On Indifferentiable Hash Functions in Multi-Stage Security Games

Yusuke Naito and Kazuki Yoneyama

Mitsubishi Electric Corporation and NTT Corporation

Abstract. Ristenpart, Shacham, and Shrimpton (EUROCRYPT 2011) demonstrated that for multi-stage security games, composability of indifferentiable hash functions does not sufficiently work. An open problem from their result is how to obtain multi-stage security when a random oracle (RO) is replaced with indifferentiable hash functions. In this paper, we positively solve this problem so that for a large class of public key encryption (PKE) scheme and ID-based encryption (IBE) scheme an important multi-stage security, the CDA security, is obtained even when the RO is replaced with important indifferentiable hash functions, Sponge, Prefix-free Merkle-Damgård, or chop Merkle-Damgård. Especially, Sponge is employed in the SHA-3 winner Keccak. 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* indifferentiability 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 PKE or IBE scheme satisfies the CPA security in the RO model, then there exists a CDA secure PKE or IBE scheme in the $\mathcal{VO}$ model. Combining these two results, we have that for a large class of PKE and IBE schemes the CDA security is guaranteed when the RO is replaced with a large class of practical hash functions.

Keywords. Indifferentiable hash function, reset indifferentiability, multi-stage security, Sponge, ChopMD, PFMD, deterministic PKE, hedged PKE, deterministic IBE, hedged IBE.

1 Introduction

The indifferentiability of Maurer, Renner, and Holenstein (MRH) [23] ensures the reducibility from one system to another system: Let $\mathcal{C}(\cdot)$ be a cryptosystem with access to a hash function H, denoted $\mathcal{C}(H)$. Then the MRH theorem is the following.

$$\begin{array}{l} H \text{ is indifferentiable from a random oracle } \mathcal{RO} \text{ (denoted } H \sqsubset \mathcal{RO}) \\ \Rightarrow \mathcal{C}(H) \text{ is at least as secure as } \mathcal{C}(\mathcal{RO}) \text{ (denoted } \mathcal{C}(H) \succ \mathcal{C}(\mathcal{RO})). \end{array}$$

The indifferentiability is important, because many practical cryptosystems e.g., RSA-OAEP [6] and RSA-PSS [7] are designed by the RO methodology, while the RO is instantiated by a hash function H such as SHA-1 and SHA-256 [28]. However, the Merkle-Damgård hash functions [17, 24] such as SHA-1 and SHA-256, are not indifferentiable from ROs [16]. So many indifferentiable (from a RO) hash functions have been proposed, e.g., Sponge [8] which is employed in the SHA-3 winner Keccak [9], Chop Merkle-Damgård (denoted ChopMD) [16], and prefix-free Merkle-Damgård (denoted PFMD) [16]. The indifferentiable security is thus an important criterion for the security of hash functions.

It has been widely believed that the equation of $\mathcal{C}(H) \succ \mathcal{C}(\mathcal{RO})$ ensures *any* security. However, Ristenpart, Shacham, and Shrimpton (RSS) [29] showed the following for some indifferentiable hash function H.

$$\exists\mathcal{C} \text{ such that } \mathcal{C}(\mathcal{RO}) \text{ is secure but } \mathcal{C}(H) \text{ is insecure (RSS result 1)}$$

where $\mathcal{C}(\cdot)$ is a hash based authentication scheme and the security game is a *multi-stage* security game. This security game is a two-stage security game where for n-bit (output length) hash function F, this game is defined as follows. In the first stage, for random messages M_1, M_2 of $2n$ bits, a first stage adversary A_1 derives the some state st of $2n$ bits. In the second stage, a 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 $z = F(M_1||M_2||C)$. Consider ChopMD $\text{chopMD}^h(M_1||M_2||C) = \mathit{chop}_n(h(h(h(IV, M_1), M_2), C))$ which is indifferentiable from a RO [16], where $h : \{0,1\}^{4n} \to \{0,1\}^{2n}$ is a RO, and $\mathit{chop}_n : \{0,1\}^{2n} \to \{0,1\}^n$ outputs the right n bits of the input. Clearly, the following adversary can win with probability 1 when $F = \text{chopMD}^h$. 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 the output of $\mathrm{chopMD}^h(M_1||M_2||C)$. On the other hand, when $F = \mathcal{RO}$, the probability that the adversary wins is negligible, since A_2 cannot receive several bits of M_1, M_2.

The RSS result 1 implies the following for any *multi-stage* security.[1]

$$H \sqsubset \mathcal{RO} \not\Rightarrow \mathcal{C}(H) \succ \mathcal{C}(\mathcal{RO}).$$

However, this does not imply that the multi-stage security of $\mathcal{C}(H)$ is broken. The RSS result 1 thus left the following open problem.

Can we prove the multi-stage security of $\mathcal{C}(H)$?

Ristenpart *et al.* [29] solved this problem for Chosen-Distribution Attack (CDA) security and NMAC hash function [18] (RSS result 2). The NMAC hash function is employed in the SHA-3 finalist Skein [19]. The CDA security notion is an important multi-stage security notion, which is the security goal for deterministic, efficiently searchable [2, 4, 10, 20, 25], and hedged [3] public key encryption (PKE), wherein there are several PKE schemes which are proven in the RO model [2, 3]. For the CDA secure PKE schemes EwH [2] and REwH1 [3] (in the RO model), they directly proved the CDA security of these PKE schemes using NMAC.

The RSS result 2 motivates us to solve the following problem which is still open.

Can we prove the multi-stage security for other indifferentiable hash functions?

1.1 Our Contributions

In this paper, we answer the question for important indifferentiable hash functions, Sponge, ChopMD, and PFMD. Especially, it is important to discuss about Sponge, because Sponge is employed in the SHA-3 winner Keccak and will be published as a standard hash function (FIPS) [27]. Also, ChopMD and PFMD are important, because these hash functions are the SHA-3 finalists [1, 21, 31]: Grøstl [21], and JH [31] have the ChopMD(-like) structure and BLAKE has the PFMD(-like) structure. We clarify that a large class of PKE and ID-based encryption (IBE) can be CDA secure even when a RO is instantiated with such a large class of hash functions. The previous result covers only PKE (i.e., not mentioned about IBE) for instantiation with only NMAC. The class of PKE and IBE, which we cover, is CPA secure schemes in the RO model. There are a lot of practical CPA secure PKE and IBE schemes in the RO model; thus, such a large class of schemes benefits from our result.

An essential difference between our result and the previous result is as follows: As the RSS result 2, we could "directly" prove the security of cryptosystems $\mathcal{C}_1, \ldots, \mathcal{C}_j$ in each hash function $H_1, \ldots, H_i$, while this approach needs to prove "many" cryptosystems $\mathcal{C}_1(H_1)$,. . .,$\mathcal{C}_1(H_i)$,. . .,$\mathcal{C}_j(H_1)$,. . ., $\mathcal{C}_j(H_i)$.[2] Moreover, when designing a hash function, we have to care the structures of all cryptosystems (for multi-stage security), and when designing a cryptosystem for multi-stage security, we have to care the structures of all hash functions. To avoid many proofs, we propose a "modular" approach, which uses a new security notion, called *reset indifferentiability from Versatile Oracle* (denoted $\mathcal{VO}$). The new security notion uses the reset indifferentiability framework [29].

(Reset) Indifferentiability [29]. Let H^f be a hash function which uses an ideal primitive f and let F' be another ideal primitive. Then the reset indifferentiability ensures the following for any cryptosystem $\mathcal{C}$ and *any* (single-stage or multi-stage) security.

H^f is reset indifferentiable from F' (denoted $H^f \sqsubset_r F'$) $\Rightarrow \mathcal{C}(H^f) \succ \mathcal{C}(F')$ (RSS theorem).

[1] Note that the MRH theorem guarantees any *single-stage* security.

[2] In [18], the multi-stage security is directly proven from the NMAC structure $H(M) = g(h(M))$, where g is a fixed input length RO and h is any Preimage Aware Function which does not use g. We note that this approach restricts the hash structure as $g(h(M))$, which does not cover many hash functions, e.g., Sponge, ChopMD, and PFMD. On the other hand, our approach covers these hash functions, Sponge, ChopMD, and PFMD.

We explain the original [23] and reset [29] indifferentiability. The original indifferentiable security game is that a distinguisher A converses either with (H^f, f) or $(F', S^{F'})$. S is a simulator which simulates f such that S is consistent with F'. If the probability that the distinguisher A hits the conversing world is small, then $H^P \sqsubset \mathcal{RO}$. In the reset indifferentiable security game, A can reset the initial state of the simulator at arbitrary times.

Our Approach and Our Results. To prove indifferentiable from RO security such as Sponge, ChopMD, and PFMD, S needs to record query-responses of S [8, 16, 14]. However, in the reset indifferentiable security game, query-responses are eliminated. We thus define $\mathcal{VO}$ such that recording the query-response history is "outsourced". The $\mathcal{VO}$ consists of a RO and auxiliary oracles. The auxiliary oracles are used to record the query-response history of S. The $\mathcal{VO}$ thus enables to construct S which does not update the internal state and which is unaffected by the reset function. Our approach using the indifferentiability from $\mathcal{VO}$ is as follows.

1. Prove that $H^f \sqsubset_r \mathcal{VO}$.
2. Prove that $\mathcal{C}(\mathcal{VO})$ is secure.

Then H^f can be plugged into $\mathcal{C}(\cdot)$ and the (multi-stage) security is ensured by the RSS theorem. This approach can divide the security proof of $\mathcal{C}(H^f)$ into the hash proof and the proof of the cryptosystem. When designing a hash function, we don't have to care the structures of cryptosystems, and when designing a cryptosystem, we don't have to care the structures of hash functions. Thus the modular approach can avoid many proofs of the direct proof approach.

We prove that

1. *Sponge* $\sqsubset_r \mathcal{VO}$, chopMD $\sqsubset_r \mathcal{VO}$, and PFMD $\sqsubset_r \mathcal{VO}$, and
2. Any CPA secure PKE and IBE scheme in the RO model can be converted to be CDA secure in the $\mathcal{VO}$ model.

For PKE schemes, we show that a scheme is CDA secure in the $\mathcal{VO}$ model if the scheme satisfies an weak property, called the IND-SIM security. Since [29] shows EwH [2] and $\mathsf{REwH1}$ [3] are IND-SIM secure if an underlying PKE scheme is CPA secure in the RO model, we obtain CDA secure PKE in the $\mathcal{VO}$ model by combining these. For IBE schemes, we propose a generic compiler of IBE, $\mathsf{IDREwH1}$. We show that $\mathsf{IDREwH1}$ is ID-based CDA (ID-CDA) secure in the $\mathcal{VO}$ model if an underlying IBE scheme is ID-based CPA (ID-CPA) secure in the RO model. As far as we know, our result of IBE is the first explicit formulation and construction of hedged IBE. It may be an independent interest.

1.2 Paper Organization.

In Section 2, we introduce some notation and explain the (reset) indifferentiability framework. In Section 3, we propose a new security notion, Reset Indifferentiability from $\mathcal{VO}$. In Section 4, we show that Sponge, ChopMD, and PFMD are reset indifferentiable from $\mathcal{VO}$s. In Section 5, we show that there exist a large class of CDA secure PKE and IBE schemes in the $\mathcal{VO}$ model.

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 $x \xleftarrow{\$} X$ 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. For sets A and C, $C \xleftarrow{\cup} A$ 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 b-bit value x, $x[i, j]$ is the value from (left) i-th bit to (left) j-th bit where $1 \leq i \leq j \leq b$. For example, let $x = 01101001$, $x[3, 5] = 101$. 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 $\mathbf{x}[1]|| \ldots ||\mathbf{x}[|\mathbf{x}|]$.

```
Algorithm $Sponge^P(M)$
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
5     $s = P(s \oplus (M_i || 0^c))$;
6 return $s[1, n]$;
```

Fig. 1. Sponge

```
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 $x[s+1, s+n]$;
```

Fig. 2. Chop Merkle-Damgård

```
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 $x$;
```

Fig. 3. Prefix-free Merkle-Damgård

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

(Reset) Indifferentiability [23, 29]. In the reset indifferentiability [29], 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 $F.priv$. Let H^f be a hash function that utilizes an ideal primitive f. The interfaces of H^f are defined by $H^f.priv = H^f$ and $H^f.pub = f$.

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.

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

$S.Rst$ takes no input and 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* (single-stage or multi-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 [29].

When $S.Rst$ and nop are removed from the reset indifferentiable security game, it is equal to the original indifferentiable security game [23]. 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. The original indifferentiability guarantees only *single* stage security [23].

Hereafter, we call the F_1 world "Real World" where A interacts with $(F_1.priv, F_1.pub)$, and the F_2 world "Ideal World" where A interacts with $(F_2.priv, F_2.pub)$. We call the oracle $F_1.priv/F_2.priv$ "Left Oracle" (denoted L) and the oracle $F_1.pub/S$ "Right Oracle" (denoted R). Thus distinguisher A interacts with (L, R) (and $nop/S.Rst$ in the reset indifferentiability). We call a query to L a "left query" (or L query). Similarly we call a query to R a 'right query" (or R query).

3 Definitions for Hash Functions

In this section, we give descriptions of hash functions, Sponge [8], chop Merkle-Damgård (ChopMD) [16], and prefix-free Merkle-Damgård (PFMD) [16]. Also, we define Sponge Graph and Merkle-Damgård Graph, which are used in the proofs of hash functions in this paper.

3.1 Descriptions of Hash Functions

Sponge. Let P be an encryption of a (fixed key) blockcipher (or a permutation) of d bits.[3] The hash function $Sponge^P : \{0,1\}^* \rightarrow \{0,1\}^n$ is defined in Fig. 1 such that $n < d$.[4] Let $c = d - n$. $\mathsf{pad}_S : \{0,1\}^* \rightarrow (\{0,1\}^n)^*$

[3] Since Sponge is in the "fixed key" setting, we don't care the key and the key length. Thus we omit the key in P.

[4] Note that if the output length (denoted l) is smaller than n, the output length is achieved by returning $s[1, l]$ at the step 6. Also note that the Sponge hash function of Fig. 1 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. 1. We conjecture that the reset indifferentiable security of the general Sponge hash function can be proven by extending the analysis of the fixed output length case.

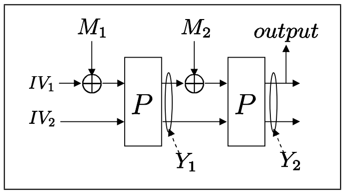

Fig. 4. Figure of Sponge

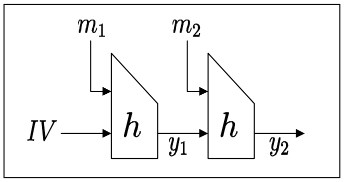

Fig. 5. Figure of Merkle-Damgård

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 the length of $M||1||0^i$ is a multiple of n.

ChopMD. Let h be a compression function which maps a value of $d+n+s$ bits to a value of $n+s$ bits. The hash function $\text{chopMD}^h : \{0,1\}^* \to \{0,1\}^n$ is defined in Fig. 2. $\mathsf{pad}_c : \{0,1\}^* \to (\{0,1\}^d)^*$ is an injective padding function such that its inverse is efficiently computable. IV is a constant value of $n+s$ bits.

PFMD. Let h be a compression function which maps a value of $d+n$ bits to a value of n bits. $\mathsf{pfpad} : \{0,1\}^* \to (\{0,1\}^d)^*$ is an injective prefix-free padding function such that for any different two values M, M' $\mathsf{pfpad}(M)$ is not a prefix of $\mathsf{pfpad}(M')$ and its inverse is efficiently computable. IV is a constant value of n bits.

3.2 Graph Representations for Sponge and Merkle-Damgård

Sponge Graph. The proof of the indifferentiability from a $\mathcal{RO}$ of Sponge is in the (fixed key) ideal cipher model (random permutation model) [8], where P is an encryption oracle of an (fixed key) ideal cipher (or a random permutation) and P^{-1} be a decryption oracle with fixed key (or an inverse oracle). The proof of the reset indifferentiability of Sponge is thus in the (fixed key) ideal cipher model. In the real world, the left oracle $L = \mathit{Sponge}^P$, and the right oracles $(R_F, R_I) = (P, P^{-1})$ (F means "forward" and I means "inverse"). In the ideal world, $(L, R_F, R_I) = (\mathcal{RO}, S_F, S_I)$ where S_F is a simulator for P and S_I is a simulator for P^{-1}.

We define a graph G_S, which is initialized with the single node IV. Edges and nodes in this graph are defined by right query-responses which follow the Sponge structure. The nodes are chaining values and the edges are message blocks. For example, if $(X_1, Y_1), (X_2, Y_2)$ are query-responses of R_F or R_I such that $X_1[n+1, d] = IV_2$ and $Y_1[n+1, d] = X_2[n+1, d]$ then IV, Y_1, Y_2 are the nodes of G_S and M_1, M_2 are the edges where $M_1 = IV_1 \oplus X_1[1, n]$ and $M_2 = Y_1[1, n] \oplus X_2[1, n]$. We denote the path by $IV \xrightarrow{M_1} Y_1 \xrightarrow{M_2} Y_2$ or $IV \xrightarrow{M_1||M_2} Y_2$ (Fig. 4 may help to understand the graph). We call a path following the Sponge structure "Sponge path".

Merkle-Damgård Graph. In the proofs of the indifferentiability from $\mathcal{RO}$s of ChopMD and PFMD [14–16], the compression function h is a (fixed input length) RO. The proofs of the reset indifferentiability of ChopMD and FPMD are thus in the RO model. So in the real world, the left oracle $L = \text{chopMD}^h/\text{PFMD}^h$, and the right oracle $R = h$. In the ideal world, $(L, R) = (\mathcal{RO}, S)$ where S is a simulator for h.

We define a graph G_{MD}, 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 $(IV, m_1, y_1), (y_1, m_2, y_2)$ are query-responses of R, IV, y_1, y_2 are the nodes of G amd 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 the path).

This graph is used in the proofs of ChopMD (Theorem 2) and PFMD (Theorem 3). For a MD path $IV \xrightarrow{M^*} y$, if $\exists M$ s.t. $\mathsf{pfpad}(M) = M^*$, then we call the MD path "PFMD path".

4 Reset Indifferentiability from Versatile Oracle

Indifferentiability [23] does not guarantee multi-stage security [29]. Hash functions relying on the indifferentiability are, for example, Sponge [8], ChopMD [16], and PFMD [16]. In this section, we introduce a new security notion, called *reset indifferentiability from Versatile Oracle* (denoted $\mathcal{VO}$) to guarantee multi-stage security when using the indifferentiable hash function, e.g, Sponge, ChopMD, and PFMD.

4.1 On the Indifferentiable Security Proof

Before defining reset indifferentiability from $\mathcal{VO}$, we recall the proof of indifferentiability from $\mathcal{RO}$. To ensure indifferentiable from $\mathcal{RO}$ security, the query-responses of S have to be satisfy two conditions. In this subsection, we explain the two conditions. After this subsection, we point out that the conditions cannot be achieved on the reset indifferentiable from $\mathcal{RO}$ security game. Then we propose $\mathcal{VO}$ so that the conditions can be achieved.

Let H^f be a target hash function and f is its underlying primitive. In the real world $(L, R) = (H^f, f)$, outputs of L are calculated by using R, while in the ideal world $(L, R) = (\mathcal{RO}, S^{\mathcal{RO}})$, outputs of L are calculated without using R. Thus S should simulate the relation between H^f and f. Also since for a repeated query to f the same value is responded, S should return the same value for a repeated query. Therefore, the following two conditions are required on query-responses of S to ensure indifferentiable from $\mathcal{RO}$ security.

- **Condition 1:** For a repeated query $S(x)$ where y was responded, the same value y must be responded.
- **Condition 2:** S has to be consistent with $\mathcal{RO}$ as well as (H^f, f).

In the indifferentiable security game, the condition 1 can be achieved by recording all query-responses.

The condition 2 depends on the hash structure H. For example, Merkle-Damgård does not achieve the condition 2 by the length extension attack [16]. Sponge, ChopMD, and PFMD are designed so that the condition 2 can be satisfied. Hereafter, we give an example of the condition 2 for Sponge.

Sponge for Condition 2. We omit the padding function (the step 1 of Fig. 1) to simply the discussion. Thus the input of $Sponge^P$ is a multiple of n. The condition 2 is that S has to be consistent with $\mathcal{RO}$. This means that since in the real world for any sponge path $IV \xrightarrow{M} Y$ the equation $L(M) = Y[1, n]$ holds, this equation should be hold in the ideal world.

Here, we explain that in the ideal world S can be constructed with the consistency. We consider the case that there is a path $IV \xrightarrow{M_1} Y_1$ and the query $S_F(X_2)$ is made such that $X_2[n+1, d] = Y_1[n+1, d]$. In this case, the response Y_2 should be such that $Y_2[1, n] = L(M_1||M_2)$, where $M_2 = X_2[1, n] \oplus Y_1[1, n]$. Since the graph G_S consists of right query-responses (query-responses for S), S can find the path $IV \xrightarrow{M_1} Y_1$ by using X_2 (due to the fact that $X_2[n+1, d] = Y_1[n+1, d]$). Then the output of $L(M_1||M_2)$ can be obtained.[5] S can thus define Y_2 such that the path $IV \xrightarrow{M_1} Y_1 \xrightarrow{M_2} Y_2$ satisfies the relation $Y_2[1, n] = L(M_1||M_2)$. Thus we can construct S which is consistent with $\mathcal{RO}$.[6]

4.2 The Conditions on Reset Indifferentiability

Problems. To satisfy these conditions, S has to record the query-responses. However, in the reset indifferentiable from $\mathcal{RO}$ setting, $S.Rst$ eliminates the query-responses. Thus, a new methodology to record the query-responses (and paths) is required.

[5] Note that if there are two collision paths, $IV \xrightarrow{M_1} Y_1$ and $IV \xrightarrow{M_1^*} Y_1$, the simulator cannot find a valid path. In this case, the simulator cannot ensure the consistency. However, this collision probability is negligible which was proven in [8].

[6] We note that there is a case that when the query $S_F(X_1)$ was made, the path $IV \xrightarrow{M_1} Y_1$ is not defined. After this query, this path is defined. Also note that there is a case that the response of a R_I query $S_I(Y_2)$ connects with the path $IV \xrightarrow{M_1} Y_1$. In these cases, S cannot ensure the consistency. But these probabilities are negligible which was proven in [8].

```
$\mathcal{RO}_n(M)$
    if $\mathsf{F}[M] = \perp$, $\mathsf{F}[M] \xleftarrow{\$} \{0,1\}^n$;
    return $\mathsf{F}[M]$;

$\mathcal{RO}^*_v(M)$
    If $\mathsf{F}^*[M] = \perp$, $\mathsf{F}^*[M] \xleftarrow{\$} \{0,1\}^v$;
    return $\mathsf{F}^*[M]$;

$\mathcal{RO}^T_w(M)$
    if $\mathsf{F}^T[M] = \perp$ then $\mathsf{F}^T[M] \xleftarrow{\$} \{0,1\}^w$;
    return $\mathsf{F}^T[M]$;

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

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

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

Fig. 6. Versatile Oracle $\mathcal{VO}$

New Methodology. We define a new oracle Versatile Oracle (denoted $\mathcal{VO}$) which consists of a random oracle $\mathcal{RO}$ and auxiliary oracles which are used to record the query-responses (and paths) of S. Here, we discuss about auxiliary oracles to achieve the conditions.

We first consider the condition 1. The underlying primitives of Sponge is an (fixed key) ideal cipher and the underlying primitive of ChopMD and PFMD is a (fixed input length) RO. We thus define an ideal cipher IC and a random oracle $\mathcal{RO}^*$ as the auxiliary oracles. Then we can outsource defining responses of S in the auxiliary oracles. For example, consider the case that S simulates a fixed input length RO. S satisfying the condition 1 can be constructed as follows: For a query $S(x)$, S returned the response of $\mathcal{RO}^*(x)$. Then for the repeated query $S(x)$, S returns the response of $\mathcal{RO}^*(x)$. So a simulator can be constructed such that the condition 1 is satisfied. Note that query-responses for the condition 1 are not related with the condition 2. Query-responses related with the condition 2 are discussed as follows.

As the example of Sponge for the condition 2, for query $S(X_2)$ S has to find the path $IV \xrightarrow{M_1} Y_1$ where $X_2[n+1,d] = Y_1[n+1,d]$. To achieve the condition 2, an oracle recording paths is required. Traceable Random Oracle (denoted TRO) proposed by Naito *et al.* [26] achieves this requirement. A TRO consists of a RO (denoted $\mathcal{RO}^T$) and a Traceable Oracle (denoted $\mathcal{TO}$). For a query $\mathcal{TO}(y)$, if the query $\mathcal{RO}^T(M)$ was made such that the response is y, $\mathcal{TO}$ returns M. For a path $IV \xrightarrow{M_1} Y_1$, S defines Y_1 such that $Y_1[n+1,d] = \mathcal{RO}^T(M_1)$. Then S can obtain M_1 by the query $\mathcal{TO}(X_2[n+1,d])$. Finally, S defines the response Y_2 so that $Y_2[1,n] = L(M_1||M_2) = \mathcal{RO}(M_1||M_2)$. We can thus outsource recording paths in a TRO and a simulator can be constructed such that the condition 2 can be satisfied.

We thus define the auxiliary oracles which consists of an ideal cipher IC, a RO $\mathcal{RO}^*$, and a TRO $\mathcal{TRO}$. In the next subsection, we give a concrete implementation of $\mathcal{VO}$.

4.3 Implementation of $\mathcal{VO}$

$\mathcal{VO}$ consists of a RO $\mathcal{RO}_n$, a RO $\mathcal{RO}^*_v$, a TRO $\mathcal{TRO}_w$, 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. 6.

$\mathcal{RO}_n$ is shown in Fig. 6 (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. 6 (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. 6 (Center) which consists of a RO $\mathcal{RO}^T_w$ and a Trace Oracle $\mathcal{TO}$. The input length of $\mathcal{RO}^T_w$ is arbitrary. The output length of $\mathcal{RO}^T_w$ and the input length of $\mathcal{TO}$ are w bits, and F^T is a (initially everywhere $\perp$) table. Note that w is defined in each hash function.

$\mathsf{IC}_{a,b}$ can be implemented as Fig. 6 (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, 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,

$\mathsf{E}[k,x] \leftarrow y, \mathsf{D}[k,y] \leftarrow x, T^+[k] \overset{\cup}{\leftarrow} \{y\}$ and $T^-[k] \overset{\cup}{\leftarrow} \{x\}$. Note that the a, b, are defined in each hash function.

4.4 Reset Indifferentiability from $\mathcal{VO}$

Let H^f be a target hash function and f is an underlying primitive. Then the advantage of the reset indifferentiability from a $\mathcal{VO}$ is defined as follows.

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff},\mathcal{VO}}_{H^f,S}(A) = |\Pr[A^{H^f,f,nop} \Rightarrow 1] - \Pr[A^{\mathcal{RO}_n,S^{\mathcal{VO}.pub},S.Rst} \Rightarrow 1]|.$$

The reset indifferentiable theorem ensures that if H^f is reset indifferentiable from a $\mathcal{VO}$, any security of any cryptosystem is preserved when $\mathcal{VO}$ is replaces with H^f.

5 Reset Indifferentiability for Hash Functions

In this section, we show that Sponge [8], ChopMD [16], and PFMD [16] are reset indifferentiable from $\mathcal{VO}$s. Then, these hash functions can be used as $\mathcal{VO}$s (in multi-stage security games). In the next section, we show that there exists a large class of CDA secure PKE and IBE schemes in the $\mathcal{VO}$ model.

5.1 Reset Indifferentiability for Sponge

We define the parameter of $\mathcal{VO}$ as $w = c$ and $b = d$. We don't care the key length a, since Sponge uses an ideal cipher with a fixed key k^*. We denote $E(k^*, \cdot)$ by $\mathcal{P}(\cdot)$ and $D(k^*, \cdot)$ by $\mathcal{P}^{-1}(\cdot)$. Thus, $\mathcal{P}$ is a random permutation $\mathcal{P}$ of d bits and $\mathcal{P}^{-1}$ is its inverse oracle. Note that in this proof, $\mathcal{RO}^*_v$ are not used. Thus, in this case, $\mathcal{VO}.priv = \mathcal{RO}_n$ and $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{TRO}_c, \mathcal{P}, \mathcal{P}^{-1})$.

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

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

where $\mathcal{A}$ can make at most q_L, q_F and q_I queries to left $L = Sponge^P/\mathcal{RO}_n$ 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 $3q$ queries and runs in time $\mathcal{O}(q)$. ♦

As the discussion in Subsection 4.2, we can construct a simulator satisfying the conditions 1 and 2. So, we can prove that Sponge is reset indifferentiable from $\mathcal{VO}$. The proof is shown in Appendix A.

5.2 Reset Indifferentiability for ChopMD

We define the parameter of $\mathcal{VO}$ as $w = s$ and $v = n + s$. Note that $\mathsf{IC}_{a,b}$ is not used. Thus, in this case, $\mathcal{VO} = (\mathcal{RO}_n, \mathcal{RO}^*_{s+n}, \mathcal{TRO}_s)$.

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

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff},\mathcal{VO}}_{\mathrm{chopMD}^h,S}(\mathcal{A}) \leq \frac{3q_R(q_R-1)}{2^{s+1}} + \frac{q_R(q_R+3)}{2^{n+s+1}} + \frac{\sigma(\sigma+1)}{2^{s+n}}$$

where $\mathcal{A}$ 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 $3q_R$ queries and runs in time $\mathcal{O}(q_R)$. ♦

Similar to the reset indifferentiability for Sponge, a simulator can be constructed so that the conditions 1 and 2 are satisfied. The condition 1 can be achieved by using $\mathcal{RO}^*_{s+n}$. The condition 2 can be achieved by using $\mathcal{TRO}_s$ which records paths. The proof is given in Appendix B.

5.3 Reset Indifferentiability for PFMD

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 $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{RO}_n^*, \mathcal{TRO}_n)$.

Theorem 3. *There exists a simulator S such that for any distinguisher $\mathcal{A}$, the following holds,*

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

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

The PFMD case is also similar to the reset indifferentiability for Sponge. A simulator can be constructed so that the conditions 1 and 2 are satisfied. The condition 1 can be achieved by using $\mathcal{RO}_n^*$. The condition 2 can be achieved by using $\mathcal{TRO}_n$ which records paths. The proof is given in Appendix C.

Remark 1. EMD [5] and MDP [22] are designed from the same design spirit as PFMD, 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.

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

In this section, we show cryptographic primitives satisfying multi-stage security in the $\mathcal{VO}$ model. Specifically, we show that for any PKE scheme, the non-adaptive CDA security [3] (including the PRIV security [2]) in the $\mathcal{VO}$ model is obtained by assuming an weak property, IND-SIM security in the RO model. Also, we show that a generic conversion of IBE scheme to satisfy the non-adaptive ID-based CDA (ID-CDA) security in the $\mathcal{VO}$ model. The previous work [29] showed the non-adaptive CDA security for PKE schemes based on the same assumption (IND-SIM) with a specific structured preimage aware hash function [18]. There was no mention about IBE. Our work focuses on how we obtain CDA secure PKE schemes and ID-CDA secure IBE schemes with large class of hash functions. For PKE, we show that if a PKE scheme is IND-SIM secure in the RO model, then it is CDA secure in the $\mathcal{VO}$ model. It is shown that EwH [2] and $\mathsf{REwH1}$ [3] satisfy IND-SIM security [29]; thus, any CPA secure PKE scheme can be converted into IND-SIM secure scheme. For IBE, we show a generic conversion of IBE, called $\mathsf{IDREwH1}$ which is an analogy of $\mathsf{REwH1}$, and is ID-CDA secure in the $\mathcal{VO}$ model if underlying IBE scheme is ID-CPA secure in the RO model. Therefore, any CPA secure PKE and ID-CPA secure IBE in the RO model can be converted into CDA secure PKE and ID-CDA secure IBE in the $\mathcal{VO}$ model. Combining with our results on reset indifferentiable hash functions in Sec. 5, such a scheme remains CDA or ID-CDA security even if $\mathcal{VO}$ is replaced with these functions. Reset indifferentiable hash functions which we prove cover wide and practical types of functions including SHA-3 though the previous result in [29] does not salvage SHA-3.

6.1 CDA Secure PKE in the $\mathcal{VO}$ 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 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 [29]). Fig. 7 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 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 $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$

```
$\mathrm{CDA}^{\mathcal{A}_1,\mathcal{A}_2}_{\mathcal{AE},F}$
$\beta \xleftarrow{\$} \{0,1\}$
$(pk, sk) \xleftarrow{\$} \mathsf{Gen}$
$(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow \mathcal{A}_1^{F.pub}$
$\mathbf{c} \leftarrow \mathsf{Enc}^{F.priv}(pk, \mathbf{m}_\beta; \mathbf{r})$
$\beta' \leftarrow \mathcal{A}_2^{F.pub}(pk, \mathbf{c})$
return $(\beta = \beta')$
```

```
$\mathrm{IND\text{-}SIM}^{\mathcal{B}}_{\mathcal{AE},\mathcal{S},F}$
$\beta \xleftarrow{\$} \{0,1\}$
$(pk, sk) \xleftarrow{\$} \mathsf{Gen}$
$\beta' \leftarrow \mathcal{B}^{\mathsf{RoS}, F.pub}(pk)$
return $(\beta = \beta')$

$\mathsf{RoS}(m, r)$
If $\beta = 1$ then return $\mathsf{Enc}^{F.priv}(pk, m; r)$
Otherwise return $\mathcal{S}^{F.priv}(pk, |m|)$
```

Fig. 7. CDA game and IND-SIM game

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 [3], 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 $\mathsf{Adv}^{\mathrm{priv}}_{\mathcal{AE},F}(\mathcal{A}_1, \mathcal{A}_2)$ which is equal to the CDA advantage with $\rho = 0$. Since the PRIV security is for deterministic encryption, our result also covers a class of deterministic encryption.

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

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

As noted in [29], 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 $\mathcal{VO}$ Model. 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 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*

$$\mathsf{Adv}^{\mathrm{cda}}_{\mathcal{AE},\mathcal{VO}}(\mathcal{A}_1, \mathcal{A}_2) \leq \mathsf{Adv}^{\text{ind-sim}}_{\mathcal{AE},\mathcal{S},\mathcal{RO}_n}(\mathcal{B}) + q_{\mathcal{RO}} \cdot \mathsf{maxpk}_{\mathcal{AE}} + \frac{q_{\mathcal{RO}} + 4q^2_{\mathcal{RO}^*} + 4q^2_{\mathcal{RO}^T} + 4q^2_{\mathcal{TO}} + 4q^2_E + 4q^2_D}{2^\mu}.$$

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

```
ID-CPA$^{\mathcal{B}}_{\mathcal{IBE},F}$
$\beta \xleftarrow{\$} \{0,1\}$
If selective-ID setting
  $id^* \leftarrow \mathcal{B}^{F.pub}$
$(params, msk) \xleftarrow{\$} \mathsf{IBE.Setup}$
If selective-ID setting
  $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{B}^{F.pub,\mathsf{IBE.Gen}(\cdot)}(params)$
If full-ID setting
  $(\mathbf{m}_0, \mathbf{m}_1, id^*) \leftarrow \mathcal{B}^{F.pub,\mathsf{IBE.Gen}(\cdot)}(params)$
$\mathbf{r} \xleftarrow{\$} (\{0,1\}^{\rho})^{l}$
$\mathbf{c} \leftarrow \mathsf{IBE.Enc}^{F.priv}(params, id^*, \mathbf{m}_\beta; \mathbf{r})$
$\beta' \leftarrow \mathcal{B}^{F.pub,\mathsf{IBE.Gen}(\cdot)}(\mathbf{c})$
return $(\beta = \beta')$
```

```
ID-CDA$^{\mathcal{A}_1,\mathcal{A}_2}_{\mathcal{IBE},F}$
$\beta \xleftarrow{\$} \{0,1\}$
If selective-ID setting
  $id^* \leftarrow \mathcal{A}_1^{F.pub}$
$(params, msk) \xleftarrow{\$} \mathsf{IBE.Setup}$
If selective-ID setting
  $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow \mathcal{A}_1^{F.pub,\mathsf{IBE.Gen}(\cdot)}$
If full-ID setting
  $(\mathbf{m}_0, \mathbf{m}_1, id^*, \mathbf{r}) \leftarrow \mathcal{A}_1^{F.pub,\mathsf{IBE.Gen}(\cdot)}$
$\mathbf{c} \leftarrow \mathsf{IBE.Enc}^{F.priv}(params, id^*, \mathbf{m}_\beta; \mathbf{r})$
$\beta' \leftarrow \mathcal{A}_2^{F.pub,\mathsf{IBE.Gen}(\cdot)}(params, id^*, \mathbf{c})$
return $(\beta = \beta')$
```

Fig. 8. ID-CPA and ID-CDA game

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 $\mathcal{RO}_n$ returns a random value for a message posed by Enc. In game $\mathbf{G}_1$, outputs of $\mathcal{RO}_n$ does not contain any information about computations to generate the challenge ciphertext. Thirdly, we transform $\mathbf{G}_1$ to game $\mathbf{G}_2$ so that ciphertext $\mathbf{c}$ is generated from a simulator $\mathcal{S}$ in the IND-SIM game. In game $\mathbf{G}_2$, 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 $\mathbf{G}_2$ to game $\mathbf{G}_3$ 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 $\mathcal{A}_2$ according to the output of $\mathcal{A}_1$. In game $\mathbf{G}_3$, queries to 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 $\mathcal{VO}$. Thus, the advantage of $\mathcal{A}_2$ in $\mathbf{G}_3$ is nothing.

The proof of Theorem 4 is shown in Appendix D.

6.2 ID-CDA Secure IBE in the $\mathcal{VO}$ Model

ID-based Encryption (IBE). An ID-based encryption scheme $\mathcal{IBE} = (\mathsf{IBE.Setup}, \mathsf{IBE.Gen}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ consists of four algorithms. Setup algorithm $\mathsf{IBE.Setup}$ outputs public parameter *params* and master secret key *msk*. Key generation algorithm $\mathsf{IBE.Gen}$ takes public parameter *params*, master secrete key *msk* and ID *id*, and outputs secret key *sk* for *id*. Encryption algorithm $\mathsf{IBE.Enc}$ takes public parameter *params*, ID *id*, plaintext m, and randomness r, and outputs ciphertext c. Decryption algorithm $\mathsf{IBE.Dec}$ takes public parameter *params*, 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 $\mathsf{IBE.Enc}(params, id, \mathbf{m}; \mathbf{r})$ the vector $(\mathsf{IBE.Enc}(params, id, \mathbf{m}[1]; \mathbf{r}[1]), \ldots, \mathsf{IBE.Enc}(params, id, \mathbf{m}[l]; \mathbf{r}[l]))$. We say that $\mathcal{IBE}$ is deterministic if $\mathsf{IBE.Enc}$ is deterministic.

ID-based CPA and CDA Security. We define the ID-CPA and the (non-adaptive) ID-CDA security. The ID-CPA security is a standard one [11–13] except that an adversary can pose multiple challenge plaintext pairs. It is known that the CPA game with multiple challenge is polynomial-time reducible to the game with single challenge. Let $\mathcal{CH}$ be the challenger of the ID-CPA game. The ID-CDA security is based on the CDA security. Fig. 8 illustrates the ID-CPA game and the non-adaptive ID-CDA game in the CPA case for $\mathcal{IBE}$ using a functionality F. As the CDA security, the ID-CDA adversary $\mathcal{A}_1$ is a (μ, ν)-mmr-source. The advantage for an ID-CPA adversary $\mathcal{B}$ against scheme $\mathcal{IBE}$ using a functionality F is defined by

$$\mathsf{Adv}^{\text{id-cpa}}_{\mathcal{IBE},F}(\mathcal{B}) = 2 \cdot \Pr[\text{ID-CPA}^{\mathcal{B}}_{\mathcal{IBE},F} \Rightarrow \mathsf{true}] - 1.$$

The advantage for an ID-CDA adversary $(\mathcal{A}_1, \mathcal{A}_2)$ against scheme $\mathcal{IBE}$ using a functionality F is defined by

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

Hedged ID-based Encryption $\mathsf{IDREwH1}$**.** We show an example of ID-CDA secure hedged IBE, $\mathsf{IDREwH1}$. The proposed scheme is a simple extension of $\mathsf{REwH1}$ [3].

Let $\mathcal{IBE}_r = (\mathsf{IBE.Setup}_r, \mathsf{IBE.Gen}_r, \mathsf{IBE.Enc}_r, \mathsf{IBE.Dec}_r)$ be an IBE scheme with plaintext length λ and randomness length ρ. $\mathcal{RO}_n$ has range size $\rho = n$ bits. $\mathsf{IDREwH1} = (\mathsf{IBE.Setup}_r, \mathsf{IBE.Gen}_r, \mathsf{IBE.Enc}, \mathsf{IBE.Dec}_r)$ uses same algorithms as $\mathcal{IBE}_r$ except $\mathsf{IBE.Enc}$ which is defined as

$$\mathsf{IBE.Enc}^{\mathcal{RO}_n}(params, id, m; r) = \mathsf{IBE.Enc}_r(params, id, m; \mathcal{RO}_n(params, id, m, r)).$$

If $|\rho| = 0$, we can obtain an ID-based version of a deterministic encryption scheme, Encrypt-with-Hash. Our theorems about $\mathsf{IDREwH1}$ also works for deterministic encryption.

ID-CPA Security in the $\mathcal{VO}$ Model. We prove the ID-CPA security of $\mathsf{IDREwH1}$; that is, we show that $\mathsf{IDREwH1}$ is selective (resp. full) ID-CPA secure in the $\mathcal{VO}$ model if $\mathcal{IBE}_r$ is selective (resp. full) ID-CPA secure in the RO model,

Theorem 5. *Let $\mathcal{IBE}_r$ be an IBE scheme. Let $\mathcal{B}$ be a selective (resp. full) CPA adversary for* $\mathsf{IDREwH1}$ *in the $\mathcal{VO}$ model, which makes at most $q_{\mathcal{RO}}, q_{\mathcal{RO}^*}, q_{\mathcal{RO}^T}, q_{\mathcal{TO}}, q_E, q_D$ queries to $\mathcal{RO}_n, \mathcal{RO}^*_v, \mathcal{TRO}_w = (\mathcal{RO}^T_w, \mathcal{TO}), \mathsf{IC}_{a,b} = (E, D)$. Then, there exists a selective (resp. full) CPA adversary $\mathcal{C}$ for $\mathcal{IBE}_r$ such that*

$$\mathsf{Adv}^{\text{id-cpa}}_{\mathsf{IDREwH1},\mathcal{VO}}(\mathcal{B}) \leq \mathsf{Adv}^{\text{id-cpa}}_{\mathcal{IBE}_r,\mathcal{RO}}(\mathcal{C}) + \frac{q_{\mathcal{RO}}}{2^{\rho}}.$$

$\mathcal{C}$ runs in time that of $\mathcal{B}$ plus $\mathcal{O}(q_{\mathcal{RO}} + q_{\mathcal{RO}^*} + q_{\mathcal{RO}^T} + q_{\mathcal{TO}} + q_E + q_D)$. ♦

The proof outline is as follows: First, we start with game $\mathbf{G}_0$ which is exactly the same game as the ID-CPA game in the $\mathcal{VO}$ model. Next, we transform $\mathbf{G}_0$ to game $\mathbf{G}_1$ so that challenge ciphertext $\mathbf{c}$ is generated from fresh randomness instead of the output of $\mathcal{RO}_n$. In game $\mathbf{G}_1$, $\mathbf{c}$ is generated by the exactly same manner as the ID-CPA game for $\mathcal{IBE}_r$. Also, oracle queries to $\mathcal{VO}$ except $\mathcal{RO}_n$ is perfectly simulated because $\mathsf{IBE.Enc}$ algorithm never use $\mathcal{RO}^*_v, \mathcal{RO}^T_w, \mathcal{TO}, E, D$. Thus, $\mathcal{B}$ can be constructed with $\mathcal{C}$.

The proof of Theorem 5 is shown in Appendix E.

ID-CDA Security in the $\mathcal{VO}$ Model. We prove the ID-CDA security of $\mathsf{IDREwH1}$; that is, we show that $\mathsf{IDREwH1}$ is selective (resp. full) ID-CDA secure in the $\mathcal{VO}$ model if $\mathcal{IBE}_r$ is selective (resp. full) ID-CPA secure in the RO model.

Theorem 6. *Let $\mathcal{IBE}_r$ be an IBE scheme. Let $(\mathcal{A}_1, \mathcal{A}_2)$ be a selective (resp. full) CDA adversary for* $\mathsf{IDREwH1}$ *in the $\mathcal{VO}$ model, which makes at most $q_{\mathcal{RO}}, q_{\mathcal{RO}^*}, q_{\mathcal{RO}^T}, q_{\mathcal{TO}}, q_E, q_D$ queries to $\mathcal{RO}_n, \mathcal{RO}^*_v, \mathcal{TRO}_w = (\mathcal{RO}^T_w, \mathcal{TO}), \mathsf{IC}_{a,b} = (E, D)$. Then, there exists a selective (resp. full) CPA adversary $\mathcal{C}$ for $\mathcal{IBE}_r$ such that*

$$\mathsf{Adv}^{\text{id-cda}}_{\mathsf{IDREwH1},\mathcal{VO}}(\mathcal{A}_1, \mathcal{A}_2) \leq 2\mathsf{Adv}^{\text{id-cpa}}_{\mathcal{IBE}_r,\mathcal{RO}}(\mathcal{C}) + q_{\mathcal{RO}} \cdot \mathsf{maxparams}_{\mathcal{IBE}_r} + \frac{q_{\mathcal{RO}} + 4q^2_{\mathcal{RO}^*} + 4q^2_{\mathcal{RO}^T} + 4q^2_{\mathcal{TO}} + 4q^2_E + 4q^2_D}{2^{\mu}}.$$

$\mathcal{C}$ runs in time that of $(\mathcal{A}_1, \mathcal{A}_2)$ plus $\mathcal{O}(q_{\mathcal{RO}} + q_{\mathcal{RO}^*} + q_{\mathcal{RO}^T} + q_{\mathcal{TO}} + q_E + q_D)$. $\mathsf{maxparams}_{\mathcal{IBE}_r}$ *is the maximum public-parameter collision probability defined as* $\mathsf{maxparams}_{\mathcal{IBE}_r} = \max_{\gamma \in \{0,1\}^*} \Pr[params = \gamma : (params, msk) \xleftarrow{\$} \mathsf{IBE.Setup}]$. ♦

The proof outline is as follows: First, we start with game $\mathbf{G}_0$ which is exactly the same game as the ID-CDA game in the $\mathcal{VO}$ model. Secondly, we transform $\mathbf{G}_0$ to game $\mathbf{G}_1$ so that challenge ciphertext $\mathbf{c}$ is generated from fresh randomness instead of the output of $\mathcal{RO}_n$. Thirdly, we transform $\mathbf{G}_1$ to game $\mathbf{G}_2$ so that challenge ciphertext $\mathbf{c}$ is generated from all zero messages instead of given messages from $\mathcal{A}_1$. In game $\mathbf{G}_2$, ciphertext $\mathbf{c}$ does not contain any information about outputs of $\mathcal{A}_1$. Finally, we transform $\mathbf{G}_2$ to game $\mathbf{G}_3$ 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 $\mathcal{A}_2$ according to the output of $\mathcal{A}_1$. In game $\mathbf{G}_3$, queries to 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 $\mathcal{VO}$. Thus, the advantage of $\mathcal{A}_2$ in $\mathbf{G}_3$ is nothing.

The proof of Theorem 6 is shown in Appendix F.

References

1. Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and Raphael C.-W. Phan. SHA-3 proposal BLAKE. Submission to NIST (Round 3). 2010.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the Indifferentiability of the Sponge Construction. In *EUROCRYPT*, pages 181–197, 2008.
9. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak SHA-3 submission. Submission to NIST (Round 3). 2011.
10. 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.
11. Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO 2001*, pages 213–229, 2001.
12. Ran Canetti, Shai Halevi, and Jonathan Katz. A Forward-Secure Public-Key Encryption Scheme. In *EUROCRYPT 2003*, pages 255–271, 2003.
13. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *EUROCRYPT 2004*, pages 207–222, 2004.
14. Donghoon Chang, Sangjin Lee, Mridul Nandi, and Moti Yung. Indifferentiable Security Analysis of Popular Hash Functions with Prefix-Free Padding. In *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 283–298. Springer, 2006.
15. Donghoon Chang and Mridul Nandi. Improved Indifferentiability Security Analysis of chopMD Hash Functionl. In *FSE*, pages pages 429–443, 2008.
16. 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.
17. Ivan Damgård. A Design Principle for Hash Functions. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.
18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. 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.
24. Ralph C. Merkle. One Way Hash Functions and DES. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.
25. Ilya Mironov, Omkant Pandey, Omer Reingold, and Gil Segev. Incremental Deterministic Public-Key Encryption, (Full Version in ePrint 2012/047). In *EUROCRYPT*, 2012.

```
$\underline{S_F(X)}$ where $x = X[1,n]$, $y = Y[n+1,d]$
1 $M \leftarrow \mathcal{TO}(y)$;
2 if $y = IV_2$ then
3     $z \leftarrow \mathcal{RO}_n(x \oplus IV_1)$; $w \leftarrow \mathcal{RO}_c^T(x \oplus IV_1)$;
4 else if $M \neq \perp$ then
5     $m \leftarrow x \oplus \mathcal{RO}_n(M)$;
6     $z \leftarrow \mathcal{RO}_n(M||m)$; $w \leftarrow \mathcal{RO}_c^T(M||m)$;
7 else $z||w \leftarrow \mathcal{P}(x||y)$;
8 **return** $z||w$;
```

```
$\underline{S_I(Y)}$ where $z = Y[1,n]$, $w = Y[n+1,d]$
1 $M \leftarrow \mathcal{TO}(w)$;
2 if $M \neq \perp$ and $|M| = n$ then
3     $x \leftarrow IV_1 \oplus M$; $y \leftarrow IV_2$;
4 if $M \neq \perp$ and $|M| > n$ then
5     let $M = M^*||m$ ($|m| = n$);
6     $x \leftarrow m \oplus \mathcal{RO}_n(M)$; $y \leftarrow \mathcal{RO}_c^T(M^*)$;
7 else $x||y \leftarrow \mathcal{P}^{-1}(z||w)$;
8 **return** $x||y$;
```

Fig. 9. Simulator S_F (left) and S_I (right)

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

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

Fig. 10. $\mathcal{Q}$ 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.

26. Yusuke Naito, Kazuki Yoneyama, Lei Wang, and Kazuo Ohta. How to Confirm Cryptosystems Security: the Original Merkle-Damgård is Still Alive! In *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*. Springer, 2009.
27. National Institute of Standards and Technology. Cryptographic Hash Algorithm Competition. `http://csrc.nist.gov/groups/ST/hash/sha-3/winner_sha-3.html`.
28. National Institute of Standards and Technoloty. FIPS PUB 180-3 Secure Hash Standard. In *FIPS PUB*, 2008.
29. 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.
30. Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. In *Cryptology ePrint Archive: 2004/332*, 2004. `http://eprint.iacr.org/2004/332`.
31. Hongjun Wu. The Hash Function JH. Submission to NIST (Round 3). 2011.

A Proof of Theorem 1

In this proof, we use the Sponge graph described in Subsection 3.2.

Simulator S**.** We define a simulator S in Fig. 9 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 S is consistent with $\mathcal{RO}_n$, that is, for any Sponge path $IV \xrightarrow{M} Y$, $Y[1,n] = \mathcal{RO}_n(M)$. In this proof, we omit the padding function pad_S. Thus the left queries must 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. The simulator in Fig. 9 is consistent with $\mathcal{RO}_n$. For example, for the ordered queries $S_F(x_1||IV_2), S_F(x_2||w_1)$ where $z_1||w_1 = S_F(x_1||IV_2), z_2||w_2 = S_F(x_2||w_1)$, the structure of S ensures that $w_1 = \mathcal{RO}_c^T(M_1)$ (the step 3 of S_F) and $w_2 = \mathcal{RO}_c^T(M_1||M_2)$ (the step 6 of S_F) where $M_1 = IV_1 \oplus x_1$ and $M_2 = z_1 \oplus x_2$. Then, for a query $S_F(x_3||w_2)$, the response $z_3||w_3$ is defined such that $z_3 = \mathcal{RO}_n(M_1||M_2||M_3)$ (the step 6 of S_F) where $M_3 = z_2 \oplus x_3$. Notice that $M_1||M_2$ can be obtained by the queries $\mathcal{TO}(w_2)$ (the step 1 of S_F) and z_2 can be obtained by the query $\mathcal{RO}_n(M_1||M_2)$ (the step 5 of S_F). Thus the simulator S succeeds in the simulation of the random permutation.

Detail. The proof is given as follows. 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 ideal 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 has oracle access to $(\mathcal{P}_1, \mathcal{P}_1^{-1})$ instead of $(\mathcal{P}, \mathcal{P}^{-1})$. $(\mathcal{P}_1, \mathcal{P}_1^{-1})$ are implemented as in Figure. 10.
- 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 real 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

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

In the following, we justify the above bound by evaluating each difference. Since S does not update the internal state, *S.Rst* does not give an advantage to $\mathcal{A}$. Thus $\Pr[G_1] = \Pr[G_2]$. Since *nop* does noting, it does not give an advantage to $\mathcal{A}$. Thus $\Pr[G_7] = \Pr[G_8]$. We thus consider Games 2, 3, 4, 5, 6, and 7.

Game 2 ⇒ 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 the 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,

$$|\Pr[G_2] - \Pr[G_3]| \leq \sum_{i=1}^{q} \frac{i-1}{2^d} = \frac{q(q-1)}{2^{d+1}}.$$

Game 3 ⇒ 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 thus bounded by the probability that in Game 3 the different value is responded. We call the event "**Diff**". Since selecting a procedure defining an output is controlled by $\mathcal{TO}$, if **Diff** occurs, the following event occurs.

- For a repeated query $\mathcal{TO}(y)$ where w was responded before, a different value w^* is responded, where if $w =\perp$ then $w^* \neq\perp$ (denoted $\mathbf{Diff}_1$), and if $w \neq\perp$ then $w^* =\perp$ (denoted $\mathbf{Diff}_2$), from the definition of $\mathcal{TRO}$.

We thus have that

$$|\Pr[G_2] - \Pr[G_3]| \leq \Pr[\mathbf{Diff}_1] + \Pr[\mathbf{Diff}_2] \leq \frac{q(q-1)}{2^c}.$$

We justify the bound as follows.

Consider $\mathbf{Diff}_1$. When the query $\mathcal{TO}(y)$ was made, no query $\mathcal{RO}_c^T(w^*)$ such that $y = \mathcal{RO}_c^T(w^*)$ had been made. And when the repeated query $\mathcal{TO}(y)$ is made, the query $\mathcal{RO}_c^T(w^*)$ such that $y = \mathcal{RO}_c^T(w^*)$ was made. Thus $\Pr[\mathbf{Diff}_1]$ is bounded by the probability that some output of $\mathcal{RO}_c^T(w^*)$ (c-bit random value) hits a fixed value y. Since the numbers of queries to $\mathcal{RO}_c^T$ and $\mathcal{TO}$ are at most q times,

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

Consider $\mathbf{Diff}_2$. From the definition of $\mathcal{TO}$, if $\mathbf{Diff}_2$ occurs, a collision of $\mathcal{RO}_c^T$ occurs, We thus have that

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

Game 4 ⇒ 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 directly but can find those by making the corresponding right queries. So we must show that the additional right query-responses that $\mathcal{A}$ obtains don't affect the $\mathcal{A}$'s behavior. We show Lemma 1 where for any Sponge path $IV \xrightarrow{M} z$, $z[1,n] = \mathcal{RO}_n(M)$ unless Bad_j occurs. Let T_i be a table which stores all values $X_t[n+1,d]$ and $Y_t[n+1,d]$ for $t = 1,\ldots,i-1$ where (X_t, Y_t) is a query-response pair defined by the t-th R_F or R_I query.

- Bad_j is that in Game j, for some i-th query $S_F(X_i)$ where Y_i is the response, $Y_i[n+1,d]$ collides with some value in $T_i \cup \{X_i[n+1,d]\} \cup \{IV_2\}$, or
  for some i-th query $S_I(Y_i)$ where X_i is the response, $X_i[n+1,d]$ collides with some value in $T_i \cup \{Y_i[n+1,d]\} \cup \{IV_2\}$.

Lemma 1 ensures that in Game 5, unless Bad_j occurs, responses which are leafs of Sponge paths[7] are defined by the same queries to $\mathcal{RO}_c^T$ and $\mathcal{RO}_n$ as in Game 4. Namely, unless the bad event occurs, the responses of the additional right queries don't affect the $\mathcal{A}$'s view. Thus, the difference $|\Pr[G_4] - \Pr[G_5]|$ is bounded by the probability of occurring the bad event. We thus have that

$$\begin{aligned}
|\Pr[G_4] - \Pr[G_5]| \leq & |\Pr[G_4|Bad_4]\Pr[Bad_4] + \Pr[G_4|\neg Bad_4]\Pr[\neg Bad_4] \\
& - (\Pr[G_5|Bad_5]\Pr[Bad_5] + \Pr[G_5|\neg Bad_5]\Pr[\neg Bad_5])| \\
\leq & |\Pr[G_4|\neg Bad_4](\Pr[Bad_5] - \Pr[Bad_4]) \\
& + (\Pr[G_4|Bad_4]\Pr[Bad_4] - \Pr[G_5|Bad_5]\Pr[Bad_5])| \\
\leq & \max\{\Pr[Bad_4], \Pr[Bad_5]\} \leq \frac{\sigma(\sigma+1)}{2^c}
\end{aligned}$$

where $\Pr[G_4|\neg Bad_4] = \Pr[G_5|\neg Bad_5]$ from Lemma 1. We justify the bound later.

Lemma 1. *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 1. Assume that Bad_j does not occur. 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 $M_1 = IV_1 \oplus X_1[1,n], \cdots, M_t = Y_{t-1}[1,n] \oplus X_t[1,n]$. We show that $z[1,n] = \mathcal{RO}_n(M)$.

Consider the case that $t = 1$. Then no pair (X, Y) which is defined by an R_I query $(X = R_I(Y))$ connects IV. Thus any path $IV \xrightarrow{M} z$ such that $|M| = n$ is defined by a R_F query. Then $z[1,n] = \mathcal{RO}_n(M)$.

Consider the case that $t \geq 2$.

No pair (X, Y) which is defined by an R_I 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 R_F queries. So, all pairs in any sponge path are defined by R_F queries.

The assumption ensures that no pair which is defined by a R_F query is connected with another pair: a path $Y \xrightarrow{M^*} Y^*$ was defined then a path $X \xrightarrow{M} Y$ is defined. More formally, the pair (X, Y) which is defined by a R_F 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. Thus $(X_1, Y_1), \ldots, (X_t, Y_t)$ are defined by the ordered R_F queries $R_F(X_1), \ldots, R_F(X_t)$.

The assumption ensures that all pairs are different, since no pair is connected with oneself. Thus for any $i, j \in \{1,\ldots,t\}$ $(X_i, Y_i) \neq (X_j, Y_j)$.

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

```
$S(x,m)$ 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. 11. Simulator S

The assumption ensures that for any i-th query-response (X,Y) defined by a R_F 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, pairs $(X_1,Y_1),\ldots,(X_t,Y_t)$ are defined by this order such that for any $i,j \in \{1,\ldots,t\}$ $(X_i,Y_i) \neq (X_j,Y_j)$. And no collision for $\mathcal{RO}_c^T$ occurs and all outputs of $\mathcal{RO}_c^T$ does not collide with IV_2.

From above discussions, for a R_F 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 $Y_t[1,n] = \mathcal{RO}_n(M)$.

□

Evaluation of $\Pr[Bad_4], \Pr[Bad_5]$**.** 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 $\{0,1\}^c$,

$$\Pr[Bad_4] \leq \sum_{i=1}^{q} \frac{(2(i-1)+2)}{2^c} = \frac{q(q+1)}{2^c},\ \Pr[Bad_5] \leq \sum_{i=1}^{\sigma} \frac{(2(i-1)+2)}{2^c} = \frac{\sigma(\sigma+1)}{2^c}$$

Game 5 ⇒ 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 1, 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. 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 have that

$$|\Pr[G_5] - \Pr[G_6]| \leq \Pr[Bad_6] \leq \frac{\sigma(\sigma+1)}{2^c}.$$

Game 6 ⇒ 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. The difference is thus bounded by the collision probability of R_F and R_I in Game 6. We thus have that

$$|\Pr[G_6] - \Pr[G_7]| \leq \sum_{i=1}^{\sigma} \frac{i-1}{2^d} = \frac{\sigma(\sigma-1)}{2^{d+1}}.$$

□

B Proof of Theorem 2

In this proof, we use the MD graph described in Subsection 3.2.

Simulator S**.** We define a simulator S in Fig. 11 which does not update the internal state to remove the attack using $S.Rst$. In this proof, 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 $IV \xrightarrow{M} z$, $z[s+1, n+s] = \mathcal{RO}_n(M)$. The simulator S is consistent with $\mathcal{RO}_n$. For example, 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 $w_1 = \mathcal{RO}_s^T(M_1)$ (the step 3), and $w_2 = \mathcal{RO}_s^T(M_1||M_2)$ (the step 7). 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 such that $z_3 = \mathcal{RO}_n(M_1||M_2||M_3)$ (the step 6). Notice that $M_1||M_2$ can be obtained by the queries $\mathcal{TO}(w_2)$ (the step 1). So we can construct a simulator such that 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 we can ensure the consistency.

Detail. 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 ideal 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 real 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

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff},\mathcal{VO}}_{\mathrm{chopMD}^h,S}(\mathcal{A}) \le \sum_{i=1}^{6} |\Pr[G_i] - \Pr[G_{i+1}]| \le \frac{3q_R(q_R-1)}{2^{s+1}} + \frac{q_R(q_R+3)}{2^{n+s+1}} + \frac{\sigma(\sigma+1)}{2^{s+n}}.$$

In the following, we justify the above bound by evaluating each difference. Since S does not update the internal state, $S.Rst$ does not give an advantage to $\mathcal{A}$. Thus $\Pr[G_1] = \Pr[G_2]$. Since nop does noting, it does not give an advantage to $\mathcal{A}$. Thus $\Pr[G_6] = \Pr[G_7]$. 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 a repeated query $R(x, m)$ the same value 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 thus bounded by the probability that in Game 2 the different value is responded. We call the event "**Diff**". Since selecting a procedure defining an output is controlled by $\mathcal{TO}$, if **Diff** occurs, the following event occurs.

- For a repeated query $\mathcal{TO}(y)$ where w was responded, a different value w^* is responded, where if $w =\perp$ then $w^* \neq\perp$ (denoted $\mathbf{Diff}_1$) , and if $w \neq\perp$ then $w^* =\perp$ (denoted $\mathbf{Diff}_2$), from the definition of $\mathcal{TRO}$.

We thus have that

$$|\Pr[G_2] - \Pr[G_3]| \le \Pr[\mathbf{Diff}_1] + \Pr[\mathbf{Diff}_2] \le \frac{q_R(q_R-1)}{2^s}.$$

We justify the bound as follows.

Consider $\mathbf{Diff}_1$. When the query $\mathcal{TO}(y)$ was made, no query $\mathcal{RO}_s^T(w^*)$ such that $y = \mathcal{RO}_s^T(w^*)$ had been made. And when the repeated query $\mathcal{TO}(y)$ is made, the query $\mathcal{RO}_s^T(w^*)$ such that $y = \mathcal{RO}_s^T(w^*)$ was made. Thus $\Pr[\mathbf{Diff}_1]$ is bounded by the probability that some output of $\mathcal{RO}_s^T(w^*)$ (s-bit random value) hits a fixed value y. Since the numbers of queries to $\mathcal{RO}_s^T$ and $\mathcal{TO}$ are at most q_R times,

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

Consider $\mathbf{Diff}_2$. From the definition of $\mathcal{TO}$, if $\mathbf{Diff}_2$ occurs, a collision of $\mathcal{RO}_s^T$ occurs, We thus have that

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

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 chopMD$^{S_1}(M)$. Note that $\mathcal{A}$ cannot find the additional right query-responses directly but can find those by making the corresponding right queries. So we must show that the additional right query-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 occurs. Let T_i be a list which records (x_t, y_t) for $t = 1, \ldots, i-1$ where (x_t, m_t, y_t) is a t-th R query-response ($y_t = R(x_t, m_t)$).

– Bad_j is that in Game j for some i-th query $R(x_i, m_i)$ the response y_i collides with some value in $T_i \cup \{x_i\} \cup \{IV\}$.

This ensures that unless the bad event occurs, in both games responses which are leafs of MD paths[8] are defined by the same query to $\mathcal{RO}_s^T$ and $\mathcal{RO}_n$. Namely, in Game 4, 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. We thus have that

$$|\Pr[G_3] - \Pr[G_4]| \le \max\{\Pr[Bad_3], \Pr[Bad_4]\} \le \frac{q_R(q_R-1)}{2^{s+1}} + \frac{q_R(q_R+3)}{2^{n+s+1}}$$

where $\Pr[G3|\neg Bad_3] = \Pr[G4|\neg Bad_4]$ from Lemma 2. 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 $j = 1$, $y[s+1, n+s] = \mathcal{RO}_n(M)$ (see the steps 2-4).

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

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

The assumption ensures that for any $u, v \in \{1, \ldots, j\}$ $(x_u, m_u, y_u) \ne (x_v, m_v, y_v)$.

The assumption ensures that no collision for $\mathcal{RO}_n^T$ occurs.

From the above discussions, for the query $S(x_j, m_j)$, $\mathcal{TO}(x_j)$ returns $m_1||\ldots||m_{j-1}$ (the step 1) and then the response y_i is defined such that $y_i[s+1, n+s] = \mathcal{RO}_n(M)$ (the step 6).

□

Evaluation of $\Pr[Bad_3], \Pr[Bad_4]$**.** In Game 3 and Game 4 S is called at most q_R and σ times, respectively. The output of R is uniformly chosen at random from $\{0,1\}^{n+s}$. Note that in Game 3, since $L = \mathcal{RO}_n$, by a L query, $\mathcal{A}$ can obtain the right n-bit value of the response of some R query in advance. On the other hand, the right s-bit value cannot be obtained in advance. We thus have that

$$\Pr[Bad_3] \le \sum_{i=1}^{q_R} \left(\frac{i-1}{2^s} + \frac{i-1+2}{2^{n+s}} \right) = \frac{q_R(q_R-1)}{2^{s+1}} + \frac{q_R(q_R+3)}{2^{n+s+1}},$$

$$\Pr[Bad_4] \le \sum_{i=1}^{\sigma} \frac{2(i-1)+2}{2^{n+s}} = \frac{\sigma(\sigma+1)}{2^{n+s}}.$$

Game 4 ⇒ Game 5. The difference between Game 4 and Game 5 is the left oracle L where in Game 4

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

```
$S(x,m)$
1 $M^* \leftarrow \mathcal{TO}(x)$;
2 if $x = IV$ then
3     if $\exists M$ s.t. $\mathsf{pfpad}(M) = m$ then $y \leftarrow \mathcal{RO}_n(M)$;
4     else $y \leftarrow \mathcal{RO}_n^T(m)$;
5 else if $M^* \neq \perp$ then
6     if $\exists M$ s.t. $\mathsf{pfpad}(M) = M^*||m$ then $y \leftarrow \mathcal{RO}_n(M)$;
7     else $y \leftarrow \mathcal{RO}_n^T(M^*||m)$;
8 else $y \leftarrow \mathcal{RO}_n^*(x,m)$;
9 **return** $y$;
```

Fig. 12. Simulator S

$L(M)$ returns $\mathcal{RO}_n(M)$, while in Game 5 $L(M)$ returns 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. Since R is called at most σ times, we have that

$$|\Pr[G_4] - \Pr[G_5]| \leq \Pr[Bad_5] \leq \frac{\sigma(\sigma+1)}{2^{n+s}}.$$

Game 5 ⇒ 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 $\Pr[G_5] = \Pr[G_6]$. □

C Proof of Theorem 3

In this proof, we use the MD graph described in Subsection 3.2.

Simulator S. We define a simulator S in Fig. 12 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 $M^* = \mathsf{pfpad}(M)$. S in Fig. 12 is consistent with $\mathcal{RO}_n$. For example, 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 $\mathcal{RO}_n^T(m_1)$ (the step 4) and $\mathcal{RO}_n^T(m_1||m_2)$ (the step 7), respectively. Then for the query $S(y_2, m_3)$, the response is defined by the output of $\mathcal{RO}_n(M)$ (the step 6) 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 $\mathcal{TO}(y_2)$ (the step 1). So the PFMD 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.

Detail. 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 ideal 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 PFMD$^{S_1}(M)$ then returns $\mathcal{RO}_n(M)$.
- Game 5 is $(L,R) = (\text{PFMD}^{S_1}, S_1)$.
- Game 6 is $(L,R) = (\text{PFMD}^h, h)$.
- Game 7 is the real world, that is, $(L,R) = (\text{PFMD}^h, h)$ 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

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

In the following, we justify the above bound by evaluating each difference. Since S does not update the internal state, $S.Rst$ does not give an advantage to $\mathcal{A}$. Thus $\Pr[G_1] = \Pr[G_2]$. Since *nop* does noting, it does not give an advantage to $\mathcal{A}$. Thus $\Pr[G_6] = \Pr[G_7]$. 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 thus bounded by the probability that in Game 2 the different value is responded. We call the event "**Diff**". Since selecting a procedure defining an output is controlled by $\mathcal{TO}$, if **Diff** occurs, the following event occurs.

- For a repeated query $\mathcal{TO}(y)$ where w was responded, a different value w^* is responded, where if $w = \perp$ then $w^* \neq \perp$ (denoted $\mathbf{Diff}_1$) , and if $w \neq \perp$ then $w^* = \perp$ (denoted $\mathbf{Diff}_2$), from the definition of $\mathcal{TRO}$.

We thus have that

$$|\Pr[G_2] - \Pr[G_3]| \leq \Pr[\mathbf{Diff}_1] + \Pr[\mathbf{Diff}_2] \leq \frac{q_R(q_R-1)}{2^n}.$$

We justify the bound as follows.

Consider $\mathbf{Diff}_1$. When the query $\mathcal{TO}(y)$ was made, no query $\mathcal{RO}_n^T(w^*)$ such that $y = \mathcal{RO}_n^T(w^*)$ had been made. And when the repeated query $\mathcal{TO}(y)$ is made, the query $\mathcal{RO}_n^T(w^*)$ such that $y = \mathcal{RO}_n^T(w^*)$ was made. Thus $\Pr[\mathbf{Diff}_1]$ is bounded by the probability that some output of $\mathcal{RO}_n^T(w^*)$ (n-bit random value) hits a fixed value y. Since the numbers of queries to $\mathcal{RO}_n^T$ and $\mathcal{TO}$ are at most q_R times,

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

Consider $\mathbf{Diff}_2$. From the definition of $\mathcal{TO}$, if $\mathbf{Diff}_2$ occurs, a collision of $\mathcal{RO}_n^T$ occurs, We thus have that

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

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 directly but can find those by making corresponding right queries. So we must show that the additional right query-responses that $\mathcal{A}$ obtains don't affect the $\mathcal{A}$'s behavior. We show Lemma 3 where for any PFMD path $IV \xrightarrow{M^*} y$ where $M^* = \mathsf{pfpad}(M)$, $y = \mathcal{RO}_n(M)$ unless Bad_j. Let T_i be a list which records (x_t, y_t) for $t = 1, \ldots, i-1$ where (x_t, m_t, y_t) is a t-th R query-response ($y_t = R(x_t, m_t)$).

- Bad_j is that in Game j for some i-th query $R(x_i, m_i)$ the response y_i collides with some value in $T_i \cup \{x_i\} \cup \{IV\}$.

This ensures that unless the bad event occurs, in both games responses which are leafs of MD paths[9] are defined by the same query to $\mathcal{RO}_n^T$ and $\mathcal{RO}_n$. Namely, in Game 4, 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. We thus have that

$$|\Pr[G_3] - \Pr[G_4]| \leq \max\{\Pr[Bad_3], \Pr[Bad_4]\} \leq \frac{\sigma(\sigma+1)}{2^n}$$

where $\Pr[G3|\neg Bad_3] = \Pr[G4|\neg Bad_4]$ from Lemma 3. We justify the bound later.

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

Lemma 3. *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 3. 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 $M^* = m_1 || \ldots || m_j$.

When $j = 1$, $y = \mathcal{RO}_n(M)$ (due to the step 1).

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

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 $(x_{i+1}, m_{i+1}, y_{i+1})$. So $(x_1, m_1, y_1), \ldots, (x_j, m_j, y_j)$ are defined by this order.

The assumption ensures that no triple connects with oneself, namely any triple (x_i, m_i, y_i) is such that $x_i \neq y_i$. Thus no triple appears twice in the PFMD path $IV \xrightarrow{M^*} y$.

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.

From the above discussions, for the query $S(x_j, m_j)$, $\mathcal{TO}(x_j)$ responses $m_1 || \ldots || m_{j-1}$ (the step 1) and then the response y_i is defined such that $y_i = \mathcal{RO}_n(M)$ (the step 3). □

Evaluation of $\Pr[Bad_3], \Pr[Bad_4]$**.** 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,

$$\Pr[Bad_3] \leq \sum_{i=1}^{q_R} \frac{2(i-1)+2}{2^n} = \frac{q_R(q_R+1)}{2^n},\ \Pr[Bad_4] \leq \sum_{i=1}^{\sigma} \frac{2(i-1)+2)}{2^n} = \frac{\sigma(\sigma+1)}{2^n}.$$

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 3, for any PFMD path $IV \xrightarrow{M^*} z$, $z = \mathcal{RO}_n(M)$ unless the bad event Bad_5, where $M^* = \mathsf{pfpad}(M)$. Since 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

$$|\Pr[G_4] - \Pr[G_5]| \leq \Pr[Bad_5] \leq \frac{\sigma(\sigma+1)}{2^n}.$$

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 $\Pr[G_5] = \Pr[G_6]$.

□

D Proof of Theorem 4

Proof. We denote $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_i)$ by the advantage of the adversary $\mathcal{A}$ when participating in experiment $\mathbf{G}_i$. We start with game $\mathbf{G}_0$ which is exactly the same game as the CDA game in the $\mathcal{VO}$ model. It means $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_0) = \mathsf{Adv}^{\mathrm{cda}}_{\mathcal{AE},\mathcal{VO}}(\mathcal{A}_1, \mathcal{A}_2)$.

Game $\mathbf{G}_1$: $\mathcal{RO}_n$ returns a random value if one of following events occur:

- Bad_1 : $\mathcal{A}_1$ poses a message M to $\mathcal{RO}_n$ where M is posed to $\mathcal{RO}_n$ by Enc to generate the challenge ciphertext.
- Bad_2 : $\mathcal{A}_2$ poses a message M to $\mathcal{RO}_n$ where M is posed to $\mathcal{RO}_n$ by Enc to generate the challenge ciphertext.

All other procedures are computed as the same way in $\mathbf{G}_0$.

Lemma 4. $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_0)| \leq \frac{q_{\mathcal{RO}}}{2^\mu} + q_{\mathcal{RO}} \cdot \mathsf{maxpk}_{\mathcal{AE}}$.

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

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

```
$\mathsf{SimB}_{\mathcal{RO}_n}(M)$
If $\mathsf{F}[M] = \perp$, $\mathsf{F}[M] \xleftarrow{\$} \{0,1\}^n$
If $\mathsf{F}[M] \neq \perp$, and $M$ is posed by $\mathsf{Enc}$,
    $\mathsf{F}[M] \xleftarrow{\$} \{0,1\}^n$
return $\mathsf{F}[M]$
```

```
$\mathsf{SimB}_{\mathcal{RO}^*_v}(M)$
If $\mathsf{F}^*[M] = \perp$, $\mathsf{F}^*[M] \xleftarrow{\$} \{0,1\}^v$
return $\mathsf{F}^*[M]$;
```

```
$\mathsf{SimB}_{\mathcal{RO}^T_w}(M)$
If $\mathsf{F}^T[M] = \perp$ 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. 13. game $\mathbf{G}_2$ and simulation SimB by adversary $\mathcal{B}$

Proof. The difference between $\mathbf{G}_0$ and $\mathbf{G}_1$ only occurs in Bad_1 and Bad_2. From Difference Lemma [30], we have that $|\mathsf{Adv}(\mathcal{B}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{B}, \mathbf{G}_0)| \leq \Pr[\mathsf{Bad}_1 \vee \mathsf{Bad}_2] \leq \Pr[\mathsf{Bad}_1] + \Pr[\mathsf{Bad}_2]$.

First, we estimate $\Pr[\mathsf{Bad}_1]$. Since pk is not given for $\mathcal{A}_1$ and is included in each query to $\mathcal{RO}_n$ by Enc, the only way to pose $(pk, *, *)$ to $\mathcal{RO}_n$ is choosing pk randomly $q_{\mathcal{RO}}$ times. We have that $\Pr[\mathsf{Bad}_1] \leq q_{\mathcal{RO}} \cdot \mathsf{maxpk}_{\mathcal{AE}}$.

Next, we estimate $\Pr[\mathsf{Bad}_2]$. Since $\mathcal{RO}_n$ is a truly random function and r (which is used to generate challenge ciphertext $\mathbf{c}$) is included in each query to $\mathcal{RO}_n$ by Enc, $\mathcal{A}_2$ cannot obtain more information of r than min-entropy μ from challenge ciphertext even if $\mathcal{A}_2$ could obtain some information about $\mathcal{RO}_n(pk, \mathbf{m}_\beta; \mathbf{r})$ from $\mathbf{c}$. Thus, the only way to pose $(*, *, r)$ to $\mathcal{RO}_n$ is guessing r under min-entropy μ $q_{\mathcal{RO}}$ times. We have that $\Pr[\mathsf{Bad}_2] \leq \frac{q_{\mathcal{RO}}}{2^\mu}$. □

Game $\mathbf{G}_2$: Ciphertext $\mathbf{c} \leftarrow \mathsf{Enc}^{\mathcal{RO}_n}(pk, \mathbf{m}_b; \mathbf{r})$ 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 $\mathbf{G}_1$.

Lemma 5. $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_1)| \leq \mathsf{Adv}^{\text{ind-sim}}_{\mathcal{AE}, \mathcal{S}, \mathcal{RO}_n}(\mathcal{B})$.

Proof. We show that if $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_1)|$ 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. 13 shows game $\mathbf{G}_2$, the construction of $\mathcal{B}$, and the simulation $\mathsf{SimB} = (\mathsf{SimB}_{\mathcal{RO}_n}, \mathsf{SimB}_{\mathcal{RO}^*_v}, \mathsf{SimB}_{\mathcal{RO}^T_w}, \mathsf{SimB}_{\mathcal{TO}}, \mathsf{SimB}_E, \mathsf{SimB}_D)$ of $\mathcal{VO}$ by $\mathcal{B}$ respectively. Note that $\mathcal{B}$ makes no RO queries, and $\mathsf{Enc}^{F.priv}(pk, \mathbf{m}_\beta; \mathbf{r})$ 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 Enc is contained both in $\mathbf{G}_1$ and $\mathbf{G}_2$. Thus, $\mathcal{A}$ cannot distinguish game $\mathbf{G}_1$ and $\mathbf{G}_2$ 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 $\mathbf{G}_1$. If $\beta = 0$ in IND-SIM game, it is clear that all interfaces for $\mathcal{A}$ is exactly same as game $\mathbf{G}_2$.

Therefore, if $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_1)|$ is non-negligible, $\mathcal{B}$ also breaks IND-SIM security of $\mathcal{AE}$. □

Game $\mathbf{G}_3$: When $\mathcal{A}_2$ poses a query related to $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ (which is the output of $\mathcal{A}_1$) to $\mathcal{RO}^*_v, \mathcal{TRO}_w = (\mathcal{RO}^T_w, \mathcal{TO})$ or $\mathsf{IC}_{a,b} = (E, D)$, then outputs are randomly chosen. That is, tables $\mathsf{F}^*, \mathsf{F}^T, \mathsf{E}$ and D are not preserved for $\mathcal{A}_1$ and $\mathcal{A}_2$ according to $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$. All other procedures are computed as the same way in $\mathbf{G}_2$.

Lemma 6. $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_2)| \leq \frac{4q^2_{\mathcal{RO}^*}+4q^2_{\mathcal{RO}^T}+4q^2_{\mathcal{TO}}+4q^2_E+4q^2_D}{2^\mu}$.

Proof. The difference between $\mathbf{G}_2$ and $\mathbf{G}_3$ only occurs when $\mathcal{A}_1$ and $\mathcal{A}_2$ poses a same query related to $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ to $\mathcal{RO}^*_v, \mathcal{TRO}_w = (\mathcal{RO}^T_w, \mathcal{TO})$ or $\mathsf{IC}_{a,b} = (E, D)$. We denote the event that a common query related to $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ is posed to $\mathcal{RO}^*_v$ by $\mathcal{A}_1$ and $\mathcal{A}_2$ as $\mathsf{Bad}_{\mathcal{RO}^*}$. Similarly, we define events $\mathsf{Bad}_{\mathcal{RO}^T}$, $\mathsf{Bad}_{\mathcal{TO}}$, Bad_E, and Bad_D. From Difference Lemma [30], we have that $|\mathsf{Adv}(\mathcal{B}, \mathbf{G}_3) - \mathsf{Adv}(\mathcal{B}, \mathbf{G}_2)| \leq \Pr[\mathsf{Bad}_{\mathcal{RO}^*} \vee \mathsf{Bad}_{\mathcal{RO}^T} \vee \mathsf{Bad}_{\mathcal{TO}} \vee \mathsf{Bad}_E \vee \mathsf{Bad}_D] \leq \Pr[\mathsf{Bad}_{\mathcal{RO}^*}] + \Pr[\mathsf{Bad}_{\mathcal{RO}^T}] + \Pr[\mathsf{Bad}_{\mathcal{TO}}] + \Pr[\mathsf{Bad}_E] + \Pr[\mathsf{Bad}_D]$.

In game $\mathbf{G}_2$ and $\mathbf{G}_3$, 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 $\mathbf{G}_2$ and $\mathbf{G}_3$ are identical. Thus, the only way to pose such a query is guessing under min-entropy μ. According to the birthday paradox, for each oracle the probability of collisions in guessing is at most $(2q_{\mathcal{RO}^*})^2/2^\mu$, $(2q_{\mathcal{RO}^T})^2/2^\mu$, $(2q_{\mathcal{TO}})^2/2^\mu$, $(2q_E)^2/2^\mu$, and $(2q_D)^2/2^\mu$, respectively. Therefore, $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_2)| \leq (4q^2_{\mathcal{RO}^*} + 4q^2_{\mathcal{RO}^T} + 4q^2_{\mathcal{TO}} + 4q^2_E + 4q^2_D)/2^\mu$. □

We estimate $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3)$. Ciphertext $\mathbf{c}$ does not give any information about $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$. Also, outputs of $\mathcal{VO}$ is independent of $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ for $\mathcal{A}_2$. Thus, the only way to win in game $\mathbf{G}_3$ is randomly guessing β. Therefore, $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3) = 0$.

To conclude, we have $\mathsf{Adv}^{\text{cda}}_{\mathcal{AE},\mathcal{VO}}(\mathcal{A}_1, \mathcal{A}_2) \leq \mathsf{Adv}^{\text{ind-sim}}_{\mathcal{AE},\mathcal{S},\mathcal{RO}_n}(\mathcal{B}) + q_{\mathcal{RO}} \cdot \mathsf{maxpk}_{\mathcal{AE}} + (q_{\mathcal{RO}} + 4q^2_{\mathcal{RO}^*} + 4q^2_{\mathcal{RO}^T} + 4q^2_{\mathcal{TO}} + 4q^2_E + 4q^2_D)/2^\mu$.

□

E Proof of Theorem 5

Proof. We denote $\mathsf{Adv}(\mathcal{B}, \mathbf{G}_i)$ by the advantage of adversary $\mathcal{B}$ when participating in experiment $\mathbf{G}_i$. We start with game $\mathbf{G}_0$ which is exactly the same game as the ID-CPA game in the $\mathcal{VO}$ model. It means $\mathsf{Adv}(\mathcal{B}, \mathbf{G}_0) = \mathsf{Adv}^{\text{id-cpa}}_{\mathsf{IDREwH1},\mathcal{VO}}(\mathcal{B})$.

Game $\mathbf{G}_1$: Challenge ciphertext $\mathbf{c} \leftarrow \mathsf{IBE.Enc}_r(params, id^*, \mathbf{m}_\beta; \mathcal{RO}(params, id^*, \mathbf{m}_\beta; \mathbf{r}))$ is replaced with $\mathbf{c} \leftarrow \mathsf{IBE.Enc}_r(params, id^*, \mathbf{m}_\beta; \mathbf{r}')$ for randomly chosen $\mathbf{r}'$. All other procedures are computed as the same way in $\mathbf{G}_0$.

Lemma 7. $|\mathsf{Adv}(\mathcal{B}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{B}, \mathbf{G}_0)| \leq \frac{q_{\mathcal{RO}}}{2^\rho}$.

Proof. The difference between $\mathbf{G}_0$ and $\mathbf{G}_1$ only occurs when adversary $\mathcal{B}$ poses $(params, id^*, m_\beta, r)$ to $\mathcal{RO}_n$ where where $m_\beta \in \mathbf{m}_\beta$, $r \in \mathbf{r}$, and $\mathbf{r}$ is the randomness vector used to generate challenge ciphertext $\mathbf{c}$. We denote this event as Bad. From Difference Lemma [30], we have that $|\mathsf{Adv}(\mathcal{B}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{B}, \mathbf{G}_0)| \leq \Pr[\mathsf{Bad}]$.

We estimate $\Pr[\mathsf{Bad}]$. Since $\mathcal{RO}_n$ is a truly random function, $\mathcal{B}$ cannot know $\mathbf{r}$ (which is used to generate challenge ciphertext $\mathbf{c}$) from challenge ciphertext even if $\mathcal{B}$ could obtain some information about $\mathcal{RO}_n(params, id^*, \mathbf{m}_\beta; \mathbf{r})$ from $\mathbf{c}$. Thus, the only way to pose $(params, id^*, m_\beta, r)$ to $\mathcal{RO}_n$ is choosing r randomly $q_{\mathcal{RO}}$ times. We have that $\Pr[\mathsf{Bad}] \leq \frac{q_{\mathcal{RO}}}{2^\rho}$. □

We estimate $\mathsf{Adv}(\mathcal{B}, \mathbf{G}_1)$. We assume that there exists $\mathcal{B}$ with $\mathsf{Adv}(\mathcal{B}, \mathbf{G}_1)$. Then, we construct adversary $\mathcal{C}$ against $\mathcal{IBE}_r$ with the same advantage as $\mathsf{Adv}(\mathcal{B}, \mathbf{G}_1)$. The simulation SimC by $\mathcal{C}$ is given in Fig. 14.

Since the generation of the challenge ciphertext is exactly same between $\mathbf{G}_0$ and the ID-CPA game for $\mathcal{IBE}_r$, $\mathcal{C}$ just forwards the challenge ciphertext to $\mathcal{B}$. The simulation of $\mathcal{VO}$ is perfect because the challenger $\mathcal{CH}$ never uses all components of $\mathcal{VO}$ with the private channel. Therefore, $\mathsf{Adv}(\mathcal{B}, \mathbf{G}_1) = \mathsf{Adv}^{\text{id-cpa}}_{\mathcal{IBE}_r,\mathcal{RO}}(\mathcal{C})$.

To conclude, we have $\mathsf{Adv}^{\text{id-cpa}}_{\mathsf{IDREwH1},\mathcal{VO}}(\mathcal{B}) \leq \mathsf{Adv}^{\text{id-cpa}}_{\mathcal{IBE}_r,\mathcal{RO}}(\mathcal{C}) + \frac{q_{\mathcal{RO}}}{2^\rho}$. □

F Proof of Theorem 6

Proof. We denote $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_i)$ by the advantage of adversary $(\mathcal{A}_1, \mathcal{A}_2)$ when participating in experiment $\mathbf{G}_i$. We start with game $\mathbf{G}_0$ which is exactly the same game as the ID-CDA game in the $\mathcal{VO}$ model. It means $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_0) = \mathsf{Adv}^{\text{id-cda}}_{\mathsf{IDREwH1},\mathcal{VO}}(\mathcal{A}_1, \mathcal{A}_2)$.

Game $\mathbf{G}_1$: Challenge ciphertext $\mathbf{c} \leftarrow \mathsf{IBE.Enc}_r(params, id^*, \mathbf{m}_\beta; \mathcal{RO}_n(params, id^*, \mathbf{m}_\beta; \mathbf{r}))$ is replaced with $\mathbf{c} \leftarrow \mathsf{IBE.Enc}_r(params, id^*, \mathbf{m}_\beta; \mathbf{r}')$ for randomly chosen $\mathbf{r}'$. All other procedures are computed as the same way in $\mathbf{G}_0$.

```
$\underline{\mathsf{SimC}_{\mathrm{main}}}$
If selective-ID setting
    receive $id^*$ from $\mathcal{B}$
    send $id^*$ to $\mathcal{CH}$
receive $params$ from $\mathcal{CH}$
send $params$ to $\mathcal{B}$
If selective-ID setting
    $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{B}$
    send $(\mathbf{m}_0, \mathbf{m}_1)$ to $\mathcal{CH}$
If full-ID setting
    $(\mathbf{m}_0, \mathbf{m}_1, id^*) \leftarrow \mathcal{B}$
    send $(\mathbf{m}_0, \mathbf{m}_1, id^*)$ to $\mathcal{CH}$
receive $\mathbf{c}$ from $\mathcal{CH}$
send $\mathbf{c}$ to $\mathcal{B}$
receive $\beta'$ from $\mathcal{B}$
return $\beta'$
```

```
$\underline{\mathsf{SimC}_{\mathsf{IBE.Gen}}(id)}$
send $id$ to $\mathsf{IBE.Gen}$ oracle
receive $sk_{id}$ from $\mathsf{IBE.Gen}$ oracle
return $sk_{id}$

$\underline{\mathsf{SimC}_{\mathcal{RO}_n}(M)}$
send $M$ to $\mathcal{RO}$
receive $\mathsf{F}[M]$ from $\mathcal{RO}$
return $\mathsf{F}[M]$

$\underline{\mathsf{SimC}_{\mathcal{RO}^*_v}(M)}$
If $\mathsf{F}^*[M] = \perp$, $\mathsf{F}^*[M] \xleftarrow{\$} \{0,1\}^v$
return $\mathsf{F}^*[M]$;

$\underline{\mathsf{SimC}_{\mathcal{RO}^T_w}(M)}$
If $\mathsf{F}^T[M] = \perp$ then $\mathsf{F}^T[M] \xleftarrow{\$} \{0,1\}^w$
return $\mathsf{F}^T[M]$;
```

```
$\underline{\mathsf{SimC}_{\mathcal{TO}}(y)}$
If $\exists_1 M$ s.t. $\mathsf{F}^T[M] = y$ then return $M$
Otherwise return $\perp$

$\underline{\mathsf{SimC}_E(k,x)}$
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 $\mathsf{E}[k,x]$

$\underline{\mathsf{SimC}_D(k,y)}$
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. 14. Simulation SimC by adversary $\mathcal{C}$

Lemma 8. $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_0)| \leq \frac{q_{\mathcal{RO}}}{2^\mu} + q_{\mathcal{RO}} \cdot \mathsf{maxparams}_{\mathcal{IBE}_r}$.

Proof. The difference between $\mathbf{G}_0$ and $\mathbf{G}_1$ only occurs in two cases: One is the case when adversary $\mathcal{A}_1$ (i.e., without knowledge of *params*) poses $(params, id^*, m_\beta, r)$ to $\mathcal{RO}_n$ where $m_\beta \in \mathbf{m}_\beta$ and $r \in \mathbf{r}$. The other is the case when adversary $\mathcal{A}_2$ (i.e., with knowledge of *params*) poses $(params, id^*, m_\beta, r)$ to $\mathcal{RO}_n$ where $m_\beta \in \mathbf{m}_\beta$ and $r \in \mathbf{r}$. We denote the former event as Bad_1, and the other as Bad_2. From Difference Lemma [30], we have that $|\mathsf{Adv}(\mathcal{B}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{B}, \mathbf{G}_0)| \leq \Pr[\mathsf{Bad}_1 \vee \mathsf{Bad}_2] \leq \Pr[\mathsf{Bad}_1] + \Pr[\mathsf{Bad}_2]$.

First, we estimate $\Pr[\mathsf{Bad}_1]$. Since *params* is not given for $\mathcal{A}_1$, the only way to pose $(params, id^*, m_\beta, r)$ to $\mathcal{RO}_n$ is choosing *params* randomly $q_{\mathcal{RO}}$ times. We have that $\Pr[\mathsf{Bad}_1] \leq q_{\mathcal{RO}} \cdot \mathsf{maxparams}_{\mathcal{IBE}_r}$.

Next, we estimate $\Pr[\mathsf{Bad}_2]$. Since $\mathcal{RO}_n$ is a truly random function, $\mathcal{A}_2$ cannot obtain more information of r (which is used to generate challenge ciphertext $\mathbf{c}$) than min-entropy μ from challenge ciphertext even if $\mathcal{A}_2$ could obtain some information about $\mathcal{RO}_n(params, id^*, \mathbf{m}_\beta; \mathbf{r})$ from $\mathbf{c}$. Thus, the only way to pose $(params, id^*, m_\beta, r)$ to $\mathcal{RO}_n$ is guessing r under min-entropy μ $q_{\mathcal{RO}}$ times. We have that $\Pr[\mathsf{Bad}_2] \leq \frac{q_{\mathcal{RO}}}{2^\mu}$. □

Game $\mathbf{G}_2$: Challenge ciphertext $\mathbf{c} \leftarrow \mathsf{IBE.Enc}_r(params, id^*, \mathbf{m}_\beta; \mathbf{r}')$ is replaced with $\mathbf{c} \leftarrow \mathsf{IBE.Enc}_r(params, id^*, \mathbf{0}; \mathbf{r}')$ for randomly chosen $\mathbf{r}'$ where $\mathbf{0}$ is a vector of l zero strings of length λ. All other procedures are computed as the same way in $\mathbf{G}_1$.

Lemma 9. $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_1)| \leq 2\mathsf{Adv}^{\text{id-cpa}}_{\mathcal{IBE}_r, \mathcal{RO}}(\mathcal{C})$.

Proof. We show that if $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_1)|$ is non-negligible, we can construct an adversary $\mathcal{C}$ breaking ID-CPA security of $\mathcal{IBE}_r$ in the RO model. Fig. 15 shows simulation $\mathsf{SimC}' = (\mathsf{SimC}'_{\mathrm{main}}, \mathsf{SimC}'_{\mathsf{IBE.Gen}}, \mathsf{SimC}'_{\mathcal{RO}}, \mathsf{SimC}'_{\mathcal{RO}^*}, \mathsf{SimC}'_{\mathcal{RO}^T}, \mathsf{SimC}'_{\mathcal{TO}}, \mathsf{SimC}'_E, \mathsf{SimC}'_D)$ by $\mathcal{C}$ respectively.

$\mathcal{C}$ simulates all queries to $\mathcal{VO}$ for $\mathcal{A}_1$ and $\mathcal{A}_2$ with simulation SimC'. SimC' is identical with the definition of $\mathcal{VO}$. Thus, $\mathcal{A}$ cannot distinguish game $\mathbf{G}_1$ and $\mathbf{G}_2$ from the simulation on the interface of $\mathcal{VO}$. If $\beta = 1$ in ID-CPA game for $\mathcal{IBE}_r$, it is clear that all interfaces for $\mathcal{A}$ is exactly same as game $\mathbf{G}_2$. If $\beta = 0$ in ID-CPA game for $\mathcal{IBE}_r$, it is clear that all interfaces for $\mathcal{A}$ is exactly same as game $\mathbf{G}_1$ if $\beta = \beta''$.

Therefore, if $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_1) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_0)|$ is non-negligible, $\mathcal{C}$ also breaks ID-CPA security of $\mathcal{IBE}_r$ if $\beta = \beta''$ (i.e., with probability 1/2). We have that $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_2) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_1)| \leq 2\mathsf{Adv}^{\text{id-cpa}}_{\mathcal{IBE}_r, \mathcal{RO}}(\mathcal{C})$. □

Game $\mathbf{G}_3$: When $\mathcal{A}_2$ poses a query related to $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ (which is the output of $\mathcal{A}_1$) to $\mathcal{RO}^*_v, \mathcal{TRO}_w = (\mathcal{RO}^T_w, \mathcal{TO})$ or $\mathsf{IC}_{a,b} = (E, D)$, then outputs are randomly chosen. That is, tables $\mathsf{F}^*, \mathsf{F}^T, \mathsf{E}$ and D are not preserved for $\mathcal{A}_1$ and $\mathcal{A}_2$ according to $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$. All other procedures are computed as the same way in $\mathbf{G}_2$.

```
$\underline{\mathsf{SimC}'_{\mathrm{main}}}$
$\beta'' \xleftarrow{\$} \{0,1\}$
If selective-ID setting
    receive $id^*$ from $\mathcal{A}_1$
    send $id^*$ to $\mathcal{CH}$
receive $params$ from $\mathcal{CH}$
If selective-ID setting
    $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow \mathcal{A}_1$
    send $(\mathbf{m}''_\beta, \mathbf{0})$ to $\mathcal{CH}$
    receive $\mathbf{c}$ from $\mathcal{CH}$
If full-ID setting
    $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}, id^*) \leftarrow \mathcal{A}_1$
    send $(\mathbf{m}''_\beta, \mathbf{0}, id^*)$ to $\mathcal{CH}$
    receive $\mathbf{c}$ from $\mathcal{CH}$
send $(params, \mathbf{c}, id^*)$ to $\mathcal{A}_2$
receive $\beta'$ from $\mathcal{A}_2$
return 0 if $\beta' = \beta''$ and 1 otherwise
```

```
$\underline{\mathsf{SimC}'_{\mathsf{IBE.Gen}}(id)}$
send $id$ to $\mathsf{IBE.Gen}$ oracle
receive $sk_{id}$ from $\mathsf{IBE.Gen}$ oracle
return $sk_{id}$

$\underline{\mathsf{SimC}'_{\mathcal{RO}_n}(M)}$
send $M$ to $\mathcal{RO}$
receive $\mathsf{F}[M]$ from $\mathcal{RO}$
return $\mathsf{F}[M]$

$\underline{\mathsf{SimC}'_{\mathcal{RO}^*_v}(M)}$
If $\mathsf{F}^*[M] = \perp$, $\mathsf{F}^*[M] \xleftarrow{\$} \{0,1\}^v$
return $\mathsf{F}^*[M]$;

$\underline{\mathsf{SimC}'_{\mathcal{RO}^T_w}(M)}$
If $\mathsf{F}^T[M] = \perp$ then $\mathsf{F}^T[M] \xleftarrow{\$} \{0,1\}^w$
return $\mathsf{F}^T[M]$;
```

```
$\underline{\mathsf{SimC}'_{\mathcal{TO}}(y)}$
If $\exists_1 M$ s.t. $\mathsf{F}^T[M] = y$ then return $M$
Otherwise return $\perp$

$\underline{\mathsf{SimC}'_E(k, x)}$
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 $\mathsf{E}[k, x]$

$\underline{\mathsf{SimC}'_D(k, y)}$
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. 15. Simulation SimC' by adversary $\mathcal{C}$

Lemma 10. $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_2)| \leq \frac{4q^2_{\mathcal{RO}^*} + 4q^2_{\mathcal{RO}^T} + 4q^2_{\mathcal{TO}} + 4q^2_E + 4q^2_D}{2^\mu}$.

Proof. The difference between $\mathbf{G}_2$ and $\mathbf{G}_3$ only occurs when $\mathcal{A}_1$ and $\mathcal{A}_2$ poses a same query related to $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ to $\mathcal{RO}^*_v, \mathcal{TRO}_w = (\mathcal{RO}^T_w, \mathcal{TO})$ or $\mathsf{IC}_{a,b} = (E, D)$. We denote the event that a common query related to $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ is posed to $\mathcal{RO}^*_v$ by $\mathcal{A}_1$ and $\mathcal{A}_2$ as $\mathsf{Bad}_{\mathcal{RO}^*}$. Similarly, we define events $\mathsf{Bad}_{\mathcal{RO}^T}$, $\mathsf{Bad}_{\mathcal{TO}}$, Bad_E, and Bad_D. From Difference Lemma [30], we have that $|\mathsf{Adv}(\mathcal{B}, \mathbf{G}_3) - \mathsf{Adv}(\mathcal{B}, \mathbf{G}_2)| \leq \Pr[\mathsf{Bad}_{\mathcal{RO}^*} \vee \mathsf{Bad}_{\mathcal{RO}^T} \vee \mathsf{Bad}_{\mathcal{TO}} \vee \mathsf{Bad}_E \vee \mathsf{Bad}_D] \leq \Pr[\mathsf{Bad}_{\mathcal{RO}^*}] + \Pr[\mathsf{Bad}_{\mathcal{RO}^T}] + \Pr[\mathsf{Bad}_{\mathcal{TO}}] + \Pr[\mathsf{Bad}_E] + \Pr[\mathsf{Bad}_D]$.

In game $\mathbf{G}_2$ and $\mathbf{G}_3$, 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 $\mathbf{G}_2$ and $\mathbf{G}_3$ are identical. Thus, the only way to pose such a query is guessing under min-entropy μ. According to the birthday paradox, for each oracle the probability of collisions in guessing is at most $(2q_{\mathcal{RO}^*})^2/2^\mu$, $(2q_{\mathcal{RO}^T})^2/2^\mu$, $(2q_{\mathcal{TO}})^2/2^\mu$, $(2q_E)^2/2^\mu$, and $(2q_D)^2/2^\mu$, respectively. Therefore, $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_2)| \leq (4q^2_{\mathcal{RO}^*} + 4q^2_{\mathcal{RO}^T} + 4q^2_{\mathcal{TO}} + 4q^2_E + 4q^2_D)/2^\mu$. □

We estimate $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3)$. Ciphertext $\mathbf{c}$ does not give any information about $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$. Also, outputs of $\mathcal{VO}$ is independent of $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ for $\mathcal{A}_2$. Thus, the only way to win in game $\mathbf{G}_3$ is randomly guessing β. Therefore, $\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3) = 0$.

To conclude, we have $\mathsf{Adv}^{\text{id-cda}}_{\mathsf{IDREwH1}, \mathcal{VO}}(\mathcal{A}_1, \mathcal{A}_2) \leq 2\mathsf{Adv}^{\text{id-cpa}}_{\mathcal{IBE}_r, \mathcal{RO}}(\mathcal{C}) + q_{\mathcal{RO}} \cdot \mathsf{maxparams}_{\mathcal{IBE}_r} + (q_{\mathcal{RO}} + 4q^2_{\mathcal{RO}^*} + 4q^2_{\mathcal{RO}^T} + 4q^2_{\mathcal{TO}} + 4q^2_E + 4q^2_D)/2^\mu$. □