# Davies-Meyer Merkle-Damgård Revisited: Variants of Indifferentiability and Random Oracles

Yusuke Naito[1], Kazuki Yoneyama[2], Lei Wang[2], and Kazuo Ohta[2]

[1] Mitsubishi Electoric Corporation
[2] The University of Electro-Communications

**Abstract.** In this paper, we succeed in analyzing practical cryptosystems that employ the Davies-Meyer Merkle-Damgård hash function $\mathsf{DM\text{-}MD}^E$ with ideal cipher $E$ by using two approaches: *indifferentiability from variants of random oracles* and *indifferentiability from a random oracle $\mathcal{RO}$ with conditions*. We show that RSA-KEM with $\mathsf{DM\text{-}MD}^E$ is secure by using the former approach and that OAEP with $\mathsf{DM\text{-}MD}^E$ is secure by using the latter approach. The public-use random oracle (pub-$\mathcal{RO}$) model is a variant of random oracle (proposed by Dodis et al. and Yoneyama et al.). We also show that cryptosystems secure under pub-$\mathcal{RO}$ model, such as FDH, Fiat-Shamir, PSS and so on, are also secure under $\mathsf{DM\text{-}MD}^E$ by using the former approach. Note that Dodis et al. failed in the paper of EUROCRYPT 2009 in analyzing the security of cryptosystems with $\mathsf{DM\text{-}MD}^E$, because they started by analyzing the underlying compression function, while our first approach starts by analyzing the hash function.

**Keywords:** Merkle-Damgård construction, Davies-Meyer mode, indifferentiability, random oracles, secure cryptosystems in the random oracle model.

## 1 Introduction

### 1.1 Background

Well-known cryptosystems such as RSA-KEM [15], OAEP [2], PSS [3] and so on have been proven secure in the random oracle $\mathcal{RO}$ model. When these cryptosystems are implemented in a real environment, $\mathcal{RO}$ is replaced by a hash function such as SHA-1, SHA-256 and so on, so their security is unknown.

The Merkle-Damgård (MD) construction is a widely used hash construction [8, 12]. The MD construction is adopted by many famous hash functions such as SHA-1, SHA-256 and so on. The MD construction digests input message $M$ by a cascade construction that uses an underlying fixed-input-length (FIL) compression function $f$. We denote a hash function with MD construction with compression function $f$ by $MD^f$

At Crypto 2005, Coron et al. [7] showed that hash functions with the $MD$ construction did not satisfy the security notion of *indifferentiability from random oracle $\mathcal{RO}$* using the *extension property*, even if it employs an ideal primitive (e.g. FIL-$\mathcal{RO}$ or ideal block cipher) as an underlying primitive. For example, the output of hash function $MD^f$ can be calculated from some output $z = MD^f(M)$ of input $M$. Namely $f(m, z) = MD^f(M||m)$ holds due to the cascade construction. On the other hand, $\mathcal{RO}$ does not have this property since it is monolithic. The indifferentiability from $\mathcal{RO}$ implies that if hash function $H$ satisfies this notion, $H$ behaves like $\mathcal{RO}$. This implies that any cryptosystem secure in the $\mathcal{RO}$ model is also secure under hash function $H$. However, if hash function $H$ does not satisfy this notion, some cryptosystem is secure in the $\mathcal{RO}$ model but insecure under $H$. Namely, *there exists some cryptosystem secure in the $\mathcal{RO}$ model but insecure under hash functions with the MD construction.* This is a serious problem, because standard cryptosystems (e.g. OAEP, RSA-KEM and so on), which are secure in the $\mathcal{RO}$ model, might be insecure under the hash functions. *We intend to prove that several important cryptosystems secure in the $\mathcal{RO}$ model are also secure under hash functions with the MD construction.*

There are several studies on this target. Dodis et al. [9] salvaged the MD construction by finding a natural property such that the security of cryptosystems can be easily analyzed; they found two properties that the strengthened MD construction preserves the property when $f$ satisfies the property. These properties are the preimage awareness (PrA) and the public-use random oracle (pub-$\mathcal{RO}$). The latter is the same as Leakey Random Oracle ($\mathcal{LRO}$) proposed by Yoneyama et al [16].

PrA is the security notion of hash functions and lies between collision resistance and $\mathcal{RO}$. PrA means that if an attacker is able to find a later useful output $y$ of the PrA function, then it must already know a corresponding preimage. Dodis et al. proved two facts: if $f$ is a FIL-PrA function, then $MD^f$ is a PrA function and if $H$ is a PrA function and $g$ is FIL-$\mathcal{RO}$, then $g(H)$ satisfies indifferentiability from $\mathcal{RO}$. As a result, $g(MD^f)$ satisfies indifferentiability from $\mathcal{RO}$. Therefore, any cryptosystem secure in the $\mathcal{RO}$ model is secure under $g(MD^f)$. The important point of this approach is, by adding just FIL-$\mathcal{RO}$ $g$, the variant function of $MD^f$ satisfies indifferentiability from $\mathcal{RO}$. Moreover, $f$ is a weaker primitive than FIL-$\mathcal{RO}$.

The second property is pub-$\mathcal{RO}$. pub-$\mathcal{RO}$ is a variant of $\mathcal{RO}$. pub-$\mathcal{RO}$ has the function of leaking all prior queries known to $\mathcal{RO}$. Dodis et al. analyzed cryptosystems with hash functions with MD construction by using pub-$\mathcal{RO}$ and found as follows: (i) if compression function $f$ is FIL-pub-$\mathcal{RO}$, then $MD^f$ satisfies indifferentiability from variable input length (VIL) pub-$\mathcal{RO}$, and (ii) several cryptosystems are secure in the pub-$\mathcal{RO}$ model. Dodis et al. claimed that FDH [1], PSS [3], PFDH [6], BLS[4], a variant of Boneh-Franklin IBE [14] and Boneh-Boyen IBE [5] are secure in the pub-$\mathcal{RO}$ model. Note these cryptosystems are secure under $MD^f$ under the condition that $f$ is FIL pub-$\mathcal{RO}$.

### 1.2 Davies-Meyer Merkle-Damgård

Davies-Meyer Merkle-Damgård, hereafter denoted by $\mathsf{DM\text{-}MD}$ construction, is the MD construction based on Davies-Meyer ($\mathsf{DM}$) mode [8, 12, 13] where underlying compression function $f(m, x)$ is implemented by $E_m(x) \oplus x$ where $E$ is a block cipher (where $m$ is a key element). Hereafter we denote $E_m(x)$ by $E(m, x)$. Standard hash function such as SHA-1, SHA-256 and so on employ the MD construction, and the underlying compression functions of SHA-1, SHA-256 and so on employ $\mathsf{DM}$ mode. Therefore, it is important to analyze the security of practical cryptosystems that employ $\mathsf{DM\text{-}MD}$ hash functions. Hereafter we deal with $E$ as an ideal block cipher and denote the $\mathsf{DM\text{-}MD}$ hash function implemented by $E$ by $\mathsf{DM\text{-}MD}^E$. However, the PrA approach and the pub-$\mathcal{RO}$ approach cannot be used to analyze cryptosystems based on $\mathsf{DM\text{-}MD}^E$. This is because, for the PrA approach, if $\mathsf{DM}$ satisfies FIL-PrA, modified $\mathsf{DM\text{-}MD}^E$ ($g(\mathsf{DM\text{-}MD}^E)$ where $g$ is FIL $\mathcal{RO}$) satisfies indifferentiability from $\mathcal{RO}$. However, $g(\mathsf{DM\text{-}MD}^E)$ is not the original Davies-Meyer Merkle-Damgård. For the pub-RO approach, since $\mathsf{DM}$ does not satisfy indifferentiability from FIL pub-$\mathcal{RO}$, the approach that uses the property where $MD^f$ preserves pub-RO does not work, because $f$ is not pub-$\mathcal{RO}$[3]. More detail is given in appendix A. Namely the approach of Dodis et al. does not work in the case of $\mathsf{DM}$ mode for analyzing cryptosystems that are secure in the pub-$\mathcal{RO}$ model.

### 1.3 Our Contribution

In this paper we succeed in analyzing the security of practical cryptosystems with $\mathsf{DM\text{-}MD}^E$ by using two approaches: *indifferentiability from variants of random oracles* $\widetilde{\mathcal{RO}}$ and *indifferentiability from* $\mathcal{RO}$ *with conditions*; it represents the first succesful rescue of $\mathsf{DM\text{-}MD}^E$ in the indifferentiability framework.

INDIFFERENTIABILITY FROM $\widetilde{\mathcal{RO}}$. For the first approach, we analyze cryptosystems with $\mathsf{DM\text{-}MD}^E$ by using indifferentiability from $\widetilde{\mathcal{RO}}$. Our approach is as follows:

1. Find a variant of random oracle $\widetilde{\mathcal{RO}}$ such that target hash function ($\mathsf{DM\text{-}MD}^E$) satisfies the indifferentiability from $\widetilde{\mathcal{RO}}$ (this hash function behaves like $\widetilde{\mathcal{RO}}$).
2. Prove that cryptosystem $\mathcal{C}$ is secure in the $\widetilde{\mathcal{RO}}$ model.

[3] Dodis et al. made a mistake in the previous version of their draft of EUROCRYPT 2009 wherein they claimed that $\mathsf{DM}$ satisfies the indifferentiability from FIL pub-$\mathcal{RO}$. In the latest version of their draft [9], it is described that we conjecture that although DM is not itself pub-PRO, applying MD to it results in a VIL pub-PRO (in the ideal cipher model).

If we can prove the above approach for cryptosystem $\mathcal{C}$, $\mathcal{C}$ under $\mathsf{DM\text{-}MD}^E$ is secure due to the indifferentiability framework. For this approach we show the following:

- For the first claim, we propose a new oracle called *extension property and inverse property simulatable random oracle* ($\mathcal{EIRO}$).
- For the second claim, we prove that RSA-KEM is secure under $\mathsf{DM\text{-}MD}^E$.

Since block ciphers have invertibility (we can calculate the inverse of $E$), there is a property derived, called the inverse property, from invertibility in $\mathsf{DM\text{-}MD}^E$. We call the property the inverse property. The inverse property is if we can get values $z = \mathsf{DM\text{-}MD}^E(M)$ and $z' = \mathsf{DM\text{-}MD}^E(M||m)$, then $E_m^{-1}(z \oplus z') = z$ holds due to invertibility of block ciphers and the cascade construction of $\mathsf{DM\text{-}MD}^E$. However, since $\mathcal{RO}$ is a monolithic random function, it inherently does not have the inverse property. We define $\mathcal{EIRO}$ so as to capture both the inverse property and the extension property.

We will prove that

- $\mathsf{DM\text{-}MD}^E$ is indifferentiable from $\mathcal{EIRO}$.

We will also prove the following:

- $\mathcal{EIRO}$ satisfies indifferentiability from pub-$\mathcal{RO}$.

This implies that any cryptosystem secure in the pub-$\mathcal{RO}$ model is also secure in the $\mathcal{EIRO}$ model. Namely, any cryptosystem secure including FDH, PSS, PFDH, BLS and so on in the pub-$\mathcal{RO}$ model is also secure under $\mathsf{DM\text{-}MD}^E$. Note that the aproach of Dodis et al. failed in proving that any cryptosystem secure in the pub-$\mathcal{RO}$ model is also secure under $\mathsf{DM\text{-}MD}^E$ (see Appendix A).

We, moreover, prove the following:

- pub-$\mathcal{RO}$ does not satisfy indifferentiability from $\mathcal{EIRO}$.

This implies that there is some cryptosystem secure in the $\mathcal{EIRO}$ model but insecure in the pub-$\mathcal{RO}$ model. We show that RSA-KEM is insecure in the pub-$\mathcal{RO}$ model (does not satisfy OW-CPA security) as an explicit example. This is evidence of the separation between pub-$\mathcal{RO}$ and $\mathcal{EIRO}$. Therefore, $\mathcal{EIRO}$ is the more powerful tool than pub-$\mathcal{RO}$ for anlyzing cryptosystems that employ $\mathsf{DM\text{-}MD}^E$.

Note that PrA and $\mathcal{EIRO}$ seems to be unrelated (since there is a relation between $\mathcal{EIRO}$ and pub-$\mathcal{RO}$ but pub-$\mathcal{RO}$ and PrA are independent). However, no cryptosystem has been shown to be secure in the $\mathcal{RO}$ model and under a PrA function.

We can conclude that RSA-KEM, FDH, PSS, Fiat-Shamir and so on are secure under $\mathsf{DM\text{-}MD}^E$, because $\mathsf{DM\text{-}MD}^E$ satisfies indifferentiability from $\mathcal{EIRO}$ and pub-$\mathcal{RO}$.

Indifferentiability from $\mathcal{RO}$ with Conditions. We propose another new approach: indifferentiability from $\mathcal{RO}$ *with conditions*. By using this approach, we can *easily* prove the security of cryptosystems that use hash functions. *Condition* refers *not* to properties of the hash functions (PrA, pub-RO and so on) but to the *condition of the hash function input*.

For example, we consider OAEP encryption with $\mathsf{DM\text{-}MD}^E$. We adopt the following procedure for OAEP.

1. Identify condition $\alpha$ that is satisfied by OAEP: $\alpha$ is the fixed input length of the hash function.
2. Identify the characteristic of condition $\alpha$: For any different but same length messages $M$ and $M'$, $M$ is not prefix of $M'$. Therefore, any input of $\mathsf{DM\text{-}MD}^E$ that satisfies condition $\alpha$ is prefix-free.
3. Identify the hash function that has the characteristic of step 2 and is indifferentiable from $\mathcal{RO}$. The hash function yielded step 2 whose input is prefix-free is prefix-free $\mathsf{DM\text{-}MD}^E$ which is indifferentiable from $\mathcal{RO}$.
4. If $\mathsf{DM\text{-}MD}^E$ with condition $\alpha$ implies the hash function identified in step 3, it implies prefix-free $\mathsf{DM\text{-}MD}^E$.
5. OAEP is secure in the $\mathcal{RO}$ model.
6. This proposed approach of indifferentiability from $\mathcal{RO}$ with conditions allows us to conclude that OAEP using $\mathsf{DM\text{-}MD}^E$ is secure.

We can easily analyze the security of OAEP by this approach, because we only identify the condition $\alpha$ and the hash function of step 3.

Adopting two approaches, we can conclude that OAEP (given indifferentiability from $\mathcal{RO}$ with condition), RSA-KEM [15], FDH [1], PSS [3], PFDH [6], BLS[4], a variant of Boneh-Franklin IBE [14] and Boneh-Boyen IBE [5] (given indifferentiability from $\widetilde{\mathcal{RO}}$) are secure under $\mathsf{DM}\text{-}\mathsf{MD}^E$.

### 1.4 Related Works

As described in Section 1.1, Dodis et al. salvaged the MD construction by either deriving the property of MD, namely PrA, or arguing the indifferentiability from weakened primitive, namely pub-$\mathcal{RO}$.

Unfortunately they failed to prove that the security will be preserved basing MD construction on $\mathsf{DM}$ mode, even though they implicitly show it might be feasible, but without any proof.

We will rescue their second approach, furthermore introduce a new primitive stronger than pub-$\mathcal{RO}$ but weak enough for MD, and propose a new approach denoted as condition $\alpha$ which is derived from the property of the protocols.

## 2 Preliminaries

### 2.1 Davies-Meyer Merkle-Damgård Construction [8, 12, 13]

We first give a short description of the Merkle-Damgård (MD) construction. Function $MD^f : \{0,1\}^* \rightarrow \{0,1\}^n$ is built by iterating compression function $f : \{0,1\}^t \times \{0,1\}^n \rightarrow \{0,1\}^n$ as follows.

- $MD^f(M)$:
  1. calculate $M' = pad(M)$ where $pad$ is a padding function such that $pad : \{0,1\}^* \rightarrow (\{0,1\}^t)^*$.
  2. calculate $c_i = f(m_i, c_{i-1})$ for $i = 1, ..., l$ where for $i = 1, ..., l$, $|m_i| = t$, $M' = m_1||...||m_l$ and $c_0$ is an initial value (s.t. $|c_0| = n$).
  3. return $c_l$

The Davies-Meyer Merkle-Damgård ($\mathsf{DM}\text{-}\mathsf{MD}$) construction is the MD construction with the underlying compression function instantiated by Davies-Meyer mode ($\mathsf{DM}$). The Davies-Meyer model is $\mathsf{DM}(m,x) = x \oplus E(m,x)$ where $m$ is a key element of the block cipher. Hereafter $E$ is an ideal block cipher and we denote the hash function $MD^{\mathsf{DM}}$ by $\mathsf{DM}\text{-}\mathsf{MD}^E$. In this paper we ignore the above padding function but this implies no loss of generality, so hereafter we discuss only $\mathsf{DM}\text{-}\mathsf{MD}^E : (\{0,1\}^t)^* \rightarrow \{0,1\}^n$.

We denote a forward query $(m,x)$ to $E$ by $(+,m,x)$ and an inverse query $(m,y)$ to $E$ by $(-,m,y)$. We call a query to hash functions "hash query".

### 2.2 Indifferentiability Framework for Hash Functions [11]

The indifferentiability framework generalizes the fundamental concept of the indistinguishability of two crypto systems $\mathcal{C}(\mathcal{U})$ and $\mathcal{C}(\mathcal{V})$ where $\mathcal{C}(\mathcal{U})$ is the cryptosystem $\mathcal{C}$ that invokes the underlying primitive $\mathcal{U}$ and $\mathcal{C}(\mathcal{V})$ is the cryptosystem $\mathcal{C}$ that invokes the underlying primitive $\mathcal{V}$. $\mathcal{U}$ and $\mathcal{V}$ have two interfaces: public and private interfaces. Adversaries can only access the public interfaces and honest parties (e.g. the cryptosystem $\mathcal{C}$) can access only the private interface.

We denote the private interface of the system $\mathcal{W}$ by $\mathcal{W}^1$ and the public interface of the system $\mathcal{W}$ by $\mathcal{W}^2$. The channel between honest party and $\mathcal{V}^1/\mathcal{U}^1$ is called private channel and the channel between adversary and $\mathcal{V}^2/\mathcal{U}^2$ is called public channel. The definition of indifferentiability is as follows.

**Definition 1.** *$\mathcal{V}$ is indifferentiable from $\mathcal{U}$, denote $\mathcal{V} \sqsubset \mathcal{U}$, if for any distinguisher $D$ with binary output (0 or 1) there is a simulator $S$ such that the advantage $|Pr[D^{\mathcal{V}^1,\mathcal{V}^2} \Rightarrow 1] - Pr[D^{\mathcal{U}^1,S(\mathcal{U}^2)} \Rightarrow 1]|$ is negligible in the security parameter $k$.*

This definition will allow us to use construction $\mathcal{V}$ instead of $\mathcal{U}$ in *any* cryptosystem and retains the same level of provable security due to the indifferentiability theory of Maurer et al. [11]. We denote the same level of probable security by $\mathcal{C}(\mathcal{V}) \succ \mathcal{C}(\mathcal{U})$. Namely we denote $\mathcal{C}(\mathcal{V}) \succ \mathcal{C}(\mathcal{U})$ in the case that if $\mathcal{C}(\mathcal{U})$ is secure, then $\mathcal{C}(\mathcal{U})$ is secure. More strictly, $\mathcal{V} \sqsubset \mathcal{U} \Leftrightarrow \mathcal{C}(\mathcal{V}) \succ \mathcal{C}(\mathcal{U})$ holds. In the proof of $\mathcal{V} \sqsubset \mathcal{U} \Leftrightarrow \mathcal{C}(\mathcal{V}) \succ \mathcal{C}(\mathcal{U})$ [11], all queries of $\mathcal{D}$ to $\mathcal{V}^1/\mathcal{U}^1$ are through the private channel.

In this paper, since we deal with cryptosystems which access hash functions, $\mathsf{DM\text{-}MD}^E$ has a private interface. Since $\mathsf{DM\text{-}MD}^E$ invokes $E$ and $E$ is a public oracle, $E$ has both private and public interfaces.

### 2.3 Distinguishing Attack using the Extension Property [7]

Coron et al. showed that $\mathsf{DM\text{-}MD}^E \not\sqsubset \mathcal{RO}$ (random oracle) using the extension property. The extension property is the property of the cascade construction where we can calculate a new hash value from some hash value. $z' = \mathsf{DM\text{-}MD}^E(M||m)$ can be calculated from only $z$ and $m$ by $z' = E(m, z) \oplus z$ where $z = \mathsf{DM\text{-}MD}^E(M)$. Namely $z'$ can be calculated without using $M$. The distinguishing attack using the extension property is as follows. Let $\mathcal{O}_1$ be $\mathsf{DM\text{-}MD}^E$ or $\mathcal{RO}$ and let $\mathcal{O}_2$ be $E$ or $S$. First, a distinguisher poses $M$ to $\mathcal{O}_1$ and gets $z$ from $\mathcal{O}_1$. Second, he poses a forward query $(+, m, z)$ to $\mathcal{O}_2$ and gets $c$ from $\mathcal{O}_2$. Finally, he poses $M||m$ to $\mathcal{O}_1$ and gets $z'$ from $\mathcal{O}_1$.

If $\mathcal{O}_1 = \mathsf{DM\text{-}MD}^E$ and $\mathcal{O}_2 = E$, then $z \oplus z' = c$, however, if $\mathcal{O}_1 = \mathcal{RO}$ and $\mathcal{O}_2 = \mathcal{S}$, then $z \oplus z' \neq c$. This is because no simulator can obtain the output value of $\mathcal{RO}(M||m)$ from just $(m, z)$ and the output value of $\mathcal{RO}(M||m)$ is independently and randomly defined from $c$. Therefore, $\mathsf{DM\text{-}MD}^E \not\sqsubset \mathcal{RO}$ holds.

### 2.4 Distinguishing Attack using the Inverse Property

There is another distinguishing attack since block ciphers are invertible. There is a property called inverse property that is one of properties of $\mathsf{DM\text{-}MD}^E$ that uses invertibility of block ciphers. Therefore $\mathsf{DM\text{-}MD}^E \not\sqsubset \mathcal{RO}$ also holds due to the inverse property. We show the distinguishing attack that uses this property as follows.

In the ideal cipher scenario, on an inverse query $(-, m, y)$ where $y = z \oplus z'$ such that $z = \mathsf{DM\text{-}MD}^E(M)$, $z' = \mathsf{DM\text{-}MD}^E(M||m)$, $E$ returns $z = \mathsf{DM\text{-}MD}^E(M)$. However, in the $\mathcal{RO}$ scenario, no simulator $\mathcal{S}$ can simulate the inverse attack. On an inverse query $(-, m, y)$ where $y = z \oplus z'$ where $z = \mathcal{RO}(M)$ and $z' = \mathcal{RO}(M||m)$, no $S$ can return $z$. since no $\mathcal{S}$ can know $z$ or $z'$ from $(m, y)$ by using just $\mathcal{RO}$.

### 2.5 Public-use Random Oracle [9, 16]

Dodis et al. proposed public-use random oracle (pub-$\mathcal{RO}$) to analyze cryptosystems that use hash functions with the MD construction. We model pub-$\mathcal{RO}$ by consisting of random oracle $\mathcal{RO}$ and leak oracle $\mathcal{LO}$. These definition is as follows: $\mathcal{RO}$ has initially empty list $\mathcal{L}_{\mathcal{RO}}$. For a query $M$, if $(M, z) \notin \mathcal{L}_{\mathcal{RO}}$, it chooses a $n$-bit random value $z$, $\mathcal{L}_{\mathcal{RO}} \leftarrow (M, z)$, and returns $z$. Otherwise ($(M, z) \in \mathcal{L}_{\mathcal{RO}}$) it returns $z$. On a query to $\mathcal{LO}$, $\mathcal{LO}$ returns $\mathcal{L}_{\mathcal{RO}}$.

pub-$\mathcal{RO}$ has interesting properties as following points: (1) We can easily argue security of several cryptosystems. (2) If $f$ is FIL pub-$\mathcal{RO}$ then $MD^f$ is the pub-$\mathcal{RO}$ function. In the first point, for any scheme and security experiment for which all messages queried to $\mathcal{RO}$ can be inferred from an adversary's queries during the experiment, one can prove straightforwardly the scheme's security in the pub-$\mathcal{RO}$ model using an existing proof in the full $\mathcal{RO}$ model as a black box. For example, these conditions are met for unforgeability under chosen-message attacks of signature schemes that use the $\mathcal{RO}$ on messages. So we can easily verity that FDH, Fiat-Shamir, PSS and some encryption schemes are secure in the pub-$\mathcal{RO}$ model. In the second point, if we can get the FIL pub-$\mathcal{RO}$ function $f$, then we can get the hash function $MD^f$ that is indifferentiable from pub-$\mathcal{RO}$. FIL $\mathcal{RO}$ is crealy indifferentiable from pub-$\mathcal{RO}$. Therefore, $MD^{\text{FIL}-\mathcal{RO}}$ is indifferentiable from pub-$\mathcal{RO}$.

### 2.6 $\mathsf{DM} \not\sqsubset$ pub-$\mathcal{RO}$

$\mathsf{DM} \not\sqsubset$ pub-$\mathcal{RO}$ holds due to invertibility of block ciphers. Therefore, the approach of Dodis et al, where MD construction is preserving the pub-$\mathcal{RO}$ property, fails in applying to $\mathsf{DM\text{-}MD}^E$. Please see Appendix A for more detail.

## 3 Indifferentiability from $\widetilde{\mathcal{RO}}$ Approach

In this section, we will introduce a new primitive called *extension property and inverse property simulatable random oracle* $\mathcal{EIRO}$ and show $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{EIRO}$. We will also show the relation among $\mathcal{EIRO}$ and pub-$\mathcal{RO}$.

### 3.1 Extension Property and Inverse Property Simulatable Random Oracle ($\mathcal{EIRO}$)

**Motibation of $\mathcal{EIRO}$.** Since no $S$ can simulate the extension property and the inverse property, $\mathsf{DM\text{-}MD} \not\sqsubset \mathcal{RO}$ holds. We consider variants of $\mathcal{RO}$ in order for $\mathcal{S}$ to be able to simulate the extension property and the inverse property. We propose new primitive $\mathcal{EIRO}$ which consists of $\mathcal{RO}$, extension property simulatable oracle $\mathcal{EO}$, and inverse property simulatable oracle $\mathcal{IO}$ such that it captures the extension property and the inverse property. The motivation of new primitives is as follows:

- $\mathcal{EO}$: Since, for a forward query $(+, m, z)$, no $\mathcal{S}$ can know $x \oplus z'$ in the $\mathcal{RO}$ scenario such that $z = \mathcal{RO}(M)$ and $z' = \mathcal{RO}(M||m)$, it is convenient for $\mathcal{S}$ to know the valid return value $z \oplus z'$. So we define $\mathcal{EO}$ that returns $z \oplus z'$ for a query $(m, x)$.
- $\mathcal{IO}$: Since, for a inverse query $(-, m, z \oplus z')$ where $z = \mathcal{RO}(M)$ and $z' = \mathcal{RO}(M||m)$, no $\mathcal{S}$ can know $z$ in the $\mathcal{RO}$ scenario, it is convenient for $\mathcal{S}$ to know the the valid return value $z$. we define $\mathcal{IO}$ such that it returns $z$ for a query $(m, y)$.

These oracles help $\mathcal{S}$ to simulate the extension property and the inverse property. We give the concrete definitions of these oracles as follows.

**Definition of $\mathcal{EIRO}$.** $\mathcal{EIRO}$ is constructed from $\mathcal{RO}$, $\mathcal{EO}$ and $\mathcal{IO}$.
$\mathcal{EO}$ has initially the empty list $\mathcal{L}_{\mathcal{EO}}$. It can look into $\mathcal{L}_{\mathcal{RO}}$. On a query $(m, z)$ to $\mathcal{EO}$,

1. If $(m, z, z') \in \mathcal{L}_{\mathcal{EO}}$, it returns $z'$.
2. Else if there exists only one pair $(M, z) \in \mathcal{L}_{\mathcal{RO}}$, it makes the query $M||m$ to $\mathcal{RO}$, receives $z'$ from $\mathcal{RO}$, $\mathcal{L}_{\mathcal{EO}} \leftarrow (m, z, z')$ and it returns $z'$.
3. Else it chooses $z' \in \{0,1\}^n$ at random, $\mathcal{L}_{\mathcal{EO}} \leftarrow (m, z, z')$ and it returns $z'$.

$\mathcal{IO}$ has initially the list $\mathcal{L}_{\mathcal{IO}} = \{(\phi, \perp, 0)\}$ and can look into $\mathcal{L}_{\mathcal{RO}}$. On a query $(m, y)$,

1. If $\exists (m, x, y) \in \mathcal{L}_{\mathcal{IO}}$, it returns $x$.
2. If $\exists (M, x), (M', x') \in \mathcal{L}_{\mathcal{RO}}$ such that $M||m = M'$ and $y = x \oplus x'$, $\mathcal{L}_{\mathcal{IO}} \leftarrow (m, x, y)$ and it returns $x$.
3. Otherwise it chooses $x \in \{0,1\}^n$ at random, $\mathcal{L}_{\mathcal{IO}} \leftarrow (m, x, y)$ and it returns $x$.

In Theorem 2, we prove that $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{EIRO}$. Before showing Theorem 2, we show the relation among pub-$\mathcal{RO}$ and $\mathcal{EIRO}$.

### 3.2 Relation among pub-$\mathcal{RO}$ and $\mathcal{EIRO}$

pub-$\mathcal{RO}$ leaks more information of $\mathcal{L}_{\mathcal{RO}}$ than $\mathcal{EIRO}$. Therefore it seems reasonable to suppose that any cryptosystem secure in the pub-$\mathcal{RO}$ model is also secure in the $\mathcal{EIRO}$ model. We prove the validity of these supposition by using the indifferentiability framework.

We will clarify the relationship between pub-$\mathcal{RO}$ and $\mathcal{EIRO}$.

**Theorem 1.** *$\mathcal{EIRO} \sqsubset$ pub-$\mathcal{RO}$ and pub-$\mathcal{RO} \not\sqsubset \mathcal{EIRO}$.*

*Proof.* We construct simulator $\mathcal{S}$ which simulates $\mathcal{EO}$ and $\mathcal{IO}$ by using pub-$\mathcal{RO}$ as follows. $S$ has initially lists $\mathcal{L}_{S_1} = \phi$ and $\mathcal{L}_{S_2} = \{(\phi, \perp, 0)\}$. On query $\mathcal{EO}$ $(m, z)$, if there is $(m, z, z') \in \mathcal{L}_{S_1}$, it returns $z'$. It makes a query to $\mathcal{LO}$ and receives $\mathcal{L}_{\mathcal{RO}}$. If there is only one pair $(M, z) \in \mathcal{L}_{\mathcal{RO}}$, it makes the query $M||m$ to $\mathcal{RO}$, receives the response $z'$ from $\mathcal{RO}$, $\mathcal{L}_{\mathcal{EO}} \leftarrow (m, z, z')$ and returns $z'$. Otherwise it chooses $x \in \{0,1\}^n$ at random, $\mathcal{L}_{\mathcal{EO}} \leftarrow (m, z, x)$ and returns $x$. On a $\mathcal{IO}$ query $(m, y)$, if $(m, x, y) \in \mathcal{L}_{S_2}$, it returns $x$. It makes $\mathcal{LO}$ query, receives $\mathcal{L}_{\mathcal{RO}}$. If $\exists(M, x), (M', x') \in \mathcal{L}_{\mathcal{RO}}$ such that $M||m = M'$ and $x \oplus x' = y$, $\mathcal{L}_{S_2} \leftarrow (m, x, y)$ and it returns $x$. Otherwise it chooses $x \in \{0,1\}^n$ at random, $\mathcal{L}_{\mathcal{EO}} \leftarrow (m, x, y)$ and returns $x$.

It is easy to see that $|Pr[D^{\mathcal{RO},\mathcal{EO},\mathcal{IO}}] - Pr[D^{\mathcal{RO},S(\mathcal{LO})}]|$ is negligible, since it is clearly that $\mathcal{S}$ is equal to $\mathcal{EO}$ and $\mathcal{IO}$.

pub-$\mathcal{RO} \not\sqsubset \mathcal{EIRO}$ is trivial, since $\mathcal{EIRO}$ does not explicitly leak more information in $\mathcal{L}_{\mathcal{RO}}$ than pub-$\mathcal{RO}$, so no $\mathcal{S}$ can know all values in $\mathcal{L}_{\mathcal{RO}}$ by using just $\mathcal{EIRO}$. □

Since $\mathcal{EIRO} \sqsubset$ pub-$\mathcal{RO}$ holds, any cryptosystem secure in the pub-$\mathcal{RO}$ model is also secure in the $\mathcal{EIRO}$ model by the indifferentiability framework. Therefore FDH, PSS, Fiat-Shamir and so on (secure in the pub-$\mathcal{RO}$ model) are secure in the $\mathcal{EIRO}$ model. Since pub-$\mathcal{RO} \not\sqsubset \mathcal{EIRO}$, there is some cryptosystem which is secure in the $\mathcal{EIRO}$ model but insecure in the pub-$\mathcal{RO}$ model. We will prove that RSA-KEM is insecure in the pub-$\mathcal{RO}$ model but secure in the $\mathcal{EIRO}$ model. Therefore RSA-KEM is evidence of the separation between $\mathcal{LRO}$ and $\mathcal{EIRO}$.

### 3.3 Proof of indifferentiability from $\mathcal{EIRO}$ for $\mathsf{DM\text{-}MD}^E$

In this subsection, we prove $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{EIRO}$.

**Theorem 2.** *$\mathsf{DM\text{-}MD}^E$ is $(t_D, t_S, q, \epsilon)$-indifferentiable from $\mathcal{EIRO}$ for any $t_D$, with $t_S = O(lq)$ and $\epsilon = O(q^2l^2)/2^n$, where $l$ is maximum message block length queried by $D$ and $q$ is maximum number of query of $D$.*

We demonstrate an intuition of this proof by using the previous result (the proof of the indifferentiability from $\mathcal{RO}$ for pre-fix free MD construction). The complete proof from scratch will be described in Appendix B.

Proof of indifferentiability from $\boldsymbol{\mathcal{RO}}$ for Prefix-Free MD. Coron et al. and Chang et al. proved that prefix-free $\mathsf{DM\text{-}MD}^E$ is indifferentiable from $\mathcal{RO}$. Hash function prefix-free $\mathsf{DM\text{-}MD}^E$ is $\mathsf{DM\text{-}MD}^E$ with the prefix-free padding function $PF$. The definition of $PF$ is that for any two different two message $M$ and $M'$, $PF(M)$ is not prefix of $PF(M')$. They consider trivial queries that might be helpful for $\mathcal{D}$ to distinguish a hash function and $\mathcal{RO}$. Let $\mathcal{D}_{TQ}$ be any distinguisher based on trivial queries and $\mathcal{D}_{\neg TQ}$ be any distinguisher based on non-trivial query. Non-trivial queries means queries excluding trivial queries. Strategy of this proof is:

1. Consider trivial queries (TQ) which might help $\mathcal{D}$ to distinguish $\mathcal{RO}$ and the hash function.
2. Prove that the advantage probability of distinguisher $\mathcal{D}_{TQ}$ is negligible.
3. Prove that the advantage probability of distinguisher $\mathcal{D}_{\neg TQ}$ is negligible.

In prefix-free $\mathsf{DM\text{-}MD}^E$ case, there are two types trivial queries: type 1 and type 2 that use the extension property. Let $\mathcal{D}_{type1}$ be any distinguisher based on trivial query type 1 and $\mathcal{D}_{type2}$ be any distinguisher based on trivial query type 2. Therefore, in prefix-free $\mathsf{DM\text{-}MD}^E$ case, $\mathcal{D}_{TQ}$ is combined with $\mathcal{D}_{type1}$ and $\mathcal{D}_{type2}$ (denote $\mathcal{D}_{TQ} = \mathcal{D}_{type1} + \mathcal{D}_{type2}$) and any distinguisher $\mathcal{D}$ is combined with $\mathcal{D}_{type1}$, $\mathcal{D}_{type2}$ and $\mathcal{D}_{\neg TQ}$ ($\mathcal{D} = \mathcal{D}_{\neg TQ} + \mathcal{D}_{type1} + \mathcal{D}_{type2}$). They showed that prefix-free $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{RO}$ holds by using above strategy.

The proof of the indifferentiability from $\mathcal{EIRO}$ based on the previous result. We give the intuition of our proof based on the previous proof of prefix-free $\mathsf{DM\text{-}MD}^E$. Full proof is given in Appendix B. The difference between *plain* $\mathsf{DM\text{-}MD}^E$ and prefix-free $\mathsf{DM\text{-}MD}^E$ is that plain $\mathsf{DM\text{-}MD}^E$ does not have a prefix-free padding function $PF$. Therefore we need to consider trivial queries type 3-6 in addition to type 1,2. In prefix-free $\mathsf{DM\text{-}MD}^E$ case, type 3-6 need not to be considered due to prefix-free padding. Type 3, 4 are based on the extension property, and type 5, 6 are based on the inverse property. Let $\mathcal{D}_{type3}$, $\mathcal{D}_{type4}$, $\mathcal{D}_{type5}$, and $\mathcal{D}_{type6}$ be distinguishers based on trivial query type 3, type 4, type 5 and type 6 respectively. We prove Theorem 2 by basing on the previous result.

- The advantage of $\mathcal{D}_{\neg TQ}$: We consider the advantage probability of $\mathcal{D}_{\neg TQ} = \mathcal{D} - \mathcal{D}_{type1} - \mathcal{D}_{type2} - \mathcal{D}_{type3} - \mathcal{D}_{type4} - \mathcal{D}_{type5} - \mathcal{D}_{type6}$. Previous work showed that the advantage of $\mathcal{D}_{\neg TQ} = \mathcal{D} - \mathcal{D}_{type1} - \mathcal{D}_{type2}$ is negligible. From the previous result, we can automatically get that the advantage probability of $\mathcal{D}_{\neg TQ} = \mathcal{D} - \mathcal{D}_{type1} - \mathcal{D}_{type2} - \mathcal{D}_{type3} - \mathcal{D}_{type4} - \mathcal{D}_{type5} - \mathcal{D}_{type6}$ is negligible, since strategy of $\mathcal{D}_{\neg TQ}$ in our case is more strict restriction than previous case.
- The advantage of $\mathcal{D}_{TQ}$: In previous case, there is prefix-free padding. In our case, prefix-free padding is removed but $\mathcal{RO}$ is replaced by $\mathcal{EIRO}$. Thanks to $\mathcal{EIRO}$, the advantage probability of $\mathcal{D}_{TQ}$ is negligible. We discuss more detail as follows.

**The advantage of $\mathcal{D}_{TQ}$:** $\mathcal{D}_{TQ}$ is any distinguisher that uses trivial queries of type 1-6. Details of trivial queries type 1-6 are as follows:

- **Type 1:** $\mathcal{D}$ makes the ordered sequences of forward queries $(+, m_1, x_1), ..., (+, m_i, x_i)$ such that (for $j = 1, ...i-1$) $x_{j+1}$ is equal to $y_j \oplus x_j$ ($y_j$ is the response of the forward query $(+, m_j, x_j)$) and $x_1 = IV$. Then we say that the fresh hash query $m_1||...||m_i$ is a trivial query if the trivial query is made when above queries were already made.
- **Type 2:** $\mathcal{D}$ makes the hash query $m_1||...||m_i$ and the ordered sequences of fresh forward queries $(+, m_1, x_1), ..., (+, m_{i-1}, x_{i-1})$ such that (for $j = 1, ..., i-2$) $x_{j+1}$ is equal to $y_j \oplus x_j$ ($y_j$ is the response of the forward query $(+, m_j, x_j)$) and $x_1 = IV$. Then we say that the fresh forward query $(+, m_i, x_i)$ such that $x_i = x_{i-1} \oplus y_{i-1}$ is a trivial query if the trivial query is made when above queries were already made.
- **Type 3:** $\mathcal{D}$ makes the ordered sequences of a fresh hash query $M$ and fresh forward queries $(+, m_1, x_1)$, ..., $(+, m_i, x_i)$ such that $z = x_1$ ($z$ is the response of $M$) and (for $j = 1, ..., i-1$) $x_{j+1}$ is equal to $x_j \oplus y_j$ ($y_j$ is the response of the query $(x_j, m_j)$). Then we say that the fresh hash query $M||m_1||...||m_i$ is a trivial query if the trivial query is made when above queries were already made.
- **Type 4:** $\mathcal{D}$ makes the fresh query $M||m_1||...||m_i$ and the ordered sequences of the fresh hash query $M$ and fresh queries $(+, m_1, x_1)$, ..., $(+, m_{i-1}, x_{i-1})$ such that $z = x_1$ ($z$ is the response of $M$) and (for $j = 1, ..., i-2$) $x_{j+1}$ is equal to $y_j \oplus x_j$ ($y_j$ is the response of the forward query $(+, m_j, x_j)$). Then we say that the forward query $(+, m_i, x_i)$ such that $x_i = x_{i-1} \oplus y_{i-1}$ is a trivial query if the trivial query is made when above queries were already made.
- **Type 5:** $\mathcal{D}$ makes the hash queries $M$ and $M||m$ where the response of $M$ is $z$ and one of $M||m$ is $z'$. Then we say that the inverse query $(-, m, z \oplus z')$ is a trivial query if the trivial query is made when above queries were already made.
- **Type 6:** $\mathcal{D}$ makes the ordered sequences of the fresh hash queries $M$ and the fresh forward queries $(+, m_1, x_1), ..., (+, m_i, x_i)$ and $M||m_1||...||m_i||m$. We denote the response of $M$ by $z$ and the response of $M||m_1||...||m_i||m$ by $z'$. Then we say that the inverse query $(-, m, z \oplus z')$ is a trivial query if the trivial query is made when above queries were already made.

In this case, since $\mathcal{S}$ can use $\mathcal{EIRO}$, $\mathcal{S}$ can simulate the extension property by using $\mathcal{EO}$ and the inverse property by using $\mathcal{IO}$. Therefore, we can prove that the advantage probability of $\mathcal{D}_{TQ}$ is negligible. Pick type 3 and 5, as an example.

- **Type 3:** In $\mathcal{EIRO}$ scenario, on the first query $(+, m_1, x_1)$ to $\mathcal{S}$, $\mathcal{S}$ send $(m_1, x_1)$ to $\mathcal{EO}$ when a hash query $M$ was already made where $z = x_1$. Since $(M, z) \in \mathcal{L}_{\mathcal{RO}}$, $\mathcal{EO}$ returns $z_1 = \mathcal{RO}(M||m_1)$. $\mathcal{S}$ returns

$y_1 = x_1 \oplus z_1$. After this procedure, there is $(M||m_1, z_1)$ in $\mathcal{L}_{\mathcal{RO}}$. Then on second query $(+, m_2, x_2)$ where $x_2 = z_1$, $\mathcal{S}$ send $(m_2, x_2)$ to $\mathcal{EO}$. Since $(M||m_1, z_1) \in \mathcal{L}_{\mathcal{RO}}$, $\mathcal{EO}$ returns $z_2 = \mathcal{RO}(M||m_1||m_2)$. $\mathcal{S}$ returns $y_2 = x_2 \oplus z_2$. After this procedure, there is $(M||m_1||m_2, z_2)$ in $\mathcal{L}_{\mathcal{RO}}$. Repeating this procedure, we can get the result where the response $y_i$ of the query $(+, m_i, x_i)$ is equal to $x_i \oplus z_i$ where $z_i = \mathcal{RO}(M||m_1||...||m_i)$. Therefore $x_i \oplus y_i = \mathcal{RO}(M||m_1||...||m_i)$ explicitly holds.
In $\mathsf{DM\text{-}MD}^E$ scenario, $x_i \oplus E(m_i, x_i) = \mathsf{DM\text{-}MD}^E(M||m_1||...||m_i)$ holds due to the cascade construction of $\mathsf{DM\text{-}MD}^E$.

- **Type 5:** In $\mathcal{EIRO}$ scenario, on query $(-, m, z \oplus z')$ to $\mathcal{S}$ where $z = \mathcal{RO}(M)$ and $z' = \mathcal{RO}(M||m)$, $\mathcal{S}$ makes a query $(m, z \oplus z')$ to $\mathcal{IO}$. Since there are pairs $(M, z)$ and $(M||m, z')$ in $\mathcal{L}_{\mathcal{RO}}$, $\mathcal{IO}$ returns $z$. Then $\mathcal{S}$ returns $z$.
In $\mathsf{DM\text{-}MD}^E$ scenario, $E(m_i, y_i)^{-1} = \mathsf{DM\text{-}MD}^E(M)$ explicitly holds.

Therefore queries of type 3 and type 5 are not helpful for $\mathcal{D}$. Similarly, queries type 1, type 2, type 4 and type 6 are not helpful for $\mathcal{D}$. Therefore the advantage probability of $\mathcal{D}_{TQ}$ is negligible.

From Theorem 1 and Theorem 2, the following corollary is obtained.

**Corollary 1.** $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{EIRO} \sqsubset$ *pub-*$\mathcal{RO}$

### 3.4 Security Analysis of RSA-KEM in $\mathcal{EIRO}$ Model

RSA-based key encapsulation mechanism (RSA-KEM) scheme [15] is secure KEM scheme in the $\mathcal{RO}$ model. In this subsection, we consider the security of RSA-KEM in the $\mathcal{TRO}$ and $\mathcal{ERO}$ models. The security notion of KEM schemes and the description of RSA-KEM will be described in Appendix C.

In [15], security of RSA-KEM in the $\mathcal{RO}$ model is proved as follows;

**Lemma 1 (Security of RSA-KEM in the $\mathcal{RO}$ model [15]).** *If RSA problem is hard, then RSA-KEM satisfies IND-CCA for KEM where $H$ is modeled as the $\mathcal{RO}$.*

Insecurity of RSA-KEM in pub-$\mathcal{RO}$ Model Though RSA-KEM is secure in the $\mathcal{RO}$ model, it is insecure in the pub-$\mathcal{RO}$ model. More specifically, we can show RSA-KEM does not even satisfy OW-CPA for KEM in the pub-$\mathcal{RO}$ model.

**Theorem 3 (Insecurity of RSA-KEM in the pub-$\mathcal{RO}$ model).** *Even if RSA problem is hard, RSA-KEM does not satisfy OW-CPA for KEM where $H$ is modeled as the pub-$\mathcal{RO}$.*

*Proof.* We construct an adversary $\mathcal{A}$ which successfully plays OW-CPA game by using the pub-$\mathcal{RO}$ $H$. The construction of $\mathcal{A}$ is as follows;

**Input :** $(n, e)$ as the public key

**Output :** $K'$

**Step 1 :** Return *state* and receive $c^*$ as the challenge cipertext. Pose the leak query to $\mathcal{LO}$ of $H$, and obtain the hash list $\{(r, K)\}$.

**Step 2 :** For all $r$ in $\{(r, K)\}$, check whether $r^e \overset{?}{\equiv} c^* \pmod{n}$. If there is $r^*$ satisfying the relation, output $K'$ which is the tally of $(r^*, K')$.

We estimate the success probability of $\mathcal{A}$. When the challenge ciphertext $c^*$ is generated, $r^*$ such that $K^* = H(r^*)$ is certainly posed to $H$ because $c^*$ is generated obeying the protocol description. Thus, $\mathcal{L}_H$ contains $(r^*, K^*)$. Therefore, $\mathcal{A}$ can successfully plays the OW-CPA game.

□

Security of RSA-KEM in $\mathcal{EIRO}$ Model We can also prove the security of RSA-KEM in the $\mathcal{EIRO}$ model as well as in the $\mathcal{RO}$ model.

**Theorem 4 (Security of RSA-KEM in the $\mathcal{EIRO}$ model).** *If RSA problem is $(t', \epsilon')$-hard, then RSA-KEM satisfies $(t, \epsilon)$-IND-CCA for KEM as follows:*

$$t' = t + (q_{RH} + q_{EH}) \cdot expo,$$
$$\epsilon' \geq \epsilon - \frac{q_D}{n} - \frac{q_{IH}}{|\mathbb{Z}_n|},$$

*where $H$ is modeled as the $\mathcal{EIRO}$, $q_{RH}$ is the number of hash query to the $\mathcal{RO}$ of $H$, $q_{EH}$ is the number of extension attack queries to the $\mathcal{EO}$ of $H$, $q_{IH}$ is the number of inverse attack queries to the $\mathcal{IO}$ of $H$, $q_D$ is the number of queries to the decryption oracle $\mathcal{DO}$, $|\mathbb{Z}_n|$ is the number of elements of $\mathbb{Z}_n$ and expo is the computational cost of exponentiation modulo $n$.*

*Proof (Sketch).*

Firstly, we show that the transformation of the experiment of IND-CCA for RSA-KEM from Exp0 to Exp5 in the full proof. By the step of the transformation from Exp2 to Exp3, we can show that the extension attack query $(x, y)$ of the hash value of the randomness $r^*$ or $r^*||x$ corresponding to the challenge ciphertext to $\mathcal{EO}$ of $H$ only gives no advantage to the adversary as Lemma 4 in the full proof. Information the adversary can obtain by the query is not useful without information of $r^*$ itself and the adversary can succeeds if the randomness is leaked. Also, by the step of the transformation from Exp3 to Exp4, we can show that the inverse attack query $(x, a)$ related to the hash value of the randomness $r^*$ or $r^*||x$ corresponding to the challenge ciphertext to $\mathcal{IO}$ of $H$ only gives negligible advantage to the adversary.

Next, we construct a reduction from RSA assumption to the transformed experiment Exp5 of IND-CCA for RSA-KEM. For the reduction part, we need to describe simulations of $\mathcal{EO}$ and $\mathcal{IO}$. However, we construct the perfect simulation of $\mathcal{EO}$ and $\mathcal{IO}$. Thus, we can show that RSA-KEM is secure by the similar proof as that in [15]. □

## 4 Indifferentiability from $\mathcal{RO}$ with Conditions

In this section, we propose another new approach, *indifferentiability from $\mathcal{RO}$* with conditions, to analyze cryptosystems that use $\mathsf{DM\text{-}MD}^E$. As an example, we demonstrate that OAEP encryption can be easily proven to be secure under $\mathsf{DM\text{-}MD}^E$ by adopting this approach.

### 4.1 Security of OAEP Encryption in $\mathcal{RO}$ model

The Optimal Asymmetric Encryption Padding (OAEP) encryption scheme [2] is a secure padding scheme for asymmetric encryption in the $\mathcal{RO}$ model. The security notion of OAEP schemes and the description of OAEP will be described in appendix D.

In [10], the security of OAEP encryption scheme in the $\mathcal{RO}$ model is proved as follows;

**Lemma 2 (Security of OAEP encryption scheme in the $\mathcal{RO}$ model [10]).** *If the trapdoor permutation $f$ is partial-domain one-way, then the OAEP encryption scheme satisfies IND-CCA where $H$ and $G$ are modeled as $\mathcal{RO}$.*

### 4.2 Security of OAEP Encryption under $\mathsf{DM\text{-}MD}^E$

This subsection details the new approach wherein OAEP that uses $\mathsf{DM\text{-}MD}^E$ is easily proven. We note conditions in the private channel between OAEP and underlying hash functions. We prove the security of OAEP encryption that uses $\mathsf{DM\text{-}MD}^E$ by the folllowing approach.

1. Identify condition $\alpha$ that is satisfied by OAEP: $\alpha$ is the fixed input length of the hash function.
2. Identify the characteristic of condition $\alpha$: For any different but same length messages $M$ and $M'$, $M$ is not prefix of $M'$. Therefore, any input of $\mathsf{DM\text{-}MD}^E$ that satisfies condition $\alpha$ is prefix-free.
3. Identify the hash function that has the characteristic of step 2 and is indifferentiable from $\mathcal{RO}$. The modified $\mathsf{DM\text{-}MD}^E$ yielded step 2 whose input is prefix-free is prefix-free $\mathsf{DM\text{-}MD}^E$ which is indifferentiable from $\mathcal{RO}$.
4. If $\mathsf{DM\text{-}MD}^E$ with condition $\alpha$ implies the hash function identified in step 3, it is prefix-free $\mathsf{DM\text{-}MD}^E$. Since prefix-free $\mathsf{DM\text{-}MD}^E$ is indifferentiable from $\mathcal{RO}$, $\mathsf{DM\text{-}MD}^E$ with condition $\alpha$ is indifferentiable from $\mathcal{RO}$.
5. OAEP is secure in the $\mathcal{RO}$ model (see Lemma 2).
6. Since OAEP has condition $\alpha$ and $\mathsf{DM\text{-}MD}^E$ with condition $\alpha$ is indifferentiable from $\mathcal{RO}$, this proposed approach of indifferentiability from $\mathcal{RO}$ with conditions allows us to conclude that OAEP using $\mathsf{DM\text{-}MD}^E$ is secure (we will prove Theorem 5 in Subsection 4.3).

We can easily analyze the security of OAEP by this approach, because we only need to identify condition $\alpha$ and the hash function of step 3.

The above procedure neither prove the consistency of prefix-free $\mathsf{DM\text{-}MD}^E$ in identified step 3 and $\mathsf{DM\text{-}MD}^E$ with $\alpha$, nor the consistency of $\mathsf{DM\text{-}MD}^E$ with $\alpha$ and OAEP. Accordingly we prove these consistencies bellow.

For the first consistency, we show that the indifferentiability proof of prefix-free $\mathsf{DM\text{-}MD}^E$ can be used as a black box in the indifferentiability proof of $\mathsf{DM\text{-}MD}^E$ with the condition. Since $\mathsf{DM\text{-}MD}^E$ with the condition is prefix-free $\mathsf{DM\text{-}MD}^E$, information from $\mathsf{DM\text{-}MD}^E$ with the condition to $\mathcal{D}$ represents information from prefix-free $\mathsf{DM\text{-}MD}^E$ to $\mathcal{D}$. In the prefix-free $\mathsf{DM\text{-}MD}^E$ case, $\mathcal{D}$ can make any query to $E$. In the $\mathsf{DM\text{-}MD}^E$ with condition $\alpha$ case, $\mathcal{D}$ can make any query to $E$. From this discussion, we can conclude that the indifferentiability proof of prefix-free $\mathsf{DM\text{-}MD}^E$ can be used as a black box in the indifferentiability proof of $\mathsf{DM\text{-}MD}^E$ with the condition, that is $\mathsf{DM\text{-}MD}^E$ with condition $\alpha \sqsubset \mathcal{RO}$.

For the second consistency, $\mathsf{DM\text{-}MD}^E$ and OAEP are connected by the private channel. $\mathcal{D}$ and $\mathsf{DM\text{-}MD}^E$ are connected by the same private channel from the proof of the indifferentiability framework [11]. Since any value from OAEP to $\mathsf{DM\text{-}MD}^E$ is a fixed length value, we can restrict $\mathcal{D}$ such that it has to make a query under this condition. Since $\mathsf{DM\text{-}MD}^E$ with condition $\alpha$ implies prefix-free $\mathsf{DM\text{-}MD}^E$ and OAEP has condition $\alpha$, $OAEP(\text{prefix-free } \mathsf{DM\text{-}MD}^E) = OAEP(\mathsf{DM\text{-}MD}^E)$ holds. Therefore, since prefix-free $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{RO}$, we can apply the result of $\mathsf{DM\text{-}MD}^E$ with condition $\alpha \sqsubset \mathcal{RO}$ to OAEP. Since prefix-free $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{RO}$ implies that $OAEP(\text{prefix-free } \mathsf{DM\text{-}MD}^E) \succ OAEP(\mathcal{RO})$ holds due to [11], we can get the result that since $\mathsf{DM\text{-}MD}^E$ with $\alpha \sqsubset \mathcal{RO}$ and OAEP has condition $\alpha$, then $OAEP(\mathsf{DM\text{-}MD}^E) \succ OAEP(\mathcal{RO})$ holds.

Considering the above discussion, we will introduce a new concept indifferentiability from $\mathcal{RO}$ with conditions, on the format of the channel between $\mathcal{D}$ and hash function $H\ P$ with underlying primitive $P$.

**Definition 2.** *Hash function $H^P$ is indifferentiable from $\mathcal{RO}$ with condition $\alpha$, denote $\mathcal{V} \sqsubset_\alpha \mathcal{U}$, if, for any distinguisher $D$ with binary output (0 or 1) such that the channel between $\mathcal{D}$ and $H^P/\mathcal{RO}$ has condition $\alpha$, there is a simulator $\mathcal{S}$ such that the advantage $|Pr[D^{H^P,P} \Rightarrow 1] - Pr[D^{\mathcal{RO},S(\mathcal{RO})} \Rightarrow 1]|$ is negligible in the security parameter $k$.*

### 4.3 Generalization of New Approach

In this subsection, we generalize our approach. The procedure of the generalized approach for cryptosystem $\mathcal{C}$ using hash function $H^P$ with underlying primitive $P$ is as follows.

1. Identify condition $\alpha$, that $\mathcal{C}$ satisfies. "$\mathcal{C}$ satisfies $\alpha$" means that there is condition $\alpha$ in the private channel between cryptosystem $\mathcal{C}$ and hash function $H^P$(for OAEP the condition is the fixed length of hash function input).
2. Identify the characteristic of condition $\alpha$.
3. Identify the hash function that has the characteristic identified in step 2 and is indifferentiable from $\mathcal{RO}$.

4. Confirm that $H^P$ with $\alpha$ implies the hash function identified in step 3.
5. Prove $\mathcal{C}$ is secure in the $\mathcal{RO}$ model.
6. This proposed approach of indifferentiability from $\mathcal{RO}$ with conditions allows us to conclude that $\mathcal{C}(H^P)$ is secure.

In step 4, we confirm that the indifferentiability proof of the hash function identified in step 3 from $\mathcal{RO}$ can be used as a black box in the indifferentiability proof of $H^P$ with condition $\alpha$ from $\mathcal{RO}$.

In step 6, we use the following Theorem. The statement of "the $\mathcal{C}$ satisfies condition $\alpha$" means that the private channel between $H^P$ and $\mathcal{C}$ satisfies condition $\alpha$.

**Theorem 5.** *For any $\mathcal{C}$ satisfying condition $\alpha$, $H^P \sqsubset_\alpha \mathcal{RO} \Leftrightarrow \mathcal{C}(H^P) \succ \mathcal{C}(\mathcal{RO})$.*

*Proof.* In indifferentiability proof of [11], the channel between $\mathcal{D}$ and $H^P/\mathcal{RO}$ is the same channel as the private channel between $\mathcal{C}$ and $H^P/\mathcal{RO}$. Therefore, we can use condition $\alpha$ to the proof of [11]. Please see the original proof of indifferentiability of $\mathcal{RO}$ in [11]. □

While our example of OAEP found only "input length of hash functions is fixed" as condition $\alpha$, we believe that there are other conditions $\alpha$ that will make the above procedure workable.

## 5 Conclusion

In this paper we analyze cryptosystems using $\mathsf{DM\text{-}MD}^E$ by two approaches. Our results represent the first succesful rescue of $\mathsf{DM\text{-}MD}^E$ in the indifferentiability framework. The first approach is the indifferentiability from $\widetilde{\mathcal{RO}}$, and the second is the indifferentiability of $\mathcal{RO}$ with conditions.

The former approach is as follows:

1. Find a valiant of random oracle $\widetilde{\mathcal{RO}}$ such that the target hash function ($\mathsf{DM\text{-}MD}^E$) satisfies the indifferentiability from $\widetilde{\mathcal{RO}}$ (this hash function behaves like $\widetilde{\mathcal{RO}}$)
2. Prove that a cryptosystem $\mathcal{C}$ is secure in the $\widetilde{\mathcal{RO}}$ model.

For this approach, we introduced the following procedure.

1. Define the extension property and inverse property simulatable random oracle $\mathcal{EIRO}$ that captures the extension property and the inverse property of $\mathsf{DM\text{-}MD}^E$.
2. Prove $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{EIRO}$ and $\mathcal{EIRO} \sqsubset$ pub-$\mathcal{RO}$.
3. Prove that RSA-KEM is secure in the $\mathcal{EIRO}$ model.

For the result $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{EIRO}$ and $\mathcal{EIRO} \sqsubset$ pub-$\mathcal{RO}$, we can obtain the result $\mathsf{DM\text{-}MD}^E \sqsubset$ pub-$\mathcal{RO}$. Since FDH, PSS, Fiat-Shamir and so on are secure in the pub-$\mathcal{RO}$ model, these cryptosystems are secure under $\mathsf{DM\text{-}MD}^E$. Since RSA-KEM is secure in the $\mathcal{EIRO}$ model, RSA-KEM is secure under $\mathsf{DM\text{-}MD}^E$ due to $\mathsf{DM\text{-}MD}^E \sqsubset \mathcal{EIRO}$. We also showed the following.

1. Prove pub-$\mathcal{RO} \not\sqsubset \mathcal{EIRO}$.
2. RSA-KEM is insecure in the pub-$\mathcal{RO}$ model.

The result pub-$\mathcal{RO} \not\sqsubset \mathcal{EIRO}$ implies that there is some cryptosystem that is secure in the $\mathcal{EIRO}$ model but insecure in the pub-$\mathcal{RO}$ model. RSA-KEM is evidence of the separation between $\mathcal{EIRO}$ and pub-$\mathcal{RO}$. Therefore $\mathcal{EIRO}$ is more useful tool than pub-$\mathcal{RO}$.

For the second approach, we showed that OAEP is secure under $\mathsf{DM\text{-}MD}^E$ by indifferentiability from $\mathcal{RO}$ with conditions agument. Therefore, we can conclude that OAEP is secure under $\mathsf{DM\text{-}MD}^E$.

Therefore OAEP, RSA-KEM, FDH, PSS, Fiat-Shamir and so on are secure under $\mathsf{DM\text{-}MD}^E$.

## References

1. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
2. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *EUROCRYPT*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 1994.
3. Mihir Bellare and Phillip Rogaway. The exact security of digital signatures - how to sign with rsa and rabin. In *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer, 1996.
4. Dan Boneh and Xavier Boyen. Short signatures from the weil pairing. In *ASIACRYPT*, pages 514–532, 2001.
5. Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.
6. Jean-Sébastien Coron. Optimal security proofs for pss and other signature schemes. In *EUROCRYPT*, pages 272–287, 2002.
7. Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-damgård revisited: How to construct a hash function. In *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer, 2005.
8. Ivan Damgård. A design principle for hash functions. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.
9. Yevgeniy Dodis, Thomas Ristenpart, and Thomas Shrimpton. Salvaging merkle-damgård for practical applications. In *EUROCRYPT*, volume ???? of *Lecture Notes in Computer Science*, pages ???–??? Springer, 2009.
10. Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. Rsa-oaep is secure under the rsa assumption. In *CRYPTO*, Lecture Notes in Computer Science, pages 260–274. Springer, 2001.
11. Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.
12. Ralph C. Merkle. One way hash functions and des. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.
13. Bart Preneel, René Govaerts, and Joos Vandewalle. Hash functions based on block ciphers: A synthetic approach. In *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 368–378. Springer, 1993.
14. Shai Halevi Ran Canetti and Jonathan Katz. A forward-secure public-key encryption scheme. In *J. Cryptology*, pages 265–294, 2007.
15. Victor Shoup. A proposal for an iso standard for public key encryption (version 2.1). 2001.
16. Kazuki Yoneyama, Satoshi Miyagawa, and Kazuo Ohta. Leaky random oracle (extended abstract). In *ProvSec*, volume 5324 of *Lecture Notes in Computer Science*, pages 226–240. Springer, 2008.

## A DM is not indifferentiable from FIL pub-$\mathcal{RO}$

It has been proven that DM is not indifferentiable from FIL pub-$\mathcal{RO}$ by the following distinguisher using only two queries. DM is defined by $h' = E_m(h) \oplus h$, where $y = h \oplus h' = E_m(h)$.

The hash list of FIL pub-$\mathcal{RO}$ is initialized as empty. Given a oracle pair $(\mathcal{O}_1, \mathcal{O}_1)$ from either (DM, $E$) or (FIL pub-$\mathcal{RO}$, $S$), the procedure of the distinguisher is as follows.

1. $D$ sends random inverse query $(-, m, y)$ to $\mathcal{O}_2$ to obtain a response $h$.
2. $D$ sends $(h, m)$ to $\mathcal{O}_1$ to obtain a response $h'$.
3. if $h' = h \oplus y$, $D$ determines $(\mathcal{O}_1, \mathcal{O}_1)$ as (DM, $E$). Otherwise $D$ determines $(\mathcal{O}_1, \mathcal{O}_1)$ as (FIL pub-$\mathcal{RO}$, $S$).

As the hash list of FIL pub-$\mathcal{RO}$ is empty, $S$ has to find a suitable $h$, where FIL pub-$\mathcal{RO}$ takes $(h, m)$ as input and outputs $y \oplus h$. No $S$ can find the suitable $h$ in polynomial time, so $D$ can succeed with a great advantage.

## B Proof of Theorem 2

*Proof.* First we define simulator $\mathcal{S}$ as follows. $S$ has a list $\mathcal{T}$ which is initially an empty list.

**Simulator $\mathcal{S}$.**

- On a forward query $(+, m, x)$,
  1. If $\exists(m, x, y) \in \mathcal{T}$, it outputs $y$.
  2. If $x = IV$, it queries $m$ to $\mathcal{RO}$, receives the output $z = \mathcal{RO}(m)$, $\mathcal{T} \leftarrow (m, x, y)$ and responds with $y = x \oplus z$.
  3. If $\exists(m_1, x_1, y_1), ..., (m_i, x_i, y_i) \in \mathcal{T}$ such that $x_j = x_{j-1} \oplus y_{j-1}$ $(j = 1, ..., i)$, $x_1 = IV$ and $x = x_i \oplus y_i$, it queries $m_1||...||m_j||m$ to $\mathcal{RO}$, receives the output $z = \mathcal{RO}(m_1||...||m_i||m)$, $\mathcal{T} \leftarrow (m, x, y)$ and responds with $y = x \oplus z$.
  4. It queries $(m, x)$ to $\mathcal{EO}$, it receives the output $z$, $\mathcal{T} \leftarrow (m, x, y)$ such that $y = x \oplus z$, and responds with $y$.
- On an inverse query $(-, m, y)$,
  1. If $\exists(m, x, y) \in \mathcal{T}$, it outputs $x$.
  2. it queries $(m, y)$ to $\mathcal{IO}$, receives the output $x$, $\mathcal{T} \leftarrow (m, x, y)$, and responds with $x$.

In the worst case of the simulator's running time, the simulator executes step 4 for every query and this requires at most $O(ql)$ time.

We need to prove that $\mathcal{S}$ cannot tell apart the two scenario, one where it has oracle access to $\mathcal{RO}$ and $\mathcal{S}$ and the other where it has access to $\mathsf{DM\text{-}MD}^E$ and $E$. The proof involves a hybrid argument starting in the $\mathcal{RO}$ scenario, and ending in the ideal cipher scenario through a sequence of mutually indistinguishable hybrid games.

**Game 1:** This is the random oracle model, where $\mathcal{D}$ has oracle access to $\mathcal{RO}$ and $\mathcal{S}$. Let $\mathsf{G1}$ denote the event that $D$ output 1 after interacting with $\mathcal{RO}$ and $\mathcal{S}$. Thus $Pr[\mathsf{G1}] = Pr[D^{\mathcal{RO},S(\mathcal{EIRO})} = 1]$.

**Game 2:** In this game, we give the distinguisher oracle access to a dummy relay algorithm $R_0$ instead of direct oracle access to $\mathcal{RO}$. $R_0$ is given oracle access to $\mathcal{RO}$. On query $M$ to $R_0$, it queries $M$ to $\mathcal{RO}$ and returns $\mathcal{RO}(M)$. Let $\mathsf{G2}$ denote the event that $\mathcal{D}$ outputs 1 in the game 2. Since the view of $\mathcal{D}$ remains unchanged in this game, so $Pr[\mathsf{G2}] = Pr[\mathsf{G1}]$.

**Game 3:** In this game, we modify the simulator $\mathcal{S}$ and the relay algorithm. In particular, we restrict the responses of the simulator such that they never satisfy certain specific failure conditions. If the simulator comes up with a response that results in its responses satisfying one of these conditions, then it fails explicitly instead of sending this response.

The failure conditions (that the new simulator $\mathcal{S}_0$ avoids) essentially describe certain dependencies that could arise among its responses that could be exploited by the distinguisher. In response to a forward query $(+, m, x)$, the new simulator chooses a response $y \in \{0, 1\}^n$ simular to $S$ and it checks for the following conditions:

- $\mathsf{FC1}$: It is the case that $y \oplus x = IV$.
- $\mathsf{FC2}$: There is a triple $(m', x', y') \in \mathcal{T}$, with $(m', x') \neq (m, x)$, such that $x' \oplus y' = x \oplus y$.
- $\mathsf{FC3}$: There is a triple $(m', x', y') \in \mathcal{T}$, with $(m', x') \neq (m, x)$, such that $x' = x \oplus y$.

If the response $y$ is chosen by $S_0$ then $S_0$ checks for these conditions and explicitly fails if any of them holds.

If an inverse query $(-, m, y)$ is made to the simulator $S_0$, then it chooses a response $x \in \{0, 1\}^n$ to this query similar to the original simulator $S$ and checks for the following failure conditions:

- $\mathsf{IC1}$: It is the case that $x = IV$ or $y \oplus x = IV$.

- $\mathsf{IC2}$: There is a triple $(m', x', y') \in \mathcal{T}$, with $(m', y') \neq (m, y)$, such that $x' \oplus y' = x \oplus y$.
- $\mathsf{IC3}$: There is a triple $(m', x', y') \in \mathcal{T}$, with $(m', y') \neq (m, y)$, such that $x' \oplus y' = x$ or $x \oplus y = x'$.

In the case of inverse queries, if the response $x$ is chosen by the simulator $\mathcal{S}_0$ then $\mathcal{S}_0$ checks for these conditions and explicitly fails if any of them holds.

We modify the relay algorithm as follows. The underlying idea is to make the responses of the relay algorithm directly dependent on the simulator. Thus, for a hash oracle query $M$, $R_1$ applies the Davies-Meyer Merkle-Damgård construction to $M$ by querying $\mathcal{S}_0$. $R_1$ is essentially the same as $\mathsf{DM}\text{-}\mathsf{MD}^E$ except that it is based on $\mathcal{S}_0$ instead of the ideal cipher $E$.

**Bad event of game 3:** In game 3, we consider the bad event of game 3. Let $(m, x, y)$ be the input-output pair of $\mathcal{S}_0$ due to a (forward or inverse) query $\mathcal{D}$. The bad event $\mathsf{B}$ occurs if $\mathcal{D}$ makes a hash query $M$ such that the last block pair of $R_1(M)$ is $(m, x, y)$ (namely $x \oplus y = R_1(M)$) and a corresponding query to $\mathcal{S}$ with $(m, x, y)$ (that is a forward query $(+, m, x)$ or an inverse query $(-, m, y)$.).

**Bad event of game 2:** As the preliminary, we consider the bad event of game 2. The bad event occurs if a response of $\mathcal{S}$ to a query $(x, m)$ satisfies some condition in $\mathsf{FC1}, \mathsf{FC2}, \mathsf{FC3}, \mathsf{IC1}, \mathsf{IC2}$ and $\mathsf{IC3}$, outputs of $R_0$ collide ($\mathsf{B1}$), some output of $R_0$ is equal to $IV$ ($\mathsf{B2}$), or some output of $\mathcal{RO}$ is equal to $x \oplus y$ or $x$ where $(m, x, y) \in \mathcal{T}$ ($\mathsf{B3}$).

Now we will show that the view of $\mathcal{D}$ remains unchange up to a negligible additive factor in the transformation from game 2 to game 3. We will assume that maximum block length of a hash input queried upon by $\mathcal{D}$ is $l$. Let the number of hash queries made by the distinguisher be $q_H$, and the number of queries to the simulator be $q_E$. Namely $q = q_H + q_E$. Let $\mathsf{G3}$ denote the event that $\mathcal{D}$ outputs 1 in game 3. We will show that $|Pr[\mathsf{G3}] - Pr[\mathsf{G2}]| = O(\frac{l^2 q^2}{2^n})$ as follows.

From the view of $\mathcal{D}$, the game 2 and game 3 differ only if $\mathcal{D}$ detects any difference in the responses of the relay algorithm or the simulator in these two games. We will prove that the view of $\mathcal{D}$ in game 3 unless $\mathcal{S}_0$ fails and the bad event of game 3 occurs is equal to the view of $\mathcal{D}$ in game 2 unless bad event of game 2 occurs. And we will show that the probability that $\mathcal{S}_0$ fails is negligible and the probability that bad events occur is negligible.

We give queries of $\mathcal{D}$ which don't help to distinguish $\mathsf{G2}$ and $\mathsf{G3}$. We consider following trivial queries.

- **Type 1:** $\mathcal{D}$ makes the ordered sequences of forward queries $(+, m_1, x_1), ..., (+, m_i, x_i)$ such that (for $j = 1, ...i-1$) $x_{j+1}$ is equal to $y_j \oplus x_j$ ($y_j$ is the response of the forward query $(+, m_j, x_j)$) and $x_1 = IV$. Then we say that the fresh hash query $m_1||...||m_i$ is a trivial query if the trivial query is made when above queries were already made.
- **Type 2:** $\mathcal{D}$ makes the hash query $m_1||...||m_i$ and the ordered sequences of fresh forward queries $(+, m_1, x_1), ..., (+, m_{i-1}, x_{i-1})$ such that (for $j = 1, ..., i-2$) $x_{j+1}$ is equal to $y_j \oplus x_j$ ($y_j$ is the response of the forward query $(+, m_j, x_j)$) and $x_1 = IV$. Then we say that the fresh forward query $(+, m_i, x_i)$ such that $x_i = x_{i-1} \oplus y_{i-1}$ is a trivial query if the trivial query is made when above queries were already made.
- **Type 3:** $\mathcal{D}$ makes the ordered sequences of a fresh hash query $M$ and fresh forward queries $(+, m_1, x_1)$, ..., $(+, m_i, x_i)$ such that $z = x_1$ ($z$ is the response of $M$) and (for $j = 1, ..., i-1$) $x_{j+1}$ is equal to $x_j \oplus y_j$ ($y_j$ is the response of the query $(x_j, m_j)$). Then we say that the fresh hash query $M||m_1||...||m_i$ is a trivial query if the trivial query is made when above queries were already made.
- **Type 4:** $\mathcal{D}$ makes the fresh query $M||m_1||...||m_i$ and the ordered sequences of the fresh hash query $M$ and fresh queries $(+, m_1, x_1)$, ..., $(+, m_{i-1}, x_{i-1})$ such that $z = x_1$ ($z$ is the response of $M$) and (for $j = 1, ..., i-2$) $x_{j+1}$ is equal to $y_j \oplus x_j$ ($y_j$ is the response of the forward query $(+, m_j, x_j)$). Then we say that the forward query $(+, m_i, x_i)$ such that $x_i = x_{i-1} \oplus y_{i-1}$ is a trivial query if the trivial query is made when above queries were already made.
- **Type 5:** $\mathcal{D}$ makes the hash queries $M$ and $M||m$ where the response of $M$ is $z$ and one of $M||m$ is $z'$. Then we say that the inverse query $(-, m, z \oplus z')$ is a trivial query if the trivial query is made when above queries were already made.

- **Type 6:** $\mathcal{D}$ makes the ordered sequences of the fresh hash queries $M$ and the fresh forward queries $(+, m_1, x_1), ..., (+, m_i, x_i)$ and $M||m_1||...||m_i||m$. We denote the response of $M$ by $z$ and the response of $M||m_1||...||m_i||m$ by $z'$. Then we say that the inverse query $(-, m, z \oplus z')$ is a trivial query if the trivial query is made when above queries were already made.

Above queries do not help $\mathcal{D}$ to distinguish game 2 and game 3, since the following claim is true both game 2 and game 3 due to Davies-Meyer Merkle-Damgård construction and constructions of $\mathcal{S}$ and $\mathcal{S}_0$.

- For type 1, the response of the hash query $m_1||...||m_i$ is equal to $x_i \oplus y_i$ due to step 2 or step 3 of the simulator.
- For type 2, $x_i \oplus y_i$ ($y_i$ is the response of the forward query $(+, m_i, x_i)$ ) is equal to the response of the hash query $m_1||...||m_i$ due to step 2 or step 3 of the simulator.
- For type 3, the response of the hash query $M||m_1||...||m_i$ is equal to $x_i \oplus y_i$ due to step 2 of $\mathcal{EO}$.
- For type 4, the response of the query $(+, m_i, x_i)$ is equal to the response of the hash query $M||m_1||...||m_i$ due to step 2 of $\mathcal{EO}$.
- For type 5 and 6, the response of the inverse query $(+, m, z \oplus z')$ is equal to $z$ due to step 2 of $\mathcal{IO}$.

Therefore we hereafter for simplicity

- **$\mathcal{D}$ does not make a trivial query and a repeated query**.

Note that we suppose that following queries is also repeated query: $\mathcal{D}$ makes an inverse query $(-, m, y)$ when $\mathcal{D}$ already made a forward query $(+, m, x)$ where $x$ is the response of the forward query, and $\mathcal{D}$ makes a forward query $(+, m, x)$ when $\mathcal{D}$ already made an inverse query $(-, m, y)$ where $x$ is the response of the inverse query. Since the simulator uses the same table $\mathcal{T}$ both a forward query and an inverse query, these queries are not helpful for $\mathcal{D}$. In the following discussion, we prove $|Pr[\mathsf{G3}] - Pr[\mathsf{G2}]| = O(q^2 l^2 / 2^n)$ by the following approach.

- **First stage:** The view of $\mathcal{D}$ in game 3 unless $\mathcal{S}_0$ fails is equal to the view of $\mathcal{D}$ in game 2 unless bad event occurs.
- **Second stage:** We estimate the probabilities that the bad event occurs or $\mathcal{S}_0$ fails.

From the first stage, the case that $\mathcal{D}$ can distinguish game 2 and game 3 is that the bad event occurs and $\mathcal{S}_0$ fails. In the second stage, we show that the probability that the bad event occurs or $\mathcal{S}_0$ fails is $O(l^2 q^2 / 2^n)$.

**First Stage:** We show that for any query the output are randomly chosen in game 2 and game 3 unless the bad event occurs and $\mathcal{S}_0$ fails. If we can show it then the view of $\mathcal{D}$ in game 3 unless $\mathcal{S}_0$ fails and the bad event occurs is equal to the view of $\mathcal{D}$ in game 2 unless bad event occurs.

First we show that all responses are chosen at random unless the bad event occurs in game 2. We start by demonstrating an useful properties (claim 1 and claim 2). After showing claim 1 and claim 2, we prove the claim 3 by using claim 1 and claim 2 where all responses to $\mathcal{D}$ is randomly chosen in game 2 unless the bad event occurs.

**Claim 1** *In game 2, if there is $(m_1||...||m_i, z)$ in $\mathcal{L}_{\mathcal{RO}}$ and the bad event does not occur, there is $k \in \{1, ..., i\}$ such that the ordered sequences of query $m_1||...||m_{k-1}$ to $R_0$ and queries $(+, m_k, x_k), ..., (+, m_i, x_i)$ were made such that*

- $x_k = z'$ *($z' = \mathcal{RO}(m_1||...||m_{k-1})$),*
- *(for $j = k+1, ..., i$) $x_j = x_{j-1} \oplus y_{j-1}$, and*
- $x_i \oplus y_i = z$

*where $y_j$ is a response of a forwar query $(+, m_i, x_i)$ to $\mathcal{S}$. Note that if $k = 0$, then a query to $R_0$ is ignored, if $k = i+1$, then queries to $\mathcal{S}$ are ignored.*

*Proof.* The pair $(m_1||...||m_i, z)$ is stored in $\mathcal{L}_{\mathcal{RO}}$ by making a query to $R_0$ or $\mathcal{S}$. By considering the definition of $\mathcal{S}$, the way to store $(m_1||...||m_i, z)$ in $\mathcal{L}_{\mathcal{RO}}$ is explicitly to make the ordered sequences of a query $m_1||...||m_{k-1}$ to $R_0$ and forward queries $(+, m_k, x_k), ..., (+, m_i, x_i)$ such that $x_k = \mathcal{RO}(m_1||...||m_{k-1})$, (for $j = k+1, ..., i$) $x_j = x_{j-1} \oplus y_{j-1}$, and $x_i \oplus y_i = z$ where $y_j$ is a response of a query $(x_j, m_j)$ unless the bad event of game 2 occurs. If there is no $k \in \{1, ..., i\}$ such that the ordered sequences of a query $m_1||...||m_{k-1}$ to $R_0$ and queries $(+, m_k, x_k), ..., (+, m_k, x_i)$ were made such that

- $x_k = z'$ $(z' = \mathcal{RO}(m_1||...||m_{k-1}))$,
- for $j = k+1, ..., i)$ $x_j = x_{j-1} \oplus y_{j-1}$, and
- $x_i \oplus y_i = z$

where $y_j$ is a response of a query $(+, m_j, x_j)$ to $\mathcal{S}$, by considering the condition invoking each step of $\mathcal{S}$, we can deduce that the bad event occurs, there is no pair $(m_1||...||m_i, z)$ in $\mathcal{L}_{\mathcal{RO}}$ or the bad event occurs. □

**Claim 2** *In game 2, step 2 of $\mathcal{IO}$ are not invoked.*

*Proof.* On an inverse query $(-, m, y)$ to $\mathcal{IO}$, the condition that invokes step 2 of $\mathcal{IO}$ is that there are pairs $(M, x)$ and $(M', x')$ in $\mathcal{L}_{\mathcal{RO}}$ such that $M||m = M'$ and $y = x \oplus x'$. Let $M$ be $m_1||...||m_i$ and $M'$ be $m_1||...||m_i||m$. Since the pair $(M, x)$ is in $\mathcal{L}_{\mathcal{RO}}$, $\mathcal{D}$ makes a query $m_1||...||m_{k-1}$ to $R_0$ and forward queries $(+, m_k, x_k), ..., (+, m_i, x_i)$ that satisfy conditions of claim 1. Since the pair $(M', x')$ is stored in $\mathcal{L}_{\mathcal{RO}}$, $\mathcal{D}$ makes a query $m_1||...||m_{s-1}$ to $R_0$ and forward queries $(+, m_s, x_s), ..., (+, m_i, x_i), (+, m, x)$ that satisfy conditions of claim 1. This implies that

1. $(m, x, y)$ is already stored in $\mathcal{T}$ when the inverse query $(-, m, y)$ is made or
2. $(m, x, y)$ is stored in $\mathcal{T}$ when the hash queries $M$ and $M||m$ were already made.

Therefore, for the first case, step 2 of $\mathcal{IO}$ is not invoked but step 1 of an inverse query of $\mathcal{S}$ is invoked. For the second case, $\mathcal{D}$ makes a trivial query, so this contradicts assumptions of $\mathcal{D}$. □

We show that all responses of $\mathcal{S}$ and $R_0$ are randomly chosen unless bad event occurs by using claim 1 and claim 2.

**Claim 3** *In game 2, all responses of $\mathcal{S}$ and $R_0$ are randomly chosen unless the bad event occurs.*

*Proof.* We prove the following claims.

- For any two different query-response pairs $(m, x, y), (m', x', y')$ from $\mathcal{D}$ to $\mathcal{S}$, $x \oplus y \neq x' \oplus y'$ holds. If this is true, all responses from $\mathcal{S}$ to $\mathcal{D}$ are independently chosen.
- All responses of $R_0$ are different unless the bad event occurs. If this is true, outputs of $R_0$ $(\mathcal{RO})$ are independently chosen.
- For any forward query $(+, m, x)$ and the response $y$, $x \oplus y$ is different from any response of $R_0$ unless the bad event occurs. If this is true, any output of $R_0$ $(\mathcal{RO})$ is independently chosen from any response of a forward query.

Since $\mathcal{D}$ does not make repeated queries, step 1 of $\mathcal{IO}$ are not invoked. Step 2 of $\mathcal{IO}$ is not invoked from claim 2. Therefore any response of an inverse query is chosen at random independently from any response of a hash query and a forward query due to the definition of step 3 of $\mathcal{IO}$. Therefore, $\mathcal{IO}$ does not invoke $\mathcal{RO}$. Namely for any inverse query the response is independently chosen from $\mathcal{RO}$. Consecuently, if we show the above claims, all responses are randomly chosen, since any fresh output is randomly chosen due to definitions of $\mathcal{RO}$, $\mathcal{EO}$ and $\mathcal{IO}$.

If the bad event (of FC2 and $\mathcal{IO}$2) does not occur, the first claim is explicitly valid. For the second claim, all responses of $R_0$ $(\mathcal{RO})$ are different since the bad event (of B1) does not occur. For the third claim, we prove this as follows.

To the contrary, assume that, for some forward query $(+, m, x)$ and the response $y$, $x \oplus y$ is equal to some response $z$ of $R_1$. Let $m_1||...||m_i$ be the query of the response $z$. From the claim 1, $\mathcal{D}$ makes a query $m_1||...||m_{k-1}$ to $R_0$ and forward queries $(+, m_k, x_k), ..., (+, m_i, x_i)$ by this order Then we can deduce that one of the following two cases occurs for choosing $y$ and $z$

1. $y$ and $z$ are independently chosen at random but $x \oplus y = z$.
2. $y$ and $z$ are dependently chosen and $x \oplus y = z$.

In the first case, since $y$ and $z$ are independently chosen at random, the bad event $\mathsf{B3}$ explicitly occurs in the first case.

In the second case, by considering the method of choosing $y$, $(m_1||...||m_{i-1}, x), (m_1||...||m_i, z)$ is stored in $\mathcal{L_{RO}}$ when the query $(+, m, x)$ is made. From claim 1, we can deduce that the hash query $m_1||...||m_{k-1}$ and forward queries $(+, x_k, m_k), ..., (+, x_{i-1}, m_{i-1})$ are made by this order and the hash query $m_1||...||m_{s-1}$ and forward queries $(+, x_s, m_s), ..., (+, x_i, m_i)$ are made by this order such that these queries satisfy the condition of claim 1. Therefore the forward query $(+, m, x)$ is a trivial query or a repeated query. This contradicts assumptions of $\mathcal{D}$.

From above discussions, we can conclude that all responses are randomly chosen unless the bad event occurs. □

Next we discuss the case of game 3. We start by demonstrating an useful properties (claim 4, claim 5 and claim 6). We show that all responses to $\mathcal{D}$ are randomly chosen unless $\mathcal{S}_0$ fails by the following approach.

**Claim 7** We prove that a response for any fresh query to $\mathcal{S}_0$ is chosen at random.
**Claim 8** We prove that all responses of $R_1$ are different unless $\mathcal{S}_0$ fails.
**Claim 9** We prove that for any different query and response pairs $(m, x, y)$ and $(m', x', y')$ to $\mathcal{S}_0$ made by $\mathcal{D}$, $x \oplus y \neq x' \oplus y'$ holds unless $\mathcal{S}_0$ fails.
**Claim 10** We prove that for any forward query $(+, m, x)$ and the response $y$ of $\mathcal{S}_0$ to $\mathcal{D}$ $x \oplus y$ is different from any response of $R_1$ to $\mathcal{D}$ unless $\mathcal{S}_0$ fails and the bad event $\mathsf{B}$ occurs, and that for any inverse query $(-, m, y)$ and the response $x$ of $\mathcal{S}_0$ to $\mathcal{D}$ $x \oplus y$ is different from any response of $R_1$ to $\mathcal{D}$ unless $\mathcal{S}_0$ fails and the bad event $\mathsf{B}$ occurs.

Claim 8-10 imply that all responses to $\mathcal{D}$ are independently chosen. By combining claim 7-10, all responses to $\mathcal{D}$ are randomly chosen.

**Claim 4** *In game 3, if there is $(m_1||...||m_i, z)$ in $\mathcal{L_{RO}}$ and $\mathcal{S}_0$ does not fail, there is the pairs $(m_1, x_1, y_1), ...,$ $(m_i, x_i, y_i)$ in $\mathcal{T}$ such that*

- $x_1 = IV$,
- *(for $j = 2, ..., i$)* $x_j = x_{i-1} \oplus y_{j-1}$,
- $x_i \oplus y_i = z$ *and*
- *these pairs were stored as a result of the ordered sequence of forward queries $(m_1, x_1), ..., (m_i, x_i)$.*

*And $(m_1||...||m_i, z)$ is stored in $\mathcal{L_{RO}}$ due to the ordered sequences of hash query $m_1||...||m_{k-1}$ to $R_1$ and forward queries from $\mathcal{D}$ to $\mathcal{S}$ $(+, m_k, x_k), ..., (+, m_i, x_i)$. Note that if $k = 0$, then a query to $R_1$ is ignored, if $k = i + 1$, then queries from $\mathcal{D}$ to $\mathcal{S}_0$ are ignored.*

*Proof.* Since $R_1$ is the Davies-Meyer Merkle-Damgård construction by making forward queries to $\mathcal{S}_0$, $\mathcal{RO}$ is only invoked by $\mathcal{S}_0$. Therefore, for any $(m_1||...||m_i, z)$ in $\mathcal{L_{RO}}$, there are pairs $(m_1, x_1, y_1), ..., (m_i, x_1, y_i)$ in $\mathcal{T}$ such that $x_1 = IV$, (for $j = 2, ..., i$) $x_j = x_{i-1} \oplus y_{j-1}$, and $x_i \oplus y_i = z$ due to the definition of $\mathcal{S}_0$. If these pairs were not stored as a result of the ordered sequence of forward queries $(+, m_1, x_1), ..., (+, m_i, x_i)$ to $\mathcal{S}_0$, we can deduce that the following must be true regarding the sequence of queries:

- for some $s \in \{1, ..., i-1\}$, the pair $(m_s, x_s, y_s)$ is stored when the pair $(x_{s+1}, m_{s+1}, y_{s+1})$ was already stored in $\mathcal{T}$, or
- there is no pair $(m_1||...||m_i, z)$ in $\mathcal{L_{RO}}$ due to the definition of $\mathcal{S}_0$:

In the first case, $\mathcal{S}_0$ explicitly fails due to $\mathsf{FC3}$ or $\mathsf{IC3}$. If these pairs were not stored as a result of the ordered sequence of queries $(+, m_1, x_1), ..., (+, m_i, x_i)$ to $\mathcal{S}_0$, $\mathcal{S}_0$ explicitly fails due to $\mathsf{FC3}$ or $\mathsf{IC3}$. Therefore, $(m_1||...||m_i, z)$ was explicitly stored as a result of the ordered sequence of forward queries $(m_1, x_1), ..., (m_i, x_i)$ to $\mathcal{S}_0$.

The pair $(m_1||...||m_i, z)$ is stored in $\mathcal{L_{RO}}$ by making a query to $R_1$ or $\mathcal{S}_0$. By considering the definition of $\mathcal{S}_0$, the way to store $(m_1||...||m_i, z)$ in $\mathcal{L_{RO}}$ is explicitly to make the ordered sequences of a hash query $m_1||...||m_{k-1}$ to $R_1$ and forward queries $(+, m_k, x_k), ..., (+, m_i, x_i)$. □

**Claim 5** *In game 3, step 2 of $\mathcal{EO}$ is not invoked unless $\mathcal{S}_0$.*

*Proof.* Recall the condition invoking step 2 of $\mathcal{EO}$: for query $(m, z)$ to $\mathcal{EO}$, there is a pair $(M, z)$ in $\mathcal{L}_{\mathcal{RO}}$. From the definition of $\mathcal{S}_0$, step 2 of $\mathcal{EO}$ is invoked due to the forward query $(+, m, z)$ to $\mathcal{S}_0$. However, if there is a pair $(M, z)$ in $\mathcal{L}_{\mathcal{RO}}$, it must be the case that there are $(m_1, x_1, y_1), ..., (m_i, x_i, y_i)$ in $\mathcal{T}$ such that these pairs satisfy conditions of claim 4. Therefore, since there are pair $(m_1, x_1, y_1), ..., (m_i, x_i, y_i)$ in $\mathcal{T}$, step 2 of $\mathcal{EO}$ is not invoked but step 3 of $\mathcal{S}_0$ is invoked. □

**Claim 6** *In game 3, step 2 of $\mathcal{IO}$ is not invoked.*

*Proof.* On an inverse query $(-, m, y)$ to $\mathcal{IO}$, the condition invoking step 2 of $\mathcal{IO}$ is that there are pairs $(M, x), (M', x')$ in $\mathcal{L}_{\mathcal{RO}}$ such that $M||m = M'$ and $y = x \oplus x'$. The pair $(M, x)$ is stored in $\mathcal{L}_{\mathcal{RO}}$ by a hash query $m_1||...||m_{k-1}$ and forward queries $(+, m_k, x_k), ..., (+, m_i, x_i)$ by this order from claim 1. The pair $(M', x')$ is stored in $\mathcal{L}_{\mathcal{RO}}$ by a hash query $m_1||...||m_{k-1}$ and forward queries $(+, m_k, x_k), ..., (+, m_i, x_i), (+, m, x)$ by this order from claim 1. Therefore the inverse query $(-, m, y)$ is explicitly trivial query type 5 or type 6. This contradicts assumptions of $\mathcal{D}$. □

**Claim 7** *In game 3, on a fresh forward query $(+, m, x)$ to $\mathcal{S}_0$, if there is no pair $(m, x, y)$ in $\mathcal{T}$ (namely $(+, m, x)$ is a fresh query), its response $y$ is chosen at random. And on a fresh inverse query $(-, m', y')$ to $\mathcal{S}_0$, if there is no pair $(m', x', y')$ in $\mathcal{T}$ (namely $(-, m', y')$ is a fresh query), its response $x'$ is chosen at random.*

*Proof.* First we consider a forward query. We consider the cases that $y$ is defined during step 2-3 of $\mathcal{S}_0$ and show that $y$ is chosen at random for all cases. The reason why we only consider step 2-3 is as follows. For step 4 of $\mathcal{S}_0$, step 1 and step 2 of $\mathcal{EO}$ are not invoked by the following reasons. Step 2 of $\mathcal{EO}$ is not invoked from claim 5. Step 1 of $\mathcal{EO}$ is not invoked since if $(m, x, y) \in \mathcal{L}_{\mathcal{EO}}$ then it should be the case that $(m, x, y) \in \mathcal{T}$. In step 3 of $\mathcal{EO}$ any response is explicitly chosen at random. Therefore all responses chosen in step 4 are chosen at random. So we consider during step 2-3 of $\mathcal{S}_0$ is that $\mathcal{RO}$ is only invoked in step 2-3.

- The case that $y$ is chosen in step 2 (namely $x = IV$):
  - The case that $(m, y) \in \mathcal{L}_{\mathcal{RO}}$: This case does not occur, since $(m, x, y)$ is already in table $\mathcal{T}$ due to claim 4, namely step 2 is not invoked but step 1 is invoked.
  - The case that $(m, y) \notin \mathcal{L}_{\mathcal{RO}}$: In this case, $y$ is chosen at random by $\mathcal{RO}$.
- The case that $y$ is chosen in step 3:
  - The case that $(m_1||...||m_k||m, z) \in \mathcal{L}_{\mathcal{RO}}$: From claim 3, there are inner pairs $(m_1, x_1, y_1), ..., (m_k, x_k, y_k)$, $(m, x, y)$ of $R_1(m_1||...||m_k||m)$ in $\mathcal{T}$. Therefore step 3 is not invoked but step 1 is invoked.
  - The case that $(m_1||...||m_k||m, z) \notin \mathcal{L}_{\mathcal{RO}}$: $y$ is chosen at random by $\mathcal{RO}$.

Therefore, for any forward query, if there is no pair $(m, x, y)$ in $\mathcal{T}$, its response $y$ is chosen at random.

Second we consider an inverse query. From claim 6, step 2 of $\mathcal{IO}$ is not invoked. When the corresponding query with the repeated query to $\mathcal{IO}$ is made to $\mathcal{S}_0$, it is explicitly the repeated inverse query to $\mathcal{S}_0$. Step 1 of $\mathcal{IO}$ is not invoked since if $(m, x, y) \in \mathcal{L}_{\mathcal{IO}}$ then it should be the case that $(m, x, y) \in \mathcal{T}$. Responses chosen step 3 of $\mathcal{IO}$ are chosen at random due to the definition of $\mathcal{IO}$. Since we assume that $\mathcal{D}$ does not a repeated query, step 1 of $\mathcal{S}_0$ is not invoked. Therefore, for any inverse query, if there is no pair $(m', x', y')$ in $\mathcal{T}$, its response $x'$ is chosen at random. □

**Claim 8** *In game 3 if $\mathcal{S}_0$ does not explicitly fail, then there are no two different sequences of $t$-bit blocks $m_1, ..., m_i$ and $m'_1, ..., m'_j$ with corresponding triples $(m_1, x_1, y_1), ..., (m_i, x_i, y_i)$ and $(m'_1, x'_1, y'_1), ..., (m'_j, x'_j, y'_j)$ in table $\mathcal{T}$ such that:*

- *It is the case that $x_1 = x'_1 = IV$, and for each $s = 1, ..., i$ and $s' = 1, ..., j$, $x_s = x_{s-1} \oplus y_{s-1}$ and $x_{s'} = x_{s'-1} \oplus y_{s'-1}$.*
- *$x_i \oplus y_i = x'_j \oplus y'_j$ holds*

*Proof.* We will prove this claim by performing an induction on the number of queries made to $\mathcal{S}_0$, and show that unless $\mathcal{S}_0$ explicitly fails, such sequence of triples cannot exists in the table $\mathcal{T}$ maintained by it.

Say there are two sequences of $t$-bit blocks $m_1, ..., m_i$ and $m'_1, ..., m'_j$ that satisfy the properties mentioned in the statement of the claim. Without loss of generality, assume that $i \leq j$.

Since $m_1, ..., m_i$ and $m'_1, ..., m'_j$ are different sequences of $t$-bit blocks, there exists $r$ such that $x_{j-r} = IV$ or $x_{i-r} \oplus y_{i-r} = x_{j-r} \oplus y'_{j-r}$ such that $(m_{i-r}, x_{i-r}) \neq (m'_{j-r}, x'_{j-r})$ (consider the output of each iteration of the scheme going backward). If such $r$ does not exists, $(m_1, ..., m_i) = (m'_1, ..., m'_j)$ explicitly holds. If $x_{j-r} = IV$, then we can deduce that $x_{j-r-1} \oplus y_{j-r-1} = IV$. Therefore, in this case, $\mathcal{S}_0$ would have explicitly failed because of failure condition FC1 or IC1. If $x_{i-r} \oplus y_{i-r} = x_{j-r} \oplus y'_{j-r}$ where $m'_{i-r}, (x'_{i-r}) \neq (m'_{j-r}, x'_{j-r})$, then $\mathcal{S}_0$ would have explicitly failed because of failure condition FC2 or IC2. □

**Claim 9** *Let In game 3, if $\mathcal{S}_0$ does not fail, for any different query-response pairs $(m, x, y), (m', x', y')$ from $\mathcal{D}$ to $\mathcal{S}_0$, $x \oplus y \neq x' \oplus y'$ holds.*

Claim 9 explicitly implies that responses for all non repeated queries are independently chosen.

*Proof.* This proof is trivial due to FC2 and IC2. □

**Claim 10** *In game 3, if $\mathcal{S}_0$ does not fail and the bad event* B *does not occur, for any (forward or inverse) query-response pair $(m, x, y)$ from $\mathcal{D}$ to $\mathcal{S}_0$ $x \oplus y$ is different from any response of $R_1$ to $\mathcal{D}$.*

*Proof.* To the contrary, assume that $x \oplus y$ is equal to some response $z$ of $R_1$ where the pair $(m, x, y)$ is some query-response pair from $\mathcal{D}$ to $\mathcal{S}_0$. Then we can deduce that one of the following cases occurs. Let $m_1||...||m_i$ be the hash query of the response $z$, $(m_1, x_1, y_1), ..., (m_i, x_i, y_i)$ be the inner pair of $R_1(m_1||...||m_i)$ such that $x_1 = IV$, $x_j = x_{j-1} \oplus y_{j-1}$ $(j = 2, ..., i)$ and $x_i \oplus y_i = z$. Then we can deduce that one of the following two cases occur for choosing $z$ and $y$.

1. $x \oplus y$ and $z$ are independently chosen but $x \oplus y = z$.
2. $x \oplus y$ and $z$ are dependently chosen and $x \oplus y = z$. This means that $x \oplus y$ and $z$ are chosen from the same pair of $\mathcal{L}_{\mathcal{RO}}$. Namely $(m, x, y) = (m_i, x_i, y_i)$ holds.

In the first case, since $x \oplus y$ and $z$ are chosen at random, $(m, x, y) \neq (m_i, x_i, y_i)$ explicitly holds. Therefore FC2 or IC2 explicitly occurs in the first case. Namely $\mathcal{S}_0$ fails.

From claim 4, ordered sequences of the hash query $m_1||...||m_{k-1}$ and forward queries $(+, m_k, x_k), ..., (+, m_i, x_i)$ are made. If these pairs were not stored these orderd sequences, $\mathcal{S}_0$ explicitly fails. Note that if $k = i + 1$, forward queries are ignored, if $k = 1$, the hash query is ignored. If $k = i + 1$, the bad event B explicitly occurs. If $k \neq i + 1$, $\mathcal{D}$ explicitly makes a trivial query.

Therefore, if $x \oplus y$ is equal to some response $z$ of $R_1$, then $\mathcal{S}_0$ fails, $\mathcal{D}$ makes a trivial query, or the bad event B occurs. □

**Claim 11** *All responses of $\mathcal{S}_0$ and $R_1$ are randomly chosen unless $\mathcal{S}_0$ fails.*

*Proof.* From claim 8, 9, and 10, all responses of $\mathcal{S}_0$ and $R_1$ are independently chosen. From claim 7, the output of $\mathcal{S}_0$ is randomly chosen. By combining these claims, we can conclude that all responses are randomly chosen. □

Since all responses in game 2 and game 3 are chosen at random from claim 3 and claim 11 unless $\mathcal{S}_0$ fails and the bad event occurs, the view of $\mathcal{D}$ in game 3 is equal to the view of $\mathcal{D}$ in game 2.

**Second Stage:** Finally we estimate the probability that the bad event occurs and $\mathcal{S}_0$ fails.

We estimate the probability that the bad event occurs in game 2. From the claim 3, all responses are chosen at random until the bad event occurs. Therefore the probability that the bad event of FC1 occurs is $O(q/2^n)$, the probability that the bad event of FC2 occurs is $O(q^2/2^n)$, the probability that the bad event of $\mathsf{FC3}$ occurs is $O(q^2/2^n)$, the probability that the bad event of $\mathsf{B1}$ occurs is $O(q^2/2^n)$, the probability that the bad event of $\mathsf{B2}$ occurs is $O(q^2/2^n)$, and the probability that the bad event of $\mathsf{B3}$ occurs is $O(q^2/2^n)$.

Next we estimate the probability that $\mathcal{S}_0$ fails and the bad event occurs in game 3. From the claim 5, all fresh responses of $\mathcal{S}_0$ are chosen at random until $\mathcal{S}_0$ fails. The maximum number of invoking $\mathcal{S}_0$ is $lq_H + q_h$. Therefore the probability that $\mathcal{S}_0$ fails by $\mathsf{FC1}$ is $O((lq_H+q_h)/2^n)$, the probability that $\mathcal{S}_0$ fails by $\mathsf{FC2}$ occurs is $O((lq_H + q_h)^2/2^n)$, the probability that $\mathcal{S}_0$ fails by $\mathsf{FC3}$ occurs is $O((lq_H + q_h)^2/2^n)$, the probability that the bad event $\mathsf{B}$ occurs is $O((lq_H + q_h)^2/2^n)$.

Therefore $|Pr[\mathsf{G3}] - Pr[\mathsf{G2}]| = O((lq_H + q_h)^2/2^n) = O(l^2q^2/2^n)$.

**Game 4.** In this game, we modify the simulator $\mathcal{S}_0$ so as to make its responses independent of $\mathcal{RO}$. For this purpose, we remove $\mathcal{RO}$ from this game entirely and the new simulator $\mathcal{S}_1$ always chosen a uniformly random $n$-bit string. Thus on a forward query $(+, m, x)$, $\mathcal{S}_1$ check if there is a triple $(m, x, y)$ in table $\mathcal{T}$. If it finds such a triple then it responds with the $n$-bit string $y$. Otherwise it chooses a uniformly random $n$-bit string and send this as the response, while string the triple $(m, x, y)$ in table $\mathcal{T}$. Thus on an inverse query $(-, m, y)$, $\mathcal{S}_1$ check if there is a triple $(m, x, y)$ in table $\mathcal{T}$. If it finds such a triple then it responds with the $n$-bit string $x$. Otherwise it chooses a uniformly random $n$-bit string and send this as the response, while string the triple $(m, x, y)$ in table $\mathcal{T}$. Note that we remove conditions $\mathsf{FC1}, \mathsf{FC2}, \mathsf{FC3}, \mathsf{IC1}, \mathsf{IC2}$ and $\mathsf{IC3}$ and bad event $\mathsf{B}$.

From claim 7, all responses of $\mathcal{S}_1$ that are not chosen from table $\mathcal{T}$ are chosen at random. But the responses of $\mathcal{S}_0$ and $\mathcal{S}_1$ are identical apart from the failure conditions which are used by $\mathcal{S}_0$ but not by $\mathcal{S}_1$. Thus the distinguisher does not notice a difference between these games if:

- In game 3, $\mathcal{S}_0$ does not fail and $\mathsf{B}$ does not occur.
- In game 4, $\mathcal{S}_1$ does not respond to its queries in such a manner that its satisfy one of the failure conditions specified in the definition of $S_0$ and $\mathsf{B}$ does not occur.

In fact, these two events are identical in terms of their probability of occurrence since the distribution of the responses of the two simulators is identical. Let $\mathsf{G4}$ denote the event that the distinguisher $\mathcal{D}$ outputs 1 in game 4. Then we can deduce that, $|Pr[\mathsf{G4}] - Pr[\mathsf{G3}]| = O(l^2q^2/2^n)$

**Game 5.** This is the final game of our argument. Here we finally replace $\mathcal{S}_1$ with the ideal cipher $E$. The outputs of the ideal cipher $E$ are not distributed uniformly like the responses of $\mathcal{S}_1$. Hence $\mathcal{D}$ may be able to differentiate between game 4 and game 5 if it can detect this. However, this happens only if $\mathcal{S}_1$ outputs an input/output collision for the same ideal cipher key. The probability of this event is easily seen to be at most the birthday bound. Let $\mathsf{G5}$ denote the event that the distinguisher $\mathcal{D}$ outputs 1 in game 5. Then we can deduce that $|Pr[\mathsf{G5}] - Pr[\mathsf{G4}]| = (O(q^2l^2/2^n))$

Now we can complete the proof of theorem by combining game 1 to 5, and observing that game 1 is same as $\mathcal{ERO}$ model while game 5 is same as the ideal cipher model. Hence we can deduce that $\epsilon = O(q^2l^2/2^n)$. □

## C Proof of Theorem 4 [15]

### C.1 Security Notion of KEM

First, we briefly review the model and the security notion of KEM schemes.

**Definition 3 (Model for KEM Schemes).**
*A KEM scheme consists of the following 3-tuple* $(\mathbf{KEM.Gen}, \mathbf{KEM.Enc}, \mathbf{KEM.Dec})$*:*

**KEM.Gen** *: a key generation algorithm which on input* $1^k$*, where* $k$ *is the security parameter, outputs a pair of keys* $(ek, dk)$*.* $ek$ *and* $dk$ *are called encryption key and decryption key respectively.*

**KEM.Enc** *: an encryption algorithm which takes as input encryption key* $ek$*, outputs key* $K$ *and ciphertext* $c$*.*

**KEM.Dec :** *a decryption algorithm which takes as input decryption key $dk$ and ciphertext $c$, output key $K$.*

In particular, a scheme which cannot even satisfy one-wayness under chosen plaintext attacks (OW-CPA) cannot be called as a KEM scheme. Generally, indistinguishability under chosen ciphertext attacks (IND-CCA) is recognized as the strongest security notion. Here, we recall definitions of OW-CPA and IND-CCA for KEM as follows.

**Definition 4 (OW-CPA).**
*A KEM scheme is $(t, \epsilon)$-OW-CPA for KEM if the following property holds for a security parameter $k$;*
*For any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\Pr[\ (ek, dk) \leftarrow \mathbf{KEM.Gen}(1^k);\ (state) \leftarrow \mathcal{A}_1(ek);\ (K^*, c^*) \leftarrow \mathbf{KEM.Enc}(ek);\ K' \leftarrow \mathcal{A}_2(c^*, state);\ K' = K^*] \leq \epsilon$, where state is state information which $\mathcal{A}$ wants to preserve from $\mathcal{A}_1$ to $\mathcal{A}_2$ and $\mathcal{A}$ runs in at most $t$ steps.*

**Definition 5 (IND-CCA for KEM).** *A KEM scheme is $(t, \epsilon)$-IND-CCA for KEM if the following property holds for security parameter $k$; For any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $|\Pr[\ (ek, dk) \leftarrow \mathbf{KEM.Gen}(1^k);\ (state) \leftarrow \mathcal{A}_1^{\mathcal{DO}(dk,\cdot)}(ek);\ b \overset{R}{\leftarrow} \{0,1\};\ (K_0^*, c_0^*) \leftarrow \mathbf{KEM.Enc}(ek);\ K_1^* \overset{R}{\leftarrow} \mathcal{K};\ b' \leftarrow \mathcal{A}_2^{\mathcal{DO}(dk,\cdot)}(ek,\ (K_b^*, c_0^*),\ state);\ b' = b] - 1/2| \leq \epsilon$, where $\mathcal{DO}$ is the decryption oracle, $\mathcal{K}$ is the space of key, state is state information which $\mathcal{A}$ wants to preserve from $\mathcal{A}_1$ to $\mathcal{A}_2$ and $\mathcal{A}$ runs in at most $t$ steps. $\mathcal{A}$ cannot submit the ciphertext $c = c_0^*$ to $\mathcal{DO}$.*

## C.2 RSA-KEM

The security of RSA-KEM is based on the RSA assumption.

**Definition 6 (RSA assumption).** *Let $n$ be an RSA modulus that is the product of two large primes $(p, q)$ for security parameter $k$ and $e$ be an exponent such that $gcd(e,\ \phi(n)) = 1$. We say that RSA problem is $(t, \epsilon)$-hard if for any adversary Alg, $\Pr[y \leftarrow \mathbb{Z}_n;\ Alg(n, e, y) = x;\ y \equiv x^e \pmod n] \leq \epsilon$, where Alg runs in at most $t$ steps.*

The description of RSA-KEM is as follows:

**Key generation :** For input $k$, outputs encryption key $(ek = (n, e))$ and decryption key $(dk = d)$ such that $n$ is an RSA modulus that is the product of two large primes $(p, q)$ for security parameter $k$, gcd($e$, $\phi(n)) = 1$ and $ed \equiv 1 \pmod{\phi(n)}$.

**Encryption :** Generates randomness $r \overset{R}{\leftarrow} \mathbb{Z}_n$, computes $c = r^e \mod n$ and $K = H(r)$, and outputs ciphertext $c$ and key $K$ where $H : \mathbb{Z}_n \rightarrow \{0,1\}^k$ is a hash function.

**Decryption :** Upon inputs of ciphertext $c$, computes $r = c^d \mod n$ and outputs $K = H(r)$.

In [15], security of RSA-KEM in the $\mathcal{RO}$ model is proved as follows;

**Lemma 3 (Security of RSA-KEM in the $\mathcal{RO}$ model [15]).** *If RSA problem is hard, then RSA-KEM satisfies IND-CCA for KEM where $H$ is modeled as the $\mathcal{RO}$.*

## C.3 Proof of Theorem 4

*Proof.* Firstly, we transform the experiment of IND-CCA for RSA-KEM to the experiment where queries to $\mathcal{DO}$ and $\mathcal{EO}$ does not give any advantage to the adversary.

Let $\mathsf{Exp0}$ be the initial experiment and $\mathsf{Succ0}$ be the probability that an adversary $\mathcal{A}$ succeeds to guess the bit $b$ in $\mathsf{Exp0}$. $\mathcal{A}$ receives $(K_b^*, c_0^*)$ as the challenge such that $c_0^* = r^{*e}$ for $r^*$.

Let $\mathsf{Exp1}$ be the same experiment as $\mathsf{Exp0}$ except when $\mathcal{A}$ queried $c_0^*$ to $\mathcal{DO}$ before receiving $c_0^*$ as the challenge ciphertext. $\mathsf{Exp1}$ aborts in the above case. Let $\mathsf{Succ1}$ be the probability that $\mathcal{A}$ succeeds to guess the bit $b$ in $\mathsf{Exp1}$ and $E_1$ be the event that the experiment aborts. Then, the probability that the event $E_1$ occurs is equal or lower than $q_D/n$ because $\mathcal{A}$ has no information about the challenge. Thus, we obtain that $|\mathsf{Succ1} - \mathsf{Succ0}| \leq q_D/n$.

Let $\mathsf{Exp2}$ be the same experiment as $\mathsf{Exp1}$ except that the challenge $(K_b^*, c_0^*)$ is generated in the beginning of the experiment. Let $\mathsf{Succ2}$ be the probability that $\mathcal{A}$ succeeds to guess the bit $b$ in $\mathsf{Exp2}$. Then, we trivially obtain that $|\mathsf{Succ2} - \mathsf{Succ1}| = 0$ because the challenge is determined independently from the behavior of $\mathcal{A}$.

Let $\mathsf{Exp3}$ be the same experiment as $\mathsf{Exp2}$ except that $\mathcal{EO}$ returns randomly chosen value $z \in \{0,1\}^k$ and adds $(x, K_b^*, z)$ to the $\mathcal{EO}$ list $\mathcal{L}_{\mathcal{EO}}$ when $\mathcal{A}$ poses query $(x, K_b^*)$ for some $x$ to $\mathcal{EO}$. Then, we consider a distinguisher $\mathcal{D}$ which tries to distinguish $\mathsf{Exp2}$ from $\mathsf{Exp3}$.

**Lemma 4.** *If the output of $\mathcal{RO}$ $H$ is independently chosen from the input, $\mathcal{D}$ cannot distinguish* $\mathsf{Exp2}$ *from* $\mathsf{Exp3}$.

*Proof.* We show that we can construct an algorithm $\mathcal{ALG}$ which can distinguish an output of $\mathcal{RO}$ $H$ from a random value if there exists $\mathcal{D}$ which can distinguish $\mathsf{Exp2}$ from $\mathsf{Exp3}$. The concrete construction of $\mathcal{ALG}$ is as follows.

**Step 1 :** Simulate $\mathsf{Exp3}$ for $\mathcal{D}$, as the adversary, except when $\mathcal{D}$ poses query $(x, K_b^*)$ for some $x$ to $\mathcal{EO}$.

**Step 2 :** On receiving query $(x, K_b^*)$ for some $x$ to $\mathcal{EO}$, forward $r^*||x$ to $\mathcal{RO}$ $H$, receive $z$ as the output where $z$ is $H(r^*||x)$ or a random value $rand$, and return $z$ to $\mathcal{D}$.

**Step 3 :** If $\mathcal{D}$ decides that he interacts with $\mathsf{Exp2}$, decide that $z$ is $H(r^*||x)$. Otherwise, decide that $z$ is $rand$.

The interface of $\mathcal{D}$ is identical with $\mathsf{Exp2}$ when $z$ is $H(r^*||x)$. Also, the interface of $\mathcal{D}$ is identical with $\mathsf{Exp3}$ when $z$ is $rand$. Therefore, if $\mathcal{D}$ succeeds, then $\mathcal{ALG}$ also succeeds.

□

Thus, $\mathsf{Exp2}$ and $\mathsf{Exp3}$ is indistinguishable for the adversary $\mathcal{A}$. Then, we obtain that $|\mathsf{Succ3} - \mathsf{Succ2}| = 0$.

Let $\mathsf{Exp4}$ be the same experiment as $\mathsf{Exp3}$ except that $\mathcal{A}$ does not pose both the hash query $r^*||m$ to $\mathcal{RO}$ and the inverse attack query $(m, H(r^*||m) \oplus K_b^*)$ to $\mathcal{IO}$ for some $m$ before posing the hash query $r^*$ to $\mathcal{RO}$. Let $\mathsf{Succ4}$ be the probability that $\mathcal{A}$ succeeds to guess the bit $b$ in $\mathsf{Exp4}$ and $E_4$ be the event that $\mathcal{A}$ poses both the hash query $r^*||m$ to $\mathcal{RO}$ and the inverse attack query $(m, H(r^*||m) \oplus K_b^*)$ to $\mathcal{IO}$ for some $m$ before posing the hash query $r^*$ to $\mathcal{RO}$. Also, let $\mathsf{AskH}$ be the event that $\mathcal{A}$ poses the hash query $r^*||m$ to $\mathcal{RO}$ before posing the hash query $r^*$ to $\mathcal{RO}$. Then, the probability that the event $E_4$ occurs is equal or lower than the probability that the event $\mathsf{AskH}$ occurs. And, the probability that the event $\mathsf{AskH}$ occurs is equal or lower than $q_{IH}/|\mathbb{Z}_n|$ because $\mathcal{A}$ does not pose $r^*$ to $\mathcal{RO}$ yet and so $r^*$ is unknown for $\mathcal{A}$ owing to that $H$ is $\mathcal{RO}$. Thus, we obtain that $|\mathsf{Succ4} - \mathsf{Succ3}| \leq q_{IH}/|\mathbb{Z}_n|$.

Let $\mathsf{Exp5}$ be the same experiment as $\mathsf{Exp4}$ except when $\mathcal{A}$ queried $r^*$ to $\mathcal{RO}$. $\mathsf{Exp5}$ aborts in the above case. Let $\mathsf{Succ5}$ be the probability that $\mathcal{A}$ succeeds to guess the bit $b$ in $\mathsf{Exp5}$ and $E_5$ be the event that the experiment aborts by this case. Then, to evaluate the probability that the event $E_5$ occurs, $\Pr[E_5]$, we show that $\Pr[E_5]$ is equal or lower than the probability that RSA problem is broken as follows.

**Lemma 5.** *If the event $E_5$ occurs with the probability $\epsilon''$ in time $t''$, we can construct an inverter $\mathcal{I}$ that breaks RSA problem with the probability $\epsilon'$ in time $t'$ as follows:*

$$t' = t'' + (q_{RH} + q_{EH}) \cdot expo,$$
$$\epsilon' = \epsilon''.$$

*Proof.* We assume that $\mathcal{A}$ does not repeat previous hash queries to the $\mathcal{ERO}$ $H$ or previous decryption queries to the $\mathcal{DO}$. Let $\mathcal{L}_H$ be the local hash list of $H$. $\mathcal{L}_H$ consists of tuples $(\delta_i, y_i, h_i)$ $(0 \leq i \leq q_{RH} + q_D + q_{EH})$. Let $\mathcal{L}_{\mathcal{EO}}$ be the local $\mathcal{EO}$ list of $H$. $\mathcal{L}_{\mathcal{EO}}$ consists of tuples $(\theta_i, h_i, z_i)$ $(0 \leq i \leq q_{EH})$. Let $\mathcal{L}_{\mathcal{IO}}$ be the local $\mathcal{IO}$ list of $H$. $\mathcal{L}_{\mathcal{IO}}$ consists of tuples $(\gamma_i, h_i, a_i)$ $(0 \leq i \leq q_{IH})$. The concrete construction of $\mathcal{I}$ is as follows.

**Input :** $(n, e, y^*)$ s.t. $n$ is RSA modulus, $e$ is the exponent where $\gcd(e, \phi(n)) = 1$ and $y^* \overset{R}{\leftarrow} \mathbb{Z}_n$

**Output :** $x^*$ s.t. $x^* \equiv {y^*}^d \pmod n$

**Input public key :** Send $(n, e)$ to $\mathcal{A}$ in $\mathsf{Exp5}$ as the input public key.

**$\mathcal{DO}$ simulation :** When $\mathcal{A}$ poses a decryption query $y_i$ to $\mathcal{DO}$, then behave as follows:
Find $(\delta_i, y_i, h_i)$ from $\mathcal{L}_H$ such that $y_i = \delta_i^e$. If there is a tuple $(\delta_i, y_i, h_i)$ satisfying the condition, then return $h_i$ as the answer. Otherwise, generate $h_i \in \{0,1\}^k$, add $(\emptyset, y_i, h_i)$ to $\mathcal{L}_H$ and return $h_i$ as the answer.

**$\mathcal{RO}$ simulation :** When $\mathcal{A}$ poses a query $\delta_i$ to $\mathcal{RO}$, then behave as follows:

```
<If $y_i = y^*$ s.t. $y_i = \delta_i^e \mod n$>
    Output $\delta_i$ as $x^*$ and halt.
<If $(\delta_i, *, h_i) \in \mathcal{L}_H$>
    Return $h_i$ to $\mathcal{A}$ as the answer.
<If $(\delta_i, *, *) \notin \mathcal{L}_H$ and $(\emptyset, y_i, h_i) \in \mathcal{L}_H$ s.t. $y_i = \delta_i^e \mod n$>
    Replace $(\emptyset, y_i, h_i)$ to $(\delta_i, y_i, h_i)$ in $\mathcal{L}_H$ and return $h_i$ to $\mathcal{A}$ as the answer.
<If $(\delta_i, *, *) \notin \mathcal{L}_H$ and $(\emptyset, y, h) \notin \mathcal{L}_H$ s.t. $y = \delta_i^e \mod n$>
    Compute $y_i = \delta_i^e \mod n$, generate $h_i \in \{0,1\}^k$, add $(\delta_i, y_i, h_i)$ and return $h_i$ to $\mathcal{A}$ as the answer.
```

**$\mathcal{EO}$ simulation :** When $\mathcal{A}$ poses an extension attack query $(\theta_i, h_i)$ to $\mathcal{EO}$, then behave as follows:
Find $(\theta_i, h_i, *)$ from $\mathcal{L}_{\mathcal{EO}}$. If there is a tuple $(\theta_i, h_i, z_i)$ satisfying the condition, then return $z_i$. Else if there is only one tuple $(\delta', *, h_i)$ in $\mathcal{L}_H$, then generate $z_i \in \{0,1\}^k$, compute $y_i = (\delta' || \theta_i)^e \mod n$, add $(\delta' || \theta_i, y_i, z_i)$ to $\mathcal{L}_H$ and $(\theta_i, h_i, z_i)$ to $\mathcal{L}_{\mathcal{EO}}$, and return $z_i$. Otherwise, generate $z_i \in \{0,1\}^k$ add $(\theta_i, h_i, z_i)$ to $\mathcal{L}_{\mathcal{EO}}$, and return $z_i$.

**$\mathcal{IO}$ simulation :** When $\mathcal{A}$ poses an inverse attack query $(\gamma_i, a_i)$ to $\mathcal{IO}$, then behave as follows:
Find $(\gamma_i, *, a_i)$ from $\mathcal{L}_{\mathcal{IO}}$. If there is a tuple $(\gamma_i, h_i, a_i)$ satisfying the condition, then return $h_i$. Else if there are tuples $(\delta', *, h_i)$ and $(\delta' || \gamma_i, *, h_i \oplus a_i)$ in $\mathcal{L}_H$, then add $(\gamma_i, h_i, a_i)$ to $\mathcal{L}_{\mathcal{IO}}$, and return $h_i$. Otherwise, generate $h_i \in \{0,1\}^k$ add $(\gamma_i, h_i, a_i)$ to $\mathcal{L}_{\mathcal{IO}}$, and return $h_i$.

**Challenge ciphertext :** When $\mathcal{A}$ outputs $(state)$, then compute $(K^*, y')$ by the encryption procedure and return $(K^*, y^*)$ as the challenge.

We determine the success probability of $\mathcal{I}$. In $\mathcal{RO}$ simulation, if $y_i = y^*$ such that $y_i = \delta_i^e \mod n$ holds, $\mathcal{I}$ successfully breaks RSA problem. This event is same as $E_5$ in $\mathsf{Exp5}$. Also, it is clear that $\mathcal{I}$ perfectly simulates $\mathsf{Exp5}$ for $\mathcal{A}$. Therefore, we obtain

$$\epsilon' = \epsilon''.$$

$\mathcal{I}$ computes at most $q_{RH} + q_{EH}$ exponentiations modulo $n$. Thus, we obtain

$$t' = t'' + (q_{RH} + q_{EH}) \cdot expo.$$

□

$\mathsf{Exp4}$ and $\mathsf{Exp5}$ are identical until $E_5$ occurs. Thus, $|\mathsf{Succ5} - \mathsf{Succ4}| = \epsilon'$.

$\mathcal{A}$ can obtain no information about the random bit $b$ because the key $K_b^*$ is independent from information which $\mathcal{A}$ can obtain in $\mathsf{Exp5}$. Therefore, $\mathsf{Succ5} = 1/2$. Since $\mathsf{Succ0} \leq |\mathsf{Succ1} - \mathsf{Succ0}| + |\mathsf{Succ2} - \mathsf{Succ1}|$

$+|\mathsf{Succ3} - \mathsf{Succ2}| + |\mathsf{Succ4} - \mathsf{Succ3}| + |\mathsf{Succ5} - \mathsf{Succ4}| + \mathsf{Succ5}$, $\mathsf{Succ0} \leq \epsilon' + \frac{q_D}{n} + \frac{q_{IH}}{|\mathbb{Z}_n|} + 1/2$. Hence, $\epsilon' \geq \epsilon - \frac{q_D}{n} - \frac{q_{IH}}{|\mathbb{Z}_n|}$.

□

## D OAEP encryption [3]

OAEP encryption scheme is based on trapdoor partial-domain one-way permutations.

**Definition 7 (Trapdoor partial-domain one-way permutation).** *Let $\mathcal{G}$ be a trapdoor permutation generator. We say that a trapdoor permutation $f$ is $(t, \epsilon)$-partial-domain one-way if*

- *for input $1^k$, $\mathcal{G}$ outputs $(f, f^{-1}, Dom)$ where $Dom$ is a subset of $\{0,1\}^{k_0} \times \{0,1\}^{k_1}$ $(k_0 + k_1 < k)$ and $f, f^{-1}$ are permutations on $Dom$ which are inverses of each other,*
- *there exist a polynomial $p$ such that $f, f^{-1}$ and $Dom$ are computable in time $p(k)$, and*
- *for any adversary $Alg$, $\Pr[(f, f^{-1}, Dom) \leftarrow \mathcal{G}(1^k);\ (x_0, x_1) \overset{R}{\leftarrow} Dom;\ Alg(f,\ Dom,\ f(x_0, x_1)) = x_0] \leq \epsilon$, where $Alg$ runs in at most $t$ steps.*

The description of OAEP encryption scheme is as follows:

**Key generation :** For input $k$, outputs encryption key $(ek = f)$ and decryption key $(dk = f^{-1})$ such that $(f, f^{-1}, Dom = \{0,1\}^{n+k_1} \times \{0,1\}^{k_0}) \leftarrow \mathcal{G}(1^k)$ where $\mathcal{G}$ is a trapdoor permutation generator and $n = k - k_0 - k_1$.

**Encryption :** Upon input of message $m \in \{0,1\}^n$, generates randomness $r \overset{R}{\leftarrow} \{0,1\}^{k_0}$, computes $x = (m||0^{k_1}) \oplus G(r)$ and $y = r \oplus H(x)$, and outputs ciphertext $c = f(x, y)$ where " || " means concatenation, $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$ and $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ are hash functions.

**Decryption :** Upon inputs of ciphertext $c$, computes $z = f^{-1}(c)$, parses $z$ as $(x, y)$ and reconstructs $r = y \oplus H(x)$ where $|x| = n + k_1$ and $|y| = k_0$. If $[x \oplus G(r)]_{k_1} \overset{?}{=} 0^{k_1}$ holds, outputs $m = [x \oplus G(r)]^n$ as the plaintext corresponding to $c$ where $[a]^b$ denotes the $b$ least significant bits of $a$ and $[a]_b$ denotes the $b$ most significant bits of $a$. Otherwise, rejects the input as an invalid ciphertext.