On Indifferentiable Hash Functions in Multi-Stage Security Games

Yusuke Naito

Mitsubishi Electric Corporation

Abstract. It had been widely believed that the indifferentiability framework ensures composition in any security game. However, Ristenpart, Shacham, and Shrimpton (EUROCRYPT 2011) demonstrated that for some multi-stage security, there exists a cryptosystem which is secure in the random oracle (RO) model but is broken when some indifferentiable hash function is used. However, this does not imply that for any multi-stage security, any cryptosystem is broken when a RO is replaced with the indifferentiable hash function. They showed that the important multi-stage security: the chosen-distribution attack (CDA) security is preserved for some public key encryption (PKE) schemes when a RO is replaced with the indifferentiable hash function proposed by Dodis, Ristenpart, and Shrimpton (EUROCRYPT 2009). An open problem from their result is the multi-stage security when a RO is replaced with other indifferentiable hash functions. We show that for *any* PKE scheme, the CDA security, which is a multi-stage security, is preserved when a RO is replaced with the important indifferentiable hash function, Prefix-free Merkle-Damgård, chop Merkle-Damgård, or Sponge.

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

1 Introduction

The indifferentiable composition theorem of Maurer, Renner, and Holenstein [25] ensures that if a functionality F (e.g., a hash function from an ideal primitive) is indifferentiable from a second functionality F' (e.g., a random oracle (RO)), the security of any cryptosystem is preserved when F' is replaced with F. The important application of this framework is the RO model security, because many practical cryptosystems e.g., RSA-OAEP [9] and RSA-PSS [10] are designed by the RO methodology. A RO is instantiated by a hash function such as SHA-1 and SHA-256 [29]. However, the Merkle-Damgård hash functions [19, 26] such as SHA-1 and SHA-256, are not indifferentiable from ROs [18]. So many indifferentiable (from a RO) hash functions have been proposed, e.g., the finalists of the SHA-3 competition [3, 12, 21, 23, 31, 1, 2, 11, 13, 18, 17, 20]. The indifferentiable security is thus an important security of hash functions.

Recently, Ristenpart, Shacham, and Shrimpton [30] showed that in some multi-stage security game some scheme secure in the RO model is broken when some indifferentiable hash function is used. They considered the multi-stage security game called CRP. The CRP security game for the n-bit (output length) hash function H is the two stage security game. In the first stage, for random messages M_1, M_2 of $2n$ bits, the first stage adversary A_1 derives the some state st of $2n$ bits. In the second stage, the second stage adversary A_2 receives st, and for a random challenge value C of $2n$ bits outputs an n-bit value z. Then, the adversary wins if $z = H(M_1||M_2||C)$. Consider the chop MD hash function $\mathrm{chopMD}^h(M_1||M_2||C) = chop_n(h(h(h(IV, M_1), M_2), C))$ which is indifferentiable from a RO [18], where $h : \{0,1\}^{4n} \rightarrow \{0,1\}^{2n}$ is a RO, and $chop_n : \{0,1\}^{2n} \rightarrow \{0,1\}^n$ outputs the right n bits of the input. Clearly, the following adversary can win with probability 1 when H is the chop MD hash function. First, A_1 receives M_1, M_2, calculates $st = h(h(IV, M_1), M_2)$, and outputs st. Second, A_2 receives st, and for a random challenge C, outputs $z = chop_n(h(st, C))$ which is equal to $\mathrm{chopMD}^h(M_1||M_2||C)$. On the other hand, when H is a RO, since A_2 cannot receive several value of M_1, M_2, the probability that the adversary wins is negligible. This result implies that the indifferentiable composition theorem does not ensure any multi-stage security when a RO is replaced with indifferentiable hash functions.

The chosen-distribution attack (CDA) security game is an important multi-stage security game, which is the security goal for deterministic, efficiently searchable [4, 6, 14, 22, 27], and hedged [5] public key encryption (PKE), wherein there are several PKE schemes which are proven in the RO model [4, 5]. For the CDA secure PKE schemes EwH [4] and REwH1 [5] (in the RO model), Ristenpart *et al.* salvaged the important indifferentiable hash function, the NMAC-type hash function [20], which was proposed by Dodis, Ristenpart, and Shrimpton, and which is employed in the SHA-3 finalist Skein [21]. They showed that these PKE schemes are non-adaptive CDA secure in the chosen-plaintext attack (CPA) case when the NMAC-type hash function is used.

The open problem from the paper of Ristenpart *et al.* is thus the CDA security when a RO replaced with other indifferentiable hash functions. Especially, it is important to consider the security when a RO is replaced with the SHA-3 finalists and the SHA-2 hash functions, because one of the SHA-3 finalists will be published as a standard hash function (FIPS) [28] and the SHA-2 hash functions were published as standard hash functions [29]. We consider the important hash functions, Prefix-free Merkle-Damgård (PFMD) [18], Sponge [11] and chop Merkle-Damgård (chop MD) [18]. The PFMD hash function is employed in the SHA-3 finalist BLAKE [3]. The Sponge hash function is employed in the SHA-3 finalist Keccak [12]. The chop Merkle-Damgård hash function is employed in SHA-224 and SHA384 [29]. We show that for *any* PKE scheme, the CDA security, which is a multi-stage security, is preserved when a RO is replaced with the indifferentiable hash function. The above result covers both the adaptive security and the non-adaptive security and both both chosen-ciphertext attack (CCA) and CPA cases. The advantages of our result to the result of Ristenpart *et al.* are that (1) our result ensures the *stronger* security (adaptive and CCA), and (2) our result ensures the CDA security of *any* PKE scheme. Since several PKE schemes in [5, 4] support the CCA case or the adaptive case, the analysis for the stronger security cases is important.

(Reset) Indifferentiability [30]. To prove the CDA security, we use the reset indifferentiability framework of Ristenpart *et al.* The reset indifferentiability ensures composition in any security game: if a hash function H^P which uses an ideal primitive P is reset indifferentiable from another ideal primitive P', any security of any cryptosystem is preserved when P' is replaced with H^P.

Recall the original [25] and reset [30] indifferentiability (from a RO) frameworks. The original indifferentiable security game from a RO for H^P is that a distinguisher A converses either with (H^P, P) or (RO, S^{RO}). S is a simulator which simulates P such that S and P' are consistent. If the probability that the distinguisher A hits the conversing world is small, then H^P is indifferentiable from a RO. In the reset indifferentiable security game, the distinguisher can reset the initial state of the simulator at arbitrary times.

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

Our Approach. We thus use the reset indifferentiability from a variant of a RO. We propose a variant which covers many indifferentiable hash functions. We call the variant "Versatile Oracle" ($\mathcal{VO}$). The $\mathcal{VO}$ consists of a RO and auxiliary oracles. The auxiliary oracles are used to record the query-response history of a simulator. The $\mathcal{VO}$ thus enables to construct a simulator which does not update the internal state and which is unaffected by the reset function. We show that the PFMD hash function, the chop MD hash function [1], and the Sponge hash function are reset indifferentiable from $\mathcal{VO}$s. Then, we show that for any PKE scheme, the CDA security is preserved when a RO is replaced with a $\mathcal{VO}$. The reset indifferentiability composition theorem ensures that the CDA security are preserved when a RO is replaced with the indifferentiable hash function. Note that this is the first time positive result for the reset indifferentiability (from a $\mathcal{VO}$). We note that since $\mathcal{VO}$ is the weaker oracle than RO, $\mathcal{VO}$ cannot cover all applications of RO. However, $\mathcal{VO}$ covers the CDA security for any PKE scheme. Again, our goal is to prove the CDA security when a RO is replaced with the indifferentiable hash function. Since these hash functions are not reset indifferentiable from ROs (one cannot directly prove from the RO model security by the reset indifferentiability), we use $\mathcal{VO}$ to prove the CDA security.

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 $x \overset{\$}{\leftarrow} X$ to mean that a value is sampled

[1] Recently, Andreeva *et al.* [1] and Chang *et al.* [17] consider the indifferentiable security of the BLAKE hash function with the more concrete structure than PFMD. In the appendix D, we prove that the BLAKE hash function with the concrete structure is reset indifferentiable from a $\mathcal{VO}$.

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

```
$\mathcal{RO}^{(i)}_{w_i}(M)$
1 if $\mathsf{F}^*_i[M] \neq \perp$ then $\mathsf{F}^*_i[M] \xleftarrow{\$} \{0,1\}^{w_i}$;
2 **return** $\mathsf{F}^*_i[M]$;
```

```
$E_t(k,x)$
1 if $\mathsf{E}_t[k,x] = \perp$, $y \xleftarrow{\$} \{0,1\}^{m_t} \backslash T^+_t[k]$;
2 $Update_t(k,x,y)$;
3 **return** $\mathsf{E}_t[k,x]$;
```

```
$\mathcal{RO}^j_{n_j}(M)$
1 If $\mathsf{F}_j[M] \neq \perp$, $\mathsf{F}_j[M] \xleftarrow{\$} \{0,1\}^{n_j}$;
2 **return** $\mathsf{F}_j[M]$;
```

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

```
$D_t(y)$
1 if $\mathsf{D}_t[k,y] = \perp$, $x \xleftarrow{\$} \{0,1\}^{m_t} \backslash T^-_t[k]$;
2 $Update_t(k,x,y)$;
3 **return** $\mathsf{D}_t[k,y]$;
```

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

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

(Reset) Indifferentiability [25, 30]. In the reset indifferentiability [30], for a functionality F, a private interface $F.priv$ and a public interface $F.pub$ are considered, where adversaries have oracle access to $F.pub$ and other parties (honest parties) have oracle access to $F.priv$. For example, for a cryptosystem in the F model, an output of the cryptosystem is calculated by accessing $F.priv$ and an adversary has oracle access to $F.pub$. In the RO model the RO has both interfaces. Let H^P be a hash function that utilizes an ideal primitive P. The interfaces of H^P are defined by $H^P.priv = H^P$ and $H^P.pub = P$.

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

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

where $\hat{S} = (S, S.Rst)$, $\bar{F_1}.priv = F_1.priv$ and $\bar{F_1}.pub = (F_1.pub, nop)$. $S.Rst$ takes no input and when run reinitializes all of S. nop takes no input and does nothing. We say F_1 is reset indifferentiable from F_2 if there exists a simulator S such that for any distinguisher A the advantage of the reset indifferentiability is negligible. This framework ensures that if F_1 is reset indifferentiable from F_2 then any stage security of any cryptosystem is preserved when F_2 is replaced with F_1. Please see Theorem 6.1 in the full version of [30].

When $S.Rst$ and nop are removed from the reset indifferentiable security game, it is equal to the original indifferentiable security game [25]. In the original indifferentiable security game, the distinguisher interacts with $(F_1.priv, F_1.pub)$ and $(F_2.priv, S^{F_2.pub})$. We denote the advantage of the indifferentiable security by $\mathsf{Adv}^{\mathsf{indif}}_{F_1,F_2,S}(A)$ for a distinguisher A. We say F_1 is indifferentiable from F_2 if there exists a simulator S such that for any distinguisher A the advantage is negligible.

3 Versatile Oracle

In this section, we propose a versatile oracle $\mathcal{VO}$. $\mathcal{VO}$ consists of a RO $\mathcal{RO}_n$, ROs $\mathcal{RO}^j_{n_j}$ ($j = 1, \ldots, v$), traceable random oracles $\mathcal{TRO}^{(i)}_{w_i}$ ($i = 1, \ldots, u$), and ideal ciphers $\mathsf{IC}^{(t)}_{k_t,m_t}$ ($t = 1, \ldots, s$). The private interface is defined by $\mathcal{VO}.priv = \mathcal{RO}_n$ and the public interface is defined by $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{RO}^j_{n_j}$ ($j = 1, \ldots, v$), $\mathcal{TRO}^{(i)}_{w_i}$ ($i = 1, \ldots, u$), $\mathsf{IC}^{(t)}_{k_t,m_t}$ ($t = 1, \ldots, s$)). $\mathcal{VO}$ can be implemented as Fig. 1.

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

$\mathcal{RO}^j_{n_j}$ is shown in Fig. 1 (Left) where the input length is arbitrary and the output length is n_j bits, and F_j is a (initially everywhere $\perp$) table. Note that n_j and v are defined in our proofs.

```
PFMD^h(M)
1 $(M_1,\ldots,M_i) \leftarrow div(m, \mathsf{pfpad}(M))$
2 $x \leftarrow IV$;
3 For $j = 1,\ldots,i$, $x \leftarrow h(x||M_j)$;
4 Ret $x$;
```

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

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

$\mathcal{TRO}_{w_i}^{(i)}$ is shown in Fig. 1 (Center) which consists of a RO $\mathcal{RO}_{w_i}^{(i)}$ and a trace oracle $\mathcal{TO}^{(i)}$. The output length of $\mathcal{RO}_{w_i}^{(i)}$ and the input length of $\mathcal{TO}^{(i)}$ are w_i bits, and F_i^* is a (initially everywhere $\perp$) table. Note that w_i and u are defined in our proofs.

$\mathsf{IC}_{k_t,m_t}^{(t)}$ can be implemented as Fig. 1 (Right) which consists of an encryption oracle E_t and a decryption oracle D_t where the first input of E_t is the key of k_t bits and the second input is the plain text of m_t bits, and the first input of D_t is the key of k_t bits and the second input is the cipher text of m_t bits. E_t and D_t are (initially everywhere $\perp$) tables where for the query $E_t(k,x)$ (resp. $D_t(k,y)$) the output is recored in $\mathsf{E}_t[k,x]$ (resp. $\mathsf{D}_t[k,y]$). $T_t^+[k]$ and $T_t^-[k]$ are (initially empty) tables which stores all values of $\mathsf{E}_t[k,\cdot]$ and $\mathsf{D}_t[k,\cdot]$, respectively. $Update_t(k,x,y)$ is the procedure wherein the tables $\mathsf{E}_t, \mathsf{D}_t, T_t^+[k]$ and $T_t^-[k]$ are updated, $\mathsf{E}_t[k,x] \leftarrow y, \mathsf{D}_t[k,y] \leftarrow x, T_t^+[k] \overset{\cup}{\leftarrow} \{y\}$ and $T_t^-[k] \overset{\cup}{\leftarrow} \{x\}$. Note that the length k_t, m_t and s are defined in our proofs.

4 Reset Indifferentiability for Hash Functions

In this section, we consider the reset indifferentiable security of the important hash functions, prefix-free Merkle-Damgård (PFMD) [18], chop Merkle-Damgård (chop MD) [18], and Sponge [11]. We show that these hash functions are reset indifferentiable from $\mathcal{VO}$s.

4.1 Reset Indifferentiability for the PFMD Hash Function

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

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

We evaluate the reset indifferentiable security from $\mathcal{VO}$ for the PFMD hash function where h is a RO. We define the parameter of $\mathcal{VO}$ as $v = 1$, $u = 1$ $n_1 = n$, and $w_1 = n$. Note that in the reset indifferentiable proof ideal ciphers are not used. Thus in this case, $\mathcal{VO}.priv = \mathcal{RO}_n$ and $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{RO}_n^1, \mathcal{TRO}_n^{(1)})$. The following theorem shows that PFMD^h is reset indifferentiable from a $\mathcal{VO}$.

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

$$\mathsf{Adv}_{\mathrm{PFMD}^h,\mathcal{VO},S}^{\mathsf{r\text{-}indiff}}(\mathcal{A}) \leq \mathcal{O}\left(\frac{(lq_H + q_h)^2}{2^n}\right)$$

where $\mathcal{A}$ can make queries to $\mathrm{PFMD}^h/\mathcal{RO}_n$ and h/S at most q_H, q_h times, respectively, and l is a maximum number of blocks of a query to $\mathrm{PFMD}^h/\mathcal{RO}_n$. S makes at most $2q_h$ queries and runs in time $\mathcal{O}(q_h)$. ♦

The Simulator S. We define the simulator S in Fig. 2 which does not update the internal state to remove the attack using $S.Rst$. The S's task is to simulate the compression function h such that $\mathcal{RO}_n$ and S are

```
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 the right $n$-bits of $x$;
```

```
S(x,m) where $x = x_1||x_2$ ($|x_1| = s$, $|x_2| = n$)
01 $M \leftarrow \mathcal{TO}^{(1)}(x_1)$;
02 if $x = IV$ then
03     $z \leftarrow \mathcal{RO}_n(m)$;
04     $w \leftarrow \mathcal{RO}_s^{(1)}(m)$;
05 else if $M \neq \perp$ then
06     $z \leftarrow \mathcal{RO}_n(M||m)$;
07     $w \leftarrow \mathcal{RO}_s^{(1)}(M||m)$;
08 else $w||z \leftarrow \mathcal{RO}^1_{n+s}(x,m)$;
09 return $w||z$;
```

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

consistent, that is, for a value M $\text{PFMD}^S(M) = \mathcal{RO}_n(M)$. We explain that the simulator S succeeds in the simulation of h. For the ordered queries $S(IV, M_1), S(z_1, M_2)$ where $z_1 = S(IV, M_1), z_2 = S(z_1, M_2)$, if there does not exists M such that $\mathsf{pfpad}(M) = M_1||M_2$, then z_1 and z_2 are defined by the responses of $\mathcal{RO}_n^{(1)}(M_1)$ and $\mathcal{RO}_n^{(1)}(M_1||M_2)$, respectively, and the Merkle-Damgård style path $(M_1||M_2, z_2)$ is recoded in the table F_1^* of $\mathcal{RO}_n^{(1)}$. Then for the query $S(z_2, M_3)$, the response is defined by the output of $\mathcal{RO}_n(M)$ if there exists M such that $\mathsf{pfpad}(M) = M_1||M_2||M_3$. Notice that $M_1||M_2$ can be obtained by the query $\mathcal{TO}^{(1)}(z_2)$. Thus the simulator S succeeds in the simulation of h. The formal proof is given in Appendix A.

Remark 1. The EMD hash function [8] and the MDP hash function [24] are designed from the same design spirit as the PFMD hash function, which are designed to resist the length extension attack. Thus, by the similar proof, one can prove that EMD and MDP are reset indifferentiable from $\mathcal{VO}$s.

4.2 Reset Indifferentiability for the Chop MD Hash Function

The chop MD hash function is employed in SHA-2 family, SHA-224 and SHA-384 [29].

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

We evaluate the reset indifferentiable security of the chop MD hash function where h is a RO. We define the parameter $\mathcal{VO}$ as $v = 1$, $u = 1$, $n_1 = s + n$ and $w_1 = s$. Note that the ideal ciphers are not used. Thus, in this case, $\mathcal{VO} = (\mathcal{RO}_n, \mathcal{RO}^1_{s+n}, \mathcal{TRO}_s^{(1)})$. The following theorem shows that chopMD^h is reset indifferentiable from a $\mathcal{VO}$.

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

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

where $\mathcal{A}$ can make queries to $\text{chopMD}^h/\mathcal{RO}_n$ *and h/S at most q_H, q_h times, respectively, and l is a maximum number of blocks of a query to* $\text{chopMD}^h/\mathcal{RO}_n$. *$S$ makes at most $3q_h$ queries and runs in time $\mathcal{O}(q_h)$.* ♦

The Simulator S. We define the simulator S in Fig. 3 which does not update the internal state to remove the attack using $S.Rst$. In the proof of Theorem 2, the padding function pad_c is removed. Thus the queries to chopMD^h and $\mathcal{RO}_n$ are in $(\{0,1\}^d)^*$. Note that the chop Merkle-Damgård hash function with the padding function is the special case of that without the padding function. The S's task is to simulate the compression function h such that $\mathcal{RO}_n$ and S are consistent, that is, for a value M, $\text{chopMD}^S(M) = \mathcal{RO}_n(M)$. For the ordered queries $S(IV, M_1), S(w_1||z_1, M_2)$ where $w_1||z_1 = S(IV, M_1), w_2||z_2 = S(w_1||z_1, M_2)$, the structure of S ensures that $z_1 = \mathcal{RO}_n(M_1)$, $w_1 = \mathcal{RO}_c^{(1)}(M_1)$, $z_2 = \mathcal{RO}_n(M_1||M_2)$, and $w_2 = \mathcal{RO}_c^{(1)}(M_1||M_2)$. Thus, the path $(M_1||M_2, w_2)$ is recoded in the table F_1^* where $\mathsf{F}_1^*[M_1||M_2] = w_2$. Then, for the query $S(w_2||z_2, M_3)$, the response $w_3||z_3$ is defined as $z_3 = \mathcal{RO}_n(M_1||M_2||M_3)$ and $w_3 = \mathcal{RO}_c^{(1)}(M_1||M_2||M_3)$. Notice that $M_1||M_2$ can be obtained by the queries $\mathcal{TO}^{(1)}(w_2)$. Thus the simulator S succeeds in the simulation of h. The formal proof is given in Appendix B.

```
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 $s = P(s \oplus (M_i||0^c))$;
5 **return** the left most $n$-bits of $s$;

$S_-(z, w)$
01 $M \leftarrow \mathcal{TO}^{(1)}(w)$;
02 if $M \neq \perp$ and $|M| = n$ then
03     $x \leftarrow IV_1 \oplus M$; $y \leftarrow IV_2$;
04 if $M \neq \perp$ and $|M| > n$ then
05     let $M = M^*||m$ ($|m| = n$);
06     if $\mathsf{unpad}_S(M^*) = M' \neq \perp$ then $x \leftarrow m \oplus \mathcal{RO}_n(M')$;
07     else $x \leftarrow m \oplus \mathcal{RO}^1_n(M^*)$;
08     $y \leftarrow \mathcal{RO}^{(1)}_c(M^*)$;
09 else $x||y \leftarrow \mathcal{P}^{-1}(z||w)$;
10 **return** $x||y$;

$S_+(x, y)$
01 $M^* \leftarrow \mathcal{TO}^{(1)}(y)$;
02 if $y = IV_2$ then
03     if $\mathsf{unpad}_S(IV_1 \oplus x) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else $z \leftarrow \mathcal{RO}^1_n(IV_1 \oplus x)$;
05     $w \leftarrow \mathcal{RO}^{(1)}_c(x \oplus IV_1)$;
06 else if $M^* \neq \perp$ then
07     if $\mathsf{unpad}(M^*) = M' \neq \perp$ then $m \leftarrow x \oplus \mathcal{RO}_n(M')$;
08     else $m \leftarrow x \oplus \mathcal{RO}^1_n(M^*)$;
08     if $\mathsf{unpad}_S(M^*||m) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
09     else $z \leftarrow \mathcal{RO}^1_n(M^*||m)$;
10     $w \leftarrow \mathcal{RO}^{(1)}_c(M^*||m)$;
11 else $z||w \leftarrow \mathcal{P}(x||y)$;
12 **return** $z||w$;
```

Fig. 4. Sponge Hash Function (left) and Simulator S (S_+ in right and S_- in left)

4.3 Reset Indifferentiability for the Sponge Hash Function

The Sponge hash function is a permutation-based hash function which employed in the SHA-3 candidate Keccak [12].

Fig. 4 (left) illustrates the Sponge hash function where IV is the initial value of b bits, $\mathsf{pad}_S : \{0,1\}^* \to (\{0,1\}^n)^*$ is an injective padding function such that the final block message $M_i \neq 0$, $P : \{0,1\}^b \to \{0,1\}^b$ is a permutation and $b = n + c$. The inverse function of pad_S is denoted by $\mathsf{unpad}_S : (\{0,1\}^n)^* \to \{0,1\}^* \cup \{\perp\}$ efficiently computable. $\mathsf{unpad}_S(M^*)$ outputs M if there exists M such that $\mathsf{pad}_S(M) = M^*$, and outputs $\perp$ otherwise. Note that the Sponge hash function of Fig. 4 is the special case of the general Sponge hash function where the output length is arbitrary. The output lengths of SHA-3 are $224, 256, 384$ and 512 bits and in this case the Keccak hash function has the structure of Fig. 4.[2] We conjecture that the reset indifferentiable security of the general Sponge hash function can be proven by extending the following analysis of the Sponge hash function. We denote the left most n-bit value and the right most c bit value of IV by IV_1 and IV_2, respectively. Namely, $IV = IV_1||IV_2$.

We evaluate the reset indifferentiable security of the Sponge hash function in the random permutation model, where P is a forward oracle of the random permutation and P^{-1} is its inverse oracle.[3] We define the parameter of $\mathcal{VO}$ as $v = 1$, $u = 1$, $s = 1$, $n_1 = n$ $w_1 = c$, and $m_1 = b$. We don't care the key length k_1, since in this proof we fix the key by some constant value, that is the fixed key ideal cipher is used, which is a random permutation of b bits. So we use the random permutation $(\mathcal{P}, \mathcal{P}^{-1})$ of b bits instead of the ideal cipher $\mathsf{IC}^{(1)}_{k_1,b}$ where $\mathcal{P}$ is a forward oracle and $\mathcal{P}^{-1}$ is an inverse oracle. Thus, in this case, $\mathcal{VO}.priv = \mathcal{RO}_n$ and $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{RO}^1_n, \mathcal{TRO}^{(1)}_c, \mathcal{P}, \mathcal{P}^{-1})$. The following theorem is that the sponge hash function $Sponge^P$ is reset indifferentiable from $\mathcal{VO}$.

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

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff}}_{Sponge^P,\mathcal{VO},S}(\mathcal{A}) \leq \frac{(1-2^{-n})q^2 + (1+2^{-n})q}{2^{c+1}} + \frac{q^2}{2^{b-3}} + \frac{q(9q+4)}{2^{c+1}}$$

where $\mathcal{A}$ can make at most q queries. S makes at most $3q$ queries and runs in time $\mathcal{O}(q)$. ♦

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

[3] The security of the Sponge hash function was evaluated in the random permutation model [11].

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

Fig. 5. CDA Security Game

The Simulator S. We define the simulator S in Fig. 4 which does not update the internal state to remove the attack using $S.Rst$. The S's task is to simulate the random permutation (P, P^{-1}) such that $\mathcal{RO}_n$ and S are consistent, that is, for a value M, $Sponge^{S_+}(M) = \mathcal{RO}_n(M)$. S_+ and S_- simulate P and P^{-1}, respectively. For the ordered queries $S_+(x_1, IV_2), S_+(x_2, w_1)$ where $z_1||w_1 = S_+(x_1, IV_2), z_2||w_2 = S_+(x_2, w_1)$, the structure of S ensures that $w_1 = \mathcal{RO}_c^{(1)}(M_1)$ and $w_2 = \mathcal{RO}_c^{(1)}(M_1||M_2)$ where $M_1 = IV_1 \oplus x_1$ and $M_2 = z_1 \oplus x_2$. Thus, the path $(M_1||M_2, w_2)$ is recoded in the table F_1^* where $\mathsf{F}_1^*[M_1||M_2] = w_2$. Then, for the query $S_+(x_3, w_2)$, the response $w_3||z_3$ is defined as $w_3 = \mathcal{RO}_n(M)$ and $z_3 = \mathcal{RO}_c^{(1)}(M_1||M_2||M_3)$, if $\mathsf{unpad}_S(M_1||M_2||M_3) = M \neq \perp$ where $M_3 = z_2 \oplus x_3$. Notice that $M_1||M_2$ can be obtained by the queries $\mathcal{TO}^{(1)}(w_2)$ and z_2 can be obtained by the query $\mathcal{RO}_n(\mathsf{unpad}_S(M_1||M_2))$ or the query $\mathcal{RO}_n^1(M_1||M_2)$. Thus the simulator S succeeds in the simulation of the random permutation. The formal proof is given in Appendix C.

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

We show that for any PKE scheme, the CDA security [5] (including the PRIV security [4]) is preserved when a RO is replaced with a $\mathcal{VO}$. This ensures both the adaptive case and the non-adaptive case and both the CCA case and the CPA case.

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

CDA Security. We explain the CDA security (we quote the explanation of the CDA security in [30]). Fig. 5 illustrates the non-adaptive CDA game in the CPA case for a PKE scheme $\mathcal{AE}$ using a functionality F. We explain the adaptive case and the CCA case, later. This notion captures the security of a PKE scheme when the randomness $\mathbf{r}$ used may not be a string of uniform bits. For the remainder of this section, fix a randomness length $\rho \geq 0$ and a message length $\omega > 0$. An (μ, ν)-mmr-source $\mathcal{M}$ is a randomized algorithm that outputs a triple of vector $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$ such that $|\mathbf{m}_0| = |\mathbf{m}_1| = |\mathbf{r}| = \nu$, 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 $(\mathbf{m}_b[i], \mathbf{r}[i]) \neq (\mathbf{m}_b[j], \mathbf{r}[j])$ for all $1 \leq i < j \leq \nu$ and all $b \in \{0,1\}$. Moreover, the source has mini-entropy μ, meaning $\Pr[(\mathbf{m}_b[i], \mathbf{r}[i]) = (m', r') | (\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow \mathcal{M}] \leq 2^{-\mu}$ for all $b \in \{0,1\}$, 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

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

In the adaptive case, the adversary $\mathcal{A}_1$ can select multiple triples $(\mathbf{m}_{0,0}, \mathbf{m}_{0,1}, \mathbf{r}_0), \ldots, (\mathbf{m}_{j,0}, \mathbf{m}_{j,1}, \mathbf{r}_j)$ adaptively, where before selecting $(\mathbf{m}_{i,0}, \mathbf{m}_{i,1}, \mathbf{r}_i)$, $\mathcal{A}_1$ can know cipher texts $\mathbf{c}_0, \ldots, \mathbf{c}_{i-1}$ of $(\mathbf{m}_{0,b}, \mathbf{r}_0), \ldots, (\mathbf{m}_{i-1,b}, \mathbf{r}_{i-1})$ where $b \in \{0,1\}$. The adversary $\mathcal{A}_2$ can receive its cipher texts $\mathbf{c}_0, \ldots, \mathbf{c}_j$. In the CCA case, the adversary has oracle access to the decryption oracle where the queries don't appear as a component of the cipher text(s). We note that the outputs of $\mathcal{A}_1$ are independent of the coins used by $F.pub$. If this condition is removed, one can easily break the CDA security for any cryptosystem using the indifferentiable hash function. By using the randomness of the underlying primitive (e.g., a RO compression function), $\mathcal{A}_1$ and $\mathcal{A}_2$ can easily share the messages $\mathbf{m}_1, \mathbf{m}_2, \mathbf{r}$. The definition of the CDA security game in [30] is that the messages are independent of the coins used by *F.pub*.

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

The CDA (PRIV) Security When a RO is replaced with a $\mathcal{VO}$. The following theorem shows that for any PKE scheme the adaptive CDA security and the non-adaptive CDA security in both CPA and CCA cases are preserved when a RO is replaced with a $\mathcal{VO}$.

Theorem 4. *Let $\mathcal{AE}$ be a PKE scheme. For any CDA adversary $\mathcal{A}_1, \mathcal{A}_2$ in the $\mathcal{VO}$ model making at most $q_1, q_2, q_3, q_{\mathcal{TO}}, q_4$ queries to $\mathcal{RO}_n, \mathcal{RO}^j_{n_j}$ $(j = 1, \ldots, v), \mathcal{TRO}^{(i)}_{w_i} = (\mathcal{RO}^{(i)}_i, \mathcal{TO}^{(i)})$ $(i = 1, \ldots, u), \mathsf{IC}^{(t)}_{k_t,m_t}$ $(t = 1, \ldots, s)$, there exists a CDA adversary A_1, A_2 in the RO model such that*

$$\mathsf{Adv}^{\mathrm{cda}}_{\mathcal{AE},\mathcal{VO}}(A_1, A_2) \le \mathsf{Adv}^{\mathrm{cda}}_{\mathcal{AE},\mathcal{RO}_n}(\mathcal{A}_1, \mathcal{A}_2) + \min_{i=1,\ldots,u}\{\frac{q_3 \times q_{\mathcal{TO}^1}}{2^{w_i - 1}}\}.$$

A_1, A_2 runs in time that of $\mathcal{A}_1, \mathcal{A}_2$ plus $\mathcal{O}(q_1 + q_2 + q_3 + q_{\mathcal{TO}} + q_4)$ and makes at most q_1 queries to $\mathcal{RO}_n$. ♦

Proof. The following proof is the CPA case. We discuss the CCA case later. Assume that there exists the CDA adversary $\mathcal{A}_1, \mathcal{A}_2$ against $\mathcal{AE}$ in $\mathcal{VO}$ model. In this proof, we construct a CDA game adversary A_1, A_2 against $\mathcal{AE}$ in the the RO ($\mathcal{RO}_n$) model by using the CDA adversary $\mathcal{A}_1, \mathcal{A}_2$. A_1, A_2 is shown in Fig. 6. A_1 uses $\mathcal{RO}_b$ and $\mathcal{A}_1$, and A_1 simulates $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{RO}^j_{n_j}$ $(j = 1, \ldots, v), \mathcal{TRO}^{(i)}_{w_i}$ $(i = 1, \ldots, u), \mathsf{IC}^{(t)}_{k_t,m_t}$ $(t = 1, \ldots, s))$ where $S1pub = (S1pub_{\mathcal{RO}_n}, S1pub_{\mathcal{RO}^j_{n_j}}, S1pub_{\mathcal{RO}^{(i)}_{w_i}}, S1pub_{\mathcal{TO}^{(i)}}, S1pub_{E_t}, S1pub_{D_t})$ is its simulation oracle. Similarly, A_2 uses $\mathcal{RO}_n$ and $\mathcal{A}_2$, and A_2 simulates $\mathcal{VO}$ where $S2pub$ is its simulation oracle. f_j, f^*_j, f^+_t, f^-_t, ff_j, ff^*_j, ff^+_t, and ff^-_t are (initially everywhere $\perp$) tables to simulate these oracles and $\mathcal{T}1^+_t$, $\mathcal{T}1^-_t$, $\mathcal{T}2^+_t$, and $\mathcal{T}2^-_t$ are (initially empty) tables to implement E_t and D_t where for any k, x such that $\mathsf{f}^+_t[k,x] \ne \perp$, $\mathsf{f}^+_t[k,x] \in \mathcal{T}1^+[k]$, for any k, y such that $\mathsf{f}^-_t[k,y] \ne \perp$, $\mathsf{f}^-_t[k,y] \in \mathcal{T}1^-_t[k]$, for any k, x such that $\mathsf{ff}^+_t[k,x] \ne \perp$, $\mathsf{ff}^+_t[k,x] \in \mathcal{T}2^+_t[k]$, and for any k, y such that $\mathsf{ff}^-_t[k,y] \ne \perp$, $\mathsf{ff}^-_t[k,y] \in \mathcal{T}2^-_t$. Note that this figure illustrates the non-adaptive case. One can easily extend this to the adaptive case by rewriting this figure where A_1 selects multiple triples adaptively, A_2 receives its cipher texts, $\mathcal{A}_1$ selects multiple triples adaptively, and $\mathcal{A}_2$ receives its cipher texts. The following proof can be applied to the adaptive case, since the number of the triples which $\mathcal{A}_1$ (A_1) outputs does not effect the simulation of $\mathcal{VO}$.

If the simulation of $\mathcal{VO}$ is success and the adversary $\mathcal{A}_1, \mathcal{A}_2$ wins the CDA game, the adversary A_1, A_2 wins. We thus show that the probability that the simulation fails is negligible.

$S1pub$ succeeds in the simulation of $\mathcal{VO}.pub$. $S2pub_{\mathcal{RO}_n}$ succeeds in the simulation of $\mathcal{RO}_n$. $S2pub_{\mathcal{RO}^j_{n_j}}$ succeeds in the simulation of $\mathcal{RO}^j_{n_j}$. $S2pub_{\mathcal{RO}^{(i)}_{n_i}}$ succeeds in the simulation of $\mathcal{RO}^i_{n_i}$. $S2pub_{E_t}$ and $S2pub_{D_t}$ succeed in the simulation of E_t and D_t. Notice that A_2 cannot see the tables of S_1pub and S_2pub. So these simulators of $S2pub$ is not consistent with those of $S1pub$ but $\mathcal{A}_2$ cannot find the inconsistency, since $\mathcal{A}_2$ is given nothing from $\mathcal{A}_1$ and the output of $\mathcal{A}_1$ is independent from these simulators due to the definition of the CDA security game (the adversary's output is independent from $\mathcal{VO}.pub$. $S2pub_{\mathcal{TO}^{(i)}}$ $i = 1, \ldots, u$ succeed in the simulation of $\mathcal{TO}^{(i)}$ with overwhelming probability. Since $S2pub_{\mathcal{RO}^{(i)}_{w_i}}$ is not consistent with $S1pub_{\mathcal{RO}^{(i)}_{w_i}}$, for the query $S2pub_{\mathcal{TO}^{(i)}}(y)$ such that there is a value w such that $\mathsf{f}^*_i[w] = y$, the output is $\perp$. In this case, thus, we fail the simulation. Since y is randomly chosen from $\{0,1\}^{w_i}$ and $\mathcal{A}_2$ is given nothing from $\mathcal{A}_1$, the failure probability is $\le \min_{i=1,\ldots,u}\{\frac{q_3 \times q_{\mathcal{TO}}}{2^{w_i}}\}$.

Thus $\Pr[\mathrm{CDA}^{A_1,A_2}_{\mathcal{AE},\mathcal{VO}} \Rightarrow \mathsf{true}] \le \Pr[\mathrm{CDA}^{\mathcal{A}_1,\mathcal{A}_2}_{\mathcal{AE},\mathcal{RO}_n} \Rightarrow \mathsf{true}] + \min_{i=1,\ldots,u}\{\frac{q_3 \times q_{\mathcal{TO}}}{2^{w_i}}\}$.

Consider the CCA case. In this case, the adversaries A_2 and $\mathcal{A}_2$ have oracle access to the decryption oracles. Thus, A_2 simulates the decryption oracle which $\mathcal{A}_2$ uses as follows: for the decryption query x from $\mathcal{A}_2$, A_2 queries x to the decryption oracle. And A_2 returns the response of the decryption oracle to $\mathcal{A}_2$. Clearly, the above proof works in the CCA case, since the decryption oracle simulation does not affect the

```
$\underline{\mathrm{CDA}^{A_1,A_2}_{\mathcal{AE},\mathcal{RO}_n}(M)}$
$b \xleftarrow{\$} \{0,1\}$
$(pk, sk) \xleftarrow{\$} \mathcal{K}$
$(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow A_1^{\mathcal{RO}_n,\mathcal{A}_1}$
$c \leftarrow \mathcal{E}^{\mathcal{RO}_n}(pk, \mathbf{m}_b; \mathbf{r})$
$b' \leftarrow A_2^{\mathcal{RO}_n,\mathcal{A}_2}(pk, c)$
Ret $(b = b')$

$\underline{A_1^{\mathcal{RO}_n,\mathcal{A}_1}}$
$(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r}) \leftarrow \mathcal{A}_1^{S1pub}$
Ret $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{r})$

$\underline{A_2^{\mathcal{RO}_n,\mathcal{A}_2}(pk, c)}$
$b' \leftarrow A_2^{S2pub}(pk, c)$
Ret $b'$
```

```
$\underline{S1pub_{\mathcal{RO}_n}(M)}$
Ret $\mathcal{RO}_n(M)$

$\underline{S1pub_{\mathcal{RO}^j_{n_j}}(M)}$
If $\mathsf{f}_j[M] = \perp$, $\mathsf{f}_j[M] \xleftarrow{\$} \{0,1\}^{n_j}$;
Ret $\mathsf{f}_j[M]$;

$\underline{S1pub_{\mathcal{RO}^{(i)}_{w_i}}(M)}$
If $\mathsf{f}^*_i[M] = \perp$, $\mathsf{f}^*_i[M] \xleftarrow{\$} \{0,1\}^{w_i}$;
Ret $\mathsf{f}^*_i[M]$;

$\underline{S1pub_{\mathcal{TO}^{(i)}}(y)}$
If $\exists M$ s.t. $\mathsf{f}^*_i[M] = y$, ret $M$;
Ret $\perp$;

$\underline{S1pub_{E_t}(k, x)}$
If $\mathsf{f}^+_t[k, x] = \perp$,
    $y \xleftarrow{\$} \{0,1\}^{m_t} \backslash \mathcal{T}1^+_t[k]$;
    $\mathsf{f}^+_t[k, x] \leftarrow y$; $\mathsf{f}^-_t[k, y] \leftarrow x$;
Ret $\mathsf{f}^+_t[k, x]$

$\underline{S1pub_{D_t}(k, y)}$
If $\mathsf{f}^-_t[k, y] = \perp$,
    $x \xleftarrow{\$} \{0,1\}^{m_t} \backslash \mathcal{T}1^-_t[k]$;
    $\mathsf{f}^+_t[k, x] \leftarrow y$; $\mathsf{f}^-_t[k, y] \leftarrow x$;
Ret $\mathsf{f}^-_t[k, y]$;
```

```
$\underline{S2pub_{\mathcal{RO}_n}(M)}$
Ret $\mathcal{RO}_n(M)$

$\underline{S2pub_{\mathcal{RO}^j_{n_j}}(M)}$
If $\mathsf{ff}_j[M] = \perp$, $\mathsf{ff}_j[M] \xleftarrow{\$} \{0,1\}^{n_j}$;
Ret $\mathsf{ff}_j[M]$;

$\underline{S2pub_{\mathcal{RO}^{(i)}_{w_i}}(M)}$
If $\mathsf{ff}^*_i[M] = \perp$, $\mathsf{ff}^*_i[M] \xleftarrow{\$} \{0,1\}^{w_i}$;
Ret $\mathsf{ff}^*_i[M]$;

$\underline{S2pub_{\mathcal{TO}^{(i)}}(y)}$
If $\exists M$ s.t. $\mathsf{ff}^*_i[M] = y$, ret $M$;
Ret $\perp$;

$\underline{S2pub_{E_t}(k, x)}$
If $\mathsf{ff}^+_t[k, x] = \perp$,
    $y \xleftarrow{\$} \{0,1\}^{m_t} \backslash \mathcal{T}2^+_t[k]$;
    $\mathsf{ff}^+_t[k, x] \leftarrow y$; $\mathsf{ff}^-_t[k, y] \leftarrow x$;
Ret $\mathsf{ff}^+_t[k, x]$

$\underline{S2pub_{D_t}(k, y)}$
If $\mathsf{ff}^-_t[k, y] = \perp$,
    $x \xleftarrow{\$} \{0,1\}^{m_t} \backslash \mathcal{T}2^-_t[k]$;
    $\mathsf{ff}^+_t[k, x] \leftarrow y$; $\mathsf{ff}^-_t[k, y] \leftarrow x$;
Ret $\mathsf{ff}^-_t[k, y]$;
```

Fig. 6. CDA game adversary A_1, A_2

simulation of $\mathcal{VO}$ (In $\mathcal{VO}$ model, $\mathcal{RO}^j_{n_j}$ $(j = 1, \ldots, v), \mathcal{TRO}^{(i)}_{w_i}$ $(i = 1, \ldots, u), \mathsf{IC}^{(t)}_{k_t,m_t}$ $(t = 1, \ldots, s))$ are not used in the PKE scheme). $\square$

6 Conclusion and Future Works

We proved that for any PKE scheme the CDA is preserved for the adaptive CDA security and the non-adaptive CDA security in both CPA and CCA cases when a RO is replaced with the indifferentiable hash functions, PFMD, chop MD, and Sponge. First, we proposed the Versatile Oracle $\mathcal{VO}$, and showed that these hash functions are reset indifferentiable from $\mathcal{VO}$s. Second, we proved that for any PKE scheme the CDA security are preserved when a RO is replaced with a $\mathcal{VO}$. Then the reset indifferentiable composition theorem ensures our result. So far, there is no positive result for the reset indifferentiability. So, our result is the first time positive result.

For other indifferentiable hash functions, e.g., the SHA-3 finalists JH [31] and Grøstl [23], the CDA security is still open. We conjecture that our approach can be applied to the CDA security proof for these indifferentiable hash functions. For other cases, e.g., deterministic and hedged identity-based encryption schemes [7], we can similarly prove the security by our approach.

Acknowledgements
The author would like to thank the anonymous Eurocrypt 2012 and CRYPTO 2012 referees for very useful comments that helped to improve the paper.

References

1. Elena Andreeva, Atul Luykx, and Bart Mennink. Provable Security of BLAKE with Non-Ideal Compression Function, ePrint 2011/620.
2. Elena Andreeva, Bart Mennink, and Bart Preneel. On the Indifferentiability of the Grøstl Hash Function. In *SCN*, volume 6280 of *LNCS*, pages 88–105. Springer, 2010.
3. Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and Raphael C.-W. Phan. SHA-3 proposal BLAKE. Submission to NIST (Round 3). 2010.
4. Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and Efficiently Searchable Encryption. In *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552. Springer, 2007.
5. Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In *ASIACRYPT*, volume 5912 of *LNCS*, pages 232–249. Springer, 2009.
6. Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. In *CRYPTO*, volume 5157 of *LNCS*, pages 360–378. Springer, 2008.
7. Mihir Bellare, Eike Kiltz, Chris Peikert, and Brent Waters. Identity-Based (Lossy) Trapdoor Functions and Applications. (Ful Version in ePrint 2011/479). In *EUROCRYPT*, 2012.
8. 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.
9. Mihir Bellare and Phillip Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *EUROCRYPT*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 1994.
10. Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer, 1996.
11. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the Indifferentiability of the Sponge Construction. In *EUROCRYPT*, pages 181–197, 2008.
12. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak SHA-3 submission. Submission to NIST (Round 3). 2011.
13. Rishiraj Bhattacharyya, Avradip Mandal, and Mridul Nandi. Security Analysis of the Mode of JH Hash Function. In *FSE*, volume 6147 of *LNCS*, pages 168–191. Springer, 2010.
14. 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.
15. 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.
16. Donghoon Chang and Mridul Nandi. Improved Indifferentiability Security Analysis of chopMD Hash Functionl. In *FSE*, pages pages 429–443, 2008.
17. Donghoon Chang, Mridul Nandi, and Moti Yung. Indifferentiability of the Hash Algorithm BLAKE, ePrint 2011/623. 2011.
18. 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.
19. Ivan Damgård. A Design Principle for Hash Functions. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.
20. 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.
21. Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The Skein Hash Function Family. Submission to NIST (Round 3). 2010.
22. Benjamin Fuller, Adam O'Neill, and Leonid Reyzin. A Unified Approach to Deterministic Encryption: New Constructions and a Connection to Computational Entropy. ePrint 2012/005.
23. Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schäffer, and Søren S. Thomsen. Grøstl – a SHA-3 candidate. Submission to NIST (Round 3). 2011.
24. 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.
25. 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.

26. Ralph C. Merkle. One Way Hash Functions and DES. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.
27. Ilya Mironov, Omkant Pandey, Omer Reingold, and Gil Segev. Incremental Deterministic Public-Key Encryption, (Full Version in ePrint 2012/047). In *EUROCRYPT*, 2012.
28. National Institute of Standards and Technology. Cryptographic Hash Algorithm Competition.
29. National Institute of Standards and Technoloty. FIPS PUB 180-3 Secure Hash Standard. In *FIPS PUB*, 2008.
30. 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.
31. Hongjun Wu. The Hash Function JH. Submission to NIST (Round 3). 2011.

```
$S^*(x,y)$
01 if $T_{S^*}[x,y] \neq \perp$ then **return** $T_{S^*}[x,y]$;
02 if $x = IV$ then
03     if $\exists M$ s.t. $\mathsf{pfpad}(M) = y$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else
05         $z \xleftarrow{\$} \{0,1\}^n$;
06         if $Path[z] = \perp$ then $Path[z] \leftarrow y$;
07 else if $Path[x] = M^* \neq \perp$ then
08     if $\exists M$ s.t. $\mathsf{pfpad}(M) = M^*||y$ then $z \leftarrow \mathcal{RO}_n(M)$;
09     else
10         $z \xleftarrow{\$} \{0,1\}^n$;
11         if $Path[z] = \perp$ then $Path[z] \leftarrow M^*||y$;
12 else $z \xleftarrow{\$} \{0,1\}^n$;
13 $T_{S^*}[x,y] \leftarrow z$;
14 **return** $T_{S^*}[x,y]$;
```

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

Fig. 7. Simulator S^* (Left) and Game PF1 (Right)

```
$\mathcal{O}(x,y)$
01 if $x = IV$ then
02     if $\exists M$ s.t. $\mathsf{pfpad}(M) = y$ then $z \leftarrow \mathcal{RO}_n(M)$;
03     else
04         if $\mathsf{F}_1^*[y] = \perp$ then $\mathsf{F}_1^*[y] \xleftarrow{\$} \{0,1\}^n$;
05         $z \leftarrow \mathsf{F}_1^*[y]$;
06 else if $\exists_1 M^*$ s.t. $\mathsf{F}_1^*[M^*] = x$ then
07     if $\exists M$ s.t. $\mathsf{pfpad}(M) = M^*||y$ then $z \leftarrow \mathcal{RO}_n(M)$;
08     else
09         if $\mathsf{F}_1^*[M^*||y] = \perp$ then $\mathsf{F}_1^*[M^*||y] \xleftarrow{\$} \{0,1\}^n$;
10         $z \leftarrow \mathsf{F}_1^*[M^*||y]$;
11 else
12     if $\mathsf{F}_1[x||y] = \perp$ then $\mathsf{F}_1[x||y] \xleftarrow{\$} \{0,1\}^n$;
13     $z \leftarrow \mathsf{F}_1[x||y]$;
14 $T_{S^*}[x,y] \leftarrow z$;
15 **return** $T_{S^*}[x,y]$;
```

```
$\mathcal{O}(x,y)$
01 if $T_{S^*}[x,y] \neq \perp$ then **return** $T_{S^*}[x,y]$;
02 if $x = IV$ then
03     if $\exists M$ s.t. $\mathsf{pfpad}(M) = y$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else
05         if $\mathsf{F}_1^*[y] = \perp$ then $\mathsf{F}_1^*[y] \xleftarrow{\$} \{0,1\}^n$;
06         $z \leftarrow \mathsf{F}_1^*[y]$;
07 else if $\exists_1 M^*$ s.t. $\mathsf{F}_1^*[M^*] = x$ then
08     if $\exists M$ s.t. $\mathsf{pfpad}(M) = M^*||y$ then $z \leftarrow \mathcal{RO}_n(M)$;
09     else
10         if $\mathsf{F}_1^*[M^*||y] = \perp$ then $\mathsf{F}_1^*[M^*||y] \xleftarrow{\$} \{0,1\}^n$;
11         $z \leftarrow \mathsf{F}_1^*[M^*||y]$;
12 else
13     if $\mathsf{F}_1[x||y] = \perp$ then $\mathsf{F}_1[x||y] \xleftarrow{\$} \{0,1\}^n$;
14     $z \leftarrow \mathsf{F}_1[x||y]$;
15 $T_{S^*}[x,y] \leftarrow z$;
16 **return** $T_{S^*}[x,y]$;
```

Fig. 8. Game PF2 (left), and Game PF3 (right)

A Proof of Theorem 1

We prove Theorem 1 by using the strategy in Appendix **??**.

Since the simulator S does not update the internal state, $\Pr[G0] - \Pr[G1] = 0$.

We use the result of the indifferentiable security by Chang *et al.* [15] They defined a simulator S^* which is shown in Fig. 7. T_{S^*} is a (initially everywhere $\perp$) table which records query-response values of S^*. For the query $S^*(x,y)$, the response is recorded in $T_{S^*}[x,y]$. $Path$ is a (initially everywhere $\perp$) table which records all paths with the Merkle-Damgård style. If triples $(z_0,y_1,z_1),(z_1,y_2,z_2),(z_2,y_3,z_3)$ are recoded in T_{S^*} where $T_{S^*}[z_{j-1},y_j] = z_j$ and $z_0 = IV$, $y_1||y_2||y_3$ is recoded in $Path[z_3]$.[4] The task of the simulator S^* is to simulate h so that $\mathcal{RO}_n$ and S^* are consistent. So the response of $S^*(x,y)$ is defined by the output of $\mathcal{RO}_n(M)$ if there exists M^* such that $Path[x] = M^*$ and there exists M such that $\mathsf{pfpad}(M) = M^*||y$. They show that the advantage p^* of the indifferentiable security is bounded by $\mathcal{O}((lq_H + q_h)^2/2^n)$.

We evaluate the difference $\Pr[G1] - \Pr[G2]$ where the distinguisher A_1 interacts with $(\mathcal{RO}_n, S)$ in Game 1 and $(\mathcal{RO}_n, S^*)$ in Game 2.

[4] Note that in [15], the paths are recorded by using another formula, which is a relation $\mathcal{R}$, but the table $Path$ realizes the same role as the relation.

```
$\underline{\mathcal{O}(x,y)}$
01 if $T_{S^*}[x,y] \neq \perp$ then **return** $T_{S^*}[x,y]$;
02 if $x = IV$ then
03     if $\exists M$ s.t. $\mathsf{pfpad}(M) = y$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else
05         $z \xleftarrow{\$} \{0,1\}^n$;
06         $\mathsf{F}_1^*[y] \leftarrow z$;
07 else if $\exists_1 M^*$ s.t. $\mathsf{F}_1^*[M^*] = x$ then
08     if $\exists M$ s.t. $\mathsf{pfpad}(M) = M^*||y$ then $z \leftarrow \mathcal{RO}_n(M)$;
09     else
10         $z \xleftarrow{\$} \{0,1\}^n$;
11         $\mathsf{F}_1^*[M^*||y] \leftarrow z$;
12 else
13     $z \xleftarrow{\$} \{0,1\}^n$;
14     $\mathsf{F}_1[x||y] \leftarrow z$;
15 $T_{S^*}[x,y] \leftarrow z$;
16 **return** $T_{S^*}[x,y]$;
```

```
$\underline{\mathcal{O}(x,y)}$
01 if $T_{S^*}[x,y] \neq \perp$ then **return** $T_{S^*}[x,y]$;
02 if $x = IV$ then
03     if $\exists M$ s.t. $\mathsf{pfpad}(M) = y$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else
05         $z \xleftarrow{\$} \{0,1\}^n$;
06         if $Path[z] = \perp$ then $Path[z] \leftarrow y$;
07 else if $Path[x] = M^* \neq \perp$ then
08     if $\exists M$ s.t. $\mathsf{pfpad}(M) = M^*||y$ then $z \leftarrow \mathcal{RO}_n(M)$;
09     else
10         $z \xleftarrow{\$} \{0,1\}^n$;
11         if $Path[z] = \perp$ then $Path[z] \leftarrow M^*||y$;
12 else $z \xleftarrow{\$} \{0,1\}^n$;
13 $T_{S^*}[x,y] \leftarrow z$;
14 **return** $T_{S^*}[x,y]$;
```

Fig. 9. Game PF4 (left), and Game PF5 (right)

We consider the five games, Game PF1, Game PF2, Game PF3, Game PF4, and Game PF5. In each game, the distinguisher A_1 interacts with $(\mathcal{RO}_n, \mathcal{O})$ where $\mathcal{O}$ is shown in Figs. 7, 8, and 9. $\mathcal{O}$ in Game PF1 is equal to S and $\mathcal{O}$ in Game PF5 is equal to S^*. Let $GPFj$ be an event that in Game PFj A_1 outputs 1. Thus

$$\Pr[G1] - \Pr[G2] = \Pr[GPF1] - \Pr[GPF5]$$
$$= \sum_{j=1}^{4} \Pr[GPFj] - \Pr[GPF(j+1)]$$

First we evaluate the difference $\Pr[GPF1] - \Pr[GPF2]$. In Game PF2, the procedures of $\mathcal{TRO}_n^{(1)}$ and $\mathcal{RO}_n^1$ are hard-coded. Thus this modification does not affect the distinguisher's behavior and $\Pr[GPF1] = \Pr[GPF2]$.

We evaluate the difference $\Pr[GPF2] - \Pr[GPF3]$. In Game PF3, for any (x, y), the table $T_{S^*}[x, y]$ is not redefined, while in Game PF2 there are cases that the table is redefined. For Game PF2 we consider the following cases which cover all cases that the table is redefined. So the modification from Game PF2 to Game PF3 does not affect the distinguisher's behavior if the following cases don't occur.

- Case 1: First $T_{S^*}[x, y]$ was defined by a value in the step 05 and then it is redefined due to another step (07, 10, or 13).
- Case 2: First $T_{S^*}[x, y]$ was defined by a value in the step 07 or 10 and then it is redefined due to another step (05 or 13).
- Case 3: First $T_{S^*}[x, y]$ was defined by a value in the step 13 and then it is redefined due to another step (05, 07, or 10).

Consider Case 1. In this case, $x = IV$. Since for any y $T_{S^*}[IV, y]$ is defined by a value from the step 05, this case does not occurs.
Consider Case 2. In this case, $x \neq IV$. So the table is redefined due to the step 13. When $T_{S^*}[x, y]$ was defined by a value from the step 07 or 10, there was just a value M such that $\mathsf{F}_1^*[M] = x$. Thus, when $T_{S^*}[x, y]$ is redefined, there is another value M' such that $\mathsf{F}_1^*[M'] = x$. So a collision for F_1^* occurs and the collision probability is bounded by $q_h^2/2^{n+1}$ from a birthday analysis.
Consider Case 3. Since $T_{S^*}[x, y]$ is defined in the step 13, $x \neq IV$, that is, $T_{S^*}[x, y]$ is not redefined in the step 05 and is redefined in the step 07 or 10. When $T_{S^*}[x, y]$ was defined in the step 13, there was not a value M^* such that $\mathsf{F}_1^*[M^*] = x$. When $T_{S^*}[x, y]$ is redefined in the step 07 or 10, there is such value. Since $\mathsf{F}_1^*[M^*]$ is uniformly chosen at random from $\{0,1\}^n$, this case is that a random value of n bits hits x. Since

the number of the $\mathcal{O}$ query is at most q_h, the probability of Case 2 is bounded by $1-(1-q_h/2^n)^{q_h} \le q_h^2/2^n$. We thus have

$$\Pr[GPF2] - \Pr[GPF3] \le \frac{3q_h^2}{2^{n+1}}.$$

We evaluate the difference $\Pr[GPF3] - \Pr[GPF4]$. In Game PF4, "if" in the steps 05, 10, and 13 is removed. So in Game PF4, F_1^* or F_1 might be redefined. However, the table T_{S^*} prevents the redefinition due to the step 01: For a pair (x,y), $T_{S^*}^*[x,y] = \mathsf{F}_1^*[y]$ if $x = IV$, $T_{S^*}^*[x,y] = \mathsf{F}_1^*[M^*||y]$ if $x \ne IV$ and there exists a value M^* such that $\mathsf{F}_1^*[M^*] = x$, and $T_{S^*}^*[x,y] = \mathsf{F}_1[x||y]$ otherwise. Thus this modification does not affect the distinguisher's behavior and $\Pr[GPF3] = \Pr[GPF4]$.

Finally, we evaluate the difference $\Pr[GPF4] - \Pr[GPF5]$. In Game PF4, the table F_1^* is replaced with the table $Path$ and F_1 is removed. Note that F_1 is not used in Game PF4. For a pair (M,z), if $\mathsf{F}_1^*[M] = z$ and no collision for the table F_1^* occurs then $Path[z] = M$, and if $Path[z] = M$ and no collision for the table F_1^* occurs then $\mathsf{F}_1^*[M] = z$. Thus if no collision for the table F_1^* occurs, this modification does not affects the distinguisher's behavior. The collision probability is at most $q_h^2/2^{n+1}$ from a birthday analysis. We thus have

$$\Pr[GPF4] - \Pr[GPF5] \le \frac{q_h^2}{2^{n+1}}.$$

□

```
$S^*(x,m)$
01 if $T_{S^*}[x,m] \neq \perp$ then return $T_{S^*}[x,m]$;
02 if $x = IV$ then
03     $z \leftarrow \mathcal{RO}_n(m)$;
04     $w \xleftarrow{\$} \{0,1\}^s \backslash \{w' : w'||z \in C \cup \{x\}\}$;
05     $Path[w||z] \leftarrow m$;
06 else if $Path[x] = M \neq \perp$ then
07     $z \leftarrow \mathcal{RO}_n(M||m)$;
08     $w \xleftarrow{\$} \{0,1\}^s \backslash \{w' : w'||z \in C \cup \{x\}\}$;
09     $Path[w||z] \leftarrow M||m$;
10 else
11     $z \xleftarrow{\$} \{0,1\}^n$;
12     $w \xleftarrow{\$} \{0,1\}^s \backslash \{w' : w'||z \in C \cup \{x\}\}$;
13 $T_{S^*}[x,m] \leftarrow w||z$; $C \xleftarrow{\cup} \{x,z\}$;
14 return $w||z$;
```

Fig. 10. Simulator S^*

B Proof of Theorem 2

We prove Theorem 2 by using the strategy shown in Appendix **??**.

Since the simulator S does not update the internal state, $\Pr[G0] - \Pr[G1] = 0$.

We use the result of the indifferentiable security by Chang and Nandi [16]. They define a simulator S^* which is shown in Fig. 10 which simulates a compression function h. T_{S^*} is a (initially everywhere $\perp$) table which records query-response values of S^*. For the query $S^*(x,m)$, the response $w||z$ is recorded in $T_{S^*_+}[x,y]$. $Path$ is a (initially everywhere $\perp$) table which records all paths with the Merkle-Damgård style. If triples $(IV, m_1, w_1||z_1), (w_1||z_1, m_2, w_2||z_2), (w_2||z_2, m_3, w_3||z_3)$ are the query-response values where $T_{S^*}[w_{j-1}||z_{j-1}, m_j] = w_j||z_j$ $(j = 1,2,3)$ and $w_0||z_0 = IV$, then $m_1||m_2||m_3$ is recoded in $Path[w_3||z_3]$. C is a (initially empty) set. They show that the advantage p^* of the indifferentiable security is bounded by $((3n+1)q_h + nq_H)/2^s + (q_H + q_h)/2^{n-1} + (lq_H + q_h)^2/2^{s+n+1}$.

We evaluate the difference $\Pr[G1] - \Pr[G2]$ where a distinguisher A_1 interacts with $(\mathcal{RO}_n, S)$ in Game 1 and $(\mathcal{RO}_n, S^*)$ in Game 2. We consider the six games Game C0, Game C1, Game C2, Game C3, Game C4, and Game C5. In each game, the distinguisher interacts with $(\mathcal{RO}_n, \mathcal{O})$ where $\mathcal{O}$ in each game is shown in Figs. 11, 12, and 13. Game C0 is equal to Game 1 and Game C5 is equal to Game 2. Let GCj be an event that A_1 output 1 in Game Cj. Thus

$$\begin{aligned}\Pr[G1] - \Pr[G2] &= \Pr[GC0] - \Pr[GC5] \\ &= \sum_{j=0}^{4}(\Pr[GCj] - \Pr[GC(j+1)]).\end{aligned}$$

In the following, we evaluate each difference $\Pr[GCj] - \Pr[GC(j+1)]$.

Game C1. In this game, the procedures of $\mathcal{RO}_s^{(1)}$, $\mathcal{TO}^{(1)}$, and $\mathcal{RO}^1_{n+s}$ are hard-coded in $\mathcal{O}$. The modification from Game C0 to Game C1 does not affect the distinguisher's behavior. Thus $\Pr[GC0] = \Pr[GC1]$. Note that T_{S^*} is a (initially everywhere $\perp$) table and this table does not affect this game.

Game C2. In this game, due to the step 01, the table F_1^* is not redefined, while in Game C1, there are cases that the table is redefined. For Game C1, we consider the following cases which cover all cases that the table is redefined.

- Case 1: First $T_{S^*}[x,m]$ was defined due to a value defined in the steps 01-04 and then the table is redefined due to the other steps.
- Case 2: First $T_{S^*}[x,m]$ was defined due to a value defined in the steps 05-08 and then the table is redefined due to the other steps.

```
$\underline{\mathcal{O}(x,m)}$ where $x = x_1||x_2$ ($|x_1| = s$, $|x_2| = n$)
01 $M \leftarrow \mathcal{TO}^{(1)}(x_1)$;
02 if $x = IV$ then
03     $z \leftarrow \mathcal{RO}_n(m)$;
04     $w \leftarrow \mathcal{RO}_s^{(1)}(m)$;
05 else if $M \neq \perp$ then
06     $z \leftarrow \mathcal{RO}_n(M||m)$;
07     $w \leftarrow \mathcal{RO}_s^{(1)}(M||m)$;
08 else $w||z \leftarrow \mathcal{RO}^1_{n+s}(x,m)$;
09 return $w||z$;
```

```
$\underline{\mathcal{O}(x,m)}$ where $x = x_1||x_2$ ($|x_1| = s$, $|x_2| = n$)
01 if $x = IV$ then
02     $z \leftarrow \mathcal{RO}_n(m)$;
03     if $\mathsf{F}_1^*[m] = \perp$ then $\mathsf{F}_1^*[m] \xleftarrow{\$} \{0,1\}^s$;
04     $w \leftarrow \mathsf{F}_1^*[m]$;
05 else if $\exists_1 M$ s.t. $\mathsf{F}_1^*[M] = x_1$ then
06     $z \leftarrow \mathcal{RO}_n(M||m)$;
07     if $\mathsf{F}_1^*[M||m] = \perp$ then $\mathsf{F}_1^*[M||m] \xleftarrow{\$} \{0,1\}^s$;
08     $w \leftarrow \mathsf{F}_1^*[M||m]$;
09 else
10     if $\mathsf{F}_1[x,m] = \perp$ then $\mathsf{F}_1[x,m] \xleftarrow{\$} \{0,1\}^{s+n}$;
11     $w||z \leftarrow \mathsf{F}_1[x,m]$;
12 $T_{S^*}[x,m] \leftarrow w||z$;
13 return $w||z$;
```

Fig. 11. Game C0 (left) and Game C1 (right)

```
$\underline{\mathcal{O}(x,m)}$ where $x = x_1||x_2$ ($|x_1| = s$, $|x_2| = n$)
01 if $T_{S^*}[x,m] \neq \perp$ then return $T_{S^*}[x,m]$;
02 if $x = IV$ then
03     $z \leftarrow \mathcal{RO}_n(m)$;
04     if $\mathsf{F}_1^*[m] = \perp$ then $\mathsf{F}_1^*[m] \xleftarrow{\$} \{0,1\}^s$;
05     $w \leftarrow \mathsf{F}_1^*[m]$;
06 else if $\exists_1 M$ s.t. $\mathsf{F}_1^*[M] = x_1$ then
07     $z \leftarrow \mathcal{RO}_n(M||m)$;
08     if $\mathsf{F}_1^*[M||m] = \perp$ then $\mathsf{F}_1^*[M||m] \xleftarrow{\$} \{0,1\}^s$;
09     $w \leftarrow \mathsf{F}_1^*[M||m]$;
10 else
11     if $\mathsf{F}_1[x,m] = \perp$ then $\mathsf{F}_1[x,m] \xleftarrow{\$} \{0,1\}^{s+n}$;
12     $w||z \leftarrow \mathsf{F}_1[x,m]$;
13 $T_{S^*}[x,m] \leftarrow w||z$;
14 return $w||z$;
```

```
$\underline{\mathcal{O}(x,m)}$
01 if $T_{S^*}[x,m] \neq \perp$ then return $T_{S^*}[x,m]$;
02 if $x = IV$ then
03     $z \leftarrow \mathcal{RO}_n(m)$;
04     if $\mathsf{F}_1^*[m] = \perp$ then $\mathsf{F}_1^*[m] \xleftarrow{\$} \{0,1\}^s$;
05     $w \leftarrow \mathsf{F}_1^*[m]$;
06     $Path[w||z] \leftarrow m$;
07 else if $Path[x] = M \neq \perp$ then
08     $z \leftarrow \mathcal{RO}_n(M||m)$;
09     if $\mathsf{F}_1^*[M||m] = \perp$ then $\mathsf{F}_1^*[M||m] \xleftarrow{\$} \{0,1\}^s$;
10     $w \leftarrow \mathsf{F}_1^*[M||m]$;
11     $Path[w||z] \leftarrow M||m$;
12 else
13     if $\mathsf{F}_1[x,m] = \perp$ then $\mathsf{F}_1[x,m] \xleftarrow{\$} \{0,1\}^{s+n}$;
14     $w||z \leftarrow \mathsf{F}_1[x,m]$;
15 $T_{S^*}[x,m] \leftarrow w||z$;
16 return $w||z$;
```

Fig. 12. Game C2 (left) and Game C3 (right)

- Case 3: First $T_{S^*}[x,m]$ was defined due to a value defined in the steps 09-11 and then the table is redefined due to the other steps.

Consider Case 1. In this case, $x = IV$. Since for any m $T_{S^*}[IV,m]$ is defined due to a value defined in the steps 01-04, this case does not occur.
Consider Case 2. In this case, $x \neq IV$. So the table is redefined due to the steps 09-11. When $T_{S^*}[x,m]$ was defined, there was a value M such that $\mathsf{F}_1^*[M] = x_1$. When the table is redefined, there is another value M' such that $\mathsf{F}_1^*[M'] = x_1$. Thus a collision for F_1^* occurs and the collision probability is bounded by $q_h^2/2^{s+1}$ from a birthday analysis.
Consider Case 3. In this case $x \neq IV$. So $T_{S^*}[x,m]$ is not redefined due to a value defined in the steps 01-04, and thus it is redefined due to a value defined in the steps 05-08. When $T_{S^*}[x,m]$ was defined, there was not M such that $\mathsf{F}_1^*[M] = x_1$ and when it is redefined, there is such M. Thus this case is that $\mathsf{F}_1[x_1,m_1] = w_1||z_1$ was defined and then $\mathsf{F}_1^*[M] = x_1$ which is uniformly chosen at random from $\{0,1\}^s$ is defined. The probability for Case 3 is bounded by $q_h^2/2^s$ since the number of queries to S^* is at most q_h.

Thus

$$\Pr[GC1] - \Pr[GC2] \leq \frac{3q_h^2}{2^{s+1}}$$

```
$\underline{\mathcal{O}(x,m)}$
01 if $T_{S^*}[x,m] \neq \perp$ then **return** $T_{S^*}[x,m]$;
02 if $x = IV$ then
03     $z \leftarrow \mathcal{RO}_n(m)$;
04     $w \xleftarrow{\$} \{0,1\}^s$;
05     $Path[w||z] \leftarrow m$;
06 else if $Path[x] = M \neq \perp$ then
07     $z \leftarrow \mathcal{RO}_n(M||m)$;
08     $w \xleftarrow{\$} \{0,1\}^s$;
09     $Path[w||z] \leftarrow M||m$;
10 else $w||z \xleftarrow{\$} \{0,1\}^{s+n}$;
11 $T_{S^*}[x,m] \leftarrow w||z$;
12 **return** $w||z$;
```

```
$\underline{\mathcal{O}(x,m)}$
01 if $T_{S^*}[x,m] \neq \perp$ then **return** $T_{S^*}[x,m]$;
02 if $x = IV$ then
03     $z \leftarrow \mathcal{RO}_n(m)$;
04     $w \xleftarrow{\$} \{0,1\}^s \backslash \{w' : w'||z \in C \cup \{x\}\}$;
05     $Path[w||z] \leftarrow m$;
06 else if $Path[x] = M \neq \perp$ then
07     $z \leftarrow \mathcal{RO}_n(M||m)$;
08     $w \xleftarrow{\$} \{0,1\}^s \backslash \{w' : w'||z \in C \cup \{x\}\}$;
09     $Path[w||z] \leftarrow M||m$;
10 else
11     $z \xleftarrow{\$} \{0,1\}^n$;
12     $w \xleftarrow{\$} \{0,1\}^s \backslash \{w' : w'||z \in C \cup \{x\}\}$;
13 $T_{S^*}[x,m] \leftarrow w||z$; $C \xleftarrow{\cup} \{x,z\}$;
14 **return** $w||z$;
```

Fig. 13. Game C4 (left) and Game C5 (right)

Game C3. In this game, a new table $Path$ is used which is initially everywhere $\perp$ and recodes paths with Merkle-Damgård style. In the step 07, $Path$ is used in this game, while F_1^* is used in Game C2. Note that if $Path[x] = M$ and no collision occurs for F_1^* then $\mathsf{F}_1^*[M] = x_1$ where $x = x_1||x_2$ and $|x_1| = s$, and if $\mathsf{F}_1^*[M] = x_1$ and no collision occurs for F_1^* then $Path[x] = M$. Thus if no collision occurs for F_1^* in Game C3 then this modification does not affect the distinguisher's behavior. We thus have via a birthday analysis that

$$\Pr[GC2] - \Pr[GC3] \leq \frac{q_h^2}{2^{s+1}}.$$

Game C4. In this game, tables F_1 and F_1^* are removed. In Game C3, for a pair (x,m), if $x = IV$ then $T_{S^*}[x,m] = \mathsf{F}_1^*[m]||\mathcal{RO}_n(m)$, and if there exists a value M such that $\mathsf{F}_1^*[M] = x_1$ then $T_{S^*}[x,m] = \mathsf{F}_1^*[M||m]||\mathcal{RO}_n(M||m)$, and $T_{S^*}[x,m] = \mathsf{F}_1[x,m]$ otherwise. Thus due to the step 01, tables F_1 and F_1^* are not used and this modification does not affect the distinguisher's behavior. So $\Pr[GC3] = \Pr[GC4]$.

Game C5. In this game, for a query $\mathcal{O}(x,m)$, w is uniformly chosen at random from $\{0,1\}^n \backslash \{w' : w' \in C \cup \{x\}\}$, while in Game C4 it is uniformly chosen at random from $\{0,1\}^n$. Thus if in Game C4 w does not collide with one of $\{w' : w' \in C \cup \{x\}\}$ then this modification does not affect the distinguisher's behavior. The number of elements in $\{w' : w' \in C \cup \{x\}\}$ is at most $2q_h + 1$. We thus have that

$$\Pr[GC4] - \Pr[GC5] \leq \frac{(2q_h+1)^2}{2^n}.$$

```
$\underline{S^*_+(x,y)}$
01 if $T_{S^*_+}[x,y] \neq \perp$ then **return** $T_{S^*_+}[x,y]$;
02 if $y = IV_2$ then
03     if $\mathsf{unpad}_S(IV_1 \oplus x) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else $z \xleftarrow{\$} \{0,1\}^n$;
05 else if $Path[y] \neq \perp$ then
06     let $Path[y] = (M^*, z^*)$;
07     if $\mathsf{unpad}_S(M^*||(z^* \oplus x)) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
08     else $z \xleftarrow{\$} \{0,1\}^n$;
09 else $z \xleftarrow{\$} \{0,1\}^n$;
10 $w \xleftarrow{\$} \{0,1\}^c \backslash T_F[z]$;
11 $Update_{S^*}(x,y,z,w)$;
12 **return** $T_{S^*_+}[x,y]$;

$\underline{S^*_-(z,w)}$
1 if $T_{S^*_-}[z,w] \neq \perp$ then $T_{S^*_-}[z,w]$;
2 $x \xleftarrow{\$} \{0,1\}^n$; $y \xleftarrow{\$} \{0,1\}^c \backslash T_I[x]$;
3 $Update_{S^*}(x,y,z,w)$;
4 **return** $T_{S^*_-}[z,w]$;
```

Fig. 14. Simulator S^*

C Proof of Theorem 3

We prove Theorem 3 by using the strategy in Appendix **??**.

Since the simulator S does not update the internal state, $\Pr[G0] - \Pr[G1] = 0$.

We use the result of the indifferentiable security by Bertoni *et al.* [11]. They define a simulator $S^* = (S^*_+, S^*_-)$ which is shown in Fig. 14. S^*_+ and S^*_- simulate the random permutation P and its inverse P^{-1}, respectively. $T_{S^*_+}$ and $T_{S^*_-}$ are (initially everywhere $\perp$) tables which records query-response values of S^*_+ and S^*_-. For the query $S^*_+(x,y)$, the response $z||w$ is recorded in $T_{S^*_+}[x,y]$ and $x||y$ is recoded in $T_{S^*_-}[z,w]$. Similarly, the response $z||w$ of the query $S^*_-(x,y)$ and $x||y$ are recoded in these tables. $Path$ is a (initially everywhere $\perp$) table which records all paths with the Sponge style. If triples $(x_1,w_0,z_1,w_1),(x_2,w_1,z_2,w_2),(x_3,w_2,z_3,w_3)$ are the query-response values where $T_{S^*_+}[x_j,w_{j-1}] = z_j||w_j$ $(j=1,2,3)$ and $w_0 = IV_2$, then $(x_1 \oplus IV_1)||(x_2 \oplus z_1)||(x_3 \oplus z_2)$ and z_3 is recoded in $Path[w_3]$. T_F and T_I are (initially everywhere empty) tables. $T_F[z]$ includes values which are all y' such that $T_{S^*_+}[\cdot,y'] \neq \perp$, IV_2, all y'' such that $Path[y''] \neq \perp$, and all w' such that $T_{S^*_-}[z,w'] \neq \perp$. $T_I[x]$ includes values which are IV_2, all y' such that $Path[y'] \neq \perp$, and all y'' such that $T_{S^*_+}[x,y''] \neq \perp$. $Update_{S^*}(x,y,z,w)$ is a procedure that the tables $T_{S^*_+}, T_{S^*_-}$, and $Path$ are updated by using (x,y,z,w), namely, $T_{S^*_+}[x,y] \leftarrow z||w$, $T_{S^*_-}[z,w] \leftarrow x||y$, and if $Path[y] = (M,z^*) \neq \perp$ then $Path[w] \leftarrow (M||(x \oplus z^*),z)$ [5]. Notice that the tables T_F and T_I are used to avoid that a multi-path is defined in $Path$. They show that the advantage p^* of the indifferentiable security is bounded by $((1-2^{-n})q^2 + (1+2^{-n})q)/2^{c+1}$.

We evaluate the difference $\Pr[G1] - \Pr[G2]$ where a distinguisher A_1 interacts with $(\mathcal{RO}_n, S)$ in Game 1 and $(\mathcal{RO}_n, S^*)$ in Game 2.

We consider the seven games Game S1, Game S2, Game S3, Game S4, Game S5, Game S6, and Game S7. In each game, the distinguisher interacts with $(\mathcal{RO}_n, \mathcal{O}_+, \mathcal{O}_-)$ where $(\mathcal{O}_+, \mathcal{O}_-)$ is shown in Figs. 15, 16, 17 18, 19, 20, and 21. Game S1 is equal to Game 1 and Game S7 is equal to Game 2. Let GSj be an event that A_1 output 1 in Game Sj. Thus

$$\begin{aligned}\Pr[G1] - \Pr[G2] &= \Pr[GS1] - \Pr[GS7] \\ &= \sum_{j=1}^{6}(\Pr[GSj] - \Pr[GS(j+1)]).\end{aligned}$$

In the following, we evaluate each difference $\Pr[GSj] - \Pr[GS(j+1)]$.

Game S2. In this game, a random permutation $(\mathcal{P}, \mathcal{P}^{-1})$ is replaced with a new function $(\mathcal{P}_1, \mathcal{P}_1^{-1})$. F^+

[5] Note that in [11], the paths and the query-response values are recorded by using a graph representation, but the table $Path$ and the tables $T_{S^*_+}, T_{S^*_-}$ realizes the same role as the graph.

```
$\underline{\mathcal{O}_+(x,y)}$
01 $M^* \leftarrow \mathcal{TO}^{(1)}(y)$;
02 if $y = IV_2$ then
03     if $\mathsf{unpad}_S(IV_1 \oplus x) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else $z \leftarrow \mathcal{RO}_n^1(IV_1 \oplus x)$;
05     $w \leftarrow \mathcal{RO}_c^{(1)}(x \oplus IV_1)$;
06 else if $M^* \neq \perp$ then
07     if $\mathsf{unpad}(M^*) = M' \neq \perp$ then $m \leftarrow x \oplus \mathcal{RO}_n(M')$;
08     else $m \leftarrow x \oplus \mathcal{RO}_n^1(M^*)$;
09     if $\mathsf{unpad}_S(M^*||m) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
10     else $z \leftarrow \mathcal{RO}_n^1(M^*||m)$;
11     $w \leftarrow \mathcal{RO}_c^{(1)}(M^*||m)$;
12 else $z||w \leftarrow \mathcal{P}(x||y)$;
13 **return** $z||w$;

$\underline{\mathcal{O}_-(z,w)}$
01 $M \leftarrow \mathcal{TO}^{(1)}(w)$;
02 if $M \neq \perp$ and $|M| = n$ then
03     $x \leftarrow IV_1 \oplus M$; $y \leftarrow IV_2$;
04 if $M \neq \perp$ and $|M| > n$ then
05     let $M = M^*||m$ ($|m| = n$);
06     if $\mathsf{unpad}_S(M^*) = M' \neq \perp$ then $x \leftarrow m \oplus \mathcal{RO}_n(M')$;
07     else $x \leftarrow m \oplus \mathcal{RO}_n^1(M^*)$;
08     $y \leftarrow \mathcal{RO}_c^{(1)}(M^*)$;
09 else $x||y \leftarrow \mathcal{P}^{-1}(z||w)$;
10 **return** $x||y$;
```

Fig. 15. Game S1

```
$\underline{\mathcal{O}_+(x,y)}$
01 $M^* \leftarrow \mathcal{TO}^{(1)}(y)$;
02 if $y = IV_2$ then
03     if $\mathsf{unpad}_S(IV_1 \oplus x) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else $z \leftarrow \mathcal{RO}_n^1(IV_1 \oplus x)$;
05     $w \leftarrow \mathcal{RO}_c^{(1)}(x \oplus IV_1)$;
06 else if $M^* \neq \perp$ then
07     if $\mathsf{unpad}(M^*) = M' \neq \perp$ then $m \leftarrow x \oplus \mathcal{RO}_n(M')$;
08     else $m \leftarrow x \oplus \mathcal{RO}_n^1(M^*)$;
09     if $\mathsf{unpad}_S(M^*||m) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
10     else $z \leftarrow \mathcal{RO}_n^1(M^*||m)$;
11     $w \leftarrow \mathcal{RO}_c^{(1)}(M^*||m)$;
12 else $z||w \leftarrow \mathcal{P}_1(x||y)$;
13 $Update_{S^*}(x,y,z,w)$;
14 **return** $T_{S^*_+}[x,y]$;

$\underline{\mathcal{P}_1(x)}$
1 if $\mathsf{F}^+[x] = \perp$, ret $\mathsf{F}^+[x]$;
2 $y \xleftarrow{\$} \{0,1\}^b$;
3 $Update_P(x,y)$;
4 **return** $\mathsf{F}^+[x]$

$\underline{\mathcal{O}_-(z,w)}$
01 $M \leftarrow \mathcal{TO}^{(1)}(w)$;
02 if $M \neq \perp$ and $|M| = n$ then
03     $x \leftarrow IV_1 \oplus M$; $y \leftarrow IV_2$;
04 else if $M \neq \perp$ and $|M| > n$ then
05     let $M = M^*||m$ ($|m| = n$);
06     if $\mathsf{unpad}_S(M^*) = M' \neq \perp$ then $x \leftarrow m \oplus \mathcal{RO}_n(M')$;
07     else $x \leftarrow m \oplus \mathcal{RO}_n^1(M^*)$;
08     $y \leftarrow \mathcal{RO}_c^{(1)}(M^*)$;
09 else $x||y \leftarrow \mathcal{P}_1^{-1}(z||w)$;
10 $Update_{S^*}(x,y,z,w)$;
11 **return** $T_{S^*_-}[z,w]$;

$\underline{\mathcal{P}_1^{-1}(y)}$
1 if $\mathsf{F}^-[y] = \perp$, ret $\mathsf{F}^-[y]$;
2 $x \xleftarrow{\$} \{0,1\}^b$;
3 $Update_P(x,y)$;
4 **return** $\mathsf{F}^-[y]$;
```

Fig. 16. Game S2

and F^- are (initially everywhere $\perp$) tables. $Update_P(x,y)$ updates the tables F^+ and F^-: $\mathsf{F}^+[x] \leftarrow y$ and $\mathsf{F}^-[y] \leftarrow x$. An output of $(\mathcal{P}_1, \mathcal{P}_1^{-1})$ is uniformly chosen at random from $\{0,1\}^b$. Thus if in Game GS2 no collision occurs for the outputs of $\mathcal{P}_1$ and the outputs of $\mathcal{P}_1^{-1}$, the modification from Game GS1 to Game GS1 does not affect the distinguisher's behavior. Since the number of the outputs of $\mathcal{P}_1$ is at most q, the probability that a collision occurs for the outputs of $\mathcal{P}_1$ is at most $q^2/2^b$. Similarly, the probability that a collision occurs for the outputs of $\mathcal{P}_1^{-1}$ is at most $q^2/2^b$. We thus have that

$$\Pr[GS1] - \Pr[GS2] \leq \frac{q^2}{2^{b-1}}.$$

Note that the procedure $Update_{S^*}$ updates tables $T^+_{S^*}$, $T^-_{S^*}$, and $Path$.

Game S3. In Game S3, the step 01 of $\mathcal{O}_+$ and the step 01 of $\mathcal{O}_-$ are new steps. The tables $T_{S^*_+}$ and $T_{S^*_-}$ are not redefined if no collision for the outputs of $\mathcal{O}_+$ or the outputs of $\mathcal{O}_-$ occurs. In Game S2, we show that if the following events don't occur then the tables $T_{S^*_+}$ and $T_{S^*_-}$ are not redefined.

```
$\underline{\mathcal{O}_+(x,y)}$
01 if $T_{S_+^*}[x,y] \neq\perp$ then **return** $T_{S_+^*}[x,y]$;
02 $M^* \leftarrow \mathcal{TO}^{(1)}(y)$;
03 if $y = IV_2$ then
04     if $\mathsf{unpad}_S(IV_1 \oplus x) = M \neq\perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
05     else $z \leftarrow \mathcal{RO}_n^1(IV_1 \oplus x)$;
06     $w \leftarrow \mathcal{RO}_c^{(1)}(x \oplus IV_1)$;
07 else if $M^* \neq\perp$ then
08     if $\mathsf{unpad}(M^*) = M' \neq\perp$ then $m \leftarrow x \oplus \mathcal{RO}_n(M')$;
09     else $m \leftarrow x \oplus \mathcal{RO}_n^1(M^*)$;
10     if $\mathsf{unpad}_S(M^*||m) = M \neq\perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
11     else $z \leftarrow \mathcal{RO}_n^1(M^*||m)$;
12     $w \leftarrow \mathcal{RO}_c^{(1)}(M^*||m)$;
13 else $z||w \leftarrow \mathcal{P}_1(x||y)$;
14 $Update_{S^*}(x,y,z,w)$;
15 **return** $T_{S_+^*}[x,y]$;

$\underline{\mathcal{O}_-(z,w)}$
01 if $T_{S_-^*}[z,w] \neq\perp$ then $T_{S_-^*}[z,w]$;
02 $M \leftarrow \mathcal{TO}^{(1)}(w)$;
03 if $M \neq\perp$ and $|M| = n$ then
04     $x \leftarrow IV_1 \oplus M$; $y \leftarrow IV_2$;
05 else if $M \neq\perp$ and $|M| > n$ then
06     let $M = M^*||m$ ($|m| = n$);
07     if $\mathsf{unpad}_S(M^*) = M_1 \neq\perp$ then $x \leftarrow m \oplus \mathcal{RO}_n(M_1)$;
08     else $x \leftarrow m \oplus \mathcal{RO}_n^1(M^*)$;
09     $y \leftarrow \mathcal{RO}_c^{(1)}(M^*)$;
10 else $x||y \leftarrow \mathcal{P}_1^{-1}(z||w)$;
11 $Update_{S^*}(x,y,z,w)$;
12 **return** $T_{S_-^*}[z,w]$;
```

Fig. 17. Game S3

- Event 1: A collision for $\mathcal{RO}_c^{(1)}$ occurs.
- Event 2: A random value of c bits hits a value in at most q values.
- Event 3: A random value of c bits hits IV_2.

One of the following cases occurs when $T_{S_+^*}$ or $T_{S_-^*}$ are redefined.

- Case 1: First, for query $\mathcal{O}_+(x,y)$, $T_{S_+^*}[x,y]$ and $T_{S_-^*}[z,w]$ were defined due to the steps 02-05 where $T_{S_+^*}[x,y] = z||w$ and $T_{S_-^*}[z,w] = x||y$, and then $T_{S_+^*}[x,y]$ or $T_{S_-^*}[z,w]$ is redefined by a new value due to other steps.
  In this case $y = IV_2$. So the redefined steps are not the steps 06-11 and 12 of $\mathcal{O}_+$. And when $T_{S_+^*}[x,y]$ and $T_{S_-^*}[z,w]$ were defined, $\mathsf{F}_1^*[y] = z$ where $|y| = n$. So the redefined steps are not the steps 02-03, and 04-08 of $\mathcal{O}_-$. That is, the redefined step is just the step 09 of $\mathcal{O}_-$.
- Case 2: First, for query $\mathcal{O}_+(x,y)$, $T_{S_+^*}[x,y]$ and $T_{S_-^*}[z,w]$ were defined due to the steps 06-11 where $T_{S_+^*}[x,y] = z||w$ and $T_{S_-^*}[z,w] = x||y$, and then $T_{S_+^*}[x,y]$ or $T_{S_-^*}[z,w]$ is redefined by a new value due to other steps.
  In this case, $y \neq IV_2$. So the redefined steps are not the steps 02-05 of $\mathcal{O}_+$. And after $T_{S_+^*}[x,y]$ defined, there is a value M such that $\mathsf{F}_1^*[M] = w$ and $|M| > n$. So the redefined steps are not the steps 02-03, and 04-08 of $\mathcal{O}_-$. That is, the redefined steps are the step 12 of $\mathcal{O}_+$ or the step 09 of $\mathcal{O}_-$.
- Case 3: First, for query $\mathcal{O}_+(x,y)$, $T_{S_+^*}[x,y]$ and $T_{S_-^*}[z,w]$ were defined due to the step 12 where $T_{S_+^*}[x,y] = z||w$ and $T_{S_-^*}[z,w] = x||y$, and then $T_{S_+^*}[x,y]$ or $T_{S_-^*}[z,w]$ is redefined by a new value due to other steps.
  In this case, $y \neq IV_2$. So the redefined steps are not the steps 02-05, that is the steps are the steps 06-11 of $\mathcal{O}_+$, the steps 02-03, 04-08, or 09 of $\mathcal{O}_-$.
- Case 4: First, for query $\mathcal{O}_-(z,w)$, $T_{S_+^*}[x,y]$ and $T_{S_-^*}[z,w]$ were defined due to the steps 02-03 where $T_{S_+^*}[x,y] = z||w$ and $T_{S_-^*}[z,w] = x||y$, and then $T_{S_+^*}[x,y]$ or $T_{S_-^*}[z,w]$ is redefined by a new value in other steps.
  In this case, when the steps 02-03 were executed, $\mathsf{F}_1^*[M]$ ($= w$) had been already defined. The defined steps are the steps 02-05 of $\mathcal{O}_+$. Thus $T_{S_+^*}[x,y]$ and $T_{S_-^*}[z,w]$ were already defined in the steps 02-05 of $\mathcal{O}_+$. So this case does not occur.
- Case 5: First, for query $\mathcal{O}_-(z,w)$, $T_{S_+^*}[x,y]$ and $T_{S_-^*}[z,w]$ were defined due to the steps 04-08 where $T_{S_+^*}[x,y] = z||w$ and $T_{S_-^*}[z,w] = x||y$, and then $T_{S_+^*}[x,y]$ or $T_{S_-^*}[z,w]$ is redefined by a new value in other steps.
  In this case, when the steps 04-08 were executed, $\mathsf{F}_1^*[M]$ ($= w$) had been defined where $|M| > n$. The steps where $\mathsf{F}_1^*[M]$ defined are the steps 06-11. Thus $T_{S_+^*}[x,y]$ and $T_{S_-^*}[z,w]$ were already defined in the steps 06-11 of $\mathcal{O}_+$. So this case does not occur.

– Case 6: First, for query $\mathcal{O}_-(z,w)$, $T_{S^*_+}[x,y]$ and $T_{S^*_-}[z,w]$ were defined due to the step 09 where $T_{S^*_+}[x,y] = z||w$ and $T_{S^*_-}[z,w] = x||y$, and then $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined by a new value in other steps.

We show that in each case some of events occurs.

First, consider Case 1. When $T_{S^*_+}[x,y]$ and $T_{S^*_-}[z,w]$ were defined, $\mathsf{F}^*_1[y] = z$ where $|y| = n$. However, when $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined, $\mathcal{TO}^{(1)}(w) = \perp$. Thus in this case, a collision for $\mathcal{RO}^{(1)}_c$ occurs, that is Event 1 occurs.

Consider Case 2. When $T_{S^*_+}[x,y]$ and $T_{S^*_-}[z,w]$ were defined, $\mathsf{F}^*_1[M^*]$ $(= y)$ had been defined and $\mathsf{F}^*_1[M]$ $(= w)$ was defined where $|M| > n$. However, when $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined due to the step 12 of $\mathcal{O}_+$, $\mathcal{TO}^{(1)}(y) = \perp$, and when $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined due to the step 09 of $\mathcal{O}_-$, $\mathcal{TO}^{(1)}(w) = \perp$. Thus in this case, a collision for $\mathcal{RO}^{(1)}_c$ occurs, that is Event 1 occurs.

Consider Case 3. When $T_{S^*_+}[x,y]$ and $T_{S^*_-}[z,w]$ were defined, there did not exist M^* such that $\mathsf{F}^*_1[M^*] = y$.
(Case 3-1) $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined in the steps 06-11 of $\mathcal{O}_+$. In this case, there is a value M^* such that $\mathsf{F}^*_1[M^*] = y$. Thus in this case, first y was defined, and then a random value hits y. Since the number of such y is at most q, when Case 3-1 occurs then Event 2 occurs.
(Case 3-2) $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined in the steps 02-03 of $\mathcal{O}_-$. In this case, there is a value M such that $\mathsf{F}^*_1[M] = w$ and $|M| = n$. $\mathsf{F}^*_1[M]$ $(= w)$ was defined in the steps 02-05 of $\mathcal{O}_+$. This implies that $y = IV_2$. This contradicts that $y \neq IV_2$ and thus this case does not occur.
(Case 3-3) $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined in the steps 04-08 of $\mathcal{O}_-$. In this case, there is a value M such that $\mathsf{F}^*_1[M] = w$ and $|M| > n$. $\mathsf{F}^*_1[M]$ $(= w)$ was defined in the steps 06-11 and thus $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ was redefined the steps 06-11. Thus this case does not occur.
(Case 3-4) $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined in the step 09 of $\mathcal{O}_-$. Since an output of $\mathcal{P}_1^{-1}$ is a random value of b bits, this case is that the random value hits $x||y$. Since the number of such y is at most q, when Case 3-4 occurs then Event 2 occurs.

Consider Case 6. When $T_{S^*_+}[x,y]$ and $T_{S^*_-}[z,w]$ were defined, there does not exist M such that $\mathsf{F}^*_1[M] = w$.
(Case 6-1) $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined due to the steps 02-05 of $\mathcal{O}_+$. In this case, $y = IV_2$. Since an output of $\mathcal{P}_1^{-1}$ is a random value of b bits, this case is that the left c bit value of the output hits IV_2, which is Event 3.
(Case 6-2) $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined due to the steps 06-11 of $\mathcal{O}_+$. In this case, there is a value M^* such that $\mathsf{F}^*_1[M^*] = y$. Note that $\mathsf{F}^*_1[M^*]$ $(= y)$ and $T_{S^*_-}[z,w]$ $(= x||y)$ are independent random values. Thus this case is that a random value of c bits hits one of q values, which is equal to Event 2.
(Case 6-3) $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined due to the step 12 of $\mathcal{O}_+$. Since an output of $\mathcal{P}_1$ is a random value of b bits, this case is that a random value of b bits hits $z||w$. So Event 2 occurs.
(Case 6-4) $T_{S^*_+}[x,y]$ or $T_{S^*_-}[z,w]$ is redefined due to the steps 02-03 or 04-08 of $\mathcal{O}_-$. In this case, there is a value M such that $\mathsf{F}^*_1[M] = w$. Thus this case is that first w is defined and then a random value of c bits hits w. This is equal to Event 2.

Thus if the following events don't occur then the modification from Game S2 to Game S3 does not affect the distinguisher's behavior. The first three events are due to Game S2 and the last two events are due to Game S3.

– Event 1: A collision for $\mathcal{RO}^{(1)}_c$ occurs.
– Event 2: A random value of c bits hits a value in at most q values.
– Event 3: A random value of c bits hits IV_2.
– Event 4: A collision for the output of $\mathcal{O}_+$ occurs.
– Event 5: A collision for the output of $\mathcal{O}_-$ occurs.

From a birthday analysis, the probability that Event 1 occurs is at most $q^2/2^{c+1}$. For Event 2, the number of a random value is at most q. So the probability that Event 2 occurs is at most $q^2/2^c$. For Event 3, the number of a random value is at most q. So the probability that Event 3 occurs is at most $q/2^c$. For Event 4, since the number of outputs of $\mathcal{O}_+$ is at most q, this event is that a random value of b bits collides with some of the q values. Since the number of the random values is at most q, the probability that Event 4 occurs is

```
$\mathcal{O}_+(x,y)$
01 if $T_{S_+^*}[x,y] \neq \perp$ then **return** $T_{S_+^*}[x,y]$;
02 $M^* \leftarrow \mathcal{TO}^{(1)}(y)$;
03 if $y = IV_2$ then
04     if $\mathsf{unpad}_S(IV_1 \oplus x) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
05     else $z \leftarrow \mathcal{RO}_n^1(IV_1 \oplus x)$;
06     $w \leftarrow \mathcal{RO}_c^{(1)}(x \oplus IV_1)$;
07 else if $M^* \neq \perp$ then
08     if $\mathsf{unpad}(M^*) = M' \neq \perp$ then $m \leftarrow x \oplus \mathcal{RO}_n(M')$;
09     else $m \leftarrow x \oplus \mathcal{RO}_n^1(M^*)$;
10     if $\mathsf{unpad}_S(M^*||m) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
11     else $z \leftarrow \mathcal{RO}_n^1(M^*||m)$;
12     $w \leftarrow \mathcal{RO}_c^{(1)}(M^*||m)$;
13 else $z||w \leftarrow \mathcal{P}_1(x||y)$;
14 $Update_{S^*}(x,y,z,w)$;
15 **return** $T_{S_+^*}[x,y]$;
```

```
$\mathcal{O}_-(z,w)$
1 if $T_{S_-^*}[z,w] \neq \perp$ then $T_{S_-^*}[z,w]$;
2 $x||y \leftarrow \mathcal{P}_1^{-1}(z||w)$;
3 $Update_{S^*}(x,y,z,w)$;
4 **return** $T_{S_-^*}[z,w]$;
```

Fig. 18. Game S4

```
$\mathcal{O}_+(x,y)$
01 if $T_{S_+^*}[x,y] \neq \perp$ then **return** $T_{S_+^*}[x,y]$;
02 if $y = IV_2$ then
03     if $\mathsf{unpad}_S(IV_1 \oplus x) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else $z \leftarrow \mathcal{RO}_n^1(IV_1 \oplus x)$;
05     $w \leftarrow \mathcal{RO}_c^{(1)}(x \oplus IV_1)$;
06 else if $Path[y] = (M^*, z^*) \neq \perp$ then
07     if $\mathsf{unpad}(M^*) = M' \neq \perp$ then $m \leftarrow x \oplus \mathcal{RO}_n(M')$;
08     else $m \leftarrow x \oplus \mathcal{RO}_n^1(M^*)$;
09     if $\mathsf{unpad}_S(M^*||m) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
10     else $z \leftarrow \mathcal{RO}_n^1(M^*||m)$;
11     $w \leftarrow \mathcal{RO}_c^{(1)}(M^*||m)$;
12 else $z||w \leftarrow \mathcal{P}_1(x||y)$;
13 $Update_{S^*}(x,y,z,w)$;
14 **return** $z||w$;
```

```
$\mathcal{O}_-(z,w)$
1 if $T_{S_-^*}[z,w] \neq \perp$ then $T_{S_-^*}[z,w]$;
2 $x||y \leftarrow \mathcal{P}_1^{-1}(z||w)$;
3 $Update_{S^*}(x,y,z,w)$;
4 **return** $x||y$;
```

Fig. 19. Game S5

at most $q^2/2^b$. The probability that Event 5 occurs is the same as that for Event 4. We thus have that

$$\Pr[GS2] - \Pr[GS3] \leq \frac{3q^2+2q}{2^{c+1}} + \frac{q^2}{2^{b-1}}.$$

Game S4. In this game, the steps 02-09 of $\mathcal{O}_-$ are removed. Since for the query $\mathcal{O}_-(z,w)$, "M $(= \mathcal{TO}^{(1)}(w)) \neq \perp$" implies that the query $\mathcal{RO}_c^{(1)}(M)$ was made by the query $\mathcal{O}_+(x,y)$ and thus when the query $\mathcal{O}_-(z,w)$ is made, the response $T_{S_-^*}[z,w]$ $(= x||y)$ has been defined. So the steps 02-09 are not executed. Note that if no collision for the outputs of $\mathcal{O}_+$ and the outputs of $\mathcal{O}_-$ occurs, then the table $T_{S_+^*}$ and $T_{S_-^*}$ are not redefined. Thus the modification does not affect the distinguisher's behavior if no collision occurs. For the collision event for the outputs of $\mathcal{O}_+$, since the number of the outputs is at most q, the probability is at most $q^2/2^b$. Similarly, the probability that a collision for the outputs of $\mathcal{O}_-$ occurs is at most $q^2/2^b$. We thus have that

$$\Pr[GS3] - \Pr[GS4] \leq \frac{q^2}{2^{b-1}}.$$

Game S5. In this game, the table $Path$ is used instead of $\mathcal{TO}^{(1)}$. In Game S5, if M^* $(= \mathcal{TO}^{(1)}(y)) \neq \perp$, then

```
$\underline{\mathcal{O}_+(x,y)}$
01 if $T_{S^*_+}[x,y] \neq \perp$ then **return** $T_{S^*_+}[x,y]$;
02 if $y = IV_2$ then
03     if $\mathsf{unpad}_S(IV_1 \oplus x) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else $z \xleftarrow{\$} \{0,1\}^n$;
05 else if $Path[y] = (M^*, z^*) \neq \perp$ then
06     if $\mathsf{unpad}_S(M^* || (z^* \oplus x)) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
07     else $z \xleftarrow{\$} \{0,1\}^n$;
08 else $z \xleftarrow{\$} \{0,1\}^n$;
09 $w \xleftarrow{\$} \{0,1\}^c$;
10 $Update_{S^*}(x,y,z,w)$;
11 **return** $z||w$;

$\underline{\mathcal{O}_-(z,w)}$
1 if $T_{S^*_-}[z,w] \neq \perp$ then $T_{S^*_-}[z,w]$;
2 $x \xleftarrow{\$} \{0,1\}^n$; $y \xleftarrow{\$} \{0,1\}^c$;
3 $Update_{S^*}(x,y,z,w)$;
4 **return** $x||y$;
```

Fig. 20. Game S6

```
$\underline{\mathcal{O}_+(x,y)}$
01 if $T_{S^*_+}[x,y] \neq \perp$ then **return** $T_{S^*_+}[x,y]$;
02 if $y = IV_2$ then
03     if $\mathsf{unpad}_S(IV_1 \oplus x) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
04     else $z \xleftarrow{\$} \{0,1\}^n$;
05 else if $Path[y] = (M^*, z^*) \neq \perp$ then
06     if $\mathsf{unpad}_S(M^* || (z^* \oplus x)) = M \neq \perp$ then $z \leftarrow \mathcal{RO}_n(M)$;
07     else $z \xleftarrow{\$} \{0,1\}^n$;
08 else $z \xleftarrow{\$} \{0,1\}^n$;
09 $w \xleftarrow{\$} \{0,1\}^c \backslash T_F[z]$;
10 $Update_{S^*}(x,y,z,w)$;
11 **return** $z||w$;

$\underline{\mathcal{O}_-(z,w)}$
1 if $T_{S^*_-}[z,w] \neq \perp$ then $T_{S^*_-}[z,w]$;
2 $x \xleftarrow{\$} \{0,1\}^n$; $y \xleftarrow{\$} \{0,1\}^c \backslash T_I[x]$;
3 $Update_{S^*}(x,y,z,w)$;
4 **return** $x||y$;
```

Fig. 21. Game S7

$Path[y] = (M^*, z^*)$. And if $Path[y] = (M^*, z^*) \neq \perp$ and no collision of $\mathcal{RO}_c^{(1)}$ occurs, then $M^* \neq \perp$ where $M^* = \mathcal{TO}^{(1)}(y)$. Thus in both games if no collision for $\mathcal{RO}_c^{(1)}$ occurs, then the modification from Game S4 to Game S5 does not affect the distinguisher's behavior. We thus have via a birthday analysis

$$\Pr[GS4] - \Pr[GS5] \leq \frac{q^2}{2^{c+1}}.$$

Game S6. In this game, $\mathcal{RO}_c^{(1)}, \mathcal{RO}_n^1, \mathcal{P}$ and $\mathcal{P}^{-1}$ are removed and z^* is used in the step 06 of $\mathcal{O}_+$. Notice that $z^* = \mathcal{RO}_n(M')$ in the step 05 of $\mathcal{O}_+$ where $\mathsf{unpad}_S(M^*) = M'$. Thus the use of z^* does not affect the distinguisher's behavior. Since outputs of $\mathcal{RO}_c^{(1)}$, $\mathcal{RO}_n^1$, $\mathcal{P}_1$, and $\mathcal{P}_1^{-1}$ are random values, the modification from Game S5 to Game S6 does not affect the distinguisher's behavior, if no collision for the outputs of $\mathcal{O}_+$ and the outputs of $\mathcal{O}_-$ occurs. Notice that if the table $T_{S^*_+}$ and $T_{S^*_-}$ are redefined then the collision occurs. We thus have that

$$\Pr[GS5] - \Pr[GS6] \leq \frac{q^2}{2^{b-1}}.$$

Game S7. In this game the table T_F (step 10 of $\mathcal{O}_+$) and the table T_I (step 2 of $\mathcal{O}_-$). Thus if in Game S6 w does not collide with $T_F[z]$ in $\mathcal{O}_+$ and y does not collide with $T_I[x]$, then Game S7 is equal to Game S6. The number of elements in $T_F[z]$ is at most $3q+1$ and the number of elements in $T_I[x]$ is at most $2q+1$. Thus the collision probabilities for $T_F[z]$ and $T_I[x]$ are $q(3q+1)/2^c$ and $q(2q+1)/2^c$, respectively. We thus have that

$$\Pr[GS6] - \Pr[GS7] \le \frac{q(5q+2)}{2^c}.$$

```
$\underline{S^*_+(k,x)}$
01 if $T^+_{S^*}[k,x] \neq \perp$ then **return** $T^+_{S^*}[k,x]$;
02 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
03 $y \xleftarrow{\$} \{0,1\}^{2n}$;
04 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
05     $P \leftarrow FindPath(z', s, k, t^{(1)}||t^{(3)})$;
06     if $P \neq \emptyset$ then
07         let $P = (M, a)$;
08         $z \leftarrow \mathcal{RO}_n(s, M)$; $y \xleftarrow{\$} \beta^{-1}_{z',s}(z)$;
09     $AddPath(\beta_{z',s}(y), z', s, k, t^{(1)}||t^{(3)})$;
10 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
11 **return** $T^+_{S^*}[k,x]$

$\underline{S^*_-(k,y)}$
01 if $T^-_{S^*}[k,y] \neq \perp$ then **return** $T^-_{S^*}[k,y]$;
02 $z||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \xleftarrow{\$} \{0,1\}^{2n}$;
03 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
04     $AddPath(\beta_{z,s}(y), z, s, k, t^{(1)}||t^{(3)})$;
05 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
06 **return** $T^-_{S^*}[k,y]$;

$\underline{AddPath(z, h, s, m, t)}$
01 for all $(M, a) \in Path[h]$ do
02     $Path[z] \xleftarrow{\cup} (M||m, a||s||m||t)$;
```

```
$\underline{FindPath(h, s, m, t)}$
01 $P \leftarrow \emptyset$;
02 for all $(M, a) \in Path[h]$ do
03     if $\exists M_1$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s, M_1) = a||s||m||t$ then
04         $P \xleftarrow{\cup} (M_1, a||s||m||t)$;
05 if $P = \emptyset$ then **return** $\perp$;
06 else **return** $(M^*, a^*) \xleftarrow{\$} P$;

$\underline{S_+(k,x)}$
01 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
02 $y \leftarrow E_I(k,x)$;
03 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
04     $a \leftarrow \mathcal{TO}^{(1)}(z')$;
05     if $a \neq \perp$ or $z' = IV$ then
06         if $\exists M$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s, M) = a||s||k||t^{(1)}||t^{(3)}$ then
07             $z \leftarrow \mathcal{RO}_n(s, M)$;
08         else $z \leftarrow \mathcal{RO}^{(1)}_n(a||s||k||t^{(1)}||t^{(3)})$;
09         $y_1 \leftarrow \mathcal{RO}^{(2)}_n(k,x)$; $y_2 \leftarrow y_1 \oplus z' \oplus [s]_2 \oplus z$; $y \leftarrow y_1||y_2$;
10 **return** $y$;

$\underline{S_-(k,y)}$
01 $k^*||x \leftarrow \mathcal{TO}^{(2)}(y^L)$;
02 if $k^*||x \neq \perp$ and $k^* = k$ then **return** $x$;
03 $x \leftarrow D_I(k,y)$;
04 **return** $x$;
```

Fig. 22. S^* (left and top of right) and S (right)

D Reset Indifferentiability for the BLAKE Hash Function

First define notations used in this subsection. $[x]_2 = x||x$ is the concatenation of two copies of x. If x is of even length, then x^L and x^R denote its left and right halves where $|x^L| = |x^R|$.

Let the output length of BLAKE be n bits. Then BLAKE takes as input a salt s of $n/2$ bits (chosen by the user), and a message M of arbitrary length. The evaluation of $\mathsf{BLAKE}^{\mathsf{BC}_{2n,2n}}(\mathsf{s}, M)$ is done as follows where a block cipher $\mathsf{BC}_{2n,2n} = (E, D)$ is used where E is the encryption function and D is the decryption function with the key size and the plain text size of $2n$ bits. Firstly, the message M is padded into message blocks $m_1, \ldots, m_k$ of $2n$ bits, where the padding function pad_B is defined as $\mathsf{pad}_B(M) = M||10^{-|M|-n/2-2 \mod 2n}1||\langle|M|\rangle_{n/2}$. Along with these message blocks, counter blocks $t_1, \ldots, t_k$ of length $n/4$ bits are generated. This counter keeps track of the number of message bits hashed so far and equals 0 if the i-th message block contains no message bits. Starting from an initial state value $z_0 \in \{0,1\}^n$, the message blocks m_i and counter blocks t_i are compressed iteratively into the state using a compression function $f : \{0,1\}^n \times \{0,1\}^{n/2} \times \{0,1\}^{2n} \times \{0,1\}^{n/4} \rightarrow \{0,1\}^n$. Here, the second input to f denotes the salt s. The outcome of the BLAKE hash function is defined as its final state value $H(\mathsf{s}, M) = z_k$. f is defined as Fig. 22. Here $C \in \{0,1\}^n$ is a constant.

```
$\underline{f(z_{i-1}, \mathsf{s}, m_i, t_i)}$
$v_i \leftarrow (z_{i-1}||\mathsf{s}||[t_i^L]_2||[t_i^R]_2]) \oplus (0^n||C)$;
$w_i \leftarrow E(m_i, v_i)$;
$z_i \leftarrow w_i^L \oplus w_i^R \oplus z_{i-1} \oplus [\mathsf{s}]_2$;
**return** $z_i$;
```

We evaluate the reset indifferentiable security from $\mathcal{VO}$ for the BLAKE hash function in the ideal cipher model. We define the parameter of $\mathcal{VO}$ as $v = 1$, $n_1 = n$, $u = 2$, $t = 1$ $w_1 = n$, $w_2 = n$, $k_1 = 2n$ and $m_1 = 2n$. Thus in this case, $\mathcal{VO}.priv = \mathcal{RO}_n$ and $\mathcal{VO}.pub = (\mathcal{RO}_n, \mathcal{TRO}^{(1)}_n, \mathcal{TRO}^{(2)}_n, \mathsf{IC}^{(1)}_{2n,2n})$. The following theorem shows that the BLAKE hash function in the ideal cipher model is reset indifferentiable from a $\mathcal{VO}$.

Theorem 5. *Let $\mathsf{IC}_{2n,2n} = (E_I, D_I)$ be an ideal cipher where the length of each elements is of $2n$ bits. There exists a simulator S such that for any distinguisher $\mathcal{A}$, the following holds,*

$$\mathsf{Adv}^{\mathsf{r\text{-}indiff}}_{\mathsf{BLAKE}^{\mathsf{IC}_{2n,2n}},\mathcal{VO},S}(\mathcal{A}) \leq 3\frac{(lq_H + q_E)(lq_H + q_E + 1)}{2^n} + \frac{5q_E^2}{2^{2n+1}} + \frac{q_E^2}{2^{n-1}}.$$

where $\mathcal{A}$ can make queries to $\mathsf{BLAKE}^{\mathsf{IC}_{2n,2n}}/\mathcal{RO}_n$ and $\mathsf{IC}_{2n,2n}/S_{\mathsf{BLAKE}}$ at most q_H, q_E times, respectively, and l is a maximum number of blocks of a query to $\mathsf{BLAKE}^{\mathsf{IC}_{2n,2n}}/\mathcal{RO}_n$. S_{BLAKE} makes at most $2q_h$ queries and runs in time $\mathcal{O}(q_h)$. ♦

First, we define a padding function $\mathsf{pad}_{\mathsf{BLAKE}}$ as $\mathsf{pad}_{\mathsf{BLAKE}}(\mathsf{s}, M) = (\mathsf{s}||m_1||t_1)||\cdots||(\mathsf{s}||m_k||t_k)$. We also define $\beta_{z,\mathsf{s}}$ and $\beta^{-1}_{z,\mathsf{s}}$ as $\beta_{z,\mathsf{s}}(w) = w^L \oplus w^R \oplus z \oplus [\mathsf{s}]_2$ for $w \in \{0,1\}^{2n}$ and $\beta^{-1}_{z,\mathsf{s}}(z') = \{w \in \{0,1\}^{2n} | w^L \oplus w^R \oplus z \oplus [\mathsf{s}]_2 = z'\}$ for $z' \in \{0,1\}^n$. In the proof of the theorem, we use the result of the indifferentiable security from a RO by Andreeva *et al.* [1] They define a simulator S^* which can be implemented as Fig. 22. S^* simulates the ideal cipher $\mathsf{IC}_{2n,2n}$ so that $\mathcal{RO}_n$ and S^* are consistent. S^*_+ and S^*_- simulate the encryption oracle E_I and the decryption oracle D_I of $\mathsf{IC}_{2n,2n}$, respectively. In this simulator, the function $FindPath$ and the procedure $AddPath$ are used.

$T^+_{S^*}$ and $T^-_{S^*}$ are (initially everywhere $\perp$) tables which record query-response values of S^*. If the query $S^*_+(k,x)$ is made, the output y is recorded in $T^+_{S^*}[k,x]$ and x is recoded in $T^-_{S^*}[k,y]$. Similarly, the query-response values for S^*_- are recoded in these tables. $Path$ is a (initially everywhere $\emptyset$) table which records all paths with the BLAKE style. Namely, if (k_1, x_1, y_1) is recoded in $T^+_{S^*}$ such that $T^+_{S^*}[k_1, x_1] = y_1$, $x_1 = z_0||\mathsf{s}_1||t^L_1||t^L_1||t^R_1||t^R_1$, and $z_1 = \beta_{z_0,\mathsf{s}_1}(y_1)$, $(k_1, \mathsf{s}_1||k_1||t_1)$ is recoded in $Path[z_1]$ [6]. Then, for the query $S^*_+(k_2, x_2)$, if the query and some query-response pairs of S^* have the BLAKE structure, the output is defined by $\mathcal{RO}_n$. Namely, if $x_2 = z_1||\mathsf{s}_2||t^L_2||t^L_2||t^R_2||t^R_2$, $\mathsf{s}_1 = \mathsf{s}_2$ and there exists M such that $\mathsf{pad}_{\mathsf{BLAKE}}(\mathsf{s}_2, M) = (\mathsf{s}_1||k_1||t_1)||(\mathsf{s}_2||k_2||t_2)$, then the output y_2 is uniformly chosen at random from $\beta^{-1}_{z_1,\mathsf{s}_2}(\mathcal{RO}_n(\mathsf{s}_2, M))$ to ensure the BLAKE consistency.

The Simulator S. We define the simulator S in Fig. 22. $\mathcal{TRO}^{(1)}_n$ and $\mathcal{TRO}^{(2)}_n$ realizes the functionality of recording a path and constructing a new path. For the query $S_+(k_1, x_1)$ where $x_1 = IV||\mathsf{s}_1||t^{(1)}_1||t^{(1)}_1||t^{(3)}_1||t^{(3)}_1$ and there does not exist M such that $\mathsf{pad}_{\mathsf{BLAKE}}(\mathsf{s}_1, M) = \mathsf{s}_1||k_1||t^{(1)}_1||t^{(3)}_1$, the simulator makes the queries $\mathcal{RO}^{(1)}_n(s_1||k_1||t^{(1)}_1||t^{(3)}_1)$ and $\mathcal{RO}^{(2)}_n(k_1, x_1)$ where the responses are z_1 and $y_{1,1}$, respectively, then $y_{1,2} = y_{1,1} \oplus z \oplus [\mathsf{s}_1]_2 \oplus IV$ and the response y_1 of the query $S_+(k_1, x_1)$ is defined by $y_1 = y_{1,1}||y_{1,2}$. Then, for the query $S_{\mathsf{BLAKE},+}(k_2, x_2)$ where $x_2 = z_1||\mathsf{s}_2||t^{(1)}_2||t^{(1)}_2||t^{(3)}_2||t^{(3)}_2$ and there exists M such that $\mathsf{pad}_{\mathsf{BLAKE}}(M) = \mathsf{s}_1||k_1||t^{(1)}_1||t^{(3)}_1||\mathsf{s}_2||k_2||t^{(1)}_2||t^{(3)}_2$, the response y_2 of the query $S_+(k_2, x_2)$ is defined by $y_{2,1}||y_{2,2}$ to ensure the BLAKE consistency. The simulator can obtain $\mathsf{s}_1||k_1||t^{(1)}_1||t^{(3)}_1$ by the query $\mathcal{TO}^{(1)}(z_1)$ and thus can make the queries $\mathcal{RO}_n(\mathsf{s}_2, M)$ and $\mathcal{RO}^{(2)}_n(k_2, x_2)$ where the outputs are z_2 and $y_{2,1}$, respectively, and $y_{2,2} = y_{2,1} \oplus z \oplus [\mathsf{s}_2]_2 \oplus z_1$. Thus the simulator S_+ can make a response with the same procedure to S^*_+. For the inverse query $S_-(k_2, y_2)$, the simulator can obtain x_2 by the query $\mathcal{TO}^{(2)}(y^R_2)$. Thus the simulator S_- can also make a response with the same procedure to S^*_-. The formal evaluation of the difference $\Pr[G1] - \Pr[G2]$ is given as follows where $\Pr[G1] - \Pr[G2] \leq 5q_E^2/2^{2n+1} + q_E^2/2^{n-1}$. Since the simulator S does not update the internal state, $\Pr[G0] = \Pr[G1]$ (in Subsection **??**). The indifferentiable bound from $\mathcal{RO}_n$ in [1] is $3(lq_H + q_E)(lq_H + q_E + 1)/2^n$. There results yield the bound of Theorem 5.

Proof. We consider ten games, Game B0, Game B1, Game B2, Game B3, Game B4, Game B5, Game B6, Game B7, Game B8, and Game B9, which are shown in Figs. 23, 24, 25, 26, 27, 28, 29, 30, 31, and 32, respectively. In each game, the distinguisher $\mathcal{A}$ interacts with $(\mathcal{O}_+, \mathcal{O}_-)$. $(\mathcal{O}_+, \mathcal{O}_-)$ in Game B0 is equal to the simulator S in Game 1, and $(\mathcal{O}_+, \mathcal{O}_-)$ in Game B7 is equal to the simulator S^* in Game 2. Notice that in this proof $\mathcal{RO}_n$ queries are removed, since the difference between Game 1 and Game 2 is just the simulator.

[6] Note that in [1], the paths are recorded by using the graph representation, but the table $Path$ realizes the same role as the graph.

```
$\underline{\mathcal{O}_+(k,x)}$
01 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
02 $y \leftarrow E_I(k,x)$;
03 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
04     $a \leftarrow \mathcal{TO}^{(1)}(z')$;
05     if $a \neq \perp$ or $z' = IV$ then
06         if $\exists M$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s,M) = a||s||k||t^{(1)}||t^{(3)}$ then
07             $z \leftarrow \mathcal{RO}_n(s,M)$;
08         else $z \leftarrow \mathcal{RO}_n^{(1)}(a||s||k||t^{(1)}||t^{(3)})$;
09         $y_1 \leftarrow \mathcal{RO}_n^{(2)}(k,x)$; $y_2 \leftarrow y_1 \oplus z' \oplus [s]_2 \oplus z$; $y \leftarrow y_1||y_2$;
10 return $y$;

$\underline{\mathcal{O}_-(k,y)}$
01 $k^*||x \leftarrow \mathcal{TO}^{(2)}(y^L)$;
02 if $k^*||x \neq \perp$ and $k^* = k$ then return $x$;
03 $x \leftarrow D_I(k,y)$;
04 return $x$;
```

Fig. 23. Game B0

```
$\underline{\mathcal{O}_+(k,x)}$
01 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
02 $y \leftarrow E^*(k,x)$;
03 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
04     $a \leftarrow \mathcal{TO}^{(1)}(z')$;
05     if $a \neq \perp$ or $z' = IV$ then
06         if $\exists M$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s,M) = a||s||k||t^{(1)}||t^{(3)}$ then
07             $z \leftarrow \mathcal{RO}_n(s,M)$;
08         else $z \leftarrow \mathcal{RO}_n^{(1)}(a||s||k||t^{(1)}||t^{(3)})$;
09         $y_1 \leftarrow \mathcal{RO}_n^{(2)}(k,x)$; $y_2 \leftarrow y_1 \oplus z' \oplus [s]_2 \oplus z$; $y \leftarrow y_1||y_2$;
10 return $y$;

$\underline{\mathcal{O}_-(k,y)}$
01 $k^*||x \leftarrow \mathcal{TO}^{(2)}(y^L)$;
02 if $k^*||x \neq \perp$ and $k^* = k$ then return $x$;
03 $x \leftarrow D^*(k,y)$;
04 return $x$;

$\underline{E^*(k,x)}$
01 if $\mathsf{E}^*[k,x] = \perp$ then $\mathsf{E}^*[k,x] \xleftarrow{\$} \{0,1\}^{2n}$;
02 $\mathsf{D}^*[k,\mathsf{E}^*[k,x]] \leftarrow x$;
03 return $\mathsf{E}^*[k,x]$;

$\underline{D^*(k,y)}$
01 if $\mathsf{D}^*[k,y] = \perp$ then $\mathsf{D}^*[k,y] \xleftarrow{\$} \{0,1\}^{2n}$;
02 $\mathsf{E}^*[k,\mathsf{D}^*[k,y]] \leftarrow y$;
03 return $\mathsf{D}^*[k,y]$;
```

Fig. 24. Game B1

Let GBj be an event that the distinguisher $\mathcal{A}$ output 1 in Game Bj. Thus

$$\begin{aligned}\Pr[G1] - \Pr[G2] &= \Pr[GB0] - \Pr[GB9] \\ &= \sum_{j=0}^{8}(\Pr[GBj] - \Pr[GB(j+1)]).\end{aligned}$$

In the following, we evaluate the each difference $\Pr[GBj] - \Pr[GB(j+1)]$.

Game B1. In Game B0 the ideal cipher (E_I, D_I) is used, while in Game B1 (E^*, D^*) is used where an output is uniformly chosen at random from $\{0,1\}^{2n}$. E^* and D^* are (initially everywhere $\perp$) tables. We thus have via birthday analysis that

$$\Pr[GB0] - \Pr[GB1] \le \frac{2q_E^2}{2^{2n+1}}.$$

Game B2. In this game, new tables $T^+_{S^*}$ and $T^-_{S^*}$ are used which are initially everywhere $\perp$. In Game B2, if no collision for the outputs of $\mathcal{O}_+$ and the output of $\mathcal{O}_-$ occurs, for a repeated query, the value which was previously returned is returned. In Game B1, the procedure of $\mathcal{O}_+$ depends on the output of $\mathcal{TO}^{(1)}$ and the procedure of $\mathcal{O}_-$ depends on the output of $\mathcal{TO}^{(2)}$. Thus in Game B1 if no collision for the outputs of $\mathcal{RO}_n^{(1)}$ and the output of $\mathcal{RO}_n^{(2)}$ occurs then for a repeated query, the value which was previously returned is returned. Thus, in both game, if o collision for the outputs of $\mathcal{RO}_n^{(1)}$, the output of $\mathcal{RO}_n^{(2)}$, the outputs of

```
$\underline{\mathcal{O}_+(k,x)}$
01 if $T^+_{S^*}[k,x] \neq \perp$ then **return** $T^+_{S^*}[k,x]$;
02 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
03 $y \leftarrow E^*(k,x)$;
04 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
05     $a \leftarrow \mathcal{TO}^{(1)}(z')$;
06     if $a \neq \perp$ or $z' = IV$ then
07         if $\exists M$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s,M) = a||s||k||t^{(1)}||t^{(3)}$ then
08             $z \leftarrow \mathcal{RO}_n(s,M)$;
09         else $z \leftarrow \mathcal{RO}_n^{(1)}(a||s||k||t^{(1)}||t^{(3)})$;
10         $y_1 \leftarrow \mathcal{RO}_n^{(2)}(k,x)$; $y_2 \leftarrow y_1 \oplus z' \oplus [s]_2 \oplus z$; $y \leftarrow y_1||y_2$;
11 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
12 **return** $T^+_{S^*}[k,x]$;

$\underline{\mathcal{O}_-(k,y)}$
01 if $T^-_{S^*}[k,y] \neq \perp$ then **return** $T^-_{S^*}[k,y]$;
02 $k^*||x^* \leftarrow \mathcal{TO}^{(2)}(y^L)$;
03 if $k^*||x^* \neq \perp$ and $k^* = k$ then $x \leftarrow x^*$;
04 else $x \leftarrow D^*(k,y)$;
05 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
06 **return** $T^-_{S^*}[k,y]$;

$\underline{E^*(k,x)}$
01 if $\mathsf{E}^*[k,x] = \perp$ then $\mathsf{E}^*[k,x] \xleftarrow{\$} \{0,1\}^{2n}$;
02 $\mathsf{D}^*[k,\mathsf{E}^*[k,x]] \leftarrow x$;
03 **return** $\mathsf{E}^*[k,x]$;

$\underline{D^*(k,y)}$
01 if $\mathsf{D}^*[k,y] = \perp$ then $\mathsf{D}^*[k,y] \xleftarrow{\$} \{0,1\}^{2n}$;
02 $\mathsf{E}^*[k,\mathsf{D}^*[k,y]] \leftarrow y$;
03 **return** $\mathsf{D}^*[k,y]$;
```

Fig. 25. Game B2

```
$\underline{\mathcal{O}_+(k,x)}$
01 if $T^+_{S^*}[k,x] \neq \perp$ then **return** $T^+_{S^*}[k,x]$;
02 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
03 $y \xleftarrow{\$} \{0,1\}^{2n}$;
04 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
05     $a \leftarrow \mathcal{TO}^{(1)}(z')$;
06     if $a \neq \perp$ or $z' = IV$ then
07         if $\exists M$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s,M) = a||s||k||t^{(1)}||t^{(3)}$ then
08             $z \leftarrow \mathcal{RO}_n(s,M)$;
09         else $z \leftarrow \mathcal{RO}_n^{(1)}(a||s||k||t^{(1)}||t^{(3)})$;
10         $y_1 \leftarrow \mathcal{RO}_n^{(2)}(k,x)$; $y_2 \leftarrow y_1 \oplus z' \oplus [s]_2 \oplus z$; $y \leftarrow y_1||y_2$;
11 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
12 **return** $T^+_{S^*}[k,x]$;

$\underline{\mathcal{O}_-(k,y)}$
01 if $T^-_{S^*}[k,y] \neq \perp$ then **return** $T^-_{S^*}[k,y]$;
02 $k^*||x^* \leftarrow \mathcal{TO}^{(2)}(y^L)$;
03 if $k^*||x^* \neq \perp$ and $k^* = k$ then $x \leftarrow x^*$;
04 else $x \xleftarrow{\$} \{0,1\}^{2n}$;
05 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
06 **return** $T^-_{S^*}[k,y]$;
```

Fig. 26. Game B3

$\mathcal{O}_+$ and the output of $\mathcal{O}_-$, the modification for Game B2 does not affect the distinguisher's behavior and so Game B2 is equal to Game B1. We thus have via a birthday analysis that

$$\Pr[GB1] - \Pr[GB2] \leq \frac{2q_E^2}{2^{n+1}} + \frac{2q_E^2}{2^{2n+1}}.$$

Game B3. In this game, (E^*, D^*) is removed. Outputs of E^* and D^* are uniformly chosen at random from $\{0,1\}^{2n}$. In Game B2, if no collision occurs for $\mathcal{O}_+, \mathcal{O}_-$, for a repeated query, the value which was previously returned is returned by the tables $T^+_{S^*}$ and $T^-_{S^*}$. Thus in both games if no collision occurs for the outputs of $\mathcal{O}_+$ and the outputs of $\mathcal{O}_-$, the modification does not affect the distinguisher's behavior. We thus have via a birthday analysis that

$$\Pr[GB2] - \Pr[GB3] \leq \frac{2q_E^2}{2^{2n+1}}.$$

Game B4. In this game, $\mathcal{TO}^{(2)}$ in $\mathcal{O}_-$ is removed. $k^*||x^*$ $(= \mathcal{TO}(y^L) \neq \perp$ means that the value corresponding with the query (k,y) is recoded. If no collision occurs for the output of $\mathcal{O}_+$ and the output of $\mathcal{O}_-$, for a repeated query, the value which was previously returned is returned. That is, if no collision occurs, $\mathcal{TO}^{(2)}$ is

```
$\underline{\mathcal{O}_+(k,x)}$
01 if $T^+_{S^*}[k,x] \neq \perp$ then **return** $T^+_{S^*}[k,x]$;
02 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
03 $y \xleftarrow{\$} \{0,1\}^{2n}$;
04 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
05     $a \leftarrow \mathcal{TO}^{(1)}(z')$;
06     if $a \neq \perp$ or $z' = IV$ then
07         if $\exists M$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s,M) = a||s||k||t^{(1)}||t^{(3)}$ then
08             $z \leftarrow \mathcal{RO}_n(s,M)$;
09         else $z \leftarrow \mathcal{RO}^{(1)}_n(a||s||k||t^{(1)}||t^{(3)})$;
10         $y_1 \leftarrow \mathcal{RO}^{(2)}_n(k,x)$; $y_2 \leftarrow y_1 \oplus z' \oplus [s]_2 \oplus z$; $y \leftarrow y_1||y_2$;
11 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
12 **return** $T^+_{S^*}[k,x]$;
```

```
$\underline{\mathcal{O}_-(k,y)}$
01 if $T^-_{S^*}[k,y] \neq \perp$ then **return** $T^-_{S^*}[k,y]$;
02 $x \xleftarrow{\$} \{0,1\}^{2n}$;
03 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
04 **return** $T^-_{S^*}[k,y]$;
```

Fig. 27. Game B4

```
$\underline{\mathcal{O}_+(k,x)}$
01 if $T^+_{S^*}[k,x] \neq \perp$ then **return** $T^+_{S^*}[k,x]$;
02 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
03 $y \xleftarrow{\$} \{0,1\}^{2n}$;
04 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
05     $a \leftarrow \mathcal{TO}^{(1)}(z')$;
06     if $a \neq \perp$ or $z' = IV$ then
07         if $\exists M$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s,M) = a||s||k||t^{(1)}||t^{(3)}$ then
08             $z \leftarrow \mathcal{RO}_n(s,M)$;
09         else $z \leftarrow \mathcal{RO}^{(1)}_n(a||s||k||t^{(1)}||t^{(3)})$;
10         $y \xleftarrow{\$} \beta^{-1}_{z',s}(z)$;
11 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
12 **return** $T^+_{S^*}[k,x]$;
```

```
$\underline{\mathcal{O}_-(k,y)}$
01 if $T^-_{S^*}[k,y] \neq \perp$ then **return** $T^-_{S^*}[k,y]$;
02 $x \xleftarrow{\$} \{0,1\}^{2n}$;
03 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
04 **return** $T^-_{S^*}[k,y]$;
```

Fig. 28. Game B5

not used and thus Game B3 is equal to Game B2. We thus have via a birthday analysis that

$$\Pr[GB3] - \Pr[GB4] \leq \frac{2q_E^2}{2^{2n+1}}.$$

Game B5. In this game, $\mathcal{RO}^{(2)}_n$ is removed. In both Game B4 and Game B5, y is uniformly chosen at random from $\{0,1\}^{2n}$ with a relation that $\beta_{z',s}(y) = z$. Thus Game B5 is equal to Game B4 and $\Pr[GB4] = \Pr[GB5]$.

Game B6. In this game, $\mathcal{TRO}^{(1)}_n$ is removed. Instead, the functions $FindPath_1$ and $AddPath_1$ are used. $Path$ is a (initially everywhere $\perp$) table. If no collision occurs for the outputs of $AddPath_1$, then $AddPath_1$ and $FindPath_1$ behave as $\mathcal{RO}^{(1)}_n$ and $\mathcal{TO}^{(1)}$, respectively. That is, if no collision occurs, Game B6 is equal to Game B5. We thus have via a birthday analysis that

$$\Pr[GB5] - \Pr[GB6] \leq \frac{q_E^2}{2^{n+1}}.$$

Game B7. In this game, $AddPath$ and $FindPath$ are used instead of $AddPath_1$ and $FindPath_1$. For some value z, in $AddPath_1$, the number of paths in $Path_1[z]$ is at most 1, while in $AddPath$, the number of paths in $Path[z]$ not limited. Thus, if for any value z the number of paths in $Path[z]$ is at most 1, Game B7 is equal to Game B6. That is, if no collision for $\beta_{z',s}$ (step 07 in $\mathcal{O}_+$) occurs then Game B7 is equal to Game B6. Since y is uniformly chosen at random from $\{0,1\}^{2n}$, an output of $\beta_{z',s}$ is a random value of n bits. We

```
$\underline{\mathcal{O}_+(k,x)}$
01 if $T^+_{S^*}[k,x] \neq \perp$ then **return** $T^+_{S^*}[k,x]$;
02 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
03 $y \xleftarrow{\$} \{0,1\}^{2n}$;
04 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
05     $a \leftarrow FindPath_1(z')$;
06     if $a \neq \perp$ or $z' = IV$ then
07         if $\exists M$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s,M) = a||s||k||t^{(1)}||t^{(3)}$ then
08             $z \leftarrow \mathcal{RO}_n(s,M)$;
09         else $z \leftarrow AddPath_1(a||s||k||t^{(1)}||t^{(3)})$;
10         $y \xleftarrow{\$} \beta^{-1}_{z',s}(z)$;
11 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
12 **return** $T^+_{S^*}[k,x]$;

$\underline{\mathcal{O}_-(k,y)}$
01 if $T^-_{S^*}[k,y] \neq \perp$ then **return** $T^-_{S^*}[k,y]$;
02 $x \xleftarrow{\$} \{0,1\}^{2n}$;
03 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
04 **return** $T^-_{S^*}[k,y]$;

$\underline{AddPath_1(M)}$
01 if $\exists z$ s.t. $Path_1[z] = M$ then **return** $z$;
02 $z \xleftarrow{\$} \{0,1\}^n$;
03 $Path_1[z] \leftarrow M$;
04 **return** $z$;

$\underline{FindPath_1(z)}$
01 if $Path_1[z] \neq \perp$ then **return** $Path[z]$;
02 **return** $\perp$;
```

Fig. 29. Game B6

```
$\underline{\mathcal{O}_+(k,x)}$
01 if $T^+_{S^*}[k,x] \neq \perp$ then **return** $T^+_{S^*}[k,x]$;
02 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
03 $y \xleftarrow{\$} \{0,1\}^{2n}$;
04 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
05     $P \leftarrow FindPath(z',s,k,t)$;
06     if $P \neq \perp$ then let $P = (M,a)$; $z \leftarrow \mathcal{RO}_n(s,M)$;
07     else $AddPath(\beta_{z',s}(y), z', s, k, t^{(1)}||t^{(3)})$;
08 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
09 **return** $T^+_{S^*}[k,x]$;

$\underline{AddPath(z,h,s,m,t)}$
01 for all $(M,a) \in Path[h]$ do
02     $Path[z] \xleftarrow{\cup} (M||m, a||s||m||t)$;

$\underline{\mathcal{O}_-(k,y)}$
01 if $T^-_{S^*}[k,y] \neq \perp$ then **return** $T^-_{S^*}[k,y]$;
02 $x \xleftarrow{\$} \{0,1\}^{2n}$;
03 $T^+_{S^*}[k,x] \leftarrow y$; $T^-_{S^*}[k,y] \leftarrow x$;
04 **return** $T^-_{S^*}[k,y]$;

$\underline{FindPath(h,s,m,t)}$
01 $P \leftarrow \emptyset$;
02 for all $(M,a) \in Path[h]$ do
03     if $\exists M_1$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s,M_1) = a||s||m||t$ then
04         $P \xleftarrow{\cup} (M_1, a||s||m||t)$;
05 if $P = \emptyset$ then **return** $\perp$;
06 else **return** $(M^*, a^*) \xleftarrow{\$} P$;
```

Fig. 30. Game B7

thus via birthday analysis that

$$\Pr[GB6] - \Pr[GB7] \leq \frac{q_E^2}{2^{n+1}}.$$

Game B8. In Game B8, "else" for the step using *AddPath* is removed. Since $\mathsf{pad}_{\mathsf{BLAKE}}$ is a prefix-free padding, the path constructed from the value defined the steps 06-08 is not used. Thus the modification does not affect the distinguisher's behavior. So we have that $\Pr[GB7] = \Pr[GB8]$.

Game B9. In this game, *AddPath* is added in $\mathcal{O}_-$. Since x is uniformly chosen at random from $\{0,1\}^{2n}$, the probability that in $\mathcal{O}_-$ a new path is added in the table *Path* is at most $q_E^2/2^n$ where the number of paths stored in *Path* is at most q_E. We thus have that

$$\Pr[GB8] - \Pr[GB9] \leq \frac{q_E^2}{2^n}.$$

□

```
$\underline{\mathcal{O}_B^+(k,x)}$
01 if $T_{S^*}^+[k,x] \neq \perp$ then **return** $T_{S^*}^+[k,x]$;
02 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
03 $y \xleftarrow{\$} \{0,1\}^{2n}$;
04 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
05     $P \leftarrow FindPath(z', s, k, t^{(1)}||t^{(3)})$;
06     if $P \neq \emptyset$ then
07         let $P = (M, a)$;
08         $z \leftarrow \mathcal{RO}_n(s, M)$; $y \xleftarrow{\$} \beta_{z',s}^{-1}(z)$;
09     $AddPath(\beta_{z',s}(y), z', s, k, t^{(1)}||t^{(3)})$:
10 $T_{S^*}^+[k,x] \leftarrow y$; $T_{S^*}^-[k,y] \leftarrow x$;
11 **return** $T_{S^*}^+[k,x]$;

$\underline{AddPath(z,h,s,m,t)}$
01 for all $(M,a) \in Path[h]$ do
02     $Path[z] \xleftarrow{\cup} (M||m, a||s||m||t)$;
```

```
$\underline{\mathcal{O}_B^-(k,y)}$
01 if $T_{S^*}^-[k,y] \neq \perp$ then **return** $T_{S^*}^-[k,y]$;
02 $x \xleftarrow{\$} \{0,1\}^{2n}$;
03 $T_{S^*}^+[k,x] \leftarrow y$; $T_{S^*}^-[k,y] \leftarrow x$;
04 **return** $T_{S^*}^-[k,y]$;

$\underline{FindPath(h,s,m,t)}$
01 $P \leftarrow \emptyset$;
02 for all $(M,a) \in Path[h]$ do
03     if $\exists M_1$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s, M_1) = a||s||m||t$ then
04         $P \xleftarrow{\cup} (M_1, a||s||m||t)$;
05 if $P = \emptyset$ then **return** $\perp$;
06 else **return** $(M^*, a^*) \xleftarrow{\$} P$;
```

Fig. 31. Game B8

```
$\underline{\mathcal{O}_+(k,x)}$
01 if $T_{S^*}^+[k,x] \neq \perp$ then **return** $T_{S^*}^+[k,x]$;
02 $z'||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \oplus (0^n||C)$;
03 $y \xleftarrow{\$} \{0,1\}^{2n}$;
04 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
05     $P \leftarrow FindPath(z', s, k, t^{(1)}||t^{(3)})$;
06     if $P \neq \emptyset$ then
07         let $P = (M, a)$;
08         $z \leftarrow \mathcal{RO}_n(s, M)$; $y \xleftarrow{\$} \beta_{z',s}^{-1}(z)$;
09     $AddPath(\beta_{z',s}(y), z', s, k, t^{(1)}||t^{(3)})$:
10 $T_{S^*}^+[k,x] \leftarrow y$; $T_{S^*}^-[k,y] \leftarrow x$;
11 **return** $T_{S^*}^+[k,x]$;

$\underline{AddPath(z,h,s,m,t)}$
01 for all $(M,a) \in Path[h]$ do
02     $Path[z] \xleftarrow{\cup} (M||m, a||s||m||t)$;
```

```
$\underline{\mathcal{O}_-(k,y)}$
01 if $T_{S^*}^-[k,y] \neq \perp$ then **return** $T_{S^*}^-[k,y]$;
02 $z||s||t^{(1)}||t^{(2)}||t^{(3)}||t^{(4)} \leftarrow x \xleftarrow{\$} \{0,1\}^{2n}$;
03 if $t^{(1)}||t^{(3)} = t^{(2)}||t^{(4)}$ then
04     $AddPath(\beta_{z,s}(y), z, s, k, t^{(1)}||t^{(3)})$;
05 $T_{S^*}^+[k,x] \leftarrow y$; $T_{S^*}^-[k,y] \leftarrow x$;
06 **return** $T_{S^*}^-[k,y]$;

$\underline{FindPath(h,s,m,t)}$
01 $P \leftarrow \emptyset$;
02 for all $(M,a) \in Path[h]$ do
03     if $\exists M_1$ s.t. $\mathsf{pad}_{\mathsf{BLAKE}}(s, M_1) = a||s||m||t$ then
04         $P \xleftarrow{\cup} (M_1, a||s||m||t)$;
05 if $P = \emptyset$ then **return** $\perp$;
06 else **return** $(M^*, a^*) \xleftarrow{\$} P$;
```

Fig. 32. Game B9