論文 / 著書情報
Article / Book Information

Title	Relationship between Standard Model Plaintext Awareness and Message Hiding
Authors	Isamu Teranishi, Wakaha Ogata
Citation	IEICE Transaction, vol. E91-A, No. 1, pp. 244-261
Pub. date	2008, 1
URL	http://search.ieice.org/
Copyright	(c) 2008 Institute of Electronics, Information and Communication Engineers

IEICE TRANS. FUNDAMENTALS, VOL.E91–A, NO.1 JANUARY 2008

PAPER Special Section on Cryptography and Information Security

Relationship between Standard Model Plaintext Awareness and Message Hiding*

Isamu TERANISHI[†,††a)], ***Nonmember and*** **Wakaha OGATA**[††b)], ***Member***

SUMMARY Recently, Bellare and Palacio defined the plaintext awareness (PA-ness) in the standard model. In this paper, we study the relationship between the standard model PA-ness and the property about message hiding, that is, IND-CPA. Although these two notions seem to be independent at first glance, we show that PA-ness in the standard model implies the IND-CPA security if the encryption function is oneway. By using this result, we also showed that "PA + Oneway ⇒ IND-CCA2." We also show that the computational PA-ness notion is strictly stronger than the statistical one.

key words: *plaintext awareness, standard model*

1. Introduction

1.1 Background

The *Plaintext Awareness* (*PA-ness*) [1], [2], [4], [12] is a notion about the security of a public-key encryption scheme. Intuitively, we say that a public-key encryption scheme satisfies the PA security, if no adversary can generate a ciphertext "without knowing" the corresponding plaintext.

The PA notion is important, because the following fundamental theorem [1], [2], [4] holds: the PA-ness implies the IND-CCA2 security [13], [14], if a public-key encryption scheme is IND-CPA secure [11]. Therefore, the PA-ness is useful when we show the IND-CCA2 security of a public-key encryption scheme, such as the Fujisaki-Okamoto padded scheme [10].

The original definition [1], [4] of the PA security was formalized in the random oracle model [3] and was highly dependent on this model, although the intuitive definition mentioned above does not depend on this model.

In Asiacrypt 2004, Bellare and Palacio [2] defined the PA-ness in the standard model (that is, the non-random oracle model). Here we briefly review their definition. They define the PA notion based on the indistinguishability of two worlds, "Dec world" and "Ext world." An adversary in the Dec world can access the decryption oracle. In contrast, the same adversary in the Ext world can access an extractor, which simulates the decryption oracle. The extractor has to simulate the decryption oracle by using only data "which the adversary knows." They define the three types of the PA-ness, named *perfect/statistical/computational* PA-ness, depending on whether the Dec world and the Ext world being perfectly/statistically/computationally indistinguishable for the adversary.

They also succeeded in proving the fundamental theorem, which states that all of these plaintext awareness notions, together with the IND-CPA security, imply the IND-CCA2 security.

1.2 Our Contributions

Main Result: We study the relationship between the standard model PA-ness and the property about message hiding, that is, IND-CPA. At first glance, these two notions seem to be independent. That is, the PA-ness does not seem to imply the IND-CPA security and the IND-CPA security does not seem to imply the PA-ness.

We, however, show that all three types of the PA security (that is, the perfect, statistical, and computational PA-nesses) imply the IND-CPA security if the encryption function is oneway. Recall that the fundamental theorem that "PA + IND-CPA ⇒ IND-CCA2" holds. Therefore, combining our result with the fundamental theorem shows "PA + Oneway ⇒ IND-CCA2."

Weakening the Onewayness Assumption: We also show that we can weaken the onewayness assumption of our result "PA + Oneway ⇒ IND-CCA2." We first give our motivation for weakening it. The fundamental theorem of the PA-ness is "PA + IND-CPA ⇒ IND-CCA2" and we show that "PA + Oneway ⇒ IND-CCA2." Therefore, it seems that we succeeded in weakening the IND-CPA assumption of the fundamental theorem into the onewayness assumption. However, this is not true because the IND-CPA security does *not* imply the onewayness. (See Sect. 4 for details.) Therefore, we present a theorem "PA + XXX ⇒ IND-CCA2," such that XXX is weaker than both the onewayness and the IND-CPA security.

We prove such a theorem for the perfect and the statistical PA-nesses. That is, we first define a new security notion, *non-triviality*, which is weaker than both the onewayness and the IND-CPA security, and show the following fact:

(Perfect or Statistical) PA + NonTriv ⇒ IND-CCA2.

The *non-triviality* is a notion given by weakening the

Manuscript received March 22, 2007.
Manuscript revised July 2, 2007.

[†]The author is with NEC Corporation, Kawasaki-shi, 211-8666 Japan.
[††]The authors are with Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

*The proceedings version of this paper was presented at Asiacrypt 2006 [15].

a) E-mail: teranisi@ah.jp.nec.com
b) E-mail: wakaha@mot.titech.ac.jp

DOI: 10.1093/ietfec/e91–a.1.244

onewayness in two ways. First, the non-triviality does not ensure that the success probability of an adversary is negligible. It only ensures that the amount of "bad" ciphertext is negligible. Here the term "bad" means that an adversary can invert the ciphertext with a probability exceeding a certain constant.

Second, the way of generating a ciphertext is generalized. In the case of onewayness, the experimenter of the onewayness game selects a random plaintext M and computes a challenge ciphertext $C = \mathsf{Enc}_{\mathsf{pk}}(M)$. This means that the experimenter has to "know" the plaintext M corresponding to the challenge ciphertext C. The non-triviality notion is weakened in this point. That is, the experimenter of the non-triviality is allowed to generate an instance ciphertext in any way. Therefore, it may generate the instance ciphertext "without knowing" the corresponding plaintext, (if possible). Note that this weakening can be important in the study of the PA-ness, because the PA-ness is a notion about the knowledge of the plaintext.

We show that the non-triviality notion is in fact weaker than both the onewayness and the IND-CPA security. Therefore, we can say that our new result "(Perfect or Statistical) PA + NonTriv ⇒ IND-CCA2" is a strengthened version of both our result "PA + Oneway ⇒ IND-CCA2" and the fundamental theorem "PA + IND-CPA ⇒ IND-CCA2" for the perfect or statistical PA-ness.

Statistical PA-ness is Strictly Stronger than Computational PA-ness: We also show that the statistical PA security is strictly stronger than the computational one. In the proof of this fact, we also show some tricky aspect of the computational PA-ness. More precisely, we show that, for some public-key encryption scheme, the extractor for the computational PA-ness never extract the correct plaintext even if the public-key encryption scheme is computationally PA secure.

Recall that this phenomenon never occurs in the case of the random oracle PA-ness, because the extractor for the random oracle PA-ness can extract the correct plaintext with overwhelming probability. Therefore, we can say that the computational PA-ness is quite different from the original random oracle PA-ness.

Comparing our result with Fujisaki's result [9] about the random oracle PA-ness is interesting. In his paper, he defined the *plaintext simulatability* (PS) notion, which was a "computational variant" of the random oracle PA-ness, and showed that the PS notion was strictly weaker than the random oracle PA. Therefore, our result can be recognized as a standard model variant of Fujisaki's result [9]. By comparing his result with ours, we can say that the statistical and computational standard model PA notions are related to the random oracle PA and the PS, respectively.

1.3 Organization

Section 2 describes preliminaries. Section 3 describes the difference between the computational PA-ness and the statistical PA-ness. Section 4 describes our main result that PA-ness together with the onewayness implies the IND-CPA security. Section 5 describes that the perfect or the statistical PA-ness together with the non-triviality implies the IND-CPA security. Section 6 describes our conclusions.

2. Preliminary

2.1 Notations and Terminologies

We let $\mathbb{N}$ and $\mathbb{Z}$ denote the set of all natural numbers and that of all integers respectively. We let $a\|b$ denote the concatenation of a bit string a and b. We let ε denote both the null string and the empty list.

We abbreviate a probabilistic Turing machine to *machine*, and an expected polynomial time probabilistic Turing machine to *polytime machine*.

For a set X, "$x \leftarrow X$" means that x is chosen from X uniformly randomly. For a machine $\mathcal{A}$, "$x \leftarrow \mathcal{A}(a; r)$" means that $\mathcal{A}$ outputs x when $\mathcal{A}$ is provided with a as an input and r as a random tape. "$x \leftarrow \mathcal{A}(a)$" means that $\mathcal{A}$ outputs x when $\mathcal{A}$ is provided with a as the input and a uniformly randomly selected bit string as a random tape. "$\Pr[x_0 \leftarrow X, x_1 \leftarrow \mathcal{A}_1(x_0), \ldots, x_n \leftarrow \mathcal{A}_n(x_{n-1}) : x_n = 1] = 1/2$" means the probability that $x_n = 1$ will hold is $1/2$ if we generate x_n as follow: $x_0 \leftarrow X, x_1 \leftarrow \mathcal{A}_1(x_0), \ldots, x_n \leftarrow \mathcal{A}_n(x_{n-1})$.

For a real-valued function f, we say that f is *negligible* if, for any positive valued polynomial p, the following property holds:

$$^{\exists}\kappa_0 \in \mathbb{N}\, ^{\forall}\kappa > \kappa_0 : |f(\kappa)| < 1/p(\kappa).$$

We say that f is *non-negligible* if f is not negligible, that is, if f satisfies the following property: there exists a positive valued polynomial p such that

$$^{\forall}\kappa_0 \in \mathbb{N}\, ^{\exists}\kappa > \kappa_0 : |f(\kappa)| \geq 1/p(\kappa).$$

2.2 Public-Key Encryption Scheme

Let κ be a security parameter. Let Gen and Enc be polytime machines, and Dec be a deterministic polytime machine satisfying the following properties:

- On inputting 1^{κ}, Gen outputs a *public key* pk and a *secret key* sk.
- On inputting pk and a *message* (or *plaintext*) M, Enc outputs a *ciphertext* C.
- On inputting sk and C, Dec outputs a message or a symbol $\perp$.

Above, the *message* is an element of a set MessSp_{κ} named *message space*.

We say that the tuple $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a *(public-key) encryption scheme* if it satisfies the following *correctness* property:

$$\forall M : \Pr\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa) \\ C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M)\end{array} : M = \mathsf{Dec}_{\mathsf{sk}}(C)\right]$$

is overwhelming.

2.3 Security Notions

Definition 2.1 (Onewayness): Let κ be a security parameter, $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, and MessSp_κ be a message space of Π. Let $\mathcal{I}$ be a polytime machine named *inverter*. For a public key/secret key pair $(\mathsf{pk}, \mathsf{sk})$ and a ciphertext C, we let $\mathsf{Oneway}^{\Pi}_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C)$ denote the following experiment:

$M' \leftarrow \mathcal{I}(\mathsf{pk}, C)$
If $M' = \mathsf{Dec}_{\mathsf{sk}}(C)$, output 1.
Otherwise output 0.

We say that Π is *oneway* (against CPA attack) if for any $\mathcal{I}$ the probability

$$\Pr\left[\begin{array}{l}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), \\ M \leftarrow \mathsf{MessSp}_\kappa, \\ C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M).\end{array} : \mathsf{Oneway}^{\Pi}_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C) = 1\right]$$

is negligible.

Definition 2.2 (IND-CPA, IND-CCA2 [11], [13], [14]). Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme and O oracle. Let $\mathcal{B} = (\mathcal{B}_{\mathsf{find}}, \mathcal{B}_{\mathsf{guess}})$ be a polytime adversary,and b be a bit. For a public key pk, we let $\mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk})$ denote the following experiment:

$(M_0, M_1, \mathsf{St}) \leftarrow \mathcal{B}^{\mathcal{O}}_{\mathsf{find}}(\mathsf{pk})$,
$C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M_b)$,
$b' \leftarrow \mathcal{B}^{\mathcal{O}}_{\mathsf{guess}}(C, \mathsf{St})$,
If $b' = 1$, output 1,
Otherwise, output 0.

Above, $\mathcal{A}$ is not allowed to make query C to O and M_0 and M_1 have to be elements of the message space MessSp_κ.

We set

$$\nu^b(\kappa) = \Pr[(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa) : \mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk}) = 1].$$

Let $\varepsilon(\cdot)$ be the oracle whose oracle-answers are always the null-string. We say that Π is *IND-CPA secure*, if $|\nu^1(\kappa) - \nu^0(\kappa)|$ is negligible for any $\mathcal{B}$ where $O = \varepsilon(\cdot)$. We say that Π is *IND-CCA2 secure*, if $|\nu^1(\kappa) - \nu^0(\kappa)|$ is negligible for any $\mathcal{B}$ where $O = \mathsf{Dec}_{\mathsf{sk}}(\cdot)$.

2.4 PA-ness

We first give intuition behind the definition of the PA-ness [2]. The definition of the PA-ness is based on the indistinguishability of two worlds named the *Dec world* and the *Ext world*, and uses entities named *adversary* and *extractor*. In the Dec world, the adversary can access the decryption oracle and the encryption oracle. In contrast, the adversary in the Ext world can access the extractor and the encryption oracle. The extractor has to simulate the decryption oracle by using only data "which the adversary can see," such as the adversary's description, its random tape, and the answers from the encryption oracle.

A characteristic feature of the definition is that it has a mechanism to hide the encryption query of the adversary from the extractor. In order to hide the encryption query, an entity named *plaintext creator* is also introduced. This is an entity which makes encryption queries as the adversary's proxy. The adversary, in both the Dec and Ext worlds, does not make encryption queries directly but sends an order to the plaintext creator, to make it send a query to the encryption oracle.

The extractor is not allowed to observe the plaintext creator's random tape, although it is allowed to observe the adversary's. Hence it cannot know what queries are made to the encryption oracle. We say that an encryption scheme satisfies the standard model PA, if the Dec and Ext worlds are indistinguishable for the adversary.

We now define the (standard model) PA-ness formally:

Definition 2.3 (PA-ness [2]). Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. Let $\mathcal{A}$, $\mathcal{P}$, $\mathcal{K}$ be polytime machines that are called *adversary*, *plaintext creator*, and *extractor*, respectively. Let $\mathcal{A}(\mathsf{pk}; R_{\mathcal{A}})$ denote the execution of an algorithm $\mathcal{A}$ on inputting pk with the random coin $R_{\mathcal{A}}$. For a security parameter $\kappa \in \mathbb{N}$, we define two experiments $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A},\mathcal{P}}(\kappa)$ and $\mathsf{PA}^{\mathcal{K}}_{\Pi,\mathcal{A},\mathcal{P}}(\kappa)$, as shown in Fig. 1. In these ex-

—$\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A},\mathcal{P}}(\kappa)$—	—$\mathsf{PA}^{\mathcal{K}}_{\Pi,\mathcal{A},\mathcal{P}}(\kappa)$—
Take coins $R_{\mathcal{A}}$ and $R_{\mathcal{P}}$ for $\mathcal{A}$ and $\mathcal{P}$ randomly.	Take coins $R_{\mathcal{A}}$, $R_{\mathcal{P}}$, and $R_{\mathcal{K}}$ for $\mathcal{A}$, $\mathcal{P}$, and $\mathcal{K}$ randomly.
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$, $\mathsf{CList} \leftarrow \varepsilon$, $\mathsf{St}_{\mathcal{P}} \leftarrow \varepsilon$.	$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$, $\mathsf{CList} \leftarrow \varepsilon$, $\mathsf{St}_{\mathcal{P}} \leftarrow \varepsilon$, $\mathsf{St}_{\mathcal{K}} \leftarrow (\mathsf{pk}, R_{\mathcal{A}})$.
(Here $\mathsf{St}_{\mathcal{P}}$ is the state of $\mathcal{P}$.)	(Here $\mathsf{St}_{\mathcal{P}}$ and $\mathsf{St}_{\mathcal{K}}$ are the states of $\mathcal{P}$ and $\mathcal{K}$.)
Run $\mathcal{A}(\mathsf{pk}; R_{\mathcal{A}})$ until it halts,	Run $\mathcal{A}(\mathsf{pk}; R_{\mathcal{A}})$ until it halts,
replying to its oracle queries as follows:	replying to its oracle queries as follows:
If $\mathcal{A}$ makes query (enc, Q)	If $\mathcal{A}$ makes query (enc, Q)
$(M, \mathsf{St}_{\mathcal{P}}) \leftarrow \mathcal{P}(Q, \mathsf{St}_{\mathcal{P}}; R_{\mathcal{P}})$, $C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M)$,	$(M, \mathsf{St}_{\mathcal{P}}) \leftarrow \mathcal{P}(Q, \mathsf{St}_{\mathcal{P}}; R_{\mathcal{P}})$, $C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M)$.
$\mathsf{CList} \leftarrow \mathsf{CList}\|C$. Send C to $\mathcal{A}$ as the reply.	$\mathsf{CList} \leftarrow \mathsf{CList}\|C$. Send C to $\mathcal{A}$ as the reply.
If $\mathcal{A}$ makes query (dec, Q)	If $\mathcal{A}$ makes query (dec, Q)
$M \leftarrow \mathsf{Dec}_{\mathsf{sk}}(Q)$.	$(M, \mathsf{St}_{\mathcal{K}}) \leftarrow \mathcal{K}(Q, \mathsf{CList}, \mathsf{St}_{\mathcal{K}}; R_{\mathcal{K}})$.
Send M to $\mathcal{A}$ as the reply.	Send M to $\mathcal{A}$ as the reply.
Return an output S of $\mathcal{A}$.	Return an output S of $\mathcal{A}$.

Fig. 1 Experiments used to define PA of [2].

periments, it is required that $\mathcal{A}$ makes no query (dec, C) for which $C \in \mathsf{CList}$.

We say that the public-key encryption scheme Π is *perfectly/statistically/computationally (standard model) PA* secure if

$$\forall \mathcal{A} \exists \mathcal{K} \forall \mathcal{P} : \mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A},\mathcal{P}}(\kappa) \text{ and } \mathsf{PA}^{\mathcal{K}}_{\Pi,\mathcal{A},\mathcal{P}}(\kappa) \text{ are perfectly/statistically/computationally indistinguishable for } \kappa.$$

Note that the PA security is called the *PA2* security in [2].

Theorem 2.4 (Fundamental Theorem for the PA-ness [1], [2], [4]). *Let Π be an IND-CPA secure public-key encryption scheme. If Π is (perfect, statistical, or computational) PA secure, then Π is IND-CCA2 secure.*

3. Statistical PA is Strictly Stronger than Computational PA

In this section, we study the difference between the statistical PA-ness and the computational PA-ness and show that the former is strictly stronger than the latter.

We see that the computational PA-ness is quite different from the statistical PA-ness. The statistical PA-ness ensures that the adversary "knows" the plaintext M in the sense that the extractor succeeds in extracting the plaintext $M = \mathsf{Dec}_{\mathsf{sk}}(C)$ with overwhelming probability. In contrast, the computational PA-ness ensures that the adversary "knows" the plaintext M only in the computational sense. More precisely, the computational PA-ness ensures that an extractor outputs a plaintext M' which is computationally indistinguishable from $M = \mathsf{Dec}_{\mathsf{sk}}(C)$, but it does not ensure that $M' = M$ holds. Therefore, there may be a case where the extractor never outputs M.

We show that this case in fact occurs, if there exists at least one public-key encryption scheme satisfying the computational PA security and the IND-CPA security. That is, we construct a computationally PA secure public-key encryption scheme $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ such that no extractor can output $M = \mathsf{Dec}'_{\mathsf{sk}'}(C)$ from a ciphertext C output by some adversary.

Recall that an extractor for the statistical PA-ness has to output $M = \mathsf{Dec}'_{\mathsf{sk}'}(C)$ with overwhelming probability. This means that our encryption scheme Π' is not statistically PA secure. Therefore, the existence of such Π' shows that there is a gap between the computational PA-ness and the statistical PA-ness, in particular:

Theorem 3.1: *Suppose that there exists at least one public-key encryption scheme which satisfies both the computational PA security and the IND-CPA security. Then there exists a computationally PA secure public-key encryption which is not statistically PA secure.*

Note that the Cramer-Shoup encryption scheme [5], [6] is computationally PA secure [8] and IND-CPA secure, if the DHK assumption [2], [7] and the DDH assumption holds.

—$\mathsf{Gen}'(1^\kappa)$—
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$
Select a message M_0 randomly.
$C_0 \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M_0)$.
$\mathsf{pk}' \leftarrow (\mathsf{pk}, C_0)$, $\mathsf{sk}' \leftarrow \mathsf{sk}$.
Output $(\mathsf{pk}', \mathsf{sk}')$.

$\mathsf{Enc}'_{\mathsf{pk}'}(M) = \mathsf{Enc}_{\mathsf{pk}}(M)$, $\mathsf{Dec}'_{\mathsf{sk}'}(C) = \mathsf{Dec}_{\mathsf{sk}}(C)$.

—$\mathcal{A}'_0(\mathsf{pk}')$—
Parse pk' as (pk, C_0).
Make decryption query C_0.
Receive M'_0 as an answer.
Output (pk, C_0, M'_0).

Fig. 2 Descriptions of $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ and $\mathcal{A}'_0$.

Therefore, Theorem 3.1 in particular shows that there is a gap between the computational PA-ness and the statistical PA-ness, if these two assumptions hold.

We now construct the encryption scheme Π' such that no extractor can output $M = \mathsf{Dec}'_{\mathsf{sk}'}(C)$ from a ciphertext C output by some adversary. Let κ be a security parameter. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme which is computationally PA secure and IND-CPA secure (and therefore IND-CCA2 secure). We construct the desired public-key encryption scheme $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ by modifying Π. The key generation algorithm $\mathsf{Gen}'(1^\kappa)$ first executes $\mathsf{Gen}(1^\kappa)$ and obtains a public key/secret key pair $(\mathsf{pk}, \mathsf{sk})$ as the output. After that, it selects a message M_0 randomly and computes a ciphertext $C_0 = \mathsf{Enc}_{\mathsf{pk}}(M_0)$. Then it sets $\mathsf{pk}' = (\mathsf{pk}, C_0)$ and $\mathsf{sk}' = \mathsf{sk}$. Finally, it outputs the public key/secret key pair $(\mathsf{pk}', \mathsf{sk}')$. We also set $\mathsf{Enc}'_{\mathsf{pk}'}(M) = \mathsf{Enc}_{\mathsf{pk}}(M)$ and $\mathsf{Dec}'_{\mathsf{sk}'}(C) = \mathsf{Dec}_{\mathsf{sk}}(C)$. See Fig. 2 also for the description of Π'.

We see that Π' satisfies the desired properties. See Appendix A for the detailed proof. We first see that there exists an adversary $\mathcal{A}'_0$ such that no extractor can extract a message from a ciphertext output by $\mathcal{A}'_0$.

Our adversary $\mathcal{A}'_0$ is the one that obtains C_0 from its input $\mathsf{pk}' = (\mathsf{pk}, C_0)$, makes decryption query C_0, receives a plaintext M'_0 as an answer, and outputs (pk, C_0, M'_0). Recall that not $\mathcal{A}'_0$ but the key generation algorithm Gen' generates M_0 and C_0. Therefore, $\mathcal{A}'_0$ "does not know" the message M_0 corresponding to C_0. Since an extractor for $\mathcal{A}'_0$ is provided with only data which the adversary can see, the extractor "cannot know" $M_0 = \mathsf{Dec}'_{\mathsf{sk}'}(C_0) = \mathsf{Dec}_{\mathsf{sk}}(C_0)$ either. Therefore, no extractor can output M_0. In particular, the encryption scheme Π' is not statistically PA secure.

We next see that Π' is computationally PA secure. As mentioned above, no extractor can output M_0 itself. However, recall that an extractor for the computational PA-ness is allowed to output a plaintext M_1 which is different from M_0, although the distribution of M_1 has to be computationally indistinguishable from M_0. Therefore, we construct an extractor which can output such M_1.

Recall that an adversary "knows" neither the plaintext M_0 nor the random number r which was used in the computation of $C_0 = \mathsf{Enc}_{\mathsf{pk}}(M_0; r)$. Since Π satisfies the IND-

CPA security and the computational PA security, Π satisfies the IND-CCA2 security. Hence, the adversary cannot distinguish a randomly selected message from M_0. Therefore, an extractor can output a randomly selected message as the answer to the decryption query C_0.

4. Main Result

Our main result is the following theorem:

Theorem 4.1 (PA + Oneway ⇒ IND-CPA): *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme satisfying the onewayness. If* Π *is perfectly, statistically, or computationally PA secure, then* Π *is IND-CPA secure (and therefore IND-CCA2 secure).*

Our result shows that a PA secure scheme satisfies the very strong message hiding property, the IND-CCA2 security, or does not satisfy even the very weak message hiding property, onewayness.

Before giving the proof of Theorem 4.1, we show that we cannot remove the onewayness assumption from Theorem 4.1:

Theorem 4.2 (PA ⇏ IND-CPA): *There is a public-key encryption which satisfies perfect, statistical, and computational PA securities, but is not IND-CPA secure.*

Proof of Theorem 4.2, sketch Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme, such that a ciphertext of an message M is M itself. Then Π is clearly not IND-CPA secure. Recall the definition of the perfect PA-ness. We say that Π satisfies the perfect PA security if, for any adversary $\mathcal{A}$, there exists an extractor $\mathcal{K}$ such that $\mathcal{K}$ succeeds in extracting the plaintext M which corresponds to a ciphertext C output by $\mathcal{A}$. Since $\mathcal{K}$ can know the message M directly from the ciphertext itself, Π satisfies the perfect PA-ness. Therefore, Π satisfies the statistical and computational PA-nesses also. □

We now give the proof of Theorem 4.1. We use a similar idea to the proof of Theorem 3.1. In both proofs, an adversary is required to output a ciphertext "without knowing" the corresponding plaintext. In the proof of Theorem 3.1, the adversary obtains such a ciphertext from the public key. In the proof of Theorem 4.1 the adversary obtains such a ciphertext by "receiving" from a plaintext creator.

Sketch of the Proof of Theorem 4.1 We here consider the special case where Π is statistically PA secure. See Appendix D for the detailed proof for the general case where Π satisfies only the computational PA security.

Let us make a contradictory supposition that there exists a statistically PA secure public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ which is not IND-CPA secure. We will show that Π is not oneway.

We would like to construct an inverter $\mathcal{I}_0(\mathsf{pk}, C)$ for the onewayness game. To this end, we will construct an adversary $\mathcal{A}_0$ and a plaintext creator $\mathcal{P}_0^C$ such that $\mathcal{P}_0^C$ can "send" the ciphertext C to $\mathcal{A}_0$ in some way. $\mathcal{A}_0$ "receives" C from $\mathcal{P}_0^C$ and makes the decryption query C.

From the definition of the statistical PA-ness, there exists an extractor $\mathcal{K}_0$ for $\mathcal{A}_0$. $\mathcal{K}_0$ can output the plaintext $M = \mathsf{Dec}_{\mathsf{sk}}(C)$ with overwhelming probability. That is, $\mathcal{K}_0$ succeeds in inverting C.

Therefore, if there exists such $\mathcal{A}_0$ and ${\mathcal{P}_0}^C$, we can construct an inverter $\mathcal{I}_0(\mathsf{pk}, C)$ that inverts the ciphertext C by executing $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk})$. Here $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk})$ is the experiment $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\kappa)$ in which pk is used as a public key.

However, there is no communication channel that allows ${\mathcal{P}_0}^C$ to send the ciphertext C to $\mathcal{A}_0$. Therefore, we construct a "virtual channel" from ${\mathcal{P}_0}^C$ to $\mathcal{A}_0$. Here we exploit the assumption that Π is not IND-CPA secure. Recall that the definition of the statistical PA security allows $\mathcal{P}_0^C$ to send plaintexts to the encryption oracle. Therefore, $\mathcal{P}_0^C$ can send to $\mathcal{A}_0$ a ciphertext c such that $\mathcal{P}_0^C$ generates the corresponding plaintext.

Since Π is not IND-CPA secure, the ciphertext c leaks information of the corresponding plaintext. This means that $\mathcal{P}_0^C$ can send to $\mathcal{A}_0$ some sort of information via the ciphertext c. Therefore, $\mathcal{P}_0^C$ can use the ciphertext c as a "virtual channel." We will describe the details of how to construct a "virtual channel" and show that $\mathcal{P}_0^C$ can "send" C to $\mathcal{A}_0$ with non-negligible probability, in later subsections.

Recall that we constructed the inverter $\mathcal{I}_0(\mathsf{pk}, C)$ which inverts the ciphertext C by executing $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk})$, and $\mathcal{K}_0$ succeeds in extracting a plaintext with overwhelming probability. Therefore, $\mathcal{I}_0(\mathsf{pk}, C)$ succeeds in inverting C with overwhelming probability if $\mathcal{A}_0(\mathsf{pk})$ succeeds in "receiving" C from ${\mathcal{P}_0}^C$. Since $\mathcal{A}_0$ can "receive" C with non-negligible probability, this means that $\mathcal{I}_0(\mathsf{pk}, C)$ succeeds in inverting C with non-negligible probability. Therefore, Π is not oneway. □

4.1 How to Construct the "Virtual Channel"

Let $\mathcal{A}_0$, $\mathcal{K}_0$, and ${\mathcal{P}_0}^C$ be machines described in the sketch of the proof of Theorem 4.1. In order to complete the proof of Theorem 4.1, we describe how to construct the "virtual channel" and show that ${\mathcal{P}_0}^C$ can "send" data to $\mathcal{A}_0$ with non-negligible probability.

We here only give the procedures on how ${\mathcal{P}_0}^C$ can "send" a bit b to $\mathcal{A}_0$. ${\mathcal{P}_0}^C$ can "send" the ciphertext C by executing the procedures for each bit of C. See Appendix B for the formal description of the "virtual channel".

From the assumption, Π is not IND-CPA secure. Therefore, there exists an adversary $\mathcal{B} = (\mathcal{B}_{\mathsf{find}}, \mathcal{B}_{\mathsf{guess}})$ which has non-negligible advantage to the IND-CPA game. We use the notations in Sect. 2.3. For a public key pk, we set

$$\nu^b(\mathsf{pk}) = \Pr[\mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk}) = 1].$$

Clearly, $\nu^b(\kappa)$ is the expected value of $\nu^b(\mathsf{pk})$ when we generate pk by using $\mathsf{Gen}(1^\kappa)$.

Before "receiving" the bit b, $\mathcal{A}_0$ has to do two preparatory things. First, $\mathcal{A}_0$ guesses the value $\nu^b(\mathsf{pk})$ as follows:

$\mathcal{A}_0$ executes, N times, the experiment $\mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk})$. Let ℓ_b be the number of times that $\mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk})$ outputs 1. $\mathcal{A}_0$ guesses that $\nu^b(\mathsf{pk})$ is $\bar{\nu}^b = \ell_b/N$.

Second, $\mathcal{A}_0$ executes $\mathcal{B}_{\mathsf{find}}(\mathsf{pk})$, N times, obtains the outputs $(m_0^{(1)}, m_1^{(1)}, \mathsf{St}_{\mathcal{B}}^{(1)})$, ..., $(m_0^{(N)}, m_1^{(N)}, \mathsf{St}_{\mathcal{B}}^{(N)})$ of $\mathcal{B}$, and sends $(m_0^{(1)}, m_1^{(1)}, \mathsf{St}_{\mathcal{B}}^{(1)})$, ..., $(m_0^{(N)}, m_1^{(N)}, \mathsf{St}_{\mathcal{B}}^{(N)})$ to ${\mathcal{P}_0}^C$.

In order to "send" a bit b to $\mathcal{A}_0$, ${\mathcal{P}_0}^C$ makes query $m_b^{(1)}$, ..., $m_b^{(N)}$ to the encryption oracle. Then, the oracle sends $c_1 = \mathsf{Enc}_{\mathsf{pk}}(m_b^{(1)})$, ..., $c_N = \mathsf{Enc}_{\mathsf{pk}}(m_b^{(N)})$ to $\mathcal{A}_0$.

$\mathcal{A}_0$ "receives" the bit b as follows: $\mathcal{A}_0$ computes $b'_1 = \mathcal{B}_{\mathsf{guess}}(c_1, \mathsf{St}_{\mathcal{B}}^{(1)})$ ..., $b'_N = \mathcal{B}_{\mathsf{guess}}(c_1, \mathsf{St}_{\mathcal{B}}^{(N)})$. Let ℓ be the number of i satisfying $b'_i = 1$. It sets $b' = 1$ or $b' = 0$, depending on whether $\ell/N \geq (\bar{\nu}^0 + \bar{\nu}^1)/2$ holds or not.

4.2 Proof that the Virtual Channel Sends Messages Correctly

We use the notations in Sect. 2.3. As described in the proof of Theorem 4.1, we can construct a virtual channel if Π is *not* IND-CPA secure. Let Π be a public-key encryption scheme that is not IND-CPA secure. Then, there exists an adversary $\mathcal{B}$ such that $|\nu^1(\kappa) - \nu^0(\kappa)|$ is non-negligible.

Therefore, there exists a non-negative valued polynomial $p_0(\kappa)$ such that $|\nu^1(\kappa) - \nu^0(\kappa)| \geq 1/p_0(\kappa)$ holds for infinitely many κ.

Let $\mathcal{B}'$ be a polytime algorithm which outputs 1 or 0 if $\mathcal{B}$ outputs 0 or 1. Recall that $\nu^b(\kappa)$ the probability that $\mathcal{B}$ outputs b. By replacing $\mathcal{B}$ with $\mathcal{B}'$, if necessary, we can suppose that $\nu^1(\kappa) - \nu^0(\kappa) \geq 1/p_0(\kappa)$ holds for infinitely many κ.

Lemma 4.3 (Informal Version): *There exists a family $\{\Omega_\kappa\}_\kappa$ of sets of public keys and a polynomial $p_0(\kappa)$ such that, for infinitely many κ, the following two properties hold:*

1. *For $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$, $\mathsf{pk} \in \Omega_\kappa$ holds with probability of at least $1/p_0(\kappa)$.*
2. *For any fixed $\mathsf{pk} \in \Omega_\kappa$, $\mathcal{A}_0(\mathsf{pk})$ can "receive" C from ${\mathcal{P}_0}^C$ with the probability of more than $4/5$.*

In particular, the bit b' which $\mathcal{A}_0$ "receives" is equal to the bit b which ${\mathcal{P}_0}^C$ "sends" with non-negligible probability.

See Appendix B for the formal description of Lemma 4.3 and the formal proof for it. We will use Lemma 4.3 in Section 5 also.

Idea of Proof of Lemma 4.3 Let $\mu(\kappa)$ and $\mu(\mathsf{pk})$ be $\nu^1(\kappa) - \nu^0(\kappa)$ and $\nu^1(\mathsf{pk}) - \nu^0(\mathsf{pk})$. As mentioned before, the inequality $\mu(\kappa) = \nu^1(\kappa) - \nu^0(\kappa) \geq 1/p_0(\kappa)$ holds for infinitely many κ. We restrict our discussion for κ satisfying this inequality.

Since $\mu(\kappa) = \nu^1(\kappa) - \nu^0(\kappa) \geq 1/p_0(\kappa)$ holds, $\nu^1(\kappa) \geq \nu^0(\kappa)$ holds.

We fix a value κ, and set $\Omega_\kappa = \{\mathsf{pk} \text{ s.t. } \mu(\mathsf{pk}) \geq \mu(\kappa)/2\}$. From the definition of $\mu(\kappa)$ and $\mu(\mathsf{pk})$, $\mu(\kappa)$ is equal to the expected value of $\mu(\mathsf{pk})$ when we take $(\mathsf{pk}, \mathsf{sk})$ according to $\mathsf{Gen}(1^\kappa)$. Therefore, many pk satisfy $\mu(\mathsf{pk}) \simeq \mu(\kappa)$. So, many pk satisfy $\mu(\mathsf{pk}) \geq \mu(\kappa)/2$. By using this fact, we can show that

$$\Pr[\mathsf{pk} \in \Omega_\kappa] \geq \frac{\mu(\kappa)}{2} \geq 1/p_0(\kappa).$$

That is, the first inequation of Lemma 4.3 holds.

Let n be the bit length of the ciphertext C. We will show that $\mathcal{A}_0(\mathsf{pk})$ can "receive" the correct bit with the probability more than $1 - (n/5)$, for any $\mathsf{pk} \in \Omega_\kappa$. Then, since C is an n bit string, $\mathcal{A}_0(\mathsf{pk})$ can receive C with probability $(1 - (n/5))^n \geq 1 - (1/5) = 4/5$. That is, the lemma holds.

We now show that $\mathcal{A}_0(\mathsf{pk})$ can "receive" the correct bit with the probability $1 - (n/5)$. Suppose that ${\mathcal{P}_0}^C$ "sends" a bit $b = 1$. Then $c_i = \mathsf{Enc}_{\mathsf{pk}}(m_1^{(i)})$ and $b'_i = \mathcal{B}_{\mathsf{guess}}(c_i, \mathsf{St}_{\mathcal{B}}^{(i)})$ are computed by the encryption oracle and $\mathcal{A}_0$ respectively. From the definition, the probability that $b'_i = 1$ holds with probability $\nu^1(\mathsf{pk})$. Therefore, the mean value $\ell/N = \sum_i b'_i/N$ satisfies $\ell/N \simeq \nu^1(\mathsf{pk})$ if N is a large number. Recall that $\bar{\nu}^0$ and $\bar{\nu}^1$ are statistical inferences of $\nu^0(\mathsf{pk})$ and $\nu^1(\mathsf{pk})$. That is, $\bar{\nu}^0 \simeq \nu^0(\mathsf{pk})$ and $\bar{\nu}^1 \simeq \nu^1(\mathsf{pk})$ hold if N is a large number.

Recall that $\nu^1(\kappa) - \nu^0(\kappa) \geq 1/p_0(\kappa) \geq 0$ holds. Hence, for any $\mathsf{pk} \in \Omega_\kappa$, it follows that $\nu^1(\mathsf{pk}) - \nu^0(\mathsf{pk}) = \mu(\mathsf{pk}) \geq \mu(\kappa)/2 = (\nu^1(\kappa) - \nu^0(\kappa))/2 \geq 0$. Therefore, $\nu^1(\mathsf{pk}) \geq \nu^0(\mathsf{pk})$ holds.

Hence, it follows that $\ell/N \simeq \nu^1(\mathsf{pk}) \geq (\nu^0(\mathsf{pk}) + \nu^1(\mathsf{pk}))/2 \simeq (\bar{\nu}^0 + \bar{\nu}^1)/2$. This means that the bit b' "received" by $\mathcal{A}_0$ is 1. That is, $b' = 1 = b$ holds with high probability, if N is large and $\mathsf{pk} \in \Omega_\kappa$ holds. By taking sufficiently large N, we can show that $b' = 1 = b$ holds with high probability $1 - (n/5)$.

If ${\mathcal{P}_0}^C$ "sends" $b = 0$ (and if $\mathsf{pk} \in \Omega_\kappa$ holds also), we can show that $\ell/N \simeq \nu^0(\mathsf{pk}) \leq (\nu^0(\mathsf{pk}) + \nu^1(\mathsf{pk}))/2 \simeq (\bar{\nu}^0 + \bar{\nu}^1)/2$ via a similar discussion. Therefore, $b' = b$ holds with high probability $1 - (n/5)$, even if N is sufficiently large and ${\mathcal{P}_0}^C$ "sends" $b = 0$. □

5. Weakening the Onewayness Assumption

5.1 Preparation

In this section, we show that we can weaken the onewayness assumption of Theorem 4.1, if Π is perfectly or statistically PA secure.

Before showing this, we describe our motivation for weakening it. Recall that our main theorem, Theorem 4.1, shows that "PA + Oneway ⇒ IND-CCA2" holds. Therefore, it seems that we succeed in strengthening the fundamental theorem of the PA-ness, "PA + IND-CPA ⇒ IND-CCA2." However, this is not true. In fact, Theorem 4.1 does not imply the fundamental theorem, because the IND-CPA security does not imply the onewayness:

Proposition 5.1 (IND-CPA ⇏ Oneway): *Suppose the existence of an IND-CPA secure public-key encryption scheme. Then, there exists a public-key encryption scheme which satisfies the IND-CPA security but does not satisfy the onewayness.*

Therefore, we would like to weaken the onewayness assumption of Theorem 4.1 and show a stronger variant of the main theorem, "PA + XXX ⇒ IND-CCA2," such that XXX is weaker than both the onewayness and the IND-CPA security.

Proof of Proposition 5.1 Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure public-key encryption scheme, and MessSp_κ be the message space of Π. If MessSp_κ has only one element, clearly Π is not IND-CPA secure. Therefore, MessSp_κ has at least two elements, M and M'.

Let Π' is the encryption scheme that is obtained from Π by restricting the message space to $\{M, M'\}$. Clearly the Π' is also IND-CPA secure.

However, we can construct an inverter $\mathcal{I}$ for the onewayness game of Π'. Here $\mathcal{I}$ is an inverter which randomly outputs M or M'. Since the message space of Π' is $\{M, M'\}$, an instance C of the onewayness game is an encryption of M or M'. Hence, the probability that $\mathcal{I}$ outputs $\mathsf{Dec}_{\mathsf{sk}}(C)$ is $1/2$, and therefore is non-negligible. This means that Π' is not oneway. □

We now weaken the onewayness assumption of Theorem 4.1. More precisely, we give the new security notion *non-triviality*, show that the non-triviality is weaker than both the onewayness and the IND-CPA security, and show the following fact:

(Perfect or Statistical) PA + NonTriv ⇒ IND-CCA2

The *non-triviality* is a notion given by weakening the onewayness. It is weaker than the onewayness in two ways. First, it does not ensure that the success probability of an inverter is negligible. It only ensures that the amount of "bad" instances $(\mathsf{pk}, \mathsf{sk}, C)$ is negligible. Here the term "bad" means that an inverter $\mathcal{I}(\mathsf{pk}, C)$ can invert C with a probability exceeding a certain constant.

Second, the way to generating a ciphertext is generalized. In the case of the onewayness, the experimenter of the onewayness game selects a random message M and computes a challenge ciphertext $C = \mathsf{Enc}_{\mathsf{pk}}(M)$. This means that the experimenter has to "know" the plaintext M corresponding to the challenge ciphertext C. The non-triviality notion is weakened in this point. That is, the experimenter of the non-triviality is allowed to generate an instance ciphertext in any way. Therefore, it may generate the instance ciphertext "without knowing" the corresponding plaintext (if possible). Note that this weakening can be important in the study of the PA-ness, because the PA-ness is a notion about the knowledge of the plaintext.

Definition 5.2 (Non-Triviality): Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. Let $\mathcal{I}$ and C be polytime machines which are respectively called *inverter* and *ciphertext generator*. For a public key/secret key pair $(\mathsf{pk}, \mathsf{sk})$ and a ciphertext C, we define $\mathsf{Oneway}^{\Pi}_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C)$ as in Definition 2.1, and we let $E_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C)$ be an event that

$$\Pr[\mathsf{Oneway}^{\Pi}_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C) = 1] \geq 2/3$$

holds. Here probability is taken over the random tape of $\mathcal{I}$.

We say that Π is *non-trivial*, if, for any $\mathcal{I}$, there exists C such that

$$\Pr\left[\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), \\ C \leftarrow C(\mathsf{pk}).\end{array} : E_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C)\right]$$

is negligible.

We should note the following points about the above definition. First, the ciphertext generator C is allowed to output a ciphertext C such that $\mathsf{Dec}_{\mathsf{sk}}(C) = \perp$. In the case where C outputs such a ciphertext, the inverter $\mathcal{I}$ has to output $\perp$ in order to win in the experiment.

Second, we can replace the constant $2/3$ with any other constant c satisfying $0 < c < 1$. Even if we define the c-non-triviality by using not $\Pr[\mathsf{Oneway}^{\Pi}_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C) = 1] \geq 2/3$ but $\Pr[\mathsf{Oneway}^{\Pi}_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C) = 1] \geq c$, the c-non-triviality notion is equivalent to the original non-triviality notion of Definition 5.2. This is because an adversary can raise its success probability by executing the same algorithm many times.

We now show that the non-triviality notion satisfies the desired property:

Proposition 5.3 (Oneway ⇒ NonTriv, IND-CPA ⇒ Non Triv) *The onewayness implies the non-triviality. Moreover, the IND-CPA security also implies the non-triviality.*

Proof

(**Oneway ⇒ NonTriv**) Suppose that a public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is not non-trivial. Then there exists an inverter $\mathcal{I}$ such that, for any ciphertext generator C, $\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), C \leftarrow C(\mathsf{pk}) : E_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C)]$ is non-negligible.

We let $C_0(\mathsf{pk})$ denote the ciphertext generator which generates a ciphertext C as follow: select a message M randomly, and set $C = \mathsf{Enc}_{\mathsf{pk}}(M)$. Then $\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), C \leftarrow C_0(\mathsf{pk}) : E_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C)]$ is non-negligible, in particular. That is, $\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), M \leftarrow (\mathrm{rand}), C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M) : E_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C)]$ is non-negligible.

Therefore, if we generate $(\mathsf{pk}, \mathsf{sk}, C)$ by using the same way as that used in the onewayness game, the event $E_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C)$ holds with non-negligible probability. This means that $\mathcal{I}$ can invert C with probability $2/3$ if $(\mathsf{pk}, \mathsf{sk}, C)$ is selected in this way. Therefore, $\mathcal{I}$ can win the onewayness game with non-negligible probability.

(**IND-CPA ⇒ NonTriv**) Suppose that a public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is not non-trivial. Then there exists an inverter $\mathcal{I}$ such that, for any ciphertext generator C, $\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), C \leftarrow C(\mathsf{pk}) : E_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C)]$ is non-negligible.

By using $\mathcal{I}$ as a subroutine, we construct an adversary $\mathcal{B}$ against the IND-CPA security. $\mathcal{B}(\mathsf{pk})$ selects two ciphertexts M_0 and M_1 randomly and sends (M_0, M_1) to the experimenter of the IND-CPA security. Then $\mathcal{B}$ receives a ciphertext $C = \mathsf{Enc}_{\mathsf{pk}}(M_b)$ and then executes $\mathcal{I}(\mathsf{pk}, C)$. $\mathcal{I}(\mathsf{pk}, C)$ outputs a message M (or $\perp$). If the output M is equal to $M_{b'}$ for some $b' \in \{0, 1\}$, $\mathcal{B}$ outputs b'. Otherwise, $\mathcal{B}$ selects a bit b' randomly and outputs b'.

—$\mathsf{Gen}(1^\kappa)$—
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{gen}(1^\kappa)$. Output $(\mathsf{pk}, \mathsf{sk})$.
—$\mathsf{Enc}_{\mathsf{pk}}(M)$—
(Here $M \in \{0,1\} \times X_\kappa$).
Parse M as $b\|M'$.
If $b = 0$, set $C = b\|M'$.
Otherwise, set $C = b\|f_{\mathsf{pk}}(M')$.
Output C.
—$\mathsf{Dec}_{\mathsf{sk}}(C)$—
Parse C as $b\|C'$.
If $b = 0$, output C'.
Otherwise, output $f_{\mathsf{sk}}^{-1}(C')$.

Fig. 3 Non-trivial scheme which is not oneway or IND-CPA.

We let $C_0(\mathsf{pk})$ be a ciphertext generator which generates a ciphertext C as follows: select two ciphertexts M_0 and M_1 randomly and a random bit b, and sets $C = \mathsf{Enc}_{\mathsf{pk}}(M_b)$.

Since Π is not non-trivial, for a public key/secret key pair $(\mathsf{pk}, \mathsf{sk})$ generated by $\mathsf{Gen}(1^\kappa)$ and for a ciphertext C generated by $C_0(\mathsf{pk})$, the event $E(\mathsf{pk}, \mathsf{sk}, C)$ occurs with non-negligible probability. Let $\mu(\kappa)$ be the (non-negligible) probability that the event $E(\mathsf{pk}, \mathsf{sk}, C)$ occurs. From the definition of $\mathcal{B}$, if the event $E(\mathsf{pk}, \mathsf{sk}, C)$ occurs, the output b' of $\mathcal{B}$ is equal to b with probability $2/3$. Even if the event $E(\mathsf{pk}, \mathsf{sk}, C)$ does not occur, the output b' of $\mathcal{B}$ is equal to b with probability $1/2$. Therefore, $b' = b$ holds with probability $\frac{2}{3}\mu(\kappa) + \frac{1}{2}(1 - \mu(\kappa)) = \frac{1}{2} + \frac{1}{6}\mu(\kappa)$. Since $\mu(\kappa)$ is non-negligible, the advantage of $\mathcal{B}$ is non-negligible. □

We finally give an example of a non-trivial public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ which satisfies neither the onewayness nor the the IND-CPA security. Let X_κ be a set, $f_{\mathsf{pk}} : X_\kappa \to X_\kappa$ be a trapdoor oneway permutation, and gen be the key generator of f. We construct the public-key encryption scheme Π as in Fig. 3.

Proposition 5.4 *The public-key encryption scheme Π depicted in Fig. 3 satisfies the non-triviality but does not satisfy onewayness or the IND-CPA security.*

Proof. (idea) From the definition of Enc, if we select a message $M = b\|M'$ randomly, $\mathsf{Enc}_{\mathsf{pk}}(M) = M$ holds with the probability $1/2$. We consider the inverter such that, by giving a ciphertext C, it outputs C itself. Clearly, the inverter succeeds in inverting the ciphertext with the probability at least $1/2$. This means that Π is not oneway.

Moreover, since $\mathsf{Enc}_{\mathsf{pk}}$ is a deterministic function, $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is not IND-CPA secure.

We finally show that Π is non-trivial. Let $\mathcal{I}$ be an arbitrary inverter. Let $C(\mathsf{pk})$ be a ciphertext creator which selects $M' \in X_\kappa$ randomly, computes $C = \mathsf{Enc}_{\mathsf{pk}}(1\|M') = 1\|f_{\mathsf{pk}}(M')$, and outputs C. Since f is oneway, $\mathcal{I}$ succeeds in inverting C with only negligible probability. That is,

$$\Pr\left[\begin{array}{l|l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), & \\ C \leftarrow C(\mathsf{pk}) & : \mathsf{Oneway}_{\mathcal{I}}^{\Pi}(\mathsf{pk}, \mathsf{sk}, C) = 1 \end{array}\right]$$

is negligible.

For any $(\mathsf{pk}_0, \mathsf{sk}_0, C_0)$, we let $P_{\mathsf{pk}_0,\mathsf{sk}_0,C_0}$ be

$$\Pr[\mathsf{Oneway}_{\mathcal{I}}^{\Pi}(\mathsf{pk}_0, \mathsf{sk}_0, C_0) = 1],$$

and $Q_{\mathsf{pk}_0,\mathsf{sk}_0,C_0}$ be

$$\Pr\left[\begin{array}{l|l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), & \\ C \leftarrow C(\mathsf{pk}) & : \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) = (\mathsf{pk}_0, \mathsf{sk}_0), \\ C = C_0 \end{array} \end{array}\right].$$

Then it follows that

$$\begin{aligned} &\Pr\left[\begin{array}{l|l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), & \\ C \leftarrow C(\mathsf{pk}) & : \mathsf{Oneway}_{\mathcal{I}}^{\Pi}(\mathsf{pk}, \mathsf{sk}, C) = 1 \end{array}\right] \\ &= \sum_{\mathsf{pk}_0,\mathsf{sk}_0,C_0} P_{\mathsf{pk}_0,\mathsf{sk}_0,C_0} Q_{\mathsf{pk}_0,\mathsf{sk}_0,C_0} \\ &\geq \sum_{\mathsf{pk}_0,\mathsf{sk}_0,C_0 \text{ s.t. } E_{\mathcal{I}}(\mathsf{pk}_0,\mathsf{sk}_0,C_0) \text{ holds.}} P_{\mathsf{pk}_0,\mathsf{sk}_0,C_0} Q_{\mathsf{pk}_0,\mathsf{sk}_0,C_0} \\ &\geq \sum_{\mathsf{pk}_0,\mathsf{sk}_0,C_0 \text{ s.t. } E_{\mathcal{I}}(\mathsf{pk}_0,\mathsf{sk}_0,C_0) \text{ holds.}} (2/3) Q_{\mathsf{pk}_0,\mathsf{sk}_0,C_0} \\ &= \frac{2}{3} \Pr\left[\begin{array}{l|l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), & \\ C \leftarrow C(\mathsf{pk}). & : E_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C) \end{array}\right]. \end{aligned}$$

Therefore,

$$\Pr\left[\begin{array}{l|l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), & \\ C \leftarrow C(\mathsf{pk}). & : E_{\mathcal{I}}(\mathsf{pk}, \mathsf{sk}, C) \end{array}\right]$$

is also negligible. That is, Π is non-trivial. □

5.2 Result

We now state our theorem formally:

Theorem 5.5 (Perfect or Statistical) PA + Non Triv ⇒ IND-CPA *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme satisfying the non-triviality. If Π is perfectly or statistically PA secure, then Π is IND-CPA secure (and therefore IND-CCA2 secure).*

The proof for Theorem 5.5 is similar to the sketch of the proof of Theorem 4.1, described in Section 4. See Appendix C for the detailed proof.

Proof. We take $\mathcal{A}_0, \mathcal{K}_0, {\mathcal{P}_0}^C, \mathcal{I}_0$ as in the sketch of the proof of Theorem 4.1. From Lemma 4.3, there exists a family $\{\Omega_\kappa\}$ of sets of public keys and a polynomial $p_0(\kappa)$ such that, for infinitely many κ, the following two properties hold:

1. For $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$, $\mathsf{pk} \in \Omega_\kappa$ holds with probability of at least $1/p_0(\kappa)$.
2. For any fixed $\mathsf{pk} \in \Omega_\kappa$, $\mathcal{A}_0(\mathsf{pk})$ can "receive" C from ${\mathcal{P}_0}^C$ with the probability of more than $4/5$.

We restrict our discussion for κ satisfying the above property.

Recall that we constructed the inverter $\mathcal{I}_0(\mathsf{pk}, C)$ which inverts the ciphertext C by executing $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,{\mathcal{P}_0}^C}(\mathsf{pk})$. $\mathcal{K}_0$ succeeds in extracting a plaintext with overwhelming probability. Therefore, $\mathcal{I}_0(\mathsf{pk}, C)$ succeeds in inverting C with overwhelming probability if $\mathcal{A}_0(\mathsf{pk})$ succeeds in "receiving" C from ${\mathcal{P}_0}^C$. In particular, $\mathcal{I}_0(\mathsf{pk}, C)$ succeeds in inverting C with probability of at least $4/5$ if $\mathsf{pk} \in \Omega_\kappa$ holds. Hence, for a public key $\mathsf{pk} \in \Omega_\kappa$ and the corresponding secret key sk,

the event $E_{I_0}(\mathsf{pk}, \mathsf{sk}, C)$ occurs for any C. Therefore, for any $\mathsf{pk} \in \Omega_\kappa$ and the corresponding sk, for any ciphertext generator C, and for a ciphertext C generated by C, the event $E_{I_0}(\mathsf{pk}, \mathsf{sk}, C)$ occurs.

Since $\mathsf{pk} \in \Omega_\kappa$ holds with probability of at least $1/p_0(\kappa)$, for any ciphertext generator C, $\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), C \leftarrow C(\mathsf{pk}) : E_{I_0}(\mathsf{pk}, \mathsf{sk}, C)]$ is more than $1/p_0(\kappa)$.That is, $\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), C \leftarrow C(\mathsf{pk}) : E_{I_0}(\mathsf{pk}, \mathsf{sk}, C)]$ is non-negligible. This means that Π is not non-trivial. □

6. Conclusion

In this paper, we studied the relationship between the standard model PA-ness and the property of message hiding, that is, IND-CPA. Although these two notions seem to be independent at first glance, we showed that all three types of PA-ness (that is, the perfect, statistical, and computational PA-ness) imply the IND-CPA security if the encryption function is oneway. We also showed that we can weaken the onewayness assumption of the above result to a new weak assumption, named non-triviality, if a public-key encryption scheme is perfectly or statistically PA secure.

Combining this result with the fundamental theorem implies the stronger variant of the fundamental theorem, "(perfect, statistical or computational) PA + NonTriv ⇒ IND-CCA2."

We also showed that the computational PA notion is strictly stronger than the statistical one. In particular, we showed a tricky aspect of the computational PA-ness. By comparing Fujisaki's result [9] with our result, we can say that statistical and computational standard model PA notions are related to the random oracle PA and the plaintext simulatability [9], respectively.

Acknowledgment

We thank Goichiro Hanaoka for his helpful comments.

References

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," CRYPTO 1998, pp.26–45, 1998.
[2] M. Bellare and A. Palacio, "Towards plaintext-aware public-key encryption without random oracles," ASIACRYPT 2004, pp.48–62, 2004.
[3] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," ACM-CCS 1993, pp.62–73, 1993.
[4] M. Bellare and P. Rogaway, "Optimal asymmetric encryption," EUROCRYPT 1994, pp.92–111, 1994.
[5] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," CRYPTO 1998, pp.13–25, 1998.
[6] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes," manuscript, 2001. Full version: SIAM J. Comput., vol.33, no.1, pp.167–226, 2004.
[7] I. Damgård, "Towards practical public key systems secure against chosen ciphertext attacks," CRYPTO'91, pp.445–456, 1991.
[8] A.W. Dent, "Cramer-shoup is plaintext-aware in the standard model," EUROCRYPT 2006.
[9] E. Fujisaki, "Plaintext simulatability," IEICE Trans. Fundamentals, vol.E89-A, no.1, pp.55–65, Jan. 2006. doi:10.1093/ietfec/e89-a.1.55. Preliminary version is available at http://eprint.iacr.org/2004/218.pdf
[10] E. Fujisaki and T. Okamoto, "How to enhance the security of public-key encryption at minimum cost," PKC'99, pp.53–68, 1999.
[11] S. Goldwasser and S. Micali. "Probabilistic encryption," J. Comput. Syst. Sci., vol.28, no.2, pp.270–299, 1984.
[12] J. Herzog, M. Liskov, and S. Micali, "Plaintext awareness via key registration," CRYPTO 2003, pp.548–564, 2003.
[13] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," STOC 1990, pp.427–437, 1990.
[14] C. Rackoff and D.R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," CRYPTO 1991, pp.433–444, 1991.
[15] I. Teranishi and W. Ogata, Extended abstract version of this paper, ASIACRYPT 2006.

Appendix A: Detailed Proof of Theorem 3.1

Proof. Let Π' be the public-key encryption scheme described in Fig. 2. We have to show that Π' is not statistically PA secure and Π' is computationally PA secure.

(Π' is Not Statistically PA Secure.) Let us make a contradictory supposition that Π' is statistically PA secure. Then we show that the original encryption scheme Π is not IND-CPA secure.

Let $\mathcal{A}'_0$ be the adversary depicted in Fig. 2. Since Π' is statistically PA secure, there exists an extractor $\mathcal{K}'_0$ for $\mathcal{A}'_0$ such that, for any $\mathcal{P}$, the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_0,\mathcal{P}}(\kappa)$ and that of $\mathsf{PA}^{\mathcal{K}'_0}_{\Pi',\mathcal{A}'_0,\mathcal{P}}(\kappa)$ have statistically indistinguishable distributions.

We fix an arbitrary plaintext creator $\mathcal{P}_0$. For a public key pk', we let $\mathsf{PA}^{\mathcal{K}'_0}_{\Pi',\mathcal{A}'_0,\mathcal{P}_0}(\mathsf{pk}')$ be the experiment $\mathsf{PA}^{\mathcal{K}'_0}_{\Pi',\mathcal{A}'_0,\mathcal{P}_0}(\kappa)$ in which pk' is used as a public key.

By using $\mathsf{PA}^{\mathcal{K}'_0}_{\Pi',\mathcal{A}'_0,\mathcal{P}_0}(\mathsf{pk}')$ as a subroutine, we construct an adversary $\mathcal{B}$ against the IND-CPA security of Π. $\mathcal{B}(\mathsf{pk})$ randomly selects two messages M_0, M_1, and makes query (M_0, M_1) to the challenge oracle for the IND-CPA security of Π. Then $\mathcal{B}$ obtains the challenge ciphertext $C = \mathsf{Enc}_{\mathsf{pk}}(M_b)$ for some $b = 0, 1$. $\mathcal{B}$ sets $\mathsf{pk}' = (\mathsf{pk}, C)$, executes $\mathsf{PA}^{\mathcal{K}'_0}_{\Pi',\mathcal{A}'_0,\mathcal{P}_0}(\mathsf{pk}')$ and obtains the output (pk', C, M') of $\mathsf{PA}^{\mathcal{K}'_0}_{\Pi',\mathcal{A}'_0,\mathcal{P}_0}(\mathsf{pk}')$. Since $\mathcal{K}'_0$ is an extractor for the statistical PA-ness, the output M' of $\mathcal{K}'_0$ is equal to $\mathsf{Dec}_{\mathsf{sk}}(C)$ with overwhelming probability. $\mathcal{B}$ compares $\mathsf{Dec}_{\mathsf{sk}}(C)$ with M_0 and M_1. If $\mathcal{B}$ outputs $b' = 0$ or $b' = 1$ depending on whether $M_1 = \mathsf{Dec}_{\mathsf{sk}}(C)$ holds or not. This means that $\mathcal{B}$ has a non-negligible advantage in the IND-CPA game. Therefore, Π is not IND-CPA secure.

(Π' is Computational PA Secure.) Let $\mathcal{A}'_1$ be an arbitrary adversary for Π'. In order to prove the PA security of Π', we have to construct an extractor $\mathcal{K}'_1$ for $\mathcal{A}'_1$. To this end,

—$\mathcal{A}_1(\mathsf{pk}; \mathcal{R}_{\mathcal{A}_1})$—
Parse $\mathcal{R}_{\mathcal{A}_1}$ as $(R, \mathcal{R}_{\mathcal{A}'_1})$.
Send a special symbol"GenM0" to a plaintext creator as a query.
Receive a ciphertext C_0 as an answer from the query from the encryption oracle.
$\mathsf{pk}' \leftarrow (\mathsf{pk}, C_0)$.
Run $\mathcal{A}'_1(\mathsf{pk}'; R_{\mathcal{A}'_1})$ until it halts, replying to its oracle as follows:
  If $\mathcal{A}'_1$ makes query (enc, Q), then
    Pass this query directly on to the plaintext creator.
    Receive the answer from the encryption oracle, and send it to $\mathcal{A}'_1$ as the reply.
  If $\mathcal{A}'_1$ makes query (dec, C_0) then
    By using the random coins R, select M_1 randomly. Send M_1 to $\mathcal{A}'_1$ as the reply.
  If $\mathcal{A}'_1$ makes query (dec, C) for $C \neq C_0$, then
    Pass this query directly on to the decryption oracle for $\mathcal{A}_1$.
    Receive the answer M from the oracle, and send M to $\mathcal{A}'_1$ as the reply.
Return the output S of $\mathcal{A}'_1$.

—$\mathcal{K}'_1(Q, \mathsf{CList}, \mathsf{St}_{\mathcal{K}'_1}; \mathcal{R}_{\mathcal{K}'_1})$—
Parse $R_{\mathcal{K}'_1}$ and $\mathsf{St}_{\mathcal{K}'_1}$ as $(R, R_{\mathcal{K}_1})$ and $(\mathsf{pk}', R_{\mathcal{A}'_1}, \mathsf{St}_{\mathcal{K}_1})$. (Initially, $\mathsf{St}_{\mathcal{K}_1} = \varepsilon$.)
Parse pk' as (pk, C_0).
If $\mathsf{St}_{\mathcal{K}_1} = \varepsilon$ then set $\mathsf{St}_{\mathcal{K}_1} = (\mathsf{pk}, (R, R_{\mathcal{A}'_1}))$.
If $Q = (\mathsf{dec}, C_0)$ then
  By using random coins R, select M_1 randomly. Return $(M_1, \mathsf{St}_{\mathcal{K}'_1})$.
Otherwise
  $(M, \mathsf{St}_{\mathcal{K}_1}) \leftarrow \mathcal{K}_1(Q, \mathsf{CList}, \mathsf{St}_{\mathcal{K}_1}; R_{\mathcal{K}_1})$, $\mathsf{St}_{\mathcal{K}'_1} \leftarrow (\mathsf{pk}', R_{\mathcal{A}'_1}, \mathsf{St}_{\mathcal{K}_1})$. Return $(M, \mathsf{St}_{\mathcal{K}'_1})$.

—$\mathcal{P}_1(Q, \mathsf{St}_{\mathcal{P}_1}; R_{\mathcal{P}_1})$—
Parse $R_{\mathcal{P}_1}$ as $(R, R_{\mathcal{P}'})$. $\mathsf{St}_{\mathcal{P}'} \leftarrow \mathsf{St}_{\mathcal{P}_1}$.
If $Q = \mathsf{GenM0}$ then
  By using random coins R, select a message M_0 randomly. Return $(M_0, \mathsf{St}_{\mathcal{P}_1})$.
Otherwise
  $(M, \mathsf{St}_{\mathcal{P}'}) \leftarrow \mathcal{P}'(Q, \mathsf{St}_{\mathcal{P}'}; R_{\mathcal{P}'})$. $\mathsf{St}_{\mathcal{P}_1} \leftarrow \mathsf{St}_{\mathcal{P}'}$. Return $(M, \mathsf{St}_{\mathcal{P}_1})$.

—$\mathcal{E}(\mathsf{pk}_0; R_{\mathcal{E}})$—
Parse $R_{\mathcal{E}}$ as $(R, R_{\mathcal{A}'_1})$.
By using the random coins R, select messages M_0 and M_1 randomly.
Make query (M_0, M_1) to $O_{\mathsf{enc}}(b, \mathsf{pk}_0, \cdot)$, and receive the answer $C_0 = \mathsf{Enc}_{\mathsf{pk}_0}(M_b)$.
Set $\mathsf{pk}' = (\mathsf{pk}_0, C_0)$.
Run $\mathcal{A}'_1(\mathsf{pk}'; R_{\mathcal{A}'_1})$ until it halts, replying to its oracle as follows:
  If $\mathcal{A}'_1$ makes a query (enc, Q) then
    Send the query Q to $\mathcal{P}'$, and receive the answer M.
    Compute $\mathsf{Enc}_{\mathsf{pk}_0}(M)$ and send it to $\mathcal{A}'_1$ as the reply.
  If $\mathcal{A}'_1$ makes a query (dec, C_0) then
    Send M_0 to $\mathcal{A}'_1$ as the reply.
  If $\mathcal{A}'_1$ makes a query (dec, C) such that $C \neq C_0$, then
    Pass this query directly on to its own decryption oracle $O_{\mathsf{dec}}(\mathsf{sk}_0, \cdot)$.
    Receive the answer, and send it to $\mathcal{A}'_1$.
$S \leftarrow$ (the output of $\mathcal{A}'_1$), return $\mathcal{D}_1(S)$.

Fig. A· 1 Descriptions of $\mathcal{A}_1(\mathsf{pk}; \mathcal{R}_{\mathcal{A}_1})$, $\mathcal{K}'_1(Q, \mathsf{CList}, \mathsf{St}_{\mathcal{K}'_1}; \mathcal{R}_{\mathcal{K}'_1})$, $\mathcal{P}_1(Q, \mathsf{St}_{\mathcal{P}_1}; R_{\mathcal{P}_1})$, and $\mathcal{E}(\mathsf{pk}_0; R_{\mathcal{E}})$.

we consider the adversary $\mathcal{A}_1$ described in Fig. A· 1. $\mathcal{A}_1$ uses $\mathcal{A}'_1$ to attack Π. Since Π satisfies the PA security, there exists an extractor $\mathcal{K}_1$ for $\mathcal{A}_1$ such that

$$\forall \mathcal{P} : \text{The distribution of the output of } \mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}}(\kappa) \text{ and that of the output of } \mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}}(\kappa) \text{ are computationally indistinguishable.} \quad (\text{A}\cdot 1)$$

Now, by using $\mathcal{K}_1$, we construct the extractor $\mathcal{K}'_1$ for $\mathcal{A}'_1$, described in Fig. A· 1. We will show that this $\mathcal{K}'_1$ is a desirable extractor for $\mathcal{A}'_1$, that is, $\mathsf{PA}^{\mathcal{K}'_1}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$ is computationally indistinguishable from $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$ for any plaintext creator $\mathcal{P}'$.

In order to show this, we construct a plaintext creator $\mathcal{P}_1$ for Π by using $\mathcal{P}'$. The description of $\mathcal{P}_1$ is presented in Fig. A· 1. From the definition of algorithms $\mathcal{A}_1$, $\mathcal{K}_1$, $\mathcal{K}'_1$, and $\mathcal{P}_1$, the distribution of the output of $\mathsf{PA}^{\mathcal{K}'_1}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$ is equal to that of the output of $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$. See Fig. A· 2.

Recall that the statement (A· 1) is satisfied. In particular, the distribution of the output of $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ and that of the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ are computationally indistinguishable from each other.

Furthermore, we show that the distribution of the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ and that of the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$ are computationally indistinguishable, by using the IND-CCA2 security of Π:

Lemma Appendix A.1: *The distribution of the output of*

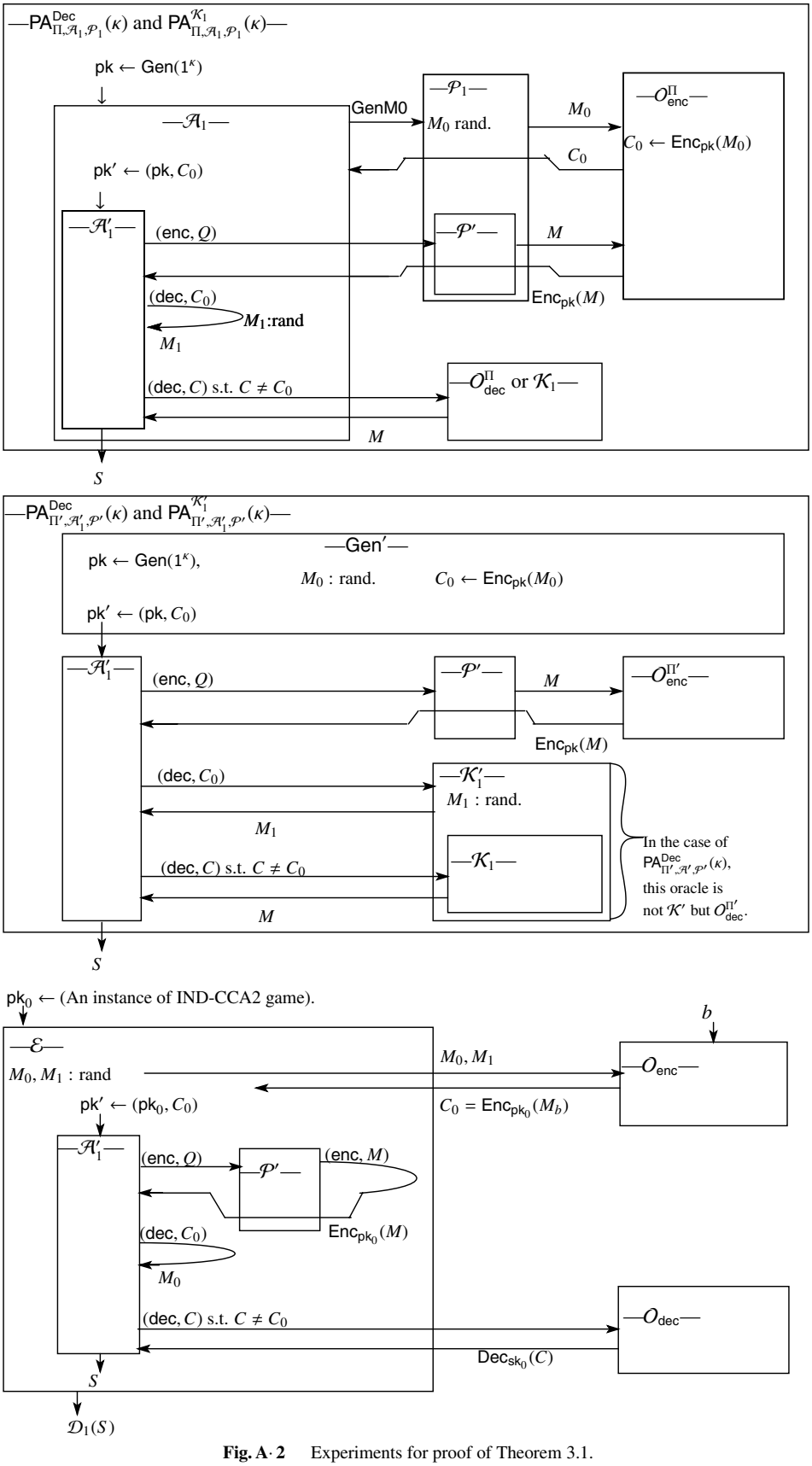

Fig. A· 2 Experiments for proof of Theorem 3.1.

$\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ *and that of the output of* $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$ *are computationally indistinguishable*.

Therefore, $\mathsf{PA}^{\mathcal{K}'_1}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$ is computationally indistinguishable from $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$ for any plaintext creator $\mathcal{P}'$. □

Proof of Lemma Appendix A.1 Before proving the lemma strictly, we give the intuition behind the proof. Recall that $\mathcal{A}_1$ uses $\mathcal{A}'_1$ as a subroutine. The two experiments $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ and $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$ are almost equal, except for who sends the answer of the query C_0 to $\mathcal{A}'_1$ and what answer $\mathcal{A}'_1$ receives. In the experiment $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$, the decryption oracle sends the answer of the query C_0 to $\mathcal{A}'_1$ and the answer which $\mathcal{A}'_1$ receives is the message $M_0 = \mathsf{Dec}_{\mathsf{sk}}(C_0)$. On the other hand, in the experiment $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$, $\mathcal{A}_1$ sends the answer to the query C_0 to $\mathcal{A}'_1$ and the answer which $\mathcal{A}'_1$ receives is a randomly selected message M_1.

Recall that Π is IND-CPA secure and computationally PA secure. In particular, Π is IND-CCA2 secure. Hence $\mathcal{A}'_1$ cannot distinguish the answer M_0 from the decryption oracle and the random answer M_1 from $\mathcal{A}_1$. Therefore, the distribution of the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ and that of the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$ are computationally indistinguishable.

Based on the above intuition, we show the lemma strictly. Let us make a contradictory supposition. That is, we suppose that there exists a polytime distinguisher $\mathcal{D}_1$ which can computationally distinguish the distribution of the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ from that of the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$.

By using algorithms $\mathcal{A}'_1$, $\mathcal{P}'$, and $\mathcal{D}_1$, we construct an adversary $\mathcal{E}(\mathsf{pk}_0)$ against the IND-CCA2 game, which satisfies the following properties. Here pk_0 is an instance of the IND-CCA2 game, b is an unknown bit, and $O_{\mathsf{enc}}(b, \mathsf{pk}_0, \cdot)$ and $O_{\mathsf{dec}}(\mathsf{sk}_0, \cdot)$ are the encryption oracle and the decryption oracle.

- If input b of the encryption oracle $O_{\mathsf{enc}}(b, \mathsf{pk}_0, \cdot)$ is 0, the distribution of the output of $\mathcal{E}$ is equal to that of $\mathcal{D}_1(S_0)$. Here S_0 is the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$.
- If input b of the encryption oracle $O_{\mathsf{enc}}(b, \mathsf{pk}_0, \cdot)$ is 1, the distribution of the output of $\mathcal{E}$ is equal to that of $\mathcal{D}_1(S_1)$. Here S_1 is the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$.

The description of $\mathcal{E}$ is depicted in Fig. A· 1. From the definition of $\mathcal{E}$, the properties mentioned above clearly hold. (See Fig. A· 2.)

Since Π is IND-CCA2 secure, the distribution of the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ has to be computationally indistinguishable from that of the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi',\mathcal{A}'_1,\mathcal{P}'}(\kappa)$. This means that the lemma is proved. □

Appendix B: Details of "Virtual Channel"

As described in the sketch of the proof of Theorems 4.1 and 5.5, we require a "virtual channel" which enables ${\mathcal{P}_0}^C$ to "send" data to $\mathcal{A}_0$. In this section, we give the definition of the virtual channel in detail.

We here only give the procedures on how ${\mathcal{P}_0}^C$ can "send" a bit b to $\mathcal{A}_0$. ${\mathcal{P}_0}^C$ can "send" the ciphertext C by executing the procedures for each bit of C.

We use the notations in Sect. 2.3. We can construct a virtual channel if Π is *not* IND-CPA secure. Let Π be a public-key encryption scheme which is not IND-CPA secure. We use the notation of Sect. 2.3.

Then there exists an adversary $\mathcal{B} = (\mathcal{B}_{\mathsf{find}}, \mathcal{B}_{\mathsf{guess}})$ such that $|\nu^1(\kappa) - \nu^0(\kappa)|$ is non-negligible.

Let $\mathcal{B}'$ be a polytime algorithm which outputs 1 or 0 if $\mathcal{B}$ outputs 0 or 1. Recall that $\nu^b(\kappa)$ the probability that $\mathcal{B}$ outputs b. By replacing $\mathcal{B}$ with $\mathcal{B}'$, if necessary, we can suppose that $\nu^1(\kappa) - \nu^0(\kappa) \geq 1/p_0(\kappa)$ holds for infinitely many κ.

For a public key pk_0, we set

$$\nu^b(\mathsf{pk}_0) = \Pr[\mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk}_0) = 1].$$

Clearly, $\nu^b(\kappa)$ is the expected value of $\nu^b(\mathsf{pk})$ when we generate pk by using $\mathsf{Gen}(1^\kappa)$.

We would like to use the value $\nu^b(\mathsf{pk}_0)$ in order to construct a "virtual channel." However, there is no way to know the value $\nu^b(\mathsf{pk}_0)$. So we prepare a function $\mathsf{guessNu}^{N,b}_{\mathcal{B}}(\mathsf{pk}_0)$, which enables us to guess the value $\nu^b(\mathsf{pk}_0)$. The description of $\mathsf{guessNu}^{N,b}_{\mathcal{B}}(\mathsf{pk}_0)$ is quite simple. That is, $\mathsf{guessNu}^{N,b}_{\mathcal{B}}(\mathsf{pk}_0)$ executes N times the experiment $\mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk}_0)$. Let ℓ_b be the number of times that $\mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk}_0)$ outputs 1. We guess that $\nu^b(\mathsf{pk}_0)$ is $\bar{\nu}^b = \ell_b/N$.

The precise description of $\mathsf{guessNu}^{N,b}_{\mathcal{B}}(\mathsf{pk}_0)$ is depicted in Fig. A· 3. We use $\mathsf{guessNu}^{N,b}_{\mathcal{B}}(\mathsf{pk}_0)$ as a subroutine when we construct a virtual channel.

The "*virtual channel*" comprises three algorithms $\mathsf{Setup}^N_{\mathcal{B}}$, $\mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}$, and $\mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}$. The roles of the three algorithms are as follow:

- $\mathsf{Setup}^N_{\mathcal{B}}$ is used to generate a parameter Param, which will be used to "send" and "receive" data.
- $\mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}$ enables ${\mathcal{P}_0}^C$ to "send" a bit.
- $\mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}$ enables $\mathcal{A}_0$ to "receive" a bit.

We stress that the "virtual channel" enables ${\mathcal{P}_0}^C$ to "send" *only one bit b*. Therefore, ${\mathcal{P}_0}^C$ have to use the "virtual channel" n times with $\mathcal{A}_0$, in order to "send" an n bit string.

We now give the description of the "virtual channel." See Fig. A· 3 for the detailed description.

$\mathsf{Setup}^N_{\mathcal{B}}(\mathsf{pk})$ executes the algorithms $\mathsf{guessNu}^{N,0}_{\mathcal{B}}(\mathsf{pk})$ and $\mathsf{guessNu}^{N,1}_{\mathcal{B}}(\mathsf{pk})$, and obtains the estimated value $\bar{\nu}^0$ and $\bar{\nu}^1$ of $\nu^0(\mathsf{pk})$ and $\nu^1(\mathsf{pk})$. Then it executes N times the algorithm $\mathcal{B}_{\mathsf{find}}(\mathsf{pk})$, and obtains the outputs $(m^{(1)}_0, m^{(1)}_1, \mathsf{St}^{(1)}_{\mathcal{B}})$, …, $(m^{(N)}_0, m^{(N)}_1, \mathsf{St}^{(N)}_{\mathcal{B}})$. $\mathsf{Setup}^N_{\mathcal{B}}(\mathsf{pk})$ finally sets $\mathsf{Param} = ((\bar{\nu}^0, \bar{\nu}^1), (m^{(1)}_0, m^{(1)}_1, \mathsf{St}^{(1)}_{\mathcal{B}}), \ldots, (m^{(N)}_0, m^{(N)}_1, \mathsf{St}^{(N)}_{\mathcal{B}}))$ and outputs Param.

$\mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(b, i)$ is the algorithm which outputs $m^{(i)}_b$. Note that ${\mathcal{P}_0}^C$ has to execute $\mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(b, i)$ for $i = 1, \ldots, N$, in order to "send" a bit b to $\mathcal{A}_0$. Then ${\mathcal{P}_0}^C$ ob-

—$\mathsf{guessNu}^{N,b}_{\mathcal{B}}(\mathsf{pk})$—
For $i = 1, \ldots, N$, execute the following procedures:
  Select r_i, s_i randomly.
  $(m_0, m_1, \mathsf{St}_{\mathcal{B}}) \leftarrow \mathcal{B}_{\mathsf{find}}(\mathsf{pk}; r_i)$,
  $c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_b; s_i)$,
  $b_i'' \leftarrow \mathcal{B}_{\mathsf{guess}}(\mathsf{pk}, c, \mathsf{St}_{\mathcal{B}})$.
$\ell_b \leftarrow$ (The number of b_i'' satisfying $b_i'' = 1$).
$\bar{\nu}^b \leftarrow \ell_b/N$, output $\bar{\nu}^b$.

—$\mathsf{Setup}^N_{\mathcal{B}}(\mathsf{pk})$—
$\bar{\nu}^0 \leftarrow \mathsf{guessNu}^{N,0}_{\mathcal{B}}(\mathsf{pk})$, $\bar{\nu}^1 \leftarrow \mathsf{guessNu}^{N,1}_{\mathcal{B}}(\mathsf{pk})$.
Select $r_1, \ldots, r_N$ randomly.
$(m_0^{(1)}, m_1^{(1)}, \mathsf{St}_{\mathcal{B}}^{(1)}) \leftarrow \mathcal{B}_{\mathsf{find}}(\mathsf{pk}; r_1)$,
$\ldots$,
$(m_0^{(N)}, m_1^{(N)}, \mathsf{St}_{\mathcal{B}}^{(N)}) \leftarrow \mathcal{B}_{\mathsf{find}}(\mathsf{pk}; r_N)$.
$\mathsf{Param} \leftarrow ((\bar{\nu}^0, \bar{\nu}^1), (m_0^{(1)}, m_1^{(1)}, \mathsf{St}_{\mathcal{B}}^{(1)}), \ldots, (m_0^{(N)}, m_1^{(N)}, \mathsf{St}_{\mathcal{B}}^{(N)}))$.
Output Param.

—$\mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(b, i)$—
Parse Param as
  $((\bar{\nu}^0, \bar{\nu}^1), (m_0^{(1)}, m_1^{(1)}, \mathsf{St}_{\mathcal{B}}^{(1)}), \ldots, (m_0^{(N)}, m_1^{(N)}, \mathsf{St}_{\mathcal{B}}^{(N)}))$.
Output $m_b^{(i)}$.

—$\mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(c_1 \ldots, c_N)$—
Select $s_1, \ldots, s_N$ randomly.
Parse Param as
  $((\bar{\nu}^0, \bar{\nu}^1), (m_0^{(1)}, m_1^{(1)}, \mathsf{St}_{\mathcal{B}}^{(1)}), \ldots, (m_0^{(N)}, m_1^{(N)}, \mathsf{St}_{\mathcal{B}}^{(N)}))$.
$b_1' \leftarrow \mathcal{B}_{\mathsf{guess}}(c_1, \mathsf{St}_{\mathcal{B}}^{(1)})$
$\ldots$,
$b_N' \leftarrow \mathcal{B}_{\mathsf{guess}}(c_N, \mathsf{St}_{\mathcal{B}}^{(N)})$.
$\ell \leftarrow$(The number of j satisfying $b_j' = 1$).
If $\ell/N \geq (\bar{\nu}^0 + \bar{\nu}^1)/2$, output $b' \leftarrow 1$.
Otherwise, $b' \leftarrow 0$.
Output b'.

—$\mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}, b)$—
$\mathsf{Param} \leftarrow \mathsf{Setup}^N_{\mathcal{B}}(\mathsf{pk})$
$m_b^{(1)} \leftarrow \mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(b, 1) \ldots, m_b^{(N)} \leftarrow \mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(b, N)$
$c_1 \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_b^{(1)}), \ldots, c_N \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_b^{(N)})$.
$b' \leftarrow \mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(c_1 \ldots, c_N)$
Output b'.

Fig. A· 3 Description of "virtual channel" and related functions.

tains the outputs $m_b^{(1)}, \ldots, m_b^{(N)}$ and sends these messages to the encryption oracle. The encryption oracle computes encryptions $c_1 = \mathsf{Enc}_{\mathsf{pk}}(m_b^{(1)}), \ldots, c_N = \mathsf{Enc}_{\mathsf{pk}}(m_b^{(N)})$.

We finally give the description of $\mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(c_1, \ldots, c_N)$. $\mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(c_1, \ldots, c_N)$ executes the algorithm $\mathcal{B}_{\mathsf{guess}}(c_i, \mathsf{St}_{\mathcal{B}}^{(i)})$ for $i = 1, \ldots, N$, and obtains the outputs $b_1', \ldots, b_N'$. Then it sets ℓ to be the number of j satisfying $b_j' = 1$. It sets $b' = 1$ or $b' = 0$, depending on whether $\ell/N \geq (\bar{\nu}^0 + \bar{\nu}^1)/2$ holds or not. It finally outputs b'. If $\mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}$ outputs b', this means that $\mathcal{A}_0$ "receives" the bit b'.

We next give the formal description of Lemma 4.3. To this end, for a public key pk and a bit b, we define an algorithm $\mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}, b)$ as follow: (See Fig. A· 3 for the formal description of it.) It executes $\mathsf{Setup}^N_{\mathcal{B}}(\mathsf{pk})$ and obtains Param as an output. Then it executes $\mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(b, 1), \ldots, \mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(b, N)$ and obtains $m_b^{(1)}, \ldots, m_b^{(N)}$ as outputs. It computes ciphertexts $c_1 = \mathsf{Enc}_{\mathsf{pk}}(m_b^{(1)}), \ldots, c_N = \mathsf{Enc}_{\mathsf{pk}}(m_b^{(N)})$. Then it computes $b' = \mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}}(c_1 \ldots, c_N)$ and outputs b'.

Lemma Appendix B.1 (Formal Version of Lemma 4.3): *We take* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*,* $\mathcal{B}$*, and* $\nu^b(\kappa)$ *as mentioned before. Let* pk *be a public key and b be a bit.*

Then, there exists a infinite set Λ *of security parameters, a polynomial* p_0*, and a family* $\{\Omega_\kappa\}_\kappa$ *of sets of public keys such that the following two properties hold for any* $\kappa \in \Lambda$:

1. $\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa) : \mathsf{pk} \in \Omega_\kappa] \geq \frac{1}{p_0(\kappa)}$,
2. $^\forall n > 0 \, ^\forall \alpha > 0 \, ^\forall \mathsf{pk} \in \Omega_\kappa \, ^\forall b \in \{0, 1\} :$
   $\Pr[\mathsf{Channel}^{N(\kappa)}_{\mathcal{B}}(\mathsf{pk}, b) = b] \geq 1 - \frac{1}{n\kappa^\alpha}$.

Here $N(\kappa) = \lceil 32n\kappa^\alpha / p_0(\kappa)^2 \rceil$.

Note that we will set n to the length of a ciphertext in the proof of Theorem 5.5.

Proof. For a public key pk and a bit b, we let $\nu^b(\mathsf{pk})$ denote $\Pr[\mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk}) = 1]$. Then clearly, the expected value $E((\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa) : \nu^b(\mathsf{pk}))$ is equal to $\nu^b(\kappa)$.

Let $\mu(\kappa)$ be $\nu^1(\kappa) - \nu^0(\kappa)$ and $\mu(\mathsf{pk})$ be $\nu^1(\mathsf{pk}) - \nu^0(\mathsf{pk})$. From the assumption, $\mu(\kappa)$ is non-negligible. Moreover, (by replacing $\mathcal{B}$ with $\mathcal{B}'$, if necessary,) we supposed that $\nu^1(\kappa) - \nu^0(\kappa) \geq 1/p_0(\kappa)$ holds for infinitely many κ. Therefore, there exists infinite set Λ and a polynomial p_0 such that, for any $\kappa \in \Lambda$, $\mu(\kappa)/2 \geq 1/2p_0(\kappa)$ holds. In particular, $\mu(\kappa) \geq 0$ holds. We define the set Ω_κ as

$$\Omega_\kappa = \{\mathsf{pk} \text{ s.t. } \mu(\mathsf{pk}) > \mu(\kappa)/2\}.$$

Let Q^κ_{pk} be the probability that pk arises. That is,

$$Q^\kappa_{\mathsf{pk}} = \Pr[(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{Gen}(1^\kappa) : \mathsf{PK} = \mathsf{pk}].$$

From the definition of $\mu(\kappa)$ and $\mu(\mathsf{pk})$, $\mu(\kappa)$ is equal to the expected value of $\mu(\mathsf{pk})$ when we take $(\mathsf{pk}, \mathsf{sk})$ according to $\mathsf{Gen}(1^\kappa)$. That is, it follows that

$$\mu(\kappa) = \sum_{\mathsf{pk}} \mu(\mathsf{pk}) Q^\kappa_{\mathsf{pk}}.$$

Therefore, it follows that

$$\begin{aligned}
&\mu(\kappa) \\
&= \textstyle\sum_{\mathsf{pk}} \mu(\mathsf{pk}) \cdot Q^\kappa_{\mathsf{pk}} \\
&= \textstyle\sum_{\mathsf{pk} \in \Omega_\kappa} \mu(\mathsf{pk}) \cdot Q^\kappa_{\mathsf{pk}} + \sum_{\mathsf{pk} \notin \Omega_\kappa} \mu(\mathsf{pk}) \cdot Q^\kappa_{\mathsf{pk}} \\
&\leq \textstyle\sum_{\mathsf{pk} \in \Omega_\kappa} Q^\kappa_{\mathsf{pk}} + \sum_{\mathsf{pk} \notin \Omega_\kappa} (\mu(\kappa)/2) \cdot Q^\kappa_{\mathsf{pk}} \\
&\leq \Pr[\mathsf{pk} \in \Omega_\kappa] + (\mu(\kappa)/2) \cdot \Pr[\mathsf{pk} \notin \Omega_\kappa] \\
&= \Pr[\mathsf{pk} \in \Omega_\kappa] + (\mu(\kappa)/2) \cdot (1 - \Pr[\mathsf{pk} \in \Omega_\kappa]) \\
&= (\mu(\kappa)/2) + (1 - \mu(\kappa)/2) \cdot \Pr[\mathsf{pk} \in \Omega_\kappa] \\
&\leq (\mu(\kappa)/2) + \Pr[\mathsf{pk} \in \Omega_\kappa].
\end{aligned}$$

Here the probabilities are taken over the random coins of Gen which generates pk.

Hence, it follows that

$$\Pr[\mathsf{pk} \in \Omega_\kappa] \geq \frac{\mu(\kappa)}{2}$$

Therefore, for any $\kappa \in \Lambda$, it follows that

$$\Pr[\mathsf{pk} \in \Omega_\kappa] \geq \frac{\mu(\kappa)}{2} \geq 1/p_0(\kappa). \tag{A·2}$$

That is, the first inequation of Lemma Appendix B.1 holds.

We next show the second inequation of Lemma Appendix B.1. We take α and n arbitrarily. Let b be a bit. We take a security parameter $\kappa \in \Lambda$ and a public key $\mathsf{pk}_0 \in \Omega_\kappa$ and study the algorithm $\mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, b)$, where $N = N(\kappa) = \lceil 32n\kappa^\alpha/p_0(\kappa)^2 \rceil$. In the execution of $\mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, b)$, $\mathsf{Setup}^N_{\mathcal{B}}(\mathsf{pk}_0)$ is executed. In the execution of $\mathsf{guessNu}^{N,0}_{\mathcal{B}}(\mathsf{pk}_0)$ of $\mathsf{Setup}^N_{\mathcal{B}}(\mathsf{pk}_0)$, the value $\bar{\nu}^0$ is computed. We estimate $|\bar{\nu}^0 - \nu^0(\mathsf{pk}_0)|$.

From the definition, $\bar{\nu}^0 = \ell_0/N$ holds, where ℓ_0 is the number of b''_i satisfying $b''_i = 1$. Therefore, it follows that

$$\bar{\nu}^0 = \frac{1}{N} \sum_{i=1,\ldots,N} b''_i.$$

From the definition of b''_i, it follows that

$$\Pr[b''_i = 0] = \nu^0(\mathsf{pk}_0).$$

Moreover, from the definition of b''_i, the data $b''_1, \ldots, b''_N$ are computed by using random tapes different from each other. That is, the distributions of $b''_1, \ldots, b''_N$ are independent from each other, (for any fixed pk_0.)

Hence, for any fixed pk_0, the distribution of $\bar{\nu}^0 = \sum_j b''_j/N$ is binomial distribution with the average $\nu^0(\mathsf{pk}_0)$ and the standard deviation $\sigma_0 = \sqrt{\nu^0(\mathsf{pk}_0)(1-\nu^0(\mathsf{pk}_0))/N} \leq 1/\sqrt{N}$.

Therefore, from Chebyshev's inequality, for any $\varepsilon > 0$, it follows that

$$\Pr[|\bar{\nu}^0 - \nu^0(\mathsf{pk}_0)| \geq \varepsilon] \leq \frac{{\sigma_0}^2}{\varepsilon^2} \leq \frac{1}{N\varepsilon^2}.$$

Hence, it follows that

$$\Pr\left[|\bar{\nu}^0 - \nu^0(\mathsf{pk}_0)| \leq \frac{\mu(\kappa)}{8}\right] \geq 1 - \frac{64}{N\mu(\kappa)^2}.$$

Via a similar discussion, we can also show that

$$\Pr\left[|\bar{\nu}^1 - \nu^1(\mathsf{pk}_0)| \leq \frac{\mu(\kappa)}{8}\right] \geq 1 - \frac{64}{N\mu(\kappa)^2}.$$

holds.

Therefore, it follows that

$$\Pr\left[\left|\frac{\bar{\nu}^0+\bar{\nu}^1}{2} - \frac{\nu^0(\mathsf{pk}_0)+\nu^1(\mathsf{pk}_0)}{2}\right| \leq \frac{\mu(\kappa)}{8}\right] \geq 1 - \frac{128}{N\mu(\kappa)^2}. \tag{A·3}$$

Here we use the fact $(1 - \frac{64}{N\mu(\kappa)^2})^2 \geq 1 - \frac{128}{N\mu(\kappa)^2}$.

We next estimate the probability

$$\Pr[\mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, 1) = 1],$$

where $\mathsf{pk}_0 \in \Omega_\kappa$ is a fixed public key. From the definition of $\mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, 1)$ and $\mathsf{Receive}^N_{\mathcal{B},\mathsf{pk}_0,\mathsf{Param}}(c_1 \ldots, c_N)$, the output $b' = \mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, 1)$ is 1 if and only if $\ell/N \geq (\bar{\nu}^0 + \bar{\nu}^1)/2$ holds. Here ℓ/N is a value computed in $\mathsf{Receive}^N_{\mathcal{B},\mathsf{pk}_0,\mathsf{Param}}(c_1 \ldots, c_N)$ of $\mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, 1)$.

We can apply a similar discussion to the case of ℓ_0/N. Hence, for any fixed pk_0, the distribution of $\ell/N = \sum_j b'_j/N$ is binomial distribution with the average $\nu^0(\mathsf{pk}_0)$ and the standard deviation $\sigma_0 = \sqrt{\nu^0(\mathsf{pk}_0)(1-\nu^0(\mathsf{pk}_0))/N} \leq 1/\sqrt{N}$.

We can apply a discussion similar to that for the case of ℓ_0/N again and can show the following inequality:

$$\Pr\left[\left|\frac{\ell}{N} - \nu^0(\mathsf{pk}_0)\right| \leq \frac{\mu(\kappa)}{8}\right] \geq 1 - \frac{64}{N\mu(\kappa)^2}. \tag{A·4}$$

From $\mathsf{pk}_0 \in \Omega_\kappa$, it follows that

$$\nu^1(\mathsf{pk}_0) - \nu^0(\mathsf{pk}_0) \geq \mu(\kappa)/2. \tag{A·5}$$

From the inequality (A·3), (A·4), and (A·5), we can conclude that

$$\Pr\left[\frac{\ell}{N} \geq \frac{\bar{\nu}^0 + \bar{\nu}^1}{2}\right] \geq 1 - \frac{32}{N\mu(\kappa)^2}. \tag{A·6}$$

Here we use the inequality $(1-\frac{128}{N\mu(\kappa)^2})(1-\frac{64}{N\mu(\kappa)^2}) \geq 1-\frac{128}{N\mu(\kappa)^2}-\frac{64}{N\mu(\kappa)^2} \geq 1 - \frac{32}{N\mu(\kappa)^2}$. From the definition of $\mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, 1)$, the inequation (A·6) means that

$$\Pr[b' \leftarrow \mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, 1) : b' = 1] \geq 1 - \frac{32}{N\mu(\kappa)^2}$$

holds.

By using similar discussions to the above one, we can also conclude that

$$\Pr[b' \leftarrow \mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, 0) : b' = 0] \geq 1 - \frac{32}{N\mu(\kappa)^2}$$

Therefore,

$$\Pr[b' \leftarrow \mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}_0, b) : b' = b] \geq 1 - \frac{32}{N\mu(\kappa)^2} \tag{A·7}$$

holds for $b = 0, 1$.

Recall that we set $N(\kappa) = \lceil 32n\kappa^\alpha/p_0(\kappa)^2 \rceil$. Recall also that $\mu(\kappa) \geq 1/p_0(\kappa)$ holds. Since we took $\kappa \in \Lambda$ arbitrarily, from inequality (A·7), the second inequality of Lemma Appendix B.1 holds. □

Appendix C: Detailed Proof of Theorem 5.5

Proof. We suppose that there exists a statistically PA secure public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ which is not IND-CPA secure, and we will show that Π is not non-trivial.

From the assumption that Π is not IND-CPA secure,

$\mathcal{A}_0(\mathsf{pk}; R_{\mathcal{A}})$	$\mathcal{P}_0^C(Q, \mathsf{St}_{\mathcal{P}}; R_{\mathcal{P}})$
$N \leftarrow \lceil n\kappa^{\alpha}/P(\kappa) \rceil$. Parse $R_{\mathcal{A}}$ as $\{R_i\}_i$. $\mathsf{Param}_i \leftarrow \mathsf{Setup}_{\mathcal{B}}^N(\mathsf{pk}; R_i)$, for $i = 1, \ldots, n$. Send $(\mathsf{SendPk}, \mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$ to a plaintext creator.	(Initially, $\mathsf{St}_{\mathcal{P}} = \varepsilon$.)
	If $Q = (\mathsf{SendPk}, \mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$ for some $(\mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$, then $\mathsf{St}_{\mathcal{P}} \leftarrow (\mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$. Return $(\mathsf{Null}, \mathsf{St}_{\mathcal{P}})$.
(Then receive an answer from the encryption oracle, and discard it.)	
Execute the following subroutine for $i = 1, \ldots, n$: { Make query (Sendb, i, j) to the plaintext creator for $j = 1, \ldots, N$.	If $Q = (\mathsf{Sendb}, i, j)$ for some (i, j), then Parse $\mathsf{St}_{\mathcal{P}}$ as $(\mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$. $b_i \leftarrow (i$-th bit of $C)$. $m_{b_i}^{(i,j)} \leftarrow \mathsf{Send}_{\mathcal{B},\mathsf{pk},\mathsf{Param}_i}^N(b_i, j)$ Return $(m_{b_i}^{(i,j)}, \mathsf{St}_{\mathcal{P}})$. (Then the encryption oracle sends the encryption of $m_{b_i}^{(i,j)}$ to the adversary.)
Receive a ciphertext $c_j^{(i)}$ as an answer for $j = 1, \ldots, N$. $b_i' \leftarrow \mathsf{Receive}_{\mathcal{B},\mathsf{pk},\mathsf{Param}_i}^N(c_1^{(i)} \ldots, c_N^{(i)})$	
$C' \leftarrow b_1' \| \cdots \| b_n'$.	
Make decryption query C'. Receive a message M' as the answer. Return (pk, C', M').	Otherwise, return $(\mathsf{Null}, \mathsf{St}_{\mathcal{P}})$.

Fig. A·4 Description of $\mathcal{A}_0$ and $\mathcal{P}_0^C$.

there exists an adversary $\mathcal{B}$ against the IND-CPA property with a non-negligible advantage.

We use the notations in Sect. 2.3 and Sect. Appendix B. As in Appendix B, we set $\nu^b(\kappa) = \Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^{\kappa}) : \mathsf{IND}_{\Pi,\mathcal{B}}^b(\mathsf{pk}) = 1]$. and $\mu(\kappa) = \nu^1(\kappa) - \nu^0(\kappa)$.

Let n be the bit length of ciphertexts of Π. We take an infinite set Λ, a polynomial $p_0(\kappa)$, and a family $\{\Omega_{\kappa}\}_{\kappa}$ of sets of public keys, whose existences are ensured in Lemma Appendix B.1. We take an arbitrary $\alpha > 0$ and set $N = N(\kappa) = 32n\kappa^{\alpha}/p_0(\kappa)^2$. Then, from Lemma Appendix B.1 and the definition of N, the following two properties hold for any $\kappa \in \Lambda$:

1. $\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^{\kappa}) : \mathsf{pk} \in \Omega_{\kappa}] \geq \frac{1}{p_0(\kappa)}$,
2. $^{\forall}\mathsf{pk} \in \Omega_{\kappa}\, ^{\forall}b \in \{0,1\} : \Pr[\mathsf{Channel}_{\mathcal{B}}^N(\mathsf{pk}, b) = b] \geq 1 - \frac{1}{n\kappa^{\alpha}}$.

By using $\mathcal{B}$ as a subroutine, we construct an adversary $\mathcal{A}_0$ as in Fig. A·4.

Since Π is PA secure, for this $\mathcal{A}_0$, there exists an extractor $\mathcal{K}_0$ such that, for any plaintext creator $\mathcal{P}$, the output of $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}}^{\mathsf{Dec}}(\kappa)$ and that of $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}}^{\mathcal{K}_0}(\kappa)$ are statistically indistinguishable.

Let C be an arbitrary ciphertext. We construct a plaintext creator $\mathcal{P}_0^C$ which "sends" C to an adversary. The precise description of $\mathcal{P}_0^C$ is depicted in Fig. A·4.

Since the output of $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}}^{\mathsf{Dec}}(\kappa)$ and that of $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}}^{\mathcal{K}_0}(\kappa)$ are statistically indistinguishable for any $\mathcal{P}$, the output of $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}^{\mathsf{Dec}}(\kappa)$ and that of $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}^{\mathcal{K}_0}(\kappa)$ are statistically indistinguishable, in particular.

For a public key/secret key pair $(\mathsf{pk}, \mathsf{sk})$, we let $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}^{\mathsf{Dec}}(\mathsf{pk}, \mathsf{sk})$ denote the experiment $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}^{\mathsf{Dec}}(\kappa)$ in which the experimenter uses pk as a public key and sk as a secret key. We also let $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}^{\mathcal{K}_0}(\mathsf{pk})$ denote the experiment $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}^{\mathcal{K}_0}(\kappa)$ in which the experimenter uses pk as a public key. (Note that no entity uses a secret key in the experiment $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}^{\mathcal{K}_0}(\kappa)$.)

Let C' be the ciphertext depicted in Fig. A·4. That is, let C' be the ciphertext which $\mathcal{A}_0$ "receives." We show that $C = C'$ holds with high probability in the experiment $\mathsf{PA}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}^{\mathsf{Dec}}(\mathsf{pk}, \mathsf{sk})$, if $(\mathsf{pk}, \mathsf{sk})$ is an element of Ω_{κ} for some $\kappa \in \Lambda$. That is, we show that $\mathcal{A}_0$ succeeds in "receiving" C with high probability if $(\mathsf{pk}, \mathsf{sk})$ is an element of Ω_{κ} for some $\kappa \in \Lambda$.

To this end, we fix $\kappa \in \Lambda$ and a public key/secret key

pair $(\mathsf{pk}_0, \mathsf{sk}_0) \in \Omega_\kappa$. We take b_i and b'_i as in Fig. A· 4. Then $C = b_1 \| \cdots \| b_n$ and $C' = b'_1 \| \cdots \| b'_n$ hold. Therefore, $C = C'$ holds if and only if $b_i = b'_i$ holds for any $i = 1, \ldots, n$. From the definition of Ω_κ, $b_i = b'_i$ holds with probability $1 - \frac{1}{n\kappa^\alpha}$ in the experiment $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk}_0, \mathsf{sk}_0)$. Hence, $C = C'$ holds with probability $(1 - \frac{1}{n\kappa^\alpha})^n \geq 1 - \frac{1}{\kappa^\alpha}$ in the experiment $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk}_0, \mathsf{sk}_0)$. Similarly, $C = C'$ holds with probability $1 - \frac{1}{\kappa^\alpha}$ even in the experiment $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk}_0)$.

Since Λ is a infinite set, there exists κ_0 such that for any $\kappa \in \Lambda$ satisfying $\kappa \geq \kappa_0$, $1 - \frac{1}{\kappa^\alpha} > 2/3$ holds. We let Λ_0 denote the set of $\kappa \in \Lambda$ satisfing $\kappa \geq \kappa_0$. Therefore, for any $\kappa \in \Lambda_0$ and for any $(\mathsf{pk}_0, \mathsf{sk}_0) \in \Omega_\kappa$, $C = C'$ holds with probability $2/3$ in both experiments $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk}_0, \mathsf{sk}_0)$ and $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk}_0)$.

By using $\mathcal{A}_0$, $\mathcal{K}_0$, and $\mathcal{P}_0^C$, we construct an inverter $\mathcal{I}_0$ against the non-triviality game. By giving an instance (pk, C) of the non-triviality game, $\mathcal{I}_0(\mathsf{pk}, C)$ executes $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk})$. Then $\mathcal{I}_0$ obtains an output (pk, C', M') of $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk})$, and outputs M'.

We now show that Π is not non-trivial. In other words, we show that $\mathcal{I}_0$ satisfies the following property: for any ciphertext generator C,

$$\Pr\left[\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), \\ C \leftarrow C(\mathsf{pk}).\end{array} : E_{\mathcal{I}_0}(\mathsf{pk}, \mathsf{sk}, C)\right]$$

is non-negligible. Here $E_{\mathcal{I}_0}(\mathsf{pk}, \mathsf{sk}, C)$ is the event that

$$\Pr[\mathsf{Oneway}^{\Pi}_{\mathcal{I}_0}(\mathsf{pk}, \mathsf{sk}, C) = 1] \geq 2/3$$

holds.

Let C_0 be an arbitrary ciphertext generator. In order to show that Π is not non-trivial, we construct a plaintext creator $\mathcal{P}_0^{C_0}$ as follows: $\mathcal{P}_0^{C_0}$ executes $C_0(\mathsf{pk})$, obtains the output C of C_0, and executes $\mathcal{P}_0^C$. From the definition of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^{C_0}}(\kappa)$, the output (pk, C', M') of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^{C_0}}(\kappa)$ satisfies $M' = \mathsf{Dec}_{\mathsf{sk}}(C')$ clearly. Since $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^{C_0}}(\kappa)$ and $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^{C_0}}(\kappa)$ are statistically indistinguishable, the output (pk, C', M') of $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^{C_0}}(\kappa)$ satisfies $M' = \mathsf{Dec}_{\mathsf{sk}}(C')$ with overwhelming probability also.

We study the experiment $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^{C_0}}(\kappa)$ step by step. In the experiment $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^{C_0}}(\kappa)$, a key pair $(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^\kappa)$ is first generated, a ciphertext $C = C_0(\mathsf{pk})$ is next generated by $\mathcal{P}_0^{C_0}$, then $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk}, \mathsf{sk})$ is executed and finally the output (pk, C', M') of $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk}, \mathsf{sk})$ is output. From the definition of $\mathcal{I}_0$, the last component M' of the output (pk, C', M') of $\mathsf{PA}^{\mathcal{K}_0}_{\Pi,\mathcal{A}_0,\mathcal{P}_0^C}(\mathsf{pk}, \mathsf{sk})$ is equal to the output of $\mathcal{I}_0(\mathsf{pk}, C)$.

We have already proved that $C = C'$ holds with probability $2/3$ if $(\mathsf{pk}, \mathsf{sk}) \in \Omega_\kappa$ holds for some $\kappa \in \Lambda_0$ Moreover, we also showed that $M' = \mathsf{Dec}_{\mathsf{sk}}(C')$ holds with overwhelming probability. This means that, if $(\mathsf{pk}, \mathsf{sk}) \in \Omega_\kappa$ holds for some some $\kappa \in \Lambda_0$, the output M' of $\mathcal{I}_0(\mathsf{pk}, C)$ is equal to $\mathsf{Dec}_{\mathsf{sk}}(C)$ with probability $2/3$. From the definition of the event $E_{\mathcal{I}_0}(\mathsf{pk}, \mathsf{sk}, C)$, this means that the event $E_{\mathcal{I}_0}(\mathsf{pk}, \mathsf{sk}, C)$ occurs if $(\mathsf{pk}, \mathsf{sk}) \in \Omega_\kappa$ holds for some $\kappa \in \Lambda_0$.

From the definition of Ω_κ, $(\mathsf{pk}, \mathsf{sk}) \in \Omega_\kappa$ holds with probability of at least $1/p_0(\kappa)$. This means that

$$\Pr\left[\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), \\ C \leftarrow C_0(\mathsf{pk}).\end{array} : E_{\mathcal{I}_0}(\mathsf{pk}, \mathsf{sk}, C)\right] \geq 1/p_0(\kappa)$$

holds for any C_0 and for any $\kappa \in \Lambda_0$ Since Λ_0 is a infinite set, this means that

$$\Pr\left[\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa), \\ C \leftarrow C_0(\mathsf{pk}).\end{array} : E_{\mathcal{I}_0}(\mathsf{pk}, \mathsf{sk}, C)\right]$$

is non-negligible for any C_0. □

Appendix D: Detailed Proof of Theorem 4.1

Theorem 4.1 for the perfect and statistical PA-nesses is clearly derived from Theorem 5.5 and Proposition 5.1. Therefore, we show Theorem 4.1 for the computational PA-ness.

Proof. We suppose that there exists a computational PA secure public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ which is not IND-CPA secure, and we will show that Π is not oneway.

From the assumption that Π is not IND-CPA secure, there exists an adversary $\mathcal{B}$ against the IND-CPA property with a non-negligible advantage. We use the notations of Sect. 2.3 and Appendix B. As in Appendix B, we set $\nu^b(\kappa) = \Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa) : \mathsf{IND}^b_{\Pi,\mathcal{B}}(\mathsf{pk}) = 1]$. and $\mu(\kappa) = \nu^1(\kappa) - \nu^0(\kappa)$. Without loss of generality, we can suppose that $\mu(\kappa) \geq 0$ holds.

Let n be the bit length of ciphertexts of Π. We take a infinite set Λ, a polynomial $p_0(\kappa)$ and a family $\{\Omega_\kappa\}_\kappa$ of sets of public keys, whose existences are ensured in Lemma Appendix B.1. We take an arbitrary $\alpha > 0$ and set $N = N(\kappa) = (n+1)\kappa^\alpha / p_0(\kappa)$. Then, from Lemma Appendix B.1 and from the definition of N, the following two properties hold for any $\kappa \in \Lambda$:

1. $\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa) : \mathsf{pk} \in \Omega_\kappa] \geq \frac{1}{p_0(\kappa)}$,
2. $^\forall \mathsf{pk} \in \Omega_\kappa\, ^\forall b \in \{0,1\} : \Pr[\mathsf{Channel}^N_{\mathcal{B}}(\mathsf{pk}, b) = b] \geq 1 - \frac{1}{(n+1)\kappa^\alpha}$.

The basic idea behind the proof of Theorem 4.1 for the computational PA-ness is similar to that of Theorem 5.5. However, these proofs are essentially different in one point. Recall that, in the proof of Theorem 5.5, we used the extractor $\mathcal{K}_0$ in order to invert a ciphertext C, because an extractor for the statistical PA-ness succeeds in outputting $\mathsf{Dec}_{\mathsf{sk}}(C)$ with overwhelming probability.

In contrast, we can not do so now. This is because an extractor for the computational PA-ness may not output

$-\mathcal{A}_1(\mathsf{pk}; R_{\mathcal{A}})-$	$-\mathcal{P}_1(Q, \mathsf{St}_{\mathcal{P}}; R_{\mathcal{P}})-$	$-{\mathcal{P}_2}^C(Q, \mathsf{St}_{\mathcal{P}}; R_{\mathcal{P}})-$
$N \leftarrow \lceil (n+1)\kappa^{\alpha}/p_0(\kappa) \rceil$. Parse $R_{\mathcal{A}}$ as $(\{R_i\}_i, R')$. $\mathsf{Param}_i \leftarrow \mathsf{Setup}^N_{\mathcal{B}}(\mathsf{pk}; R_i)$, for $i = 1, \ldots, n$. Send $(\mathsf{SendPk}, \mathsf{pk}, \{\mathsf{Param}\}_i, N)$ to a plaintext creator.	(Initially, $\mathsf{St}_{\mathcal{P}} = \varepsilon$.)	(Initially, $\mathsf{St}_{\mathcal{P}} = \varepsilon$.)
	If $Q = (\mathsf{SendX}, \mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$ for some $(\mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$, then $M \leftarrow (\text{rand})$, $C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M)$. $\mathsf{St}_{\mathcal{P}} \leftarrow (\mathsf{pk}, \{\mathsf{Param}_i\}_i, N, C, M)$. Return $(\mathsf{Null}, \mathsf{St}_{\mathcal{P}})$.	If $Q = (\mathsf{SendX}, \mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$ for some $(\mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$, then $\mathsf{St}_{\mathcal{P}} \leftarrow (\mathsf{pk}, \{\mathsf{Param}_i\}_i, N, C)$. Return $(\mathsf{Null}, \mathsf{St}_{\mathcal{P}})$.
(Then receive an answer from the encryption oracle, and discard it.)		
Execute the following subroutine for $i = 1, \ldots, n$: Make query (Sendb, i, j) to the plaintext creator for $j = 1, \ldots, N$.	If $Q = (\mathsf{Sendb}, i, j)$ for some (i, j), then Parse $\mathsf{St}_{\mathcal{P}}$ as $(\mathsf{pk}, \{\mathsf{Param}_i\}_i, N, C, M)$. $b_i \leftarrow (i\text{-th bit of } C)$. $m^{(i,j)}_{b_i} \leftarrow \mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}_i}(b_i, j)$ Return $(m^{(i,j)}_{b_i}, \mathsf{St}_{\mathcal{P}})$. (Then the encryption oracle sends the encryption of $m^{(i,j)}_{b_i}$ to the adversary.)	If $Q = (\mathsf{Sendb}, i, j)$ for some (i, j), then Parse $\mathsf{St}_{\mathcal{P}}$ as $(\mathsf{pk}, \{\mathsf{Param}_i\}_i, N, C)$. $b_i \leftarrow (i\text{-th bit of } C.)$ $m^{(i,j)}_{b_i} \leftarrow \mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}_i}(b_i, j)$ Return $(m^{(i,j)}_{b_i}, \mathsf{St}_{\mathcal{P}})$. (Then the encryption oracle sends the encryption of $m^{(i,j)}_{b_i}$ to the adversary.)
Receive ciphertext $c^{(i)}_j$ as an answer for $j = 1, \ldots, N$. $b'_i \leftarrow \mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}_i}(c^{(i)}_1 \ldots, c^{(i)}_N; \hat{R}_i)$		
$C' \leftarrow b'_1 \| \cdots \| b'_n$.		
Make decryption query C'. Receive a message M' as the answer.		
$\mathsf{Param}' \leftarrow \mathsf{Setup}^N_{\mathcal{B}}(\mathsf{pk}; R')$. Send $(\mathsf{SendCMPram}', C', M', \mathsf{Param}')$ to a plaintext creator.		
	If $Q = (\mathsf{SendCMParam}', C', M', \mathsf{Param}')$ for some $(C', M', \mathsf{Param}')$ then Parse $\mathsf{St}_{\mathcal{P}}$ as $(\mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$. $T \leftarrow \begin{cases} 1 & \text{if } M = M', \\ 0 & \text{otherwise.} \end{cases}$ $\mathsf{St}_{\mathcal{P}} \leftarrow (\mathsf{pk}, \mathsf{Param}', N, M', T)$. Return $(\mathsf{Null}, \mathsf{St}_{\mathcal{P}})$.	If $Q = (\mathsf{SendCMParam}', C', M', \mathsf{Param}')$ for some $(C', M', \mathsf{Param}')$ then Parse $\mathsf{St}_{\mathcal{P}}$ as $(\mathsf{pk}, \{\mathsf{Param}_i\}_i, N)$. $T \leftarrow 1$ $\mathsf{St}_{\mathcal{P}} \leftarrow (\mathsf{pk}, \mathsf{Param}', N, M', T)$. Return $(\mathsf{Null}, \mathsf{St}_{\mathcal{P}})$.
(Then receive an answer from the encryption oracle, and discard it.)		
Execute the following subroutine: Make query (SendT, j) to the plaintext creator for $j = 1, \ldots, N$.	If $Q = (\mathsf{SendT}, j)$ for some j, then $m^{(j)}_T \leftarrow \mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}'}(T, j)$ Return $(m^{(j)}_T, \mathsf{St}_{\mathcal{P}})$. (Then the encryption oracle sends the encryption of $m^{(j)}_T$ to the adversary.)	If $Q = (\mathsf{SendT}, j)$ for some j, then $m^{(j)}_T \leftarrow \mathsf{Send}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}'}(T, j)$ Return $(m^{(j)}_T, \mathsf{St}_{\mathcal{P}})$. (Then the encryption oracle sends the encryption of $m^{(j)}_T$ to the adversary.)
Receive ciphertexts c'_j as an answer for $j = 1, \ldots, N$. $T' \leftarrow \mathsf{Receive}^N_{\mathcal{B},\mathsf{pk},\mathsf{Param}'}(c'_1 \ldots, c'_N; \hat{R}')$		
If $T' = 1$, then return (pk, C', M'). Otherwise, return $\perp$.	Otherwise, return $(\mathsf{Null}, \mathsf{St}_{\mathcal{P}})$.	Otherwise, return $(\mathsf{Null}, \mathsf{St}_{\mathcal{P}})$.

Fig. A· 5 Description of $\mathcal{A}_1$, $\mathcal{P}_1$, and ${\mathcal{P}_2}^C$.

$\mathsf{Dec}_{\mathsf{sk}}(C)$, although it outputs a plaintext which is computationally indistinguishable from $\mathsf{Dec}_{\mathsf{sk}}(C)$.

Therefore, we have to use a more elaborate technique. In order to prove Theorem 4.1 for the computational PA-ness, we construct an adversary $\mathcal{A}_1$, by modifying the description of the adversary $\mathcal{A}_0$ of the proof of Theorem 5.5.

The precise description of $\mathcal{A}_1$ is depicted in Fig. A· 5. (We will later give the intuitive description of $\mathcal{A}_1$.) Since Π is computationally PA secure, for this $\mathcal{A}_1$, there exists an extractor $\mathcal{K}_1$ such that, for any plaintext creator $\mathcal{P}$, the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}}(\kappa)$ and that of $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}}(\kappa)$ are computationally indistinguishable.

We construct a plaintext creator $\mathcal{P}_1$, by modifying the description of $\mathcal{P}_0$ of the proof of Theorem 5.5. The precise description of $\mathcal{P}_1$ is depicted in Fig. A· 5. We now give the intuitive descriptions of $\mathcal{A}_1$ and $\mathcal{P}_1$. In the experiment $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$, $\mathcal{P}_1$ generates a plaintext M randomly, computes $C = \mathsf{Enc}_{\mathsf{pk}}(M)$, and "sends" C to $\mathcal{A}_1$ via the "virtual channel." Then $\mathcal{A}_1$ "receives" C' (which is expected to be equal to C), makes a decryption query C', obtains a message M' as an answer, and sends M' to $\mathcal{P}_1$. $\mathcal{P}_1$ sets $T = 1$ or $T = 0$, depending on whether $M = M'$ holds or not. $\mathcal{P}_1$ "sends" T to $\mathcal{A}_1$.

Then $\mathcal{A}_1$ "receives" T' (which is expected to be equal to T), and outputs $(\mathsf{pk}, C', M', T')$.

We will show that $\mathcal{K}_1$ succeeds in outputting $M = \mathsf{Dec}_{\mathsf{sk}}(C)$ with non-negligible probability in the experiment $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$. In other words, we will show that $M' = M$ holds with non-negligible probability in this experiment. To this end, we study the experiment $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$. In the experiment $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$, the decryption oracle answers the query C'. Therefore, $M' = \mathsf{Dec}_{\mathsf{sk}}(C')$ holds. From Lemma Appendix B.1, $C = C'$ holds with non-negligible probability. Therefore, $M' = \mathsf{Dec}_{\mathsf{sk}}(C') = \mathsf{Dec}_{\mathsf{sk}}(C) = M$ holds with non-negligible probability. From the definition of T, this means that $T = 1$ holds with non-negligible probability. From Lemma Appendix B.1, $T = T'$ holds with non-negligible probability. Therefore, $T' = 1$ holds with non-negligible probability.

Recall that the output of $\mathsf{PA}^{\mathsf{Dec}}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ and that of $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$ are computationally indistinguishable. Therefore, even in the experiment $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$, $T' = 1$ holds with non-negligible probability. From Lemma Appendix B.1, $T = T'$ holds with non-negligible probability. Therefore, $T = 1$ holds with non-negligible probability. From the definition of T, this means that $M = M'$ holds with non-negligible probability, even in the experiment $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\kappa)$. That is, $\mathcal{K}_1$ succeeds in outputting $M = \mathsf{Dec}_{\mathsf{sk}}(C)$ with non-negligible probability.

Let C be a ciphertext. We cannot use $\mathcal{P}_1$ itself in order to invert a ciphertext. So, we construct another plaintext creator ${\mathcal{P}_2}^C$ by modifying $\mathcal{P}_1$, and show that we can invert a ciphertext by using ${\mathcal{P}_2}^C$. There are two differences between the description $\mathcal{P}_1$ and that of ${\mathcal{P}_2}^C$. First, ${\mathcal{P}_2}^C$ "sends" the input ciphertext C to $\mathcal{A}_1$, although $\mathcal{P}_1$ generates a ciphertext itself and "sends" the ciphertext to $\mathcal{A}_1$. Second, ${\mathcal{P}_2}^C$ always sets $T = 1$, although $\mathcal{P}_1$ sets $T = 1$ or $T = 0$, depending on whether $M = M'$ holds or not. The precise description of ${\mathcal{P}_2}^C$ is depicted in Fig. A· 5.

We give the description of $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,{\mathcal{P}_2}^C}(\kappa)$ step by step. ${\mathcal{P}_2}^C$ "sends" C to $\mathcal{A}_1$. Then $\mathcal{A}_1$ receives C' (which is expected to be equal to C), makes decryption query C' and receives a plaintext M', and sends M' to ${\mathcal{P}_2}^C$. ${\mathcal{P}_2}^C$ sets $T = 1$ and "sends" T to $\mathcal{A}_1$. Then $\mathcal{A}_1$ "receives" T' (which is expected to be equal to T), and outputs $(\mathsf{pk}, C', M', T')$.

We let $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,{\mathcal{P}_2}^C}(\mathsf{pk})$ denote the experiment $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,{\mathcal{P}_2}^C}(\kappa)$ in which pk is used as a public key. By using $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,{\mathcal{P}_2}^C}(\mathsf{pk})$ as a subroutine, we construct an inverter $\mathcal{I}_1$ for the onewayness game as follows. $\mathcal{I}_1(\mathsf{pk}, C)$ executes $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,{\mathcal{P}_2}^C}(\mathsf{pk})$ to obtain the output $(\mathsf{pk}, C', M', T')$ and outputs M'.

We compare $\mathcal{I}_1(\mathsf{pk}, C)$ and $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\mathsf{pk})$. $\mathcal{K}_1$ is executed in both $\mathcal{I}_1(\mathsf{pk}, C)$ and $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\mathsf{pk})$, which are perfectly indistinguishable from the viewpoint of $\mathcal{K}_1$. More precisely, the input of $\mathcal{K}_1$ in $\mathcal{I}_1(\mathsf{pk}, C)$ and that in $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\mathsf{pk})$ are perfectly indistinguishable, (if the input (pk, C) of $\mathcal{I}_1$ is chosen randomly by the experimenter of the onewayness.) Recall that $\mathcal{K}_1$ succeeds in outputting the decrypted plaintext in $\mathsf{PA}^{\mathcal{K}_1}_{\Pi,\mathcal{A}_1,\mathcal{P}_1}(\mathsf{pk})$ with non-negligible probability. Therefore, even in the execution of $\mathcal{I}_1(\mathsf{pk}, C)$, $\mathcal{K}_1$ succeeds in outputting the decrypted plaintext $M = \mathsf{Dec}_{\mathsf{sk}}(C)$ with non-negligible probability. This means that $\mathcal{I}_1(\mathsf{pk}, C)$ succeeds in inverting C with non-negligible probability. Therefore, Π is not oneway. □

Isamu Teranishi received B.S. and M.E. degrees in mathematics in 2000 and 2002 respectively, from Tokyo Institute of Technology. He joined the NEC as a researcher in 2002.

Wakaha Ogata received B.S., M.E. and D.E. degrees in electrical and electronic engineering in 1989, 1991 and 1994, respectively, from Tokyo Institute of Technology. From 1995 to 2000, she was an Assistant Professor at Himeji Institute of Technology. Since 2000, she has been an Associate Professor at Tokyo Institute of Technology. Her current interests are information security and cryptography.