How to Confirm Cryptosystems Security: The Original Merkle-Damgård Is Still Alive!

Yusuke Naito[1], Kazuki Yoneyama[2], Lei Wang[3], and Kazuo Ohta[3]

[1] Mitsubishi Electric Corporation
[2] NTT Corporation
[3] The University of Electro-Communications

Abstract. At Crypto 2005, Coron et al. showed that Merkle-Damgård hash function (MDHF) with a fixed input length random oracle is not indifferentiable from a random oracle RO due to the extension attack. Namely MDHF does not behave like RO. This result implies that there exists some cryptosystem secure in the RO model but insecure under MDHF. However, this does not imply that no cryptosystem is secure under MDHF. This fact motivates us to establish a criteria methodology for confirming cryptosystems security under MDHF.

In this paper, we confirm cryptosystems security by using the following approach:

1. Find a variant, $\widetilde{\mathsf{RO}}$, of RO which leaks the information needed to realize the extension attack.
2. Prove that MDHF is indifferentiable from $\widetilde{\mathsf{RO}}$.
3. Prove cryptosystems security in the $\widetilde{\mathsf{RO}}$ model.

From the indifferentiability framework, a cryptosystem secure in the $\widetilde{\mathsf{RO}}$ model is also secure under MDHF. Thus we concentrate on finding $\widetilde{\mathsf{RO}}$, which is weaker than RO.

We propose the Traceable Random Oracle (TRO) which leaks enough information to permit the extension attack. By using TRO, we can *easily* confirm the security of OAEP and variants of OAEP. However, there are several practical cryptosystems whose security cannot be confirmed by TRO (e.g. RSA-KEM). This is because TRO leaks information that is irrelevant to the extension attack. Therefore, we propose another $\widetilde{\mathsf{RO}}$, the Extension Attack Simulatable Random Oracle, ERO, that leaks *just* the information needed for the extension attack. Fortunately, ERO is *necessary and sufficient* to confirm the security of cryptosystems under MDHF. This means that the security of *any* cryptosystem under MDHF is *equivalent* to that under the ERO model. We prove that RSA-KEM is secure in the ERO model.

Keywords: Indifferentiability, Merkle-Damgård hash function, Variants of Random Oracle, Cryptosystems Security.

1 Introduction

Indifferentiability Framework. Maurer et al. [9] introduced the indifferentiable framework as a notion stronger than indistinguishability. This framework

M. Matsui (Ed.): ASIACRYPT 2009, LNCS 5912, pp. 382–398, 2009.

deals with the security of two systems $\mathcal{C}(\mathcal{V})$ and $\mathcal{C}(\mathcal{U})$: for cryptosystem $\mathcal{C}$, $\mathcal{C}(\mathcal{V})$ retains at least the same level of provable security of $\mathcal{C}(\mathcal{U})$ if primitive $\mathcal{V}$ is indifferentiable from primitive $\mathcal{U}$, denoted by $\mathcal{V} \sqsubset \mathcal{U}$. This definition will allow us to use construction $\mathcal{V}$ instead of $\mathcal{U}$ in any cryptosystem $\mathcal{C}$ and retain the same level of provable security due to the indifferentiability framework of Maurer et al. [9]. We denote "$\mathcal{C}(\mathcal{V})$ is at least as secure as $\mathcal{C}(\mathcal{U})$" by $\mathcal{C}(\mathcal{V}) \succ \mathcal{C}(\mathcal{U})$. More strictly, $\mathcal{V} \sqsubset \mathcal{U} \Leftrightarrow \mathcal{C}(\mathcal{V}) \succ \mathcal{C}(\mathcal{U})$ holds. This result implies that if cryptosystem $\mathcal{C}$ is secure in the $\mathcal{U}$ model and $\mathcal{V} \sqsubset \mathcal{U}$ holds, $\mathcal{C}$ is secure in the $\mathcal{V}$ model, and if $\mathcal{U} \not\sqsubset \mathcal{V}$ holds, there is some cryptosystem that is secure in the $\mathcal{U}$ model but insecure in the $\mathcal{V}$ model.

Indifferentiability and the MD Construction. While many cryptosystems have been proven to be secure in the random oracle (RO) model [3] (e.g. FDH [3], OAEP[4], RSA-KEM[11], Prefix-MAC[12] and so on), where RO is modeled as a monolithic entity (i.e. a black box working in domain $\{0,1\}^*$), in practice most instantiations that use a hash function are usually constructed by iterating a fixed input length primitive (e.g. a compression function). There are many architectures based on iterated hash functions. The most well-known one is the Merkle-Damgård (MD) construction [6,10]. A hash function with MD construction iterates underlying compression function $f : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^n$ as follows.

$\mathsf{MD}^f(m_1, ..., m_l)$ $(|m_i| = t, i = 1, ..., l)$:
  let $y_0 = IV$ be some n bit fixed value.
  for $i = 1$ to l do $y_i = f(y_{i-1}, m_i)$
  return y_l

There is a significant gap between RO and hash functions, since hash functions are constructed from a small primitive f while RO is a monolithic random function.

Coron et al. [5] made important observations on the cryptosystems that use the indifferentiable framework. They introduced the new iterated hash function property of indifferentiability from RO. In this framework, the underlying primitive, G, is a fixed input length random oracle (denoted here as FILRO or h) or an ideal block cipher. We say that hash function H^G is indifferentiable from RO if there exists simulator S such that no distinguisher can distinguish H^G from RO (S mimics G). The distinguisher can access RO/H^G and S/G; S can access RO. A hash function that satisfies this property, H^G, behaves like RO. Therefore, replacing the RO of any cryptosystem by H^G does not destroy its security.

Coron et al. analyzed the indifferentiability from RO for several specific constructions. For example, they have shown that MD^h is not indifferentiable from RO due to the extension attack which uses the following property: The output value $z' = \mathsf{MD}^h(M||m)$ can be calculated by $c = h(z, m)$ where $z = \mathsf{MD}^h(M)$, so $z' = c$. On the other hand, no S can return the output value $z' = \mathsf{RO}(M||m)$ from query (z, m) where $z = \mathsf{RO}(M)$, since no S knows z' from z and m, and z' is chosen at random. Therefore, no S can simulate the extension attack. This result implies that MD^h does not behave like RO and there exists some cryptosystem

that is secure in the RO model but insecure under MD^h due to the indifferentiability framework. Their solution was to propose several constructions such as Prefix-Free MD, chop MD, NMAC and HMAC. Hash functions with these constructions are, under h, indifferentiable from RO. It seems impossible to prove that the important *original* MD cryptosystem is secure.

MD Construction Dead? The MD construction is among the most important foundations of modern cryptosystems [2,5,8]. There are two main reasons:

1. MD construction is employed by many popular hash functions such as SHA-1 and SHA-256, and
2. MD construction is more efficient than other iterated hash functions such as Prefix-Free MD, and chop MD.

Since $\mathsf{MD}^h \not\sqsubset \mathsf{RO}$ holds, there is some cryptosystem $\mathcal{C}^*$ that is secure in the RO model but insecure under MD^h. Thus the important question is "can we confirm that a given cryptosystem is secure in the RO model and secure under MD^h?" There might be several cryptosystems that remain secure when RO is replaced by MD^h. If we can confirm this for many cryptosystems that are widely used, the *original* MD construction remains alive in the indifferentiability framework!

Our Contribution. Since $\mathsf{MD}^h \not\sqsubset \mathsf{RO}$ holds, we modify RO such that MD^h is indifferentiable from the modified RO. Then we analyze cryptosystems security within the modified RO model. Concretely, we adopt the following approach.

1. Find a variant $\widetilde{\mathsf{RO}}$ of RO that leaks enough information such that S can simulate the extension attack.
2. Prove that $\mathsf{MD}^h \sqsubset \widetilde{\mathsf{RO}}$ holds.
3. Prove the cryptosystem's security in the $\widetilde{\mathsf{RO}}$ model.

Secure cryptosystems in the $\widetilde{\mathsf{RO}}$ model are also secure under MD^h due to the indifferentiability framework. Therefore, we concentrate on proposing $\widetilde{\mathsf{RO}}$ that can support many applications.

First we propose *Traceable Random Oracle* TRO as $\widetilde{\mathsf{RO}}$.

Traceable Random Oracle. Our proposal of TRO is motivated by the following points:

- Applications of TRO hide the outputs of hash functions from adversaries. One example is OAEP encryption: Adversaries cannot know the outputs of the hash functions that are used for calculating a cipher text, since these values are hidden by a random value or a trapdoor one-way permutation.
- TRO leaks useful information such that S can run the extension attack.

By considering the above points, it is convenient for S to obtain useful information from value z which is the output of $\mathsf{RO}(M)$. Thus we define TRO that leaks input M on query z such that $\mathsf{RO}(M) = z$. Since S can obtain value M such that $z = \mathsf{RO}(M)$, S can know value $z' = \mathsf{RO}(M||m)$ by using TRO. Therefore, S can run the extension attack. We will prove that $\mathsf{MD}^h \sqsubset \mathsf{TRO}$ holds (Corollary 2).

Since the hash function outputs for OAEP and variants of OAEP (e.g. OAEP+) are hidden, adversaries cannot use TRO effectively. So we can easily confirm that these cryptosystems are secure in the TRO model.

Limitation of TRO. Though TRO can easily confirm the security of many cryptosystems under MD^h, there are several cryptosystems whose security we cannot confirm by TRO. For example, RSA-KEM is insecure in the TRO model (Theorem 7). It is possible that there are cryptosystems that are secure under MD^h because TRO leaks information beyond that needed to simulate the extension attack. The essential information to simulate the extension attack is just $z' = \mathsf{RO}(M||m)$, but TRO leaks M, which is not essential.

Our response is to propose *Extension Attack Simulatable Random Oracle* ERO as $\widetilde{\mathsf{RO}}$.

Extension Attack Simulatable Random Oracle. We define ERO that leaks just z' $(= \mathsf{RO}(M||m))$. By using ERO, S can run the extension attack, since S can know z'. We will prove that $\mathsf{MD}^h \sqsubset \mathsf{ERO}$ holds (Theorem 5). We will also prove that RSA-KEM is secure in the ERO model (Theorem 8). Therefore, we can confirm RSA-KEM security under MD^h by using ERO. Fortunately, MD^h is equivalent to ERO, since $\mathsf{ERO} \sqsubset \mathsf{MD}^h$ holds (Theorem 6). Namely, any cryptosystem that is secure under MD^h is equally secure in the ERO model and vice versa. Therefore, ERO is necessary and sufficient to confirm the security of cryptosystems under MD^h. When we analyze a cryptosystem under MD^h, all that is needed is to prove cryptosystems security in the ERO model.

TRO v.s. ERO. Since TRO leaks more information than ERO, we will prove $\mathsf{ERO} \sqsubset \mathsf{TRO}$. Since ERO has wider applicability, we recommend that ERO be used for cryptosystems whose security cannot be proven in the TRO model.

ERO v.s. RO. Since ERO leaks several bits of information in permitting the simulation of the extension attack, $\mathsf{RO} \sqsubset \mathsf{ERO}$ and $\mathsf{ERO} \not\sqsubset \mathsf{RO}$ explicitly hold. As evidence of the separation between RO and ERO, we pick up prefix MAC [12] which is secure in the RO model, and prove that prefix MAC is insecure in the ERO model (Theorem 4). Since ERO is equivalent to MD^h, prefix MAC is also insecure in the MD^h model.

Leakey Random Oracle. Leaky random oracle LRO was proposed by Yoneyama et al. [13] but with a different motivation. LRO has a function that leaks all query-response pairs of RO. In this paper, we will prove that $\mathsf{TRO} \sqsubset \mathsf{LRO}$ and $\mathsf{LRO} \not\sqsubset \mathsf{TRO}$ hold. Therefore, all cryptosystems secure in the LRO model are also secure in the TRO model and there is some cryptosystem that is insecure in the LRO model but secure in the TRO model. Since FDH is secure in LRO model [13], FDH is secure under MD^h. Since OAEP is insecure in the LRO model [13] and secure in the TRO model, OAEP is evidence of the separation between LRO and TRO.

Remarks. First we compare LRO, TRO and ERO from the viewpoint of security proofs of cryptosystems. LRO, TRO, and ERO consist of RO and the additional

oracle (denote LO, TO and EO respectively). Since LO leaks more information to adversaries than TO, adversaries that are given LRO have more flexible strategies than adversaries given TRO. That is, security proofs in the LRO model are more complex than those in the TRO model. The same is true for TRO and ERO.

Finally, for the security proof of cryptosystem $\mathcal{C}(\mathsf{MD}^h)$ we compare the direct proof in MD^h with the proof via ERO. Since MD^h has the MD structure, we must consider this structure in the direct proof. On the other hand, since ERO does not have this structure, we does not need to consider it. For example we must consider the events of inner collisions for MD^h in the direct proof. However this is not necessary for the proof in the ERO model. Moreover, since we can reuse existing proofs for the simulation of RO in the security proof in the ERO model, we only consider the simulation of EO in the security proof. Therefore, the security proof in the ERO model is easier than the direct proof in MD^h. Since $\mathsf{ERO} = \mathsf{MD}^h$ holds, we can confirm a cryptosystems security under MD^h by proving its security in ERO, an easier task than a direct proof.

Related Works. Recently, Dodis et al. independently proposed a methodology to salvage the original and modified MD constructions in many applications [7]. They found two properties: one is preimage awareness (PrA), and the other is public-use random oracle (pub-RO). pub-RO is the same as LRO. The approach of pub-RO is almost same as our approach of LRO. Dodis et al. pointed out that the security of cryptosystems that satisfy the following property can be easily proven in the pub-RO model: all inputs of hash functions are public to the adversaries. Therefore, PSS and the Fiat-Shamir signature scheme, and other, are easily proven to be secure in the pub-RO model by using existing proofs in the RO model. Since $\mathsf{LRO}(\text{pub-RO}) \not\sqsubset \mathsf{TRO}$ and $\mathsf{TRO} \sqsubset \mathsf{LRO}(\text{pub-RO})$ hold, TRO and ERO have more applications than LRO(pub-RO) (e.g. OAEP is secure in the TRO model but insecure in the pub-RO model). The approach of PrA is interesting in that this approach can treat the case where the compression function f requirement is relaxed from FILRO to property PrA. It seems, however, that this approach is not effective in saving the original MD construction, since this approach *modifies* MD construction by processing the output of the MD construction by FILRO.

Cryptosystems Security under the Merkle-Damgård Hash Function. PSS, Fiat-Shamir, and so on are secure under MD^h thanks to pub-RO [7], OAEP and variants of OAEP are secure under MD^h thanks to TRO, and RSA-KEM is secure under MD^h thanks to ERO. Since many cryptosystems are secure under MD^h, the original Merkle-Damgård construction is still alive!

2 Preliminaries

2.1 Merkle-Damgård Construction

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

– $\mathsf{MD}^f(M)$:
  1. calculate $M' = pad(M)$ where pad is a padding function such that $pad : \{0,1\}^* \to (\{0,1\}^t)^*$.
  2. calculate $c_i = f(c_{i-1}, m_i)$ 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_n

In this paper we ignore the above padding function, this does not degrade generality, so hereafter we discuss $\mathsf{MD}^f : (\{0,1\}^t)^* \to \{0,1\}^n$. We use random oracle compression function h as f where $h : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^n$. Thus we discuss below hash function MD^h with MD construction using h.

2.2 Random Oracle

$\mathsf{RO} : \{0,1\}^* \to \{0,1\}^n$ can be realized as follows. RO has initially the empty hash list $\mathcal{L}_{\mathsf{RO}}$. On query M, if $\exists(M, z) \in \mathcal{L}_{\mathsf{RO}}$, it returns z. Otherwise, it chooses $z \in \{0,1\}^n$ at random, adds (M, z) to the $\mathcal{L}_{\mathsf{RO}}$, hereafter denoted by $\mathcal{L}_{\mathsf{RO}} \leftarrow (M, z)$, and returns z.

2.3 Leaky Random Oracle

LRO was proposed by Yoneyama et al. [13]. LRO can be realized as follows. LRO consists of RO and LO. On a leak query to LO, LO outputs the entire contents of $\mathcal{L}_{\mathsf{RO}}$. We can define S that can simulate the extension attack by using LRO, since S can know M from z by using LO and can know z' by posing $M||m$ to RO.

2.4 Indifferentiability

The indifferentiability framework generalizes the fundamental concept of the indistinguishability of two cryptosystems $\mathcal{C}(\mathcal{U})$ 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 $\mathcal{W}^{\mathsf{priv}}$ and the public interface of the system $\mathcal{W}$ by $\mathcal{W}^{\mathsf{pub}}$. 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 polynomial time simulator S such that $|Pr[D^{\mathcal{V}^{\mathsf{priv}}, \mathcal{V}^{\mathsf{pub}}} \Rightarrow 1] - Pr[D^{\mathcal{U}^{\mathsf{priv}}, S(\mathcal{U}^{\mathsf{pub}})} \Rightarrow 1]| < \epsilon$. Simulator S has oracle access to $\mathcal{U}^{\mathsf{pub}}$ and runs in time at most t_{S}. Distinguisher D runs in time at most t_D and makes at most q queries. ϵ is negligible in security parameter k.*

This definition will allow us to use construction $\mathcal{V}$ instead of $\mathcal{U}$ in *any* cryptosystem $\mathcal{C}$ and retain the same level of provable security due to the indifferentiability theory of Maurer et al. [9]. We denote "$\mathcal{C}(\mathcal{V})$ is at least as secure as $\mathcal{C}(\mathcal{U})$" by $\mathcal{C}(\mathcal{V}) \succ \mathcal{C}(\mathcal{U})$. Namely, $\mathcal{C}(\mathcal{V}) \succ \mathcal{C}(\mathcal{U})$ denotes the case that if $\mathcal{C}(\mathcal{U})$ is secure, then $\mathcal{C}(\mathcal{V})$ is secure. More strictly, $\mathcal{V} \sqsubset \mathcal{U} \Leftrightarrow \mathcal{C}(\mathcal{V}) \succ \mathcal{C}(\mathcal{U})$ holds.

2.5 Extension Attack

Coron et al. showed that MD^h is not indifferentiable from RO due to the extension attack. The extension attack targets MD^h where we can calculate a new hash value from some hash value. Namely $z' = \mathsf{MD}^h(M||m)$ can be calculated from only z and m by $z' = h(z, m)$ where $z = \mathsf{MD}^h(M)$. Note that z' can be calculated without using M. The differentiable attack with extension attack is as follows. Let $\mathcal{O}_a$ be MD^h or RO and let $\mathcal{O}_b$ be h or S. First, a distinguisher poses M to $\mathcal{O}_a$ and gets z from $\mathcal{O}_a$. Second, he poses (z, m) to $\mathcal{O}_b$ and gets c from $\mathcal{O}_b$. Finally, he poses $M||m$ to $\mathcal{O}_a$ and gets z' from $\mathcal{O}_a$.

If $\mathcal{O}_a = \mathsf{MD}^h$ and $\mathcal{O}_b = h$, then $z' = c$, while, if $\mathcal{O}_a = \mathsf{RO}$ and $\mathcal{O}_b = S$, then $z' \neq c$. This is because no simulator can obtain the output value of $\mathsf{RO}(M||m)$ from just (z, m) and the output value of $\mathsf{RO}(M||m)$ is independently and randomly defined from c. Therefore, $\mathsf{MD}^h \not\preceq \mathsf{RO}$ holds.

3 Variants of Random Oracles

In this section, we will introduce several variants of random oracles in order for S to simulate the extension attack described above, and then show the relationships among these oracles within the indifferentiability framework.

3.1 Definition of Variants of Random Oracles

Traceable Random Oracle: TRO consists of RO and TO. On trace query z,

1. If there exist pairs such that $(M_i, z) \in \mathcal{L}_{\mathsf{RO}}$ $(i = 1, ..., n)$, it returns $(M_1, ..., M_n)$.
2. Otherwise, it returns $\perp$.

We can define S that can simulate the extension attack by using TRO, since S can know M from z by using TO and can know z' by posing $M||m$ to RO.

Extension Attack Simulatable Random Oracle: TRO leaks too much information to simulate the extension attack. So we define ERO such that S is given just the important information. The important information is value z' such that $z' = \mathsf{RO}(M||m)$. Therefore, we define ERO as follows. ERO consists of RO and EO. EO has initially the empty list $\mathcal{L}_{\mathsf{EO}}$ and can look into $\mathcal{L}_{\mathsf{RO}}$. On simulation query (m, z) to EO where $|m| = t$,

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

We can construct S that can simulate the extension attack by using ERO, since S can obtain z' from (m, z) where $z' = \mathsf{RO}(M||m)$ by using EO.

3.2 Relationships among LRO, TRO, ERO, and RO Models within the Indifferentiability Framework

LRO leaks more information of $\mathcal{L}_{\mathsf{RO}}$ than TRO, and TRO leaks more information of $\mathcal{L}_{\mathsf{RO}}$ than ERO. Therefore, it seems reasonable to suppose that anything secure in the LRO model is also secure in the TRO model, anything secure in the TRO model is also secure in the ERO model, and any cryptosystem secure in the ERO model is also secure in the RO model. We prove the validity of these suppositions by using the indifferentiability framework.

First we clarify the relationship between TRO and LRO.

Theorem 1. $\mathsf{TRO} \sqsubset \mathsf{LRO}$ *and* $\mathsf{LRO} \not\sqsubset \mathsf{TRO}$.

Proof. We construct S which simulates TO by using LRO as follows. Given query z, S poses a leak query to LO and receives the entire information of $\mathcal{L}_{\mathsf{RO}}$. If there exists pairs such that $(M_i, z) \in \mathcal{L}_{\mathsf{RO}}$ $(i = 1, ..., n)$, it returns $(M_1, ..., M_n)$. Otherwise it returns $\perp$.

It is easy to see that $|Pr[D^{\mathsf{RO},\mathsf{TO}} \Rightarrow 1] - Pr[D^{\mathsf{RO},\mathsf{S}(\mathsf{LRO})} \Rightarrow 1]| = 0$, since the output from each step of S is equal to that from each step of TO.

LRO $\not\sqsubset$ TRO is trivial, since no S cannot acquire all values in $\mathcal{L}_{\mathsf{RO}}$ by using TRO. □

Since TRO $\sqsubset$ LRO, any cryptosystem secure in the LRO model is also secure in the TRO model by the indifferentiability framework. Since LRO $\not\sqsubset$ TRO, there exists some cryptosystem that is secure in the TRO model but insecure in the LRO model. For example, Yoneyama et al. proved that OAEP is insecure in the LRO model [13]. Since OAEP is secure in the TRO model, OAEP is evidence of the separation between LRO and TRO.

Next we will clarify the relationship between ERO and TRO.

Theorem 2. $\mathsf{ERO} \sqsubset \mathsf{TRO}$ *and* $\mathsf{TRO} \not\sqsubset \mathsf{ERO}$.

Proof. We construct S which simulates EO by using TRO as follows. S initially has the empty list $\mathcal{L}_S$. On query (m, z), if $\exists(m, z, z') \in \mathcal{L}_S$, it returns z'. Otherwise S poses query z to TO, and receives string X. If X consists of one value, it poses query $X||m$ to RO, receives z', $\mathcal{L}_S \leftarrow (m, z, z')$ and returns z'. Otherwise, it chooses $z' \in \{0,1\}^n$ at random, $\mathcal{L}_S \leftarrow (m, z, z')$ and returns z'.

It is easy to see that $|Pr[D^{\mathsf{RO},\mathsf{EO}} \Rightarrow 1] - Pr[D^{\mathsf{RO},\mathsf{S}(\mathsf{TRO})} \Rightarrow 1]| = 0$, since the output from each step of S is equal to that from each step of EO.

TRO $\not\sqsubset$ ERO is trivial, since no S cannot decide whether there exists (M, z) in $\mathcal{L}_{\mathsf{RO}}$ or not by using ERO. □

Since ERO $\sqsubset$ TRO, any cryptosystem secure in the TRO model is also secure in the ERO model in the indifferentiability framework. Since TRO $\not\sqsubset$ ERO, there exists some cryptosystem that is secure in the ERO model but insecure in the TRO model. We will prove that RSA-KEM is secure in the ERO model but insecure in the TRO model in Section 5. Therefore, RSA-KEM is evidence of the separation between TRO and ERO.

Finally we will clarify the relationship between RO and ERO.

Theorem 3. $\mathsf{RO} \sqsubset \mathsf{ERO}$ *and* $\mathsf{ERO} \not\sqsubset \mathsf{RO}$.

This proof of theorem 3 is trivial because ERO consists of RO and the additional oracle EO which leaks some information of $\mathcal{L}_{\mathsf{RO}}$. Since $\mathsf{RO} \sqsubset \mathsf{ERO}$, any cryptosystem secure in the ERO model is also secure in the RO model by the indifferentiability framework. Since $\mathsf{ERO} \not\sqsubset \mathsf{RO}$, there exists some cryptosystem which is secure in the RO model but insecure in the ERO model. We can show simple evidence of the separation between ERO and RO as follows: We consider the following Prefix-MAC protocol which is unforgeable in the RO model. Note that the concept of unforgeability with regard to MAC schemes is defined in [1].

Prefix MAC [12]: Alice and Bob share one secret key, K, as an authentication key. Before sending message M to Bob, Alice sends $K||M$ to RO H to obtain a MAC value denoted as y. Finally, Alice sends (M, y) to Bob. When Bob obtains (M, y), he sends $K||M$ to H to obtain another MAC value y'. If y' is equal to y, then Bob is convinced that message M is from Alice. Otherwise, Bob will reject message M.

We will show that Prefix MAC fails to satisfy unforgeability for MAC schemes in the ERO model.

Theorem 4 (Insecurity of Prefix MAC in the ERO model). *Prefix MAC does not satisfy unforgeability for MAC schemes where H is modeled as* ERO.

Proof. A forgery procedure is as follows: forger $\mathcal{F}$ obtains a valid pair of (M, h) from MAC, where $h = H(K||M)$. $\mathcal{F}$ sends (h, m) to EO, and obtains $h' = H(K||M||m)$. Since $M||m$ is not queried to MAC, $\mathcal{F}$ succeeds in Existential forgery of known message attack (EF-KMA) attack using ERO H. □

Therefore, Prefix-MAC is secure in the RO model but insecure in the ERO model. Consequently, Prefix-MAC is evidence of the separation between ERO and RO.

From the above discussions, the following corollary is obtained.

Corollary 1. $\mathsf{RO} \sqsubset \mathsf{ERO} \sqsubset \mathsf{TRO} \sqsubset \mathsf{LRO}$, *and* $\mathsf{LRO} \not\sqsubset \mathsf{TRO} \not\sqsubset \mathsf{ERO} \not\sqsubset \mathsf{RO}$.

4 Relationship between MD^h and ERO in the Indifferentiability Framework

In this section we prove that $\mathsf{MD}^h \sqsubset \mathsf{ERO}$ and $\mathsf{ERO} \sqsubset \mathsf{MD}^h$ hold as follows. In theorem 5, we use statements σ_H and q_h instead of the total number of queries q. σ_H is the total number of message blocks for $\mathsf{RO}/\mathsf{MD}^h$ and q_h is the total number of queries to S/h

Theorem 5. $\mathsf{MD}^h \sqsubset \mathsf{ERO}$, *for any* t_D, *with* $t_S = O(q_h^2)$ *and* $\epsilon \leq \frac{4(\sigma_H+q_h)^2+2(\sigma_H+q_h)}{2^n}$.

This proof is given in subsection 4.1.

In theorem 6, we use statements σ_H and q_{EO} instead of the total number of queries q. σ_H is the total number of message blocks for $\mathsf{RO}/\mathsf{MD}^h$ and q_{EO} is the total number of queries to EO/S

Theorem 6. $\mathsf{ERO} \sqsubset \mathsf{MD}^h$, *for any* t_D, *with* $t_S = O(q_{\mathsf{EO}})$ *and* $\epsilon \leq \frac{2(\sigma_H+q_{\mathsf{EO}})^2+(\sigma_H+q_{\mathsf{EO}})}{2^n}$.

This proof is given in subsection 4.2.

From Theorem 5 and Theorem 6, ERO is equivalent to MD^h in the indifferentiability framework. From Corollary 1, Theorem 5 and Theorem 6, the following corollary is obtained.

Corollary 2. $\mathsf{RO} \sqsubset \mathsf{MD}^h = \mathsf{ERO} \sqsubset \mathsf{TRO} \sqsubset \mathsf{LRO}$, *and* $\mathsf{LRO} \not\sqsubset \mathsf{TRO} \not\sqsubset \mathsf{ERO} = \mathsf{MD}^h \not\sqsubset \mathsf{RO}$

4.1 Proof of Theorem 5

First we define simulator S as follows. S has a list $\mathcal{T}$ which is initially empty. We define chain triples as follows.

Definition 2 (Chain Triples). *Triples* $(x_1, m_1, y_1), ..., (x_i, m_i, y_i)$ *are chain triples if* $x_1 = IV$ *and* $y_j = x_{j+1}$ $(j = 1, ..., j-1)$ *holds.*

Simulator S: On a query (x, m),

1. If $\exists(x, m, y) \in \mathcal{T}$, it outputs y.
2. Else if chain triples $\exists(x_1, m_1, y_1), ..., (x_i, m_i, y_i) \in \mathcal{T}$ such that $x = y_i$, $y \leftarrow \mathsf{RO}(m_1||...||m_i||m)$.
3. Else, $y \leftarrow \mathsf{EO}(m, x)$.
4. $\mathcal{T} \leftarrow (x, m, y)$.
5. S returns y.

Since S needs to search pairs in $\mathcal{T}$, this requires at most $O(q_h^2)$ time.

We need to prove that S cannot tell apart two scenarios, ERO and MD^h. In one scenario D has oracle access to RO and S while in the other D has access to MD^h and h. The proof involves a hybrid argument starting in the ERO scenario, and ending in the MD^h scenario through a sequence of mutually indistinguishable hybrid games.

We give six events that allow D to distinguish MD^h from ERO. These events arise from the fact that MD^h has the MD construction but ERO does not. We explain these events as follows. Details of these events are given in Game 3.

First we discuss distinguishing events that occur due to differences among RO and MD^h. RO and MD^h return a random value unless collision occurs. Therefore, distinguishing events occur when collision occurs. When a collision of MD^h occurs, one of following events occurs due to the MD construction: an output of h is equal to IV (event E1) or a collision of h occurs (event E2). On the other hand, since RO is a monolithic function, these events don't occur. Therefore, these events are distinguishing events between MD^h and ERO.

Second, we discuss distinguishing events that occur due to differences among S and h. Since for h there is the relation that $h(x, m) = \mathsf{RO}(M||m)$ where $\mathsf{MD}^h(M) = x$, S must simulate the relation such that $\mathsf{S}(x, m) = \mathsf{RO}(M||m)$ where $\mathsf{RO}(M) = x$. On query (x, m) to S, if only one pair exists $(M, x) \in \mathcal{L}_{\mathsf{RO}}$

such that $x \neq IV$ holds, S can know $\mathsf{MD}^h(M||m)$ by using EO. Therefore, S can simulate the relation. If such a pair does not exist ($(M, x) \notin \mathcal{L}_{\mathsf{RO}}$), since S cannot know M, S cannot know the value of $\mathsf{RO}(M||m)$. Therefore, S cannot simulate the relation (event $\mathsf{E3}$ and event $\mathsf{E5}$). If two or more such pairs exist ($(M, x), (M', x), \cdots \in \mathcal{L}_{\mathsf{RO}}$), S must simulate the relation such that $\mathsf{RO}(M||m) = \mathsf{RO}(M'||m) =, \cdots$. However, since S cannot control the outputs of RO, it cannot simulate the relation (event $\mathsf{E4}$).

On the other hand, if $\exists (M, x) \in \mathcal{L}_{\mathsf{RO}}$ such that $x = IV$, S must simulate the relation such that $\mathsf{RO}(m) = \mathsf{RO}(M||m)$. However, since S cannot control the outputs of RO, it cannot simulate the relation (event $\mathsf{E6}$).

In following game transforms, since the MD construction is considered in Game 3 for the first time, we discuss these events in the transform from Game 2 to Game 3. In this discussion, we show that if distinguishing events don't occur, Game 3 is identical to Game 2, and the probability that one of the events will occur is negligible.

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

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

Game 3: In this game, we modify the relay algorithm R_0 into R_1 as follows. For hash oracle query M, R_1 applies the MD construction to M by querying S. R_1 is essentially the same as MD^h except that R_1 is based on S instead of the fixed input length random oracle h.

We show that Game 3 is identical with Game 2 unless the following bad events occur. In response to query (x, m), S chooses response $y \in \{0,1\}^n$:

- $\mathsf{E1}$: It is the case that $y = IV$.
- $\mathsf{E2}$: There is a triple $(x', m', y') \in \mathcal{T}$, with $(x', m') \neq (x, m)$, such that $y' = y$.
- $\mathsf{E3}$: There is a triple $(x', m', y') \in \mathcal{T}$, with $(x', m') \neq (x, m)$, such that $x' = y$ and (x', m', y') is defined exept for step 3 of EO.

and in a response to a query M to RO, RO returns z:

- $\mathsf{E4}$: There is a pair $(M', z') \in \mathcal{L}_{\mathsf{RO}}$, with $M \neq M'$ such that $z = z'$.
- $\mathsf{E5}$: There is a triple $(x', m', y') \in \mathcal{T}$ such that $z = x'$.
- $\mathsf{E6}$: $z = IV$.

We demonstrate that Game 3 is identical with Game 2 unless bad events occur and the probability that bad events occur is negligible. Before we demonstrate these facts, we give an useful property as follows.

Lemma 1. *For any chain triples* $(x_1, m_1, y_1), ..., (x_i, m_i, y_i)$ *in* $\mathcal{T}$, $y_i = \mathsf{RO}(m_1||...||m_i)$ *holds unless bad events occur.*

Proof. To contrary, assume that $y_i \neq \mathsf{RO}(m_1||...||m_i)$. Since y_i is defined in step 2 of S (case A), step 2 of EO (case B), step 3 of EO (case C), or step 4 of EO (case D), we show that when y_i is defined in each step, bad events occur.

First, we discuss the case A. In this case, we divided two case: When (x_i, m_i, y_i) is stored, another chain triples (x'_1, m'_1, y'_1), ..., (x'_t, m'_t, y'_t) are already stored in $\mathcal{T}$ such that $y_t = y_{i-1}$ (case A-1) and chain triples are not stored in $\mathcal{T}$ (case A-2). The case A-1 is equal to collision of MD^{S}. Therefore a collision of S occurs or an output of S is equal to IV in this case. Therefore event $\mathsf{E1}$ or $\mathsf{E2}$ occurs. In the case A-2, since $y_i = \mathsf{RO}(m_1||...||m_i)$ holds from the definition of S, this is contrary to the assumption.

We discuss the case B. In this case, we divided two cases: $i = 1$ (case B-1) and $i \neq 1$ (case B-2). In the case B-1, $y_1 = \mathsf{RO}(m_1)$ holds due to the definition of S. This is contrary to the assumption. In the case B-2, since $x_i = IV$, $y_{i-1} = IV$ holds. Therefore event $\mathsf{E1}$ or $\mathsf{E6}$ occurs.

We discuss the case C. In this case, (M, x_i) is already in $\mathcal{L}_{\mathsf{RO}}$, when y_i is defined. We consider two cases: $M = m_1||...||m_{i-1}$ (case C-1) and $M \neq m_1||...||m_{i-1}$ (case C-2). In the case C-1, $y_i = \mathsf{RO}(m_1||...||m_i)$ holds and this is contrary to the assumption. In the case C-2, we consider two case: y_{i-1} is chosen at random by EO (case C-2-1) and y_{i-1} is defined by RO (case C-2-2). For the case C-2-1, from the definition of S, when $(x_{i-1}, m_{-1i}, y_{i-1})$ is stored in $\mathcal{T}$, some triple (x_j, m_j, y_j) is not in $\mathcal{T}$. Assume that j is the maximum number. Therefore $y_{j+1}, ..., y_{i-1}$ are defined at random by EO and independent from RO. $(x_{j+1}, m_{j+1}, y_{j+1})$ is stored in $\mathcal{T}$ before (x_j, m_j, y_j) is stored in $\mathcal{T}$. If y_j is defined at random by EO and independent from RO, event $\mathsf{E3}$ occurs. If y_j is defined by RO ($y_i = \mathsf{RO}(m_1||...||m_j)$), event $\mathsf{E5}$ occurs. The case C-2-2 is equal to event $\mathsf{E4}$.

Finally we discuss the case D. From the same discussion of the case C-2-1, bad event $\mathsf{E3}$ or $\mathsf{E5}$ occurs. □

For the view of D for R_0 and R_1, from Lemma 1, for any M, $R_1(M) = \mathsf{RO}(M)$ holds unless bad events occur. Therefore the view of D for R_0 is equal to that for R_1. For consistency in Game 2, from the definition of S and Lemma 1, for any chain triples $(x_1, m_1, y_1), ..., (x_i, m_i, y_i) \in \mathcal{T}$, $y_i = \mathsf{RO}(m_1||...||m_i) = R_0(m_1||...||m_i)$ holds unless bad events occur. Therefore, the answers given by S are consistent with those given by R_0. For consistency in Game 3, from the definition of S, the definition of R_1 and Lemma 1, for any chain triples $(x_1, m_1, y_1), ..., (x_i, m_i, y_i) \in \mathcal{T}$, $y_i = R_1(m_1||...||m_i) = \mathsf{RO}(m_1||...||m_i)$ holds unless bad events occur. Therefore, the answers given by S are consistent with those given by R_1. Therefore, Game 3 is identical with Game 2 unless bad events occur.

Next we examine the probability that bad events occur as follows.

Lemma 2. $Pr[\mathsf{E1} \vee \mathsf{E2} \vee \mathsf{E3} \vee \mathsf{E4} \vee \mathsf{E5} \vee \mathsf{E6}] \leq \frac{2q_1^2+q_2^2+q_1q_2+q_1+q_2}{2^n}$ *where* q_1 *is the maximum number of invoking the simulator and* q_2 *is the maximum number of invoking* RO.

Proof. We will examine each of the three events and bound their probability. Since outputs of S are chosen at random, $Pr[\mathsf{E1}] \leq \frac{q_1}{2^n}$. Since $\mathsf{E2}$ is the event where a collision occurs, $Pr[\mathsf{E2}] \leq 1 - \frac{2^n-1}{2^n} \cdots \frac{2^n-q_1+1}{2^n} \leq \frac{q_1^2}{2^n}$. Since y is chosen at random, the probability that event $\mathsf{E3} \leq \frac{q_1^2}{2^n}$. Since $\mathsf{E4}$ is the event that a RO collision occurs, $\Pr[\mathsf{E4}] \leq \frac{q_2^2}{2^n}$. Since $\mathsf{E5}$ is the event that a random value is equal to some fixed value, $Pr[\mathsf{E5}] \leq \frac{q_1 q_2}{2^n}$. Since $\mathsf{E6}$ is the event that a random value is equal to IV, $Pr[\mathsf{E6}] \leq \frac{q_2}{2^n}$. Therefore $Pr[\mathsf{E1} \vee \mathsf{E2} \vee \mathsf{E3} \vee \mathsf{E4} \vee \mathsf{E5} \vee \mathsf{E6}] \leq Pr[\mathsf{E1}] + Pr[\mathsf{E2}] + Pr[\mathsf{E3}] + Pr[\mathsf{E4}] + Pr[\mathsf{E5}] + Pr[\mathsf{E6}] \leq \frac{2q_1^2+q_2^2+q_1q_2+q_1+q_2}{2^n}$. □

Let $\mathsf{G3}$ denote the event that the distinguisher D outputs 1 in Game 3, $\mathsf{B2}$ be the event wherein $\mathsf{E1} \vee \mathsf{E2} \vee \mathsf{E3} \vee \mathsf{E4} \vee \mathsf{E5} \vee \mathsf{E6}$ occurs in Game 2 and $\mathsf{B3}$ be the event wherein $\mathsf{E1} \vee \mathsf{E2} \vee \mathsf{E3} \vee \mathsf{E4} \vee \mathsf{E5} \vee \mathsf{E6}$ occurs in Game 3. From Lemma 2, the probability that bad events occur in Game 2 is less than $\frac{\sigma_H^2+3q_h^2+3q_j\sigma_H+2q_h+\sigma_H}{2^n}$ and the probability that bad events occur in Game 3 is less than $\frac{4(\sigma_H+q_h)^2+2(\sigma_H+q_h)}{2^n}$. Therefore $|Pr[\mathsf{G3}]-Pr[\mathsf{G2}]| = |Pr[\mathsf{G3} \wedge \mathsf{B3}]+Pr[\mathsf{G3} \wedge \neg \mathsf{B3}]-Pr[\mathsf{G2} \wedge \mathsf{B2}]-Pr[\mathsf{G2} \wedge \neg \mathsf{B2}]| \leq |Pr[\mathsf{G3}|\mathsf{B3}] \times Pr[\mathsf{B3}] - Pr[\mathsf{G2}|\mathsf{B2}] \times Pr[\mathsf{B2}]| \leq \max\{Pr[\mathsf{B2}], Pr[\mathsf{B3}]\} = \frac{4(\sigma_H+q_h)^2+2(\sigma_H+q_h)}{2^n}$.

Game 4: In this Game, we modify simulator S to S_1. RO is removed from simulator S_1 as follows.

Simulator S_1: On query (x, m),

1. If $\exists(x, m, y) \in \mathcal{T}$, it responds with y.
2. Else S_1 chooses $y \leftarrow \{0,1\}^n$ at random.
3. $\mathcal{T} \leftarrow (x, m, y)$.
4. S_1 responds with y.

The output of S is chosen at random or chosen by RO. Therefore, for any fresh query to S, the response is chosen at random. Since RO is invoked only by S, no D can access RO. Namely, no D distinguish S_1 from S, though RO is removed in S_1, so Game 4 is identical to Game 3. Let $\mathsf{G4}$ denote the event that distinguisher D outputs 1 in Game 4. $Pr[\mathsf{G4}] = Pr[\mathsf{G3}]$ holds.

Game 5. This is the final game of our argument. Here we finally replace S_1 with the fixed input length random oracle h. Let $\mathsf{G5}$ denote the event that distinguisher D outputs 1 in Game 5. Since for a new query S_1 responds with a random value and for a repeated query S_1 responds a repeated value, Game 5 is identical to Game 4. Therefore, we can deduce that $Pr[\mathsf{G5}] = Pr[\mathsf{G4}]$.

Now we can complete the proof of Theorem by combining Games 1 to 5, and observing that Game 1 is the same as ERO scenario while Game 5 is same as MD^h scenario. Hence we can deduce that $\epsilon \leq \frac{4(\sigma_H+q_h)^2+2(\sigma_H+q_h)}{2^n}$. □

4.2 Proof of Theorem 6

We define simulator S that simulates EO. S has initially empty list $\mathcal{L}_S$. On query (m, z), S is defined as follows: $z' \leftarrow h(z, m)$, and it returns z'. The simulator's running time requires at most $O(q_{\mathsf{EO}})$ time.

We need to prove that S cannot tell apart two scenarios, MD^h and ERO scenarios, one where D has oracle access to MD^h and S and the other where D has access to RO and EO. The proof involves a hybrid argument starting in the MD^h scenario, and ending in the ERO scenario through a sequence of mutually indistinguishable hybrid games.

Game 1: This is the MD^h scenario, where D has oracle access to MD^h and $\mathsf{S}(h)$. Let $\mathsf{G1}$ denote the event that D outputs 1 after interacting with MD^h and $\mathsf{S}(h)$. Thus $Pr[\mathsf{G1}] = Pr[D^{\mathsf{MD}^h,\mathsf{S}(h)} = 1]$.

Game 2: In this game, we change the underlying primitive of MD from h to S. Thus D interacts with MD^{S} and $\mathsf{S}(h)$. For any query to S, S poses it to h and returns the value received from h. Let $\mathsf{G2}$ denote the event that D outputs 1 in Game 2. Since the view of D remains unchanged in this game, so $Pr[\mathsf{G2}] = Pr[\mathsf{G1}]$.

Game 3: In this game, we remove S and h and insert EO and RO. In this game, D interacts with $\mathsf{MD}^{\mathsf{EO}}$ and EO and does not access to RO. Since for a fresh query EO returns a fresh random value and for a repeated query EO returns the corresponding value, Game 3 is identical with Game 2. Let $\mathsf{G3}$ denote the event that D outputs 1 in Game 3. Since the view of D remains unchanged in this game, so $Pr[\mathsf{G3}] = Pr[\mathsf{G2}]$.

Game 4. This is the final game of our argument. In this game, we remove $\mathsf{MD}^{\mathsf{EO}}$ and D interacts with RO and EO. We show that Game 4 is identical with Game 3 unless following bad events occur and probability that bad events occur is negligible.

Bad events are as follows. On query (m, x), EO returns y:

- $\mathsf{Bad1}$: $y = IV$.

On query M, RO returns z:

- $\mathsf{Bad2}$: There is a pair (M', z') in $\mathcal{L}_{\mathsf{EO}}$, with $M \neq M'$, such that $z = z'$.
- $\mathsf{Bad3}$: There is a triple (m, x, y) in $\mathcal{L}_{\mathsf{EO}}$ such that $z = x$.

We demonstrate that Game 4 is identical with Game 3 unless bad events occur and the probability that bad events occur is negligible. Before we demonstrate these facts, we give an useful property as follows.

Lemma 3. *For any chain triples* $(x_1, m_1, y_1), ..., (x_i, m_i, y_i)$ *in* $\mathcal{L}_{\mathsf{EO}}$, $y_i = \mathsf{RO}(m_1||...||m_i)$ *holds unless bad events occur.*

Due to lack of space, we omit this proof. We will show this in the full version.

For the view of D for $\mathsf{MD}^{\mathsf{EO}}$ and RO, from Lemma 3, the view of D for $\mathsf{MD}^{\mathsf{EO}}$ is equal to that for RO. For consistency in Game 3, from the definition of MD and Lemma 3, for any chain triples $(m_1, x_1, y_1), ..., (m_i, x_i, y_i) \in \mathcal{L}_{\mathsf{EO}}$, $y_i = \mathsf{RO}(m_1||...||m_i) = \mathsf{MD}^{\mathsf{EO}}(m_1||...||m_i)$ holds unless bad events occur. Therefore,

the answers given by S are consistent with those given by $\mathsf{MD}^{\mathsf{EO}}$. For consistency in Game 4, from Lemma 3, for any chain triples $(x_1, m_1, y_1), ..., (x_i, m_i, y_i) \in \mathcal{L}_{\mathsf{EO}}$, $y_i = \mathsf{RO}(m_1||...||m_i)$ holds unless bad events occur. Therefore, the answers given by S are consistent with those given by RO. Therefore, Game 4 is identical with Game 3 unless bad events occur.

Next we examine the probability that bad events occur as follows.

Lemma 4. *$Pr[\mathsf{Bad1} \vee \mathsf{Bad2} \vee \mathsf{Bad3}] \leq \frac{q_1+q_2^2+q_1q_2}{2^n}$ where q_1 is the maximum number of invoking EO and q_2 is the maximum number of invoking RO.*

Due to lack of space we omit this proof.

Let $\mathsf{G4}$ denote the event that the distinguisher D outputs 1 in Game 4, $\mathsf{B3}$ be the event that $\mathsf{Bad1} \vee \mathsf{Bad2} \vee \mathsf{Bad3}$ occurs in Game 3 and $\mathsf{B4}$ be the event that $\mathsf{Bad1} \vee \mathsf{Bad2} \vee \mathsf{Bad3}$ occurs in Game 4. Therefore $|Pr[\mathsf{G4}] - Pr[\mathsf{G3}]| \leq \max\{Pr[\mathsf{B3}], Pr[\mathsf{B4}]\} = \frac{2(\sigma_H+q_{\mathsf{EO}})^2+(\sigma_H+q_{\mathsf{EO}})}{2^n}$. □

4.3 MGF1 Transform

In the above discussions, we ignored range extension algorithms such as MGF1 which is an instantiated hash function of OAEP. When we consider these algorithms, we need to modify TRO and ERO. Due to the lack of space, we only modify TRO for MGF1 as follows and will discuss ERO in the full paper.

Let $H : \{0,1\}^* \to \{0,1\}^n$ be some hash function and $MGF1 : \{0,1\}^* \to \{0,1\}^{jn}$ be $H(M||[1])||H(M||[2])||...||H(M||[j])$ where M is the input of the hash function and $[s]$ is the encoding value of s. We confirm the security of cryptosystems that use MGF1 transform with MD^h by the following approach. Let $MGF1 : \{0,1\}^* \to \{0,1\}^{jn}$.

- Propose the modification of TRO (denote TRO' that consists of random oracle $\mathsf{RO}' : \{0,1\}^* \to \{0,1\}^{jn}$ and TO of RO') such that $MGF1(\mathsf{TRO}) \sqsubset \mathsf{TRO}'$.
- Prove cryptosystems security in TRO' model.

If we can find above TRO', since $\mathsf{MD}^h \sqsubset \mathsf{TRO}$, cryptosystems that are secure in TRO' model are secure under MD^h.

TRO' is as follows. TRO' consists of random oracle $\mathsf{RO}' : \{0,1\}^* \to \{0,1\}^{jn}$ and TO', a variant of TO. Let $z[s]$ be the s-th block of z. On trace query (j, w) to TO',

- If there exist pairs such that $(M, z) \in \mathcal{L}_{\mathsf{RO}}$ such that $z[j] = w$, TO' returns all such pairs.
- Otherwise, TO' returns $\perp$.

When H is a random oracle, we can see $H(*||[1]), ..., H(*||[j])$ as independent random oracles $\mathsf{RO}_1, ..., \mathsf{RO}_j$. In order to prove $MGF1(\mathsf{TRO}) \sqsubset \mathsf{TRO}'$, we need to find a simulator that simulates each TO of $\mathsf{RO}_1, ..., \mathsf{RO}_j$. The simulator of TO of RO_s can be easily shown by using queries $(s, *)$ to TO'. Therefore, we can prove $MGF1(\mathsf{TRO}) \sqsubset \mathsf{TRO}'$.

Cryptosystems that are secure in the TRO model are also secure in the TRO′ model by discussions similar to those for the cases of TRO. Note that security bound of these cryptosystems is dependent on n, not jn.

The same discussion can be applied to KDF3 which is an instantiated hash function of RSA-KEM[11].

5 Security Analysis of RSA-KEM in TRO and ERO Models

The RSA-based key encapsulation mechanism (RSA-KEM) scheme [11] is a secure KEM scheme in the RO model. In this section, we consider the security of RSA-KEM in the TRO and ERO models.

The notation of the scheme follows that in [11]. The security of RSA-KEM in the RO model is proved as follows;

Lemma 5 (Security of RSA-KEM in the RO model [11]). *If the RSA problem is hard, then RSA-KEM satisfies IND-CCA for KEM where KDF is modeled as* RO*.*

5.1 Insecurity of RSA-KEM in TRO Model

Though RSA-KEM is secure in the RO model, it is insecure in the TRO model. More specifically, we can show that RSA-KEM does not even satisfy IND-CPA for KEM in the TRO model. Note that IND-CPA means IND-CCA without $\mathcal{DO}$.

Theorem 7 (Insecurity of RSA-KEM in the TRO model). *Even if the RSA problem is hard, RSA-KEM does not satisfy IND-CPA for KEM where KDF is modeled as* TRO*.*

Proof. We construct an adversary, $\mathcal{A}$, which successfully plays the IND-CPA by using TRO KDF. The construction of $\mathcal{A}$ is as follows;

Input : (n, e) as the public key
Output : b' as the guessed bit
Step 1 : Return *state* and receive (K_b^*, C_0^*) as the challenge. Pose the trace query K_b^* to KDF, and obtain $\{r\}$.
Step 2 : For all r in $\{r\}$, check whether $r^e \stackrel{?}{\equiv} C_0^* \pmod{n}$. If there is r^* that satisfies the relation, output $b' = 0$. Otherwise, output $b' = 1$.

We estimate the success probability of $\mathcal{A}$. When challenge ciphertext C_0^* is generated, r^* such that $K_0^* = KDF(r^*)$ is certainly posed to KDF because C_0^* is generated following the protocol description. Thus, $\mathcal{L}_{KDF}$ contains (r^*, C_0^*, K_0^*). If (r^*, C_0^*, K_b^*) is not in $\mathcal{L}_{KDF}$, then $b = 1$. Therefore, $\mathcal{A}$ can successfully play the IND-CPA game. □

5.2 Security of RSA-KEM in ERO Model

We can also prove the security of RSA-KEM in the ERO model as well as in the RO model.

Theorem 8 (Security of RSA-KEM in the ERO model). *If the RSA problem is (t', ϵ')-hard, then RSA-KEM satisfies (t, ϵ)-IND-CCA for KEM as follows: $t' = t + (q_{RKDF} + q_{EKDF}) \cdot expo$, $\epsilon' \geq \epsilon - \frac{q_D}{n}$, where KDF is modeled as ERO, q_{RKDF} is the number of hash queries posed to the RO of KDF, q_{EKDF} is the number of extension attack queries posed to the EO of KDF, q_D is the number of queries posed to the decryption oracle $\mathcal{DO}$ and expo is the running time of exponentiation modulo n.*

The proof will be described in the full paper.

Acknowledgements. We would like to thank the anonymous referees for their many useful comments.

References

1. An, J.H., Bellare, M.: Constructing vil-macsfrom fil-macs: Message authentication under weakened assumptions. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 252–269. Springer, Heidelberg (1999)
2. Bellare, M., Ristenpart, T.: Multi-property-preserving hash domain extension and the EMD transform. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 299–314. Springer, Heidelberg (2006)
3. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM Conference on Computer and Communications Security, pp. 62–73 (1993)
4. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
5. Coron, J.-S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-damgård revisited: How to construct a hash function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (2005)
6. Damgård, I.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (1990)
7. Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging merkle-damgård for practical applications. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 371–388. Springer, Heidelberg (2009)
8. Hirose, S., Park, J.H., Yun, A.: A Simple Variant of the Merkle-Damgård Scheme with a Permutation. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 113–129. Springer, Heidelberg (2007)
9. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004)
10. Merkle, R.C.: One way hash functions and des. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 428–446. Springer, Heidelberg (1990)
11. Shoup, V.: A proposal for an iso standard for public key encryption, version 2.1 (2001)
12. Tsudik, G.: Message authentication with one-way hash functions. In: INFOCOM, pp. 2055–2059 (1992)
13. Yoneyama, K., Miyagawa, S., Ohta, K.: Leaky random oracle (extended abstract). In: Baek, J., Bao, F., Chen, K., Lai, X. (eds.) ProvSec 2008. LNCS, vol. 5324, pp. 226–240. Springer, Heidelberg (2008)