Security Analysis of the Mode of JH Hash Function

Rishiraj Bhattacharyya[1], Avradip Mandal[2], and Mridul Nandi[3,⋆]

[1] Indian Statistical Institute, Kolkata, India
rishi_r@isical.ac.in
[2] Université du Luxembourg, Luxembourg
avradip.mandal@uni.lu
[3] NIST, USA and Computer Science Department, The George Washington University
mridul.nandi@gmail.com

Abstract. Recently, NIST has selected 14 second round candidates of SHA3 competition. One of these candidates will win the competition and eventually become the new hash function standard. In TCC'04, Maurer *et al* introduced the notion of indifferentiability as a generalization of the concept of the indistinguishability of two systems. Indifferentiability is the appropriate notion of modeling a random oracle as well as a strong security criteria for a hash-design. In this paper we analyze the indifferentiability and preimage resistance of JH hash function which is one of the SHA3 second round candidates. JH uses a $2n$ bit fixed permutation based compression function and applies chopMD domain extension with specific padding.

- We show under the assumption that the underlying permutations is a $2n$-bit random permutation, JH mode of operation with output length $2n - s$ bits, is indifferentiable from a random oracle with distinguisher's advantage bounded by $O(\frac{q^2\sigma}{2^s} + \frac{q^3}{2^n})$ where σ is the total number of blocks queried by distinguisher.
- We show that the padding rule used in JH is essential as there is a simple indifferentiablity distinguisher (with constant query complexity) against JH mode of operation without length padding outputting n bit digest.
- We prove that a little modification (namely chopping different bits) of JH mode of operation enables us to construct a hash function based on random permutation (without any length padding) with similar bound of sponge constructions (with fixed output size) and with same efficiency.
- On the other hand, we improve the preimage attack of query complexity $2^{510.3}$ due to Mendel and Thompson. Using multicollisions in both forward and reverse direction, we show a preimage attack on JH with $n = 512, s = 512$ in 2^{507} queries to the permutation.

Keywords: JH, SHA-3 candidate, Indifferentiability, chop-MD, random permutation.

1 Introduction

Designing secure hash function is a primary objective of symmetric key cryptography. Popular methods to build a hash function involve two steps. First, one designs a

⋆ Supported in part by the National Science Foundation, Grant CNS-0937267.

S. Hong and T. Iwata (Eds.): FSE 2010, LNCS 6147, pp. 168–191, 2010.

compression function $f : \{0,1\}^m \rightarrow \{0,1\}^n$ where $m > n$. Then a domain extension algorithm that utilizes f as a black box[1] is applied to implement the hash function $H^f : \{0,1\}^* \rightarrow \{0,1\}^n$. This is also known as design or mode of the hash function. The well known Merkle-Damgård domain extension technique is very popular as it preserves the collision resistance property of the compression function: If f is collision resistant then so is H^f. This enables the designers to focus on designing collision resistant compression functions.

INDIFFERENTIABILITY. While collision resistance remains an essential property of a cryptographic hash function, current usage indicates that it no more suffices the modern security goals. Today hash functions are used as PRFs, MACs, (2nd) preimage-secure or even as to replace Random Oracles in different Cryptographic Protocols. In [6], Coron et al considered the problem of designing secure cryptographic hash function based on the *indifferentiability* framework of Maurer et al [15]. Informally speaking, to prove indifferentiablity of an iterated hash function H (based on some ideal primitive f), one has to design a simulator S. The job of S is to simulate the behavior of f while maintaining consistency with the random oracle R. If no distinguisher D can distinguish the output distribution of the pair (H^f, f) from that of (R, S^R), the construction H is said to be indifferentiable from a Random Oracle (RO). By proving indifferentiability, we are guaranteed that there is no trivial flaw in the design of the hash function; the design is secure against generic attacks. Today, indifferentiability is considered to be a desirable property of any secure hash function design. Coron et al showed in [6], the design principle (Strengthened Merkle-Damgård) behind the current standard hash functions like MD5 or SHA-1 does not satisfy indifferentiability from RO property. They also proved that different variant of MD constructions, including chopped MD constructions can be proven indifferentiable from a Variable Input Length Random Oracle if the compression function is constructed as an ideal component like Fixed Input Length Random Oracle or from Ideal Cipher with Davis Meyer technique. Subsequently, authors of [2,4,9,12] proved indifferentiability of different constructions of iterated hash functions. In [5], Chang and Nandi proved an indifferentiability bound beyond birthday bound for chopped MD constructions under the assumption that the compression function is a fixed input length random oracle.

In 2007, NIST announced a competition for a new hash function standard, to be called SHA-3. 64 designs were submitted and after an internal review of the submissions, 51 were selected for meeting the minimum submission requirements and accepted as the First Round Candidates. Recently, NIST declared the names of 14 candidates for the second round of the competition. One of these candidates will win the competition and eventually become the next standard cryptographic hash function. Hence, it is essential for these candidate designs to be indifferentiable from an RO to guarantee its robustness against generic attacks.

In this paper, we consider the mode of operation of the JH hash function, one of the second round candidates of SHA3 competition. It uses a novel construction, somewhat reminiscent of a sponge construction [4], to build a hash algorithm out of a single, large, fixed permutation using chopped-MD domain extension [21]. We also consider a little

[1] The domain extension can be applied independent of compression functions except that it depends on the parameters m and n.

modified mode of operation of JH where the chopping is done on the other bits. For a formal and detailed description of mode of operation of JH and the modified mode of operation, we refer the reader to Section 2. Although the mode of JH is novel, it has withstand many cryptanalysis attempts so far. The only noticeable attack is due to Mendel and Thompson who has recently shown a preimage attack on JH mode of operation through finding r- multicollisions in the forward direction of JH mode [16]. The query complexity of their attack is $2^{510.3}$ to get a preimage of JH outputting 512-bits.

1.1 Our Result

In this paper we examine the indifferentiability and preimage resistance of JH mode of operation in $2n$ bit random permutation model. Let s denote the number of chopped bits. We extend the technique of Chang and Nandi [5] to random permutation model. We prove that under the assumption that the fixed permutation of JH is a random permutation, JH mode of operation with specific length padding is indifferentiable from random oracle with distinguisher's advantage bounded by $O(\frac{q^2\sigma}{2^s} + \frac{q^3}{2^n})$. When $s = 3n/2$ (as in case of JH hash function with 256 bit output), our result gives beyond the birthday barrier security guarantee for JH [2]. This implies that finding collision in the output is not enough to distinguish a Random Oracle from JH hash function with $n/2$-bit output. Although chopMD constructions do not need the length padding in general, we show the padding is *essential* for JH mode. We construct one indifferentiability attacker, working in *constant* number of queries against JH mode of operation without length padding at last block with n-bit output. This result also shows that the method used in [4] to prove indifferentiability of sponge constructions (where length padding in last block is not required) based on random permutations cannot be readily extended to prove indifferentiability of JH.

Next we consider the preimage resistance of JH mode of operation and improve the preimage attack of Mendel and Thompson [16]. Our preimage attack works with query complexity 2^{507} for finding a preimage of 512-bit JH hash function. Even though it marginally reduces the complexity of the previous known attack (with $2^{510.3}$ queries), theoretically the new attack requires asymptotically less complexity. Looking ahead, we exploit the multicollision in both forward and backward direction unlike in only forward direction used in [16].

Simultaneously, we look at other constructions, modifying JH mode of operation, where the chopping is done on the first instead of last s bits.

- We show that when the length of longest query is less than $2^{n/2}$, then the modified JH mode of operation without the length padding is indifferentiable from an RO with distinguisher's advantage bounded by $\mathcal{O}(\frac{q^2}{2^{\min(s,n)}})$ where q is the maximum number of queries made by the distinguisher.
- We show one indifferentiability attacker against modified JH mode of operation with $\Omega(2^{n/2})$ query complexity. This shows for $s \geq n$ the previous security bound is actually optimal.

[2] According to birthday paradox, for a uniform random function with n-bit digest, collision can be found with significant probability in $\mathcal{O}(2^{\frac{n}{2}})$ queries. This is known as the birthday barrier as security against more than $\mathcal{O}(2^{\frac{n}{2}})$ queries is non-trivial; when at all possible.

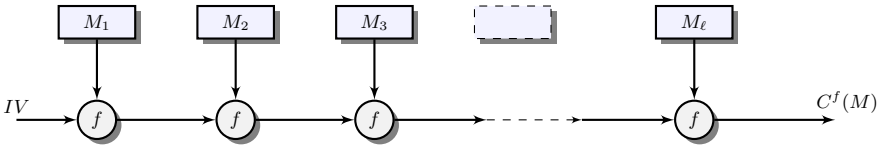

Fig. 1. Merkle-Damgård mode of operation based on compression function f

– If we set $s = n$, we get a random permutation based secure mode of operation with n-bit digest using $2n$ bit permutation. We note that this construction is atleast as secure as the sponge construction based on $2n$ bit random permutation. where the indifferentiability bound is $\mathcal{O}(\frac{\sigma^2}{2^n})$ [4]. Here σ is the number blocks that the adversary queries.

On a secondary note, even though our proof techniques for indifferentiable security bounds are closely related to the techniques used in [5,12], we give a more formal argument behind some implicit assumptions made over there.

The rest of the paper is organized as follows. In next section, we mention the notations, formal description of JH mode of operation and modified mode of operation, a short introduction to Indifferentiability of hash functions and some useful definitions and facts. In Section 3, we build our tools for extending Chang and Nandi's proof to random permutation model. For simplicity of of explanation, first we describe the indifferentiability of modified JH mode without length padding at last block in Section 4 followed by indifferentiability of original JH mode with padding in Section 5. In Section 6 and Section 7, we describe our indifferentiability distinguisher against JH mode of operation and modified mode of operation without the padding. Finally in Section 8, we present our improved preimage attack on JH mode of operation with padding.

2 Preliminaries

In this section we describe the notations and definitions used throughout the paper. Let us begin with a formal definition of mode of operation.

`Mode of Operation:` Informally speaking, a mode of operation is an algorithm to construct a hash function from a compression function.

Definition 1. *A mode of operation C with oracle access to compression function $f\{0,1\}^m \to \{0,1\}^n$ an algorithm which defines a function $C^f : \{0,1\}^* \to \{0,1\}^n$.*

Let $IV \in \{0,1\}^n$ be a fixed initial value. It is well known that given a compression function $f : \{0,1\}^m \to \{0,1\}^n$, Merkle-Damgård mode of operation is defined as

$$MD^f(m_1\|m_2\|\ldots\|m_l) = f(f(\ldots f(f(IV\|m_1)\|m_2)\ldots)\|m_l)$$

where $m_1, m_2, \ldots m_l \in \{0,1\}^{m-n}$.

There is a subtle difference between a hash function and a mode of operation. The mode of operation is actually a domain extension algorithm. If we supply a particular compression function f to the mode of operation algorithm we get a particular hash function. So when we think about a hash function, the compression function is fixed.

`JH Mode of Operation:` The compression function of JH, $f^\pi : \{0,1\}^{3n} \rightarrow \{0,1\}^{2n}$ is defined as follows:

$$f^\pi(h_1\|h_2\|m) = \pi(h_1\|(h_2 \oplus m)) \oplus (m\|0^n)$$

where $h_1, h_2, m \in \{0,1\}^n$ and $\pi : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ is a fixed permutation.

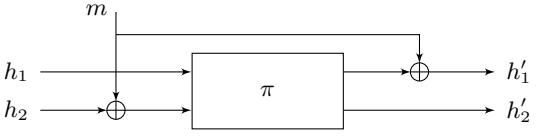

Fig. 2. The JH compression function

The JH mode of operation based on a permutation π is the chopMD mode of operation based on the above compression function f^π. The usual Merkle-Damgård technique is applied on f^π and the output of the hash function is the first $2n - s$ bits of the final f^π query output. For any, $0 \leq s \leq |m|$, $\text{CHOPR}_s(m)$ is defined as m_L where $m = m_L\|m_R$ and $|m_R| = s$. Formally the JH mode of operation based on a permutation π with initial value $IV_1\|IV_2$ is defined as

$$JH^\pi(\cdot) : (\{0,1\}^n)^+ \rightarrow \{0,1\}^{2n-s} \equiv \text{CHOPR}_s(MD^{f^\pi}(\cdot)).$$

Where, MD^{f^π} is the Merkle-Damgård mode of operation with initial value as $IV_1\|IV_2$ and compression function as f^π. According to [21], typically $s = n$. Also it is suggested to have $s \geq n$.

We also define a modified version of JH mode of operation (referred as JH$'$ throughout the paper) where instead of chopping right most s bits we chop left most s bits. Let for $0 \leq s \leq |m|$, $\text{CHOPL}_s(m)$ is defined as m_R where $m = m_L\|m_R$ and $|m_L| = s$.

$$JH'^\pi(\cdot) : (\{0,1\}^n)^+ \rightarrow \{0,1\}^{2n-s} \equiv \text{CHOPL}_s(MD^{f^\pi}(\cdot)).$$

Throughout the paper JH-t denotes the JH mode of operation with t bit output. Similarly JH$'$-t denotes JH' mode of operation with t bit output.

`Padding Rule:` To encode messages whose lengths are not multiple of block size (n bit) we need some padding rule, so that padded message becomes a multiple of block size. A simple padding rule can be zero padding, that is adding sufficient number of zero bits so that the padded message becomes a multiple of block size, even though this is not secure. We will see as in the case of JH a well designed padding rule leads to additional security guarantee.

Definition 2. *A padding rule P is a tuple of two efficiently computable functions*

$$P \equiv (\text{PAD} : \{0,1\}^* \to (\{0,1\}^n)^+, \text{DEPAD} : (\{0,1\}^n)^+ \to \{0,1\}^* \cup \{\perp\})$$

such that for any $M \in \{0,1\}^*$ we have

$$\text{DEPAD}(\text{PAD}(M)) = M.$$

$\text{DEPAD}(y)$ *outputs $\perp$ if there exists no $M \in \{0,1\}^*$ such that,* $\text{PAD}(M) = y$.

The function PAD takes a message of arbitrary length and outputs the padded message which is multiple of block length. Where as, the function DEPAD takes the padded message which is multiple of block length and outputs the original message. Normally, when we specify a padding rule we only specify the function PAD, but usually definition of DEPAD can be trivially derived from the description of PAD. In our context, we are interested in a specialized class of padding rules, namely with the following additional properties.

1. $\frac{|\text{PAD}(M)|}{n} = \lceil \frac{|M|}{n} \rceil + 1$.
2. For any $M \in (\{0,1\}^n)^+$, $LB(M) \subseteq \{0,1\}^n$ be the set of n-bit elements (possible last blocks) such that,
   $\text{DEPAD}(M \| m) \neq \perp$ for any $m \in LB(M)$. We want, $|LB(M)|$ to be small for all $M \in \{0,1\}^*$(smaller than some constant).

Here, if $x \in \{0,1\}^*$, $|x|$ denotes the length of x in bits. Also, if A is a set, $|A|$ denotes the number of elements in A. Any padding which satisfies the above two properties is called *good* padding rule. Now we are ready to define the JH mode of operation with padding.

Definition 3. *With respect to a padding rule $P = (\text{PAD}, \text{DEPAD})$ and a permutation π, the JH_P mode of operation is defined as follows,*

$$JH_P^{\pi}(\cdot) : \{0,1\}^* \to \{0,1\}^{2n-s} \equiv JH^{\pi}(\text{PAD}(\cdot)) \equiv \text{CHOPL}_s(MD^{f^{\pi}}(\text{PAD}(\cdot))).$$

`The JH Padding rule:` In [21], the following padding rule is mentioned for JH hash function with block length $n = 512$. *Suppose that the length of the message M is $\ell(M)$ bits. Append the bit* 1 *to the end of the message, followed by* $384 - 1 + (-\ell(M) \bmod 512)$ *zero bits. Then the binary representation of $\ell(M)$ in big endian form is concatenated. This padding rule ensures that at least one block of* 512 *bits is padded after the message (irrespective of whether the message length is multiple of 512)* . It is easy to check the above padding rule is actually a *good* padding rule with $|LB(M)| \leq 2$.

`Indifferentiability:` The notion of indifferentiability, introduced by Maurer et. al. in [15], is a generalization of classical notion of indistinguishability. Loosely speaking, if an ideal primitive $\mathcal{G}$ is indifferentiable with a construction C based on another ideal primitive $\mathcal{F}$, then $\mathcal{G}$ can be safely replaced by $C^{\mathcal{F}}$ in any cryptographic construction. In other terms if a cryptographic construction is secure in $\mathcal{G}$ model then it is secure in $\mathcal{F}$ model.

Definition 4. Advantage
Let F_i, G_i be probabilistic oracle algorithms. We define advantage *of the distinguisher $\mathcal{A}$ at distinguishing (F_1, F_2) from (G_1, G_2) as*

$$\mathsf{Adv}_{\mathcal{A}}((F_1, F_2), (G_1, G_2)) = |\Pr[\mathcal{A}^{F_1,F_2} = 1] - \Pr[\mathcal{A}^{G_1,G_2} = 1]|.$$

Definition 5. Indifferentiability [15]
A Turing machine C with oracle access to an ideal primitive $\mathcal{F}$ is said to be $(t, q_C, q_{\mathcal{F}}, \varepsilon)$ indifferentiable from an ideal primitive $\mathcal{G}$ if there exists a simulator S with an oracle access to $\mathcal{G}$ and running time at most t, such that for any distinguisher D, it holds that

$$\mathsf{Adv}_D((C^{\mathcal{F}}, \mathcal{F}), (\mathcal{G}, S^{\mathcal{G}})) < \varepsilon.$$

The distinguisher makes at most q_C queries to C or $\mathcal{G}$ and at most $q_{\mathcal{F}}$ queries to $\mathcal{F}$ or S. Similarly, $C^{\mathcal{F}}$ is said to be (computationally) indifferentiable from $\mathcal{G}$ if running time of D is bounded above by some polynomial in the security parameter k and ε is a negligible function of k.

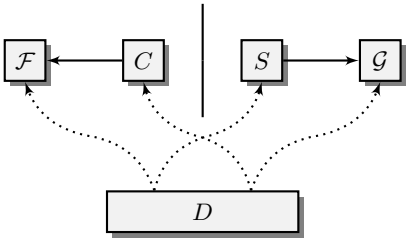

Fig. 3. The indifferentiability notion

We stress that in the above definition $\mathcal{G}$ and $\mathcal{F}$ can be two completely different primitives. As shown in Fig 3 the role of the simulator is to not only simulate the behavior of $\mathcal{F}$ but also remain consistent with the behavior of $\mathcal{G}$. Note that, the simulator does not know the queries made directly to $\mathcal{G}$, although it can query $\mathcal{G}$ whenever it needs.

In this paper $\mathcal{G}$ is a variable input length Random oracle and $\mathcal{F}$ is a random permutation. Intuitively a random function (oracle) is a function $f : X \to Y$ chosen uniformly at random from the set of all functions from X to Y.

Definition 6. *$f : X \to Y$ is said to be a* random oracle *if for each $x \in X$ the value of $f(x)$ is chosen uniformly at random from Y. More precisely,*

$$\Pr[f(x) = y \mid f(x_1) = y_1, f(x_2) = y_2, \ldots, f(x_q) = y_q] = \frac{1}{|Y|}$$

where $|Y|$ is finite and $x \notin \{x_1, \ldots, x_q\}$ and $y, y_1, \ldots, y_q \in Y$.

A random permutation is similar to random oracle except that it is a permutation. So similarly one can view a random permutation $\pi : X \to X$ as a permutation chosen uniformly at random from the set of all permutation from X to X.

Definition 7. *$\pi : X \rightarrow X$ is said to be a* random permutation *if for each $x \in X$ we have,*

$$\Pr[\pi(x) = y \mid \pi(x_1) = y_1, \pi(x_2) = y_2, \ldots, \pi(x_q) = y_q] = \frac{1}{|X| - q}$$

where $|X|$ is finite and $x \notin \{x_1, \ldots, x_q\}$, $y_1, \ldots, y_q \in X$ and $y \in X \setminus \{y_1, \ldots, y_q\}$.

Definition 8. *$FH : \{0,1\}^{2n} \rightarrow \{0,1\}^n$ is a function which outputs first n bit of any $2n$ bit number. Similarly, $LH : \{0,1\}^{2n} \rightarrow \{0,1\}^n$ is a function which outputs last n bit of any $2n$ bit number.*

Often we refer FH as left half and LH as right half. Below we state a few basic inequalities as a lemma which will be useful later.

Lemma 1. *For any $y \in \{0,1\}^{2n-s}$, $c \in \{0,1\}^n$, $\mathcal{S} \subseteq \{0,1\}^{2n}$ and $\mathcal{T} \subseteq \{0,1\}^n$ we have,*

1. $|\{z \in \{0,1\}^s : y\|z \in \mathcal{S}\}| \leq |\mathcal{S}|$ *and* $|\{z \in \{0,1\}^s : z\|y \in \mathcal{S}\}| \leq |\mathcal{S}|$
2. $|\{z \in \{0,1\}^s : FH(y\|z) \oplus c \in \mathcal{T}\}| \leq 2^n|\mathcal{T}|$ *and* $|\{z \in \{0,1\}^s : FH(z\|y) \oplus c \in \mathcal{T}\}| \leq \frac{2^s}{2^{\min(s,n)}}|\mathcal{T}|$

3 Main Tools for Bounding Distinguisher's Advantage

We follow a similar approach to [5,12] for proving indifferentiability security over here. We start with modeling the attacker. Then we construct a simulator, for which the information the attacker sees remain statistically close whether the attacker is interacting with JH Hash function and the random permutation it is based on, or it is interacting with a random function and the simulator. Compared to [5] we do not restrict ourselves to some particular type of *irreducible views*. The underlying small domain oracle being a random permutation we also need to answer inverse queries.

`Consistent Oracles`
Intuitively, a small domain oracle is said to be consistent to a big domain oracle with respect to some mode of operation if querying the mode of operation based on the small domain oracle is equivalent to querying the big domain oracle.

Definition 9. *A (small domain) probabilistic oracle algorithm G_2 is said to be* consistent *to a (big domain) probabilistic oracle algorithm G_1 with respect to MO-mode of operation if for any point x (from the big domain), we have*

$$\Pr[G_1(x) = MO^{G_2}(x)] = 1.$$

The notion of consistent oracles is nothing new. In fact, in all the previous works e.g. [4,5,6,7,9,10,12] and many others, the simulators mentioned over there are always consistent to the big domain oracle (or they abort, when they fail to be consistent). Also note, π is always consistent to JH^{π} with respect to JH-mode of operation.

`Evaluatable queries`
There might be some point x for which the value of $MO^{G_2}(x)$ gets fixed by the relations $G_2(x_1) = y_1, \cdots, G_2(x_q) = y_q$. Such x's are called evaluatable by the relations $G_2(x_1) = y_1, \cdots, G_2(x_q) = y_q$. Formally,

Definition 10. *A point $x \in Domain(MO^{G_2})$ is called* evaluatable *with respect to MO-mode of operation (based on G_2) by the relations $G_2(x_1) = y_1, \cdots, G_2(x_q) = y_q$, if there exist a deterministic algorithm $\mathcal{B}$ such that,*

$$\Pr[MO^{G_2}(x) = \mathcal{B}(x, (x_1, y_1), \cdots, (x_q, y_q)) | G_2(x_1) = y_1, \cdots, G_2(x_q) = y_q] = 1.$$

`Modeling the adversary`

In this paper the adversary is modeled as a deterministic, computationally unbounded[3] distinguisher $\mathcal{A}$ which has access to two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. Recall that $\mathcal{A}$ tries to distinguish the output distribution of (JH^π, π) from that of (R, S^R). We say $\mathcal{A}$ queries $\mathcal{O}_1$ when it queries the oracle JH^π or R and queries $\mathcal{O}_2$ when it queries the oracle π or S^R. As we model π as a random permutation, the distinguisher is allowed to make inverse queries to oracle $\mathcal{O}_2$. We denote the forward query as $(\mathcal{O}_2(+, \cdot, \cdot))$ and inverse query as $(\mathcal{O}_2(-, \cdot, \cdot))$. The view $\mathcal{V}$ of the distinguisher is the list query-response tuple

$$((M_1, h_1), \ldots, (M_{q_1}, h_{q_1}), (x_1^1, x_1^2, y_1^1, y_1^2), \ldots, (x_{q_2+q_3}^1, x_{q_2+q_3}^2, y_{q_2+q_3}^1, y_{q_2+q_3}^2)) \quad (1)$$

Where,

$$\mathcal{O}_1(M_1) = h_1, \ldots, \mathcal{O}_1(M_{q_1}) = h_{q_1}$$
$$\mathcal{O}_2(+, x_1^1, x_1^2) = (y_1^1, y_1^2), \ldots, \mathcal{O}_2(+, x_{q_2}^1, x_{q_2}^2) = (y_{q_2}^1, y_{q_2}^2)$$
$$\mathcal{O}_2(-, y_{q_2+1}^1, y_{q_2+1}^2) = (x_{q_2+1}^1, x_{q_2+1}^2), \ldots, \mathcal{O}_2(-, y_{q_2+q_3}^1, y_{q_2+q_3}^2) = (x_{q_2+q_3}^1, x_{q_2+q_3}^2)$$

Definition 11. *For any view $\mathcal{V}$ as in (1), we define* Input View $\mathcal{I}(\mathcal{V})$ *and* Output View $\mathcal{O}(\mathcal{V})$ *as follows,*

$$\mathcal{I}(\mathcal{V}) = (M_1, \ldots, M_q, (x_1^1, x_1^2), \ldots, (x_{q_2}^1, x_{q_2}^2), (y_{q_2+1}^1, y_{q_2+1}^2), \ldots, (y_{q_2+q_3}^1, y_{q_2+q_3}^2))$$
$$\mathcal{O}(\mathcal{V}) = (h_1, \ldots, h_q, (y_1^1, y_1^2), \ldots, (y_{q_2}^1, y_{q_2}^2), (x_{q_2+1}^1, x_{q_2+1}^2), \ldots, (x_{q_2+q_3}^1, x_{q_2+q_3}^2))$$

Below we point out some important observations,

1. $\mathcal{V}, \mathcal{I}(\mathcal{V})$ and $\mathcal{O}(\mathcal{V})$ are actually ordered tuples. That means, the position of any element inside the tuple actually denotes the corresponding query number. So, in general $\mathcal{O}_1(.)$, $\mathcal{O}_2(+, (., .))$ and $\mathcal{O}_2(-, (., .))$ queries should not be grouped together. But we write it like this to avoid further notational complexity.
2. For any deterministic *non-adaptive* attacker $\mathcal{I}(\mathcal{V})$ is always fixed.
3. For any deterministic *adaptive* attacker $\mathcal{I}(\mathcal{V})$ is actually determined by $\mathcal{O}(\mathcal{V})$ [18].
4. For any deterministic attacker (adaptive or non-adaptive) $\mathcal{V}$ is actually determined by $\mathcal{O}(\mathcal{V})$.

`Irreducible Views`

Loosely speaking an irreducible view does not contain any duplicate query, and none of the $\mathcal{O}_1$ queries are evaluatable from the $\mathcal{O}_2$ queries present in the view.

[3] Any deterministic adversary with unlimited resource is as powerful as a randomized adversary [18].

Definition 12. *A view,*

$$\mathcal{V}=((M_1,h_1),\ldots,(M_{q_1},h_{q_1}),(x_1^1,x_1^2,y_1^1,y_1^2),\ldots,(x_{q_2+q_3}^1,x_{q_2+q_3}^2,y_{q_2+q_3}^1,y_{q_2+q_3}^2))$$

is called irreducible *if*

- $M_1,\ldots,M_{q_1}$ *are distinct,*
- $(x_1^1,x_1^2),\ldots,(x_{q_2+q_3}^1,x_{q_2+q_3}^2)$ *are distinct,*
- $(y_1^1,y_1^2),\ldots,(y_{q_2+q_3}^1,y_{q_2+q_3}^2)$ *are distinct,*
- $M_1,\cdots,M_{q_1}$ *are not evaluatable by the relations*

$$\pi(x_1^1,x_1^2)=(y_1^1,y_1^2),\ldots,\pi(x_{q_2+q_3}^1,x_{q_2+q_3}^2)=(y_{q_2+q_3}^1,y_{q_2+q_3}^2)$$

with respect to MD-mode of operation based on f^π.

Also, any view which is not irreducible is called reducible view.

Definition 13. *For an attacker $\mathcal{A}$, an output view $\mathcal{OV}$ is called* irreducible *if the corresponding view $\mathcal{V}$ is irreducible. Any output view which is not irreducible is called* reducible *output view.*

Let $\mathcal{OV}_{\mathcal{O}_1,\mathcal{O}_2}^{\mathcal{A}}$ be the random variable corresponding to the output view of attacker $\mathcal{A}$, obtained after interacting with $\mathcal{O}_1,\mathcal{O}_2$. Also, $\mathcal{V}_{\mathcal{O}_1,\mathcal{O}_2}^{\mathcal{A}}$ be the random variable corresponding to the view of attacker $\mathcal{A}$, obtained after interacting with $\mathcal{O}_1,\mathcal{O}_2$.

The theorem below shows, if the probability distributions for all possible output views in two scenarios are close, then the attacker advantage is small. Theorems similar to this were mentioned in literatures before [5,12,18]. The only difference is, here we concentrate on *output views* instead of *views*. In fact, for a fixed attacker $\mathcal{A}$, there is always an one to one mapping between any view and output view.

Theorem 1. *F_i,G_i be the probabilistic oracle algorithms. If for an attacker $\mathcal{A}$, the relation*

$$\Pr[\mathcal{OV}_{F_1,F_2}^{\mathcal{A}}=\mathcal{OV}]\geq(1-\varepsilon)\Pr[\mathcal{OV}_{G_1,G_2}^{\mathcal{A}}=\mathcal{OV}],$$

holds for all possible output views $\mathcal{OV}$, then we have, $\mathsf{Adv}_{\mathcal{A}}((F_1,F_2),(G_1,G_2))\leq\varepsilon$.

In general it is hard to show the necessary condition of Theorem 1 for all possible output views. Theorem 2 proves that it is sufficient to work with irreducible output views instead of all possible output views. In fact, one can reduce any output view to an irreducible output view and then can apply Theorem 1.

Theorem 2. *If there exists a simulator S^R consistent to a random oracle R with respect to JH-mode of operation, such that for any attacker $\mathcal{A}$ making at most q queries, the relation*

$$\Pr[\mathcal{OV}_{JH^\pi,\pi}^{\mathcal{A}}=\mathcal{OV}]\geq(1-\varepsilon)\Pr[\mathcal{OV}_{R,S^R}^{\mathcal{A}}=\mathcal{OV}],$$

holds for all possible irreducible *output views $\mathcal{OV}$ (with respect to $\mathcal{A}$); then for any attacker $\mathcal{A}$ making at most q queries, we have*

$$\mathsf{Adv}_{\mathcal{A}}((JH^\pi,\pi),(R,S^R))\leq\varepsilon.$$

Proof. This theorem differs from Theorem 1, only in the aspect that here probability distributions are close only for the irreducible output views. For any reducible output view $\mathcal{OV}$ and the corresponding attacker $\mathcal{A}$, let $\mathcal{V}$ be the view fixed by $\mathcal{OV}$ and $\mathcal{A}$. Let, $\mathcal{V}'$ be the view obtained by deleting the computable $\mathcal{O}_1$ queries and repeated $\mathcal{O}_2$ queries of $\mathcal{V}$. The input view $\mathcal{I}(\mathcal{V}')$ actually specifies a non-adaptive attacker $\mathcal{A}'$. The output view $\mathcal{OV}' = \mathcal{O}(\mathcal{V}')$ is actually an irreducible output view with respect to $\mathcal{A}'$. As, π is consistent to JH^π and S^R is consistent to R with respect to JH-mode of operation we have,

$$\Pr[\mathcal{OV}^{\mathcal{A}}_{JH^\pi,\pi} = \mathcal{OV}] = \Pr[\mathcal{OV}^{\mathcal{A}'}_{JH^\pi,\pi} = \mathcal{OV}']$$
$$\Pr[\mathcal{OV}^{\mathcal{A}}_{R,S^R} = \mathcal{OV}] = \Pr[\mathcal{OV}^{\mathcal{A}'}_{R,S^R} = \mathcal{OV}'].$$

Note, $\mathcal{A}'$ actually makes less number of queries compared to $\mathcal{A}$. Hence, even for reducible views, we have

$$\begin{aligned}\Pr[\mathcal{OV}^{\mathcal{A}}_{JH^\pi,\pi} = \mathcal{OV}] &= \Pr[\mathcal{OV}^{\mathcal{A}'}_{JH^\pi,\pi} = \mathcal{OV}'] \\ &\geq (1-\varepsilon)\Pr[\mathcal{OV}^{\mathcal{A}'}_{R,S^R} = \mathcal{OV}'] \\ &= (1-\varepsilon)\Pr[\mathcal{OV}^{\mathcal{A}}_{R,S^R} = \mathcal{OV}].\end{aligned}$$

So the required condition of Theorem 1 remains true. Now, by applying Theorem 1 we get the result. □

In many previous works e.g. [4,5,12] ideas similar to Theorem 2 have been used implicitly. But to our knowledge, we are the first to formalize it.

4 Indifferentiability Security Analysis of JH'

4.1 Simulator and Its Interpolation Probability

The simulator maintains one partial permutation $e_1 : \{0,1\}^{2n} \to \{0,1\}^{2n}$ initially empty, one partial function $e_1^* : (\{0,1\}^n)^* \to \{0,1\}^{2n}$ initialized with $e_1^*(\phi) = IV_1 \| IV_2$. It also maintains two sets C_1, C_2 initialized as $C_1 = \{IV_1\}$ and C_2 as empty. Let I_1 denotes the set of points on which e_1 is defined, O_1 denotes the output points of e_1. $FH, LH : \{0,1\}^{2n} \to \{0,1\}^n$ be the two functions outputting first n-bits and last n-bits of any $2n$-bit number respectively.

The goal of the simulator is to remain consistent to R with respect to JH-mode of operation while behaving like a random permutation. Before describing the simulator, we give some insight informally on how the simulator works.

1. In the partial permutation e_1, the simulator maintains its history.
2. In the partial function e_1^*, the simulator maintains the list of queries evaluatable by e_1 with respect to JH-mode of operation.
3. C_1 is the set of *first half* (first n-bits) of e_1^* outputs.
4. Even though e_1^* is evaluatable by the partial permutation e_1, it might happen that e_1 is also defined at some points which do not help in evaluating e_1^*. C_2 is the set of first half of such points.

5. The simulator makes sure, C_1 and C_2 always remain mutually exclusive.
6. Because of 5, there are no so called *accidents*. That means when the attacker is interacting with (R, S^R) and it wants to evaluate $\mathcal{O}_1(m_1\|\cdots\|m_\ell)$ through a series of $\mathcal{O}_2$ queries, she will always have to make a series of ℓ queries starting with $\mathcal{O}_2(IV_1, IV_2 \oplus m_1)$. The attacker can not hope to skip a query in the middle.

We note at any point of time, the following conditions hold.

$$|O_1| \leq q_2 + q_3 \text{ and } |I_1| \leq q_2 + q_3 \text{ and } |C_1 \cup C_2| \leq q_2 + q_3 \text{ and } |C_1| \leq q_2 + 1$$

Theorem 3. *For any attacker $\mathcal{A}$ against JH$'$ and any irreducible output view $\mathcal{OV}$ with respect to it, we have*

$$\Pr[\mathcal{OV}^{\mathcal{A}}_{R,S'^R} = \mathcal{OV}] \leq \frac{1}{2^{(2n-s)q_1+2n(q_2+q_3)}} \times \frac{1}{(1-\frac{2(q_2+q_3)}{2^{\min(s,n)}})^{q_2}} \times \frac{1}{(1-\frac{2(q_2+q_3)}{2^n})^{q_3}}$$

where $2^s > 2(q_2+q_3)2^{min(s,n)}$.

Proof. As $\mathcal{OV}$ is irreducible, R query outputs are independent of the other queries, hence R being a Random Function for q_1 many R queries we get the term $\frac{1}{2^{(2n-s)q_1}}$. For an $S'^R(+,\cdot,\cdot)$ queries, simulator is giving output as $w\|y$, there are two scenarios.

1. y is distributed uniformly over $\{0,1\}^{2n-s}$ and w is distributed uniformly over $\{0,1\}^s \setminus \{z \in \{0,1\}^s : z\|y \in O_1 \text{ or } FH(z\|y) \oplus (x' \oplus x_2) \in C_1 \cup C_2\}$.
2. $w\|y$ is distributed uniformly over $\{0,1\}^{2n} \setminus O_1$.

By Lemma 1 we know,

$$|\{z \in \{0,1\}^s : y\|z \in O_1\}| \leq |O_1| \leq (q_2+q_3).$$

On the other hand, using Lemma 1 here we have,

$$\begin{aligned}|\{z \in \{0,1\}^s : FH(z\|y) \oplus (x' \oplus x_2) \in C_1 \cup C_2\}| &\leq \frac{2^s}{2^{\min(s,n)}}|C_1 \cup C_2| \\ &\leq \frac{2^s}{2^{\min(s,n)}}(q_2+q_3).\end{aligned}$$

Hence, for $2^s > 2(q_2+q_3)2^{min(s,n)}$ and any $(w\|y) \in \{0,1\}^{2n}$ we have,

$$\begin{aligned}&\Pr[S'^R(+,\cdot,\cdot) \text{ query outputs } (w\|y)] \\ &\leq \max\left(\frac{1}{2^{2n-s}}\frac{1}{2^s - \frac{2^s}{2^{\min(s,n)}}(q_2+q_3) - (q_2+q_3)}, \frac{1}{2^{2n}-(q_2+q_3)}\right) \\ &\leq \frac{1}{2^{2n}}\frac{1}{(1-\frac{2(q_2+q_3)}{2^{\min(s,n)}})}\end{aligned}$$

For $S^R(-,\cdot,\cdot)$ query giving output as $z_1\|z_2$ we know,

1. z_1 is uniformly distributed over $\{0,1\}^n \setminus C_1$
2. z_2 is uniformly distributed over $\{0,1\}^n \setminus \{w \in \{0,1\}^n : z_1\|w \in I_1\}$

We know, $|C_1| \leq (q_2+1)$ and $|I_1| \leq (q_2+q_3)$. Hence, for any $(z_1\|z_2) \in \{0,1\}^{2n}$ we have

$$\begin{aligned}\Pr[S^R(-,\cdot,\cdot) \textit{ query outputs } (z_1\|z_2)] &\leq \frac{1}{2^n-(q_2+1)}\frac{1}{2^n-(q_2+q_3)} \\ &\leq \frac{1}{2^{2n}}\frac{1}{1-\frac{2(q_2+q_3)}{2^n}}\end{aligned}$$

Hence, all together we get

$$\Pr[\mathcal{OV}^{\mathcal{A}}_{R,S^R} = \mathcal{OV}] \leq \frac{1}{2^{(2n-s)q_1+2n(q_2+q_3)}} \times \frac{1}{(1-\frac{2(q_2+q_3)}{2^{\min(s,n)}})^{q_2}} \times \frac{1}{(1-\frac{2(q_2+q_3)}{2^n})^{q_3}}$$

□

Next we wish to show that our simulator is efficient. The condition $2^{\min(s,n)} > 4(q_2+q_3)2^n$ ensures the GOTO statement at Step 5 in forward query in Figure 4 gets executed with probability less than $\frac{1}{2}$ at each iteration. We also know $|O_1| \leq (q_2+q_3)$ and $|C_1 \cup C_2| \leq (q_2+q_3)$. Hence except with negligible probability, Step 5 takes at most $\mathcal{O}(q_2+q_3)$ time to satisfy the condition. The same argument holds for other GOTO statements as well. Hence we get the following result.

Theorem 4. *If $2^{\min(s,n)} > 4(q_2+q_3)$, the simulator S'^R takes at most $\mathcal{O}(q_2+q_3)$ time to answer any query (except with exponentially small probability).*

$S'^R(+, x_1, x_2)$

- IF $e_1(x_1\|x_2) = z$ RETURN z
- IF *there exists* M, *s.t* $e_1^*(M) = x_1\|x'$
  1. $m = x' \oplus x_2$
  2. $y = R(M\|m) \oplus \text{CHOPL}(m\|0^n)$
  3. $w \in_R \{0,1\}^s$
  4. $z = w\|y$
  5. IF ($z \in O_1$ OR $FH(z) \oplus m \in C_1 \cup C_2$)
     - GOTO 3
  6. $C_1 = C_1 \cup \{FH(z) \oplus m\}$
  7. $e_1^*(M\|m) = z \oplus (m\|0^n)$
  8. $e_1(x_1\|x_2) = z$
  9. RETURN z
- ELSE
  10. $z \in_R \{0,1\}^{2n}$
  11. IF $z \in O_1$
      - GOTO 10
  12. $e_1(x_1\|x_2) = z$
  13. $C_2 = C_2 \cup \{x_1\}$
  14. RETURN z

$S'^R(-, y_1, y_2)$

- IF *there exists* $z_1\|z_2$ *such that* $e_1(z_1\|z_2) = y_1\|y_2$
  - RETURN $z_1\|z_2$
- ELSE
  1. $z_1 \in_R \{0,1\}^n$
  2. IF $z_1 \in C_1$
     - GOTO 1
  3. $z_2 \in_R \{0,1\}^n$
  4. IF $z_1\|z_2 \in I_1$
     - GOTO 3
  5. $C_2 = C_2 \cup \{z_1\}$
  6. RETURN $z_1\|z_2$

Fig. 4. Simulator for JH′

4.2 Interpolation Probability of $\mathcal{OV}^{\mathcal{A}}_{JH'^{\pi},\pi}$

In Theorem 3 we have shown upper bound for $\Pr[\mathcal{OV}^{\mathcal{A}}_{R,S'^R} = \mathcal{OV}]$ for any irreducible output views $\mathcal{OV}$. The Theorem below gives a lower bound for $\Pr[\mathcal{OV}^{\mathcal{A}}_{JH'^{\pi},\pi} = \mathcal{OV}]$ for any irreducible output view $\mathcal{OV}$. Later we will apply Theorem 2 to prove the indifferentiability bound using these upper and lower bounds.

Theorem 5. *For any attacker $\mathcal{A}$ and any irreducible output view $\mathcal{OV}$ with respect to it, we have*

$$\Pr[\mathcal{OV}^{\mathcal{A}}_{JH'^{\pi},\pi} = \mathcal{OV}] \geq \frac{1}{2^{(2n-s)q_1+2n(q_2+q_3)}} \times (1 - \frac{2\sigma^2}{2^{2n}}) \times (1 - \frac{2q_1(q_1+q_2+q_3)}{2^{\min(s,n)}}).$$

The proof of the above theorem involves two steps. Starting with an attacker $\mathcal{A}$ against $JH'^{\pi} \equiv \text{CHOPL}(MD^{f^{\pi}})$ we construct another attacker $\mathcal{A}'$ against $MD^{f^{\pi}}$ which essentially makes same queries as $\mathcal{A}$ but has access to unchopped output view.

- First we define the notion of *MD-irreducible* view (irreducible view with respect to Merkle-Damgård mode of operation) and then we show for the output view $\mathcal{OV}_{MD}$ corresponding to any *MD-irreducible* view we actually have,

$$\Pr[\mathcal{OV}^{\mathcal{A}'}_{MD^{f^{\pi}},\pi} = \mathcal{OV}_{MD}] \geq \frac{1}{2^{2nq_1+2n(q_2+q_3)}} \times (1 - \frac{2\sigma^2}{2^{2n}})$$

- In Theorem 7 we show, given an irreducible output view $\mathcal{OV}$ and an attacker $\mathcal{A}$, if $\mathbf{OV}_{MD}$ is the set of all *MD-irreducible* output views for the attacker $\mathcal{A}'$ such that,

$$\Pr[\mathcal{OV}^{\mathcal{A}}_{JH'^{\pi},\pi} = \mathcal{OV} | \mathcal{OV}^{\mathcal{A}'}_{MD^{f^{\pi}},\pi} = \mathcal{OV}_{MD}] = 1$$

  for all $\mathcal{OV}_{MD} \in \mathbf{OV}_{MD}$; then

$$|\mathbf{OV}_{MD}| \geq 2^{sq_1} \times (1 - \frac{2q_1(q_1+q_2+q_3)}{2^{\min(s,n)}})$$

The above two results will readily imply Theorem 5.

Definition 14. *The set of relations*

$$MD^{f^{\mathcal{O}_2}}(M_1\|m_1) = g_1, \ldots, MD^{f^{\mathcal{O}_2}}(M_{q_1}\|m_{q_1}) = g_{q_1}$$
$$\mathcal{O}_2(x^1_1, x^2_1) = (y^1_1, y^2_1), \ldots, \mathcal{O}_2(x^1_{q_2+q_3}, x^2_{q_2+q_3}) = (y^1_{q_2+q_3}, y^2_{q_2+q_3}) \ldots \textit{ Rel A}$$

is MD-irreducible *if,*

1. *$g_1 \oplus (m_1\|0^n), \ldots, g_{q_1} \oplus (m_{q_1}\|0^n), y^1_1\|y^2_1, \ldots, y^1_{q_2+q_3}\|y^2_{q_2+q_3}$ are all different.*
2. *For $i = 1, \ldots, q_1$, one of the following two conditions hold*
   (a) *$FH(g_i)$ is different from $x^1_1, \ldots, x^1_{q_2+q_3}$ and IV_1.*
   (b) *Σ be the set of all message blocks present in $MD^{f^{\mathcal{O}_2}}$ queries. If $FH(g_i) = IV_1$, then $LH(g_i) \oplus IV_2 \notin \Sigma$. If $FH(g_i) = x^1_j$ for some $1 \leq j \leq q_2 + q_3$, then $LH(g_i) \oplus x^2_j \notin \Sigma$.*

3. $M_1\|m_1, \ldots, M_{q_1}\|m_{q_1}$ *are not evaluatable by the relations*

$$\mathcal{O}_2(x_1^1, x_1^2) = (y_1^1, y_1^2), \ldots, \mathcal{O}_2(x_{q_2+q_3}^1, x_{q_2+q_3}^2) = (y_{q_2+q_3}^1, y_{q_2+q_3}^2)$$

with respect to MD*-mode of operations based on* $f^{\mathcal{O}_2}$.

We also say the tuple,

$$v = ((M_1\|m_1, g_1), \ldots, (M_{q_1}\|m_{q_1}, g_{q_1}), (x_1^1, x_1^2, y_1^1, y_1^2), \ldots, (x_{q_2+q_3}^1, x_{q_2+q_3}^2, y_{q_2+q_3}^1, y_{q_2+q_3}^2))$$

is MD-irreducible *if and only if the corresponding Rel-A is MD-irreducible.*

The definition above is similar to the definition of irreducible view (Definition 12). But here we are interested in the view without any chopping. Note, condition 2 ensures $M_i\|m_i$ is not evaluatable even with the help of the relations $MD^{f^{\mathcal{O}_2}}(M_j\|m_j) = h_j$ for $j \neq i$. Loosely speaking, the Theorem below gives a lower bound of the probability of getting a particular MD-irreducible tuple v, when a attacker interacts with (MD^{f^π}, π).

Theorem 6. *Let a tuple*

$$v = ((M_1\|m_1, g_1), \ldots, (M_{q_1}\|m_{q_1}, g_{q_1}), (x_1^1, x_1^2, y_1^1, y_1^2), \ldots, (x_{q_2+q_3}^1, x_{q_2+q_3}^2, y_{q_2+q_3}^1, y_{q_2+q_3}^2))$$

is MD-irreducible*, then the number of permutations* π *such that,*

$$MD^{f^\pi}_{IV_1\|IV_2}(M_1\|m_1) = g_1, \ldots, MD^{f^\pi}_{IV_1\|IV_2}(M_{q_1}\|m_{q_1}) = g_{q_1}$$
$$\pi(x_1^1, x_1^2) = (y_1^1, y_1^2), \ldots, \pi(x_{q_2+q_3}^1, x_{q_2+q_3}^2) = (y_{q_2+q_3}^1, y_{q_2+q_3}^2) \ldots \textit{Rel B}$$

is at least

$$\frac{|\Pi|}{2^{2nq_1+2n(q_2+q_3)}} \times (1 - \frac{2\sigma^2}{2^{2n}}),$$

where $|\Pi| = (2^{2n})!$ *is the total number of permutations from* $\{0,1\}^{2n}$ *to* $\{0,1\}^{2n}$ *and* σ *is the total number of message blocks queried. Also for a MD-irreducible tuple* v*, the probability that Rel B holds is at least*

$$\frac{1}{2^{2nq_1+2n(q_2+q_3)}} \times (1 - \frac{2\sigma^2}{2^{2n}}),$$

when π *is a random permutation.*

Proof. Let D be the set of all elements from $(\{0,1\}^n)^+$ whose $MD^{f^\pi}_{IV_1\|IV_2}$ values are determined from the relations

$$\pi(x_1^1, x_1^2) = (y_1^1, y_1^2), \ldots, \pi(x_{q_2+q_3}^1, x_{q_2+q_3}^2) = (y_{q_2+q_3}^1, y_{q_2+q_3}^2).$$

Since v is MD-irreducible, $M_i \| m_i \notin D$ for all $1 \leq i \leq q_1$. let P denote the set of all nonempty prefixes of M_i's. More precisely,

$$P = \{M \in (\{0,1\}^n)^+ : M \textit{ is prefix of } M_i \textit{ for some } 1 \leq i \leq q_1\}.$$

We enumerate the set $P \setminus D \equiv \{N_1, \ldots, N_{\sigma'}\}$. Note that, $|P| + q_1 \leq \sum_i \|M_i\|$. Now, we have

$$\sigma = q_2 + q_3 + \sum_i \|M_i\| \geq q_2 + q_3 + |P| + q_1 \geq q_1 + q_2 + q_3 + \sigma' \equiv \sigma''$$

Similar to the proof of Lemma 1 in [5], we can choose outputs of $MD^{f^{\mathcal{O}_2}}_{IV_1 \| IV_2}(N_1), \ldots, MD^{f^{\mathcal{O}_2}}_{IV_1 \| IV_2}(N_{\sigma'})$ in at least

$$(2^{2n} - 2(q_1+q_2+q_3))(2^{2n} - 2(q_1+q_2+q_3+1)) \ldots (2^{2n} - 2(q_1+q_2+q_3+\sigma'-1))$$

ways. (In the negative term, the factor 2 comes because, any output value should not be same as other output values and the next input value induced by the output value should not be same as other input values.) Hence,

$$\begin{aligned}
&|\{\pi : \{0,1\}^{2n} \to \{0,1\}^{2n} \textit{ such that } \pi \textit{ is a permutation and satisfies Rel B}\}| \\
&\geq (2^{2n} - \sigma'')! \times \prod_{i=0}^{\sigma'-1} (2^{2n} - 2(q_1+q_2+q_3+i)) \\
&\geq \frac{(2^{2n})!}{2^{2n\sigma''}} \times 2^{2n\sigma'} \times (1 - \frac{2{\sigma'}^2}{2^{2n}}) \geq \frac{|\pi|}{2^{2nq_1 + 2n(q_2+q_3)}} \times (1 - \frac{2\sigma^2}{2^{2n}})
\end{aligned}$$

□

Definition 15. *With respect to an irreducible view*

$$\mathcal{V} = ((M_1\|m_1, h_1), \ldots, (M_{q_1}\|m_{q_1}, h_{q_1}), (x_1^1, x_1^2, y_1^1, y_1^2), \ldots, (x_{q_2+q_3}^1, x_{q_2+q_3}^2, y_{q_2+q_3}^1, y_{q_2+q_3}^2))$$

an MD-irreducible tuple v is said to be CHOPL-matching *if*

$$v = ((M_1\|m_1, w_1\|h_1), \ldots, (M_{q_1}\|m_{q_1}, w_{q_1}\|h_{q_1}), (x_1^1, x_1^2, y_1^1, y_1^2), \ldots, (x_{q_2+q_3}^1, x_{q_2+q_3}^2, y_{q_2+q_3}^1, y_{q_2+q_3}^2)),$$

for some q_1-tuple $w = (w_1, \ldots, w_{q_1})$.

Let $M'_{\mathcal{V}}$ be the set of all such CHOPL-*matching* MD-irreducible tuples.

Theorem 7. *For any irreducible view*

$$\mathcal{V} = ((M_1\|m_1, h_1), \ldots, (M_{q_1}\|m_{q_1}, h_{q_1}), (x_1^1, x_1^2, y_1^1, y_1^2), \ldots, (x_{q_2+q_3}^1, x_{q_2+q_3}^2, y_{q_2+q_3}^1, y_{q_2+q_3}^2))$$

we have,

$$|M'_{\mathcal{V}}| \geq 2^{sq_1} \times (1 - \frac{q_1(q_1+q_2+q_3)}{2^{\min(s,n)}}).$$

For the proof of this theorem, the reader is referred to the full version of the paper.

Now we are ready to prove Theorem 5, with help of Theorem 6 and Theorem 7.Let $\mathcal{V}$ be the irreducible view determined by $\mathcal{A}$ and irreducible output view $\mathcal{OV}$. Consider an Attacker $\mathcal{A}'$, which makes queries at the same input points as of $\mathcal{A}$, but has access to $MD^{f^{\mathcal{O}_2}}$ instead of $JH'^{\mathcal{O}_2}$. Hence,

$$\Pr[\mathcal{OV}^{\mathcal{A}}_{JH'^{\pi},\pi} = \mathcal{OV}] = \Pr[\mathcal{V}^{\mathcal{A}}_{JH'^{\pi},\pi} = \mathcal{V}] = \sum_{v \in M_{\mathcal{V}}} \Pr[\mathcal{V}^{\mathcal{A}'}_{MD^{f^{\pi}},\pi} = v]$$

$$\geq \sum_{v \in M_{\mathcal{V}}} \frac{1}{2^{2nq_1+2n(q_2+q_3)}} \times (1 - \frac{2\sigma^2}{2^{2n}})$$

$$\geq \frac{1}{2^{2nq_1+2n(q_2+q_3)}} \times (1 - \frac{2\sigma^2}{2^{2n}}) \times 2^{sq_1} \times (1 - \frac{2q_1(q_1+q_2+q_3)}{2^{\min(s,n)}})$$

$$= \frac{1}{2^{(2n-s)q_1+2n(q_2+q_3)}} \times (1 - \frac{2\sigma^2}{2^{2n}}) \times (1 - \frac{2q_1(q_1+q_2+q_3)}{2^{\min(s,n)}}) \qquad \square$$

4.3 Indifferentiability Security Bound

We are now ready to prove the main result of this section. For any attacker $\mathcal{A}$, making at most q_1, q_2, q_3 queries to the oracles $\mathcal{O}_1, \mathcal{O}_2(+,\cdot,\cdot), \mathcal{O}_2(-,\cdot,\cdot)$ respectively we show an upper bound for $\mathsf{Adv}_{\mathcal{A}}$.

Theorem 8. *The JH'^{π}-construction (with $(2n-s)$-bit output) based on a random permutation π is $(\mathcal{O}(q_2+q_3)), q_1, q_2+q_3, \epsilon)$ indifferentiable from a random oracle R, with*

$$\epsilon \leq \frac{2\sigma^2}{2^{2n}} + \frac{2q_3(q_2+q_3)}{2^n} + \frac{2q_2(q_2+q_3)+2q_1(q_1+q_2+q_3)}{2^{\min(s,n)}},$$

where σ is the maximum number of message blocks queried, q_1 is the maximum number queries to JH'^{π} or R, q_2+q_3 is the maximum number of queries to π, π^{-1} or $S'^R(+,\cdot,\cdot), S^R(-,\cdot,\cdot)$. Here we also assume, $q_2+q_3 < 2^{\min(s,n)}/4$.

Proof. For any attacker $\mathcal{A}$ and an irreducible output view $\mathcal{OV}$ from Theorem 3 and Theorem 5 we have,

$$\Pr[\mathcal{OV}^{\mathcal{A}}_{JH^{\pi},\pi} = \mathcal{OV}]$$

$$\geq \left(1 - \left(\frac{2\sigma^2}{2^{2n}} + \frac{2q_3(q_2+q_3)}{2^n} + \frac{2q_2(q_2+q_3)+2q_1(q_1+q_2+q_3)}{2^{\min(s,n)}}\right)\right)$$

$$\times \Pr[\mathcal{OV}^{\mathcal{A}}_{R,S'^R} = \mathcal{OV}]$$

Now, applying Theorem 2 we get the required result. $\square$

When maximum query length ℓ is smaller than $2^{n/2}$, for any attacker $\mathcal{A}$ (making at most q many queries) against the JH′ construction we have

$$\mathsf{Adv}_{\mathcal{A}} = \mathcal{O}(\frac{q^2}{2^{\min(s,n)}})$$

5 Indifferentiability Security Analysis of JH_P

In this section we prove the indifferentiability of JH mode of operation with padding.

5.1 Simulator and Its Interpolation Probability

We describe our simulator in Fig 5. Similar to previous section, the following notation we used in describing the simulator.

- Partial permutation $e : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$, initially empty. I denotes set of points where e is defined and O denotes the output points of e.
- Partial function $e^* : (\{0,1\}^n)^* \rightarrow \{0,1\}^{2n}$ initialized to $e^*(\phi) = IV_1 \| IV_2$.
- Set $C \subseteq \{0,1\}^n$ initialized to $C = \{IV_1\}$ is the FH (first half) of e^* outputs.

For a padding rule $P = (\text{PAD}, \text{DEPAD})$ and $M \in (\{0,1\}^n)^+$, we recall $LB(M) \subseteq \{0,1\}^n$ is defined as $\{m \in \{0,1\}^n : \text{DEPAD}(M\|m) \neq \perp\}$. As in case of the actual JH padding rule we assume, $|LB(M)| \leq 2$.

We recall the design philosophy behind the JH' simulator from Section 4.1. Over there the simulator was maintaining a list of *evaluatable* queries and their non-chopped outputs in the partial permutation e_1^*. When the simulator receives some query the goal of the simulators are three fold.

1. Give a random output keeping in mind the permutation property.
2. Do not create some new evaluatable query unless forced to do so. That means output of the simulator will never create a new evaluatable query with the exception of the following scenario.
3. It might happen, only the input of the simulator forces another new evaluatable query. (This happens if attacker is trying to find some $\mathcal{O}_1$ query output through $\mathcal{O}_2$ query.) If this happens, then adjust the output of the simulator so that it remains *consistent* to R, w.r.t. the new evaluatable query.

One crucial point is, during one simulator query the simulator must prevent creation of more than one evaluatable query. Otherwise, the simulator can not remain consistent to both of them. In forward queries to JH' simulator with $s = n$, when the attacker has forced creation of one new evaluatable query the LH (last half) of the possible output gets fixed by R response of that evaluatable query, but the simulator has control over FH output with which it makes sure, another evaluatable query is not created.

Here the situation is reversed. FH gets fixed by R, the simulator has control only over LH. This is problematic, because only FH can lead to creation of more evaluatable queries (with one more message block after the current evaluatable query). In fact, in Section 6 the attacker against JH mode operation (without length padding at last block) exploits this fact. But the simulator can play with LH to change the actual evaluatable query (even though it can not prevent the creation.) By doing so, the simulator ensures the new evaluatable query is not a valid padded message, hence for that query the simulator does not need to be consistent with R. The simulator also need to be careful such that no new evaluatable queries of length (current evaluatable query length + 2) or more are created. However, that can easily be handled.

$S^R(+, x_1, x_2)$

1. IF $e(x_1\|x_2) = z$ RETURN z
2. IF *there exists* M, s.t. $e^*(M) = x_1\|x'$
   (a) $m = x' \oplus x_2$
   (b) IF $M \neq \phi$ AND $m \in LB(M)$
      i. $y = R(\text{DEPAD}(M\|m)) \oplus \text{CHOPR}(m\|0^n)$
      ii. $w \in_R \{0,1\}^s$
      iii. $z = y\|w$
   (c) ELSE
      i. $z \in_R \{0,1\}^{2n}$
   (d) IF $z \in O$ GOTO 2b
   (e) $e^{*\prime} = e^*$
   (f) $C' = C$
   (g) FOR EACH $i_1\|i_2 \in I \cup \{x_1\|x_2\}$
      i. IF $FH(z) \oplus m \neq i_1$ CONTINUE
      ii. IF $LH(z) \oplus i_2 \in LB(M\|m)$
         – GOTO 2b
      iii. IF $i_1\|i_2 = x_1\|x_2$
         – $o_1\|o_2 = z$
      iv. ELSE
         – $o_1\|o_2 = e(i_1\|i_2)$
      v. $e^{*\prime}(M\|m\|LH(z) \oplus i_2) = (o_1 \oplus LH(z) \oplus i_2)\|o_2$
      vi. $C' = C' \cup \{o_1 \oplus LH(z) \oplus i_2\}$
      vii. FOR EACH $i_1'\|i_2' \in I \cup \{x_1\|x_2\}$
         – IF $LH(z) \oplus i_2 = o_1 \oplus i_1'$
            • GOTO 2b
   (h) $e^* = e^{*\prime}$
   (i) $C = C'$
   (j) $e^*(M\|m) = z \oplus (m\|0^n)$
   (k) $C = C \cup \{FH(z) \oplus m\}$
3. ELSE
   (a) $z \in_R \{0,1\}^{2n}$
   (b) IF $z \in O$ GOTO 3a
4. $e(x_1\|x_2) = z$
5. RETURN z

$S^R(-, y_1, y_2)$

1. IF *there exists* $z_1\|z_2$ *such that* $e(z_1\|z_2) = y_1\|y_2$
   – RETURN $z_1\|z_2$
2. ELSE
   (a) $z_1 \in_R \{0,1\}^n$
   (b) IF $z_1 \in C$
      – GOTO 2a
   (c) $z_2 \in_R \{0,1\}^n$
   (d) IF $z_1\|z_2 \in I$
      – GOTO 2c
   (e) $e(z_1\|z_2) = y_1\|y_2$
   (f) RETURN $z_1\|z_2$

Fig. 5. Simulator for JH with padding

The next two theorems describe the running time and interpolation probability upper bound corresponding to the simulator.

Theorem 9. *For any attacker $\mathcal{A}$ against JH_P^π mode of operation and any irreducible output view $\mathcal{OV}$ with respect to it, we have*

$$\Pr[\mathcal{OV}^{\mathcal{A}}_{R,S^R} = \mathcal{OV}] \leq \frac{1}{2^{(2n-s)q_1 + 2n(q_2+q_3)}} \times \frac{1}{(1 - \frac{(q_2+q_3+3)^2}{2^s})^{q_2}} \times \frac{1}{(1 - \frac{(q_2+q_3+1)^2}{2^n})^{q_3}}$$

when $(q_2 + q_3 + 3)^2 < 2^{\min(s,n)}$.

Theorem 10. *If* $2(q_2+q_3+3)^2 < 2^{\min(s,n)}$*, the simulator* S^R *takes at most* $\mathcal{O}((q_2+q_3)^2)$ *time to answer any query (except with exponentially negligible probability).*

The proof of two theorems above is similar to the proof of Theorem 3 and Theorem 4. Due to space constraint we skip the proof and refer the reader to full version of the paper.

5.2 Interpolation Probability of $\mathcal{OV}^{\mathcal{A}}_{JH^{\pi}_{P},\pi}$

The following theorem is analogous to Theorem 5, used in Section 4.

Theorem 11. *For any attacker* $\mathcal{A}$ *and any irreducible output view* $\mathcal{OV}$ *with respect to it, we have*

$$\Pr[\mathcal{OV}^{\mathcal{A}}_{JH^{\pi}_{P},\pi} = \mathcal{OV}] \geq \frac{1}{2^{(2n-s)q_1+2n(q_2+q_3)}} \times (1-\frac{2\sigma^2}{2^{2n}}) \times (1-\frac{2\sigma q_1(q_1+q_2+q_3)}{2^s}).$$

For the proof of above theorem we refer the reader to the full version.

5.3 Indifferentiability Security Bound

Theorem 12. *The* JH^{π}_{P} *mode of operation (with* $(2n-s)$*-bit output) based on a random permutation* π *is* $(\mathcal{O}((q_2+q_3)^2), q_1, q_2+q_3, \epsilon)$ *indifferentiable from a random oracle* R*, with*

$$\epsilon \leq \frac{2\sigma^2}{2^{2n}} + \frac{q_2(q_2+q_3+3)^2}{2^s} + \frac{q_3(q_2+q_3+1)^2}{2^n} + \frac{2\sigma q_1(q_1+q_2+q_3)}{2^s},$$

where σ *is the maximum number of message blocks queried,* q_1 *is the maximum number queries to* JH^{π}_{P} *or* R*,* q_2+q_3 *is the maximum number of queries to* π, π^{-1} *or* $S'^R(+,\cdot,\cdot), S^R(-,\cdot,\cdot)$*. Here we also assume,* $2(q_2+q_3+3)^2 < 2^{\min(s,n)}$*.*

Under reasonable assumptions, for an attacker making at most q queries with total σ many compression function invocations we have

$$\mathsf{Adv}_{\mathcal{A}} = \mathcal{O}(\frac{\sigma^2}{2^{2n}} + \frac{q^3}{2^n} + \frac{q^2\sigma}{2^s}).$$

6 Distinguisher $\mathcal{A}$ for JH without Length Padding at Last Block

Recall that the compression function of JH is based on a fixed permutation π. On input of the n-bit message block m and $2n$-bit chaining value $h_1||h_2$ the compression function outputs $f(m,h_1,h_2) = \pi(h_1, h_2+m)+m||0^n$. JH applies chopped Merkle-Damgård transformation and outputs first t ($t = 2n-s$) bits of the output of final compression function. Here s denotes the number of chopped bits.

In case of JH-n, we have $s = n$. Our distinguisher first queries $h = C^{\pi}(M)$ with a random n-bit message M. The distinguisher appends 0^n with h and queries $t_1\|t_2 = \pi(+, h\|0^n)$. Note that when the distinguisher is interacting with (π, C^{π}), the second π query made by $C^{\pi}(M||M_2)$ will be on the input $(h||z)$ where z is the last n bit

output of $\pi(+, IV_1, IV_2 \oplus M_1)$ xor-ed with M_2. So if we set M_2 to be the last n bit output of $\pi(+, IV_1, IV_2 \oplus M_1)$ then $z = 0^n$. Note that in case of JH with length padding, we could not choose M_2 this way as the length block is fixed. To get M_2, the distinguisher queries $z_1 \| z_2 = \mathcal{O}_2(+, IV_1 \| IV_2 \oplus M)$. Now $\mathcal{D}$ sets $M_2 = z_2$ and queries $h_2 = C^{\pi}(M \| z_2)$. Finally the distinguisher checks whether $h_2 = t_1 \oplus z_2$. Formal algorithm of the distinguisher is described in Figure 6(a).

Theorem 13. *If the simulator $\mathcal{S}$ makes at most k many $\mathcal{R}$ queries for answering a single query, then* $\mathsf{Adv}_{\mathcal{A}} \geq 1 - \frac{2k+1}{2^n}$.

Distinguisher A

1. $M \in_R \{0,1\}^n$.
2. $h = \mathcal{O}_1(M)$.
3. $t_1 \| t_2 = \mathcal{O}_2(+, h \| 0^n)$.
4. $z_1 \| z_2 = \mathcal{O}_2(+, IV_1 \| IV_2 \oplus M)$.
5. $h_2 = \mathcal{O}_1(M \| z_2)$.
6. IF $t_1 \neq h_2 \oplus z_2$
   - return 1.
7. return 0.

(a)

Distinguisher for JH′ without length padding at last block

- Choose distinct n-bit numbers $m_1, \ldots, m_k$
- For $i = 1, \ldots, k$
  $y_i^1 \| y_i^2 = \mathcal{O}_2(+, IV_1 \| IV_2 \oplus m_i)$
- If for $i = 1, \ldots, k$, $(y_1^i \oplus m_i)$'s are distinct return 1
- else
  - Find distinct j_1, j_2 such that $(y_{j_1}^1 \oplus m_{j_1}) = (y_{j_2}^1 \oplus m_{j_2})$
  - $m \in_R \{0,1\}^n$
  - $x_1 = \mathcal{O}_1(m_{j_1} \| (m \oplus y_{j_1}^2))$
  - $x_2 = \mathcal{O}_1(m_{j_2} \| (m \oplus y_{j_2}^2))$
  - if $x_1 \oplus \textsc{ChopL}((m \oplus y_{j_1}^2) \| 0^n) \neq x_2 \oplus \textsc{ChopL}((m \oplus y_{j_2}^2) \| 0^n)$
    - return 1
- return 0

(b)

Fig. 6. (a): Distinguisher for JH-n without length padding, (b): Distinguisher for JH′ without length padding

7 Distinguisher for JH′

In this section, we show one distinguisher with $\Omega(2^{n/2})$ many queries, which is successful against any simulator with non-negligible probability. Hence, when maximum query length ℓ is bounded by $2^{n/2}$, we get tight security bound.

The distinguisher has access to two oracles $\mathcal{O}_1, \mathcal{O}_2$ and is trying to differentiate between the two scenarios whether $(\mathcal{O}_1, \mathcal{O}_2)$ is (JH^{π}, π) or (R, S^R). Formal description of our distinguisher is given in Fig 6(b). The success probability of the distinguisher is established by following theorem. For a proof we refer the reader to the full version of the paper.

Theorem 14. *With $k = \Omega(2^{n/2})$, $\mathsf{Adv}_{\mathcal{A}}$ is non-negligible for any simulator S.*

Note, if we use CHOPR instead of CHOPL then the same attack actually applies for the original JH mode of construction without length padding at last block as well.

8 Preimage Attack on JH

In this section we demonstrate a preimage attack on Merkle-Damgård based the JH compression function. As the JH hash output is a part of MD^{f^π}, having preimage attack on MD^{f^π} immediately translate a preimage attack on JH hash function. We use multicollision as it has been used in [16]. Let $Q(r)$ denote expected number of queries to get r-collision of a n-bit random oracle. In [20], it was shown that $Q(r) \sim 2^{n(r-1)/r}(r!)^{1/r}$. In [16], a preimage attack on JH has been shown based on multicollision of the forward direction of the JH mode. The query-complexity of the attack is $\mathcal{O}(Q(r))$ where r is a solution of the equation $r^{1/2}Q(r) = 2^n$. We use two sided multicollision (both from forward and backward direction) to improve the attack complexity little bit. The new query-complexity is $\mathcal{O}(Q(r))$ where $Q(r)r = 2^n$. Now we describe our preimage attack for MD^{f^π} where f^π is the compression function defined in JH based on a permutation π (see Fig 2). Let $h\|h' \in \{0,1\}^{2n}$ be a randomly chosen target. Note that given any m, h, h', $f^{-1}(h, h', m)$ is easily computable by making only one π^{-1} query.

1. Choose an arbitrary message block M_5 with correct padding, and compute $H_4 := h_4\|h'_4 = f^{-1}(h\|h', M_5)$.
2. Compute $Q(r)$ candidates for $H_3 = f^{-1}(H_4, M_4)$ to obtain r-collision on the last half of H_3. This is possible since we assume that π is a random permutation. Let L be the list of r many H_3's such that $LH(H_3)$'s are identical to say h'_3.
3. Similarly we do it for forward computation of f for the first message block M_1. We have a list L' of r values of H_1 such that $FH(H_1) = h_1$ for all $H_1 \in L'$.
4. Now we run a kind of meet-in-the-middle attack for the chaining value H_2. We compute $Q(r)$ values of $\pi(h_1, h'_1)$ and $\pi^{-1}(h_3, h'_3)$ for $Q(r)$ choices of h'_1 and h_3. Note that h_1 and h'_3 are fixed from the previous two steps. Find h'_1 and h_3 such that

   $$FH(\pi(h_1, h'_1) \oplus \pi^{-1}(h_3, h'_3)) \oplus h'_1 \in L',\ LH(\pi(h_1, h'_1) \oplus \pi^{-1}(h_3, h'_3)) \oplus h_3 \in L.$$

   For any pair (h'_1, h_3) the probability of the above event is $\frac{r^2}{2^{2n}}$. Since we have $Q^2(r)$ such pairs we can expect one such pair (h'_1, h_3) satisfying the above condition provided r is the at least the solution of the equation $rQ(r) = 2^n$. Let $M_2 = FH(\pi(h_1, h'_1) \oplus \pi^{-1}(h_3, h'_3))$, $M_3 = LH(\pi(h_1, h'_1) \oplus \pi^{-1}(h_3, h'_3))$. Moreover we choose M_1 and M_4 from the list L' and L respectively such that $H_1 := h_1\|h'_1$ and $H_3 := h_3\|h'_3$ are the corresponding chaining value.

It is easy to verify that $MD^{f^\pi}(M_1\|M_2\|M_3\|M_4\|M_5) = h\|h'$. In [16], $r = 51$ to satisfy the equation $r^{1/2}Q(r) = 2^{512}$ where $n = 512$. The query complexity of their attack is roughly 2^{510}. We can choose $r = 46$ a solution of $rQ(r) \sim 2^{512}$. In this case the query complexity of π and π^{-1} is roughly 2^{507}. Compared with the previous preimage attack, it does not have significant reduction in complexity. However, asymptotically it has non-trivial reduction finding preimage of JH. The solution of r in $r^{1/2}Q(r) = 2^n$ is larger than that of $rQ(r) = 2^n$. Since $Q(r)$ is strictly increasing function in r our attack complexity is asymptotically less than that of [16]. However, we do not know any concrete forms of the query complexities for these two attacks.

9 Conclusion

14 candidates has been selected for the second round of SHA3 competition. Over the next few years one of these candidates will win and become the next hash function standard. In this paper we considered the security of a second round candidate, JH, in the indifferentiability framework. We showed that under the assumption that the underlying permutation is a random permutation, JH mode of operation with specific padding rule is indifferentiable from a Random Oracle. We also considered a modified design of JH, called JH′ , by chopping different bits. We analyzed the indifferentiability of JH' mode with optimal bounds. We also presented a distinguisher for JH mode without length padding (with any other prefix free padding). Finally we constructed a preimage attack with 2^{507} query which is better than the complexity of known preimage attacks. However, our attack doesnot pose any serious threat to JH hash function.

Acknowledgements

We sincerely thank Jean Sébastien Coron for his valuable comments on initial drafts of this paper. We also thank anonymous reviewers for their thoughtful suggestions.

References

1. Bellare, M., Rogaway, P.: Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols. In: 1st Conference on Computing and Communications Security, pp. 62–73. ACM, New York (1993)
2. Bellare, M., Ristenpart, T.: Multi-Property-Preserving Hash Domain Extension and the EMD Transform. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 299–314. Springer, Heidelberg (2006)
3. Barke, R.: On the Security of Iterated MACs. Diploma Thesis 2003. ETH Zurich (2003)
4. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer, Heidelberg (2008)
5. Chang, D., Nandi, M.: Improved Indifferentiability Security Analysis of chopMD Hash Function. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 429–443. Springer, Heidelberg (2008)
6. Coron, J.S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgard Revisited: How to Construct a Hash Function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (2005)
7. Coron, J.S., Patarin, J., Seurin, Y.: The Random Oracle Model and the Ideal Cipher Model Are Equivalent. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 1–20. Springer, Heidelberg (2008)
8. Damgård, I.: A Design Principles for hash functions. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (1990)
9. Dodis, Y., Pietrzak, K., Puniya, P.: A new mode of operation for block ciphers and length-preserving MACs. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 198–219. Springer, Heidelberg (2008)
10. Dodis, Y., Reyzin, L., Rivest, R., Shen, E.: Indifferentiability of Permutation-Based Compression Functions and Tree-Based Modes of Operation, with Applications to MD6. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 104–121. Springer, Heidelberg (2009)

11. Dodis, Y., Ristenpart, T., Shrimpton, T.: Salvaging Merkle-Damgård for Practical Applications. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 371–388. Springer, Heidelberg (2009)
12. Chang, D., Lee, S., Nandi, M., Yung, M.: Indifferentiable security analysis of popular hash functions with prefix-free padding. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 283–298. Springer, Heidelberg (2006)
13. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: STOC 1998, ACM, New York (1998)
14. Maurer, U.: Indistinguishability of Random Systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002)
15. Maurer, U., Renner, R., Holenstein, C.: Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004)
16. Mendel, F., Thomsen, S.: An Observation on JH-512, `http://ehash.iaik.tugraz.at/uploads/d/da/Jh_preimage.pdf`
17. Nielsen, J.: Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Noncommitting Encryption Case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, p. 111. Springer, Heidelberg (2002)
18. Nandi, M.: A Simple and Unified Method of Proving Indistinguishability. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 317–334. Springer, Heidelberg (2006)
19. SHA 3 official website, `http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/submissions_rnd1.html`
20. Suzuki, K., Tonien, K.D., Kurosawa, K., Toyota, K.: Birthday Paradox for Multi-collisions. In: Rhee, M.S., Lee, B. (eds.) ICISC 2006. LNCS, vol. 4296, pp. 29–40. Springer, Heidelberg (2006)
21. Wu, H.: The Hash Function JH. Submission to NIST (2008), `http://icsd.i2r.a-star.edu.sg/staff/hongjun/jh/jh.pdf`
22. Vaudenay, S.: Decorrelation: A Theory for Block Cipher Security. J. Cryptology 16(4), 249–286 (2003)