

Adaptive Preimage Resistance and Permutation-based Hash Functions

Jooyoung Lee, Je Hong Park

The Attached Institute of Electronics and Telecommunications Research Institute
Yuseong-gu, Daejeon, Korea 305-390
jlee05@ensec.re.kr, jhpark@ensec.re.kr

Abstract. In this paper, we introduce a new notion of security, called *adaptive preimage resistance*. We prove that a compression function that is collision resistant and adaptive preimage resistant can be combined with a public random function to yield a hash function that is indistinguishable from a random oracle. Specifically, we analyze adaptive preimage resistance of $2n$ -bit to n -bit compression functions that use three calls to n -bit public random permutations. This analysis also provides a simpler proof of their collision resistance and preimage resistance than the one provided by Rogaway and Steinberger [19]. By using such compression functions as building blocks, we obtain permutation-based pseudorandom oracles that outperform the Sponge construction [4] and the MD6 compression function [9] both in terms of security and efficiency.

Keywords: hash function, indistinguishability, blockcipher, provable security

1 Introduction

A cryptographic hash function takes a message of arbitrary length, and returns a bit string of fixed length. The most common way of hashing variable length messages is to iterate a fixed-size compression function according to the Merkle-Damgård paradigm. The underlying compression function can either be constructed from scratch, or be built upon off-the-shelf cryptographic primitives. For example, the Whirlpool hash function, adopted as ISO/IEC 10118-3 standard, is based on the Miyaguchi-Preneel construction using a modified version of AES. Compression functions based on blockciphers have been widely studied [6, 11, 12, 15, 17, 24, 25]. Recently, researchers has begun to pay attention to building compression functions from fixed key blockciphers, where just a small number of constants are used as keys [4, 5, 18, 19, 22]. Since each key of a blockcipher defines an independent random permutation in the ideal cipher model, such compression functions are often called *permutation-based*. Permutation-based compression functions have an obvious advantage over conventional blockcipher-based ones, since fixing the keys allows to save computational overload for key scheduling.

The security of a permutation-based compression function is usually analyzed in the ideal cipher model. The goal of the analysis has been to prove security notions such as collision resistance and preimage resistance in an information theoretic sense. However, in many cryptographic protocols including OAEP [2], PSS [3] and HMQV [13], to name a few, those notions do not seem to suffice as the security requirement for the underlying hash functions. Instead, their security is guaranteed under a stronger assumption that the underlying hash functions behave as *random oracles*, publicly available random functions that take a message of arbitrary length. Such a hash function, called a *pseudorandom oracle*, is rigorously defined in the indistinguishability framework [16]. To the authors' knowledge, the Sponge construction [4] and the MD6 compression function [9] are the only constructions of permutation-based hash functions whose security is proved in the indistinguishability framework.

In this paper, we attempt to find an alternative method of building a pseudorandom oracle from public random permutations. A natural approach is to extend the domain of a permutation-based compression function using the Merkle-Damgård transform, and apply an independent random

function to the output of the MD chaining. This approach is essentially same as the NMAC construction [7]. However, we found that collision resistance of the compression function is not enough to guarantee the indifferentiability of the “NMAC-type” construction from a random oracle. This observation motivated us to introduce a new notion of security, called *adaptive preimage resistance*. Adaptive preimage resistance can be regarded as a strengthening of preimage resistance. In a conventional definition of preimage resistance, an adversary chooses a point in the range of a hash function at the beginning of the experiment, and tries to find a preimage of the point by making adaptive queries to the underlying ideal primitives. In our strengthening, an adversary is allowed to adaptively choose a target point and include it into a *commitment list* during making queries. The only constraint on the choice is that the previous queries should not determine any preimage of the target point. We say that a hash function is adaptive preimage resistant if no adversary is able to find a preimage for any point in the commitment list except with negligible probability.

We prove that a hash/compression function that is collision resistant and adaptive preimage resistant can be composed with a public random function to yield a hash function that is indifferentiable from a random oracle (Theorem 2). We also show that the plain Merkle-Damgård transform preserves adaptive preimage resistance and collision resistance as a combination (Theorem 1). As a related work, Ristenpart and Shrimpton proposed so called Mix-Compress-Mix construction that transforms any collision resistant functions into a pseudorandom oracle [18]. However, their work is based on a complexity-theoretic definition of collision resistance. We also note that Dodis et. al. independently introduced a security notion called *preimage awareness* [10]. The authors proved a similar result that preimage awareness is preserved by the strengthened Merkle-Damgård transform, and any preimage aware function can be composed with a public random function to yield a pseudorandom oracle. However, we emphasize that adaptive preimage resistance is a separate notion from collision resistance, while preimage awareness is a strengthening of collision resistance. We believe that our notion is more suitable for the modular approach of security analysis. A detailed comparison between adaptive preimage resistance and preimage awareness is made in Section 3.2.

Results We summarize our results as follows.

- We introduce a new notion of security, called adaptive preimage resistance. This notion can be regarded as a natural strengthening of preimage resistance. We prove that a compression function that is collision resistant and adaptive preimage resistant can be composed with a public random function to yield a hash function that is indifferentiable from a random oracle. We also show that the plain Merkle-Damgård transform preserves adaptive preimage resistance and collision resistance as a combination.
- We prove adaptive preimage resistance for a certain class of permutation-based compression functions. They are $2n$ -bit to n -bit compression functions that use three calls to public random permutations, denoted “LP231” in [19]. A unified approach allows us to prove their collision resistance and preimage resistance within the proof of adaptive preimage resistance. Our analysis is not only simpler than the one given in [19], but also provides for better asymptotic bounds (at least for the bounds that are explicitly approximated). In [19], the bounds on the number of queries for collision resistance and preimage resistance are given by $O(2^{n(1/2-\epsilon)})$ and $O(2^{n(2/3-\epsilon)})$, respectively, for any $\epsilon > 0$. (The bounds are misprinted as $O(2^{n/2-\epsilon})$ and $O(2^{2n/3-\epsilon})$ in [19].) Our proof guarantees the two properties up to $O(2^{n/2}/n)$ and $O(f(n)2^{2n/3})$ queries, respectively, for any decreasing function $f(n)$. The adaptive preimage resistance of LP231 is guaranteed up to $O(2^{n/2}/n)$ queries. LP231 is the first permutation-based compression function whose adaptive preimage resistance is proved. (However, we also note that Steinberger independently proved the preimage awareness of LP231 [23].)
- With a “filtering” function built upon public random permutations, we obtain an alternative construction of permutation-based pseudorandom oracles, comparable to the Sponge construction and the MD6 compression function. As an example, one might use $F = \sigma \oplus \sigma^{-1}$ for a

public random permutation σ . This function is known to be indifferentiable from a noncompressive public random function up to the birthday bound [8]. Combined with LP231, the resulting hash function is purely permutation-based with rate 1/3. The indifferentiability from a random oracle is guaranteed up to the birthday bound. On contrast, the Sponge construction and the MD6 compression function with rate 1/3 achieve the indifferentiability only up to $O(2^{n/3})$ queries.

2 Preliminaries

General Notations For a positive integer n , we let $I_n = \{0, 1\}^n$ denote the set of all bitstrings of length n , and let $I_n^* = \bigcup_{i=1}^{\infty} I_n^i$. We simply write $I^* = I_1^*$. We let \mathbb{F}_{2^n} denote a finite field of order 2^n . Throughout our work, we will identify \mathbb{F}_{2^n} and I_n , assuming a fixed mapping between the two sets. We write Π_n for the set of permutations on I_n .

For a set U , we write $u \xleftarrow{\$} U$ to denote uniform random sampling from the set U and assignment to u . The multiplicity of u in a multiset U is denoted as $\text{mult}_U(u)$. For an event \mathcal{E} , the complement of \mathcal{E} is denoted as $\bar{\mathcal{E}}$.

Linearly-dependent Permutation-based Compression Functions For positive integers m , k and r with $m > r$, let $\mathcal{M}_{\mathbb{F}_{2^n}}(m, k, r)$ be a set of $(k+r) \times (m+k)$ matrices $A = (a_{ij})$ over \mathbb{F}_{2^n} such that

$$a_{ij} = 0 \text{ for } 1 \leq i \leq k \text{ and } j \geq m+i.$$

Then each matrix $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(m, k, r)$ defines a compression function $H[A]$ with oracle access to independent random permutations $\pi_1, \dots, \pi_k \in \Pi_n$ as follows.

$$\begin{aligned} H[A] : I_n^m &\longrightarrow I_n^r \\ (v_1, \dots, v_m) &\longmapsto (w_1, \dots, w_r), \end{aligned} \tag{1}$$

where (w_1, \dots, w_r) is computed by the algorithm described in Figure 1(a). In [19], such a compression function is called *linearly-dependent permutation-based* (or simply LP), and denoted as LP_{mkr}^A . A compression function $H[A]$ for $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(2, 3, 1)$ is separately described in Figure 1(b).

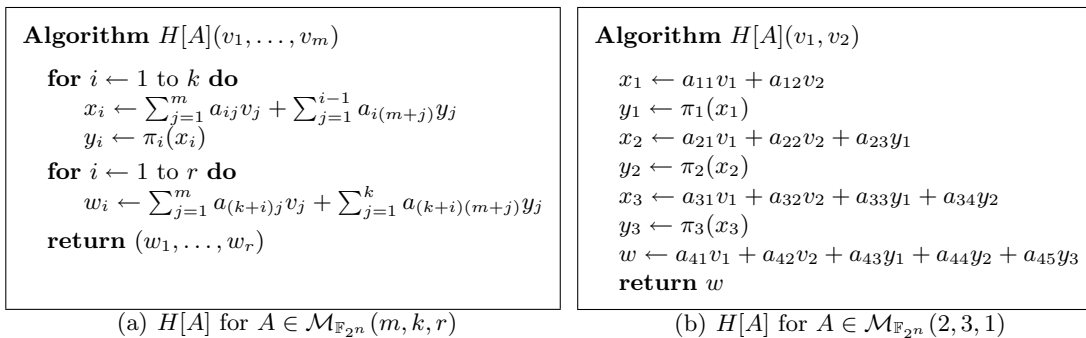


Fig. 1. Compression function $H[A]$

Collision Resistance and Preimage Resistance For simplicity of notations, we will define security notions including collision resistance, preimage resistance and adaptive preimage resistance for linearly-dependent permutation-based compression functions. However, we note that these security notions can be extended in an obvious way to any hash function based on public ideal primitives.

Given a compression function $H = H[A]$ for $A \in \mathcal{M}_{\mathbb{F}_2^n}(m, k, r)$ and an information-theoretic adversary \mathcal{A} with oracle access to π_i and π_i^{-1} , $i = 1, \dots, k$, we execute the experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{coll}}$ described in Figure 2(a) in order to quantify the collision resistance of H . The experiment records the queries that the adversary \mathcal{A} makes into a *query history* \mathcal{Q} . A pair (i, x, y) is in the query history if \mathcal{A} asks $\pi_i(x)$ and gets back y , or it asks $\pi_i^{-1}(y)$ and gets back x . For $k = 1$, we simply write (x, y) for $(1, x, y)$. Given a query history \mathcal{Q} , then $\text{Map}_H(\mathcal{Q}) \subset I_n^m \times I_n^r$ is defined to be the set of pairs (v, w) such that there exist query-response pairs $(i, x_i, y_i) \in \mathcal{Q}$, $i = 1, \dots, k$, satisfying the following equations.

$$\begin{aligned} x_i &= \sum_{j=1}^m a_{ij} v_j + \sum_{j=1}^{i-1} a_{i(m+j)} y_j, & i = 1, \dots, k, \\ w_i &= \sum_{j=1}^m a_{(k+i)j} v_j + \sum_{j=1}^k a_{(k+i)(m+j)} y_j, & i = 1, \dots, r, \end{aligned} \quad (2)$$

for $v = (v_1, \dots, v_m)$ and $w = (w_1, \dots, w_r)$. Informally, $\text{Map}_H(\mathcal{Q})$ is the set of the evaluations of H that are determined by the query history \mathcal{Q} . Now the *collision-finding advantage* of \mathcal{A} is defined to be

$$\mathbf{Adv}_H^{\text{coll}}(\mathcal{A}) = \Pr [\mathbf{Exp}_{\mathcal{A}}^{\text{coll}} = 1]. \quad (3)$$

The probability is taken over the random permutations π_1, \dots, π_k , and \mathcal{A} 's coins (if any). For $q > 0$, we define $\mathbf{Adv}_H^{\text{coll}}(q)$ as the maximum of $\mathbf{Adv}_H^{\text{coll}}(\mathcal{A})$ over all adversaries \mathcal{A} making at most q queries.

The preimage resistance of H is quantified similarly using the experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{pre}}$ described in Figure 2(b). The adversary \mathcal{A} chooses a single ‘‘commitment’’ point $w \in I_n^r$ before it begins making queries to $\pi_i^{\pm 1}$, $i = 1, \dots, k$. (In a weaker version, the point w is chosen uniformly at random.) The *preimage-finding advantage* of \mathcal{A} is defined to be

$$\mathbf{Adv}_H^{\text{pre}}(\mathcal{A}) = \Pr [\mathbf{Exp}_{\mathcal{A}}^{\text{pre}} = 1]. \quad (4)$$

For $q > 0$, $\mathbf{Adv}_H^{\text{pre}}(q)$ is the maximum of $\mathbf{Adv}_H^{\text{pre}}(\mathcal{A})$ over all adversaries \mathcal{A} making at most q queries.

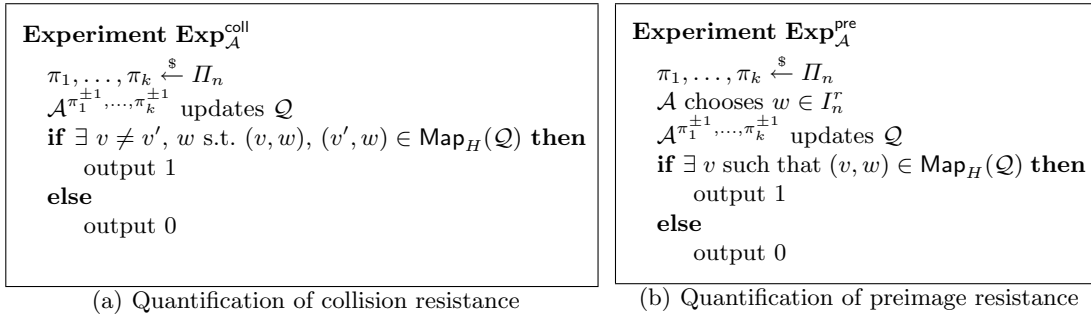


Fig. 2. Experiments for quantification of collision resistance and preimage resistance

Indifferentiability The indifferentiability framework was introduced by Maurer et al. in [16] as an extension of the classical notion of indistinguishability. This general notion allows to discuss secure construction of a public ideal primitive that uses another public ideal primitives as building blocks. In this paper, we are interested in construction of a random oracle based on a certain number of independent random permutations. In the indifferentiability framework, a distinguisher is given

two systems $(\mathcal{C}^{\mathcal{F}}, \mathcal{F})$ and $(\mathcal{P}, \mathcal{S}^{\mathcal{P}})$. Here \mathcal{F} is an ideal primitive used as a building block for the construction of $\mathcal{C}^{\mathcal{F}}$. An ideal primitive \mathcal{P} and a probabilistic Turing machine $\mathcal{S}^{\mathcal{P}}$ with oracle access to \mathcal{P} have the same interfaces as $\mathcal{C}^{\mathcal{F}}$ and \mathcal{F} , respectively. $\mathcal{S}^{\mathcal{P}}$, called a *simulator*, tries to emulate the ideal primitive \mathcal{F} so that no distinguisher can tell apart the two systems $(\mathcal{P}, \mathcal{S}^{\mathcal{P}})$ and $(\mathcal{C}^{\mathcal{F}}, \mathcal{F})$ with non-negligible probability, based on their responses to queries that the distinguisher may send. We say that the construction $\mathcal{C}^{\mathcal{F}}$ is indistinguishable from \mathcal{P} if the existence of such a simulator is proved. The indistinguishability implies the absence of a generic attack against $\mathcal{C}^{\mathcal{F}}$ that regards \mathcal{F} merely as a black-box. Here we give an information-theoretic definition of indistinguishability. For more comprehensive introduction of the indistinguishability framework, we refer to [7, 16].

Definition 1. A Turing machine \mathcal{C} with oracle access to an ideal primitive \mathcal{F} is said to be (q, ϵ) -indistinguishable from an ideal primitive \mathcal{P} if there exists a simulator \mathcal{S} with oracle access to \mathcal{P} such that for any distinguisher \mathcal{D} making at most q queries, it holds that

$$\left| \Pr \left[\mathcal{D}^{\mathcal{C}^{\mathcal{F}}, \mathcal{F}} = 1 \right] - \Pr \left[\mathcal{D}^{\mathcal{P}, \mathcal{S}^{\mathcal{P}}} = 1 \right] \right| < \epsilon.$$

3 Adaptive Preimage Resistance

3.1 Definition

In this section, we define a new notion of security, called *adaptive preimage resistance*. Given a compression function $H = H[A]$ for $A \in \mathcal{M}_{\mathbb{F}_2^n}(m, k, r)$ and an information-theoretic adversary \mathcal{A} with oracle access to $\pi_i^{\pm 1}$, $i = 1, \dots, k$, the adaptive preimage resistance of H is quantified by the experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{pre}}$ described in Figure 3. At any point during the experiment, the adversary \mathcal{A} can choose a commitment point $w \in I_n^r \setminus \text{Range}_H(\mathcal{Q})$, where

$$\text{Range}_H(\mathcal{Q}) = \{w \in I_n^r : (v, w) \in \text{Map}_H(\mathcal{Q}) \text{ for some } v \in I_n^m\}.$$

Then the experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{pre}}$ records the element w into a *commitment list* $\mathcal{L} \subset I_n^r$. At the end of the experiment, \mathcal{A} would like to succeed in finding a preimage of some element in the commitment list. Now the *adaptive preimage-finding advantage* of \mathcal{A} is defined to be

$$\mathbf{Adv}_H^{\text{pre}}(\mathcal{A}) = \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{pre}} = 1]. \quad (5)$$

For $q, q_c > 0$, we define $\mathbf{Adv}_H^{\text{pre}}(q, q_c)$ as the maximum of $\mathbf{Adv}_H^{\text{pre}}(\mathcal{A})$ over all adversaries \mathcal{A} that make at most q queries and at most q_c commitments.

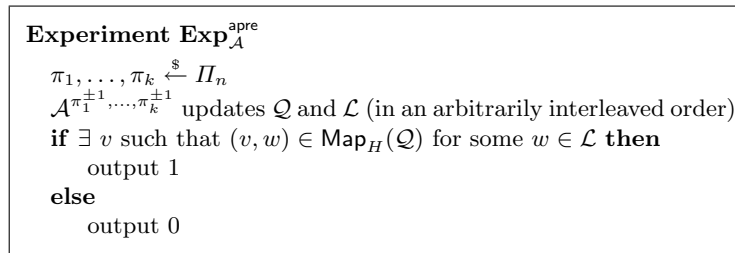


Fig. 3. Experiment for quantification of adaptive preimage resistance

3.2 Comparison of Adaptive Preimage Resistance and Preimage Awareness

From the definition, it is easy to prove that

$$\mathbf{Adv}_H^{\text{pre}}(q) \leq \mathbf{Adv}_H^{\text{pre}}(q, 1), \quad (6)$$

for any compression function H . Therefore, adaptive preimage resistance can be regarded as a natural strengthening of preimage resistance. As seen in the security proof of LP231, a bound for adaptive preimage resistance allows us to instantly upper bound preimage resistance (by setting $q_c = 1$). In our framework, it is sufficient to prove adaptive preimage resistance for existing compression functions whose collision resistance are proved. We note that adaptive preimage resistance is much easier to prove than collision resistance in general. This property is in sharp contrast to preimage awareness that is a strengthening of collision resistance. In the most security proofs of preimage awareness, they begin by bounding away collision resistance and continue in a setting where no collisions are allowed. Thus, we believe that adaptive preimage resistance better captures the modular approach for security analysis compared to preimage awareness.

Interestingly, adaptive preimage resistance has a symbiotic relationship with collision resistance. Neither is preserved by the plain Merkle-Damgård transform on its own, yet the combination is (Theorem 1). On contrast, preimage awareness is preserved by the strengthened Merkle-Damgård transform. The plain Merkle-Damgård transform preserves preimage awareness with an additional condition of difficulty of inverting IV .

3.3 Construction of Hash Functions

We show that if a compression function $H : I_s \times I_t \rightarrow I_s$ is collision resistant and adaptive preimage resistant, then these properties are preserved by the Merkle-Damgård transform:

```

Function  $MD^H(p_1, \dots, p_l)$ 
   $x_0 \leftarrow IV$ 
  for  $i \leftarrow 1$  to  $l$  do
     $x_i \leftarrow H(p_i, x_{i-1})$ 
  return  $x_l$ 

```

Here IV is a predetermined constant and $|x_i| = s$ and $|p_i| = t$ for all i .

Theorem 1. *Let $H : I_s \times I_t \rightarrow I_s$ be a compression function, and let $MD^H : I_t^* \rightarrow I_s$ be the Merkle-Damgård transform based on the function H . Then it holds that*

$$\mathbf{Adv}_{MD^H}^{\text{coll}}(q) \leq \mathbf{Adv}_H^{\text{coll}}(q) + \mathbf{Adv}_H^{\text{apre}}(q, 1), \quad (7)$$

$$\mathbf{Adv}_{MD^H}^{\text{apre}}(q, q_c) \leq \mathbf{Adv}_H^{\text{coll}}(q) + \mathbf{Adv}_H^{\text{apre}}(q, q_c). \quad (8)$$

Proof. The proof of (7) is simple. Let \mathcal{A} be a collision-finding adversary for MD^H . If \mathcal{A} inverts IV with nonnegligible probability, then it can be transformed into an adaptive preimage-finding adversary for H that makes a single commitment of IV . Otherwise, \mathcal{A} can be transformed into a collision-finding adversary for H .

In order to prove (8), suppose that \mathcal{A} is an adaptive preimage-finding adversary for MD^H such that

$$\epsilon = \mathbf{Adv}_{MD^H}^{\text{apre}}(q, q_c) = \mathbf{Adv}_{MD^H}^{\text{apre}}(\mathcal{A}).$$

Then \mathcal{A} can be used for construction of an algorithm \mathcal{A}^* that with probability ϵ succeeds in either finding a collision in H or finding a preimage of some committed point under H . The algorithm \mathcal{A}^* runs \mathcal{A} as a subroutine.

- When \mathcal{A} makes a query to one of the underlying primitives, \mathcal{A}^* makes the same query to the primitive, and relays the response to \mathcal{A} . \mathcal{A}^* records a query history \mathcal{Q} , and updates $\text{Map}_H(\mathcal{Q})$. \mathcal{A}^* also grows a directed graph \mathcal{T} on I_s , where $((c_1, p), c_2) \in \text{Map}_H(\mathcal{Q})$ if and only if $\overrightarrow{c_1 c_2} \in \mathcal{T}$ with label p .
- When \mathcal{A} chooses a commitment point c , \mathcal{A}^* finds an element $\theta(c) = c_0 (\neq IV)$ such that there exists a path $\overrightarrow{c_0 c_1}, \overrightarrow{c_1 c_2}, \dots, \overrightarrow{c_{l-1} c_l} \in \mathcal{T}$ and $c_l = c$ for $l \geq 0$, but there exists no incoming edge to c_0 . Informally, $\theta(c)$ is the start vertex of a path to the vertex c in \mathcal{T} . If there exists two such vertices, then it means that \mathcal{A}^* succeeded in finding a collision in the compression function H . Otherwise, \mathcal{A}^* chooses the unique element c_0 as a commitment point.

Now it is easy to show that if \mathcal{A} finds a preimage of a committed element c under MD^H without finding a collision in H , then \mathcal{A}^* succeeds in finding a preimage of the corresponding commitment $\theta(c)$ under H . This completes the proof of (8). \square

The following theorem states that the domain of a public random function can be extended with any compression function that is collision resistant and adaptive preimage resistant.

Theorem 2. *For $V \subset I^*$, let $H : V \rightarrow I_s$ be a hash/compression function that uses at most L calls to the underlying ideal primitives. If $F : I_s \rightarrow I_t$ is a public random function, then the composite function $F \circ H$ is $(q/L, \epsilon)$ -indifferentiable from a public random function $G : V \rightarrow I_t$, where*

$$\epsilon = \mathbf{Adv}_H^{\text{coll}}(q) + \mathbf{Adv}_H^{\text{apre}}(q, q_c).$$

Proof (Sketch). For simplicity of notations, we assume that $H = H[A]$ is a linearly-determined permutation-based compression function defined by $A \in \mathcal{M}_{\mathbb{F}_2^n}(m, k, r)$, with oracle access to public random permutations π_1, \dots, π_k .

In Figure 4, we present simulator \mathcal{S}^G that makes oracle queries to a public random function G . In the description of \mathcal{S}^G , we define

$$\text{Inv}_{H,w}(\mathcal{Q}) = \{v \in I_n^m : (v, w) \in \text{Map}_H(\mathcal{Q})\},$$

for $w \in I_n^r$. The simulator \mathcal{S}^G faithfully responds to the queries $\pi_i^{\pm 1}(\cdot)$, $i = 1 \dots, k$. On the query $F(w)$, \mathcal{S}^G checks if it has determined a preimage of w under H . If so, it chooses a preimage v uniformly at random from the set $\text{Inv}_{H,w}(\mathcal{Q})$ and replies with $G(v)$. Otherwise, \mathcal{S}^G outputs a random value. Note that if H is collision resistant, then there would be a unique preimage of w , if any, except with negligible probability. In a complexity-theoretic framework, the simulator \mathcal{S}^G can be slightly modified so that it runs in time $O(q^{k-1})$. The modified simulator $\hat{\mathcal{S}}^G$ updates $\text{Map}_H(\mathcal{Q})$ whenever it updates the query history \mathcal{Q} . The update of $\text{Map}_H(\mathcal{Q})$ requires $O(q^{k-1})$ steps. On a query $F(w)$, $\hat{\mathcal{S}}^G$ chooses an arbitrary element in $\text{Inv}_{H,w}(\mathcal{Q})$ by searching the list $\text{Map}_H(\mathcal{Q})$. It does not need to find every element of $\text{Inv}_{H,w}(\mathcal{Q})$.

Simulator \mathcal{S}^G

<p><u>Initialize</u></p> <p>$\pi_1, \dots, \pi_k \xleftarrow{\\$} \Pi_n$ $\mathcal{Q} \leftarrow \emptyset$</p> <p><u>Interface $\pi_i(x)$ ($i = 1, \dots, k$)</u></p> <p>$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(i, x, \pi_i(x))\}$ return $\pi_i(x)$</p> <p><u>Interface $\pi_i^{-1}(y)$ ($i = 1, \dots, k$)</u></p> <p>$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(i, \pi_i^{-1}(y), y)\}$ return $\pi_i^{-1}(y)$</p>	<p><u>Interface $F(w)$</u></p> <p>if $F(w) = \perp$ then if $\text{Inv}_{H,w}(\mathcal{Q}) = \emptyset$ then $F(w) \xleftarrow{\\$} I_n$ else $v \xleftarrow{\\$} \text{Inv}_{H,w}(\mathcal{Q})$ $F(w) \leftarrow G(v)$ return $F(w)$</p>
---	--

Fig. 4. Simulator \mathcal{S}^G

A distinguisher interacts with one of two systems: one consists of a set $\mathcal{F} = \{\pi_1, \dots, \pi_k, F\}$ of ideal primitives and the compression function $\mathcal{C}^{\mathcal{F}} = F \circ H[A]$ based on the primitives, while the other consists of the simulator \mathcal{S}^G that has the same interfaces as the ideal primitives, and the public random function G . Our main observation is that one can distinguish the two systems, only by using one of the following two strategies.

- Find distinct v and v' such that $H[A](v) = H[A](v')$, and check if $F \circ H[A](v) = F \circ H[A](v')$.

- Choose w whose preimage is not determined by \mathcal{Q} and make a query $F(w)$. Then find a preimage v of w under $H[A]$ and make a query $F \circ H[A](v)$. Check if $F(w) = F \circ H[A](v)$.

It can be shown that these strategies are successful in distinguishing the two systems at most with probability ϵ . We give a detailed proof in Appendix A. \square

4 Concrete Security Bounds of $H[A]$ for $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(2, 3, 1)$

In this section, we prove that a linearly-determined permutation-based compression function $H[A]$ achieves good adaptive preimage resistance if its matrix $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(2, 3, 1)$ satisfies a certain condition. For such a compression function, we also analyze preimage resistance and collision resistance in a simpler way than the one given in [19]. First, we describe the condition.

4.1 Condition on the matrix A

For $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(2, 3, 1)$, the system of equations (2) is reduced to the following system of equations.

$$\begin{aligned} x_1 &= a_{11}v_1 + a_{12}v_2 \\ x_2 &= a_{21}v_1 + a_{22}v_2 + a_{23}y_1 \\ x_3 &= a_{31}v_1 + a_{32}v_2 + a_{33}y_1 + a_{34}y_2 \\ w &= a_{41}v_1 + a_{42}v_2 + a_{43}y_1 + a_{44}y_2 + a_{45}y_3. \end{aligned} \tag{9}$$

If we regard every variable in system (9) as constant except four variables v_1 , v_2 , x_3 and y_3 , then the system (9) is rewritten as the following system of equations in the four variables.

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & 1 & 0 \\ a_{41} & a_{42} & 0 & a_{45} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 + a_{23}y_1 \\ a_{33}y_1 + a_{34}y_2 \\ a_{43}y_1 + a_{44}y_2 + w \end{bmatrix}. \tag{10}$$

If $a_{11}a_{22} + a_{12}a_{21} \neq 0$ and $a_{45} \neq 0$, then we can solve system (10) to obtain an equation of the following form.

$$A_1x_1 + A_2y_1 + A_3x_2 + A_4y_2 + A_5x_3 + A_6y_3 = Cw, \tag{11}$$

where A_i and C are 2×1 matrices over \mathbb{F}_{2^n} . Let $\mathcal{M}_{\mathbb{F}_{2^n}}^*(2, 3, 1)$ be the set of matrices $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(2, 3, 1)$ such that

1. $a_{11}a_{22} + a_{12}a_{21} \neq 0$ and $a_{45} \neq 0$,
2. $[A_{i_1}, A_{i_2}]$ are invertible for $1 \leq i_1 < i_2 \leq 6$.

If $A \in \mathcal{M}_{\mathbb{F}_{2^n}}^*(2, 3, 1)$, then we can rewrite equation (11) in the following ways.

$$\begin{aligned} x_1 &= \Phi_1(x_2, y_2, x_3, y_3, w) = \alpha_{11}x_2 + \alpha_{12}y_2 + \alpha_{13}x_3 + \alpha_{14}y_3 + \alpha_{15}w, \\ y_1 &= \Psi_1(x_2, y_2, x_3, y_3, w) = \beta_{11}x_2 + \beta_{12}y_2 + \beta_{13}x_3 + \beta_{14}y_3 + \beta_{15}w, \\ x_2 &= \Phi_2(x_1, y_1, x_3, y_3, w) = \alpha_{21}x_1 + \alpha_{22}y_1 + \alpha_{23}x_3 + \alpha_{24}y_3 + \alpha_{25}w, \\ y_2 &= \Psi_2(x_1, y_1, x_3, y_3, w) = \beta_{21}x_1 + \beta_{22}y_1 + \beta_{23}x_3 + \beta_{24}y_3 + \beta_{25}w, \\ x_3 &= \Phi_3(x_1, y_1, x_2, y_2, w) = \alpha_{31}x_1 + \alpha_{32}y_1 + \alpha_{33}x_2 + \alpha_{34}y_2 + \alpha_{35}w, \\ y_3 &= \Psi_3(x_1, y_1, x_2, y_2, w) = \beta_{31}x_1 + \beta_{32}y_1 + \beta_{33}x_2 + \beta_{34}y_2 + \beta_{35}w, \end{aligned}$$

where $\alpha_{ij} \in \mathbb{F}_{2^n}$ and $\beta_{ij} \in \mathbb{F}_{2^n}$ are nonzero coefficients determined by A . Note that matrices

$$\begin{bmatrix} \alpha_{ij_1} & \alpha_{ij_2} \\ \beta_{ij_1} & \beta_{ij_2} \end{bmatrix} \text{ are invertible for } i \in \{1, 2, 3\}, j_1 \in \{1, 2\} \text{ and } j_2 \in \{3, 4\}.$$

Example 1. Let $n = 128$ and let $\mathbb{F}_{2^{128}} = \mathbb{F}[\zeta]/(\zeta^{128} + \zeta^7 + \zeta^2 + \zeta + 1)$ be a finite field, where $f(\zeta) = \zeta^{128} + \zeta^7 + \zeta^2 + \zeta + 1$ is an irreducible polynomial over \mathbb{F}_2 . For simplicity of computation, assume that $a_{23} = 0$, $a_{45} = 1$, and

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Then we have

$$[A_1, A_2, A_3, A_4, A_5, A_6] = \begin{bmatrix} a_{31} & a_{33} & a_{32} & a_{34} & 1 & 0 \\ a_{41} & a_{43} & a_{42} & a_{44} & 0 & 1 \end{bmatrix}.$$

If we set $a_{31} = a_{33} = a_{34} = a_{41} = a_{42} = 1$, $a_{32} = a_{43} = \zeta$ and $a_{44} = \zeta^2$, then

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & \zeta & 1 & 1 & 0 \\ 1 & 1 & \zeta & \zeta^2 & 1 \end{bmatrix}$$

is contained in $\mathcal{M}_{\mathbb{F}_2^n}^*(2, 3, 1)$.

Remark 1. Our condition that consists of 16 inequalities seems to be equivalent to the ‘‘independence criterion’’ described in [19]. (Note that $[A_5, A_6]$ is always the identity matrix.) Either condition would be achieved by most choices of the matrix A .

4.2 Security Bounds of $H[A]$

In this section, we analyze adaptive preimage resistance, collision resistance and preimage resistance of a compression function $H[A]$ for $A \in \mathcal{M}_{\mathbb{F}_2^n}^*(2, 3, 1)$. We begin with the following three lemmas.

Lemma 1. *Suppose that an adversary \mathcal{A} makes q ($\leq 2^{n-1}$) adaptive queries to a random permutation π and its inverse π^{-1} , and updates a query history \mathcal{Q} . Let*

$$U = U(\alpha_1, \alpha_2) = \{\alpha_1 x + \alpha_2 y : (x, y) \in \mathcal{Q}\}$$

be a multiset defined for nonzero elements $\alpha_1, \alpha_2 \in \mathbb{F}_2^n$. Then, for $l > 0$,

$$\text{prob}_1(l) = \Pr[\mathcal{A} \text{ sets } \text{mult}_U(c) \geq l \text{ for some } c \in \mathbb{F}_2^n] \leq 2^n \binom{q}{l} \left(\frac{1}{2^{n-1}} \right)^l.$$

Proof. Fix $c^* \in \mathbb{F}_2^n$. When \mathcal{A} makes the j -th query $\pi(x)$, the probability that $\alpha_1 x + \alpha_2 \pi(x) = c^*$, or $\pi(x) = \alpha_2^{-1}(c^* + \alpha_1 x)$, is not greater than $1/(2^n - (j - 1))$. Similarly, when \mathcal{A} makes the j -th query $\pi^{-1}(y)$, the probability that $\alpha_1 \pi^{-1}(y) + \alpha_2 y = c^*$, or $\pi^{-1}(y) = \alpha_1^{-1}(c^* + \alpha_2 y)$, is not greater than $1/(2^n - (j - 1))$.

Note that the event ‘‘ $\text{mult}_U(c^*) \geq l$ ’’ occurs when there are at least l queries among the total q queries such that $\alpha_1 x + \alpha_2 \pi(x) = c^*$ or $\alpha_1 \pi^{-1}(y) + \alpha_2 y = c^*$. Since

$$\frac{1}{2^n - (j - 1)} \leq \frac{1}{2^n - q} \leq \frac{1}{2^{n-1}},$$

and

$$\Pr[\mathcal{A} \text{ sets } \text{mult}_U(c) \geq l \text{ for some } c \in \mathbb{F}_2^n] \leq \sum_{c^* \in \mathbb{F}_2^n} \Pr[\text{mult}_U(c^*) \geq l],$$

it follows that

$$\text{prob}_1(l) \leq 2^n \binom{q}{l} \left(\frac{1}{2^{n-1}} \right)^l.$$

□

Lemma 2. *Suppose that an adversary \mathcal{A} makes q ($\leq 2^{n-1}$) adaptive queries to random permutations π_i and their inverses π_i^{-1} , $i = 1, 2$, and updates a query history \mathcal{Q} . Let*

$$\begin{aligned} U &= U(\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}) \\ &= \{\alpha_{11}x_1 + \alpha_{12}y_1 + \alpha_{21}x_2 + \alpha_{22}y_2 : (1, x_1, y_1), (2, x_2, y_2) \in \mathcal{Q}\} \end{aligned}$$

be a multiset defined for nonzero elements $\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22} \in \mathbb{F}_{2^n}$. Then, for $L > 0$ and $l > 1$,

$$\begin{aligned} \text{prob}_2(L) &= \Pr[\mathcal{A} \text{ sets } \text{mult}_U(c) \geq L \text{ for some } c \in \mathbb{F}_{2^n}] \\ &\leq 2^{n+1} \binom{q}{l} \left(\frac{1}{2^{n-1}}\right)^l + 2^n \binom{q}{\lceil L/(l-1) \rceil} \left(\frac{q}{2^{n-1}}\right)^{\lceil L/(l-1) \rceil}. \end{aligned}$$

Proof. First, we define multisets

$$U_1 = \{\alpha_{11}x + \alpha_{12}y : (x, y) \in \mathcal{Q}\} \text{ and } U_2 = \{\alpha_{21}x + \alpha_{22}y : (x, y) \in \mathcal{Q}\}.$$

These multisets are associated with events

$$\begin{aligned} \mathcal{E}_1 &\Leftrightarrow \mathcal{A} \text{ sets } \text{mult}_{U_1}(c) \geq l \text{ for some } c \in \mathbb{F}_{2^n}, \\ \mathcal{E}_2 &\Leftrightarrow \mathcal{A} \text{ sets } \text{mult}_{U_2}(c) \geq l \text{ for some } c \in \mathbb{F}_{2^n}. \end{aligned}$$

Let $\mathcal{E}_3 = \mathcal{E}_1 \cup \mathcal{E}_2$, and let

$$\mathcal{E} \Leftrightarrow \mathcal{A} \text{ sets } \text{mult}_U(c) \geq L \text{ for some } c \in \mathbb{F}_{2^n}.$$

Then it follows that

$$\text{prob}_2(L) = \Pr[\mathcal{E}] \leq \Pr[\mathcal{E} \cap \overline{\mathcal{E}_3}] + \Pr[\mathcal{E}_3]. \quad (12)$$

By Lemma 1, we have

$$\Pr[\mathcal{E}_3] \leq 2^{n+1} \binom{q}{l} \left(\frac{1}{2^{n-1}}\right)^l. \quad (13)$$

Now we estimate the probability $\Pr[\mathcal{E} \cap \overline{\mathcal{E}_3}]$. For a fixed $c^* \in \mathbb{F}_{2^n}$, each query increases $\text{mult}_U(c^*)$ at most by $l-1$ without the occurrence of \mathcal{E}_3 . Therefore, there should be at least $d = \lceil L/(l-1) \rceil$ queries that increase $\text{mult}_U(c^*)$ at least by one. Since, for each query, the number of responses that increase $\text{mult}_U(c^*)$ is at most q , we obtain

$$\Pr[\mathcal{E} \cap \overline{\mathcal{E}_3}] \leq 2^n \binom{q}{\lceil L/(l-1) \rceil} \left(\frac{q}{2^{n-1}}\right)^{\lceil L/(l-1) \rceil}. \quad (14)$$

This completes the proof together with (12) and (13). \square

The following lemma is used only for the proof of collision resistance.

Lemma 3. *Suppose that an adversary \mathcal{A} makes q ($\leq 2^{n-2}$) adaptive queries to random permutations π_i and their inverses π_i^{-1} , $i = 1, 2$, and updates a query history \mathcal{Q} . Let*

$$U = U(A) = \{[A_1, A_2] \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + [A_3, A_4] \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} : (1, x_1, y_1), (2, x_2, y_2) \in \mathcal{Q}\}$$

be a multiset, where $A = [A_1, A_2, A_3, A_4]$ is a 2×4 matrix over \mathbb{F}_{2^n} such that submatrices $[A_1, A_3]$, $[A_1, A_4]$, $[A_2, A_3]$ and $[A_2, A_4]$ are invertible. Then, for $L > 1$, it holds that

$$\text{prob}_3 = \Pr[\mathcal{A} \text{ sets } \text{mult}_U(c) \geq 2 \text{ for some } c \in \mathbb{F}_{2^n}] \leq \frac{q^2 L}{2^{n-1}} + 4\text{prob}_2(L).$$

Proof. For a collision

$$[A_1, A_2] \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + [A_3, A_4] \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = [A_1, A_2] \begin{bmatrix} x'_1 \\ y'_1 \end{bmatrix} + [A_3, A_4] \begin{bmatrix} x'_2 \\ y'_2 \end{bmatrix}, \quad (15)$$

it should be the case that $(x_1, y_1) \neq (x'_1, y'_1)$ and $(x_2, y_2) \neq (x'_2, y'_2)$. Suppose that \mathcal{A} makes a query $\pi_1(x^*) (= y)$. With (x_1, y_1) replaced by (x^*, y) , the equality (15) is rewritten as

$$A_1 x'_1 + A_2 y'_1 + A_3 x'_2 + A_4 y'_2 + A_3 x_2 + A_4 y_2 = A_1 x^* + A_2 y. \quad (16)$$

Any response y satisfying (16) corresponds to a triple $((1, x'_1, y'_1), (2, x'_2, y'_2), (2, x_2, y_2)) \in \mathcal{Q}^3$ such that

$$(BA_1 x'_1 + BA_2 y'_1 + BA_3 x'_2 + BA_4 y'_2) + (BA_3 x_2 + BA_4 y_2) = BA_1 x^*. \quad (17)$$

Here B is a 1×2 matrix such that $BA_2 = 0$. For each $(2, x_2, y_2) \in \mathcal{Q}$, the number of pairs $((1, x'_1, y'_1), (2, x'_2, y'_2))$ satisfying (17) is smaller than L except with probability $\text{prob}_2(L)$. It means that the number of triples satisfying (16) is smaller than qL except with probability $\text{prob}_2(L)$.

Taking into account symmetry, we conclude that the probability that the j -th query makes a collision in U is not greater than $qL/(2^n - j) \leq qL/2^{n-1}$, with some exceptions that occur with probability at most $4\text{prob}_2(L)$. It completes the proof. \square

We are now ready to prove the following theorem.

Theorem 3. *Let $H[A]$ be a compression function for $A \in \mathcal{M}_{\mathbb{F}_{2^n}}^*(2, 3, 1)$, with oracle access to random permutations π_i , $i = 1, 2, 3$. Suppose that an adversary \mathcal{A} makes q ($\leq 2^{n-2}$) adaptive queries to π_i and π_i^{-1} , $i = 1, 2, 3$, and q_c commitments, and updates a query history \mathcal{Q} and a commitment list \mathcal{L} . Then, for $L > 0$ and $l > 1$, it holds that*

$$\text{Adv}_H^{\text{apre}}(q, q_c) \leq \frac{qq_c L}{2^{n-1}} + 6\epsilon, \quad (18)$$

as well as,

$$\text{Adv}_H^{\text{pre}}(q) \leq \frac{qL}{2^{n-1}} + 6\epsilon, \quad \text{Adv}_H^{\text{coll}}(q) \leq \frac{q^2(L^2 + 3L)}{2^{n-1}} + 18\epsilon, \quad (19)$$

where

$$\epsilon = 2^{n+1} \binom{q}{l} \left(\frac{1}{2^{n-1}} \right)^l + 2^n \binom{q}{\lceil L/(l-1) \rceil} \left(\frac{q}{2^{n-1}} \right)^{\lceil L/(l-1) \rceil}. \quad (20)$$

Proof. Let Φ_i and Ψ_i , $1 \leq i \leq 3$, be the functions defined in Section 4.1. For $x^*, y^*, w^* \in \mathbb{F}_{2^n}$, define

$$\Gamma_{\Phi_1}(x^*, w^*) = \{((x_2, y_2), (x_3, y_3)) \in \mathcal{Q}^2 : x^* = \Phi_1(x_2, y_2, x_3, y_3, w^*)\},$$

and

$$\Gamma_{\Psi_1}(y^*, w^*) = \{((x_2, y_2), (x_3, y_3)) \in \mathcal{Q}^2 : y^* = \Psi_1(x_2, y_2, x_3, y_3, w^*)\}.$$

$\Gamma_{\Phi_i}(x^*, w^*)$ and $\Gamma_{\Psi_i}(y^*, w^*)$ are similarly defined for $i = 2, 3$. Then, for each $i = 1, 2, 3$, the probability that

$$\exists x^*, w^* \in \mathbb{F}_{2^n}, |\Gamma_{\Phi_i}(x^*, w^*)| \geq L \quad (21)$$

is not greater than ϵ . For example, suppose that $|\Gamma_{\Phi_1}(x^*, w^*)| \geq L$ for some $x^*, w^* \in \mathbb{F}_{2^n}$. It means that $\text{mult}_U(x^* + \alpha_{15}w^*) \geq L$, where

$$U = \{\alpha_{11}x_2 + \alpha_{12}y_2 + \alpha_{13}x_3 + \alpha_{14}y_3 : (x_2, y_2), (x_3, y_3) \in \mathcal{Q}\}.$$

We see that such an event occurs with probability at most ϵ by Lemma 2. Similarly, for each $i = 1, 2, 3$, the probability that

$$\exists y^*, w^* \in \mathbb{F}_{2^n}, |\Gamma_{\Psi_i}(y^*, w^*)| \geq L \quad (22)$$

is not greater than ϵ . Let \mathcal{E}_{1i} and \mathcal{E}_{2i} , respectively, denote the event (21) and (22) for $i = 1, 2, 3$, and let $\mathcal{E} = \mathcal{E}_{11} \cup \mathcal{E}_{12} \cup \mathcal{E}_{13} \cup \mathcal{E}_{21} \cup \mathcal{E}_{22} \cup \mathcal{E}_{23}$. Then it follows that $\Pr[\mathcal{E}] \leq 6\epsilon$.

Adaptive Preimage Resistance. For the proof of (18), it is sufficient to show that

$$\Pr [\mathbf{Exp}_{\mathcal{A}}^{\text{apre}} = 1 \cap \overline{\mathcal{E}}] \leq \frac{qq_c L}{2^{n-1}}, \quad (23)$$

for any adaptive preimage-finding adversary \mathcal{A} . Fix $w^* \in \mathcal{L}$, and suppose that \mathcal{A} makes a query, say, $\pi_1(x^*)$ for some $x^* \in \mathbb{F}_{2^n}$. In order for the pair $(x^*, \pi_1(x^*))$ to determine $(v, w^*) \in \text{Map}_H(\mathcal{Q})$ for some v , it should be the case that $\pi_1(x^*) = \Psi_1(x_2, y_2, x_3, y_3)$ for $((2, x_2, y_2), (3, x_3, y_3)) \in \Gamma_{\Phi_1}(x^*, w^*)$. At the j -th query, such an event occurs with probability at most $L / (2^n - (j-1)) \leq L/2^{n-1}$, since $|\Gamma_{\Phi_1}(x^*, w^*)| \leq L$ without the occurrence of \mathcal{E} . Now the inequality (23) is proved since $|\mathcal{Q}| \leq q$ and $|\mathcal{L}| \leq q_c$.

Preimage Resistance. The proof is straightforward from (6) and (18).

Collision Resistance. First, we estimate the probability P_1 that two distinct queries make a collision. Fix $w^* \in \mathbb{F}_{2^n}$ and let $1 \leq j_1 < j_2 \leq q$. For any collision-finding adversary \mathcal{A} , the probability that the j_1 -th query and the j_2 -th query, respectively, determine evaluations (v, w^*) and (v', w^*) in $\text{Map}_H(\mathcal{Q})$ for some v and v' without the occurrence of \mathcal{E} is not greater than $(L/2^{n-1})^2$. Therefore, we have

$$P_1 \leq 2^n \binom{q}{2} \left(\frac{L}{2^{n-1}} \right)^2 + \Pr[\mathcal{E}] \leq \frac{q^2 L^2}{2^{n-1}} + 6\epsilon, \quad (24)$$

Next, we estimate the probability P_2 that a single query, say $\pi_1(x^*)$, makes a collision. This case implies the occurrence of a collision

$$\begin{bmatrix} \Phi_1(x_2, y_2, x_3, y_3, w^*) \\ \Psi_1(x_2, y_2, x_3, y_3, w^*) \end{bmatrix} = \begin{bmatrix} \Phi_1(x'_2, y'_2, x'_3, y'_3, w^*) \\ \Psi_1(x'_2, y'_2, x'_3, y'_3, w^*) \end{bmatrix} \left(= \begin{bmatrix} x^* \end{bmatrix} \right), \quad (25)$$

for some w^* . Considering queries for π_2 and π_3 , we have

$$P_2 \leq 3\text{prob}_3 \leq \frac{3q^2 L}{2^{n-1}} + 12\epsilon, \quad (26)$$

by Lemma 3. From (24) and (26), we have

$$\Pr [\mathbf{Exp}_{\mathcal{A}}^{\text{coll}} = 1] \leq P_1 + P_2 \leq \frac{q^2(L^2 + 3L)}{2^{n-1}} + 18\epsilon. \quad (27)$$

□

Corollary 1. *Let $H_n = H[A_n]$ be a compression function for $A_n \in \mathcal{M}_{\mathbb{F}_{2^n}}^*(2, 3, 1)$. If $q = f(n)2^{\frac{2n}{3}}$ and $\lim_{n \rightarrow \infty} f(n) = 0$, then $\lim_{n \rightarrow \infty} \mathbf{Adv}_{H_n}^{\text{pre}}(q) = 0$.*

Proof. Set $L = 2 \cdot 2^{\frac{n}{3}}$ and $l = 3$, and use Theorem 3. The detailed proof is given in Appendix 1. □

Corollary 2. *Let $H_n = H[A_n]$ be a compression function for $A_n \in \mathcal{M}_{\mathbb{F}_{2^n}}^*(2, 3, 1)$. If $q = 2^{\frac{n}{2}}/n$, then $\lim_{n \rightarrow \infty} \mathbf{Adv}_{H_n}^{\text{pre}}(q) = \lim_{n \rightarrow \infty} \mathbf{Adv}_{H_n}^{\text{coll}}(q) = 0$.*

Proof. Set $L = n/\log_2 n$ and $l = 2$, and use Theorem 3. The proof is similar to that of Corollary 1. □

5 Permutation-based Hash Functions

Our results allow us to construct a secure hash function in the indistinguishability framework using only a small number of public random permutations as follows.

1. Based on a small number of public random permutations, construct a compression function H that is collision resistant and adaptive preimage resistant.

2. Apply the plain Merkle-Damgård transform to the compression function H .
3. Construct a public random function F based on a small number of public random permutations.
4. Define the composite $F \circ MD^H$ as the resulting hash function.

In the third step, we might use $F = \sigma \oplus \sigma^{-1}$ for a public random permutation σ . This function is known to be indifferentiable from a noncompressive public random function up to the birthday bound [8]. Combined with LP231, the resulting hash function is purely permutation-based with rate 1/3. The indifferentiability from a random oracle is guaranteed up to the birthday bound. On contrast, the Sponge construction and the MD6 compression function with rate 1/3 achieve the indifferentiability only up to $O(2^{n/3})$ queries.

If the filtering function F is given as a public random permutation, then the resulting hash function is not guaranteed to be a pseudorandom oracle. A distinguisher might exploit the interface F^{-1} in order to tell apart $(F \circ H, \mathcal{F})$ and (G, \mathcal{S}^G) . However, it seems that such a construction still gives a *public-use pseudorandom oracle* introduced and studied in [10]. Finally, we notice that the single-permutation-based variant of LP231 was also proved to be adaptive preimage resistant up to the birthday bound [14].

Acknowledgements

The authors are grateful to Martijn Stam and John Steinberger for their valuable comments.

References

1. P. S. L. M. Barreto and V. Rijmen. The Whirlpool hashing function. Primitve submitted to NESSIE, September 2000, revised on May 2003.
2. M. Bellare, P. Rogaway. Optimal asymmetric encryption-how to encrypt with RSA. Eurocrypt 1994, LNCS 950, pp. 92–111, Springer-Verlag, 1994.
3. M. Bellare, P. Rogaway. The exact security of digital signatures-how to sign with RSA and Rabin. Eurocrypt 1996, LNCS 1070, pp. 399–416, Springer-Verlag, 1996.
4. G. Bertoni, J. Daemen, M. Peeters and G. Van Assche. On the indifferentiability of the Sponge construction. Eurocrypt 2008, LNCS 4965, pp. 181–197, Springer-Verlag, 2008.
5. J. Black, M. Cochran and T. Shrimpton. On the impossibility of highly-efficient blockcipher-based hash functions. Eurocrypt 2005, LNCS 3494, pp. 526–541, Springer-Verlag, 2005.
6. J. Black, P. Rogaway and T. Shrimpton. Black-box analysis of the block-cipher-based hash-function construction from PGV. Crypto 2002, LNCS 2442, pp. 320–325, Springer-Verlag, 2002.
7. J. Coron, Y. Dodis, C. Malinaud and P. Puniya. Merkle-Damgård revisited: How to construct a hash function. Crypto 2005, LNCS 3621, pp. 430–448, Springer-Verlag, 2005.
8. Y. Dodis, K. Pietrzak and P. Puniya. A new mode of operation for block ciphers and length-preserving MACs. Eurocrypt 2008, LNCS 4965, pp. 198–219, Springer-Verlag, 2008.
9. Y. Dodis, L. Reyzin, R. L. Rivest and E. Shen. Indifferentiability of permutation-based compression functions and tree-based modes of operation, with application to MD6. Preproceedings of FSE 2009, pp. 106–122, 2009.
10. Y. Dodis, T. Ristenpart and T. Shrimpton. Salvaging Merkle-Damgård for practical applications. Eurocrypt 2009, To appear. Available at <http://www.cs.nyu/~dodis>.
11. S. Hirose. Provably secure double-block-length hash functions in a black-box model. ICISC 2004, LNCS 3506, pp. 330–342, Springer-Verlag, 2005.
12. S. Hirose. Some plausible construction of double-block-length hash functions. FSE 2006, LNCS 4047, pp. 210–225, Springer-Verlag, 2006.
13. H. Krawczyk. HMACV: A high-performance secure Diffie-Hellman protocol. CRYPTO 2005. LNCS 3621, pp. 546–566. Springer-Verlag, 2005. Full version available at <http://eprint.iacr.org/2005/176>.
14. J. Lee and D. Kwon. Security of single-permutation-based compression functions. Available at <http://eprint.iacr.org/2009/145>.
15. S. Matyas, S. Meyer and J. Oseas. Generating strong one-way functions with cryptographic algorithm. IBM Technical Disclosure Bulletin 27, 10a, pp. 5658–5659, 1985.

16. U. Maurer, R. Renner and R. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. TCC 2004, LNCS 2951, pp. 21–39, Springer-Verlag, 2008.
17. B. Preneel, R. Govaerts and J. Vandewalle. Hash functions based on block ciphers: A synthetic approach. Crypto 1993, LNCS 773, pp. 368–378, Springer-Verlag, 1994.
18. T. Ristenpart and T. Shrimpton. How to build a hash function from any collision-resistant function. Asiacrypt 2007, LNCS 4833, pp. 147–163, Springer-Verlag, 2007.
19. P. Rogaway and J. Steinberger. Constructing cryptographic hash functions from fixed-key blockciphers. Crypto 2008, LNCS 5157, pp. 433–450, Springer-Verlag, 2008.
20. P. Rogaway and J. Steinberger. Security/efficiency tradeoffs for permutation-based hashing. Eurocrypt 2008, LNCS 4965, pp. 220–236, Springer-Verlag, 2008.
21. T. Shrimpton and M. Stam. Building a collision-resistant function from non-compressing primitives. ICALP 2008, LNCS 5126, pp. 643–654, Springer-Verlag, 2008.
22. M. Stam. Beyond uniformity: Better security/efficiency tradeoffs for compression functions. Crypto 2008, LNCS 5157, pp. 397–412, Springer-Verlag, 2008.
23. J. Steinberger. Private communications.
24. J. Steinberger. The collision intractability of MDC-2 in the ideal-cipher model. Eurocrypt 2007, LNCS 4515, pp. 34–51, Springer-Verlag, 2008.
25. R. Winternitz. A secure one-way hash function built from DES. IEEE Symposium on Information Security and Privacy, pp. 88–90, IEEE Press, 1984.

A Proof of Theorem 2

The proof is based on the standard game-hopping technique. Let \mathcal{G} and \mathcal{H} be the games defined in Figure 5. Without loss of generality, we can make the following assumptions for the games and a distinguisher \mathcal{D} that interacts with the games.

1. Before \mathcal{D} makes a query $G(v)$, \mathcal{D} computes $H[A](v)$. It means that \mathcal{D} has made queries to π_1, \dots, π_k , that are required to compute $H[A](v)$.
2. Each game records a query history \mathcal{Q} for π_1, \dots, π_k . Any interface of the games do not make a private query to the permutations due to the first assumption. Therefore we can regard the query history as public to the distinguisher \mathcal{D} .

We observe that the subroutine **Sample- G** is called only within the subroutine **Sample- F** in game \mathcal{G} . Therefore, **Sample- G** can be merged into **Sample- F** , yielding game \mathcal{H} . In this way, games \mathcal{G} and \mathcal{H} are equivalent, where \mathcal{H} implements $(F \circ H[A], \mathcal{F})$. Now we prove the following claim.

Claim. Game \mathcal{G} faithfully simulates (G, \mathcal{S}^G) , as long as none of the flags bad_1 and bad_2 is set to true.

proof of claim When $G(v)$ is asked, game \mathcal{G} computes $w = H[v]$. If the previous queries have not made a collision for H , then $\text{Inv}_{H,w}(\mathcal{Q}) = \{v\}$. Assuming $F(w) = \perp$, the subroutine **Sample- F** returns **Sample- $G(v)$** . On the contrary, suppose that $F(w)$ is already defined, say at the j -th query. Then, it should be the case $\text{Inv}_{H,w}(\mathcal{Q}) = \{v\}$ at the point where the subroutine **Sample- $F(w)$** is called for the first time; if $\text{Inv}_{H,w}(\mathcal{Q}) = \emptyset$, then $G(v)$ (or any previous query) determines $w = H(v)$ at some point after $F(w)$ is defined, setting bad_2 to true. If $\text{Inv}_{H,w}(\mathcal{Q})$ contains v' which is not v , then the query $G(v)$ (or any previous query) would result in a collision such that $H(v) = H(v') = w$. Therefore at the j -th query, the subroutine **Sample- F** returns **Sample- $G(v)$** . Now the claim is followed since the other interfaces are faithfully implemented.

Any distinguisher \mathcal{D} that interacts with \mathcal{G} can be regarded as a collision-finding adversary \mathcal{A} for $H[A]$ such that $\Pr(\mathcal{D}^{\mathcal{G}} \text{ sets } \text{bad}_1) \leq \mathbf{Adv}_H^{\text{coll}}(\mathcal{A})$. Also, \mathcal{D} can be transformed into an adaptive preimage-finding adversary \mathcal{B} such that $\Pr(\mathcal{D}^{\mathcal{G}} \text{ sets } \text{bad}_2) \leq \mathbf{Adv}_H^{\text{apre}}(\mathcal{B})$. Such an adversary \mathcal{B} runs \mathcal{D} as a subroutine and, when \mathcal{D} makes a query $F(w)$ such that $\text{Inv}_{H,w}(\mathcal{Q}) = \emptyset$, chooses the element w as a commitment. Therefore we conclude that

$$\begin{aligned} \left| \Pr \left[\mathcal{D}^{F \circ H[A], \mathcal{F}} = 1 \right] - \Pr \left[\mathcal{D}^{G, \mathcal{S}^G} = 1 \right] \right| &\leq \Pr(\mathcal{D}^{\mathcal{G}} \text{ sets } \text{bad}_1) + \Pr(\mathcal{D}^{\mathcal{G}} \text{ sets } \text{bad}_2) \\ &\leq \mathbf{Adv}_H^{\text{coll}}(q) + \mathbf{Adv}_H^{\text{apre}}(q, q_c), \end{aligned} \quad (28)$$

for any distinguisher \mathcal{D} that makes at most q/L queries.

Games \mathcal{G}

<p><u>Initialize</u></p> $\pi_1, \dots, \pi_k \stackrel{\$}{\leftarrow} \Pi_n$ <p><u>Interface $\pi_i(x)$ ($i = 1, \dots, k$)</u></p> <p>if $\pi_i(x)$ makes a collision for H then $\text{bad}_1 \leftarrow \text{true}$ else if $\pi_i(x)$ determines $w = H(v)$ s.t. $F(w) \neq \perp$ then $\text{bad}_2 \leftarrow \text{true}$ return $\pi_i(x)$</p> <p><u>Interface $\pi_i^{-1}(y)$ ($i = 1, \dots, k$)</u></p> <p>if $\pi_i^{-1}(y)$ makes a collision for H then $\text{bad}_1 \leftarrow \text{true}$ else if $\pi_i^{-1}(y)$ determines $w = H(v)$ s.t. $F(w) \neq \perp$ then $\text{bad}_2 \leftarrow \text{true}$ return $\pi_i^{-1}(y)$</p> <p><u>Interface $F(w)$</u></p> <p>return $\text{Sample-}F(w)$</p> <p><u>Interface $G(v)$</u></p> <p>for $i \leftarrow 1$ to k do $x_i \leftarrow \sum_{j=1}^m a_{ij}v_j + \sum_{j=1}^{i-1} a_{i(m+j)}y_j$ $y_i \leftarrow \pi_i(x_i)$ for $i \leftarrow 1$ to r do $w_i \leftarrow \sum_{j=1}^m a_{(k+i)j}v_j + \sum_{j=1}^k a_{(k+i)(m+j)}y_j$ return $\text{Sample-}F(w)$</p>	<p><u>Subroutine $\text{Sample-}F(w)$</u></p> <p>if $F(w) = \perp$ then if $\text{Inv}_{H,w}(\mathcal{Q}) = \emptyset$ then $F(w) \stackrel{\\$}{\leftarrow} I_n$ else $v \stackrel{\\$}{\leftarrow} \text{Inv}_{H,w}(\mathcal{Q})$ $F(w) \leftarrow \text{Sample-}G(v)$ return $F(w)$</p> <p><u>Subroutine $\text{Sample-}G(v)$</u></p> <p>if $G(v) = \perp$ then $G(v) \stackrel{\\$}{\leftarrow} I_n$ return $G(v)$</p>
--	--

Game \mathcal{H}

<p><u>Initialize</u></p> $\pi_1, \dots, \pi_k \stackrel{\$}{\leftarrow} \Pi_n$ <p><u>Interface $\pi_i(x)$ ($i = 1, \dots, k$)</u></p> <p>return $\pi_i(x)$</p> <p><u>Interface $\pi_i^{-1}(y)$ ($i = 1, \dots, k$)</u></p> <p>return $\pi_i^{-1}(y)$</p> <p><u>Interface $F(w)$</u></p> <p>return $\text{Sample-}F(w)$</p>	<p><u>Interface $G(v)$</u></p> <p>for $i \leftarrow 1$ to k do $x_i \leftarrow \sum_{j=1}^m a_{ij}v_j + \sum_{j=1}^{i-1} a_{i(m+j)}y_j$ $y_i \leftarrow \pi_i(x_i)$ for $i \leftarrow 1$ to r do $w_i \leftarrow \sum_{j=1}^m a_{(k+i)j}v_j + \sum_{j=1}^k a_{(k+i)(m+j)}y_j$ return $\text{Sample-}F(w)$</p> <p><u>Subroutine $\text{Sample-}F(w)$</u></p> <p>if $F(w) = \perp$ then $F(w) \stackrel{\\$}{\leftarrow} I_n$ return $F(w)$</p>
---	---

Fig. 5. Games \mathcal{G} and \mathcal{H}

B Proof of Corollary 1

Set $L = 2 \cdot 2^{\frac{2}{3}}$ and $l = 3$, and use the inequality

$$\mathbf{Adv}_H^{\text{pre}}(q) \leq \frac{qL}{2^{n-1}} + 6\epsilon. \quad (29)$$

For the first term on the right-hand side of the bound (29), we have

$$\lim_{n \rightarrow \infty} \frac{qL}{2^{n-1}} = \lim_{n \rightarrow \infty} (4f(n)) = 0. \quad (30)$$

Since

$$\begin{aligned} \log_2 \left(2^{n+1} \binom{q}{l} \left(\frac{1}{2^{n-1}} \right)^l \right) &\leq \log_2 \left(2^{n+1} \left(\frac{qe}{l} \right)^l \left(\frac{1}{2^{n-1}} \right)^l \right) \\ &\leq n + 1 + 3 \left(\frac{2n}{3} + \log_2 f(n) - n + C \right) \\ &= 3 \log_2 f(n) + 3C + 1 \rightarrow -\infty, \end{aligned} \quad (31)$$

as $n \rightarrow 0$, where $C = \log_2(2e/3)$, and

$$\begin{aligned} \log_2 \left(2^n \binom{q}{d} \left(\frac{q}{2^{n-1}} \right)^d \right) &\leq \log_2 \left(2^n \left(\frac{qe}{d} \right)^d \left(\frac{q}{2^{n-1}} \right)^d \right) \\ &\leq n + 2^{\frac{2}{3}} \left(2 \left(\frac{2n}{3} + \log_2 f(n) \right) - \frac{4n}{3} + C' \right) \\ &= n + 2^{\frac{2}{3}} (2 \log_2 f(n) + C') \rightarrow -\infty, \end{aligned} \quad (32)$$

as $n \rightarrow 0$, where $d = \lceil L/(l-1) \rceil = 2^{\frac{2}{3}}$ and $C' = \log_2(2e)$, it follows that

$$\lim_{n \rightarrow \infty} \epsilon(n) = 0. \quad (33)$$

Now (29), (30) and (33) complete the proof.