On Indifferentiable Hash Functions in Multi-Stage Security Games

Yusuke Naito, Kazuki Yoneyama, and Kazuo Ohta

Mitsubishi Electric Corporation, NTT Corporation and UEC

Abstract. Ristenpart, Shacham, and Shrimpton (RSS) demonstrated that there exists a cryptosystem $\mathcal{C}$ which is secure in a certain multi-stage game in the random oracle (RO) model but insecure even when using an indifferentiable (from a RO) hash function H such as the chop Merkle-Damgård hash function (ChopMDHF) and the fixed output length Sponge hash function (FOLSpongeHF). However, the RSS result does not imply that for every cryptosystem $\mathcal{C}$ which is secure in a multi-stage game in the RO model, $\mathcal{C}(H)$ is always insecure. There might exist a cryptosystem $\mathcal{C}$ which is secure in a multi-stage game when using H. In this paper, we show that for each of cryptosystems EwH, REwH1 and IDREwH1 and each of the FOLSpongeHF and the ChopMDHF $\mathcal{C}(H)$ is secure in the Chosen Distribution Attack (CDA) game. EwH and REwH1 are public key encryption schemes. IDREwH1 is an ID-based encryption scheme which is an ID-based version of REwH1. To prove the security, we propose a modular approach by adopting the reset indifferentiability framework of RSS, which covers all games. Firstly, we introduce a new weakened RO, called "Versatile Oracle" ($\mathcal{VO}$), since H might not be used as a RO in the CDA game. Secondly, we show that H are reset indifferentiable from a $\mathcal{VO}$. That is, H can be used as a $\mathcal{VO}$. Finally, we show that $\mathcal{C}(\mathcal{VO})$ are CDA secure. Consequently, the reset indifferentiability framework ensures that $\mathcal{C}(H)$ is CDA secure.
Keywords. Indifferentiable hash function, reset indifferentiability, multi-stage game, Sponge, ChopMD.

1 Introduction

The indifferentiability framework of Maurer, Renner, and Holenstein (MRH) [19] ensures reducibility from one system to another system. Coron *el al.* suggested applying the indifferentiability framework to hash function design [13]. Let $\mathcal{C}(\cdot)$ be a cryptosystem with access to a hash function H, denoted by $\mathcal{C}(H)$. The MRH theorem states that

H is indifferentiable from a random oracle (RO), denoted $H \sqsubset \mathcal{RO}$
$\Rightarrow$ For any cryptosystem $\mathcal{C}(\mathcal{RO})$ which is **S** secure, $\mathcal{C}(H)$ is **S** secure, denoted $\mathcal{C}(H) \succ \mathcal{C}(\mathcal{RO})$.

Note that we say a cryptosystem $\mathcal{C}(H)$ is **S** secure if there does not exist an adversary which wins a game **S**. After their suggestion, several hash constructions which are indifferentiable from ROs have been proposed such as the Sponge construction [5] and the chop Merkle-Damgård (ChopMD) construction [13]. Hereafter, we omit the term "from a RO". It has been widely believed that the MRH theorem ensures all games.

However, Ristenpart, Shacham, and Shrimpton (RSS) [23] gave a counter example of the MRH theorem: Let H be an indifferentiable hash function such as the Sponge hash function (SpongeHF) and the ChopMD hash function (ChopMDHF).

$$\exists \mathcal{C} \text{ such that } \mathcal{C}(\mathcal{RO}) \text{ is } \mathbf{S} \text{ secure but } \mathcal{C}(H) \text{ is insecure, even though } H \sqsubset \mathcal{RO}.$$

RSS considered a hash based authentication protocol as $\mathcal{C}$. The client sends a random challenge C to the server; the server proves possession of the file M by computing $z = H(M||C)$ where H is a hash function and the client compares the result to that sent by the server. In this case, the server cannot cheat in the $\mathcal{RO}$ model, while the server can cheat when using an indifferentiable hash function. Consider the case that $|M| = |C| = 4n$ and H is the ChopMDHF, $\mathrm{chopMD}^f(M||C) = chop_n(f(f(IV||M)||C))$, where $f : \{0,1\}^{6n} \rightarrow \{0,1\}^{2n}$ is a RO and $chop_n$ returns the first n bits. The server can response a valid value for a challenge C by possessing only the chaining value $cv = f(IV||M)$. That is, he can discard M. RSS formalized this situation as the following game **S**: In this game, two adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ are appeared. In the first stage, $\mathcal{A}_1$ receives a random file M of $4n$ bit and outputs a value v such that $|v| \leq 2n$. In the second stage, $\mathcal{A}_2$ receives the value v and a random challenge C of $4n$ bit and outputs

an n-bit value z. An adversary wins if $z = H(M||C)$. When $H = \mathcal{RO}$, an adversary cannot win, while when $H = \mathrm{chopMD}^f$, he wins, since $\mathcal{A}_1$ can send $cv = f(IV||M)$ to $\mathcal{A}_2$.

The above game is a member of *multi-stage* games, where there are multi adversaries and the size of a value sent from some-stage adversary to the next-stage adversary is restricted. From the RSS result, it is doubtful whether the MRH theorem supports multi-stage games or not. RSS revisited the MRH theorem and found that the MRH theorem supports *only* single-stage games. On the other hand, the RSS result does not imply that for every cryptosystem $\mathcal{C}$ which has been proven secure in a multi-stage game $\mathbf{S}$ in the $\mathcal{RO}$ model, $\mathcal{C}(H)$ is always insecure. There might exist cryptosystems $\mathcal{C}$ that remain $\mathbf{S}$ secure when using an indifferentiable hash function H. Thus it is important to verify whether $\mathcal{C}(H)$ remains $\mathbf{S}$ secure or not.

There are several multi-stage games such as the Chosen Distribution Attack (CDA) game [1, 2]. The CDA notion captures message privacy of a public key encryption (PKE) scheme when messages and randomness are (jointly) unpredictable. There are two PKE settings in the CDA game, which are a deterministic public key encryption (DPKE) setting [1, 3, 7, 16, 20] and a hedged PKE (HPKE) setting [2]. DPKE schemes don't use randomness, while HPKE schemes use randomness. Note that when considering a DPKE scheme in the CDA game, randomness is omitted. CDA secure DPKE schemes realize efficient searchable PKE schemes. CDA secure HPKE schemes can ensure message privacy from bad randomness. Therefore, CDA secure schemes can be used in many practical cases. Thus, it is important to consider the CDA secure schemes as the above cryptosystem $\mathcal{C}$.

Let $\mathcal{C}^{\mathsf{pke}}$ be a PKE scheme which uses a semantically secure (CPA secure) PKE scheme $\mathcal{C}^{\mathsf{cpa}}$ and a hash function H as building blocks, denoted by $\mathcal{C}^{\mathsf{pke}}(\mathcal{C}^{\mathsf{cpa}}, H)$. Several CDA secure PKE schemes such as the EwH DPKE scheme [1] and the $\mathsf{REwH1}$ PKE scheme [2] have been proposed. $\mathsf{EwH}(\mathcal{C}^{\mathsf{cpa}}, H)$ and $\mathsf{REwH1}(\mathcal{C}^{\mathsf{cpa}}, H)$ are CDA secure if H is a RO. Therefore, EwH (resp. $\mathsf{REwH1}$) can generically convert a CPA secure PKE scheme to a CDA secure DPKE (resp. HPKE) scheme. If $H \sqsubset \mathcal{RO}$ and $\mathsf{EwH}(\mathcal{C}^{\mathsf{cpa}}, H)$ and $\mathsf{REwH1}(\mathcal{C}^{\mathsf{cpa}}, H)$ are CDA secure, H can be used as hash functions in many CDA secure PKE schemes via EwH and $\mathsf{REwH1}$. Because, we have thousands of CPA secure PKE schemes. Thus it is important to consider EwH and $\mathsf{REwH1}$.

The above case considers a PKE setting, while it is natural to consider an extension to an ID-based setting. It is a meaningful and interesting problem to clarify if we generically construct an ID-based CDA (ID-CDA) secure ID-based encryption (IBE) scheme $\mathcal{C}^{\mathsf{ibe}}(\mathcal{C}^{\mathsf{idcpa}}, H)$ from an ID-based CPA (ID-CPA) secure IBE $\mathcal{C}^{\mathsf{idcpa}}$ and an indifferentiable hash function H, similarly.

RSS [23] *directly* proved that $\mathsf{EwH}(\mathcal{C}^{\mathsf{cpa}}, H)$ and $\mathsf{REwH1}(\mathcal{C}^{\mathsf{cpa}}, H)$ are CDA secure if H is the NMAC hash function [15]. On the other hand, it is unclear whether or not the same results hold when H is an indifferentiable hash function other than NMAC. Also, RSS did not consider the ID-based setting. Especially, it is important to consider the fixed output length Sponge (FOLSponge) construction and the ChopMD construction which are employed in standard hash functions: FOLSponge, which is the special case of Sponge, is employed in the SHA-3 hash functions Keccak-l $l = 224, 256, 384, 512$ [6], and ChopMD is employed in SHA-512/224 and SHA-512/256 [22]. Moreover, the RSS approach requires complex and many proofs: it makes proofs complex that the structures of a hash function and of a cryptosystem must be considered at the same time, and when considering cryptosystems $\mathcal{C}_i$ $(i = 1, \ldots, a)$ and hash functions H_j $(j = 1, \ldots, b)$, we need many distinct proofs because the security of $\mathcal{C}_i(H_j)$ has to be proven for each i, j respectively. Therefore, it is desirable to consider how to avoid the direct proofs by taking a *modular* approach.

Our Result. In this paper, we solve the open problem related to the RSS result. We show that for each H of the FOLSponge hash function (FOLSpongeHF) and the ChopMDHF, and each $\mathcal{C}^{\mathsf{pke}}$ of EwH and $\mathsf{REwH1}$, $\mathcal{C}^{\mathsf{pke}}(\mathcal{C}^{\mathsf{cpa}}, H)$ is CDA secure.

We also consider an ID-based setting. We find an IBE scheme $\mathcal{C}^{\mathsf{ibe}}(\mathcal{C}^{\mathsf{idcpa}}, H)$ which is ID-CDA secure for each H of the FOLSpongeHF and the ChopMDHF. We introduce a new IBE scheme $\mathsf{IDREwH1}$ which

is an ID-based version of REwH1. IDREwH1 ensures that any ID-CPA secure IBE can be converted to ID-CDA secure. So, we can construct various ID-CDA secure IBE with the FOLSpongeHF and the ChopMDHF via IDREwH1, since we have hundreds of ID-CPA secure IBE schemes.

Our Approach. In this paper, we propose a "modular" approach. Since indifferentiable hash functions such as the ChopMDHF and the FOLSpongeHF cannot be used as ROs in a certain multi-stage game from the RSS result, we weaken a RO such that (1) indifferentiable hash functions H can be used as weakened ROs (WROs) in all games, and (2) it is sufficient to prove that target cryptosystems $\mathcal{C}$ are (ID-)CDA secure in the WRO model. Hereafter, we assume that when considering a PKE scheme such as EwH and REwH1, $\mathcal{C}(\cdot) = \mathcal{C}^{\mathsf{pke}}(\mathcal{C}^{\mathsf{cpa}}, \cdot)$, and when considering a IBE scheme such as IDREwH1, $\mathcal{C}(\cdot) = \mathcal{C}^{\mathsf{ibe}}(\mathcal{C}^{\mathsf{idcpa}}, \cdot)$. Though EwH and REwH1 have been proven to be CDA secure in the RO model, we will prove that the similar result in the WRO model. Since the reset indifferentiability framework [23] introduced by RSS fits this approach, we use this framework to combine the point (1) with the point (2). See the following explanation of this framework.

In the point (1), we will define a WRO $\mathcal{WRO}$ such that H can be used as a $\mathcal{WRO}$. $\mathcal{WRO}$ consists of a RO $\mathcal{RO}$ and a sub oracle $\mathcal{O}^*$. $\mathcal{O}^*$ leaks several values of $\mathcal{RO}$ such as query-responses of $\mathcal{RO}$. On the other hand, in the point (2), a CDA adversary might obtain advantages to win the CDA game from $\mathcal{O}^*$. Therefore, we must carefully define $\mathcal{WRO}$ such that both the point (1) and the point (2) hold, simultaneously.

By the modular approach, we can avoid complex and many proofs.

- Our approach can avoid complex proofs, since the structure of $\mathcal{C}$ and the structure of H separately via the $\mathcal{WRO}$ model.
- Our approach can avoid many proofs, since what we have to prove is only that (1) for each i, $\mathcal{C}_i(\mathcal{WRO})$ is secure in the (ID-)CDA game, and (2) for each j, H_j can be used as a $\mathcal{WRO}$.

Reset Indifferentiability [23]. The reset indifferentiability framework is an extension of the original indifferentiability framework [19]. This framework ensures that for *all* games

$$H \text{ is reset indifferentiable from an oracle } \mathcal{O} \text{ (denoted } H \sqsubset_r \mathcal{O}) \Rightarrow \mathcal{C}(H) \succ \mathcal{C}(\mathcal{O}).$$

In this paper, we discuss $\mathcal{WRO}$ as $\mathcal{O}$. Therefore, the point (1) is to show that $H \sqsubset_r \mathcal{WRO}$, and thus, we can combine the point (1) with the point (2) from the reset indifferentiability framework.

Roughly speaking, H using an underlying primitive $\mathcal{U}$ (denoted by $H^{\mathcal{U}}$) is reset indifferentiable from a $\mathcal{WRO}$ if there exists a simulator S such that no distinguisher A can distinguish a real world from an ideal world. A interacts with a left oracle L and a right oracle R. In the real world, $(L, R) = (H, \mathcal{U})$ and A has access to *nop* which does nothing. In the ideal world, $(L, R) = (\mathcal{RO}, S)$ and A has access to *S.Ret* which reinitializes the internal values of S to the initial values. The role of S is to simulate $\mathcal{U}$ such that S is consistent with $\mathcal{RO}$ as the real world. That is, since in the real world, the relation $L(M)$ $(= H^{\mathcal{U}}(M)) = H^R(M)$ holds, S must ensure that this relation must be hold in the ideal world. Note that $H^R(M)$ means that all query-response values to calculate $H^R(M)$ are obtained by only R queries. Note that A does not have oracle access to $\mathcal{O}^*$ and S has oracle access $\mathcal{RO}$ and $\mathcal{O}^*$.

In this paper, we consider a stateless simulator for simplicity. Under a stateless simulator, we can omit *S.Rst* and *nop*. The reset indifferentiable security with *S.Rst* and *nop* can be ensured by one without *S.Rst* and *nop* where S is stateless, since we restrict the structure of S and a stateless simulator is not affected by *S.Rst*.

Versatile Oracle. We define $\mathcal{WRO}$ such that the two points hold: (1) $H^{\mathcal{U}} \sqsubset_r \mathcal{WRO}$ and (2) the target cryptosystems $\mathcal{C}$ are CDA secure in the $\mathcal{WRO}$ model. We call the WRO "Versatile Oracle" denoted by $\mathcal{VO}$. "Versatile" means that $\mathcal{VO}$ supports several hash constructions such as FOLSponge and ChopMD.

For the point (1), to prove that $H^{\mathcal{U}} \sqsubset_r \mathcal{VO}$, we must construct a stateless simulator S which achieves two situations, *extractable* and *unchangeable* situations. In order for a simulator to realize these situations, we define a sub oracle as $\mathcal{O}^* = (\mathcal{TRO}, \mathcal{RO}^*, \mathsf{IC})$ where IC is an ideal cipher, $\mathcal{RO}^*$ is a RO and $\mathcal{TRO}$ is a traceable RO (TRO) introduced by Naito *et al.* [21] (See the following)

- **Extractable Situation** was pointed out by Dodis *et al.* [15], where S needs to extract a message from a query to ensure the consistency with $\mathcal{RO}$. Consider the NMAC construction $H^{g,h}(M) = g(h(M))$ as an example case. In this case, a simulator $S = (S_h, S_g)$ must be such that for any M the relation $L(M)$ $(= \mathcal{RO}(M)) = S_g(S_h(M))$ holds. S_g is a simulator of g and S_h is a simulator of h. When a query $S_g(z)$ is made such that $z = S_h(M)$, S_g needs to extract M from the query-response pair (M, z) of S_h to define the response which is equal to the output of $\mathcal{RO}(M)$. However, since a stateless simulator S cannot record its query-response pairs, S cannot extract M.
  In order for S to achieve the extractable situation, we add $\mathcal{TRO}$ to $\mathcal{O}^*$. $\mathcal{TRO}$ consists of a RO $\mathcal{RO}^T$ and a trace oracle $\mathcal{TO}$, where for a query $\mathcal{TO}(z)$, if a query $\mathcal{RO}^T(M)$ which is equal to z was already made, $\mathcal{TO}$ returns M, and otherwise, $\mathcal{TO}$ returns $\perp$. By using $\mathcal{TRO}$, S can extract M from z. Consider the NMAC case, again. First, for a query $S_h(M)$ S_h returns $z = \mathcal{RO}^T(M)$. Then, for a query $S_g(z)$ S_g can extract M by the query $\mathcal{TO}(z)$ and returns $z = \mathcal{RO}(M)$. That is, S can achieve the extractable situation. Note that in this case, S does not use memory.
- **Unchangeable Situation** is that for a repeated query to a simulator S S must respond the same value which was returned before. Usually, the underlying primitives $\mathcal{U}$ of indifferentiable hash functions are an ideal cipher, a random permutation and a fixed input length RO (FILRO). These ideal primitives are such that when a query $\mathcal{U}(w)$ was already made and the response was v, the response of the repeated query $\mathcal{U}(w)$ is v. On the other hand, since a stateless simulator S cannot record query-response values, for a repeated query to S, S cannot response the same value which was returned before.
  In order for S to realize the unchangeable situation, we add an ideal cipher IC and a RO $\mathcal{RO}^*$. Consider the case that S simulates a FILRO. For a query $S(w)$ S returns $v = \mathcal{RO}^*(w)$. For a repeated query $S(w)$, S can respond the same value v by setting $v = \mathcal{RO}^*(w)$. Note that in this case, the simulator does not use memory. Similarly, by using IC, we can construct a stateless simulator which achieves the unchangeable situation, when simulating an ideal cipher or a random permutation.

For the point (2), we must ensure that the sub oracle $\mathcal{O}^*$ gives no advantage to an adversary in the CDA game. The CDA game consists of two stages, where a first stage adversary $\mathcal{A}_1$ sends no value to a second stage adversary $\mathcal{A}_2$.[1] First, we suppose that the challenge ciphertext c_β does not leak any information of messages (m_0, m_1) even with access to $\mathcal{RO}$. This property is guaranteed by assuming that a scheme satisfies some weak properties such as the IND-SIM security [23] in the RO model. Then, since $\mathcal{RO}^T$ of $\mathcal{TRO}$, $\mathcal{RO}^*$ are ideal primitives whose outputs do not leak no information for the inputs, these oracles give no advantage to the adversary. $\mathcal{A}_1$ might deliver some information about messages via interfaces of $\mathcal{TO}$ and IC by posing a value to $\mathcal{RO}^T$, E, or D, where E is an encryption oracle of IC and D is a decryption oracle. If $\mathcal{A}_2$ could pose the corresponding output value to $\mathcal{TO}$, D, or E, he would obtain the value from $\mathcal{A}_1$. However, $\mathcal{A}_2$ cannot find the output value which are defined in the first stage, because c_β does not leak any information of (m_0, m_1), and outputs of $\mathcal{RO}^T$, E, and D are uniformly random. Therefore, $\mathcal{TO}$ and IC also give no advantage to the adversary.

Result for Point (1). In Section 4, we show that *Sponge* $\sqsubset_r \mathcal{VO}$ and chopMD $\sqsubset_r \mathcal{VO}$. Since a stateless simulator can realize extractable and unchangeable situations by using a $\mathcal{VO}$, we can prove

[1] In the first stage, an adversary $\mathcal{A}_1$ outputs two messages (m_0, m_1) and a random value r such that the jointed values $m_i||r$ have sufficient min-entropy. In the second stage, an adversary $\mathcal{A}_2$ receives the challenge cipher text $c_\beta = \mathcal{E}(m_\beta; r)$ from the game where β is a random value of a single bit, and outputs a bit b, where $\mathcal{E}$ is an encryption function. The adversary wins if $b = \beta$. Note that the length $|r|$ of a random value is 0 when considering DPKE schemes.

them. Similarly, we can obtain the same results for the prefix-free MD construction [13], the EMD construction [4], and the MDP construction [17]. As an example, in Appendix E, we prove that PFMD $\sqsubset_r$ $\mathcal{VO}$.

Result for Point (2). In Section 5, we show that EwH, REwH1 and IDREwH1 are 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 the IND-SIM security. RSS [23] showed that EwH and REwH1 are IND-SIM secure if an underlying PKE scheme is CPA secure in the RO model. Thus, we obtain CDA secure PKE in the $\mathcal{VO}$ model from any CPA secure PKE in the RO model by combining them.

For IBE schemes, we propose a generic construction of IBE, IDREwH1, which is an ID-based version of the REwH1 PKE scheme. IDREwH1 contains an ID-CPA secure IBE scheme as a building block. We show that IDREwH1 is ID-CDA secure in the $\mathcal{VO}$ model if an underlying IBE scheme is ID-CPA secure in the RO model. As far as we know, IDREwH1 is the first explicit formulation and construction for deterministic IBE and hedged IBE.

Related Works. There are two independent works regarding the RSS results [23]. However, these papers don't include results to salvage indifferentiable hash constructions such as Sponge and ChopMD in the CDA games.

Luykx *et al.* showed that there are no meaningful hash constructions which are reset indifferentiable from ROs [18]. This does not salvage indifferentiable hash constructions. Since the reset indifferentiability from a RO cannot salvage meaningful hash constructions, it is plausible to consider a WRO to salvage them.

Demay *et al.* introduced an extended framework of the indifferentiability [14]. This framework restricts the size of simulator's memory. This covers multi-stage games where for each stage, the size of a value sent from the adversary to the next stage adversary is over those of simulator's memory. They clarified the size of simulator's memory to ensure the restricted indifferentiability from a RO. The size of simulator's memory must not be 0, while in the CDA game, the first stage adversary sends no value to the second stage adversary. Therefore, their result does not salvage indifferentiable hash constructions in the CDA game.

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 Boolean function F, we denote by "$\exists_1 M$ s.t. $F(M)$ is true" "there exists just a value M such that $F(M)$ is true". Vectors are written in boldface, e.g., $\mathbf{x}$. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes its length and $\mathbf{x}[i]$ denotes its i-th component for $1 \leq i \leq |\mathbf{x}|$. $bit_j(\mathbf{x})$ is the left j-th bit of $\mathbf{x}[1]||\ldots||\mathbf{x}[|\mathbf{x}|]$.

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

Functionalities. This papers treat *functionalities*. A functionality provides two interfaces. These interfaces are referred to as private and public. A private interface is accessed by honest parties such as a cryptosystem. A public interface is accessed by adversaries. In this paper, the private interface of a functionality F is denoted by $F.priv$ and the public interface is denoted by $F.pub$. Functionalities

treated in this paper are ideal primitives $\mathcal{I}$ and hash functions $H^{\mathcal{U}}$ using an ideal primitive $\mathcal{U}$. Ideal primitives $\mathcal{I}$ discussed in this paper are an arbitrary input length random oracle (simply denoted by RO) and a weakened RO ($\mathcal{WRO}$) defined in the next section. Hash functions discussed in this paper are the fixed output length Sponge hash function (FOLSpongeHF) [5], and the chop Merkle-Damgård hash function (ChopMDHF) [13]. The underlying ideal primitives $\mathcal{U}$ are a random permutation (RP) when H is the FOLSpongeHF, and a fixed input length RO (FILRO) when H is the ChopMDHF.

Since a RO offers a single interface, this paper does *not* distinguish the public interface and the private interface. On the other hand, we distinguish the interfaces of a $\mathcal{WRO}$ defined in the next section and the interfaces of hash functions. The interfaces of a hash function $H^{\mathcal{U}}$ using an ideal primitive $\mathcal{U}$ are defined as $H^{\mathcal{U}}.priv = H^{\mathcal{U}}$ and $H^{\mathcal{U}}.pub = \mathcal{U}$.

Games. A game G consists of a single main procedure. A game can make use of a functionality F, a cryptographic scheme $\mathcal{C}$ and a number of adversarial procedures $\mathcal{A}_1, \ldots, \mathcal{A}_m$ together referred to as the adversary. We denote this game by $G^{\mathcal{A}_1,\ldots,\mathcal{A}_m}_{\mathcal{C},F}$. $\mathcal{C}$ has access to $F.priv$. $\mathcal{A}_1, \ldots, \mathcal{A}_m$ have accesses to $F.pub$ and $\mathcal{C}$. Running a game $G^{\mathcal{A}_1,\ldots,\mathcal{A}_m}_{\mathcal{C},F}$ means executing the sequence of statements of the game's procedure and the output of G is true or false.

This paper divides games into single-stage games and multi-stage games. In single-stage games, a single adversary is appeared or the size of a value sent from every adversary to the next-stage adversary is not restricted. Examples of single-stage games are the collision security game and the IND-CCA game. On the other hand, in multi-stage games, multi-adversaries are appeared and the size of a value sent from some-stage adversary to the next-stage adversary is restricted. An example of multi-state games is the Chosen Distribution Attack (CDA) security game [1, 2]. The CDA security game is the two-stage game, where the first-stage adversary sends no value to the second-stage adversary. Please see the explicit formula of the CDA security game in Section 5.

Indifferentiability Frameworks [19, 23]. The indifferentiability framework ensures reducibility from one system to another system in single-stage security games. Let F_1 and F_2 are two functionalities. The indifferentiable advantage of F_1 from F_2 is defined as follows [19].

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

$S^{F_2.pub}$ is a simulator which has access to $F_2.pub$. S's task is to simulate $F_1.pub$ such that S is consistent with $F_2.priv$ as $(F_1.priv, F_1.pub)$. A is a distinguisher which has accesses to two interfaces of F_1, or $F_2.priv$ and S. In the F_1 case, the distinguisher is denoted by $A^{F_1.priv,F_1.pub}$. In the F_2 case, the distinguisher is denoted by $A^{F_2.priv,S^{F_2.pub}}$. "$A^{\mathcal{O}_1,\mathcal{O}_2} \Rightarrow 1$" is that the distinguisher A, which has accesses to two interfaces $\mathcal{O}_1, \mathcal{O}_2$, outputs a bit 1. We say F_1 is indifferentiable from F_2 if there exists a simulator such that for any distinguisher A the advantage is negligible in the security parameter. If F_1 is indifferentiable from F_2, then for any cryptosystem $\mathcal{C}$, $\mathcal{C}(F_1)$ is at least as secure as $\mathcal{C}(F_2)$ when the security game is a single-stage security game.

Reset indifferentiability [23] is the stronger security notion than indifferentiability. In this case, a distinguisher A can reinitialize internal values of a simulator to the initial values. When a simulator is stateless, one can omit such a reset capacity. Therefore, the reset indifferentiable advantage is bounded by the indifferentiable advantage where a simulator is stateless. This fact is also referred by Demay *et al.* [14]. We call the stateless indifferentiability "SL-indifferentiability". A functionality F_1 is SL-indifferentiable from a functionality F_2 if there exists a stateless simulator S such that for any distinguisher A the SL-indifferentiable advantage is negligible in the security parameter. We denote by $\mathsf{Adv}^{\mathsf{sl\text{-}indiff},F_2}_{F_1,S}(A)$ the SL-indifferentiable advantage of F_1 from F_2. 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 $(F_2.priv, S^{F_2.pub})$. 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 A interacts with (L, R). 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).

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 $\quad s = P(s \oplus (M_i||0^c))$;
6 **return** $s[1, n]$;

Fig. 1. Sponge

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

Fig. 2. Chop Merkle-Damgård

The reset indifferentiability framework ensures that if F_1 is SL-indifferentiable from F_2, then for any cryptosystem $\mathcal{C}(\cdot)$ and any game, the cryptosystem $\mathcal{C}(F_1)$ is at least as secure as the cryptosystem $\mathcal{C}(F_2)$. More precisely, the reset indifferentiability framework ensures the following theorem [23].

Theorem 1 ([23]). *Let G be any security game. Let F_1 and F_2 be functionalities. Let $\mathcal{A}_1, \ldots, \mathcal{A}_m$ be adversary and let S be a stateless simulator. Then there exist an adversary $\mathcal{B}_1, \ldots, \mathcal{B}_m$ and distinguisher A such that*

$$\Pr[G^{\mathcal{A}_1,\ldots,\mathcal{A}_m}_{\mathcal{C},F_1} \Rightarrow \mathsf{true}] \leq \Pr[G^{\mathcal{B}_1,\ldots,\mathcal{B}_m}_{\mathcal{C},F_2} \Rightarrow \mathsf{true}] + \mathsf{Adv}^{\mathsf{sl\text{-}indiff},F_2}_{F_1,S}(A).$$

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

Definitions of Hash Functions. We give the description of the FOLSponge construction [5]. Let P be a permutation of d bits. The FOLSongeHF $Sponge^P : \{0,1\}^* \to \{0,1\}^n$ is defined in Fig. 1 such that $n < d$.[2] Let $c = d - n$. $\mathsf{pad}_S : \{0,1\}^* \to (\{0,1\}^n)^*$ is an injective padding function such that the last n-bit value is not 0. IV is a constant value of d bits. $IV_1 = IV[1, n]$ and $IV_2 = IV[n+1, d]$. For example, $\mathsf{pad}_S(M) = M||1||0^i$ where i is a smallest value such that $|M||1||0^i|$ is a multiple of n.

We give the description of the ChopMD construction [13]. Let h be a compression function which maps a value of $d + n + s$ bits to a value of $n + s$ bits. The ChopMDHF $\mathrm{chopMD}^h : \{0,1\}^* \to \{0,1\}^n$ 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.

3 Versatile Oracle

In this section, we define a new $\mathcal{WRO}$, called Versatile Oracle ($\mathcal{VO}$) such that indifferentiable hash functions such as the ChopMDHF and the FOLSpongeHF can be used as $\mathcal{VO}$s and the CDA security of schemes, which are CDA secure in the RO model, is ensured in the $\mathcal{VO}$ model. We use the reset indifferentiability framework to ensure that a hash function can be used as a $\mathcal{VO}$. In Section 4, we prove that the ChopMDHF and the FOLSpongeHF are SL-indifferentiable from $\mathcal{VO}$s. In Section 5, we prove that the several schemes are CDA secure in the $\mathcal{VO}$ model. Consequently, Theorem 1 ensures the CDA security of these schemes for each of the ChopMDHF and the FOLSpongeHF.

We define $\mathcal{VO}$ as $(\mathcal{RO}_n, \mathcal{TRO}_w, \mathcal{RO}^*_v, \mathsf{IC}_{a,b})$. The interfaces are defined by $\mathcal{VO}.priv = \mathcal{RO}_n$ and $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{RO}^*_v, \mathcal{TRO}_w, \mathsf{IC}_{a,b})$. $\mathcal{RO}_n$ and $\mathcal{RO}^*_v$ are arbitrary input length ROs, where the output lengths are n bit and v bit, respectively. $\mathsf{IC}_{a,b} = (E, D)$ is an ideal cipher where the key length is a bit and the block length is b bit. $E : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}^b$ is an encryption oracle and $D : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}^b$ is a decryption oracle. $\mathcal{TRO}_w = (\mathcal{RO}^T_w, \mathcal{TO})$ is a Traceable RO (TRO) which

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

was proposed by Naito *et al.* [21]. $\mathcal{RO}_w^T$ is an arbitrary input length RO where the output length is w bit. $\mathcal{TO}$ is a trace oracle where for query $\mathcal{TO}(z)$ such that $|z| = w$, it returns M if $\exists_1 M$ such that a query $\mathcal{RO}_w^T(M)$ was already made, and it returns $\perp$ otherwise. Note that the parameters (n, w, v, a, b) are defined in each hash function. Note that $\mathcal{RO}_n$ offers the both public and private interfaces, while $\mathcal{RO}_v^*, \mathcal{RO}_w^T$ offer only public interfaces. In Appendix B, we give the method of implementing a $\mathcal{VO}$.

SL-Indifferentiability from $\mathcal{VO}$. Let $H^{\mathcal{U}}$ be a target hash function and $\mathcal{U}$ is the underlying ideal primitive. Then the advantage of the SL-indifferentiability from a $\mathcal{VO}$ is defined as follows.

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

where S is a *stateless* simulator.

The $\mathcal{VO}$ Model. From the definition of the reset indifferentiability (See Section 2), in the $\mathcal{VO}$ model, an distinguisher has access to the public interface $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{TRO}_w, \mathcal{RO}_v^*, \mathsf{IC}_{a,b})$. Cryptosystems have access to the private interface $\mathcal{VO}.priv = \mathcal{RO}_n$.

4 $\mathcal{VO}$ Hash Functions

In this section, we show that the ChopMDHF and the FOLSpongeHF are SL-indifferentiable from $\mathcal{VO}$s. To prove the SL-indifferentiable security, we must construct a stateless simulator S which is consistent with $\mathcal{RO}_n$ as in the real world. In order to construct such simulator S, S must achieve two situations which are extractable and unchangeable situations. The extractable situation is that S extracts a message from a query to ensure the consistency with $\mathcal{RO}_n$. The unchangeable situation is that for a repeated query to S, S always returns the same value. In the following, we demonstrate that S which achieves these situations can be constructed.

4.1 SL-Indifferentiability for ChopMD

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

Theorem 2. *There exists a stateless simulator S such that for any distinguisher A,*

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

where 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 $\mathcal{O}(q_R)$. ♦

To simplify the explanation, we omit the padding function pad_c. In this case, we must construct a stateless simulator S which achieves unchangeable and extractable situations. The unchangeable situation can be achieved by using $\mathcal{RO}^*_{d+n+s}$. The extractable situation can be achieved by using $\mathcal{TRO}_s$. We give an example for the extractable situation. For a query $S(IV||m_1)$, S chooses the response $y_1 = y_{1,1}||y_{1,2}$ as $y_{1,1} = \mathcal{RO}_s^T(m_1)$ and $y_{1,2} = \mathcal{RO}_n(m_1)$ and returns y_1, where $|x| = n + s$ and $|m| = d$. For a query $S(y_1||m_2)$, S *extracts* m_1 by the query $\mathcal{TO}(y_{1,1})$, chooses the response y_2 as $y_{2,1} = \mathcal{RO}_s^T(m_1||m_2)$ and $y_{2,2} = \mathcal{RO}_n(m_1||m_2)$, and returns y_2. In this case, $\mathrm{chopMD}^S(m_1||m_2) = \mathcal{RO}_n(m_1||m_2)$. This case is a two message block case, while we can ensure the consistency for other message block cases. Therefore, for the ChopMDHF, we can construct a stateless simulator which achieves the unchangeable and extractable situations, and we can prove that the ChopMDHF is SL-indifferentiable from a $\mathcal{VO}$. The proof for the ChopMDHF is given in Appendix C.

4.2 SL-Indifferentiability for FOLSponge

We define the parameter of $\mathcal{VO}$ as $w = c$ and $b = d$. We don't care the key length a, since $\mathsf{IC}_{a,b}$ can be regarded as permutation P by fixing a 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 3. *There exists a stateless simulator $S = (S_F, S_I)$ such that for any distinguisher A,*

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

where 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 $\mathcal{O}(q)$. ♦

Similar to the ChopMDHF case, for the FOLSpongeHF, we can construct a stateless simulator S which achieves the unchangeable and extractable situations. The unchangeable situation can be achieved by using $(\mathcal{P}, \mathcal{P}^{-1})$. The extractable situation can be achieved by using $\mathcal{TRO}_c$. S can extract a message from $\mathcal{TO}$ when the right c-bit values of responses of queries to S_F are defined by $\mathcal{RO}_c^T$. Namely, S_F uses a message, which will be extracted, as an input of $\mathcal{RO}_c^T$. The proof for the FOLSpongeHF is given in Appendix D.

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

In this section, we show cryptosystems satisfying multi-stage security in the $\mathcal{VO}$ model. Specifically, we show that for any PKE scheme, the non-adaptive CDA 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 construction of IBE scheme which satisfies the non-adaptive ID-based CDA (ID-CDA) security in the $\mathcal{VO}$ model. The previous work [23] showed the non-adaptive CDA security for PKE schemes based on the same assumption (IND-SIM) only with the specific NMAC hash function. That work does not mention about CDA secure 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 was shown that EwH [1] and $\mathsf{REwH1}$ [2] satisfy IND-SIM security [23]; thus, any CPA secure PKE scheme can be converted into IND-SIM secure scheme. For IBE, we show a generic construction 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 generically converted into CDA secure PKE and ID-CDA secure IBE in the $\mathcal{VO}$ model.

5.1 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 [23]). Fig. 3 illustrates the non-adaptive CDA game for a PKE scheme $\mathcal{AE}$ using a functionality F. This notion

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

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

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

Fig. 3. CDA game and IND-SIM game

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

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

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

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. 3 shows the IND-SIM game. We define the IND-SIM advantage of an adversary $\mathcal{B}$ by

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

As noted in [23], 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 $\mathcal{VO}$ 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*

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

$\mathcal{B}$ 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 $\mathsf{maxpk}_{\mathcal{AE}} = \max_{\gamma \in \{0,1\}^*} \Pr[pk = \gamma : (pk, sk) \xleftarrow{\$} \mathsf{Gen}]$. ♦

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 F.

5.2 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 [8–10] 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. 4 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. (1) The advantage for an ID-CPA adversary $\mathcal{B}$ against scheme $\mathcal{IBE}$ using a functionality F and (2) the advantage for an ID-CDA adversary $(\mathcal{A}_1, \mathcal{A}_2)$ against scheme $\mathcal{IBE}$ using a functionality F are defined by

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

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

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 $\mathsf{REwH1}$ [2].

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

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

$\underline{\text{ID-CPA}^{\mathcal{B}}_{\mathcal{IBE},F}}$
$\beta \xleftarrow{\$} \{0,1\}$
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 $(\beta = \beta')$

$\underline{\text{ID-CDA}^{\mathcal{A}_1,\mathcal{A}_2}_{\mathcal{IBE},F}}$
$\beta \xleftarrow{\$} \{0,1\}$
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. 4. ID-CPA and ID-CDA game

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

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 5. *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*

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

$\mathcal{C}$ 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 5 is shown in Appendix H.

References

1. 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.

2. 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.
3. 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.
4. 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.
5. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the Indifferentiability of the Sponge Construction. In *EUROCRYPT*, pages 181–197, 2008.
6. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak SHA-3 submission. Submission to NIST (Round 3). 2011.
7. 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.
8. Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO 2001*, pages 213–229, 2001.
9. Ran Canetti, Shai Halevi, and Jonathan Katz. A Forward-Secure Public-Key Encryption Scheme. In *EUROCRYPT 2003*, pages 255–271, 2003.
10. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *EUROCRYPT 2004*, pages 207–222, 2004.
11. 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.
12. Donghoon Chang and Mridul Nandi. Improved Indifferentiability Security Analysis of chopMD Hash Functionl. In *FSE*, pages pages 429–443, 2008.
13. 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.
14. Gregory Demay, Peter Gazi, Martin Hirt, and Ueli Maurer. Resource-restricted indifferentiability. In *ePrint 2012/613*, 2012.
15. 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.
16. Benjamin Fuller, Adam O'Neill, and Leonid Reyzin. A Unified Approach to Deterministic Encryption: New Constructions and a Connection to Computational Entropy. In *TCC2012*, pages 582–599, 2012.
17. 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.
18. Atul Luykx, Elena Andreeva, Bart Mennink, and Bart Preneel. Impossibility results for indifferentiability with resets. In *ePrint 2012/644*, 2012.
19. 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.
20. Ilya Mironov, Omkant Pandey, Omer Reingold, and Gil Segev. Incremental Deterministic Public-Key Encryption, (Full Version in ePrint 2012/047). In *EUROCRYPT*, 2012.
21. 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.
22. National Institute of Standards and Technoloty. FIPS PUB 180-4 Secure Hash Standard. In *FIPS PUB*, 2011.
23. 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.
24. 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`.

A Graph Representations for FOLSponge and Merkle-Damgård

Sponge Graph. The proof of the indifferentiability from a RO of the SpongeHF is in the random permutation model [5], where P is a random permutation and P^{-1} be an inverse oracle. The proof of the SL-indifferentiability of the SpongeHF is thus in the random permutation model. In the real world, the left oracle $L = Sponge^P$, and the right oracles $(R_F, R_I) = (P, P^{-1})$ (F means "forward" and I

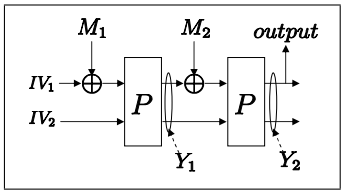

Fig. 5. Figure of Sponge

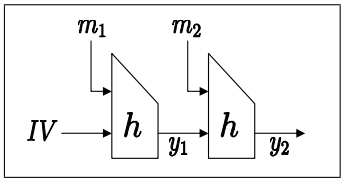

Fig. 6. Figure of Merkle-Damgård

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

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

$\underline{E(k,x)}$
1 if $\mathsf{E}[k,x] = \perp$, $y \xleftarrow{\$} \{0,1\}^b \backslash T^+[k]$;
2 $Update(k,x,y)$;
3 **return** $\mathsf{E}[k,x]$;

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

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

$\underline{D(y)}$
1 if $\mathsf{D}[k,y] = \perp$, $x \xleftarrow{\$} \{0,1\}^b \backslash T^-[k]$;
2 $Update(k,x,y)$;
3 **return** $\mathsf{D}[k,y]$;

Fig. 7. Versatile Oracle $\mathcal{VO}$

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. 5 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 ROs of the ChopMDHF and the PFMD hash function (PFMDHF) [11–13], the compression function h is a RO. The proofs of the SL-indifferentiability of the ChopMDHF and the FPMDHF 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. 6 may help to understand the path).

This graph is used in the proofs of the ChopMDHF (Theorem 2) and the PFMDHF (Theorem 6). 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".

B Implementation of $\mathcal{VO}$

In this Appendix, we give an implementation method of $\mathcal{VO} = (\mathcal{RO}_n, \mathcal{RO}_v^*, \mathcal{TRO}_w, \mathsf{IC}_{a,b})$. $\mathcal{VO}$ can be implemented as Fig. 7.

$\underline{S(x||m)}$ where $x = x_1||x_2$ ($|x_1| = s$, $|x_2| = n$) and $|m| = d$
$M \leftarrow \mathcal{TO}(x_1)$;
if $x = IV$ then
    $z \leftarrow \mathcal{RO}_n(m)$;
    $w \leftarrow \mathcal{RO}_s^T(m)$;
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. 8. Simulator S

$\mathcal{RO}_n$ is shown in Fig. 7 (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. 7 (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. 7 (Center) which consists of a RO $\mathcal{RO}_w^T$ and a Trace Oracle $\mathcal{TO}$. The input length of $\mathcal{RO}_w^T$ is arbitrary. The output length of $\mathcal{RO}_w^T$ 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. 7 (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, $\mathsf{E}[k,x] \leftarrow y, \mathsf{D}[k,y] \leftarrow x, T^+[k] \xleftarrow{\cup} \{y\}$ and $T^-[k] \xleftarrow{\cup} \{x\}$. Note that the a, b, are defined in each hash function.

C Proof of Theorem 2

In this proof, we use the MD graph described in Appendix A. And we use the implementation method defined in Appendix B of a $\mathcal{VO}$.

Simulator S. We define a stateless simulator S in Fig. 8. In this proof, the padding function pad_c is removed. Thus the left queries are in $(\{0,1\}^d)^*$. Note that the ChopMDHF with the padding function is the special case of one without the padding function. Thus the security of the ChopMDHF without the padding function ensures the security of one with the padding function. 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 defined in Fig. 8 is consistent with $\mathcal{RO}_n$. We give an example that S is consistent with $\mathcal{RO}_n$. For 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 $w_1 = \mathcal{RO}_s^T(M_1)$ (due to the step 4), and $w_2 = \mathcal{RO}_s^T(M_1||M_2)$ (due to 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 query $S(w_2||z_2||M_3)$, the response $w_3||z_3$ is defined such that $z_3 = \mathcal{RO}_n(M_1||M_2||M_3)$ (due to the step 6). Notice that $M_1||M_2$ can be obtained by the queries $\mathcal{TO}(w_2)$ (due to the step 1). So in this case S is consistent with $\mathcal{RO}_n$, that is, the path $IV \xrightarrow{M_1||M_2||M_3} w_3||z_3$ is such that $z_3 = \mathcal{RO}_n(M_1||M_2||M_3)$. Though this case is a three message block case, S ensures the consistency for other message block cases.

Detail. To evaluate the SL-indifferentiable advantage, we consider five games. In each game, distinguisher A has oracle access to left oracle L and right oracle R.

- Game 1 is the ideal world, that is, $(L, R) = (\mathcal{RO}_n, S)$.
- Game 2 is $(L, R) = (\mathcal{RO}_n, S_1)$, where S_1 keeps all query-response pairs. For a 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 3 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 $\mathrm{chopMD}^{S_1}(M)$ and returns the output of $\mathcal{RO}_n(M)$.
- Game 4 is $(L, R) = (\mathrm{chopMD}^{S_1}, S_1)$.
- Game 5 is the real world, that is, $(L, R) = (\mathrm{chopMD}^h, h)$.

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

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

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

Game 1 $\Rightarrow$ Game 2. In Game 2, use of the query-response history ensures that for a repeated query $R(x||m)$ the same value is responded, while in Game 1 there is a case that for some repeated query $R(x||m)$ where y was responded, a different value y^* ($\neq y$) is responded due to the definition of $\mathcal{TO}$. The difference $|\Pr[G_1] - \Pr[G_2]|$ is thus bounded by the probability that in Game 1 a different value is responded. We call the event "**Diff**". Procedures of S are controlled by $\mathcal{TO}$ (See the steps 2, 5, and 8). Therefore, if **Diff** occurs, some output of $\mathcal{TO}(x_1)$ is changed. More precisely, 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. There are two cases for (w, w^*).
  - $\mathbf{Diff}_1$: $w = \perp$ and $w^* \neq \perp$.
  - $\mathbf{Diff}_2$: $w \neq \perp$ and $w^* = \perp$.

We thus have that

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

We justify the bound as follows.

Consider $\mathbf{Diff}_1$ which occurs when the following case occurs. When query $\mathcal{TO}(y)$ was made where the response was w ($=\perp$), a query $\mathcal{RO}^T_s(w^*)$ such that $y = \mathcal{RO}^T_s(w^*)$ had not been made. When the repeated query $\mathcal{TO}(y)$ is made where the response is w^*, the query $\mathcal{RO}^T_s(w^*)$ such that $y = \mathcal{RO}^T_s(w^*)$ was already made. Therefore, first y is defined. Second, the output of $\mathcal{RO}^T_s(w^*)$ is defined. Thus, $\Pr[\mathbf{Diff}_1]$ is bounded by the probability that an output of $\mathcal{RO}^T_s(w^*)$ (s-bit random value) hits a value y. Since the numbers of queries to $\mathcal{RO}^T_s$ and $\mathcal{TO}$ are at most q_R times,

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

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

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

Game 2 ⇒ Game 3. The difference between Game 2 and Game 3 is that for a left query $L(M)$, in Game 2 L does not make a right query, while in Game 3 L makes additional right queries corresponding with $\mathrm{chopMD}^{S_1}(M)$. Note that 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 A obtains don't affect the A's behavior. We show Lemma 1 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 $(x_t[1,s], y_t[1,s])$ for $t = 1, \ldots, i-1$ where $(x_t||m_t, y_t)$ is a t-th R query-response ($y_t = S_1(x_t||m_t)$).

- Bad_j is that in Game j for some i-th query $S_1(x_i||m_i)$ the response y_i is such that $y_i[1,s]$ collides with some value in $T_i \cup \{x_i[1,s]\} \cup \{IV[1,s]\}$.

Lemma 1 is also used in the evaluation between Game 3 and Game 4. Therefore, Lemma 1 ensures that unless the bad event occurs, in both games, responses of right queries which are leafs of MD paths[3] are defined by the same queries to $\mathcal{RO}_s^T$ and $\mathcal{RO}_n$. Namely, in Game 3, unless the bad event occurs, the responses of the additional right queries which A obtains are chosen from the same distribution as in Game 2. Thus, the difference $|\Pr[G_2] - \Pr[G_3]|$ is bounded by the probability of occurring the bad event. Precisely,

$$\begin{aligned}
|\Pr[G_2] - \Pr[G_3]| \leq & |\Pr[G_2|Bad_2]\Pr[Bad_2] + \Pr[G_2|\neg Bad_2]\Pr[\neg Bad_2] \\
& - (\Pr[G_3|Bad_3]\Pr[Bad_3] + \Pr[G_3|\neg Bad_3]\Pr[\neg Bad_3])| \\
\leq & |\Pr[G_2|\neg Bad_2](\Pr[Bad_3] - \Pr[Bad_2]) \\
& + (\Pr[G_2|Bad_2]\Pr[Bad_2] - \Pr[G_3|Bad_3]\Pr[Bad_3])| \\
\leq & \max\{\Pr[Bad_2], \Pr[Bad_3]\} \leq \frac{\sigma(\sigma+1)}{2^s}
\end{aligned}$$

where $\Pr[G2|\neg Bad_2] = \Pr[G3|\neg Bad_3]$ from Lemma 1. We justify the bound later.

Lemma 1. *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 1. 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 $(x_1||m_1, y_1), \ldots, (x_t||m_t, y_t)$ be query-response pairs of S_1 which correspond with the MD path where $x_1 = IV$, $x_i = y_{i-1}$ $(i = 2, \ldots, t)$, $y_t = y$, and $M = m_1|| \ldots ||m_t$.

When $t = 1$, $y[s+1, n+s] = \mathcal{RO}_n(M)$ (see the steps 2-4).

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

Since Bad_j does not occur, the case that for some $i \in \{1, \ldots, t-1\}$, $(x_i||m_i, y_i)$ is defined after $(x_{i+1}||m_{i+1}, y_{i+1})$ was defined does not occur. So $(x_1||m_1, y_1), \ldots, (x_t||m_t, y_t)$ are defined by this order. Therefore, when the query $S_1(x_t||m_t)$ is made, the pair $(m_1|| \ldots ||m_{j-1}, y_{t-1})$ is already recorded in F^T, that is, $\mathsf{F}^T[m_1|| \ldots ||m_{t-1}] = y_{t-1} = x_t$.

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

Thus, for the query $S(x_t||m_t)$, $\mathcal{TO}(x_t)$ returns $m_1|| \ldots ||m_{t-1}$ (the step 1) and then the response y_t is defined such that $y_t[s+1, n+s] = \mathcal{RO}_n(M)$ (the step 6).

□

Evaluations of $\Pr[Bad_2], \Pr[Bad_3]$. In Game 2 and Game 3 S_1 is called at most q_R and σ times, respectively. The left s-bit values of all outputs of S_1 are uniformly chosen at random from $\{0,1\}^s$. We

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

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

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

thus have that

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

Game 3 ⇒ Game 4. The difference between Game 3 and Game 4 is the left oracle L where in Game 3 $L(M)$ returns $\mathcal{RO}_n(M)$, while in Game 4 $L(M)$ returns $\mathrm{chopMD}^{S_1}(M)$. Thus, the difference does not change behavior of A iff in Game 4 for any query $L(M)$, $L(M)$ returns $\mathcal{RO}_n(M)$. From Lemma 1, for any MD path $IV \xrightarrow{M} z$, $z[s+1, s+n] = \mathcal{RO}_n(M)$ unless the bad event Bad_4 occurs. Since S_1 is called at most σ times, we have that

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

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

D Proof of Theorem 3

In this proof, we use the Sponge graph described in Appendix A. And we use the implementation method defined in Appendix B of a $\mathcal{VO}$.

Simulator S**.** We define a stateless simulator S in Fig. 9. 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 are in $(\{0,1\}^n)^*$. Note that the FOLSpongeHF with the padding function is the special case of one without the padding function. Thus the security of the FOLSpongeHF 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 $\mathcal{RO}_n$. We give an example that S is consistent with $\mathcal{RO}_n$. For 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). Though this case is three message block case, S ensures cases for other message blocks.

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

Fig. 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.

Detail. The proof is given as follows. To evaluate the SL-indifferentiable advantage, we consider six 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, $(L, R_F, R_I) = (\mathcal{RO}_n, S_F, S_I)$.
- Game 2 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 3 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 4 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 5 is $(L, R_F, R_I) = (Sponge^{S1_F}, S1_F, S1_I)$.
- Game 6 is the real world, that is, $(L, R_F, R_I) = (Sponge^P, P, P^{-1})$.

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

$$\mathsf{Adv}^{\mathsf{sl\text{-}indiff},\mathcal{VO}}_{Sponge^P,S}(\mathcal{A}) \leq \sum_{i=1}^{5} |\Pr[G_i] - \Pr[G_{i+1}]| \leq \frac{2\sigma(\sigma+1) + q(q-1)}{2^c} + \frac{\sigma(\sigma-1) + q(q-1)}{2^{d+1}}.$$

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

Game 1 $\Rightarrow$ Game 2. In Game 1, a random permutation $\mathcal{P}$ and its inverse $\mathcal{P}^{-1}$ are uses, while in Game 2, $\mathcal{P}_1$ and $\mathcal{P}_1^{-1}$ are used where the outputs are uniformly chosen at random from $\{0,1\}^d$. Thus $|\Pr[G_1] - \Pr[G_2]|$ 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,

$$|\Pr[G_1] - \Pr[G_2]| \leq \sum_{i=1}^{q} \frac{i-1}{2^d} = \frac{q(q-1)}{2^{d+1}}.$$

Game 2 $\Rightarrow$ Game 3. In Game 3, use of the query-response 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 2 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, a 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**". Procedures of S are controlled by $\mathcal{TO}$ (See the steps 2, 4, and 7 of S_F and the steps 2, 4, and 7 of S_I). Therefore, if **Diff** occurs, some output of $\mathcal{TO}(y)$ is changed. More precisely, 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. There are two cases for (w, w^*).
  - **Diff**$_1$: $w = \perp$ and $w^* \neq \perp$.

- **Diff**$_2$: $w \neq \perp$ and $w^* = \perp$.

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 **Diff**$_1$. When a query $\mathcal{TO}(y)$ was made, the response was $w = \perp$ and a query $\mathcal{RO}_c^T(w^*)$ such that $y = \mathcal{RO}_c^T(w^*)$ had not been made. And when the repeated query $\mathcal{TO}(y)$ is made, the response is w^* and 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 an output of $\mathcal{RO}_c^T(w^*)$ (c-bit random value) hits a 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 **Diff**$_2$. From the definition of $\mathcal{TO}$, if **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 3 $\Rightarrow$ 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 $Sponge^{S1_F}$. Note that 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 A obtains don't affect the A's behavior. We show Lemma 2 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 2 is also used in the evaluation between Game 4 and Game 5. Lemma 2 ensures that in Game 4, unless Bad_j occurs, responses which are leafs of Sponge paths[4] are defined by the same queries to $\mathcal{RO}_c^T$ and $\mathcal{RO}_n$ as in Game 3. Namely, unless the bad event occurs, the responses of the additional right queries don't affect the 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

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

where $\Pr[G_3|\neg Bad_3] = \Pr[G_4|\neg Bad_4]$ from Lemma 2. We justify the bound later.

Lemma 2. *In Game j, unless Bad_j occurs, for any Sponge path $IV \xrightarrow{M} z$ $z[1,n] = \mathcal{RO}_n(M)$.* ♦

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

Proof of Lemma 2. 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 $t = 1$. Since Bad_j does not occur, 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. Therefore $z[1, n] = \mathcal{RO}_n(M)$ due to the steps 2,3.

Consider the case that $t \geq 2$.

Since Bad_j does not occur, no pair which is defined by a R_I query is connected with another pair and also no pair which is defined by a R_F query is connected with another pair. Namely, $(X_1, Y_1), \ldots, (X_t, Y_t)$ are defined by R_F queries and are defined by the ordered R_F queries $S1_F(X_1), \ldots, S1_F(X_t)$. Therefore, the structure of S_F ensures that when the query $R_F(X_t)$ is made, the pair $(M_1 || \cdots || M_{t-1}, Y_{t-1}[n+1, d])$ is stored in the table F^T, that is, $\mathsf{F}^T[M_1 || \cdots || M_{t-1}] = Y_{t-1}[n+1, d]$.

Since no collision for $\mathcal{RO}_c^T$ occurs, when for the query $S1_F(X_t)$ S_F makes the query $\mathcal{TO}(X_t[n+1, d])$, $M_1 || \cdots || M_{t-1}$ is returned from $\mathcal{TO}$.

From above discussions, when the query $S1_F(X_t)$ is made, $S1_F$ returns the output of $\mathcal{RO}_n(M)$ due to the structure of S_F. Thus $Y_t[1, n] = \mathcal{RO}_n(M)$.

□

Evaluations of $\Pr[Bad_3], \Pr[Bad_4]$**.** Since in Game 3 and Game 4 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 $\{0, 1\}^c$,

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

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 $Sponge^{S_1}(M)$. 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 Sponge path $IV \xrightarrow{M} z$ the relation $z[1, n] = \mathcal{RO}_n(M)$ holds unless the bad event Bad_5 occurs. In Game 5 R is called at most σ times and for any query to S the response is chosen uniformly at random from $\{0, 1\}^c$. We have that

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

Game 5 ⇒ Game 6. In Game 5, outputs of R_F and R_I are chosen uniformly at random from $\{0, 1\}^d$, while in Game 6, 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 5. We thus have that

$$|\Pr[G_5] - \Pr[G_6]| \leq \sum_{i=1}^{\sigma} \frac{i-1}{2^d} = \frac{\sigma(\sigma-1)}{2^{d+1}}.$$

□

E SL-Indifferentiability for PFMD

The PFMD construction is shown in Fig. 11. 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.

$\underline{\mathrm{PFMD}^h(M)}$
1 $(M_1, \ldots, M_i) \leftarrow div(d, \mathsf{pfpad}(M))$
2 $x \leftarrow IV$;
3 For $j = 1, \ldots, i$, $x \leftarrow h(x||M_j)$;
4 **return** x;

Fig. 11. Prefix-free Merkle-Damgård

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

E.1 SL-Indifferentiability for PFMD

We define the parameter of $\mathcal{VO}$ as $v = n$ and $w = n$. Note that in the SL-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 6. *There exists a stateless simulator S such that for any distinguisher $\mathcal{A}$, the following holds,*

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

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 $\mathcal{O}(q_R)$. ♦

Remark 1. EMD [4] and MDP [17] 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 SL-indifferentiable from $\mathcal{VO}$s.

E.2 Proof

In this proof, we use the MD graph described in Appendix A. And we use the implementation method defined in Appendix B of a $\mathcal{VO}$.

Simulator S**.** We define a stateless simulator S in Fig. 12. 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 defined in Fig. 12 is consistent with $\mathcal{RO}_n$. We give an example that S is consistent with $\mathcal{RO}_n$. For 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 $\mathsf{pfpad}(M) = m_1||m_2||m_3$. So, in this case, the simulator S succeeds in the simulation of h. Though this case is three message block case, S ensures cases for other message blocks.

Detail. To evaluate the SL-indifferentiable advantage, we consider five games. In each game, distinguisher A has oracle access to left oracle L and right oracle R.

- Game 1 is the ideal world, that is, $(L, R) = (\mathcal{RO}_n, S)$.
- Game 2 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 3 is $(L, R) = (L_1, S_1)$, where on query $L_1(M)$ L_1 first makes queries to S_1 which correspond with $\mathrm{PFMD}^{S_1}(M)$ then returns $\mathcal{RO}_n(M)$.
- Game 4 is $(L, R) = (\mathrm{PFMD}^{S_1}, S_1)$.
- Game 5 is the real world, that is, $(L, R) = (\mathrm{PFMD}^h)$.

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

$$\mathsf{Adv}^{\mathsf{sl\text{-}indiff},\mathcal{VO}}_{\mathrm{PFMD}^h,S}(\mathcal{A}) \leq \sum_{i=1}^{4} |\Pr[G_i] - \Pr[G_{i+1}]| \leq \frac{2\sigma(\sigma+1) + q_R(q_R-1)}{2^n}.$$

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

Game 1 $\Rightarrow$ Game 2. In Game 2, use of the history ensures that for any repeated query $R(x, m)$ the same value y is responded, while in Game 1 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 $|\Pr[G_1] - \Pr[G_2]|$ is thus bounded by the probability that in Game 1 a different value is responded. We call the event "**Diff**". Procedures of S are controlled by $\mathcal{TO}$ (see the steps 2,5,8). Therefore, if **Diff** occurs, some output of $\mathcal{TO}(y)$ is changed. More precisely, if **Diff** occurs, the following event occurs.

- For a repeated query $\mathcal{TO}(y)$ where w was responded, a different value w^* is responded. There are two cases for (w, w^*).
  - $\mathbf{Diff}_1$: $w = \perp$ and $w^* \neq \perp$.
  - $\mathbf{Diff}_2$: $w \neq \perp$ and $w^* = \perp$.

We thus have that

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

We justify the bound as follows.

Consider $\mathbf{Diff}_1$. When a query $\mathcal{TO}(y)$ was made where the response was w ($=\perp$), a query $\mathcal{RO}^T_n(w^*)$ such that $y = \mathcal{RO}^T_n(w^*)$ had not been made. And when the repeated query $\mathcal{TO}(y)$ is made where the response is w^*, the query $\mathcal{RO}^T_n(w^*)$ such that $y = \mathcal{RO}^T_n(w^*)$ was already made. Thus $\Pr[\mathbf{Diff}_1]$ is bounded by the probability that an output of $\mathcal{RO}^T_n(w^*)$ (n-bit random value) hits a value y. Since the numbers of queries to $\mathcal{RO}^T_n$ 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 $\mathcal{RO}^T_n$ 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 2 ⇒ Game 3. The difference between Game 2 and Game 3 is that in Game 2 L does not make a right query, while in Game 3 L makes additional right queries corresponding with $\mathrm{PFMD}^{S_1}(M)$. Note that 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 A obtains don't affect the 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 $S_1(x_i||m_i)$ 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[5] are defined by the same query to $\mathcal{RO}_n^T$ and $\mathcal{RO}_n$. Namely, in Game 3, unless the bad event occurs, the responses of the additional right queries which A obtains are chosen from the same distribution as in Game 2. Thus, the difference $|\Pr[G_2] - \Pr[G_3]|$ is bounded by the probability of occurring the bad event. We thus have that

$$|\Pr[G_2] - \Pr[G_3]| \leq \max\{\Pr[Bad_2], \Pr[Bad_3]\} \leq \frac{\sigma(\sigma+1)}{2^n}$$

where $\Pr[G2|\neg Bad_2] = \Pr[G3|\neg Bad_3]$ from Lemma 3. We justify the bound later.

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 $(x_1||m_1, y_1), \ldots, (x_t||m_t, y_t)$ be query-response pairs of S which correspond with the PFMD path where $x_1 = IV$, $x_i = y_{i-1}$ $(i = 2, \ldots, t)$, $y_j = y$, 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 $j \geq 2$.

Since Bad_j does not occur, no pair $(x_i||m_i, y_i)$ is defined after $(x_{i+1}||m_{i+1}, y_{i+1})$ was defined. Therefore, $(x_1||m_1, y_1), \ldots, (x_t||m_t, y_t)$ are defined by this order. The structure of S ensures that when a query $S_1(x_t||m_t)$ is made, the pair $(m_1|| \cdots ||mt-1, y_{t-1})$ is already stored in F^T, that is, $\mathsf{F}^T[m_1|| \cdots ||m_{t-1}] = y_{t-1}$.

Since Bad_j does not occur, no collision for $\mathcal{RO}_n^T$ occurs. Therefore, when a query $\mathcal{TO}(x_t)$ is made for the query $S_1(x_t||m_t)$, $\mathcal{TO}$ returns $m_1|| \cdots ||m_{t-1}$.

From the above discussions, for the query $S_1(x_t||m_t)$, $\mathcal{TO}$ responses $m_1|| \ldots ||m_{t-1}$ (the step 1) and then the response y_t is defined such that $y_i = \mathcal{RO}_n(M)$ (the step 3). □

Evaluation of $\Pr[Bad_2], \Pr[Bad_3]$. Since in Game 2 and Game 3 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,

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

Game 3 ⇒ Game 4. The difference between Game 3 and Game 4 is the left oracle L where in Game

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

3 $L(M)$ returns $\mathcal{RO}_n(M)$, while in Game 4 $L(M)$ returns $\mathrm{PFMD}^{S_1}(M)$. Thus, the difference does not change behavior of A iff in Game 4 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_4, where $M^* = \mathsf{pfpad}(M)$. Since in Game 4 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

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

Game 4 ⇒ Game 5. Since outputs of S are uniformly chosen at random from $\{0,1\}^n$, the difference for R does not affect the A's behavior. We thus have that $\Pr[G_4] = \Pr[G_5]$.

□

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

Proof. The difference between $\mathbf{G}_0$ and $\mathbf{G}_1$ only occurs in Bad_1 and Bad_2. From Difference Lemma [24], 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 $\mathsf{SimB} = (\mathsf{SimB}_{\mathcal{RO}_n}, \mathsf{SimB}_{\mathcal{RO}^*_v}, \mathsf{SimB}_{\mathcal{RO}^T_w}, \mathsf{SimB}_{\mathcal{TO}}, \mathsf{SimB}_E, \mathsf{SimB}_D)$

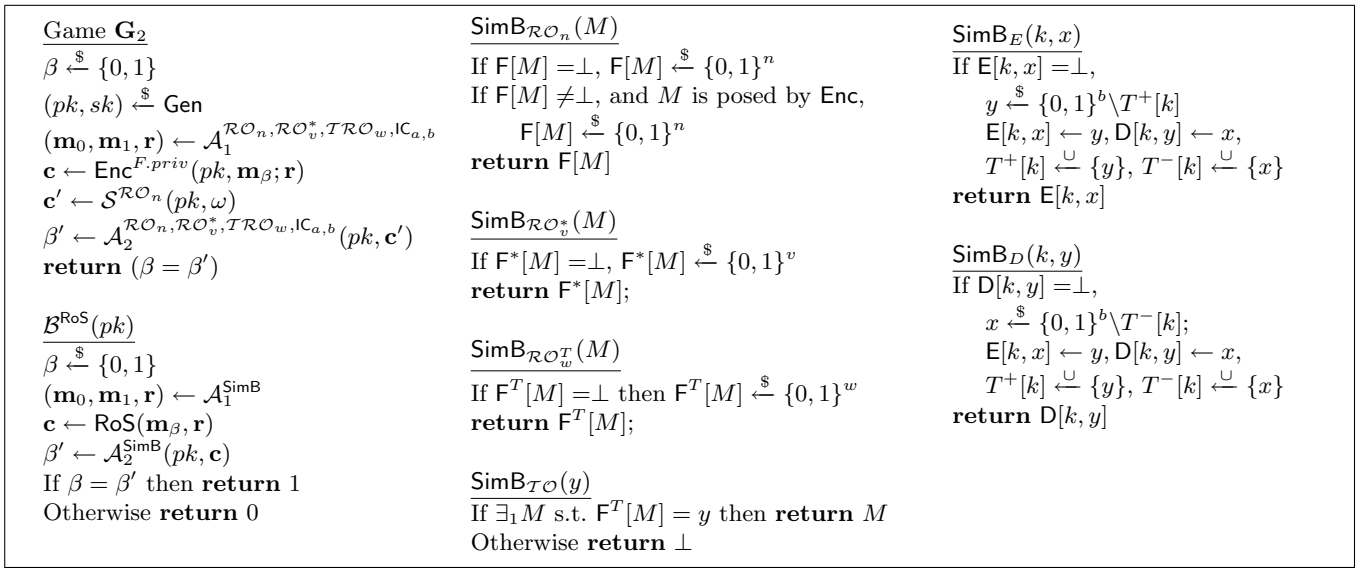

Fig. 13. game $\mathbf{G}_2$ and simulation SimB by adversary $\mathcal{B}$

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 $\mathcal{RO}_v^*, \mathcal{TRO}_w = (\mathcal{RO}_w^T, \mathcal{TO})$ 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. $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_2)| \leq \frac{4q_{\mathcal{RO}^*}^2 + 4q_{\mathcal{RO}^T}^2}{2^\mu} + max\left\{\frac{4q_{\mathcal{TO}}^2}{2^\mu}, \frac{4q_{\mathcal{TO}}^2}{2^w}\right\} + max\left\{\frac{4q_E^2 + 4q_D^2}{2^\mu}, \frac{4q_E^2 + 4q_D^2}{2^b}\right\}$.

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 $\mathcal{RO}_v^*, \mathcal{TRO}_w = (\mathcal{RO}_w^T, \mathcal{TO})$ 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 [24], 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, or the output length w of $\mathcal{RO}_T$ and the output length b of (E, D). According to the birthday paradox, for oracles $\mathcal{RO}^*$ and $\mathcal{RO}^T$ the probability of collisions in guessing is at most $(2q_{\mathcal{RO}^*})^2/2^\mu$, and $(2q_{\mathcal{RO}^T})^2/2^\mu$, respectively. Also, for oracle $\mathcal{TO}$ the probability of collision in guessing is at most $(2q_{\mathcal{TO}})^2/2^\mu$ if $\mu < w$, $(2q_{\mathcal{TO}})^2/2^w$ otherwise, and for oracles E and D the probability of collisions in guessing is at most $(2q_E)^2/2^\mu$ and $(2q_D)^2/2^\mu$ if $\mu < b$, $(2q_E)^2/2^b$ and $(2q_D)^2/2^b$ otherwise. Therefore, $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_2)| \leq (4q_{\mathcal{RO}^*}^2 + 4q_{\mathcal{RO}^T}^2)/2^\mu + max\left\{4q_{\mathcal{TO}}^2/2^\mu, 4q_{\mathcal{TO}}^2/2^w\right\} + max\left\{(4q_E^2 + 4q_D^2)/2^\mu, (4q_E^2 + 4q_D^2)/2^b\right\}$.

□

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

□

G ID-CPA Security of IDREwH1 in the $\mathcal{VO}$ Model

We can prove the ID-CPA security of IDREwH1; that is, we show that 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 7. *Let $\mathcal{IBE}_r$ be an IBE scheme. Let $\mathcal{B}$ be a selective (resp. full) CPA adversary for* 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 IBE.Enc algorithm never use $\mathcal{RO}^*_v, \mathcal{RO}^T_w, \mathcal{TO}, E, D$. Thus, $\mathcal{B}$ can be constructed with $\mathcal{C}$.

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 $m_\beta \in \mathbf{m}_\beta$, $r \in \mathbf{r}$, and $\mathbf{r}$ is the randomness vector used to generate challenge ciphertext $\mathbf{c}$. We denote this event as Bad. From Difference Lemma [24], 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.

$\underline{\mathsf{SimC}_{\mathrm{main}}}$
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 β' from $\mathcal{B}$
return β'

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

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

H Proof of Theorem 5

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 $\mathbf{G}_0$.

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 [24], 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 $\Pr[\mathsf{Bad}_2] \leq \frac{q_{\mathcal{RO}}}{2^\mu}$. □

$\underline{\mathsf{SimC}'_{\text{main}}}$
$\beta'' \xleftarrow{\$} \{0,1\}$
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 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 c from $\mathcal{CH}$
send $(params, \mathbf{c}, id^*)$ to $\mathcal{A}_2$
receive β' from $\mathcal{A}_2$
return 0 if $\beta' = \beta''$ and 1 otherwise

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

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 $\mathsf{SimC}' = (\mathsf{SimC}'_{\text{main}}, \mathsf{SimC}'_{\mathsf{IBE.Gen}}, \mathsf{SimC}'_{\mathcal{RO}}$ $\mathsf{SimC}'_{\mathcal{RO}^*}, \mathsf{SimC}'_{\mathcal{RO}^T}, \mathsf{SimC}'_{\mathcal{TO}}, \mathsf{SimC}'_E, \mathsf{SimC}'_D)$ 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 $\mathbf{G}_2$.

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

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 [24], 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, or the output length w of $\mathcal{RO}_T$ and the output length b of (E, D). According to the birthday paradox, for oracles $\mathcal{RO}^*$ and $\mathcal{RO}^T$ the probability of collisions in guessing is at most $(2q_{\mathcal{RO}^*})^2/2^\mu$, and $(2q_{\mathcal{RO}^T})^2/2^\mu$, respectively. Also, for oracle $\mathcal{TO}$ the probability of collision in guessing is at most $(2q_{\mathcal{TO}})^2/2^\mu$ if $\mu < w$, $(2q_{\mathcal{TO}})^2/2^w$ otherwise, and for oracles E and D the probability of collisions in guessing is at most $(2q_E)^2/2^\mu$ and $(2q_D)^2/2^\mu$ if $\mu < b$, $(2q_E)^2/2^b$ and $(2q_D)^2/2^b$ otherwise. Therefore, $|\mathsf{Adv}(\mathcal{A}, \mathbf{G}_3) - \mathsf{Adv}(\mathcal{A}, \mathbf{G}_2)| \leq (4q^2_{\mathcal{RO}^*} + 4q^2_{\mathcal{RO}^T})/2^\mu + max\left\{4q^2_{\mathcal{TO}}/2^\mu, 4q^2_{\mathcal{TO}}/2^w\right\} + max\left\{(4q^2_E + 4q^2_D)/2^\mu, (4q^2_E + 4q^2_D)/2^b\right\}$.

□

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