ISSN 1342-2812

Research Reports on
Mathematical and
Computing Sciences

Public-Key Encryption with Masking

Ryotaro Hayashi and Keisuke Tanaka

August 2006, C–226

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES C: Computer Science

Public-Key Encryption with Masking

Ryotaro Hayashi and Keisuke Tanaka

Dept. of Mathematical and Computing Sciences
Tokyo Institute of Technology
W8-55, 2-12-1 Ookayama Meguro-ku, Tokyo 152-8552, Japan
{hayashi9, keisuke}@is.titech.ac.jp

August 21, 2006

Abstract

Timed-release encryption, first mentioned by May [12] and discussed by Rivest, Shamir, and Wanger [15], is a cryptographic primitive which enables us to "send a message into the future."

We take a simple approach, called public-key encryption with masking, in order to realize this requirement. In our model, the sender first encrypts a plaintext and "masks" the ciphertext, then send it the receiver. Given a masked ciphertext, not only the person who does not have a secret key but also the secret-key holder (the receiver) cannot know the plaintext underlying the masked ciphertext. If the sender wants the receiver to decrypt the ciphertext, the sender makes some actions to reveal the mask of the masked ciphertext. Then, the secret-key holder can decrypt the unmasked ciphertext. However, the person who does not have a secret key cannot still get any information about the plaintext underlying the unmasked ciphertext. Moreover, the sender cannot change the message underlying the (un)masked ciphertext after sending the masked ciphertext.

In this paper, we formalize the model of public-key encryption with masking, and also propose its concrete scheme.

Keywords: timed-release encryption, public-key encryption, OAEP, one-time pad.

1 Introduction

Timed-Release Encryption. Timed-release encryption is a cryptographic primitive which enables us to "send a message into the future." One way to do this is to encrypt a message such that the receiver cannot decrypt the ciphertext until some specific time in the future. Such a primitive would have many applications such as electronic auctions, key escrow, scheduled payment methods, sealed-bid auctions, lotteries, etc. This idea was first mentioned by May [12] and then discussed in detail by Rivest, Shamir, and Wanger [15].

The previously proposed schemes fall into two categories, the time-lock puzzle approach [15, 1, 4, 11, 8, 9] and the time-server-based approach [12, 15, 13, 3, 10, 14, 16, 5]. In the time-lock puzzle approach, the sender encrypts a message and the receiver needs to perform non-parallelizable computation to decrypt it. This approach does not require a trusted third party. However, it turns out to be computationally expensive for the receiver, and the time when the receiver decrypts the ciphertext depends on the receiver's computational power. Thus, the sender cannot control when the receiver decrypts the ciphertext in the sense of the absolute time (e.g. 21:59, August 18, 2006 GMT). In the time-server-based approach, the sender encrypts a message such that the receiver needs some secret value, published by a trusted agent on the target date, in order to decrypt the ciphertext. Although this approach requires the trusted third party, the sender can control the absolute time when the receiver can decrypt the ciphertext.

PUBLIC-KEY ENCRYPTION WITH MASKING. In this paper, we take another approach to realize this requirement. In the time-server-based approach, as mentioned above, once the sender sends a ciphertext to the receiver, the sender does not have to do anything, since the receiver can decrypt the ciphertext by using the receiver's secret key and the secret information published by the trusted agent. However, in some applications, it is natural and useful that the sender makes some actions to the receiver when the sender wants the receiver to decrypt the ciphertext. That is, we consider the following model.

- The sender first encrypts a plaintext and "masks" the ciphertext, then send it the receiver. Given a masked ciphertext, not only the person who does not have a secret key but also the secret-key holder (the receiver) cannot know the plaintext underlying the masked ciphertext.

- If the sender wants the receiver to decrypt the ciphertext, the sender makes some actions to reveal the mask of the masked ciphertext. Then, the secret-key holder can decrypt the unmasked ciphertext. However, the person who does not have a secret key cannot still get any information about the plaintext underlying the unmasked ciphertext.

- The sender cannot change the message underlying the (un)masked ciphertext after sending the masked ciphertext.

In this model, the sender can control the time when the receiver can decrypt the message *after* the sender sends the ciphertext. This property seems useful in the situation that the time when the ciphertext is decrypted depends on some circumstances.

Furthermore, in this model, the sender can cancel to open the message which the receiver has already received. For example, let us consider the paper review process. The author encrypts the paper and sends it to the reviewers by the deadline. The reviewers cannot read the paper until the review process begins. Then, the author opens the paper to the reviewer when the review process begins. Here, suppose that the author who has submitted the paper finds some mistake in the paper, and wants to withdraw the paper before the deadline. Then, in our model, the author can withdraw the submission without revealing the content of the paper to anyone, including the reviewers. In addition to this property, it is preferable that the author cannot change the content of the paper after the deadline.

In this paper, in order to realize the above idea, we propose a special type of public-key encryption, called *public-key encryption with masking.* A public-key encryption scheme with masking consists of three algorithms, that is, a key generation algorithm $\mathcal{K}$, an encryption-and-masking algorithm $\mathcal{EM}$, and a decryption algorithm $\mathcal{D}$ for unmasked ciphertexts. The sender computes a masked ciphertext c and an unmasking information $\tilde{r}$ by using the algorithm $\mathcal{EM}$. Given only a masked ciphertext c, the receiver cannot know the plaintext underlying c. Once the receiver gets the unmasking information $\tilde{r}$, the secret-key holder can decrypt the unmasked ciphertext. However, the person who does not have a secret key cannot still decrypt the unmasked ciphertext even if the person knows $\tilde{r}$. Note that we do not require the trusted third party.

We also formalize the security properties for public-key encryption with masking. As mentioned above, we require the following properties.

Security Property 1. Given a masked ciphertext c, not only the person who does not have a secret key but also the secret-key holder cannot still get any information about the plaintext underlying the masked ciphertext.

Security Property 2. Given an unmasked ciphertext $(c, \tilde{r})$, which contains a masked ciphertext c and an unmasking information $\tilde{r}$, the person who does not have a secret key cannot get any information about the plaintext underlying the unmasked ciphertext.

We formalize the security properties 1 and 2 as "indistinguishability of masked ciphertexts" and "indistinguishability of unmasked ciphertexts," respectively. These properties are derived naturally from the standard property of the indistinguishability of ciphertexts.

Moreover, we formalize the following security property, called "binding." This property claims that the sender cannot change the message underlying the (un)masked ciphertext after sending the masked ciphertext c.

Security Property 3. The sender cannot produce one masked ciphertext c and two unmasking informations $\tilde{r}_0, \tilde{r}_1$ such that two unmasked ciphertexts, $(c, \tilde{r}_0)$ and $(c, \tilde{r}_1)$, are valid, and the plaintexts of these two unmasked ciphertexts are different.

The Constructions of Public-Key Encryption with Masking. It seems possible to construct a public-key encryption scheme with masking by combining a public-key encryption scheme with a commitment scheme. For example, the sender first encrypts a message by using the receiver's public key, and commit the ciphertext. Then, only the receiver can decrypt the ciphertext if the sender opens the commitment. However, we require some efficient string commitment scheme in order to construct such a scheme. Furthermore, it is not clear that this scheme is secure even if we combine a secure public-key encryption scheme with a secure commitment scheme.

It also seems possible to construct a public-key encryption scheme with masking by using a multiple encryption scheme, that is, we employ two pairs of public and secret keys. For example, the sender first encrypts a message by using the receiver's public key, and encrypts the resulting ciphertext by the sender's public key. If the sender opens the sender's public key, the receiver can decrypt the ciphertext. However, it is not clear that this scheme satisfies the security properties for public-key encryption with masking. Note that multiple encryption schemes are not always secure even if the basic encryption schemes are secure [17]. Furthermore, in this scheme, the sender has to reveal the secret key to (at least) the receiver, and the cost for key generation is required for each run of the protocol. The computational cost for the multiple encryption is basically twice as the basic encryption scheme.

Thus, in order to construct efficient schemes for public-key encryption with masking, we have to consider how to mask the ciphertext reasonably. One standard way to mask some data is using one-time pad. That is, we employ a (standard) public-key encryption scheme and apply a one-time pad to the plaintext or the ciphertext of the scheme. However, these schemes do not satisfy the security properties (See Section 4.).

In this paper, we also propose a concrete public-key encryption scheme with masking based on OAEP (Bellare and Rogaway [2], Fujisaki, Okamoto, Pointcheval, and Stern [6, 7]). In our scheme, we apply the one-time pad neither to the plaintext nor the ciphertext of OAEP, but to the randomness of OAEP (See Section 5). We prove that our scheme satisfies the three security properties described above in the random oracle model.

Organization. In Section 2, we review the definitions of families of trap-door permutations and partial one-wayness. In Section 3, we propose the definition of public-key encryption with masking and its security properties. In Section 4, we consider two trivial constructions of public-key encryption with masking, and point out their weakness. In Section 5, we propose a concrete scheme based on OAEP for public-key encryption with masking, and prove its security.

2 Preliminaries

In this paper, we use the following notations. If A is a probabilistic algorithm, then $A(x_1, x_2, \ldots, x_n; r)$ is the result of running A on inputs $x_1, x_2, \ldots, x_n$ and coins r. We let $y \leftarrow A(x_1, x_2, \ldots, x_n)$ denote the experiment of picking r at random and letting y be $A(x_1, x_2, \ldots, x_n; r)$. If S is a finite set then $x \overset{R}{\leftarrow} S$ is the operation of picking an element uniformly from S. If α is not an algorithm then $x \leftarrow \alpha$ is a simple assignment statement.

We say that the function $\epsilon : \mathbb{N} \rightarrow \mathbb{R}^+$ is negligible (in k) if for every constant $c > 0$ there exists an integer k' such that $\epsilon(k) < 1/k^c$ for all $k \geq k'$.

2.1 Families of Trap-Door Permutations

In this section, we review the definitions of families of trap-door permutations and θ-partial one-wayness.

Definition 1 (Families of Trap-Door Permutations). *A family of trap-door permutations $\mathcal{TP} = (K, F, F^{-1})$ is described as follows.*

- *The key generation algorithm K takes as input a security parameter 1^k and outputs a public key pk and a matching secret key (trap-door) sk. For given pk, the domain $\mathrm{Dom}(pk)$ and the range $\mathrm{Rng}(pk)$ of the permutation are uniquely determined where $\mathrm{Dom}(pk) = \mathrm{Rng}(pk)$.*
- *The evaluation algorithm F is a deterministic algorithm that takes a public key pk and an element $x \in \mathrm{Dom}(pk)$ and returns an element $y \in \mathrm{Rng}(pk)$. We require that F_{pk} is bijective for any pk, that is, F_{pk} is a permutation over $\mathrm{Dom}(pk)$ for any pk.*
- *The inversion algorithm F^{-1} is a deterministic algorithm that takes a secret key sk and an element $y \in \mathrm{Rng}(pk)$ and returns an element $x \in \mathrm{Dom}(pk)$. We require that for any $(pk, sk) \leftarrow K(1^k)$ and $x \in \mathrm{Dom}(pk)$, if $y = F_{pk}(x)$ then $x = F^{-1}_{sk}(y)$.*

Definition 2 (θ-Partial One-Wayness). *Let $k \in \mathbb{N}$ be a security parameter, and $0 < \theta \leq 1$ a constant. Let $\mathcal{TP} = (K, F, F^{-1})$ be a family of trap-door permutations, and A an adversary. We consider the following experiment:*

```
Experiment $\mathbf{Exp}^{\theta\text{-pow}}_{\mathcal{TP},A}(k)$
  $(pk, sk) \leftarrow K(1^k)$; $x \overset{R}{\leftarrow} \mathrm{Dom}(pk)$; $y \leftarrow F_{pk}(x)$
  $x_1 \leftarrow A(pk, y)$ where $|x_1| = \lceil \theta \cdot |x| \rceil$
  if ($F_{pk}(x_1||x_2) = y$ for some $x_2$) return 1 else return 0
```

Here, "$||$" denotes concatenation. We define the advantage of the adversary via

$$\mathbf{Adv}^{\theta\text{-pow}}_{\mathcal{TP},A}(k) = \Pr[\mathbf{Exp}^{\theta\text{-pow}}_{\mathcal{TP},A}(k) = 1]$$

where the probability is taken over K, $x \overset{R}{\leftarrow} \mathrm{Dom}(pk)$, and A. We say that $\mathcal{TP}$ is θ-partial one-way if the function $\mathbf{Adv}^{\theta\text{-pow}}_{\mathcal{TP},A}(k)$ is negligible for any poly-time adversary A.

Note that when $\theta = 1$ the notion of θ-partial one-wayness coincides with the standard notion of one-wayness.

3 Public-Key Encryption with Masking

In this section, we propose the definition of public-key encryption with masking and its security properties.

3.1 The Definition of Public-Key Encryption with Masking

We propose the definition of public-key encryption with masking as follows.

Definition 3. *A public-key encryption scheme $\mathcal{PEM} = (\mathcal{K}, \mathcal{EM}, \mathcal{D})$ with masking consists of three algorithms.*

- *The key generation algorithm $\mathcal{K}$ is a randomized algorithm that takes as input a security parameter 1^k and returns a pair (pk, sk) of keys, a public key and a matching secret key. We note that for given pk, the message space $\mathtt{MSPC}(pk)$ is uniquely determined.*

- *The encryption and masking algorithm $\mathcal{EM}$ is a randomized algorithm that takes a public key pk and a plaintext $m \in \mathtt{MSPC}(pk)$, and returns a masked ciphertext c and some unmasking information $\tilde{r}$. We call the pair $(c, \tilde{r})$ an unmasked ciphertext.*

- *The decryption algorithm $\mathcal{D}$ (for unmasked ciphertexts) is a deterministic algorithm that takes a secret key sk and an unmasked ciphertext $(c, \tilde{r})$, and returns the corresponding plaintext m or a special symbol $\perp$ to indicate that the unmasked ciphertext $(c, \tilde{r})$ is invalid.*

We require that, for any $k \in \mathbb{N}$, if $(pk, sk) \leftarrow \mathcal{K}(1^k)$, $m \in \mathtt{MSPC}(pk)$, and $(c, \tilde{r}) \leftarrow \mathcal{EM}_{pk}(m)$, then $m = \mathcal{D}_{sk}(c, \tilde{r})$.

It is easy to see that we can use a public-key encryption scheme with masking as a (standard) public-key encryption scheme if we always use a pair $(c, \tilde{r})$ for a standard ciphertext.

3.2 Security Properties of Public-Key Encryption with Masking

We define security properties with respect to public-key encryption with masking.

Indistinguishability of Masked Ciphertexts. First, we formalize the security notion called "indistinguishability of masked ciphertexts." This security notion captures the property that, given a masked ciphertext, not only the person who does not have a secret key but also the secret-key holder cannot get any information about the plaintext underlying the masked ciphertext. In the following definition, the adversary gets not only the public key but also the corresponding secret key.

Definition 4. *Let $\mathcal{PEM} = (\mathcal{K}, \mathcal{EM}, \mathcal{D})$ be a public-key encryption scheme with masking. Let $b \in \{0, 1\}$ and $k \in \mathbb{N}$. Let $A = (A_1, A_2)$ be an adversary that runs in two stages. Note that* si *is the state information. It contains pk, m_0, m_1, and so on. We consider the following experiment:*

```
Experiment $\mathbf{Exp}^{\text{ind-mc-}b}_{\mathcal{PEM},A}(k)$
  $(pk, sk) \leftarrow \mathcal{K}(1^k)$; $(m_0, m_1, \mathsf{si}) \leftarrow A_1(pk, sk)$; $(c, \tilde{r}) \leftarrow \mathcal{EM}_{pk}(m_b)$; $d \leftarrow A_2(c, \mathsf{si})$
  return $d$
```

Note that $m_0, m_1 \in \mathtt{MSPC}(pk)$. We define the advantage via

$$\mathbf{Adv}^{\text{ind-mc}}_{\mathcal{PEM},A}(k) = \left| \Pr[\mathbf{Exp}^{\text{ind-mc-1}}_{\mathcal{PEM},A}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ind-mc-0}}_{\mathcal{PEM},A}(k) = 1] \right|.$$

We say that a public-key encryption scheme $\mathcal{PEM}$ with masking meets IND-MC if the function $\mathbf{Adv}^{\text{ind-mc}}_{\mathcal{PEM},A}(k)$ is negligible for any poly-time adversary A.

Indistinguishability of Unmasked Ciphertexts. Second, we formalize the security notion called "indistinguishability of unmasked ciphertexts." This security notion captures the property that, given an unmasked ciphertext, the person who does not have a secret key cannot get any information about the plaintext underlying the unmasked ciphertext.

Definition 5. *Let $\mathcal{PEM} = (\mathcal{K}, \mathcal{EM}, \mathcal{D})$ be a public-key encryption scheme with masking. Let $b \in \{0, 1\}$ and $k \in \mathbb{N}$. Let $A^{\rm cpa} = (A_1^{\rm cpa}, A_2^{\rm cpa})$, $A^{\rm cca} = (A_1^{\rm cca}, A_2^{\rm cca})$ be adversaries that run in two stages and where $A^{\rm cca}$ has access to the decryption oracle $\mathcal{D}_{sk}(\cdot)$. For* atk $\in$ {cpa, cca}*, we consider the following experiment:*

```
Experiment $\mathbf{Exp}^{\text{ind-umc-atk-}b}_{\mathcal{PEM},A^{\rm atk}}(k)$
  $(pk, sk) \leftarrow \mathcal{K}(1^k)$; $(m_0, m_1, \mathsf{si}) \leftarrow A_1^{\rm atk}(pk)$; $(c, \tilde{r}) \leftarrow \mathcal{EM}_{pk}(m_b)$; $d \leftarrow A_2^{\rm atk}((c, \tilde{r}), \mathsf{si})$
  return $d$
```

Note that $m_0, m_1 \in \mathtt{MSPC}(pk)$. Above it is mandated that $A_2^{\rm cca}$ never queries the challenge $(c, \tilde{r})$ to the decryption oracle $\mathcal{D}_{sk}(\cdot)$. For atk $\in$ {cpa, cca}*, we define the advantage via*

$$\mathbf{Adv}^{\text{ind-umc-atk}}_{\mathcal{PEM},A^{\rm atk}}(k) = \left|\Pr[\mathbf{Exp}^{\text{ind-umc-atk-1}}_{\mathcal{PEM},A^{\rm atk}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ind-umc-atk-0}}_{\mathcal{PEM},A^{\rm atk}}(k) = 1]\right|.$$

We say that a public-key encryption scheme $\mathcal{PEM}$ with masking meets IND-UMC-CPA (respectively IND-UMC-CCA) if the function $\mathbf{Adv}^{\text{ind-umc-cpa}}_{\mathcal{PEM},A^{\rm cpa}}(k)$ (resp. $\mathbf{Adv}^{\text{ind-umc-cca}}_{\mathcal{PEM},A^{\rm cca}}(k)$) is negligible for any poly-time adversary $A^{\rm cpa}$ (resp. $A^{\rm cca}$).

The difference between the definitions of IND-MC and IND-UMC is as follows. The adversary in the IND-MC game has not only a public key but also a secret key, while the adversary in the IND-UMC game has only a public key. The adversary in the IND-UMC game gets the unmasking information $\tilde{r}$, while the adversary in the IND-MC game cannot get it. We cannot generally say that IND-MC implies IND-UMC, or vice versa.

Binding. Finally, we formalize the security notion called "binding." This security notion captures the property that the sender cannot produce one masked ciphertext c and two unmasking informations $\tilde{r}_0, \tilde{r}_1$ such that two unmasked ciphertexts, $(c, \tilde{r}_0)$ and $(c, \tilde{r}_1)$, are valid, and the two corresponding plaintexts are different.

Definition 6. *Let $\mathcal{PEM} = (\mathcal{K}, \mathcal{EM}, \mathcal{D})$ be a public-key encryption scheme with masking. Let $b \in \{0, 1\}$ and $k \in \mathbb{N}$. Let $A^{\rm cpa}$, $A^{\rm cca}$ be adversaries where $A^{\rm cca}$ has access to the decryption oracle $\mathcal{D}_{sk}(\cdot)$. For* atk $\in$ {cpa, cca}*, we consider the following experiment:*

```
Experiment $\mathbf{Exp}^{\text{bind-atk}}_{\mathcal{PEM},A^{\rm atk}}(k)$
  $(pk, sk) \leftarrow \mathcal{K}(1^k)$; $(c, \tilde{r}_0, \tilde{r}_1) \leftarrow A^{\rm atk}(pk)$; $m_0 \leftarrow \mathcal{D}_{sk}(c, \tilde{r}_0)$; $m_1 \leftarrow \mathcal{D}_{sk}(c, \tilde{r}_1)$
  if $((m_0 \neq \perp) \wedge (m_1 \neq \perp) \wedge (m_0, m_1 \in \mathtt{MSPC}(pk)) \wedge (m_0 \neq m_1))$
  then return $1$ else return $0$
```

For atk $\in$ {cpa, cca}*, we define the advantage via*

$$\mathbf{Adv}^{\text{bind-atk}}_{\mathcal{PEM},A^{\rm atk}}(k) = \Pr[\mathbf{Exp}^{\text{bind-atk}}_{\mathcal{PEM},A^{\rm atk}}(k) = 1].$$

We say that a public-key encryption scheme $\mathcal{PEM}$ with masking meets BIND-CPA (respectively BIND-CCA) if the function $\mathbf{Adv}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\rm cpa}}(k)$ (resp. $\mathbf{Adv}^{\text{bind-cca}}_{\mathcal{PEM},A^{\rm cca}}(k)$) is negligible for any poly-time adversary $A^{\rm cpa}$ (resp. $A^{\rm cca}$).

Remark 1. In our model, we only consider the situation that the sender first sends the masked ciphertext c to the receiver, and reveals the unmasking information $\tilde{r}$ later on. That is, we do not consider the situation that only the unmasking information is sent to the receiver (the secret-key holder). Therefore, in our formalization, we do not define the security notion "indistinguishability of unmasking informations" which captures that the unmasking information $\tilde{r}$ leaks some information about the plaintext to the secret-key holder.

Although we do not define such a security notion, the unmasking information $\tilde{r}$ of our proposed scheme in Section 5 does not leak any information about the plaintext, since $\tilde{r}$ is a random string and is independent of any other information.

Remark 2. We have defined the indistinguishability of masked ciphertexts and the binding. If the scheme satisfies these two security notions, it assures that the third party (i.e. neither the sender nor the receiver) who eavesdrops a masked ciphertext c cannot compute an unmasking information $\tilde{r}'$ such that $(c, \tilde{r}')$ is a valid unmasked ciphertext (i.e. $\mathcal{D}_{sk}(c, \tilde{r}') \in \mathtt{MSPC}(pk)$).

If the third party can compute the unmasking information $\tilde{r}'$ such that $\mathcal{D}_{sk}(c, \tilde{r}') = m$, then the scheme does not satisfy IND-MC. If the third party can compute an unmasking information $\tilde{r}'$ such that $\mathcal{D}_{sk}(c, \tilde{r}') = m' \neq m$ ($m' \in \mathtt{MSPC}(pk)$) in the CPA setting (respectively in the CCA setting), then the scheme does not satisfy BIND-CPA (resp. BIND-CCA).

4 The Weakness on Trivial Constructions

In this section, we consider the trivial constructions of public-key encryption with masking by simply applying the one-time pad to the plaintext or the ciphertext. Although they seem to be secure, we point out that these schemes do not satisfy the security properties.

Let Π be a standard public-key encryption scheme. It consists of three algorithms, that is, the key generation algorithm $\mathtt{Key}$, the encryption algorithm $\mathtt{Enc}$, and the decryption algorithm $\mathtt{Dec}$. For simplicity, we assume that the message space and the ciphertext space are $\{0,1\}^p$ and $\{0,1\}^q$, respectively.

4.1 The One-Time Pad for Ciphertexts

One standard way to mask the data is using the one-time pad. It seems good to apply the one-time pad to the ciphertext of a standard public-key encryption scheme. Thus, we can define the encryption algorithm $\mathcal{EM}_{pk}$ with masking as

$$\mathtt{Algorithm}\ \mathcal{EM}_{pk}(m):\ \tilde{r} \overset{R}{\leftarrow} \{0,1\}^q;\ c \leftarrow \mathtt{Enc}_{pk}(m) \oplus \tilde{r};\ \mathtt{return}\ (c, \tilde{r}).$$

The secret-key holder can decrypt the unmasked ciphertext $(c, \tilde{r})$ by computing $\mathtt{Dec}_{sk}(c \oplus \tilde{r})$.

It is easy to see that this scheme satisfies IND-MC, since $\tilde{r}$ is a perfect one-time pad. We can also prove that this scheme meets IND-UMC-CPA if Π satisfies IND-CPA.

However, this scheme does not meet IND-UMC-CCA even if Π meets IND-CCA. Suppose that the challenge for the adversary for the IND-UMC-CCA game is $(c, \tilde{r})$ where $c = \mathtt{Enc}_{pk}(m_b) \oplus \tilde{r}$. Then, the adversary can ask $(c \oplus \tilde{r}', \tilde{r} \oplus \tilde{r}')$ to the decryption oracle where $\tilde{r}' \in \{0,1\}^p$, and can get the plaintext m_b underlying the challenge unmasked ciphertext. Therefore, the adversary always wins the IND-UMC-CCA game.

Moreover, this scheme does not satisfy BIND-CPA even if Π satisfies IND-CCA. In the BIND-CCA game, the adversary chooses m_0, m_1 ($m_0 \neq m_1$) from $\{0,1\}^p$, and computes $c_0' \leftarrow \mathtt{Enc}_{pk}(m_0)$ and $c_1' \leftarrow \mathtt{Enc}_{pk}(m_1)$. Then, the adversary chooses $\tilde{r}_0 \in \{0,1\}^q$, computes $c \leftarrow c_0' \oplus \tilde{r}_0$ and $\tilde{r}_1 \leftarrow c_1' \oplus c$, and outputs $(c, \tilde{r}_0, \tilde{r}_1)$. Since $\mathtt{Dec}_{sk}(c \oplus \tilde{r}_0) = \mathtt{Dec}_{sk}(c_0') = m_0$ and $\mathtt{Dec}_{sk}(c \oplus \tilde{r}_1) = \mathtt{Dec}_{sk}(c_1') = m_1$, this adversary always wins the BIND-CPA game.

4.2 The One-Time Pad for Plaintexts

We can also apply the one-time pad to the message. That is, first we mask the message with some random string, then encrypt it. We can define the encryption algorithm $\mathcal{EM}_{pk}$ with masking as

$$\texttt{Algorithm } \mathcal{EM}_{pk}(m): \ \tilde{r} \overset{R}{\leftarrow} \{0,1\}^p; \ c \leftarrow \texttt{Enc}_{pk}(m \oplus \tilde{r}); \ \texttt{return } (c, \tilde{r}).$$

The secret-key holder can decrypt the unmasked ciphertext $(c, \tilde{r})$ by computing $\texttt{Dec}_{sk}(c) \oplus \tilde{r}$.

We can see that this scheme satisfies IND-MC, since $\tilde{r}$ is a perfect one-time pad. We can also prove that this scheme meets IND-UMC-CPA if Π satisfies IND-CPA.

However, this scheme does not meet IND-UMC-CCA even if Π meets IND-CCA. Suppose that the challenge for the adversary for the IND-UMC-CCA game is $(c, \tilde{r})$ where $c = \texttt{Enc}_{pk}(m_b \oplus \tilde{r})$. Then, the adversary can ask $(c, \tilde{r} \oplus \tilde{r}')$ to the decryption oracle where $\tilde{r}' \in \{0,1\}^p$. If the answer of the decryption oracle is m', this means that $m' = \texttt{Dec}_{sk}(c) \oplus \tilde{r}'$. Therefore, the adversary can compute m_b as $m_b = \texttt{Dec}_{sk}(c) \oplus \tilde{r} = m' \oplus \tilde{r}' \oplus \tilde{r}$, and can always win the IND-UMC-CCA game.

In addition, this scheme does not meet BIND-CPA even if Π meets IND-CCA, similar to the previous scheme. In the BIND-CCA game, the adversary chooses m_0, m_1 $(m_0 \neq m_1)$ from $\{0,1\}^p$ and $\tilde{r}_0 \in \{0,1\}^p$. Then, the adversary computes $\tilde{r}_1 = m_0 \oplus \tilde{r}_0 \oplus m_1$ and $c \leftarrow \texttt{Enc}_{pk}(m_0 \oplus \tilde{r}_0)$, and outputs $(c, \tilde{r}_0, \tilde{r}_1)$. Since $\texttt{Dec}_{sk}(c) \oplus \tilde{r}_0 = (m_0 \oplus \tilde{r}_0) \oplus \tilde{r}_0 = m_0$ and $\texttt{Dec}_{sk}(c) \oplus \tilde{r}_1 = (m_0 \oplus \tilde{r}_0) \oplus \tilde{r}_1 = m_1$, this adversary always wins the BIND-CPA game.

5 A Concrete Scheme based on OAEP

In this section, we propose a scheme based on OAEP for public-key encryption with masking, and prove its security.

5.1 Our Proposed Scheme

We now describe our proposed public-key encryption scheme with masking. We apply the one-time pad not to the plaintext or to the ciphertext of OAEP, but to the randomness of OAEP.

Definition 7. *Our proposed public-key encryption scheme $\mathcal{PEM} = (\mathcal{K}, \mathcal{EM}, \mathcal{D})$ with masking is as follows. Let k, k_0, and k_1 be security parameters such that $k_0 + k_1 < k$. This defines an associated plaintext-length $n = k - k_0 - k_1$. Let $\mathcal{TP} = (K, F, F^{-1})$ be a family of trap-door permutations such that $\mathrm{Dom}(pk) = \{0,1\}^k$ for any pk. The key generation algorithm $\mathcal{K}$ takes as input a security parameter 1^k, runs the key generation algorithm of $\mathcal{TP}$ as $(pk, sk) \leftarrow K(1^k)$, and outputs the public key pk and the secret key sk. The other algorithms are as follows. Let $G : \{0,1\}^{k_0} \to \{0,1\}^{k-k_0}$ and $H : \{0,1\}^{k-k_0} \to \{0,1\}^{k_0}$ be hash functions. Note that $[x]^{\ell}$ denotes the ℓ most significant bits of x, and $[x]_{\ell'}$ denotes the ℓ' least significant bits of x.*

```
Algorithm $\mathcal{EM}_{pk}(m)$
  $r, \tilde{r} \overset{R}{\leftarrow} \{0,1\}^{k_0}$
  $s \leftarrow (m||0^{k_1}) \oplus G(r \oplus \tilde{r})$
  $t \leftarrow r \oplus H(s)$
  $c \leftarrow F_{pk}(s||t)$
  return $(c, \tilde{r})$
```

```
Algorithm $\mathcal{D}_{sk}(c, \tilde{r})$
  $s \leftarrow [F_{sk}^{-1}(c)]^{n+k_1}$; $t \leftarrow [F_{sk}^{-1}(c)]_{k_0}$
  $r \leftarrow t \oplus H(s)$
  $m \leftarrow [s \oplus G(r \oplus \tilde{r})]^n$; $p \leftarrow [s \oplus G(r \oplus \tilde{r})]_{k_1}$
  if $(p = 0^{k_1})$ $z \leftarrow m$ else $z \leftarrow \perp$
  return $z$
```

We compute $G(r \oplus \tilde{r})$ in the encryption-and-masking algorithm and the decryption algorithm of our scheme, while $G(r)$ is computed in the encryption and decryption algorithms of OAEP.

5.2 Security

In the following, we show that our proposed scheme meets IND-MC, IND-UMC-CCA, and BIND-CCA.

First, we prove that our scheme provides IND-MC.

Theorem 1. *For any poly-time algorithm A making at most q_G queries to G, $\mathbf{Adv}^{\text{ind-mc}}_{\mathcal{PEM},A}(k) \leq 2q_G/2^{n+k_1}$.*

Proof. Assume that the challenge for the adversary is c where

$$b \xleftarrow{R} \{0,1\}, \quad r, \tilde{r} \xleftarrow{R} \{0,1\}^{k_0}, \quad s = (m_b||0^{k_1}) \oplus G(r \oplus \tilde{r}), \quad t = r \oplus H(s), \quad c = F_{pk}(s||t).$$

We define the event AskR as "the adversary makes a query $r^* \in \{0,1\}^{k_0}$ to G such that $(m_0||0^{k_1}) \oplus G(r^*) = s$ or $(m_1||0^{k_1}) \oplus G(r^*) = s$." Then the advantage of the adversary can be written as

$$\begin{aligned}
&\mathbf{Adv}^{\text{ind-mc}}_{\mathcal{PEM},A}(k) \\
&\quad = |\Pr[\mathsf{AskR}] \cdot (\Pr[\mathbf{Exp}^{\text{ind-mc-atk-1}}_{\mathcal{PEM},A}(k) = 1|\mathsf{AskR}] - \Pr[\mathbf{Exp}^{\text{ind-mc-atk-0}}_{\mathcal{PEM},A}(k) = 1|\mathsf{AskR}]) \\
&\qquad + \Pr[\neg\mathsf{AskR}] \cdot (\Pr[\mathbf{Exp}^{\text{ind-mc-atk-1}}_{\mathcal{PEM},A}(k) = 1|\neg\mathsf{AskR}] - \Pr[\mathbf{Exp}^{\text{ind-mc-atk-0}}_{\mathcal{PEM},A}(k) = 1|\neg\mathsf{AskR}])| \\
&\quad \leq \Pr[\mathsf{AskR}] + |\Pr[\mathbf{Exp}^{\text{ind-mc-atk-1}}_{\mathcal{PEM},A}(k) = 1|\neg\mathsf{AskR}] - \Pr[\mathbf{Exp}^{\text{ind-mc-atk-0}}_{\mathcal{PEM},A}(k) = 1|\neg\mathsf{AskR}]|.
\end{aligned}$$

First, we evaluate the probability $\Pr[\mathsf{AskR}]$. Since F is a permutation and the adversary has a secret-key sk, she knows the values s and t such that $c = F_{pk}(s||t)$, and she can compute $r = H(s) \oplus t$. Furthermore, the adversary knows m_0 and m_1.

However, the values $(pk, sk), m_0, m_1, s, t, r$ does not restrict the range of r^* such that $(m_0||0^{k_1}) \oplus G(r^*) = s$ or $(m_1||0^{k_1}) \oplus G(r^*) = s$. That is, even if the adversary knows the values $(pk, sk), m_0, m_1, s, t, r$, the probability that the adversary asks $r' \in \{0,1\}^{k_0}$ to G such that $(m_0||0^{k_1}) \oplus G(r') = s$ or $(m_1||0^{k_1}) \oplus G(r') = s$ is $2/2^{n+k_1}$, since G is the random oracle. Therefore, $\Pr[\mathsf{AskR}] \leq 2q_G/2^{n+k_1}$.

Next, we consider the situation that $\neg\mathsf{AskR}$ holds. If $\neg\mathsf{AskR}$ holds, then the adversary does not know the value $G(r \oplus \tilde{r})$ which was used for computing the challenge. Then, the adversary cannot gain any advantage in the experiment without asking $r \oplus \tilde{r}$ to G. Since it does not depend on the value $b \in \{0,1\}$, we have

$$\Pr[\mathbf{Exp}^{\text{ind-mc-atk-1}}_{\mathcal{PEM},A}(k) = 1|\neg\mathsf{AskR}] = \Pr[\mathbf{Exp}^{\text{ind-mc-atk-0}}_{\mathcal{PEM},A}(k) = 1|\neg\mathsf{AskR}] = \frac{1}{2}.$$

Hence, we have $\mathbf{Adv}^{\text{ind-mc}}_{\mathcal{PEM},A}(k) \leq 2q_G/2^{n+k_1}$ for any poly-time adversary A. □

Second, we prove that our scheme provides IND-UMC-CCA.

Theorem 2. *For any adversary A attacking the IND-UMC-CCA security of our scheme $\mathcal{PEM}$ with $\mathcal{TP}$, and making at most q_D queries to decryption oracle, q_G G-oracle queries, and q_H H-oracle queries, there exists a θ-partial inverting adversary B for $\mathcal{TP}$, such that for any k, k_0, k_1, and $\theta = \frac{k-k_0}{k}$,*

$$\mathbf{Adv}^{\text{ind-umc-cca}}_{\mathcal{PEM},A}(k) \leq \frac{q_G + q_D + q_D q_G}{2^{k_0-1}} + \frac{q_D}{2^{k_1-1}} + 2q_H \cdot \mathbf{Adv}^{\theta\text{-pow}}_{\mathcal{TP},B}(k)$$

and the running time of B is that of A plus $q_D \cdot q_G \cdot q_H \cdot (T_F + O(1))$ where T_F denotes the time for evaluating the permutation F.

Proof. The proof of the IND-UMC-CCA security for our scheme is similar to that of the IND-CCA security for OAEP by Fujisaki, Okamoto, Pointcheval, and Stern [7]. We define a sequence Game_1, Game_2, etc., of modified attack games starting from the actual game Game_0. Each of the games operates on the same underlying probability space: the public and secret keys of the cryptosystems, the coin tosses of the adversary A, the random oracles G and H and the hidden bit b for the challenge.

In the following, all variables with asterisk refer to the challenge unmasked ciphertext, and all variables with no asterisk refer to the decryption queries.

Game_0. A pair of keys (pk, sk) is generated by $\mathcal{K}(1^k)$, and the adversary A_1 takes pk and outputs two messages (m_0, m_1) and the state information si. Then, the adversary A_2 takes si and the challenge unmasked ciphertexts $(c^*, \tilde{r}^*)$ where $b \overset{R}{\leftarrow} \{0,1\}$ and

$$r^*, \tilde{r}^* \overset{R}{\leftarrow} \{0,1\}^{k_0}, \quad s^* = (m_b||0^{k_1}) \oplus G(r^* \oplus \tilde{r}^*), \quad t^* = r^* \oplus H(s^*), \quad c^* = F_{pk}(s^*||t^*),$$

and A_2 outputs d. In the above experiment, the adversary A can make access to the random oracles G, H, and the decryption oracle $\mathcal{D}_{sk}$. However, A_2 cannot ask the challenge unmasked ciphertext $(y^*, \tilde{r}^*)$ to the decryption oracle.
We denote by S_0 the event "$d = b$" and use a similar notation S_i in any Game_i below. By definition, we have $\Pr[S_0] = 1/2 + \epsilon/2$ where $\epsilon = \mathbf{Adv}^{\text{ind-umc-cca}}_{\mathcal{PEM},A}(k)$.

Game_1. We choose three random values $r^+ \overset{R}{\leftarrow} \{0,1\}^{k_0}$, $\tilde{r}^+ \overset{R}{\leftarrow} \{0,1\}^{k_0}$, and $g^+ \overset{R}{\leftarrow} \{0,1\}^{k-k_0}$ in advance (i.e. before the adversary A_1 runs), and use r^+, $\tilde{r}^+$, and g^+, instead of r^*, $\tilde{r}^*$, and $G(r^* \oplus \tilde{r}^*)$ respectively. In Game_1, we apply the following special rules.

R1: We compute the challenge unmasked ciphertext $(c^*, \tilde{r}^*)$ by setting

$$r^* \leftarrow r^+, \ \tilde{r}^* \leftarrow \tilde{r}^+, \text{ and } s^* \leftarrow (m_b||0^{k_1}) \oplus g^+.$$

R2: Whenever the random oracle G is queried at $r^+ \oplus \tilde{r}^+$, the answer is g^+.

Since we replace a triplet of elements $(r^*, \tilde{r}^*, G(r^* \oplus \tilde{r}^*))$ by a different, but identically distributed (by the definition of the random oracle G), set of random variables, we have $\Pr[S_1] = \Pr[S_0]$.

Game_2. In this game, we drop the rule **R2** from Game_1. Therefore, g^+ is just used for computing the challenge unmasked ciphertext, and if $r^* \oplus \tilde{r}^*$ is queried to G then we respond not g^+ but $G(r^* \oplus \tilde{r}^*)$ by using the random oracle G. Then, g^+ is never revealed to the adversary and the input $(c^*, \tilde{r}^*)$ to A_2 follows a distribution that does not depend on b. Therefore, we have $\Pr[S_2] = 1/2$.
One may note that Game_1 and Game_2 may differ if $r^* \oplus \tilde{r}^*$ is queried to G. Let AskG_2 denotes the event that, in Game_2, $r^* \oplus \tilde{r}^*$ is queried to G (except by the encryption oracle, for producing the challenge). We use an identical notation AskG_i for any Game_i below. Then, $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[\mathsf{AskG}_2]$.

Game_3. We now define s^* independently of anything else, as well as $H(s^*)$. We choose two random values $s^+ \overset{R}{\leftarrow} \{0,1\}^{k-k_0}$ and $h^+ \overset{R}{\leftarrow} \{0,1\}^{k_0}$ in advance (i.e. before the adversary A_1 runs), and use s^+ instead of s^*, as well as h^+ instead of $H(s^*)$. In Game_3, we apply the following special rules. Note that we change the way to compute g^+ (but identically distributed as that in Game_2).

R1′: We compute the challenge unmasked ciphertext $(c^*, \tilde{r}^*)$ by setting

$$s^* \leftarrow s^+, \quad g^+ \leftarrow (m_b||0^{k_1}) \oplus s^+, \quad t^* \leftarrow r^* \oplus h^+.$$

R2′: Whenever the random oracle H is queried at s^+, the answer is h^+.

Since we replace the set of elements $(s^*, H(s^*), g^+, b)$ by a different, but identically distributed (by the definition of the random oracle H), set of random variables, we have $\Pr[\mathsf{AskG}_3] = \Pr[\mathsf{AskG}_2]$.

Game_4. In this game, we drop the rule **R2′** from Game_3. Therefore, h^+ is just used for computing the challenge unmasked ciphertext, and if s^* is queried to H then we respond not h^+ but $H(s^*)$ by using the random oracle H.
One may note that Game_3 and Game_4 may differ if s^* is queried to H. Let AskH_4 denotes the event that, in Game_4, s^* is queried to H (except by the encryption oracle, for producing the challenge). We use an identical notation AskH_i for any Game_i below. Then, $|\Pr[\mathsf{AskG}_4] - \Pr[\mathsf{AskG}_3]| \leq \Pr[\mathsf{AskH}_4]$.
Furthermore, $r^* = t^* \oplus h^+$ is uniformly distributed, and independent of the adversary's view, since h^+ is never revealed. Therefore, $r^* \oplus \tilde{r}^*$ is also independent of the adversary's view, and we have $\Pr[\mathsf{AskG}_4] \leq (q_G + q_D)/2^{k_0}$, where q_G and q_D denote the number of queries asked to G and that asked to the decryption oracle, respectively.

Game_5. In Game_5, in order to evaluate AskH_4, we again modify the previous game. That is, when manufacturing the challenge unmasked ciphertext, we randomly choose $c^+ \overset{R}{\leftarrow} \{0,1\}^k$, and simply set $c^* \leftarrow c^+$, ignoring the encryption oracle altogether.
Since F is a permutation, and $s^* = s^+$ and $t^* = h^+ \oplus r^+$ are uniformly distributed over $\{0,1\}^{k-k_0}$ and $\{0,1\}^{k_0}$, respectively, the distribution of $c^* = F_{pk}(s^*||t^*)$ is the same as that of c^+. Thus, we have $\Pr[\mathsf{AskH}_5] = \Pr[\mathsf{AskH}_4]$.

In the following, we deal with the random oracles and the decryption oracle.

Game_6. In this game, we do not use the random oracles G, H, and simulating these oracles. We use two lists, G-List and H-List, for simulating the random oracles G and H, respectively, both are initially set to empty list.

- When the adversary or the decryption oracle makes a query $\gamma \in \{0,1\}^{k_0}$ to G, if there exist a pair $(\gamma, G_\gamma) \in G$-List then we respond G_γ. Otherwise, we respond a random string $G_\gamma \overset{R}{\leftarrow} \{0,1\}^{k-k_0}$ and put (γ, G_γ) into the G-List.
- When the adversary or the decryption oracle makes a query $\delta \in \{0,1\}^{k-k_0}$ to H, if there exist a pair $(\delta, H_\delta) \in H$-List then we respond H_δ. Otherwise, we respond a random string $H_\delta \overset{R}{\leftarrow} \{0,1\}^{k_0}$ and put (δ, H_δ) into the H-List.

Since we can simulate the random oracles perfectly, we have $\Pr[\mathsf{AskH}_6] = \Pr[\mathsf{AskH}_5]$.

Game_7. We make the decryption oracle reject any unmasked ciphertext $(c, \tilde{r})$ such that the corresponding value $r \oplus \tilde{r}$ has not been previously queried to G by the adversary (i.e. there exists no element $(\gamma, G_\gamma) \in G$-List such that $(r \oplus \tilde{r}) = \gamma$). This makes a difference only if $(c, \tilde{r})$ is a valid unmasked ciphertext, while $G(r \oplus \tilde{r})$ has not been asked. Since $G(r \oplus \tilde{r})$ is uniformly distributed, the equation $[s \oplus G((r \oplus \tilde{r}))]_{k_1} = 0^{k_1}$ holds with probability $1/2^{k_1}$. Summing up for all decryption queries, we get $|\Pr[\mathsf{AskH}_7] - \Pr[\mathsf{AskH}_6]| \leq q_D/2^{k_1}$.

Game_8. We now make the decryption oracle reject any unmasked ciphertext $(c, \tilde{r})$ such that the corresponding value s has not been previously queried to H by the adversary. (i.e. there exists no element $(\delta, H_\delta) \in H$-List such that $s = \delta$). This makes a difference only if $(c, \tilde{r})$ is a valid unmasked ciphertext, and $r \oplus \tilde{r}$ has been queried to G, while $H(s)$ has not been asked. Since $r = H(s) \oplus t$ is uniformly distributed, $r \oplus \tilde{r}$ is also uniformly distributed. Thus, $r \oplus \tilde{r}$ has been queried to G with probability less than $q_G/2^{k_0}$ (note that in the previous game, the decryption oracle makes no additional query to G). Summing up for all decryption queries, we get $|\Pr[\mathsf{AskH}_8] - \Pr[\mathsf{AskH}_7]| \le q_D q_G / 2^{k_0}$.

Game_9. We replace the decryption oracle by the plaintext extractor described as follows.

Plaintext extractor. The plaintext extractor takes an unmasked ciphertext $(c, \tilde{r})$ and two lists, G-List and H-List. Then, for each $(\gamma, G_\gamma) \in G$-List and $(\delta, H_\delta) \in H$-List, the plaintext extractor checks whether

$$c = F_{pk}(\delta || (\gamma \oplus \tilde{r} \oplus H_\delta)) \text{ and } [\delta \oplus G_\gamma]_{k_1} = 0^{k_1}.$$

If both equations hold, the plaintext extractor outputs $[\delta \oplus G_\gamma]^n$. If no such pair is found, the plaintext extractor outputs $\perp$.

If the adversary has made queries $r \oplus \tilde{r}$ and s to G and H, respectively, then the plaintext extractor can decrypt the unmasked ciphertext correctly. Therefore, $\Pr[\mathsf{AskH}_9] = \Pr[\mathsf{AskH}_8]$.

We now construct an algorithm B attacking θ-partial one-wayness of $\mathcal{TP}$ by using A against Game_9.

1. B takes pk and y^* where $y^* = F_{pk}(x^*)$ and $x^* \xleftarrow{R} \{0,1\}^k$, and runs A against Game_9 where $c^* \leftarrow y^*$.
2. When A terminates, B outputs $\delta' \xleftarrow{R} \{\delta | (\delta, H_\delta) \in H\text{-List}\}$.

Note that B simulates the random oracles and the decryption oracle for A as in Game_9 (by using G-List, H-List, and the plaintext extractor). We also note that the distribution of c^* in Game_9 is the same as that of y^*. If AskH_9 holds then there exists an element $s^* = [x^*]^{k-k_0}$ in H-List such that $F_{pk}(x^*) = y^*$. Therefore, we have $\Pr[\mathsf{AskH}_9] \le q_H \cdot \mathbf{Adv}^{\theta\text{-pow}}_{\mathcal{TP},B}(k)$.

In conclusion, we have

$$\begin{aligned}
\frac{1}{2} \cdot \mathbf{Adv}^{\text{ind-umc-cca}}_{\mathcal{PEM},A}(k) &= |\Pr[S_2] - \Pr[S_0]| \le \Pr[\mathsf{AskG}_2] \le \Pr[\mathsf{AskG}_4] + \Pr[\mathsf{AskH}_4] \\
&\le \frac{q_G + q_D}{2^{k_0}} + \Pr[\mathsf{AskH}_6] \le \frac{q_G + q_D}{2^{k_0}} + \frac{q_D}{2^{k_1}} + \Pr[\mathsf{AskH}_7] \\
&\le \frac{q_G + q_D}{2^{k_0}} + \frac{q_D}{2^{k_1}} + \frac{q_D q_G}{2^{k_0}} + \Pr[\mathsf{AskH}_8] \\
&\le \frac{q_G + q_D + q_G q_G}{2^{k_0}} + \frac{q_D}{2^{k_1}} + q_H \cdot \mathbf{Adv}^{\theta\text{-pow}}_{\mathcal{TP},B}(k).
\end{aligned}$$

We now estimate the running time of B. It is the running time of A plus that of the plaintext extractor. For each decryption query $(c, \tilde{r})$, the plaintext extractor has to look at all pairs $(\gamma, G_\gamma) \in G$-List, and $(\delta, H_\delta) \in H$-List, and to compute $F_{pk}(\delta || (\gamma \oplus \tilde{r} \oplus H_\delta))$. Therefore, the running time of B is bounded by that of A plus $q_D \cdot q_G \cdot q_H \cdot (T_F + O(1))$ where T_F denotes the time for evaluating the permutation F. □

Remark 3. The reduction cost of our scheme in the proof of Theorem 2 and that of OAEP in [7] are different with respect to the running time. This is because the difference of the running time of the plaintext extractor.

In the proof of the IND-CCA2 security for OAEP, Fujisaki, Okamoto, Pointcheval, and Stern defined the plaintext extractor for OAEP. In order to run the plaintext extractor for OAEP, it is sufficient to compute the value $F_{pk}(\delta || (\gamma \oplus H_\delta))$ for every $(\gamma, G_\gamma) \in G$-List and $(\delta, H_\delta) \in H$-List. Therefore, the running time of the plaintext extractor is bounded by $q_G \cdot q_H \cdot (T_F + O(1))$ where T_F is the time for evaluating the trap-door permutation.

In the proof for our scheme, in order to run the plaintext extractor for our scheme, we have to compute $F_{pk}(\delta||(\gamma \oplus \tilde{r} \oplus H_\delta))$ for every $(\gamma, G_\gamma) \in G$-List, $(\delta, H_\delta) \in H$-List, and $\tilde{r}$ queried to the decryption oracle. Thus, the running time of the plaintext extractor is bounded by $q_D \cdot q_G \cdot q_H \cdot (T_F + O(1))$.

Finally, we show that our scheme provides BIND-CCA.

Theorem 3. *For any poly-time algorithm A^{cca} making at most q_G queries to G, and q_D queries to the decryption oracle,*

$$\mathbf{Adv}^{\text{bind-cca}}_{\mathcal{PEM},A^{\text{cca}}}(k) \le \frac{q_G(q_G-1)+1+q_D}{2^{k_1}} + \frac{q_D q_G}{2^{k_0}}.$$

In order to prove this theorem, we first show that our scheme meets BIND-CPA. Then, we show that if our scheme meets BIND-CPA then it also meets BIND-CCA.

First, we show that our scheme meets BIND-CPA.

Lemma 1. *For any poly-time algorithm A^{cpa} making at most q_G queries to G, $\mathbf{Adv}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k) \le (q_G(q_G-1)+1)/2^{k_1}$.*

Proof. Assume that the output of the adversary is $(c, \tilde{r}_0, \tilde{r}_1)$. Then, the values s, t, r such that $c = F_{pk}(s||t)$ and $r = t \oplus H(s)$ are uniquely determined since F is a permutation. Therefore, the adversary wins the game if and only if these values satisfy

$$(m_0||0^{k_1}) \oplus G(r \oplus \tilde{r}_0) = (m_1||0^{k_1}) \oplus G(r \oplus \tilde{r}_1) = s$$

for some $m_0, m_1 \in \mathtt{MSPC}(pk)$ $(m_0 \ne m_1)$.

We define the event $\mathsf{AskR2}$ as "the adversary makes two queries $r^*, r^{**} \in \{0,1\}^{k_0}$ to G such that $(m_0||0^{k_1}) \oplus G(r^*) = (m_1||0^{k_1}) \oplus G(r^{**}) = s$ for some $m_0, m_1 \in \mathtt{MSPC}(pk)$ $(m_0 \ne m_1)$." Then the advantage of the adversary can be written as

$$\begin{aligned}
&\mathbf{Adv}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k) \\
&\quad = \Pr[\mathsf{AskR2}] \cdot \Pr[\mathbf{Exp}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k) = 1|\mathsf{AskR2}] + \Pr[\neg\mathsf{AskR}] \cdot \Pr[\mathbf{Exp}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k) = 1|\neg\mathsf{AskR2}] \\
&\quad \le \Pr[\mathsf{AskR2}] + \Pr[\mathbf{Exp}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k) = 1|\neg\mathsf{AskR2}].
\end{aligned}$$

First, we evaluate the probability $\Pr[\mathsf{AskR2}]$. In order to satisfy $(m_0||0^{k_1}) \oplus G(r^*) = (m_1||0^{k_1}) \oplus G(r^{**}) = s$ for some $m_0 \ne m_1$, it is necessary that $r^* \ne r^{**}$ and the k_1 least significant bits of $G(r^*)$ are equal to those of $G(r^{**})$.

Assume that the adversary makes q_G queries $g_1, \cdots, g_{q_G}$ to G. Without loss of generality, we assume that $g_j \ne g_{j'}$ for any $j, j' \in \{1, \cdots, q_G\}$.

Then, for every $j \in \{2, \cdots, q_G\}$, we have that $\Pr[$the k_1 least significant bits of $G(g_j)$ are equal to those of $G(g_i)$ for some $0 \le i < j] \le (j-1)/2^{k_1}$, since G is the random oracle. Therefore, $\Pr[\mathsf{AskR2}] \le (1 + 2 + \cdots + (q_G - 1))/2^{k_1} = q_G(q_G-1)/2^{k_1}$.

Next, we consider the situation that $\neg\mathsf{AskR2}$ holds. We have

$$\begin{aligned}
&\Pr[\mathbf{Exp}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k) = 1|\neg\mathsf{AskR2}] \\
&\quad = \Pr[(m_0||0^{k_1}) \oplus G(r \oplus \tilde{r}_0) = (m_1||0^{k_1}) \oplus G(r \oplus \tilde{r}_1) = s \\
&\qquad\qquad \text{for some } m_0, m_1 \in \mathtt{MSPC}(pk)\ (m_0 \ne m_1)|\neg\mathsf{AskR2}] \\
&\quad \le \Pr[\text{the } k_1 \text{ least significant bits of } G(r \oplus \tilde{r}_0) \text{ are equal to those of } G(r \oplus \tilde{r}_1)|\neg\mathsf{AskR2}].
\end{aligned}$$

If $\neg\mathsf{AskR2}$ holds, the adversary does not know either the value $G(r \oplus \tilde{r}_0)$ or the value $G(r \oplus \tilde{r}_1)$ where $(c, \tilde{r}_0, \tilde{r}_1)$ is the output of the adversary. Therefore, the probability that the k_1 least significant bits of $G(r \oplus \tilde{r}_0)$ are equal to those of $G(r \oplus \tilde{r}_1)$ is bounded by $1/2^{k_1}$, since G is the random oracle. Therefore, $\Pr[\mathbf{Exp}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k) = 1|\neg\mathsf{AskR2}] \le 1/2^{k_1}$.

Hence, we have $\mathbf{Adv}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k) \le q_G(q_G-1)/2^{k_1} + 1/2^{k_1} = (q_G(q_G-1)+1)/2^{k_1}$ for any poly-time adversary A^{cpa}. □

Next, we show that if our scheme meets BIND-CPA then it also meets BIND-CCA.

Lemma 2. *For any adversary A^{cca} attacking the BIND-CCA security of our scheme $\mathcal{PEM}$, and making at most q_D queries to decryption oracle, q_G G-oracle queries, and q_H H-oracle queries, there exists a BIND-CPA adversary A^{cpa} of $\mathcal{PEM}$ making at most q_G G-oracle queries and q_H H-oracle queries, such that for any k, k_0, k_1,*

$$\mathbf{Adv}^{\text{bind-cca}}_{\mathcal{PEM},A^{\mathrm{cca}}}(k) \le \frac{q_D}{2^{k_1}} + \frac{q_D q_G}{2^{k_0}} + \mathbf{Adv}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\mathrm{cpa}}}(k)$$

and the running time of A^{cpa} is that of A^{cca} plus $q_D \cdot q_G \cdot q_H \cdot (T_F + O(1))$ where T_F denotes the time for evaluating the permutation F.

Proof. The proof is similar to that for Theorem 2. We define a sequence Game'_1, Game'_2, and Game'_3, of modified attack games starting from the actual game Game'_0. Each of the games operates on the same underlying probability space: the public and secret keys of the cryptosystems, the coin tosses of the adversary A^{cca}, the random oracles G and H.

We define two lists, G-List and H-List. G-List contains of all pairs (γ, G_γ) where γ is a query to G by the adversary, and G_γ is the corresponding answer of G. Similarly, H-List contains of all pairs (δ, H_δ) where δ is a query to H by the adversary, and H_δ is the corresponding answer of H.

In the following, all variables with asterisk refer to the output of the adversary, and all variables with no asterisk refer to the decryption queries.

Game'_0. A pair of keys (pk, sk) is generated by $\mathcal{K}(1^k)$, and the adversary A^{cca} takes pk and outputs $(c^*, \tilde{r}^*_0, \tilde{r}^*_1)$. In the above experiment, the adversary A^{cca} can make access to the random oracles G, H, and the decryption oracle $\mathcal{D}_{sk}$.
We denote by S'_0 the event "$(m^*_0 \ne \perp) \wedge (m^*_1 \ne \perp) \wedge (m^*_0, m^*_1 \in \mathtt{MSPC}(pk)) \wedge (m^*_0 \ne m^*_1)$" where $m^*_0 \leftarrow \mathcal{D}_{sk}(c^*, \tilde{r}^*_0)$ and $m^*_1 \leftarrow \mathcal{D}_{sk}(c^*, \tilde{r}^*_1)$. We use a similar notation S'_i in any Game'_i below. By definition, we have $\Pr[S'_0] = \mathbf{Adv}^{\text{bind-cca}}_{\mathcal{PEM},A^{\mathrm{cca}}}(k)$.

Game'_1. We make the decryption oracle reject any unmasked ciphertext $(c, \tilde{r})$ such that the corresponding value $r \oplus \tilde{r}$ has not been previously queried to G by the adversary (i.e. there exists no element $(\gamma, G_\gamma) \in G$-List such that $(r \oplus \tilde{r}) = \gamma$). By the similar discussion as that in Game_7 in the proof of Theorem 2, we have $|\Pr[S'_1] - \Pr[S'_0]| \le q_D/2^{k_1}$.

Game'_2. We now make the decryption oracle reject any unmasked ciphertext $(c, \tilde{r})$ such that the corresponding value s has not been previously queried to H by the adversary. (i.e. there exists no element $(\delta, H_\delta) \in H$-List such that $s = \delta$). By the similar discussion as that in Game_8 in the proof of Theorem 2, we have $|\Pr[S'_2] - \Pr[S'_1]| \le q_D q_G/2^{k_0}$.

Game'_3. We replace the decryption oracle by the plaintext extractor. The definition of the plaintext extractor is the same as that in Game_9 in the proof of Theorem 2. By the similar discussion as that in Game_9 in the proof of Theorem 2, we have $\Pr[S'_3] = \Pr[S'_2]$.

We now construct an algorithm A^{cpa} attacking the BIND-CPA security by using A^{cca} against Game'_3.

1. A^{cpa} takes pk and runs $A^{\mathrm{cca}}(pk)$ against Game'_3 where,

   - if A^{cca} makes a query γ to G, A^{cpa} makes a query γ to its own oracle G, and gets an answer G_γ. Then, A^{cpa} responds G_γ to A^{cca} and puts (γ, G_γ) to G-List, and similarly, A^{cpa} responds H-oracle query from A^{cca} and make H-List.

2. When A^{cca} outputs $(c, \tilde{r}_0, \tilde{r}_1)$ then A^{cpa} outputs $(c, \tilde{r}_0, \tilde{r}_1)$.

Note that A^{cpa} simulates the decryption oracle for A^{cca} as in Game'_3 (by using G-List, H-List, and the plaintext extractor). We can easily see that $\Pr[S'_3] = \mathbf{Adv}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k)$.

In conclusion, we have

$$\begin{aligned}&|\mathbf{Adv}^{\text{bind-cca}}_{\mathcal{PEM},A^{\text{cca}}}(k) - \mathbf{Adv}^{\text{bind-cpa}}_{\mathcal{PEM},A^{\text{cpa}}}(k)|\\ &\quad = |\Pr[S'_3] - \Pr[S'_0]| \le |\Pr[S'_2] - \Pr[S'_1]| + |\Pr[S'_1] - \Pr[S'_0]| \le \frac{q_D}{2^{k_1}} + \frac{q_D q_G}{2^{k_0}}.\end{aligned}$$

We can bound the running time of A^{cpa} by that of A^{cca} plus $q_D \cdot q_G \cdot q_H \cdot (T_F + O(1))$ where T_F denotes the time for evaluating the permutation F, by the similar discussion as that in the proof of Theorem 2. □

From Lemmas 1 and 2, we get the claimed result in Theorem 3.

References

[1] Bellare, M., and Goldwasser, S. Encapsulated Key Escrow. Technical Report MIT-LCS-TR-688, Massachusetts Institute of Technology, 1996. Online available at http://www.lcs.mit.edu/publications/pubs/pdf/MIT-LCS-TR-688.pdf.

[2] Bellare, M., and Rogaway, P. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Advances in Cryptology – EUROCRYPT '94* (Perugia, Italy, May 1994), A. De Santis, Ed., vol. 950 of *LNCS*, Springer-Verlag, pp. 92–111.

[3] Blake, I. F., and Chan, A. C.-F. Scalable, Server-Passive, User-Anonymous Timed Release Public Key Encryption from Bilinear Pairing. IACR Cryptology ePrint Archive, http://eprint.iacr.org/2004/211.pdf, 2004.

[4] Boneh, D., and Naor, M. Timed Commitments. In *Advances in Cryptology – CRYPTO 2000* (Santa Barbara, California, USA, August 2000), M. Bellare, Ed., vol. 1880 of *LNCS*, Springer-Verlag, pp. 236–254.

[5] Cheon, J. H., Hopper, N., Kim, Y., and Osipkov, I. Authenticated Key-Insulated Public-Key Encryption and Time-Release Cryptography. In *Financial Cryptography – FC 2006* (Anguilla, British West Indies, February 2006).

[6] Fujisaki, E., Okamoto, T., Pointcheval, D., and Stern, J. RSA-OAEP is Secure under the RSA Assumption. In *Advances in Cryptology – CRYPTO 2001* (Santa Barbara, California, USA, August 2001), J. Kilian, Ed., vol. 2139 of *LNCS*, Springer-Verlag, pp. 260–274.

[7] Fujisaki, E., Okamoto, T., Pointcheval, D., and Stern, J. RSA-OAEP is Secure under the RSA Assumption. *Journal of Cryptology 17*, 2 (2004), 81–104.

[8] Garay, J. A., and Jakobsson, M. Timed Release of Standard Digital Signatures. In *Financial Cryptography – FC 2002* (Southampton, Bermuda, March 2002), M. Blaze, Ed., vol. 2357 of *LNCS*, Springer-Verlag, pp. 168–182.

[9] Garay, J. A., and Pomerance, C. Timed Fair Exchange of Standard Signatures. In *Financial Cryptography – FC 2003* (Guadeloupe, French West Indies, January 2003), R. N. Wright, Ed., vol. 2742 of *LNCS*, Springer-Verlag, pp. 190–207.

[10] Hwang, Y. H., Yum, D. H., and Lee, P. J. Timed-Release Encryption with Pre-open Capability and Its Application to Certified E-mail System. In Zhou et al. [18], pp. 344–358.

[11] Mao, W. Timed-Release Cryptography. In *Selected Areas in Cryptography (SAC 2001)* (Toronto, Ontario, Canada, August 2001), S. Vaudenay and A. M. Youssef, Eds., vol. 2259 of *LNCS*, Springer-Verlag, pp. 342–358.

[12] May, T. Timed-Release Crypto. manuscript, 1993.

[13] Mont, M., Harrison, K., and Sadler, M. The HP Time Vault Service: Innovating the way confidential information is disclosed at the Right Time. HP Lab. Report, `http://www.hpl.hp.com/techreports/2002/HPL-2002-243.pdf`, 2002.

[14] Nali, D., Adams, C. M., and Miri, A. Time-Based Release of Confidential Information in Hierarchical Settings". In Zhou et al. [18], pp. 29–43.

[15] Rivest, R. L., Shamir, A., and Wagner, D. A. Time-lock puzzles and timed-release crypto. Technical Report MIT/LCS/TR-684, Massachusetts Institute of Technology, 1996. Online available at `http://theory.lcs.mit.edu/ rivest/RivestShamirWagner-timelock.ps`.

[16] Yoshida, M., Mitsunari, S., and Fujiwara, T. A Timed-Release Key Management Scheme for Backward Recovery. In *Information Security and Cryptology - ICISC 2005 8th International Conference* (Seoul, Korea, December 2005), D. Won and S. Kim, Eds., vol. 3935 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 3–14.

[17] Zhang, R., Hanaoka, G., Shikata, J., and Imai, H. On the Security of Multiple Encryption or CCA-security + CCA-security = CCA-security? In *PKC 2004 – 7th International Workshop on Theory and Practice in Public Key Cryptography* (Singapore, March 2004), F. Bao, R. H. Deng, and J. Zhou, Eds., vol. 2947 of *LNCS*, Springer-Verlag, pp. 360–374.

[18] Zhou, J., Lopez, J., Deng, R. H., and Bao, F., Eds. *Information Security, 8th International Conference, ISC 2005* (Singapore, September 2005), vol. 3650 of *Lecture Notes in Computer Science*, Springer.