Reset Indifferentiability from Weakened Random Oracle Salvages One-Pass Hash Functions

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

[1] Mitsubishi Electric Corporation
Naito.Yusuke@ce.MitsubishiElectric.co.jp
[2] NTT Secure Platform Laboratories
yoneyama.kazuki@lab.ntt.co.jp
[3] The University of Electro-Communications
kazuo.ohta@uec.ac.jp

Abstract. Ristenpart et al. (EUROCRYPT 2011) showed that the indifferentiability theorem of Maurer et al. (TCC 2004) does not cover all multi-stage security notions; it only covers single-stage security notions. They defined reset indifferentiability, and proved the reset indifferentiability theorem, which covers all security notions; if a hash function is reset indifferentiable from a random oracle denoted by $\mathcal{RO}$, for any security, any cryptosystem is at least as secure under the hash function as in the $\mathcal{RO}$ model. Unfortunately, they also proved the impossibility of one-pass hash functions such as ChopMD and Sponge; there exists a multi-security notion such that some cryptosystem is secure in the $\mathcal{RO}$ model but insecure when $\mathcal{RO}$ is replaced with a one-pass hash function.

In order to ensure other multi-stage security notions, we propose a new methodology, called the $\mathcal{WRO}$ methodology, instead of the $\mathcal{RO}$ methodology. We consider "Reset Indifferentiability from Weakened Random Oracle" which salvages ChopMD and Sponge. The concrete procedure of the $\mathcal{WRO}$ methodology is as follows:

1. Define a new concept of $\mathcal{WRO}$ instead of $\mathcal{RO}$,
2. Prove that a hash function H is reset indifferentiable from $\mathcal{WRO}$, (here the examples are ChopMD and Sponge), and
3. For multi-stage security $\mathcal{G}$, prove that a cryptosystem $\mathcal{C}$ is $\mathcal{G}$-secure in the $\mathcal{WRO}$ model.

As a result, $\mathcal{C}$ with H is $\mathcal{G}$-secure by combining the results of Steps 2, 3, and the theorem of Ristenpart et al. Moreover, for a public-key encryption scheme (as $\mathcal{C}$) and the chosen-distribution attack game (as the game of $\mathcal{G}$) we prove that $\mathcal{C}(\mathcal{WRO})$ is $\mathcal{G}$-secure, which implies the appropriateness of the new concept of the $\mathcal{WRO}$ methodology.

Keywords: Indifferentiable hash function, reset indifferentiability, multi-stage game, Sponge, ChopMD.

1 Introduction

1.1 Indifferentiability

The Indifferentiability theorem [12] of Maurer, Renner, and Holenstein (MRH), called MRH theorem, covers all single-stage security notions $\mathbf{G}_s$ and all cryptosystems $\mathbf{C}$; for

I. Boureanu, P. Owesarski, and S. Vaudenay (Eds.): ACNS 2014, LNCS 8479, pp. 235–252, 2014.

$\forall \mathcal{G} \in \mathbf{G}_s$ and $\forall C \in \mathbf{C}$, C is at least as $\mathcal{G}$-secure in the F_1 model as in the F_2 model, denoted by $C(F_1) \succ_{\mathcal{G}} C(F_2)$, if "$F_1$ is indifferentiable from F_2", denoted by $F_1 \sqsubset F_2$. The game of $F_1 \sqsubset F_2$ is a simulation based game where some stateful simulator S is constructed, which represents some adversary in the F_2 model, thereby this game ensures that any adversary in the F_1 model can obtain some information in the F_2 model due to S. Thus, this framework distinguishes interfaces of F which are adversarial and honest interfaces, denoted by $F.adv$ and $F.hon$, respectively. Adversaries are permitted to access to the adversarial interface, and honest parties are permitted to access to the honest interface. The definition of $F_1 \sqsubset F_2$ is that there exists a stateful simulator S such that for any distinguisher $\mathcal{D}$ which interacts with two oracles (L, R), no $\mathcal{D}$ can distinguish real world $(L, R) = (F_1.hon, F_1.adv)$ from ideal world $(L, R) = (F_2.hon, S^{F_2.adv})$ where S has access to $F_2.adv$. MRH proved the following theorem:

MRH Theorem.[12] $F_1 \sqsubset F_2 \Rightarrow \forall C \in \mathbf{C}, \forall \mathcal{G} \in \mathbf{G}_s$: $C(F_1) \succ_{\mathcal{G}} C(F_2)$.

1.2 $\mathcal{RO}$ Methodology

Coron, Dodis, Malinaud, and Puniya [7] pointed out that the MRH theorem opened a nice modular approach for security proofs of cryptosystems using hash function $H^{\mathcal{U}}$; $H^{\mathcal{U}} \sqsubset \mathcal{RO} \Rightarrow \forall C \in \mathbf{C}, \forall \mathcal{G} \in \mathbf{G}_s$: $C(H^{\mathcal{U}}) \succ_{\mathcal{G}} C(\mathcal{RO})$ where it is assumed that the underlying primitive $\mathcal{U}$ is ideal. Thus designers of hash functions only have to concentrate on designing H such that $H^{\mathcal{U}} \sqsubset \mathcal{RO}$, and ones of cryptosystems concentrate on designing of C such that C is $\mathcal{G}$-secure in the $\mathcal{RO}$ model. This approach is called as the Random Oracle ($\mathcal{RO}$) methodology. In the proof of $H^{\mathcal{U}} \sqsubset \mathcal{RO}$, the real world is $(L, R) = (H^{\mathcal{U}}, \mathcal{U})$ and the ideal world is $(L, R) = (\mathcal{RO}, S^{\mathcal{RO}})$. Hereafter, we call hash function $H^{\mathcal{U}}$ such that $H^{\mathcal{U}} \sqsubset \mathcal{RO}$ an "IFRO (indifferentiable from a $\mathcal{RO}$) hash function" and its construction the "IFRO hash construction".

So far, many IFRO hash constructions have been proposed such as the Chop Merkle-Damgård (ChopMD) construction [7] and the Sponge construction [4]. SHA-512/224 and SHA-512/256, which are standarized in FIPS 180-4 [17], employ the ChopMD construction, and the SHA-3 winner Keccak [16,5] employs the Sponge construction. Therefore, IFRO security is an important criterion of designing hash functions.

1.3 Impossibility of IFRO Security in Multi-Stage Security Games

However, Ristenpart, Shacham, and Shrimpton (RSS) [18] pointed out that the MRH theorem covers all single-stage security notions $\mathbf{G}_s$, while it does not cover all multi-stage security notions $\mathbf{G}_m$.

The impossibility is from a difference of conditions of "state" sizes between indifferentiability and multi-stage security. Indifferentiability deals with a "stateful" simulator, that is, the size of the sate of the simulator is not restricted. On the other hand, in multi-stage games, the size of the state shared among adversaries is restricted.

They gave an example meeting the impossibility. They defined a two party challenge response protocol $\mathcal{CR}$ and multi-stage security, called CRP-security. They showed that $\mathcal{CR}$ is CRP-secure in the $\mathcal{RO}$ model but insecure when using IFRO one-pass hash functions such as the ChopMD hash function and the Sponge hash function.

Note that the RSS result does not always imply that for $\forall \mathcal{G} \in \mathbf{G}_m$ and $\forall C \in \mathbf{C}$, C is $\mathcal{G}$-secure in the $\mathcal{RO}$ model and insecure when using $H^{\mathcal{U}}$. So we have the following question:

"Can we prove the $\mathcal{G}$-security of $C(H^{\mathcal{U}})$?"

This paper tackles how to solve this question. The candidate to solve this question is reset indifferentiability of RSS [18].

1.4 Reset Indifferentiability

The reset indifferentiability framework is the extension of the indifferentiability framework and this theorem, called RSS theorem, covers all security notions $\mathbf{G}$ ($= \mathbf{G}_s \cup \mathbf{G}_m$). The RSS theorem ensures that for $\forall \mathcal{G} \in \mathbf{G}$ and $\forall C \in \mathbf{C}$, $C(F_1) \succ_{\mathcal{G}} C(F_2)$ if F_1 is reset indifferentiable from F_2, denoted by $F_1 \sqsubset_r F_2$. The reset indifferentiable game is the same simulation based game as the indifferentiable game [12]. The difference is that indifferentiability deals with a stateful simulator, while reset indifferentiability deals with a *stateless* simulator. The "stateless" setting reflects the settings of multi-stage security games where the state size among adversaries is restricted. So the definition of $F_1 \sqsubset_r F_2$ is that there exists a stateless simulator S such that for any distinguisher $\mathcal{D}$ which interacts with two oracles (L, R), no $\mathcal{D}$ can distinguish a real world $(L, R) = (F_1.hon, F_2.adv)$ from an ideal world $(L, R) = (F_2.hon, S^{F_2.adv})$. RSS proved the following theorem.

RSS Theorem. [18] $F_1 \sqsubset_r F_2 \Rightarrow \forall \mathcal{G} \in \mathbf{G}, \forall C \in \mathbf{C}$: $C(F_1) \succ_{\mathcal{G}} C(F_2)$.

And, the RSS theorem offers the corollary: $H^{\mathcal{U}} \sqsubset_r \mathcal{RO} \Rightarrow \forall \mathcal{G} \in \mathbf{G}, \forall C \in \mathbf{C} : C(H^{\mathcal{U}}) \succ_{\mathcal{G}} C(\mathcal{RO})$.

Unfortunately, RSS also proved the *impossibility* of $H^{\mathcal{U}} \sqsubset_r \mathcal{RO}$ where H is a one-pass hash construction such as the ChopMD construction and the Sponge construction. Therefore, we have to consider another solution than the $\mathcal{RO}$ methodology.

1.5 Our Contributions – A New Proposal of $\mathcal{WRO}$ Methodology –

We propose a $\mathcal{WRO}$ methodology which is based on "Reset Indifferentiability from Weakened Random Oracle ($\mathcal{WRO}$)" in order to ensure the $\mathcal{G}$-security of $C(H^{\mathcal{U}})$. This paper deals with the ChopMD construction and the fixed output length Sponge (FOL-Sponge) construction as H, because these are employed in important hash functions such as SHA-512/224, SHA-512/256, and SHA-3 winner Keccak.

The concrete proof procedure of the $\mathcal{WRO}$ methodology is as follows:

1. Define a new concept of $\mathcal{WRO}$ instead of $\mathcal{RO}$,
2. Prove that $H^{\mathcal{U}} \sqsubset_r \mathcal{WRO}$ assuming $\mathcal{U}$ is ideal, and
3. Prove that C is $\mathcal{G}$-secure in the $\mathcal{WRO}$ model.

As a result we can ensure that $C(H^{\mathcal{U}})$ is $\mathcal{G}$-secure by combining the results of Steps 2 and 3, and the RSS theorem. Moreover, for public-key encryption (as cryptosystem C) and Chosen Distribution Attack [1,2] (as game $\mathcal{G}$) we prove that $C(\mathcal{WRO})$ is $\mathcal{G}$-secure, which implies the appropriateness of the new concept of the $\mathcal{WRO}$ model.

$\mathcal{D}$'s Procedure 1 (Condition 1)
1. $\mathcal{D}$ makes a query x to R and receives the response y_1.
2. $\mathcal{D}$ makes a query x to R and receives the response y_2.

Fig. 1. Distinguisher's Procedure 1

$\mathcal{D}$'s Procedure 2 (Condition 2)
1. $\mathcal{D}$ makes a query $IV||M_1$ to R and receives the response y_1
2. $\mathcal{D}$ makes a query $y_1||M_2$ to R and receives the response y_2

Fig. 2. Distinguisher's Procedure 2

We define $\mathcal{WRO}$ so that one can construct a stateless simulator such that $H^{\mathcal{U}} \sqsubset_r \mathcal{WRO}$, that is, an adversary can simulate information of $\mathcal{U}$ (= $H^{\mathcal{U}}.adv$) from $\mathcal{WRO}.adv$. Thus $\mathcal{WRO}$ consists of $\mathcal{RO}$ and sub oracle O^* which leaks information to simulate $\mathcal{U}$, and the interfaces are defined as $\mathcal{WRO}.hon = \mathcal{RO}$ and $\mathcal{WRO}.adv = (\mathcal{RO}, O^*)$.

To our knowledge, our result is the first result to ensure the reducibility from a real model to an ideal model for the important hash constructions, ChopMD and FOLSponge.

How to Define O^*. We explain how to define O^* by basing on the proof of ChopMD$^h \sqsubset \mathcal{RO}$, where $h : \{0,1\}^{m+2n} \to \{0,1\}^{2n}$ is a random oracle compression function. For two block message $M_1||M_2$, the output of ChopMD is calculated as ChopMD$^h(M_1||M_2) = chop_n(h(h(IV||M_1)||M_2))$ where $chop_n$ accepts $2n$ bit value $x'||x^*$ and returns the right n bit value x^*. In this case, the real world is $(L, R) = (\text{ChopMD}^h, h)$. In the indifferentiable game, distinguisher $\mathcal{D}$ tries to distinguish the real world from the ideal world by using query-response values of (L, R). Therefore, the following two points are required to construct a simulator S. The first point is the simulation of h. The second point is the simulation of the relation between L and R in the real world, because L uses R in the real world. We explain the simulations by considering the use of the S's state.

Simulation of h: We explain the simulation of h by using Fig. 1. This example is that $\mathcal{D}$ makes a repeated query. In the real world the responses y_1 and y_2 satisfy the following conditions, since R is a random oracle h,

- **Condition 1:** y_1 is a random value and $y_2 = y_1$.

The following demonstrates that S can return responses satisfying the condition by using the S's state.

- **Constructing S:** In Step 1 S chooses a random value as the response y_1 for the query x. Then S *records* the query response pair (x, y_1). In Step 2 S finds y_1 from the recorded pair (x, y_1), and defines $y_2 := y_1$.

Simulation of the L-R Relation: We explain the simulation of the relation between L and R by using Fig. 2. In the real world, since $(L, R) = (\text{ChopMD}^h, h)$, the query response values in Fig. 2 satisfy the following conditions.

- **Condition 2:** $chop_n(y_1) = \text{ChopMD}^f(M_1)$ and $chop_n(y_2) = \text{ChopMD}^h(M_1||M_2)$.

The following shows that S can return responses satisfying the condition by using the S's state.

– **Constructing S:** In Step 1 S defines $y_1^* := \mathcal{RO}(M_1)$ for the query $IV||M_1$, chooses a random value y_1', and defines $y_1 := y_1'||y_1^*$. Then S *records* the pair (M_1, y_1). In Step 2, for the query $y_1||M_2$, S finds M_1 from the recorded pair (M_1, y_1). Then S chooses a random value y_2', defines $y_2^* := \mathcal{RO}(M_1||M_2)$, and $y_2 := y_2'||y_2^*$. This procedure ensures that $chop_n(y_1) = \mathcal{RO}(M_1)$ and $chop_n(y_2) = \mathcal{RO}(M_1||M_2)$.

Thus, we can construct a stateful simulator S which ensures the two points. On the other hand, we cannot construct a stateless simulator S which ensures the two points. So we compensate the stateless setting by using sub oracle $\mathcal{O}^*$. We define the sub oracle as follows.

Sub Oracle for Simulation of h: In order to ensure the condition 1, we add random oracle $\mathcal{RO}^*$ to $\mathcal{O}^*$. Then we can construct a stateless simulator S which ensures the condition 1: In Step 1 S defines $y_1 := \mathcal{RO}^*(x)$. In Step 2 S defines $y_2 := \mathcal{RO}^*(x)$. This procedure ensures that $y_1 = y_2$.

Sub Oracle for Simulation of L-R Relation: In order to ensure the condition 2, we add random oracle $\mathcal{RO}^\dagger$ and trace oracle $\mathcal{TO}$ to $\mathcal{O}^*$. The definition of $\mathcal{TO}$ is that for query y_1' to $\mathcal{TO}$, $\mathcal{TO}$ returns M_1 if a query M_1 to $\mathcal{RO}^\dagger$ was made such that $y_1' = \mathcal{RO}^\dagger(M_1)$, otherwise $\mathcal{TO}$ returns $\perp$. Then we can construct a stateless simulator S which ensures the condition 2: In Step 1, for query $IV||M_1$, S defines $y_1' := \mathcal{RO}^\dagger(M_1)$ and $y_1^* := \mathcal{RO}(M_1)$, and $y_1 := y_1'||y_1^*$. In Step 2, for query $y_1||M_2$, S obtains y_1' from y_1 and makes a query y_1' to $\mathcal{TO}$. Then M_1 is returned from $\mathcal{TO}$. Finally S defines $y_2^* := \mathcal{RO}(M_1||M_2)$ and $y_2' := \mathcal{RO}^\dagger(M_1||M_2)$, and $y_2 := y_2'||y_2^*$. This procedure ensures that $chop_n(y_1) = \mathcal{RO}(M_1)$ and $chop_n(y_2) = \mathcal{RO}(M_1||M_2)$.

We thus define $\mathcal{O}^* := (\mathcal{RO}^*, \mathcal{RO}^\dagger, \mathcal{TO})$, thereby we can construct a stateless simulator which ensures the above two conditions, and can prove ChopMD$^h \sqsubset_r \mathcal{WRO}$ (Theorem 2).

Similarly, for the FOLSponge construction, we define $\mathcal{O}^* := (\mathsf{IC}, \mathcal{RO}^\dagger, \mathcal{TO})$, thereby we can construct a stateless simulator which ensures the above two simulations, and can prove FOLSponge $\sqsubset_r \mathcal{WRO}$ (Theorem 3) where $\mathsf{IC} = (E, D)$ is an ideal cipher. E is an encryption oracle and D is a decryption oracle.

Consequently, we define the sub oracle as $\mathcal{O}^* := (\mathcal{RO}^*, \mathcal{RO}^\dagger, \mathcal{TO}, \mathsf{IC})$ in order to evaluate the ChopMD and the FOLSponge constructions by the single $\mathcal{WRO}$. Thus, $\mathcal{WRO}$ consists of $(\mathcal{RO}, \mathcal{RO}^*, \mathcal{RO}^\dagger, \mathcal{TO}, \mathsf{IC})$, and the interfaces are defined as $\mathcal{WRO}.hon = \mathcal{RO}$ and $\mathcal{WRO}.adv = (\mathcal{RO}, \mathcal{RO}^*, \mathcal{RO}^\dagger, \mathcal{TO}, \mathsf{IC})$.

Appropriateness of $\mathcal{WRO}$. We succeed to bypass the impossible result in [18] by introducing the $\mathcal{WRO}$ model; however, it is non-trivial if previous cryptosystems that are secure for multi-stage games in the $\mathcal{RO}$ model are still secure in the $\mathcal{WRO}$ model. Thus, the next step is to show that there exists a secure cryptosystem for a multi-stage game in the $\mathcal{WRO}$ model. We consider public-key encryption (PKE) (as cryptosystem C) for the Chosen Distribution Attack (CDA) game [1,2] (as game $\mathcal{G}$). Roughly, we say a PKE scheme is CDA secure if message privacy is preserved even if an adversary can control distributions of messages and randomness in generating the challenge ciphertext. The CDA game captures several flavors of PKE settings (e.g., deterministic PKE

(DPKE) [1,3,6,10,13], hedged PKE (HPKE) [2], and message-locked PKE (MLPKE)), and such PKE settings are tools for many practical applications. Thus, our target is to find a CDA secure cryptosystem in the $\mathcal{WRO}$ model.

First, we start with the result in [18]. They showed that any CPA secure PKE scheme in the $\mathcal{RO}$ model can be (redundancy-freely) transformed to an IND-SIM secure PKE scheme in the $\mathcal{RO}$ model via conversion REwH1 [2]. The IND-SIM security is a very weak property that an adversary cannot distinguish between encryptions of chosen messages under chosen randomness and the output of a simulator.[1] We show that any IND-SIM secure [18] PKE scheme in the $\mathcal{RO}$ model is also CDA secure in the $\mathcal{WRO}$ model (Theorem 4). The combination of our theorem and the previous result implies that a CDA secure PKE scheme in the $\mathcal{WRO}$ model can be obtained from any CPA secure PKE scheme in the $\mathcal{RO}$ model.[2]

To prove the CDA security in the $\mathcal{WRO}$ model, we must ensure that the sub oracle O^* gives no advantage to an adversary in the CDA game. The CDA game consists of two stages, where a first stage adversary $\mathcal{A}_1$ sends no value to a second stage adversary $\mathcal{A}_2$.[3] First, the challenge ciphertext c_β does not leak any information of messages (m_0, m_1) and r even with access to $\mathcal{RO}$. This property is guaranteed by the IND-SIM security. Next, if $\mathcal{RO}^\dagger$ and $\mathcal{RO}^*$ are ideal primitives whose outputs do not leak no information for the inputs, these oracles give no advantage to the adversary. Finally, $\mathcal{A}_1$ might deliver some information about (m_0, m_1) or r via interfaces of IC, $\mathcal{TO}$ and $\mathcal{RO}^\dagger$. $\mathcal{A}_1$ can pose (m_0, m_1) or r (or a related value) to $\mathcal{RO}^\dagger$, E, and D, where E and D are an encryption oracle and a decryption oracle of IC. If $\mathcal{A}_2$ could pose the corresponding output value of $\mathcal{RO}^\dagger$, E, or D to $\mathcal{TO}$, D, or E, $\mathcal{A}_2$ would obtain information from $\mathcal{A}_1$. However, indeed, $\mathcal{A}_2$ cannot find the corresponding output value except negligible probability because of following two reasons: 1) Any meaningful information from $\mathcal{A}_1$ is not obtained from any of c_β, $\mathcal{RO}$, $\mathcal{RO}^\dagger$ and $\mathcal{RO}^*$ as discussed above. 2) Outputs of $\mathcal{RO}^\dagger$, E, and D are uniformly random, and then a possible action of $\mathcal{A}_2$ is randomly guessing these values. Therefore, $\mathcal{TO}$ and IC also give no advantage to the adversary.

1.6 Related Works

RSS gave a "from scratch" proof where REwH1 using the NMAC hash function [9] is CDA secure. This approach has to consider structures of hash functions, while our approach does not have to consider them. We only have to deal with the handy tool $\mathcal{WRO}$. Moreover, the NMAC hash construction does not cover important hash constructions ChopMD and Sponge.

[1] This definition is meaningless in the standard model because the encryption algorithm uses no further randomness beyond that input.

[2] From Theorem 2 and 3, the CDA security in the $\mathcal{WRO}$ model is preserved if $\mathcal{WRO}$ is replaced with the ChopMD construction and the FOLSponge construction. Therefore, our result achieves that a CDA secure PKE scheme with such practical hash functions can be obtained from any CPA secure PKE scheme in the $\mathcal{RO}$ model.

[3] In the first stage, an adversary $\mathcal{A}_1$ outputs two messages (m_0, m_1) and a random value r such that the jointed values $m_i \| r$ have sufficient min-entropy. In the second stage, an adversary $\mathcal{A}_2$ receives the challenge ciphertext $c_\beta = \mathcal{E}(m_\beta; r)$ from the game where β is a random value of a single bit, and outputs a bit b, where $\mathcal{E}$ is an encryption function. The adversary wins if $b = \beta$.

Two papers [8,11] independently show that for any domain extender H it is impossible to prove $H^{\mathcal{U}} \sqsubset_r \mathcal{RO}$. Because of the impossibility result, it cannot be guaranteed to securely instantiate $\mathcal{RO}$ by $H^{\mathcal{U}}$ via the reset indifferentiability. Thus, they try to salvage H by relaxing limitations of S and/or $\mathcal{D}$. Conversely, we salvage H by showing instantiability from $\mathcal{WRO}$.

Demay et al. [8] propose a relaxed model that is called *resource-restricted indifferentiability*. This model allows simulator S to have a fixed size state while the reset indifferentiability restrict S to be stateless. That means, adversaries in a multi-stage game can share a fixed size (denoted by parameter s) state. They show that it is possible to securely instantiate $\mathcal{RO}$ by $H^{\mathcal{U}}$ via the resource-restricted indifferentiability. Specifically, they define that F_1 is s-resource-restricted indifferentiable from F_2 (denoted by $F_1 \sqsubset_{rr,s} F_2$) if $\exists S$ with the state size s bit s.t. no $\mathcal{D}$ distinguishes the real world $(F_1.hon, F_1.adv)$ from the ideal world $(F_2.hon, S^{F_2.adv})$. They prove that for any multi-stage game security $\mathcal{G}$ that the size of shared state between adversaries in multi-stage is restricted to equal or lower than s bit, $F_1 \sqsubset_{rr,s} F_2 \Rightarrow \forall C \in \mathbf{C}\, C(F_1) \succ_{\mathcal{G}} C(F_2)$.

They also show a necessary condition of parameter s (i.e., $s = l - m - \log q > 0$) to prove $H^{\mathcal{U}} \sqsubset_{rr,s} \mathcal{RO}$ for any domain extender H, where l is the maximal input length of H, m is the input length of the ideal primitive of H (e.g., compression function) and q is the number of query of S. Their theorem is only valid for the case $s > 0$; that is, their result is still restricted to *specific* multi-stage games. Indeed, unfortunately, their approach *cannot* cover security games that shared state between adversaries in multi-stage is restricted to zero (i.e., $s = 0$). Because the CDA game is the case $s = 0$, they cannot salvage H for the CDA game while our result can do that.

Luykx et al. [11] propose a relaxed model that is called *i-reset indifferentiability*. This model restricts distinguisher $\mathcal{D}$ so that $\mathcal{D}$ is allowed to reset the memory of simulator S only i times while the reset indifferentiability allows $\mathcal{D}$ to reset any times. That means, the number of stages in multi-stage games is equal or lower than i. They define that F_1 is i-reset indifferentiable from F_2 (denoted by $F_1 \sqsubset_{r,i} F_2$) if $\exists S$ which is stateful s.t. no $\mathcal{D}$ distinguishes the real world $(F_1.hon, F_1.adv)$ from the ideal world $(F_2.hon, S^{F_2.adv})$, where $\mathcal{D}$ can reset S up to i times. They prove that for any i'-stage $(1 \leq i' \leq i)$ game security $\mathcal{G}$, $F_1 \sqsubset_{r,i} F_2 \Rightarrow \forall C \in \mathbf{C}\, C(F_1) \succ_{\mathcal{G}} C(F_2)$.

Unfortunately, they show the impossibility that $H^{\mathcal{U}} \sqsubset_{r,i} \mathcal{RO}$ cannot be proved for *any one-pass hash construction* even if $i = 1$, and Baecher et al. clarifies that 1-reset indifferentiability is equivalent to the reset indifferentiability. Hence, their approach *cannot* salvage practical H. On the other hand, our result can salvage important and practical one-pass H such as ChopMD and FOLSponge (Theorems 2 and 3); therefore, our methodology with $\mathcal{WRO}$ is more suitable in a practical sense.

Recently, an independent paper from this paper was accepted at EUROCRYPT 2014 [14]. This independent paper proposed the unsplittability approach and showed that this approach salvages some cryptosystems using Merkle-Damgård type hash constructions in some multi-stage security games. Note that the unsplittability approach is different from the $\mathcal{WRO}$ methodology. Moreover, these hash constructions do not include Sponge, while these include ChopMD.

2 Preliminaries

Notations. Given two strings x and y, we use $x||y$ to denote the concatenation of x and y. Given a value y, $x \leftarrow y$ means assigning y to x. When X is a non-empty finite set, we write $x \xleftarrow{\$} X$ to mean that a value is sampled uniformly at random from X and assign to x. $\oplus$ is bitwise exclusive or. $|x|$ is the bit length of x. Given two sets A and C, $C \xleftarrow{\cup} A$ means assign $A \cup C$ to C. For any $l \times r$-bit value M, $div(r, M)$ divides M into r-bit values $(M_1, \ldots, M_l)$ and outputs them where $M_1||\cdots||M_l = M$. For a b-bit value x, $x[i, j]$ is the value from (left) i-th bit to (left) j-th bit where $1 \le i \le j \le b$. For example, let $x = 01101001$, $x[3, 5] = 101$. For a Boolean function F, we denote by "$\exists_1 M$ s.t. $F(M)$ is true" "there exists just a value M such that $F(M)$ is true". Vectors are written in boldface, e.g., $\mathbf{x}$. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes its length and $\mathbf{x}[i]$ denotes its i-th component for $1 \le i \le |\mathbf{x}|$. $bit_j(\mathbf{x})$ is the left j-th bit of $\mathbf{x}[1]||\ldots||\mathbf{x}[|\mathbf{x}|]$.

Throughout this paper, we assume that any algorithm and game is implicitly given a security parameter as input if we do not explicitly state.

Indifferentiability Frameworks [12,18]. The indifferentiability framework [12] ensures reducibility from one system F_1 to another system F_2 in any single-stage security game, where an adversary uses a single state; for any single-stage security, any cryptosystem is at least as secure in F_1 model as in F_2 model. This framework considers two interfaces of system F. One is an adversarial interface, denoted by $F_i.adv$ to which adversaries have access. The other is an honest interface, denoted by $F_i.hon$ to which honest parties have access. In this framework, the reducibility reflects in a simulation based game, called an indifferentiability game. When considering the reducibility from F_1 to F_2, the advantage of this game is defined as follows.

$$\mathsf{Adv}^{\mathsf{indiff}}_{F_1,F_2,S}(A) = |\Pr[\mathcal{D}^{F_1.hon,F_1.adv} \Rightarrow 1] - \Pr[\mathcal{D}^{F_2.hon,S^{F_2.adv}} \Rightarrow 1]|$$

where S is a simulator which has access to $F_2.adv$ and $\mathcal{D}$ is a distinguisher which has access to left oracle L and right oracle R. The F_1 case is that $(L, R) = (F_1.hon, F_1.adv)$, called Real World. The F_2 case is that $(L, R) = (F_2.hon, S^{F_2.adv})$, called Ideal World. The reducibility from F_1 to F_2 is ensured if F_1 is indifferentiable from F_2; there exists a stateful simulator S such that for any $\mathcal{D}$ the indifferentiable advantage is negligible in the security parameter [12].

The reset indifferentiability framework [18] is the extension of the indifferentiability framework and covers any multi-stage security game in addition to any single-stage security game. A multi-stage game is that the size of the state shared among adversaries are restricted. The restricted situation is covered by dealing with a *stateless* simulator. When considering the reducibility from F_1 to F_2, the advantage of this game is defined as follows.

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff}}_{F_1,F_2,S}(A) = |\Pr[\mathcal{D}^{F_1.hon,F_1.adv} \Rightarrow 1] - \Pr[\mathcal{D}^{F_2.hon,S^{F_2.adv}} \Rightarrow 1]|$$

The reducibility from F_1 to F_2 is ensured if F_1 is reset indifferentiable from F_2; there exists a *stateless* simulator S such that for any $\mathcal{D}$ the indifferentiable advantage is negligible in the security parameter [18]. More precisely, RSS gave the following theorem.

$\underline{\text{ChopMD}^h(M)}$
1 $M' \leftarrow \mathsf{pad}_c(M)$;
2 $(M_1, \ldots, M_i) \leftarrow div(d, M')$;
3 $x \leftarrow IV$;
4 for $j = 1, \ldots, i$ do $x \leftarrow h(x \| M_j)$;
5 **return** $x[s+1, s+n]$;

$\underline{\text{FOLSponge}^P(M)}$
1 $M' \leftarrow \mathsf{pad}_S(M)$;
2 $(M_1, \ldots, M_i) \leftarrow div(n, M')$;
3 $s = IV$;
4 for $i = 1, \ldots, i$ do $s = P(s \oplus (M_i \| 0^c))$;
5 **return** $s[1, n]$;

Fig. 3. Chop Merkle-Damgård and Sponge

$\underline{\mathcal{RO}^\dagger_w(M)}$
1 if $\mathsf{F}^\dagger[M] = \perp$ then $\mathsf{F}^\dagger[M] \xleftarrow{\$} \{0,1\}^w$;
2 **return** $\mathsf{F}^\dagger[M]$;

$\underline{\mathcal{TO}(y)}$
1 if $\exists_1 M$ s.t. $\mathsf{F}^\dagger[M] = y$ then **return** M;
2 **return** $\perp$;

Fig. 4. $\mathcal{RO}^\dagger_w$ and $\mathcal{TO}$ where $\mathsf{F}^\dagger$ is a (initially everywhere $\perp$) table

Theorem 1 (RSS Theorem [18]). *Let G be any game. Let F_1 and F_2 be cryptographic systems. Let S be a stateless simulator. For any adversary $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_m)$, there exist an adversary $\mathcal{B} = (\mathcal{B}_1, \ldots, \mathcal{B}_m)$ and a distinguisher $\mathcal{D}$ such that*

$$\Pr[\mathcal{A} \text{ wins in } F_1 \text{ model in } G] \leq \Pr[\mathcal{B} \text{ wins in } F_2 \text{ model in } G] + \mathsf{Adv}^{\mathsf{r\text{-}indiff}}_{F_1,F_2,S}(\mathcal{D}).$$

Moreover, $t_{\mathcal{B}_i} \leq t_{\mathcal{A}_i} + q_{\mathcal{A}_i} t_S, q_{\mathcal{B}_i} \leq q_{\mathcal{A}_i} q_S, t_A \leq m + t_G + \sum_{i=1}^m q_{G,i} t_{\mathcal{A}_i}, q_A \leq q_{G,0} + \sum_{i=1}^m q_{G,i} t_{\mathcal{A}_i}$ where $t_{\mathcal{A}}, t_{\mathcal{B}}, t_{\mathcal{D}}$ are the maximum running times of $\mathcal{A}, \mathcal{B}, \mathcal{D}$; $q_{\mathcal{A}}, q_{\mathcal{B}}$ are the maximum number of queries made by $\mathcal{A}$ and $\mathcal{B}$ in a single execution; and $q_{G,0}, q_{G,1}$ are the maximum number of queries made by G to the private interface and to the adversary.

Definitions of Hash Functions. We give the description of the ChopMD construction [7]. Let h be a compression function which maps a value of $d + n + s$ bits to a value of $n + s$ bits. The ChopMD $\text{ChopMD}^h : \{0,1\}^* \to \{0,1\}^n$ is defined in Fig. 3. $\mathsf{pad}_c : \{0,1\}^* \to (\{0,1\}^d)^*$ is an injective padding function such that its inverse is efficiently computable. IV is a constant value of $n + s$ bits.

We give the description of the FOLSponge construction [4]. Let P be a permutation of d bits. The FOLSonge $\text{FOLSponge}^P : \{0,1\}^* \to \{0,1\}^n$ is defined in Fig. 3 such that $n < d$.[4] Let $c = d - n$. $\mathsf{pad}_S : \{0,1\}^* \to (\{0,1\}^n)^*$ is an injective padding function such that the last n-bit value is not 0. IV is a constant value of d bits. $IV_1 = IV[1, n]$ and $IV_2 = IV[n+1, d]$. For example, $\mathsf{pad}_S(M) = M \| 1 \| 0^i$ where i is a smallest value such that $|M\|1\|0^i|$ is a multiple of n.

3 Reset Indifferentiability from $\mathcal{WRO}$

RSS [18] proved the impossibility of proving that the ChopMD and the FOLSponge are reset indifferentiable from random oracles. To compensate the impossibility, we change

[4] Note that if the output length (denoted by l) is smaller than n, the output length is achieved by returning $s[1, l]$.

the ideal world from a random oracle to a weakened random oracle ($\mathcal{WRO}$). We define $\mathcal{WRO}$ such that both of the ChopMD and the FOLSponge are reset indifferentiable from $\mathcal{WRO}$s.

We define $\mathcal{WRO}$ as $(\mathcal{RO}_n, \mathcal{RO}_v^*, \mathcal{RO}_w^\dagger, \mathcal{TO}, \mathsf{IC}_{a,b})$, where $\mathcal{RO}_n, \mathcal{RO}_v^*$, and $\mathcal{RO}_w^\dagger$ are arbitrary input length random oracles whose output lengths are n bit, v bit, and w bit, respectively, $\mathcal{TO}$ is a trace oracle, and $\mathsf{IC}_{a,b}$ is an ideal cipher with key length a and block length b. The definition of $\mathcal{TO}$ is that for query y to $\mathcal{TO}$, it returns M if $\exists_1 M$ such that a query M to $\mathcal{RO}_w^\dagger$ such that $y = \mathcal{RO}_w^\dagger(M)$ was made, and otherwise it returns $\perp$. Fig. 4 shows the method of implementing a $\mathcal{RO}_w^\dagger$ and a $\mathcal{TO}$. $E : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}^b$ denotes the encryption oracle of $\mathsf{IC}_{a,b}$, and $D : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}^b$ denotes the decryption oracle. The interfaces are defined by $\mathcal{WRO}.hon = \mathcal{RO}_n$ and $\mathcal{WRO}.adv = (\mathcal{RO}_n, \mathcal{RO}_v^*, \mathcal{RO}_w, \mathcal{TO}, \mathsf{IC}_{a,b})$. Note that the parameters (n, v, w, a, b) are defined in each hash function.

For a hash function $H^{\mathcal{U}}$ using an ideal primitive $\mathcal{U}$, the advantage of reset indifferentiability from $\mathcal{WRO}$ is defined as follows.

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff}}_{H^{\mathcal{U}},\mathcal{WRO},S}(\mathcal{D}) = |\Pr[\mathcal{D}^{H^{\mathcal{U}},\mathcal{U}} \Rightarrow 1] - \Pr[\mathcal{D}^{\mathcal{WRO}.hon, S^{\mathcal{WRO}.adv}} \Rightarrow 1]|.$$

The RSS theorem ensures that if $H^{\mathcal{U}}$ is reset indifferentiable from a $\mathcal{WRO}$, any security of any cryptosystem is preserved when a $\mathcal{WRO}$ is replaced by $H^{\mathcal{U}}$, where in the $\mathcal{WRO}$ model adversaries have access to $\mathcal{WRO}.adv$ and the cryptosystem has access to $\mathcal{WRO}.hon$, and for the $H^{\mathcal{U}}$ case, adversaries have access to $\mathcal{U}$ and the cryptosystem has access to $H^{\mathcal{U}}$.

3.1 Reset Indifferentiability for ChopMD

In this proof, we define the parameter of $\mathcal{WRO}$ as $w = s$ and $v = n + s$. Note that $\mathsf{IC}_{a,b}$ is not used. Therefore, $\mathcal{WRO} = (\mathcal{RO}_n, \mathcal{RO}_{n+s}^*, \mathcal{RO}_s^\dagger, \mathcal{TO})$.

Theorem 2. *Let the compression function h be a random oracle. There exists a stateless simulator S such that for any distinguisher $\mathcal{D}$,*

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff}}_{\mathrm{ChopMD}^h,\mathcal{WRO},S}(\mathcal{D}) \leq \frac{q_R(q_R-1) + 2\sigma(\sigma+1)}{2^s}$$

where $\mathcal{D}$ can make queries to left oracle $L = \mathrm{ChopMD}^h/\mathcal{RO}_n$ and right oracle $R = h/S$ at most q_L, q_R times, respectively, and l is a maximum number of blocks of a query to L. $\sigma = lq_L + q_R$. S makes at most $4q_R$ queries and runs in time $O(q_R)$. ♦

An intuition of this proof is shown in Subsection 1.5. This proof is given in Section 4.

3.2 Reset Indifferentiability for FOLSponge

We define the parameter of $\mathcal{WRO}$ as $w = c$ and $b = d$. We don't care the key size a, since $\mathsf{IC}_{a,b}$ can be regarded as random permutation by fixing a key k^*. We denote $E(k^*, \cdot)$ by a random permutation $\mathcal{P}(\cdot)$ of d bit and $D(k^*, \cdot)$ by its inverse oracle $\mathcal{P}^{-1}(\cdot)$. Note that in this proof, $\mathcal{RO}_v^*$ are not used. Therefore, $\mathcal{WRO} = (\mathcal{RO}_n, \mathcal{RO}_c^\dagger, \mathcal{TO}, \mathcal{P}, \mathcal{P}^{-1})$.

Theorem 3. *Assume that the underlying permutation P is a random permutation and P^{-1} is its inverse oracle. There exists a stateless simulator $S = (S_F, S_I)$ such that for any distinguisher $\mathcal{D}$,*

$$\mathsf{Adv}^{\text{r-indiff}}_{\text{FOLSponge}^P, \mathcal{WRO}, S}(\mathcal{D}) \leq \frac{2\sigma(\sigma+1)+q(q-1)}{2^c} + \frac{\sigma(\sigma-1)+q(q-1)}{2^{d+1}}$$

where $\mathcal{D}$ can make at most q_L, q_F and q_I queries to left $L = \text{FOLSponge}^P/\mathcal{RO}_n$ and right oracles $R_F = P/S_F, R_I = P^{-1}/S_I$. l is a maximum number of blocks of a query to L. $\sigma = lq_L + q_F + q_I$ and $q = q_F + q_I$. S makes at most $4q$ queries and runs in time $O(q)$. ♦

In the following, we outline why a stateless simulator can be constructed. To simplify the explanation, we omit the padding function of FOLSponge^P. Therefore, queries to L are in $(\{0,1\}^n)^*$. Since $\mathcal{D}$ interacts with (L, R_F, R_I), helpful information for $\mathcal{D}$ is obtained from these oracles. Thus, the S's tasks are to simulate the following two points.

- Simulation of P and P^{-1}: Since in the real world $R_F = P$ and $R_I = P^{-1}$, S must simulate P and P^{-1}.
- Simulation of L-R relation: Since there is a relation based on the FOLSponge construction among query-response values of L and of R_F in the real world, S must simulate such relation.

Using $\mathcal{WRO}$, we can construct a stateless simulator which succeeds in these simulations.

- Simulation of P and P^{-1}: S succeeds in this simulation by using $\mathcal{P}$ and $\mathcal{P}^{-1}$; S returns the response of $\mathcal{P}(x)$ for query x, and returns the response of $\mathcal{P}^{-1}(y)$ for query y.
- Simulation of L-R relation: S succeeds in this simulation by using $\mathcal{RO}_c^\dagger$ and $\mathcal{TO}$. We explain this simulation by using the following example.
  - $\mathcal{D}$ makes query X_1 $(:= IV \oplus (M_1 \| 0^c))$ to R_F and receives the response Y_1.
  - $\mathcal{D}$ makes query X_2 $(:= Y_1 \oplus (M_2 \| 0^c))$ to R_F and receives the response Y_2.

  In the real world, there are the relations $Y_1[1,n] = L(M_1)$ and $Y_2[1,n] = L(M_1 \| M_2)$. Then S succeeds in this simulation by the following procedures.
  - For query X_1 to S_F, S_F parses $X_1 = W_1 \| IV_2$, $M_1 = W_1 \oplus IV_1$, $Y_1^* := \mathcal{RO}_n(M_1)$, $Y_1' := \mathcal{RO}_c^\dagger(M_1)$ and $Y_1 = Y_1^* \| Y_1'$.
  - For query X_2 S_F parses $X_2 = W_2 \| Y_1'$, $M_1 = \mathcal{TO}(Y_1')$, $Y_1^* = \mathcal{RO}_n(M_1)$, $M_2 = W_2 \oplus Y_1^*$ and $Y_2 := \mathcal{RO}_n(M_1 \| M_2) \| \mathcal{RO}_c^\dagger(M_1 \| M_2)$.

  These procedures ensure that in the ideal world, $Y_1[1,n] = L(M_1)$ and $Y_2[1,n] = L(M_1 \| M_2)$.

As a result, we can construct a stateless simulator S which succeeds in the simulations of (P, P^{-1}) and of the L-R relation. Thus we can prove Theorem 3. The proof is given in the full version of this paper [15].

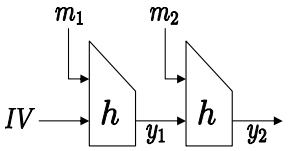

Fig. 5. Figure of Merkle-Damgård

$\underline{S(x\|m)}$ where $x_1 = x[1, s], x_2 = x[s+1, n]$ and $|m| = d$
$M \leftarrow \mathcal{TO}(x_1)$;
if $x = IV$ then $z \leftarrow \mathcal{RO}_n(m)$; $w \leftarrow \mathcal{RO}_s^{\dagger}(m)$;
else if $M \neq \perp$ and $x_2 \neq \mathcal{RO}_n(M)$ then $z \leftarrow \mathcal{RO}_n(M\|m)$; $w \leftarrow \mathcal{RO}_s^{\dagger}(M\|m)$;
else $w\|z \leftarrow \mathcal{RO}_{n+s}^{*}(x\|m)$;
return $w\|z$;

Fig. 6. Simulator S

4 Proof of Theorem 2

First we define a graph G_{MD}, which is initialized with a single node IV. Edges and nodes in this graph are defined by query-response values to R, which follow the MD structure. The nodes are chaining values and the edges are message blocks. For example, if $(IV, m_1, y_1), (y_1, m_2, y_2)$ are query response values of R, (IV, y_1, y_2) are the nodes of the graph and (m_1, m_2) are the edges. We denote the MD path by $IV \xrightarrow{m_1} y_1 \xrightarrow{m_2} y_2$ or $IV \xrightarrow{m_1\|m_2} y_2$ (Fig. 5 may help to understand this path).

To simplify this proof, we omit the padding function pad_c. Thus queries to L are in $(\{0,1\}^d)^*$. Note that ChopMD with pad_c is the special case of that without pad_c, thereby the security of ChopMD without pad_c ensures one with pad_c.

We define a stateless simulator S in Fig. 6. Step 4 ensures the simulation of h and Steps 2 and 3 ensure the simulation of the L-R relation.

Detail. In the following, for the simulator S in Fig. 6 and any distinguisher $\mathcal{D}$, we evaluate the bound of the reset indiferentiable advantage of ChopMD^h from $\mathcal{WRO}$. To evaluate the bound we consider the following five games. In each game, $\mathcal{D}$ has access to (L, R).

- Game 1 is the ideal world, that is, $(L, R) = (\mathcal{RO}_n, S)$.
- Game 2 is $(L, R) = (\mathcal{RO}_n, S_1)$, where S_1 keeps all query-response pairs. For a query $x\|m$ to S_1, if there is $(x\|m, w\|z)$ in the query response history, then S_1 returns $w\|z$, otherwise, S_1 returns the output of $S(x\|m)$.
- Game 3 is $(L, R) = (L_1, S_1)$, where for a query M to L_1 L_1 makes S_1 queries corresponding with $\mathrm{ChopMD}^{S_1}(M)$ and returns the response of $\mathcal{RO}_n(M)$.
- Game 4 is $(L, R) = (\mathrm{ChopMD}^{S_1}, S_1)$.
- Game 5 is the real world, that is, $(L, R) = (\mathrm{ChopMD}^h, h)$.

Let G_i be an event that $\mathcal{D}$ outputs 1 in Game i. We thus have that

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff}}_{\mathrm{ChopMD}^h,\mathcal{WRO},S}(\mathcal{D}) \leq \sum_{i=1}^{4} |\Pr[G_i] - \Pr[G_{i+1}]| \leq \frac{q_R(q_R-1)+2\sigma(\sigma+1)}{2^s}.$$

In the following, we justify the above bound by evaluating each difference.

Game 1 ⇒ Game 2. From Game 1 to Game 2, we change R from S to S_1 where S_1 records query response values, while S does not record them. The query-response history ensures that in Game 2 if a query $x\|m$ to S_1 was made and y was responded, for the repeated query $x\|m$ to S_1 the same value y is responded, while in Game 1 there is a case that for some repeated query $x\|m$ to S_1 where y was responded, a distinct value y^* ($\neq y$) is responded. The difference $|\Pr[G_1] - \Pr[G_2]|$ is thus bounded by the probability that in Game 1 the distinct value is responded. We call the event "**Diff**". Since the procedure of S to define outputs is controlled by $\mathcal{TO}$ (See the steps 2-4), the event **Diff** relies on outputs of $\mathcal{TO}$. Thus, if **Diff** occurs, for some repeated query to $\mathcal{TO}$ the distinct value is responded. More precisely, if **Diff** occurs, the following event occurs.

– For a query y to $\mathcal{TO}$, w was responded, and then for the repeated query a different value w^* is responded. From the definition of $\mathcal{TO}$, there are two cases for (w, w^*); $\mathbf{Diff}_1$: $w = \perp$ and $w^* \neq \perp$, $\mathbf{Diff}_2$: $w \neq \perp$ and $w^* = \perp$.

We thus have that

$$|\Pr[G_1] - \Pr[G_2]| \leq \Pr[\mathbf{Diff}_1] + \Pr[\mathbf{Diff}_2] \leq \frac{q_R(q_R-1)}{2^s}.$$

We justify the bound as follows.

First we bound the probability of $\Pr[\mathbf{Diff}_1]$. Since the response w of the first query is $\perp$, when the first query is made, the query w^* to $\mathcal{RO}^\dagger_s$ such that $y = \mathcal{RO}^\dagger_s(w^*)$ was not made. Since the response w^* of the repeated query is not $\perp$, when the repeated query is made, the query w^* to $\mathcal{RO}^\dagger_s$ was made such that $y = \mathcal{RO}^\dagger_s(w^*)$. Therefore, first y is defined. Second, the output of $\mathcal{RO}^\dagger_s(w^*)$ is defined. Thus, $\Pr[\mathbf{Diff}_1]$ is bounded by the probability that the response of $\mathcal{RO}^\dagger_s(w^*)$, which is an s-bit random value, collides with the value y. Since the numbers of queries to $\mathcal{RO}^\dagger_s$ and $\mathcal{TO}$ are at most q_R times, respectively, we have that

$$\Pr[\mathbf{Diff}_1] \leq \sum_{i=1}^{q_R} \frac{i-1}{2^s} \leq \frac{q_R(q_R-1)}{2^{s+1}}.$$

Next we bound the probability of $\Pr[\mathbf{Diff}_2]$. Since the response w of the first query is not $\perp$, when the first query is made, the query w to $\mathcal{RO}^\dagger_s$ was made such that $y = \mathcal{RO}^\dagger_s(w)$. Since the response w^* of the repeated query is $\perp$, when the repeated query is made, a query w' to $\mathcal{RO}^\dagger_s$ was made such that $w \neq w'$ and $\mathcal{RO}^\dagger_s(w) = \mathcal{RO}^\dagger_s(w')$. Therefore, $\Pr[\mathbf{Diff}_2]$ is bounded by the collision probability of $\mathcal{RO}^\dagger_s$. We thus have that

$$\Pr[\mathbf{Diff}_2] \leq \sum_{i=1}^{q_R} \frac{i-1}{2^s} \leq \frac{q_R(q_R-1)}{2^{s+1}}.$$

Game 2 ⇒ Game 3. From Game 2 to Game 3, we change L from $\mathcal{RO}_n$ to L_1 where in Game 3 L makes additional queries to R corresponding with the calculation of $\mathrm{ChopMD}^{S_1}(M)$. Note that $\mathcal{D}$ cannot directly observe the additional query response values but can observe those by making the queries to R. So we have to show that in Game 3 the additional queries by L don't affect $\mathcal{D}$'s behavior. We ensure this by Lemma 1 where in Game j, for any MD path $IV \xrightarrow{M} z$, $z = \mathcal{RO}_s^{\dagger}(M)||\mathcal{RO}_n(M)$ unless Bad_j occurs. By Lemma 1, in both games, unless the bad event occurs, all responses to R are defined by the same queries to $\mathcal{RO}_s^{\dagger}$ and to $\mathcal{RO}_n$. Namely, in Game 3, the responses of the additional queries to R which $\mathcal{D}$ observes are chosen from the same distribution as in Game 2 unless the bad event occurs. Thus, the difference $|\Pr[G_2] - \Pr[G_3]|$ is bounded by the probability of occurring the bad event.

First we define the bad event. Let T_i be a list which records $(x_t[1,s], y_t[1,s])$ for $t = 1, \ldots, i-1$ where $(x_t||m_t, y_t)$ is a t-th query response pair of S where $y_t = S(x_t||m_t)$.

- Bad_j is that in Game j for some i-th query $x_i||m_i$ to S, the response y_i is such that $y_i[1,s]$ collides with some value in $T_i \cup \{x_i[1,s]\} \cup \{IV[1,s]\}$.

Note that since all outputs of S_1 are defined by using S, we deal with S instead of S_1.

Next we give Lemma 1 as follows. Note that Lemma 1 is also used when evaluating the difference between Game 3 and Game 4.

Lemma 1. *In Game j, for any MD path* $IV \xrightarrow{M} y$ $y = \mathcal{RO}_s^{\dagger}(M)||\mathcal{RO}_n(M)$ *unless* Bad_j *occurs.* ♦

Proof of Lemma 1. Assume that Bad_j does not occur. We show that for any MD path $IV \xrightarrow{M} y$, $y = \mathcal{RO}_s^{\dagger}(M)||\mathcal{RO}_n(M)$. Let $(x_1||m_1, y_1), \ldots, (x_t||m_t, y_t)$ be query response pairs to S which correspond with the MD path where $x_1 = IV$, $x_i = y_{i-1}$ $(i = 2, \ldots, t)$, $y_t = y$, and $M = m_1||\ldots||m_t$.

When $t = 1$, $y = \mathcal{RO}_s^{\dagger}(M)||\mathcal{RO}_n(M)$ (see Step 2).

We consider the case that $t \geq 2$.

Since Bad_j does not occur, the following case does not occur; for some $i \in \{1, \ldots, t-1\}$, $(x_i||m_i, y_i)$ is defined after $(x_{i+1}||m_{i+1}, y_{i+1})$ was defined. So $(x_1||m_1, y_1), \ldots, (x_t||m_t, y_t)$ are defined by this order.

Since Bad_j does not occur, no collision of outputs of $\mathcal{RO}_s^{\dagger}$ occurs. Therefore, when the query $S_1(x_t||m_t)$ is made, the pair $(m_1||\ldots||m_{j-1}, y_{t-1})$ has been recorded in the table $\mathsf{F}^{\dagger}$ of $\mathcal{RO}_s^{\dagger}$, that is, $\mathsf{F}^{\dagger}[m_1||\ldots||m_{t-1}] = y_{t-1} = x_t$.

Since Bad_j does not occur, no collision of outputs of $\mathcal{RO}_s^{\dagger}$ occurs. Therefore, there is no value M^* such that $M^* \neq m_1||\ldots||m_{t-1}$ and $\mathsf{F}^{\dagger}[M^*] = x_t$.

Thus, for the query $x_t||m_t$ to S, S makes the query $x_t[1,s]$ to $\mathcal{TO}$, receives $m_1||\ldots||m_{t-1}$ (Step 1), and returns the response y_t such that $y_t = \mathcal{RO}_s^{\dagger}(M)||\mathcal{RO}_n(M)$ (Step 3). □

By Lemma 1, the difference $|\Pr[G_2] - \Pr[G_3]|$ is bounded by

$$\max\{\Pr[Bad_2], \Pr[Bad_3]\} \leq \frac{\sigma(\sigma+1)}{2^s}.$$

Finally we justify the bound. The left s-bit values of all outputs of S_1 are uniformly chosen at random from $\{0,1\}^s$. The probability of occurring the bad event is that for

some i-th query to S the left s-bit value of the response, which is a random value, hits some of $T_i \cup \{x_i[1,s]\} \cup \{IV[1,s]\}$. We thus have

$$\Pr[Bad_2] \leq \sum_{i=1}^{q_R} \frac{2(i-1)+2}{2^s} = \frac{q_R(q_R+1)}{2^s},\ \Pr[Bad_3] \leq \sum_{i=1}^{\sigma} \frac{2(i-1)+2}{2^s} = \frac{\sigma(\sigma+1)}{2^s}$$

where S_1 is called at most q_R times in Game 2 and σ times in Game 3.

Game 3 ⇒ Game 4. From Game 3 to Game 4, we change L where in Game 3 $L(M) = \mathcal{RO}_n(M)$, while in Game 4 $L(M) = \mathrm{ChopMD}^{S_1}(M)$. Therefore, the modification does not change $\mathcal{D}$'s behavior iff in Game 4 $\mathrm{ChopMD}^{S_1}(M) = \mathcal{RO}_n(M)$. Since Lemma 1 ensures that for any MD path $IV \xrightarrow{M} z$, $z = \mathcal{RO}_s^{\dagger}(M) \| \mathcal{RO}_n(M)$ unless the bad event Bad_4 occurs, the modification does not change $\mathcal{D}$'s behavior. Thus the difference $|\Pr[G_3] - \Pr[G_4]|$ is bounded by the probability of occurring Bad_4. Since S_1 is called at most σ times, we have

$$|\Pr[G_3] - \Pr[G_4]| \leq \Pr[Bad_4] \leq \frac{\sigma(\sigma+1)}{2^s}.$$

Game 4 ⇒ Game 5. From Game 4 to Game 5, we change R from S_1 to h. Since outputs of S_1 are uniformly chosen at random from $\{0,1\}^{n+s}$, the modification of R does not affect $\mathcal{D}$'s behavior. We thus have that $\Pr[G_4] = \Pr[G_5]$. □

5 Multi-Stage Security in the $\mathcal{WRO}$ Model

In this section, we show appropriateness of our $\mathcal{WRO}$ methodology. We construct a (non-adaptive) CDA secure [2] PKE scheme in the $\mathcal{WRO}$ model. Specifically, we show that if a PKE scheme satisfies an weak security (i.e., IND-SIM security [18]) in the $\mathcal{RO}$ model, then it is also CDA secure in the $\mathcal{WRO}$ model.

An IND-SIM secure PKE in the $\mathcal{RO}$ model is easily obtained by applying a known technique [18] that any CPA secure PKE scheme can be converted into IND-SIM secure by using EwH [1] and REwH1 [2] in the $\mathcal{RO}$ model. Therefore, our result implies that a very large class of PKE schemes is CDA secure in the $\mathcal{WRO}$ model (e.g., factoring-based, Diffie-Hellman-based, lattice-based, etc.).

Furthermore, our result in Section 3 guarantees to instantiate $\mathcal{WRO}$ by ChopMD or FOLSponge. Hence, finally, we have that any CPA secure PKE in the $\mathcal{RO}$ model can be converted into CDA secure with ChopMD or FOLSponge. While the previous work [18] showed CDA secure PKE schemes only with the specific NMAC hash function, our work achieves CDA secure PKE schemes with large class of hash functions (i.e., ChopMD and FOLSponge).

5.1 CDA Secure PKE in the $\mathcal{WRO}$ Model

Public Key Encryption (PKE). A public key encryption scheme $\mathcal{AE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ consists of three algorithms. Key generation algorithm Gen outputs public key

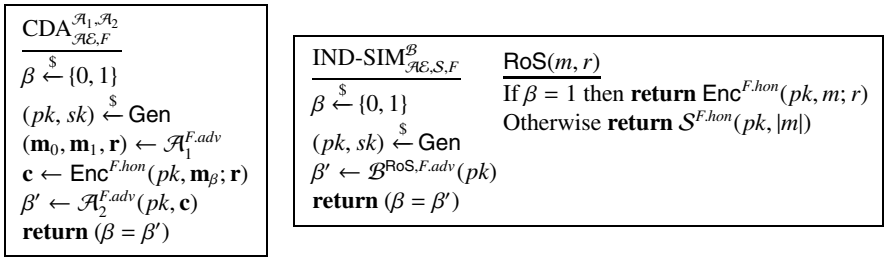

Fig. 7. CDA game and IND-SIM game

pk and secret key sk. Encryption algorithm Enc takes public key pk, plaintext m, and randomness r, and outputs ciphertext c. Decryption algorithm Dec takes secret key sk and ciphertext c, and outputs plaintext m or distinguished symbol $\perp$. For vectors $\mathbf{m}, \mathbf{r}$ with $|\mathbf{m}| = |\mathbf{r}| = l$ which is the size of vectors, we denote by $\mathsf{Enc}(pk, \mathbf{m}; \mathbf{r})$ the vector $(\mathsf{Enc}(pk, \mathbf{m}[1]; \mathbf{r}[1]), \ldots, \mathsf{Enc}(pk, \mathbf{m}[l]; \mathbf{r}[l]))$. We say that $\mathcal{AE}$ is deterministic if Enc is deterministic.

CDA Security. We explain the CDA security (we quote the explanation of the CDA security in [18]). Fig. 7 illustrates the non-adaptive CDA game for a PKE scheme $\mathcal{AE}$ using a functionality F. This notion captures the security of a PKE scheme when randomness $\mathbf{r}$ used in encryption may not be a string of uniform bits. For the remainder of this section, fix a randomness length $\rho \geq 0$ and a plaintext length $\omega > 0$. An (μ, ν)-mmr-source $\mathcal{M}$ is a randomized algorithm that outputs a triple of vector $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ such that $|\mathbf{m}_0| = |\mathbf{m}_1| = |\mathbf{r}| = \nu$, all components of $\mathbf{m}_0$ and $\mathbf{m}_1$ are bit strings of length ω, all components of $\mathbf{r}$ are bit strings of length ρ, and $(\mathbf{m}_\beta[i], \mathbf{r}[i]) \neq (\mathbf{m}_\beta[j], \mathbf{r}[j])$ for all $1 \leq i < j \leq \nu$ and all $\beta \in \{0, 1\}$. Moreover, the source has min-entropy μ, meaning $\Pr[(\mathbf{m}_\beta[i], \mathbf{r}[i]) = (m', r') | (\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow \mathcal{M}] \leq 2^{-\mu}$ for all $\beta \in \{0, 1\}$, all $1 \leq i \leq \nu$, and all (m', r'). A CDA adversary $\mathcal{A}_1, \mathcal{A}_2$ is a pair of procedures, the first of which is a (μ, ν)-mmr-source. The CDA advantage for a CDA adversary $\mathcal{A}_1, \mathcal{A}_2$ against scheme $\mathcal{AE}$ using a functionality F is defined by

$$\mathsf{Adv}^{\mathrm{cda}}_{\mathcal{AE},F}(\mathcal{A}_1, \mathcal{A}_2) = 2 \cdot \Pr[\mathrm{CDA}^{\mathcal{A}_1,\mathcal{A}_2}_{\mathcal{AE},F} \Rightarrow \mathsf{true}] - 1.$$

As noted in [2], in the RO model, mmr-sources have access to the RO. In this setting, the min-entropy requirement is independent of the coins used by the RO, meaning the bound must hold for any fixed choice of function as the RO. If this condition is removed, one can easily break the CDA security (i.e., $\mathcal{A}_1$ and $\mathcal{A}_2$ can easily share the messages $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{r})$) for any cryptosystem using any indifferentiable hash function.

IND-SIM Security. The IND-SIM security is a special notion for PKE schemes. It captures that an adversary cannot distinguish outputs from the encryption algorithm and from a simulator $\mathcal{S}$ even if the adversary can choose plaintext and randomness. Fig. 7 shows the IND-SIM game. We define the IND-SIM advantage of an adversary $\mathcal{B}$ by

$$\mathsf{Adv}^{\text{ind-sim}}_{\mathcal{AE},\mathcal{S},F}(\mathcal{B}) = 2 \cdot \Pr[\text{IND-SIM}^{\mathcal{B}}_{\mathcal{AE},F} \Rightarrow \mathsf{true}] - 1.$$

As noted in [18], in the standard model this security goal is not achievable because $\mathcal{AE}$ uses no randomness beyond that input. In the RO model, we will use it when the adversary does not make any RO queries. A variety of PKE schemes is shown to satisfy IND-SIM security in the RO model.

CDA Security in the $\mathcal{WRO}$ Model. The following theorem shows that for any PKE scheme the non-adaptive CDA security in the $\mathcal{WRO}$ model is obtained from IND-SIM security in the RO model.

Theorem 4. *Let $\mathcal{AE}$ be a PKE scheme. Let $(\mathcal{A}_1, \mathcal{A}_2)$ be a CDA adversary in the $\mathcal{WRO}$ model making at most $q_{\mathcal{RO}}, q_{\mathcal{RO}^*}, q_{\mathcal{RO}^\dagger}, q_{\mathcal{TO}}, q_E, q_D$ queries to $\mathcal{RO}_n, \mathcal{RO}_v^*, \mathcal{RO}_w^\dagger, \mathcal{TO}, \mathsf{IC}_{a,b}$ = (E, D). For any simulator $\mathcal{S}$ there exists an IND-SIM adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathrm{cda}}_{\mathcal{AE},\mathcal{WRO}}(\mathcal{A}_1,\mathcal{A}_2) \le \mathsf{Adv}^{\text{ind-sim}}_{\mathcal{AE},\mathcal{S},\mathcal{RO}_n}(\mathcal{B}) + q_{\mathcal{RO}} \cdot \mathsf{maxpk}_{\mathcal{AE}} + \frac{q_{\mathcal{RO}} + 4q_{\mathcal{RO}^*}^2}{2^\mu}$$
$$+ max\left\{\frac{4q_{\mathcal{RO}^\dagger}^2 + 4q_{\mathcal{TO}}^2}{2^\mu}, \frac{q_{\mathcal{TO}}}{2^{w-\log q_{\mathcal{RO}^\dagger}}}\right\} + max\left\{\frac{4q_E^2 + 4q_D^2}{2^\mu}, \frac{q_D}{2^{b-\log q_E}}, \frac{q_E}{2^{b-\log q_D}}\right\}.$$

$\mathcal{B}$ makes no RO queries, makes ν RoS-queries, and runs in time that of $(\mathcal{A}_1, \mathcal{A}_2)$ plus $O(q_{\mathcal{RO}} + q_{\mathcal{RO}^*} + q_{\mathcal{RO}^\dagger} + q_{\mathcal{TO}} + q_E + q_D)$. $\mathsf{maxpk}_{\mathcal{AE}}$ is the maximum public key collision probability defined as $\mathsf{maxpk}_{\mathcal{AE}} = \max_{\gamma\in\{0,1\}^*} \Pr[pk = \gamma : (pk, sk) \xleftarrow{\$} \mathsf{Gen}]$. μ is min-entropy of the mmr-source. ♦

The proof outline is as follows: First, we start with game $\mathbf{G}_0$ which is exactly the same game as the CDA game in the $\mathcal{WRO}$ model. Secondly, we transform $\mathbf{G}_0$ to game $\mathbf{G}_1$ so that $\mathcal{RO}_n$ returns a random value when $\mathcal{A}_1$ poses a message that is posed to $\mathcal{RO}_n$ by Enc to generate the challenge ciphertext. In game $\mathbf{G}_1$, outputs of $\mathcal{RO}_n$ does not contain any information about computations to generate the challenge ciphertext for $\mathcal{A}_1$. Thirdly, we transform $\mathbf{G}_1$ to game $\mathbf{G}_2$ so that the table of inputs and outputs of each oracle in $\mathcal{WRO}$ (except $\mathcal{RO}_n$) for $\mathcal{A}_1$ is independent of the table for $\mathcal{A}_2$ according to the output of $\mathcal{A}_1$. In game $\mathbf{G}_2$, queries to sub-oracles for $\mathcal{A}_2$ does not contain any information about the output of $\mathcal{A}_1$, and $\mathcal{A}_1$ cannot hand over any information to $\mathcal{A}_2$ with sub-oracles. Fourthly, we transform $\mathbf{G}_2$ to game $\mathbf{G}_3$ so that ciphertext $\mathbf{c}$ is generated from a simulator $\mathcal{S}$ in the IND-SIM game. In game $\mathbf{G}_3$, ciphertext $\mathbf{c}$ does not contain any information about outputs of $\mathcal{A}_1$. Thus, $\mathcal{A}_1$ cannot hand over any information to $\mathcal{A}_2$ with $\mathbf{c}$. Finally, we transform $\mathbf{G}_3$ to game $\mathbf{G}_4$ so that $\mathcal{RO}_n$ returns a random value when $\mathcal{A}_2$ poses a message that is posed to $\mathcal{RO}_n$ by Enc to generate the challenge ciphertext. In game $\mathbf{G}_4$, outputs of $\mathcal{RO}_n$ does not contain any information about computations to generate the challenge ciphertext for $\mathcal{A}_2$. Thus, the advantage of $\mathcal{A}_2$ in $\mathbf{G}_4$ is nothing.

The proof of Theorem 4 is shown in the full version of this paper [15].

References

1. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and Efficiently Searchable Encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)

2. Bellare, M., Brakerski, Z., Naor, M., Ristenpart, T., Segev, G., Shacham, H., Yilek, S.: Hedged public-key encryption: How to protect against bad randomness. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 232–249. Springer, Heidelberg (2009)
3. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008)
4. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the Indifferentiability of the Sponge Construction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer, Heidelberg (2008)
5. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak SHA-3 submission. Submission to NIST, Round 3 (2011)
6. Boldyreva, A., Fehr, S., O'Neill, A.: On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
7. 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)
8. Demay, G., Gaži, P., Hirt, M., Maurer, U.: Resource-restricted indifferentiability. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 664–683. Springer, Heidelberg (2013)
9. 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); Full Version in ePrint 2009/177
10. Fuller, B., O'Neill, A., Reyzin, L.: A Unified Approach to Deterministic Encryption: New Constructions and a Connection to Computational Entropy. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 582–599. Springer, Heidelberg (2012)
11. Luykx, A., Andreeva, E., Mennink, B., Preneel, B.: Impossibility results for indifferentiability with resets. ePrint 2012/644 (2012)
12. Maurer, U.M., Renner, R.S., 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)
13. Mironov, I., Pandey, O., Reingold, O., Segev, G.: Incremental Deterministic Public-Key Encryption. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 628–644. Springer, Heidelberg (2012); Full Version in ePrint 2012/047
14. Mittelbach, A.: Salvaging indifferentiability in a multi-stage setting. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 603–621. Springer, Heidelberg (2014)
15. Naito, Y., Yoneyama, K., Ohta, K.: Reset Indifferentiability from Weakened Random Oracle Salvages One-pass Hash Functions. In: ePrint 2012/014 (2012); Full Version of this Paper
16. National Institute of Standards and Technology. Cryptographic Hash Algorithm Competition. http://csrc.nist.gov/groups/ST/hash/sha-3/winner_sha-3.html
17. National Institute of Standards and Technoloty. FIPS PUB 180-4 Secure Hash Standard. In: FIPS PUB (2012)
18. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with Composition: Limitations of the Indifferentiability Framework. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 487–506. Springer, Heidelberg (2011); Full Version: ePrint 2011/339