On Indifferentiable Hash Functions in Multi-Stage Security Games

Yusuke Naito and Kazuki Yoneyama

Mitsubishi Electric Corporation and NTT Corporation

Abstract. It had been widely believed that the indifferentiability framework ensures composition in any security game. However, Ristenpart, Shacham, and Shrimpton (EUROCRYPT 2011) demonstrated that for some multi-stage security, there exists a cryptosystem which is secure in the random oracle (RO) model but is broken when some indifferentiable hash function is used. However, this does not imply that for any multi-stage security, any cryptosystem is broken when a RO is replaced with the indifferentiable hash function. They showed that the important multi-stage security, the chosen-distribution attack (CDA) security, is preserved for some public key encryption (PKE) schemes when a RO is replaced with the indifferentiable hash function proposed by Dodis, Ristenpart, and Shrimpton (EUROCRYPT 2009). An open problem from their result is how to obtain the multi-stage security when a RO is replaced with other indifferentiable hash functions. In this paper, we positively solve this problem so that for some PKE scheme the CDA security is obtained even when the RO is replaced with important indifferentiable hash functions, Prefix-free Merkle-Damgård, chop Merkle-Damgård, or Sponge. First, we introduce a new weakened RO model, called Versatile Oracle ($\mathcal{VO}$) model, as a tool for bridging the multi-stage security and such hash functions. We prove *reset* indifferentiablity of these hash functions from a $\mathcal{VO}$; thus, if a cryptosystem is secure in the $\mathcal{VO}$ model, then it is also secure when instantiating the $\mathcal{VO}$ by these hash functions. Next, we show that if a cryptosystem satisfies an weak property, the IND-SIM security, in the RO model, then it is also CDA secure in the $\mathcal{VO}$ model. Combining these two results, we have that for PKE schemes satisfiying the IND-SIM security in the RO model the CDA security is guaranteed when the RO is replaced with a large class of practical hash functions.

Keywords. Indifferentiable Hash Function, Reset Indifferentiable Security, Multi-Stage Security

1 Introduction

The indifferentiable composition theorem of Maurer, Renner, and Holenstein [22] ensures that if a functionality F (e.g., a hash function from an ideal primitive) is indifferentiable from a second functionality F' (e.g., a random oracle (RO)), the security of any cryptosystem is preserved when F' is replaced with F. The important application of this framework is the RO model security, because many practical cryptosystems e.g., RSA-OAEP [8] and RSA-PSS [9] are designed by the RO methodology. A RO is instantiated by a hash function such as SHA-1 and SHA-256 [26]. However, the Merkle-Damgård hash functions [16, 23] such as SHA-1 and SHA-256, are not indifferentiable from ROs [15]. So many indifferentiable (from a RO) hash functions have been proposed, e.g., the finalists of the SHA-3 competition [3, 11, 18, 20, 28, 1, 2, 10, 12, 15, 14, 17]. The indifferentiable security is thus an important security of hash functions.

Recently, Ristenpart, Shacham, and Shrimpton [27] showed that in some multi-stage security game some scheme secure in the RO model is broken when some indifferentiable hash function is used. They considered the multi-stage security game called CRP. The CRP security game for the n-bit (output length) hash function H is the two stage security game. In the first stage, for random messages M_1, M_2 of $2n$ bits, the first stage adversary A_1 derives the some state st of $2n$ bits. In the second stage, the second stage adversary A_2 receives st, and for a random challenge value C of $2n$ bits outputs an n-bit value z. Then, the adversary wins if $z = H(M_1||M_2||C)$. Consider the chop MD hash function $\mathrm{chopMD}^h(M_1||M_2||C) = chop_n(h(h(h(IV, M_1), M_2), C))$ which is indifferentiable from a RO [15], where $h : \{0,1\}^{4n} \to \{0,1\}^{2n}$ is a RO, and $chop_n : \{0,1\}^{2n} \to \{0,1\}^n$ outputs the right n bits of the input. Clearly, the following adversary can win with probability 1 when H is the chop MD hash function. First, A_1 receives M_1, M_2, calculates $st = h(h(IV, M_1), M_2)$, and outputs st. Second, A_2 receives st, and for a random challenge C, outputs $z = chop_n(h(st, C))$ which is equal to $\mathrm{chopMD}^h(M_1||M_2||C)$. On the other hand, when H is a RO, since A_2 cannot receive several value of M_1, M_2, the probability that the adversary wins is negligible. This result implies that the indifferentiable composition theorem does not ensure any multi-stage security when a RO is replaced with indifferentiable hash functions.

The chosen-distribution attack (CDA) security game is an important multi-stage security game, which is the security goal for deterministic, efficiently searchable [4, 6, 13, 19, 24], and hedged [5] public key encryption (PKE), wherein there are several PKE schemes which are proven in the RO model [4, 5]. For the CDA secure PKE schemes EwH [4] and REwH1 [5] (in the RO model), Ristenpart *et al.* salvaged the important indifferentiable hash function, the NMAC-type hash function [17], which was proposed by Dodis, Ristenpart, and Shrimpton, and which is employed in the SHA-3 finalist Skein [18]. They showed that these PKE schemes are non-adaptive CDA secure in the chosen-plaintext attack (CPA) case when the NMAC-type hash function is used.

The open problem from the paper of Ristenpart *et al.* is thus the CDA security when a RO replaced with other indifferentiable hash functions. Especially, it is important to consider the security when a RO is replaced with the SHA-3 finalists and the SHA-2 hash functions, because one of the SHA-3 finalists will be published as a standard hash function (FIPS) [25] and the SHA-2 hash functions were published as standard hash functions [26]. We consider the important hash functions, Prefix-free Merkle-Damgård (PFMD) [15], Sponge [10] and chop Merkle-Damgård (chop MD) [15]. The PFMD hash function is employed in the SHA-3 finalist BLAKE [3]. The Sponge hash function is employed in the SHA-3 finalist Keccak [11]. The chop Merkle-Damgård hash function is employed in SHA-224 and SHA-384 [26]. We show that for the same class of PKE schemes as the Ristenpart *et al.*'s result, the CDA security is guaranteed when a RO is replaced with these indifferentiable hash functions. The above result covers the non-adaptive security in the CPA case (i.e., the same setting as [27]). The advantages of our result to the result of Ristenpart *et al.* are that our result can salvage other types of practical hash functions to obtain the CDA security.

(Reset) Indifferentiability [27]. To prove the CDA security, we use the reset indifferentiability framework of Ristenpart *et al.* The reset indifferentiability ensures composition in any security game: if a hash function H^P which uses an ideal primitive P is reset indifferentiable from another ideal primitive P', any security of any cryptosystem is preserved when P' is replaced with H^P.

Recall the original [22] and reset [27] indifferentiability (from a RO) frameworks. The original indifferentiable security game from a RO for H^P is that a distinguisher A converses either with (H^P, P) or (RO, S^{RO}). S is a simulator which simulates P such that S and P' are consistent. If the probability that the distinguisher A hits the conversing world is small, then H^P is indifferentiable from a RO. In the reset indifferentiable security game, the distinguisher can reset the initial state of the simulator at arbitrary times.

To prove the original indifferentiable security, the simulator needs to record the query-response history. When for a query $P(x)$ z was returned, for a repeated query $P(x)$, z is returned. So, when for a query $S(x)$ z was returned, for a repeated query $S(x)$, the simulator should return z. When the internal state is reseted, the simulator forgets the value and cannot return. Thus one cannot use the reset indifferentiability from a RO to prove the CDA security when a RO is replaced with the indifferentiable hash function.

Our Approach. First, we thus use the reset indifferentiability from a variant of a RO. We propose an weakened variant which covers many indifferentiable hash functions. We call the variant "Versatile Oracle" ($\mathcal{VO}$). The $\mathcal{VO}$ consists of a RO and auxiliary oracles. The auxiliary oracles are used to record the query-response history of a simulator. The $\mathcal{VO}$ thus enables to construct a simulator which does not update the internal state and which is unaffected by the reset function. We show that the PFMD hash function, the chop MD hash function [1], and the Sponge hash function are reset indifferentiable from $\mathcal{VO}$s.

Next, we show that some PKE scheme satisfies the CDA security in the $\mathcal{VO}$ model. The reset indifferentiability composition theorem ensures that the CDA security are preserved when a $\mathcal{VO}$ is replaced with the indifferentiable hash function (i.e., PFMD, Chop MD, and Sponge). This is a positive result for applicability of the reset indifferentiability (from a $\mathcal{VO}$). We note that since $\mathcal{VO}$ is the weaker oracle than RO, $\mathcal{VO}$ cannot cover all applications of RO. However, we can show that $\mathcal{VO}$ covers the CDA security for PKE schemes satisfying an weak property, called the IND-SIM security. Thus, PKE schemes which are proved to be IND-SIM secure such as EwH [4] and REwH1 [5] are also CDA secure in the $\mathcal{VO}$ model.

[1] Recently, Andreeva *et al.* [1] and Chang *et al.* [14] consider the indifferentiable security of the BLAKE hash function with the more concrete structure than PFMD. Similarly, one can prove that the BLAKE hash function is reset indifferentiable from a $\mathcal{VO}$.

Again, our goal is to prove the CDA security when a RO is replaced with the indifferentiable hash function. Though these hash functions are not reset indifferentiable from ROs (one cannot directly prove from the RO model security by the reset indifferentiability), our first result ensures that these hash functions are reset indifferentiable from $\mathcal{VO}$s. Therefore, we have that some PKE scheme is CDA secure when a RO is replaced with practical hash functions.

2 Preliminaries

Notation. For two values x, y, $x||y$ is the concatenated value of x and y. For some value y, $x \leftarrow y$ means assigning y to x. When X is a non-empty finite set, we write $x \stackrel{\$}{\leftarrow} X$ 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. For sets A and C, $C \stackrel{\cup}{\leftarrow} A$ means assign $A \cup C$ to C. For $l \times r$-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 formula F, if there exists just a value M such that $F(M)$ is true, we denote $\exists_1 M$ s.t. $F(M)$. 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 $1 \leq i \leq |\mathbf{x}|$. $bit_j(\mathbf{x})$ is the left j-th bit of $\mathbf{x}[1]|| \ldots ||\mathbf{x}[|\mathbf{x}|]$.

(Reset) Indifferentiability [22, 27]. In the reset indifferentiability [27], for a functionality F, a private interface $F.priv$ and a public interface $F.pub$ are considered, where adversaries have oracle access to $F.pub$ and other parties (honest parties) have oracle access to $F.priv$. For example, for a cryptosystem in the F model, an output of the cryptosystem is calculated by accessing $F.priv$ and an adversary has oracle access to $F.pub$. In the RO model the RO has both interfaces. Let H^P be a hash function that utilizes an ideal primitive P. The interfaces of H^P are defined by $H^P.priv = H^P$ and $H^P.pub = P$.

For two functionalities F_1 (e.g., hash function) and F_2 (e.g. a variant of a RO), the advantage of the reset indifferentiability for F_1 from F_2 is as follows.

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff},F_2}_{F_1,S}(A) = |\Pr[A^{\bar{F}_1.priv,\bar{F}_1.pub} \Rightarrow 1] - \Pr[A^{F_2.priv,\hat{S}^{F_2.pub}} \Rightarrow 1]|$$

where $\hat{S} = (S, S.Rst)$, $\bar{F}_1.priv = F_1.priv$ and $\bar{F}_1.pub = (F_1.pub, nop)$. $S.Rst$ takes no input and when run reinitializes all of S. nop takes no input and does nothing. We say F_1 is reset indifferentiable from F_2 if there exists a simulator S such that for any distinguisher A the advantage of the reset indifferentiability is negligible. This framework ensures that if F_1 is reset indifferentiable from F_2 then any stage security of any cryptosystem is preserved when F_2 is replaced with F_1. Please see Theorem 6.1 in the full version of [27].

When $S.Rst$ and nop are removed from the reset indifferentiable security game, it is equal to the original indifferentiable security game [22]. In the original indifferentiable security game, the distinguisher interacts with $(F_1.priv, F_1.pub)$ and $(F_2.priv, S^{F_2.pub})$. We denote the advantage of the indifferentiable security by $\mathsf{Adv}^{\mathsf{indif}}_{F_1,F_2,S}(A)$ for a distinguisher A. We say F_1 is indifferentiable from F_2 if there exists a simulator S such that for any distinguisher A the advantage is negligible.

3 Versatile Oracle

In this section, we propose a versatile oracle $\mathcal{VO}$. $\mathcal{VO}$ consists of a RO $\mathcal{RO}_n$, a RO $\mathcal{RO}^*_v$, a traceable random oracle $\mathcal{TRO}_w$, and ideal ciphers $\mathsf{IC}_{a,b}$. The private interface is defined by $\mathcal{VO}.priv = \mathcal{RO}_n$ and the public interface is defined by $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{RO}^*_v, \mathcal{TRO}_w, \mathsf{IC}_{a,b})$. $\mathcal{VO}$ can be implemented as Fig. 1.

$\mathcal{RO}_n$ is shown in Fig. 1 (Left) where the input length is arbitrary and the output length is n bits. F is a (initially everywhere $\perp$) table.

$\mathcal{RO}^*_v$ is shown in Fig. 1 (Left) where the input length is arbitrary and the output length is v bits, and F^* is a (initially everywhere $\perp$) table. Note that v is defined in each hash function.

$\mathcal{TRO}_w$ is shown in Fig. 1 (Center) which consists of a RO $\mathcal{RO}^T_w$ and a trace oracle $\mathcal{TO}$. The output length of $\mathcal{RO}^T_w$ and the input length of $\mathcal{TO}$ are w bits, and F^*_i is a (initially everywhere $\perp$) table. Note that w is defined in each hash function.

$\mathsf{IC}_{a,b}$ can be implemented as Fig. 1 (Right) which consists of an encryption oracle E and a decryption oracle D where the first input of E is the key of a bits and the second input is the plain text of b bits,

$\underline{\mathcal{RO}_n(M)}$

if $\mathsf{F}[M] = \perp$, $\mathsf{F}[M] \xleftarrow{\$} \{0,1\}^n$;
return $\mathsf{F}[M]$;

$\underline{\mathcal{RO}^T_w(M)}$

if $\mathsf{F}^T[M] \neq \perp$ then $\mathsf{F}^T[M] \xleftarrow{\$} \{0,1\}^w$;
return $\mathsf{F}^T[M]$;

$\underline{E(k,x)}$

if $\mathsf{E}[k,x] = \perp$, $y \xleftarrow{\$} \{0,1\}^b \backslash T^+[k]$;
$Update(k,x,y)$;
return $\mathsf{E}[k,x]$;

$\underline{\mathcal{RO}^*_v(M)}$

If $\mathsf{F}^*[M] \neq \perp$, $\mathsf{F}^*[M] \xleftarrow{\$} \{0,1\}^v$;
return $\mathsf{F}^*[M]$;

$\underline{\mathcal{TO}(y)}$

if $\exists_1 M$ s.t. $\mathsf{F}^T[M] = y$ then **return** M;
return $\perp$;

$\underline{D(y)}$

if $\mathsf{D}[k,y] = \perp$, $x \xleftarrow{\$} \{0,1\}^b \backslash T^-[k]$;
$Update(k,x,y)$;
return $\mathsf{D}[k,y]$;

Fig. 1. Versatile Oracle $\mathcal{VO}$

$\underline{\mathrm{PFMD}^h(M)}$

$(M_1, \ldots, M_i) \leftarrow div(d, \mathsf{pfpad}(M))$
$x \leftarrow IV$;
For $j = 1, \ldots, i$, $x \leftarrow h(x||M_j)$;
return x;

$\underline{S(x,m)}$

$M^* \leftarrow \mathcal{TO}(x)$;
if $x = IV$ then
  if $\exists M$ s.t. $\mathsf{pfpad}(M) = m$ then $y \leftarrow \mathcal{RO}_n(M)$;
  else $y \leftarrow \mathcal{RO}^T_n(m)$;
else if $M^* \neq \perp$ then
  if $\exists M$ s.t. $\mathsf{pfpad}(M) = M^*||m$ then $y \leftarrow \mathcal{RO}_n(M)$;
  else $y \leftarrow \mathcal{RO}^T_n(M^*||m)$;
else $y \leftarrow \mathcal{RO}^*_n(x,m)$;
return y;

Fig. 2. PFMD Hash Function (left) and Simulator S (right)

and the first input of D is the key of a bits and the second input is the cipher text of b bits. E and D are (initially everywhere $\perp$) tables where for the query $E(k,x)$ (resp. $D(k,y)$) the output is recored in $\mathsf{E}[k,x]$ (resp. $\mathsf{D}[k,y]$). $T^+[k]$ and $T^-[k]$ are (initially empty) tables which store all values of $\mathsf{E}[k,\cdot]$ and $\mathsf{D}[k,\cdot]$, respectively. $Update(k,x,y)$ is the procedure wherein the tables $\mathsf{E}, \mathsf{D}, T^+[k]$ and $T^-[k]$ are updated, $\mathsf{E}[k,x] \leftarrow y, \mathsf{D}[k,y] \leftarrow x, T^+[k] \xleftarrow{\cup} \{y\}$ and $T^-[k] \xleftarrow{\cup} \{x\}$. Note that the length a, b, are defined in each hash function.

4 Reset Indifferentiability for Hash Functions

In this section, we consider the reset indifferentiable security of the important hash functions, prefix-free Merkle-Damgård (PFMD) [15], chop Merkle-Damgård (chop MD) [15], and Sponge [10]. We show that these hash functions are reset indifferentiable from $\mathcal{VO}$s.

4.1 Reset Indifferentiability for the PFMD Hash Function

The PFMD hash function is employed in the SHA-3 finalist BLAKE hash function [3]. In the document of [3], the indifferentiable security is proven when the compression function is a RO.

The PFMD hash function is illustrated in Fig. 2 (Left) where IV is the initial value of n bits, $h : \{0,1\}^{n+d} \to \{0,1\}^n$ is a compression function, and $\mathsf{pfpad} : \{0,1\}^* \to (\{0,1\}^d)^*$ is an injective prefix-free padding where for any different values M, M', $\mathsf{pfpad}(M)$ is not a prefix of $\mathsf{pfpad}(M')$ and the inverse function of pfpad is efficiently computable.

We show that PFMD^h is reset indifferentiable from $\mathcal{VO}$ where h is a RO. We define the parameter of $\mathcal{VO}$ as $v = n$ and $w = n$. Note that in the reset indifferentiable proof ideal ciphers are not used. Thus in this case, $\mathcal{VO}.priv = \mathcal{RO}_n$ and $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{RO}^*_n, \mathcal{TRO}^T_n)$.

Theorem 1. *There exists a simulator S such that for any distinguisher $\mathcal{A}$, the following holds,*

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff},\mathcal{VO}}_{\mathrm{PFMD}^h,S}(\mathcal{A}) \leq \frac{2\sigma(\sigma+2) + q_R(q_R+1)}{2^n}$$

where $\mathcal{A}$ can make queries to left oracle $L = \mathrm{PFMD}^h/\mathcal{RO}_n$ (left queries) and right oracle $R = h/S$ (right queries) at most q_L, q_R times, respectively, and l is a maximum number of blocks of a left query. $\sigma = lq_L + q_R$. S makes at most $2q_R$ queries and runs in time $\mathcal{O}(q_R)$. ♦

We define a graph G, which is initialized with a single node IV. Edges and nodes in this graph are defined by right query-responses which follow the MD structure. The nodes are chaining values and the edges are message blocks. For example, if $(x_1, m_1, y_1), (x_2, m_2, y_2), (x_3, m_3, y_3)$ are query-responses of R such that $x_1 = IV$, $y_1 = x_2$ and $y_2 = x_3$ then IV, y_1, y_2, y_3 are the nodes of G amd m_1, m_2, m_3 are the edges. We denote the MD path by $IV \xrightarrow{m_1} y_1 \xrightarrow{m_2} y_2 \xrightarrow{m_3} y_3$ or $IV \xrightarrow{m_1||m_2||m_3} y_3$. If there exists M such that $\mathsf{pfpad}(M) = m_1||m_2||m_3$, then we call the MD path "PFMD path".

The Simulator S. We define a simulator S in Fig. 2 which does not update the internal state to remove the attack using $S.Rst$. The S's task is to simulate the compression function h such that S is consistent with $\mathcal{RO}_n$, namely, any PFMD path $IV \xrightarrow{M^*} y$ is such that $y = \mathcal{RO}_n(M)$ where $M^* = \mathsf{pfpad}(M)$. For the ordered queries $S(IV, m_1), S(y_1, m_2)$ where $y_1 = S(IV, m_1), y_2 = S(y_1, m_2)$, if there does not exists M such that $\mathsf{pfpad}(M) = m_1||m_2$, then y_1 and y_2 are defined by the responses of $\mathcal{RO}_n^T(m_1)$ and $\mathcal{RO}_n^T(m_1||m_2)$, respectively. Then for the query $S(y_2, m_3)$, the response is defined by the output of $\mathcal{RO}_n(M)$ if there exists M such that $\mathsf{pfpad}(M) = m_1||m_2||m_3$. Notice that $m_1||m_2$ can be obtained by the query $\mathcal{TO}(y_2)$. So the path $IV \xrightarrow{m_1||m_2||m_3} y_3$ is such that $y_3 = \mathcal{RO}_n(M)$ where $\mathsf{pfpad}(M) = m_1||m_2||m_3$. Thus the simulator S succeeds in the simulation of h. The proof is given as follows.

Proof. To evaluate the indifferentiable advantage, we consider seven games. In each game, distinguisher $\mathcal{A}$ has oracle access to left oracle L and right oracle R.

- Game 1 is the $\mathcal{VO}$ world, that is, $(L, R) = (\mathcal{RO}_n, S)$ and $\mathcal{A}$ has oracle access to $S.Rst$.
- Game 2 is $(L, R) = (\mathcal{RO}_n, S)$. Note that $S.Rst$ is removed.
- Game 3 is $(L, R) = (\mathcal{RO}_n, S_1)$. S_1 keeps all query-responses. For query $S_1(x, m)$, if there is a tuple (x, m, y) in the query-response history, then S_1 returns y, otherwise, S_1 returns $S(x, m)$.
- Game 4 is $(L, R) = (L_1, S_1)$, where on query $L_1(M)$ L_1 first makes queries to S_1 which correspond with the calculation of $\mathrm{PFMD}^{S_1}(M)$ then returns $\mathcal{RO}_n(M)$.
- Game 5 is $(L, R) = (\mathrm{PFMD}^{S_1}, S_1)$.
- Game 6 is $(L, R) = (\mathrm{PFMD}^h, h)$.
- Game 7 is the PFMD world, that is, $(L, R) = (\mathrm{PFMD}^h, h)$ and $\mathcal{A}$ has oracle access to *nop*.

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

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff},\mathcal{VO}}_{\mathrm{PFMD}^h,S}(\mathcal{A}) \leq \sum_{i=1}^{6} |\Pr[G_i] - \Pr[G_{i+1}]| \leq \frac{2\sigma(\sigma+2) + q_R(q_R+1)}{2^n}.$$

In the following, we justify the above bound by evaluating each difference. Since S does not update the internal state, $S.Rst$ does not affect the $\mathcal{A}$'s behavior and thus $\Pr[G_1] = \Pr[G_2]$. Since *nop* does noting, $\Pr[G_6] = \Pr[G_7]$. We thus consider game sequences Game 2, Game 3, Game 4, Game 5, and Game 6.

Game 2 ⇒ Game 3. In Game 3, use of the history ensures that for any repeated query $R(x, m)$ the same value y is responded, while in Game 2 there is a case that for some repeated query $R(x, m)$ where y was responded, different value y^* ($\neq y$) is responded due to the definition of $\mathcal{TO}$. The difference $|\Pr[G_2] - \Pr[G_3]|$ is bounded by the probability that in Game 2 the different value is responded. The different value are not responded unless an event Bad_j occurs: Let T_i be a list which records all responses y of S and the first values x of all queries to S before the i-th query to S. Bad_j is that in Game j for some i-th query $S(x_i, m_i)$ the response y_i collides with some value in T_i. This is because an output of $\mathcal{TO}(x)$ is determined by responses of $\mathcal{RO}_n^T$ and the query x to $\mathcal{TO}$ is the first value of the query to S. We thus have that $|\Pr[G_2] - \Pr[G_3]| \leq \Pr[Bad_2]$.

Since S is called at most q_R time and outputs of S are chosen uniformly at random from $\{0,1\}^n$, $\Pr[Bad_2] \leq \sum_{i=2}^{q_R} 2i/2^n = (q_R - 1)(q_R + 2)/2^n$. We thus have that $|\Pr[G_2] - \Pr[G_3]| \leq q_R(q_R + 1)/2^n$.

Game 3 ⇒ Game 4. The difference between Game 3 and Game 4 is that in Game 3 L does not make a right query, while in Game 4 L makes additional right queries corresponding with $\mathrm{PFMD}^{S_1}(M)$. Note that $\mathcal{A}$ cannot find the additional right query-responses but can find those by making corresponding right queries. So we must show that the additional right queries and the responses that $\mathcal{A}$ obtains don't affect the $\mathcal{A}$'s behavior. We show Lemma 1 where for any PFMD path $IV \xrightarrow{M^*} y$ where $M^* = \mathsf{pfpad}(M)$, $y = \mathcal{RO}_n(M)$ unless Bad_j or Bad_j^* occurs where Bad_j^* is an event that in Game j the response of some query $S(x,m)$ collides with IV. Namely, unless the bad event occurs, in the both games, responses which are leafs in PFMD paths[2] are defined by $\mathcal{RO}_n$, and other responses are defined by random choices of $\mathcal{RO}_n^T$ or $\mathcal{RO}_n^*$. Namely, unless the bad event occurs, the responses of the additional right queries which $\mathcal{A}$ obtains are chosen from the same distribution as in Game 3. Thus, the difference $|\Pr[G_3] - \Pr[G_4]|$ is bounded by the probability of occurring the bad event. Let $bad_j = Bad_j \vee Bad_j^*$. We thus have that

$$\begin{aligned}|\Pr[G_3] - \Pr[G_4]| =& |\Pr[bad_3 \wedge G_3] + \Pr[G_3|\neg bad_3]\Pr[\neg bad_3] - (\Pr[bad_4 \wedge G_4] + \Pr[G_4|\neg bad_4]\Pr[\neg bad_4])| \\ \le& |\Pr[bad_3 \wedge G_3] - \Pr[bad_4 \wedge G_4] - \Pr[G3|\neg bad_3](\Pr[bad_3] - \Pr[bad_4])| \\ \le& \max\{\Pr[bad_3], \Pr[bad_4]\} \le \frac{\sigma(\sigma+2)}{2^n}\end{aligned}$$

where $\Pr[G3|\neg bad_3] = \Pr[G4|\neg bad_4]$ from Lemma 1 and $\Pr[bad_3] \le \Pr[bad_4]$. We justify the bound later.

Lemma 1. *In Game j, unless bad_j occurs, for any PFMD path $IV \xrightarrow{M^*} y$ $y = \mathcal{RO}_n(M)$ where $M^* =$* $\mathsf{pfpad}(M)$. ♦

Proof of Lemma 1. Assume that bad_j does not occur. Let $IV \xrightarrow{M^*} y$ be any PFMD path. We show that $y = \mathcal{RO}_n(M)$ where $M^* = \mathsf{pfpad}(M)$. Let $(x_1, m_1, y_1), \ldots, (x_j, m_j, y_j)$ be query-responses of S which correspond with the PFMD path where $x_1 = IV$, $x_i = y_{i-1}$ $(i = 2, \ldots, j)$, $y_j = y$, and $M^* = m_1||\ldots||m_j$.

If $j = 1$ then $y = \mathcal{RO}_n(M)$.

We consider the case that $j \ge 2$.

If some triple (x_i, m_i, y_i) is defined after $(x_{i+1}, m_{i+1}, y_{i+1})$ was defined, the assumption ensures that (x_i, m_i, y_i) does not connect with $(x_{i+1}, m_{i+1}, y_{i+1})$, namely, $y_i \ne x_i$. So $(x_1, m_1, y_1), \ldots, (x_j, m_j, y_j)$ are defined by this order.

Since $\mathcal{RO}_n^T$ is used to define an output of S, the assumption ensures that no collision for $\mathcal{RO}_n^T$ occurs and no output of $\mathcal{RO}_n^T$ collides with IV. Thus for the query $S(x_j, m_j)$, $\mathcal{TO}(x_j)$ responses $m_1||\ldots||m_{j-1}$ and then the response y_i is the output of $\mathcal{RO}_n(M)$. □

Evaluation of $\Pr[Bad_3], \Pr[Bad_4], \Pr[Bad_3^*], \Pr[Bad_4^*]$**.** Since in Game 3 and Game 4 S is called at most q_R and σ times, respectively, and for any query to S the response is chosen uniformly at random from $\{0,1\}^n$ and is independent from the table T_i due to the prefix-free padding, $\Pr[Bad_3] \le \sum_{i=2}^{q_R} 2i/2^n = (q_R-1)(q_R+2)/2^n$ and $\Pr[Bad_4] \le \sum_{i=2}^{\sigma} 2i/2^n = (\sigma-1)(\sigma+2)/2^n$. $\Pr[Bad_3^*] \le q_R/2^n$ and $\Pr[Bad_4^*] \le \sigma/2^n$.

Game 4 ⇒ Game 5. The difference between Game 4 and Game 5 is the left oracle L where in Game 4 $L(M)$ returns $\mathcal{RO}_n(M)$, while in Game 5 $L(M)$ returns $\mathrm{PFMD}^{S_1}(M)$. Thus, the difference does not change behavior of $\mathcal{A}$ iff in Game 5 for any query $L(M)$, $L(M)$ returns $\mathcal{RO}_n(M)$. From Lemma 1, for any PFMD path $IV \xrightarrow{M^*} z$, $z = \mathcal{RO}_n(M)$ unless the bad event bad_5 occurs in Game 5, where $M^* = \mathsf{pfpad}(M)$. We have that $|\Pr[G_4] - \Pr[G_5]| \le \Pr[Bad_5] \le \sigma(\sigma+2)/2^n$.

In the following, we justify the bound. In Game 5 R is called at most σ times and for any query to S the response is chosen uniformly at random from $\{0,1\}^n$. We thus have that $\Pr[bad_5] \le ((\sigma-1)(\sigma+2)+\sigma)/2^n$.

Game 5 ⇒ Game 6. Since outputs of S are uniformly chosen at random from $\{0,1\}^n$, the difference for R does not affect the $\mathcal{A}$'s behavior. We thus have that $\Pr[G_5] = \Pr[G_6]$. □

Remark 1. The EMD hash function [7] and the MDP hash function [21] are designed from the same design spirit as the PFMD hash function, which are designed to resist the length extension attack. Thus, by the similar proof, one can prove that EMD and MDP are reset indifferentiable from $\mathcal{VO}$s.

[2] A leaf of a PFMD path $IV \xrightarrow{M^*} y$ is y.

$\underline{\mathrm{chopMD}^h(M)}$

$M' \leftarrow \mathsf{pad}_c(M)$;
$(M_1, \ldots, M_i) \leftarrow div(d, M')$;
$x \leftarrow IV$;
for $j = 1, \ldots, i$ do $x \leftarrow h(x, M_j)$;
return $x[s+1, s+n]$;

$\underline{S(x, m)}$ where $x = x_1 || x_2$ ($|x_1| = s$, $|x_2| = n$)

$M \leftarrow \mathcal{TO}(x_1)$;
if $x = IV$ then
    $z \leftarrow \mathcal{RO}_n(m)$;
    $w \leftarrow \mathcal{RO}_s^T(m)$;
else if $M \neq \perp$ then
    $z \leftarrow \mathcal{RO}_n(M||m)$;
    $w \leftarrow \mathcal{RO}_s^T(M||m)$;
else $w||z \leftarrow \mathcal{RO}^*_{n+s}(x, m)$;
return $w||z$;

Fig. 3. chop MD (left) and S (right)

4.2 Reset Indifferentiability for the Chop MD Hash Function

The chop MD hash function is employed in SHA-2 family, SHA-224 and SHA-384 [26].

Fig. 3 illustrates the chop MD hash function $\mathrm{chopMD}^h : \{0,1\}^* \to \{0,1\}^n$. $h : \{0,1\}^{d+n+s} \to \{0,1\}^{n+s}$ is a compression function. $\mathsf{pad}_c : \{0,1\}^* \to (\{0,1\}^d)^*$ is an injective padding function such that the inverse function is efficiently computable.

We evaluate the reset indifferentiable security of the chop MD hash function where h is a RO. We define the parameter of $\mathcal{VO}$ as $w = s$ and $v = n+s$. Note that the ideal cipher in $\mathcal{VO}$ is not used. Thus, in this case, $\mathcal{VO} = (\mathcal{RO}_n, \mathcal{RO}^*_{s+n}, \mathcal{TRO}_s)$. The following theorem shows that chopMD^h is reset indifferentiable from $\mathcal{VO}$.

Theorem 2. *There exists a simulator S such that for any distinguisher $\mathcal{A}$, the following holds,*

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff},\mathcal{VO}}_{\mathrm{chopMD}^h,S}(\mathcal{A}) \leq \frac{2\sigma(\sigma+1) + q_R(q_R+1)}{2^s} + \frac{\sigma}{2^{s+n}}$$

where $\mathcal{A}$ can make queries to left oracle $L = \mathrm{chopMD}^h/\mathcal{RO}_n$ (left queries) and right oracle $R = h/S$ (right queries) at most q_L, q_R times, respectively, and l is a maximum number of blocks of a query to $\mathrm{chopMD}^h/\mathcal{RO}_n$. S makes at most $3q_h$ queries and runs in time $\mathcal{O}(q_h)$. ♦

In the following proof, we use the graph G which are defined in the Subsection 4.1. The graph is constructed from right query-responses.

The Simulator S. We define the simulator S in Fig. 3 which does not update the internal state to remove the attack using $S.Rst$. In the proof of Theorem 2, the padding function pad_c is removed. Thus the left queries should be in $(\{0,1\}^d)^*$. Note that the chop MD hash function with the padding function is the special case of one without the padding function. Thus the security of the chop MD hash function without the padding function ensures the security of one with the padding function. The S's task is to simulate the compression function h such that $\mathcal{RO}_n$ and S are consistent, that is, for any MD path $IV \xrightarrow{M} z$, $z[s+1, n+s] = \mathcal{RO}_n(M)$. For the ordered queries $S(IV, M_1), S(w_1||z_1, M_2)$ where $w_1||z_1 = S(IV, M_1), w_2||z_2 = S(w_1||z_1, M_2)$, the structure of S ensures that $z_1 = \mathcal{RO}_n(M_1)$, $w_1 = \mathcal{RO}_s^T(M_1)$, $z_2 = \mathcal{RO}_n(M_1||M_2)$, and $w_2 = \mathcal{RO}_s^T(M_1||M_2)$. Thus, the path $(M_1||M_2, w_2)$ is recorded in the table F^T where $\mathsf{F}^T[M_1||M_2] = w_2$. Then, for the query $S(w_2||z_2, M_3)$, the response $w_3||z_3$ is defined as $z_3 = \mathcal{RO}_n(M_1||M_2||M_3)$ and $w_3 = \mathcal{RO}_s^T(M_1||M_2||M_3)$. Notice that $M_1||M_2$ can be obtained by the queries $\mathcal{TO}(w_2)$. So the path $IV \xrightarrow{M_1||M_2||M_3} w_3||z_3$ is such that $z_3 = \mathcal{RO}_n(M_1||M_2||M_3)$. Thus the simulator S succeeds in the simulation. The proof is given as follows.

Proof. To evaluate the indifferentiable advantage, we consider seven games. In each game, distinguisher $\mathcal{A}$ has oracle access to left oracle L and right oracle R.

- Game 1 is the $\mathcal{VO}$ world, that is, $(L, R) = (\mathcal{RO}_n, S)$ and $\mathcal{A}$ has oracle access to $S.Rst$.
- Game 2 is $(L, R) = (\mathcal{RO}_n, S)$. Note that $S.Rst$ is removed.
- Game 3 is $(L, R) = (\mathcal{RO}_n, S_1)$, where S_1 keeps all query-responses (x, m, y). For the query $S_1(x, m)$, if there is (x, m, y) in the query-response history, then S_1 returns y, otherwise, S_1 returns $S(x, m)$.

– Game 4 is $(L, R) = (L_1, S_1)$, where on a query $L_1(M)$ L_1 first makes queries to S_1 which correspond with $\mathrm{chopMD}^{S_1}(M)$ then returns $\mathcal{RO}_n(M)$.
– Game 5 is $(L, R) = (\mathrm{chopMD}^{S_1}, S_1)$.
– Game 6 is $(L, R) = (\mathrm{chopMD}^h, h)$.
– Game 7 is the chop MD world, that is, $(L, R) = (\mathrm{chopMD}^h, h)$ and $\mathcal{A}$ has oracle access to *nop*.

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

$$\mathsf{Adv}^{\text{r-indiff},\mathcal{VO}}_{\mathrm{PFMD}^h,S}(\mathcal{A}) \le \sum_{i=1}^{6} |\Pr[G_i] - \Pr[G_{i+1}]| \le \frac{2\sigma(\sigma+1) + q_R(q_R+1)}{2^s} + \frac{\sigma}{2^{s+n}}.$$

In the following, we justify the above bound by evaluating each difference. Since S does not update the internal state, $S.Rst$ does not affect $\mathcal{A}$'s behavior between Game 1 and Game 2 and thus $\Pr[G_1] = \Pr[G_2]$. Since *nop* does noting, $\Pr[G_6] = \Pr[G_7]$. We thus consider game sequences Game 2, Game 3, Game 4, Game 5, and Game 6.

Game 2 ⇒ Game 3. In Game 3, use of the history ensures that $R(x, m)$ the same value y is responded, while in Game 2 there is a case that for some repeated query $R(x, m)$ where y was responded, different value y^* $(\neq y)$ is responded due to the definition of $\mathcal{TO}$. The difference $|\Pr[G_2] - \Pr[G_3]|$ is bounded by the probability that in Game 2 the different value is responded. The different value are not responded unless an event Bad_j occurs: Let T_i be a list which records the s-bit values $y[1, s]$ of all responses y of S and the s-bit value $x[1, s]$ of all queries x to S before the i-th query to S. Bad_j is that in Game j for some i-th query $S(x, m)$ the s-bit value $y[1, s]$ of the responses y collides with some value in T_i. This is because an output of $\mathcal{TO}(x_1)$ is determined by responses of $\mathcal{RO}^T_n$ and the query x_1 to $\mathcal{TO}$ is $x[1, s]$ where x is the first value of the query to S. We thus have that $|\Pr[G_2] - \Pr[G_3]| \le \Pr[Bad_2]$.

Since S is called at most q_R time and outputs of S are chosen uniformly at random from $\{0, 1\}^{s+n}$, $\Pr[Bad_2] \le \sum_{i=2}^{q_R} 2i/2^s = (q_R - 1)(q_R + 2)/2^s$. We thus have that $|\Pr[G_2] - \Pr[G_3]| \le q_R(q_R + 1)/2^s$.

Game 3 ⇒ Game 4. The difference between Game 3 and Game 4 is that for a left query $L(M)$, in Game 3 L does not make a right query, while in Game 4 L makes additional right queries corresponding with $\mathrm{chopMD}^{S_1}(M)$. Note that $\mathcal{A}$ cannot find the additional right query-responses but can find those by making corresponding right queries. So we must show that the additional right queries and responses that $\mathcal{A}$ obtains don't affect the $\mathcal{A}$'s behavior. We show Lemma 2 where for any MD path $IV \xrightarrow{M} z$, $z[s+1, n+s] = \mathcal{RO}_n(M)$ unless Bad_j or Bad^*_j occurs where Bad^*_j is that in Game j for some query $S(x, m)$ the response y collides with IV. This ensures that unless the bad event occurs, in both games responses which are leafs of MD paths[3] are defined by $\mathcal{RO}^T_s$ and $\mathcal{RO}_n$, and other responses are defined by $\mathcal{RO}^*_{n+s}$. Namely, unless the bad event occurs, the responses of the additional right queries which $\mathcal{A}$ obtains are chosen from the same distribution as in Game 3. Thus, the difference $|\Pr[G_3]-\Pr[G_4]|$ is bounded by the probability of occurring the bad event. Let $bad_j = Bad_j \vee Bad^*_j$. We thus have that $|\Pr[G_3]-\Pr[G_4]| \le \max\{\Pr[bad_3], \Pr[bad_4]\} \le \sigma(\sigma+1)/2^s + \sigma/2^{s+n}$ where $\Pr[G3|\neg bad_3] = \Pr[G4|\neg bad_4]$ from Lemma 1 and $\Pr[bad_3] \le \Pr[bad_4]$. We justify the bound later.

Lemma 2. *In Game j, unless bad_j occurs, for any MD path $IV \xrightarrow{M} y$ $y[s+1, n+s] = \mathcal{RO}_n(M)$.* ♦

Proof of Lemma 2. Assume that bad_j does not occur. Let $IV \xrightarrow{M} y$ be any MD path. We show that $y[s+1, n+s] = \mathcal{RO}_n(M)$. Let $(x_1, m_1, y_1), \ldots, (x_j, m_j, y_j)$ be query-responses of S which correspond with the MD path where $x_1 = IV$, $x_i = y_{i-1}$ $(i = 2, \ldots, j)$, $y_j = y$, and $M = m_1||\ldots||m_j$.

When $j = 1$, $y[s+1, n+s] = \mathcal{RO}_n(M)$.

We consider the case that $j \ge 2$.

If some triple (x_i, m_i, y_i) is defined after $(x_{i+1}, m_{i+1}, y_{i+1})$ was defined, the assumption ensures that (x_i, m_i, y_i) does not connect with $(x_{i+1}, m_{i+1}, y_{i+1})$, namely, $y_i \neq x_i$. So $(x_1, m_1, y_1), \ldots, (x_j, m_j, y_j)$ are defined by this order.

Since $\mathcal{RO}^T_n$ is used to define outputs of S, the assumption ensures that no collision for $\mathcal{RO}^T_n$ occurs. And no output of S collides with IV. Thus for the query $S(x_j, m_j)$, $\mathcal{TO}(x_j)$ responses $m_1||\ldots||m_{j-1}$ and then for the response y_i $y_i[s+1, n+s]$ is the output of $\mathcal{RO}_n(M)$. □

[3] A leaf of the MD path $IV \xrightarrow{M} z$ is z.

Algorithm $Sponge^P(M)$	$S_F(X)$ where $x = X[1,n]$, $y = Y[n+1,d]$	$S_I(Y)$ where $z = Y[1,n]$, $w = Y[n+1,d]$
$M' \leftarrow \mathsf{pad}_S(M)$;	$M \leftarrow \mathcal{TO}(y)$;	$M \leftarrow \mathcal{TO}(w)$;
$(M_1, \ldots, M_i) \leftarrow div(n, M)$;	if $y = IV_2$ then	if $M \neq \perp$ and $\|M\| = n$ then
$s = IV$;	$\quad z \leftarrow \mathcal{RO}_n(x \oplus IV_1)$; $w \leftarrow \mathcal{RO}_c^T(x \oplus IV_1)$;	$\quad x \leftarrow IV_1 \oplus M$; $y \leftarrow IV_2$;
for $i = 1, \ldots, i$ do	else if $M \neq \perp$ then	if $M \neq \perp$ and $\|M\| > n$ then
$\quad s = P(s \oplus (M_i \|\| 0^c))$;	$\quad m \leftarrow x \oplus \mathcal{RO}_n(M)$;	$\quad$ let $M = M^* \|\| m$ ($\|m\| = n$);
return $s[1,n]$;	$\quad z \leftarrow \mathcal{RO}_n(M \|\| m)$; $w \leftarrow \mathcal{RO}_c^T(M \|\| m)$;	$\quad x \leftarrow m \oplus \mathcal{RO}_n(M)$; $y \leftarrow \mathcal{RO}_c^T(M^*)$;
	else $z \|\| w \leftarrow \mathcal{P}(x \|\| y)$;	else $x \|\| y \leftarrow \mathcal{P}^{-1}(z \|\| w)$;
	return $z \|\| w$;	**return** $x \|\| y$;

Fig. 4. Sponge Hash Function (left) and Simulator S (S_F in center and S_I in right)

Evaluation of $\Pr[Bad_3], \Pr[Bad_4], \Pr[Bad_3^*], \Pr[Bad_4^*]$**.** Since in Game 3 and Game 4 S is called at most q_R and σ times, respectively, and for any query to S the response is chosen uniformly at random from $\{0,1\}^n$ and is independent from the list T_i, $\Pr[Bad_3] \leq \sum_{i=2}^{q_R} 2i/2^s = (q_R - 1)(q_R + 2)/2^s$, $\Pr[Bad_4] \leq \sum_{i=2}^{\sigma} 2i/2^s = (\sigma - 1)(\sigma + 2)/2^s$, $\Pr[Bad_3^*] \leq q_R/2^{s+n}$, and $\Pr[Bad_4^*] \leq \sigma/2^{s+n}$.

Game 4 $\Rightarrow$ Game 5. The difference between Game 4 and Game 5 is the left oracle L where in Game 4 $L(M)$ returns $\mathcal{RO}_n(M)$, while in Game 5 $L(M)$ returns $\mathrm{chopMD}^{S_1}(M)$. Thus, the difference does not change behavior of $\mathcal{A}$ iff in Game 5 for any query $L(M)$, $L(M)$ returns $\mathcal{RO}_n(M)$. From Lemma 2, for any MD path $IV \xrightarrow{M} z$, $z[s+1, s+n] = \mathcal{RO}_n(M)$ unless the bad event bad_5 occurs. We have that $|\Pr[G_4] - \Pr[G_5]| \leq \Pr[bad_5] \leq \sigma(\sigma+1)/2^s + \sigma/2^{n+s}$. In the following, we justify the bound. In Game 5 R is called at most σ times and for any query to S the response is chosen uniformly at random from $\{0,1\}^{n+s}$. We thus have that $\Pr[Bad_5] \leq (\sigma - 1)(\sigma + 2)/2^s + \sigma/2^{n+s}$.

Game 5 $\Rightarrow$ Game 6. Since outputs of S are uniformly chosen at random from $\{0,1\}^n$, the difference of R does not affect the $\mathcal{A}$'s behavior. We thus have that $\Pr[G_5] = \Pr[G_6]$. $\square$

4.3 Reset Indifferentiability for the Sponge Hash Function

The Sponge hash function is a permutation-based hash function which employed in the SHA-3 candidate Keccak [11].

Fig. 4 (left) illustrates the Sponge hash function where IV is the initial value of d bits, $\mathsf{pad}_S : \{0,1\}^* \rightarrow (\{0,1\}^n)^*$ is an injective padding function such that the final block message $M_i \neq 0$, $P : \{0,1\}^b \rightarrow \{0,1\}^b$ is a permutation and $d = n + c$. The inverse function of pad_S is denoted by $\mathsf{unpad}_S : (\{0,1\}^n)^* \rightarrow \{0,1\}^* \cup \{\perp\}$ efficiently computable. $\mathsf{unpad}_S(M^*)$ outputs M if there exists M such that $\mathsf{pad}_S(M) = M^*$, and outputs $\perp$ otherwise. Note that the Sponge hash function of Fig. 4 is the special case of the general Sponge hash function where the output length is arbitrary. The output lengths of SHA-3 are 224, 256, 384 and 512 bits and in this case the Keccak hash function has the structure of Fig. 4.[4] We conjecture that the reset indifferentiable security of the general Sponge hash function can be proven by extending the following analysis of the Sponge hash function. We denote the left most n-bit value and the right most c bit value of IV by IV_1 and IV_2, respectively. Namely, $IV = IV_1 || IV_2$.

We evaluate the reset indifferentiable security of the Sponge hash function in the random permutation model, where P is a random permutation and P^{-1} is its inverse oracle.[5] We define the parameter of $\mathcal{VO}$ as $w = c$ and $b = d$. We don't care the key length a, since in this proof we fix the key by some constant value, that is the fixed key ideal cipher is used, which is a random permutation of d bits. So we use the random permutation $(\mathcal{P}, \mathcal{P}^{-1})$ of d bits instead of the ideal cipher $\mathsf{IC}_{a,b}$ where $\mathcal{P}$ is a forward oracle and $\mathcal{P}^{-1}$ is an inverse oracle. Note that in this proof random oracles $\mathcal{RO}^*$ are not used. Thus, in this case, $\mathcal{VO}.priv = \mathcal{RO}_n$

[4] In the Keccak case, $b = 1600$ and $c = 576$. So, the output length of Keccak is shorter than n. Since a chopped RO is also a RO, the reset indifferentiable security of Sponge with the n-bits output length implies that of Sponge with the shorter output length.

[5] The security of the Sponge hash function was evaluated in the random permutation model [10].

$\mathcal{P}_1(X)$	$\mathcal{P}_1^{-1}(X)$
1 if $\exists(j,X,Y) \in \mathcal{Q}$ then **return** Y;	1 if $\exists(j,X,Y) \in \mathcal{Q}$ then **return** X;
2 $Y \xleftarrow{\$} \{0,1\}^d$; $\mathcal{Q} \xleftarrow{\cup} (t,X,Y)$; $t \leftarrow t+1$;	2 $X \xleftarrow{\$} \{0,1\}^d$; $\mathcal{Q} \xleftarrow{\cup} (t,X,Y)$; $t \leftarrow t+1$;
3 **return** Y;	3 **return** X;

Fig. 4.3. $\mathcal{Q}$ is a (initially empty) list and initially $t = 1$. In the step 1 of $\mathcal{P}_1, \mathcal{P}_1^{-1}$, j is a maximum value.

and $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{TRO}_c, \mathcal{P}, \mathcal{P}^{-1})$. The following theorem is that the sponge hash function $Sponge^P$ is reset indifferentiable from $\mathcal{VO}$.

Theorem 3 (Sponge is reset indifferentiable from a $\mathcal{VO}$). *There exists a simulator $S = (S_F, S_I)$ such that for any distinguisher $\mathcal{A}$, the following holds.*

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

where $\mathcal{A}$ can make at most q_L, q_F and q_I queries to left $L = Sponge^P/\mathcal{RO}_n$ (left queries) and right $R_F = P/S_F, R_I = P^{-1}/S_I$ oracles (right queries). $\sigma = lq_L + q_F + q_I$ and $q = q_F + q_I$. S makes at most $7q$ queries and runs in time $\mathcal{O}(q)$. ♦

We define a graph G, which is initialized with the single node IV. Edges and nodes in this graph are defined by right query-responses which follows the Sponge structure. The nodes are chaining values and the edges are message blocks. For example, if $(X_1,Y_1),(X_2,Y_2),(X_3,Y_3)$ are query-responses of R such that $X_1[n+1,c+n] = IV_2$, $Y_1[n+1,c+n] = X_2[n+1,c+n]$ and $Y_2[n+1,c+n] = X_3[n+1,c+n]$ then IV, Y_1, Y_2, Y_3 are the nodes of G and M_1, M_2, M_3 are the edges where $M_1 = IV_1 \oplus X_1[1,n]$. We denote the path by $IV \xrightarrow{M_1} Y_1 \xrightarrow{M_2} Y_2 \xrightarrow{M_3} Y_3$ or $IV \xrightarrow{M_1||M_2||M_3} Y_3$. We call the path "Sponge path".

The Simulator S. We define the simulator S in Fig. 4 which does not update the internal state to remove the attack using $S.Rst$. The S's task is to simulate (P, P^{-1}) such that $\mathcal{RO}_n$ and S are consistent, that is, for any Sponge path $IV \xrightarrow{M} Y$, $Y[1,n] = \mathcal{RO}_n(M)$. In the proof of Theorem 3, the padding function pad_S is removed. Thus the left queries should be in $(\{0,1\}^n)^*$. Note that the Sponge with the padding function is the special case of one without the padding function. Thus the security of the Sponge without the padding function ensures the security of one with the padding function. S_F and S_I simulate P and P^{-1}, respectively. For the ordered queries $S_F(x_1||IV_2), S_F(x_2||w_1)$ where $z_1||w_1 = S_F(x_1||IV), z_2||w_2 = S_F(x_2||w_1)$, the structure of S ensures that $w_1 = \mathcal{RO}_c^T(M_1)$ and $w_2 = \mathcal{RO}_c^T(M_1||M_2)$ where $M_1 = IV_1 \oplus x_1$ and $M_2 = z_1 \oplus x_2$. Then, for the query $S_F(x_3||w_2)$, the response $z_3||w_3$ is defined as $z_3 = \mathcal{RO}_n(M_1||M_2||M_3)$ and $w_3 = \mathcal{RO}_c^T(M_1||M_2||M_3)$ where $M_3 = z_2 \oplus x_3$. Notice that $M_1||M_2$ can be obtained by the queries $\mathcal{TO}(w_2)$ and z_2 can be obtained by the query $\mathcal{RO}_n(M_1||M_2)$. Thus the simulator S succeeds in the simulation of the random permutation. The proof is given as follows.

Proof. To evaluate the indifferentiable bound, we consider eight games. In each game, distinguisher $\mathcal{A}$ has oracle access to the left oracle L and the right oracles R_F, R_I.

- Game 1 is the $\mathcal{VO}$ world, that is, $(L, R_F, R_I) = (\mathcal{RO}_n, S_F, S_I)$ and $\mathcal{A}$ has oracle access to $S.Rst$.
- Game 2 is $(L, R_F, R_I) = (\mathcal{RO}_n, S_F, S_I)$. Note that $S.Rst$ is removed.
- Game 3 is that a random permutation $\mathcal{P}$ and its inverse $\mathcal{P}^{-1}$ are changed into $\mathcal{P}_1$ and $\mathcal{P}_1^{-1}$, respectively. So the simulator uses $(\mathcal{P}_1, \mathcal{P}_1^{-1})$ instead of $(\mathcal{P}, \mathcal{P}^{-1})$. $(\mathcal{P}_1, \mathcal{P}_1^{-1})$ are implemented as in Figure. 4.3.
- Game 4 is $(L, R_F, R_I) = (\mathcal{RO}_n, S1_F, S1_I)$, where $S1$ keeps all query-responses (X,Y) where $Y = S1_F(X)$ or $X = S1_I(Y)$. For query $S1_F(X)$, if there is (X,Y) in the query-response history, then $S1_F$ returns Y, otherwise, $S1_F$ returns $S_F(X)$. For query $S1_I(Y)$, if there is (X,Y) in the query-response history, then $S1_I$ returns X, otherwise, $S1_I$ returns $S_I(Y)$.
- Game 5 is $(L, R_F, R_I) = (L_1, S1_F, S1_I)$, where on a query $L_1(M)$ L_1 first makes $S1_F$ queries which correspond with $Sponge^{S1_F}(M)$ then returns $\mathcal{RO}_n(M)$.
- Game 6 is $(L, R_F, R_I) = (Sponge^{S1_F}, S1_F, S1_I)$.
- Game 7 is $(L, R_F, R_I) = (Sponge^P, P, P^{-1})$.

– Game 8 is the Sponge world, that is, $(L, R_F, R_I) = (Sponge^P, P, P^{-1})$ and $\mathcal{A}$ has oracle access to *nop*.

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

$$\mathsf{Adv}^{\text{r-indiff},\mathcal{VO}}_{Sponge^P,S}(\mathcal{A}) \leq \sum_{i=1}^{7} |\Pr[G_i] - \Pr[G_{i+1}]| \leq \frac{2\sigma^2 + q(q+1)}{2^c} + \frac{\sigma(\sigma+1) + q(q+1)}{2^{d+1}}.$$

In the following, we justify the above bound by evaluating each difference. Since S does not update the internal state, $S.Rst$ does not affect the $\mathcal{A}$'s behavior between Game 1 and Game 2 and thus $\Pr[G_1] = \Pr[G_2]$. Since *nop* does noting, $\Pr[G_7] = \Pr[G_8]$. We thus consider Games 2, 3, 4, 5, 6, 7. We call a query to R_F "forward query" and a query to R_I "inverse query".

Game 2 $\Rightarrow$ Game 3. In Game 2, a random permutation $\mathcal{P}$ and its inverse $\mathcal{P}^{-1}$ are uses, while in Game 3, $\mathcal{P}_1$ and $\mathcal{P}_1^{-1}$ are used where the outputs are uniformly chosen at random from $\{0,1\}^d$. Thus $|\Pr[G_2] - \Pr[G_3]|$ is bounded by a collision probability of $(\mathcal{P}_1, \mathcal{P}_1^{-1})$. Since $\mathcal{P}_1$ and $\mathcal{P}_1^{-1}$ are called at most q times, $|\Pr[G_2] - \Pr[G_3]| \leq \sum_{t=2}^{q} t/2^d = q(q+1)/2^{d+1}$.

Game 3 $\Rightarrow$ Game 4. In Game 4, use of the history ensures that for any repeated query $R_F(X)$ (resp. $R_I(Y)$) the same value Y (resp. X) is responded, while in Game 3 there is a case due to the definition of $\mathcal{TO}$ where for some repeated query $R_F(X)$ (or $R_I(Y)$) where Y (or X) was responded, different value Y^* (or X^*) is responded. The difference $|\Pr[G_2] - \Pr[G_3]|$ is bounded by the probability that in Game 2 the different value is responded. The different value are not responded unless an event Bad_j occurs: Let T_i be a list which records the c-bit values $X[n+1,d], Y[n+1,d]$ of all query-responses (X,Y) of S_F, S_I before the i-th query to S. Bad_j is that in Game j for some i-th query $S(X_i)$ $Y_i[n+1,d]$ where Y_i is the response collides with some value in T_i or for some i-th query $S_I(Y_i)$ $X_i[n+1,d]$ where X_i is the response collides with some value in T_i. This is because outputs of $\mathcal{TO}(y)$ (or $\mathcal{TO}(w)$) are determined by query-responses of $\mathcal{RO}_n^T$ and the value y (or w) which is a query to $\mathcal{TO}$ is $X[n+1,d]$ (or $Y[n+1,d]$) where (X,Y) is the query-response of the simulator. We thus have that $|\Pr[G_3] - \Pr[G_4]| \leq \Pr[Bad_3]$.

Since the simulator is called at most q time and outputs of S_F and S_I are chosen uniformly at random from $\{0,1\}^d$, $\Pr[Bad_3] \leq \sum_{i=2}^{q} 2i/2^c = q(q+1)/2^c$. We thus have that $|\Pr[G_3] - \Pr[G_4]| \leq q(q+1)/2^c$.

Game 4 $\Rightarrow$ Game 5. The difference between Game 4 and Game 5 is that in Game 4 L does not make a right query, while in Game 5 L makes additional right queries corresponding with $Sponge^{S1_F}(M)$. Note that $\mathcal{A}$ cannot find the additional right query-responses but can find those by making corresponding right queries. So we must show that the additional right queries and responses that $\mathcal{A}$ obtains don't affect the $\mathcal{A}$'s behavior. We show Lemma 3 where for any Sponge path $IV \xrightarrow{M} z$, $z[1,n] = \mathcal{RO}_n(M)$ unless Bad_j or Bad_j^* occur where Bad_j^* is an event that in Game j, for some query $S_F(X)$ $Y[n+1,d]$ where Y is the response collides with IV_2 or for some query $S_I(Y)$ $X[n+1,d]$ where X is the response collides with IV_2. This ensures that unless Bad_j or Bad_j^* occurs, responses which are leafs of Sponge paths[6] are defined by $\mathcal{RO}_c^T$ and $\mathcal{RO}_n$, and other responses are defined by random choices of $\mathcal{P}_1$ or $\mathcal{P}_1^{-1}$. Namely, unless the bad event occurs, the responses of the additional right queries which $\mathcal{A}$ obtains are chosen from the same distribution as in Game 4. Thus, the difference $|\Pr[G_4] - \Pr[G_5]|$ is bounded by the probability of occurring the bad event. Let $bad_j = Bad_j \vee Bad_j^*$. We thus have that $|\Pr[G_4] - \Pr[G_5]| \leq \max\{\Pr[bad_4], \Pr[bad_5]\} \leq \sigma(\sigma+1)/2^c + \sigma/2^c$ where $\Pr[G_4|\neg bad_4] = \Pr[G_5|\neg bad_5]$ from Lemma 3 and $\Pr[bad_4] \leq \Pr[bad_5]$. We justify the bound later.

Lemma 3. *In Game j, unless bad_j occurs, for any Sponge path $IV \xrightarrow{M} z$ $z[1,n] = \mathcal{RO}_n(M)$.* ♦

Proof of Lemma 3. Assume that bad_j does not occur. Then no pair (X,Y) which is defined by an inverse query connects IV. Thus any path $IV \xrightarrow{M} z$ such that $|M| = n$ is defined by a forward query. And no pair (X,Y) which is defined by an inverse query connects the leaf z of some sponge path $IV \xrightarrow{M} z$. Thus any path $IV \xrightarrow{M} z$ such that $|M| > n$ is defined by forward queries. So, any pair in any sponge path is defined by forward queries.

---

[6] The leaf of the Sponge path $IV \xrightarrow{M} Y$ is Y.

The assumption ensures that no pair which is defined by a forward query connect another pair, namely, the pair (X,Y) which is defined by a forward query is such that $Y[n+1,d] \neq X^*[n+1,d]$ where (X^*,Y^*) is any pair defined before (X,Y) is defined. Let $IV \xrightarrow{M} z$ be any sponge path and $(X_1,Y_1),\ldots,(X_t,Y_t)$ be the corresponding pairs where $X_1[n+1,d] = IV_2$, $X_i[n+1,d] = Y_{i-1}[n+1,d]$ $(i=2,\ldots,t)$, $Y_t[n+1,d] = z$, and $M = M_1||\ldots||M_t$ where $M_1 = IV_1 \oplus X_1[1,n],\cdots,M_t = Y_{t-1}[1,n] \oplus X_t[1,n]$. Thus $(X_1,Y_1),\ldots,(X_t,Y_t)$ are defined by this order and forward queries.

The assumption ensures that for any i-th query-response (X,Y) such that it is defined by a forward query $Y[n+1,d]$ does not collide with IV_2 or some value in T_i. Since $\mathcal{RO}_c^T$ are used as defining the right c-bit values of outputs of S_F, the assumption ensures that no collision occur for $\mathcal{RO}_c^T$. Thus for a forward query $R_F(X_t)$, S_F can obtain $M_1||\ldots||M_{t-1}$ by the query $\mathcal{TO}(y)$ where $y = X_t[n+1,d]$. Thus $Y_t[1,n] = \mathcal{RO}_n(M)$. $\square$

Evaluation of $\Pr[Bad_4], \Pr[Bad_5], \Pr[Bad_4^*], \Pr[Bad_5^*]$**.** Since in Game 4 and Game 5 the simulator is called at most q and σ times, respectively, and for any query to S the right c-bit value of the response is chosen uniformly at random from $\{0,1\}^c$, $\Pr[Bad_4] \leq \sum_{i=2}^{q} 2i/2^c = q(q-1)/2^c$, $\Pr[Bad_5] \leq \sum_{i=2}^{\sigma} 2i/2^c = \sigma(\sigma-1)/2^c$, $\Pr[Bad_4^*] \leq q/2^c$, and $\Pr[Bad_5^*] \leq \sigma/2^c$.

Game 5 $\Rightarrow$ Game 6. The difference between Game 5 and Game 6 is the left oracle L where in Game 5 $L(M)$ returns $\mathcal{RO}_n(M)$, while in Game 6 $L(M)$ returns $Sponge^{S_1}(M)$. Thus, the difference does not change behavior of $\mathcal{A}$ iff in Game 6 for any query $L(M)$, $L(M)$ returns $\mathcal{RO}_n(M)$. From Lemma 3, for any Sponge path $IV \xrightarrow{M} z$ the relation $z[1,n] = \mathcal{RO}_n(M)$ holds unless the bad event bad_6 occurs. We have that $|\Pr[G_5] - \Pr[G_6]| \leq \Pr[bad_6] \leq \sigma(\sigma+1)/2^c + \sigma/2^c$.

In the following, we justify the bound. In Game 6 R is called at most σ times and for any query to S the response is chosen uniformly at random from $\{0,1\}^c$. We thus have that $\Pr[Bad_5] \leq \sigma(\sigma+1)/2^c + \sigma/2^c$.

Game 6 $\Rightarrow$ Game 7. In Game 6, outputs of R_F and R_I are chosen uniformly at random from $\{0,1\}^d$, while in Game 7, those are a random permutation and its inverse oracle. We thus have that $|\Pr[G_6] - \Pr[G_7]| \leq \sum_{i=2}^{\sigma} i/2^d = \sigma(\sigma+1)/2^{d+1}$.

$\square$

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

In this section, we show that there are cryptographic primitives satsfying multi-stage security in the $\mathcal{VO}$ model. Specifically, we show that for any PKE scheme, the non-adaptive CDA security [5] (including the PRIV security [4]) in the $\mathcal{VO}$ model is obtained by assuming an weak property, IND-SIM security in the RO model. The previous work [27] showed the non-adaptive CDA security for PKE schemes based on the same assumption (IND-SIM) with a specific structured preimage aware [17] hash function. Our work focuses on how we obtain CDA secure PKE schemes with large class of hash functions. If a PKE scheme is IND-SIM secure in the RO model, then it is CDA secure in the $\mathcal{VO}$ model. Combining with our results on reset indifferentiable hash functions in Sec. 4, the scheme is CDA secure with these hash functions. Hash functions we prove reset indifferentiabiluty cover other types of functions compared with the result in [27].

Public Key Encryption (PKE). A public key encryption scheme $\mathcal{AE} = (\mathcal{K},\mathcal{E},\mathcal{D})$ consists of three algorithms. Key generation $\mathcal{K}$ outputs a public key, secret key pair. Encryption $\mathcal{E}$ takes a public key pk, a message m, and randomness r and outputs a cipher text. Decryption $\mathcal{D}$ takes a secret key, a cipher text, and outputs a plaintext or a distinguished symbol $\perp$. For vectors $\mathbf{m},\mathbf{r}$ with $|\mathbf{m}| = |\mathbf{r}| = l$ which is the size of vectors, we denote by $\mathcal{E}(pk,\mathbf{m};\mathbf{r})$ the vector $(\mathcal{E}(pk,\mathbf{m}[1];\mathbf{r}[1]),\ldots,\mathcal{E}(pk,\mathbf{m}[l];\mathbf{r}[l]))$. We say that $\mathcal{AE}$ is deterministic if $\mathcal{E}$ is deterministic.

CDA Security. We explain the CDA security (we quote the explanation of the CDA security in [27]). Fig. 5 illustrates the non-adaptive CDA game in the CPA case for a PKE scheme $\mathcal{AE}$ using a functionality F. This notion captures the security of a PKE scheme when the randomness $\mathbf{r}$ used may not be a string of uniform bits. For the remainder of this section, fix a randomness length $\rho \geq 0$ and a message 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

$\text{CDA}^{\mathcal{A}_1,\mathcal{A}_2}_{\mathcal{AE},F}$

$\beta \xleftarrow{\$} \{0,1\}$
$(pk, sk) \xleftarrow{\$} \mathcal{K}$
$(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow \mathcal{A}_1^{F.pub}$
$\mathbf{c} \leftarrow \mathcal{E}^{F.priv}(pk, \mathbf{m}_\beta, \mathbf{r})$
$\beta' \leftarrow \mathcal{A}_2^{F.pub}(pk, \mathbf{c})$
return $(\beta = \beta')$

$\text{IND-SIM}^{\mathcal{B}}_{\mathcal{AE},\mathcal{S},F}$

$\beta \xleftarrow{\$} \{0,1\}$
$(pk, sk) \xleftarrow{\$} \mathcal{K}$
$\beta' \leftarrow \mathcal{B}^{\mathsf{RoS}, F.pub}(pk)$
return $(\beta = \beta')$

$\mathsf{RoS}(m, r)$

If $\beta = 1$ then **return** $\mathcal{E}^{F.priv}(pk, m, r)$
Otherwise **return** $\mathcal{S}^{F.priv}(pk, |m|)$

Fig. 5. CDA game and IND-SIM game

$|\mathbf{m}_0| = |\mathbf{m}_1| = |\mathbf{r}| = \nu$, 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

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

As noted in [5], 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 for any cryptosystem using the indifferentiable hash function. That is, $\mathcal{A}_1$ and $\mathcal{A}_2$ can easily share the messages $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{r})$.

PRIV Security. The PRIV security is the special case of the CDA security when the PKE scheme $\mathcal{AE}$ being considered has randomness length $\rho = 0$. Thus the PRIV security game for a PKE scheme $\mathcal{AE}$ using a functionality F against adversary $\mathcal{A}_1, \mathcal{A}_2$ is equal to the CDA game when $\rho = 0$. The PRIV advantage for a PRIV adversary $\mathcal{A}_1, \mathcal{A}_2$ is denoted by $\mathsf{Adv}^{\text{priv}}_{\mathcal{AE},F}(\mathcal{A}_1, \mathcal{A}_2)$ which is equal to the CDA advantage with $\rho = 0$.

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 algorythm and from a simulator $\mathcal{S}$ even if the adversary can choose message and randomness. Fig. 5 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 [27], 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.

The CDA (PRIV) Security When a RO is replaced with a $\mathcal{VO}$. The following theorem shows that for any PKE scheme the non-adaptive CDA security in the CPA case in the $\mathcal{VO}$ 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 $(\mathcal{A}_1, \mathcal{A}_2)$ in the $\mathcal{VO}$ model making at most $q_{\mathcal{RO}}, q_{\mathcal{RO}^*}, q_{\mathcal{RO}^T}, q_{\mathcal{TO}}, q_E, q_D$ queries to $\mathcal{RO}_n, \mathcal{RO}^*_v, \mathcal{TRO}_w = (\mathcal{RO}^T_w, \mathcal{TO}), \mathsf{IC}_{a,b} = (E, D)$. For any simulator $\mathcal{S}$ there exists an IND-SIM adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\text{cda}}_{\mathcal{AE},\mathcal{VO}}(\mathcal{A}_1, \mathcal{A}_2) \leq \mathsf{Adv}^{\text{ind-sim}}_{\mathcal{AE},\mathcal{S},\mathcal{RO}_n}(\mathcal{B}) + \frac{\nu q_{\mathcal{RO}}}{2^\mu}.$$

$\mathcal{B}$ makes no RO queries, makes ν RoS-queries, and runs in time that of $(\mathcal{A}_1, \mathcal{A}_2)$ plus $\mathcal{O}(q_{\mathcal{RO}} + q_{\mathcal{RO}^*} + q_{\mathcal{RO}^T} + q_{\mathcal{TO}} + q_E + q_D)$. ♦

Proof. 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 $\mathcal{VO}$ model. Secondly, we transform $\mathbf{G}_0$ to game $\mathbf{G}_1$ so that ciphertext $\mathbf{c}$ is generated from a

Game $\mathbf{G}_1$

$\beta \xleftarrow{\$} \{0,1\}$
$(pk, sk) \xleftarrow{\$} \mathcal{K}$
$(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow \mathcal{A}_1^{\mathcal{RO}_n, \mathcal{RO}_v^*, \mathcal{TRO}_w, \mathsf{IC}_{a,b}}$
$\mathbf{c} \leftarrow \mathcal{E}^{F.priv}(pk, \mathbf{m}_\beta, \mathbf{r})$
$\mathbf{c}' \leftarrow \mathcal{S}^{\mathcal{RO}_n}(pk, \omega)$
$\beta' \leftarrow \mathcal{A}_2^{\mathcal{RO}_n, \mathcal{RO}_v^*, \mathcal{TRO}_w, \mathsf{IC}_{a,b}}(pk, \mathbf{c}')$
return $(\beta = \beta')$

$\mathcal{B}^{\mathsf{RoS}}(pk)$

$\beta \xleftarrow{\$} \{0,1\}$
$(pk, sk) \xleftarrow{\$} \mathcal{K}$
$(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow \mathcal{A}_1^{\mathsf{SimB}}$
$\mathbf{c} \leftarrow \mathsf{RoS}(\mathbf{m}_\beta, \mathbf{r})$
$\beta' \leftarrow \mathcal{A}_2^{\mathsf{SimB}}(pk, \mathbf{c})$
If $\beta = \beta'$ then **return** 1
Otherwise **return** 0

$\mathsf{SimB}_{\mathcal{RO}}(M)$

If $\mathsf{F}[M] = \perp$, $\mathsf{F}[M] \xleftarrow{\$} \{0,1\}^n$
return $\mathsf{F}[M]$

$\mathsf{SimB}_{\mathcal{RO}^*}(M)$

If $\mathsf{F}^*[M] \neq \perp$, $\mathsf{F}^*[M] \xleftarrow{\$} \{0,1\}^v$
return $\mathsf{F}^*[M]$;

$\mathsf{SimB}_{\mathcal{RO}^T}(M)$

If $\mathsf{F}^T[M] \neq \perp$ then $\mathsf{F}^T[M] \xleftarrow{\$} \{0,1\}^w$
return $\mathsf{F}^T[M]$;

$\mathsf{SimB}_{\mathcal{TO}}(y)$

If $\exists_1 M$ s.t. $\mathsf{F}^T[M] = y$ then **return** M
Otherwise **return** $\perp$

$\mathsf{SimB}_E(k, x)$

If $\mathsf{E}[k,x] = \perp$,
$y \xleftarrow{\$} \{0,1\}^b \backslash T^+[k]$
$\mathsf{E}[k,x] \leftarrow y, \mathsf{D}[k,y] \leftarrow x,$
$T^+[k] \xleftarrow{\cup} \{y\}$, $T^-[k] \xleftarrow{\cup} \{x\}$
return $\mathsf{E}[k,x]$

$\mathsf{SimB}_D(k, y)$

If $\mathsf{D}[k,y] = \perp$,
$x \xleftarrow{\$} \{0,1\}^b \backslash T^-[k]$;
$\mathsf{E}[k,x] \leftarrow y, \mathsf{D}[k,y] \leftarrow x,$
$T^+[k] \xleftarrow{\cup} \{y\}$, $T^-[k] \xleftarrow{\cup} \{x\}$
return $\mathsf{D}[k,y]$

Fig. 6. game $\mathbf{G}_1$ and adversary $\mathcal{B}$

simulator $\mathcal{S}$ in the IND-SIM game. In game $\mathbf{G}_1$, 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 $\mathbf{c}$. Thirdly, we transform $\mathbf{G}_1$ to game $\mathbf{G}_2$ so that the table of inputs and outputs of each oracle in $\mathcal{VO}$ (except $\mathcal{RO}_n$) for $\mathcal{A}_1$ is independent of the table for $\mathcal{A}_2$. In game $\mathbf{G}_2$, queries to oracles for $\mathcal{A}_2$ does not contain any information about that of $\mathcal{A}_1$. Thus, $\mathcal{A}_1$ cannot hand over any information to $\mathcal{A}_2$ with $\mathcal{VO}$. Finally, we estimate that bad events in $\mathbf{G}_2$ occurs only with negligible probability.

We denote $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_i)$ by the advantage of the adversary $\mathcal{A}$ when participating in experiment $\mathbf{G}_i$. It means $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_0) = \mathsf{Adv}^{\text{cda}}_{\mathcal{AE},F}(\mathcal{A}_1, \mathcal{A}_2)$.

Game $\mathbf{G}_1$: Ciphertext $\mathbf{c} \leftarrow \mathcal{E}^{\mathcal{RO}_n}(pk, \mathbf{m}_b, \mathbf{r})$ is replaced with outputs of a simulator $\mathcal{S}^{\mathcal{RO}_n}(pk, \omega)$ in the IND-SIM game. All other procedures are computed as the same way in $\mathbf{G}_0$.

Lemma 4. $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_0)| \leq \mathsf{Adv}^{\text{ind-sim}}_{\mathcal{AE},\mathcal{S},\mathcal{RO}_n}(\mathcal{B})$.

Proof. We show that if $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_0)|$ is non-negligible, for any simulator $\mathcal{S}$ we can construct an adversary $\mathcal{B}$ breaking IND-SIM security of $\mathcal{AE}$ in the RO model. Fig. 6 shows game $\mathbf{G}_1$, the construction of $\mathcal{B}$, and the simulation $\mathsf{SimB} = (\mathsf{SimB}_{\mathcal{RO}}, \mathsf{SimB}_{\mathcal{RO}^*}, \mathsf{SimB}_{\mathcal{RO}^T}, \mathsf{SimB}_{\mathcal{TO}}, \mathsf{SimB}_E, \mathsf{SimB}_D)$ of $\mathcal{VO}$ by $\mathcal{B}$ respectively. Note that $\mathcal{B}$ makes no RO queries, and $\mathcal{E}^{F.priv}(pk, \mathbf{m}_\beta, \mathbf{r})$ is executed with return value ignored. $\mathcal{B}$ simulates all queries to $\mathcal{VO}$ for $\mathcal{A}_1$ and $\mathcal{A}_2$ with simulation SimB. SimB is identical with the definition of $\mathcal{VO}$. Also, queries to $\mathcal{RO}_n$ by $\mathcal{E}$ is contained both in $\mathbf{G}_0$ and $\mathbf{G}_1$. Thus, $\mathcal{A}$ cannot distinguish game $\mathbf{G}_0$ and $\mathbf{G}_1$ from the simulation on the interface of $\mathcal{VO}$. If $\beta = 1$ in IND-SIM game, it is clear that all interfaces for $\mathcal{A}$ is exactly same as game $\mathbf{G}_0$. If $\beta = 0$ in IND-SIM game, it is clear that all interfaces for $\mathcal{A}$ is exactly same as game $\mathbf{G}_1$.

Therefore, if $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_0)|$ is non-negligible, $\mathcal{B}$ also breaks IND-SIM security of $\mathcal{AE}$. $\square$

Game $\mathbf{G}_2$: Outputs of $\mathcal{RO}_v^*, \mathcal{TRO}_w = (\mathcal{RO}_w^T, \mathcal{TO})$ and $\mathsf{IC}_{a,b} = (E, D)$ for $\mathcal{A}_1$ and for $\mathcal{A}_2$ are changed to be independent. That is, tables $\mathsf{F}^*, \mathsf{F}^T, \mathsf{E}$ and D are not preserved for $\mathcal{A}_1$ and $\mathcal{A}_2$. All other procedures are computed as the same way in $\mathbf{G}_1$.

Lemma 5. $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_1)| = 0$.

Proof. In game $\mathbf{G}_1$ and $\mathbf{G}_2$, ciphertext $\mathbf{c}$ does not give any information about $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ and queries to $\mathcal{VO}$ by $\mathcal{A}_1$ to $\mathcal{A}_2$. On queries to $\mathcal{RO}_n$, interfaces of $\mathcal{A}_2$ in $\mathbf{G}_1$ and $\mathbf{G}_2$ are identical. On queries to $\mathcal{RO}^*$ and $\mathcal{RO}^T$, $\mathcal{A}_2$ cannot find inconsistency even if $\mathcal{A}_1$ and $\mathcal{A}_2$ pose a common input to these oracles. On queries to $\mathcal{TO}$, E

and D, $\mathcal{A}_2$ may find inconsistency so that outputs of $\mathcal{TO}$ is inconsistent to inputs to $\mathcal{RO}^T$ by $\mathcal{A}_1$, or outputs of D is inconsistent to inputs to E by $\mathcal{A}_1$. However, since $\mathcal{A}_2$ does not have any information of obtained outputs of $\mathcal{RO}^T$ and E by $\mathcal{A}_1$, she still cannot find inconsistency. Therefore, $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_1)| = 0$. $\square$

We estimate $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2)$. The only way to win in game $\mathbf{G}_2$ is if $\mathcal{A}_2$ poses some message M to $\mathcal{RO}_n$ and M is also posed to $\mathcal{RO}_n$ by $\mathcal{E}$. The probability this event occurs can be bounded by $\frac{\nu q_{\mathcal{RO}}}{2^\mu}$ based on the fact that $\mathcal{A}_1$ is an mmr-source with min-entropy μ as Theorem 9.1 in [27]. Therefore, $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2) \leq \frac{\nu q_{\mathcal{RO}}}{2^\mu}$.

To conclude, we have $\mathsf{Adv}^{\text{cda}}_{\mathcal{AE},\mathcal{VO}}(\mathcal{A}_1, \mathcal{A}_2) \leq \mathsf{Adv}^{\text{ind-sim}}_{\mathcal{AE},\mathcal{S},\mathcal{RO}_n}(\mathcal{B}) + \frac{\nu q_{\mathcal{RO}}}{2^\mu}$. $\square$

References

1. Elena Andreeva, Atul Luykx, and Bart Mennink. Provable Security of BLAKE with Non-Ideal Compression Function, ePrint 2011/620.
2. Elena Andreeva, Bart Mennink, and Bart Preneel. On the Indifferentiability of the Grøstl Hash Function. In *SCN*, volume 6280 of *LNCS*, pages 88–105. Springer, 2010.
3. Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and Raphael C.-W. Phan. SHA-3 proposal BLAKE. Submission to NIST (Round 3). 2010.
4. Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and Efficiently Searchable Encryption. In *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552. Springer, 2007.
5. Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In *ASIACRYPT*, volume 5912 of *LNCS*, pages 232–249. Springer, 2009.
6. Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. In *CRYPTO*, volume 5157 of *LNCS*, pages 360–378. Springer, 2008.
7. Mihir Bellare and Thomas Ristenpart. Multi-Property-Preserving Hash Domain Extension and the EMD Transform. In *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 299–314. Springer, 2006.
8. Mihir Bellare and Phillip Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *EUROCRYPT*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 1994.
9. Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer, 1996.
10. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the Indifferentiability of the Sponge Construction. In *EUROCRYPT*, pages 181–197, 2008.
11. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak SHA-3 submission. Submission to NIST (Round 3). 2011.
12. Rishiraj Bhattacharyya, Avradip Mandal, and Mridul Nandi. Security Analysis of the Mode of JH Hash Function. In *FSE*, volume 6147 of *LNCS*, pages 168–191. Springer, 2010.
13. Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles. In *CRYPTO*, volume 5157 of *LNCS*, pages 335–359. Springer, 2008.
14. Donghoon Chang, Mridul Nandi, and Moti Yung. Indifferentiability of the Hash Algorithm BLAKE, ePrint 2011/623. 2011.
15. Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård Revisited: How to Construct a Hash Function. In *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer, 2005.
16. Ivan Damgård. A Design Principle for Hash Functions. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.
17. Yevgeniy Dodis, Thomas Ristenpart, and Thomas Shrimpton. Salvaging Merkle-Damgård for Practical Applications. In *EUROCRYPT (Full Version in ePrint 2009/177)*, volume 5479 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2009.
18. Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The Skein Hash Function Family. Submission to NIST (Round 3). 2010.
19. Benjamin Fuller, Adam O'Neill, and Leonid Reyzin. A Unified Approach to Deterministic Encryption: New Constructions and a Connection to Computational Entropy. ePrint 2012/005.
20. Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schäffer, and Søren S. Thomsen. Grøstl – a SHA-3 candidate. Submission to NIST (Round 3). 2011.

21. Shoichi Hirose, Je Hong Park, and Aaram Yun. A Simple Variant of the Merkle-Damgård Scheme with a Permutation. In *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 113–129. Springer, 2007.
22. Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.
23. Ralph C. Merkle. One Way Hash Functions and DES. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.
24. Ilya Mironov, Omkant Pandey, Omer Reingold, and Gil Segev. Incremental Deterministic Public-Key Encryption, (Full Version in ePrint 2012/047). In *EUROCRYPT*, 2012.
25. National Institute of Standards and Technology. Cryptographic Hash Algorithm Competition.
26. National Institute of Standards and Technoloty. FIPS PUB 180-3 Secure Hash Standard. In *FIPS PUB*, 2008.
27. Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with Composition: Limitations of the Indifferentiability Framework. In *EUROCRYPT (Full Version: ePrint 2011/339)*, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506. Springer, 2011.
28. Hongjun Wu. The Hash Function JH. Submission to NIST (Round 3). 2011.