Adaptive Preimage Resistance and Permutation-based Hash Functions

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 indifferentiable 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. By using such compression functions as building blocks, we obtain a method for construction of permutation-based pseudorandom oracles that is comparable to the Sponge construction [4] both in terms of security and efficiency.

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, 10, 11, 13, 15, 21, 23]. 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, 16, 17, 20]. 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 [12], to name a few, these notions do not seem to suffice as the security requirement for the underlying hash functions. Instead, the security of the protocols 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 indifferentiability framework [14]. To the authors' knowledge, the Sponge construction [4] is the first construction of permutation-based hash functions whose security is proved in the indifferentiability framework. Later, Dodis et. al. proved the security of the MD6 compression function based on a public random permutation [8]. The MD6 compression function can be used for construction of a pseudorandom oracle via appropriate variants of the Merkle-Damgård transform. However, any of the constructions does not achieve the indifferentiability as high as the birthday bound (for the size of the underlying permutation) without significant degradation of efficiency.

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 the same as the NMAC construction [7]. Using the "filtering" function, we do not require the indifferentiability of the compression function from a public random function. However, we found that weaker notions such as collision resistance and preimage resistance do not guarantee the indifferentiability of the resulting hash function 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 strengthened version of preimage resistance. In a conventional definition of preimage resistance, an adversary receives a random point in the range of a hash function, and tries to find a preimage of the point by making adaptive queries to the underlying ideal primitives. Even in a stronger version of the definition, the target range point is fixed before the adversary begins to make queries. In our extension, 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 premiage 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 as long as the underlying compression function is collision resistant (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 [16]. We note that their work is based on a complexity-theoretic definition of collision resistance.

Shortly before submission of this paper, we learned that an independent work [9] introduced a security notion of *preimage awareness*. The authors proved 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. Since any hash function that is both collision resistant and adaptive preimage resistant can be shown to be preimage aware, Theorem 1 and Theorem 2 of this paper are immediate from the results of [9]. However, we note that adaptive preimage resistance is a separate notion from collision resistance, and a weaker notion than preimage awareness. We also emphasize that our main result is to give a simple proof of adaptive preimage resistance for linearly-dependent permutation-based compression functions analyzed in [16, 17].

Results We summarize our results as follows.

- 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 composed with a public random function to yield a hash function that is indifferentiable from a random oracle. We also show that the Merkle-Damgård transform preserves adaptive preimage resistance as long as the underlying compression function is collision resistant.
- 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 [17]. Our approach allows us to prove their preimage resistance within the proof of adaptive preimage resistance. The adaptive preimage resistance of LP231 is guaranteed up to $O\left(2^{n/2}/n\right)$ queries. This bound is better than the one that Dodis et. al. [9] presented for the Shrimpton and Stam's compression function [19].
- With a filtering function built upon public random permutations, we obtain an alternative construction of permutation-based pseudorandom oracles, comparable to the Sponge construction. In this paper, we do not propose any specific construction of the filtering function. However, even applying the Sponge construction to the (fixed-size) output of an "LP231-based" MD hash function would yield a pseudorandom oracle that in terms of efficiency outperforms the original Sponge construction. We also might use $F(\cdot) = E_{(\cdot)}(IV)$ as a filtering function, where $E_{(\cdot)}(\cdot)$ is an ideal cipher and IV is a fixed constant. Since F is a truly random function, we can obtain a hash function of rate $1/3$ that achieves the indifferentiability from a random oracle up to $O\left(2^{n/2}/n\right)$ queries. In this case, we cannot say that the resulting hash function is fully permutation-based, while this might be acceptable in terms of efficiency since the encryption is called only once per message.

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. For any set $U \subset \mathbb{F}_{2^n}$ and $a \in \mathbb{F}_{2^n}$, we use the notations $U + a = \{u + a : u \in U\}$, $aU = \{au : u \in U\}$ and $\overline{U} = I_n \backslash U$. For any miltisets $U,\ V \subset \mathbb{F}_{2^n}$, let $U + V = \{u + v : u \in U,\ v \in V\}$, where $U + V$ is also a multiset. The multiplicity of u in a multiset U is denoted as $\mathsf{mult}_U(u)$.

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

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

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 $\pi_1, \ldots, \pi_k \in \Pi_n$ as follows.

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

where $(w_1, \ldots, w_r)$ is computed by the algorithm described in Figure 1(a). In [17], 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).

Algorithm $H[A](v_1, \ldots, v_m)$

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 $(w_1, \ldots, w_r)$

(a) $H[A]$ for $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(m,k,r)$

Algorithm $H[A](v_1, v_2)$

$x_1 \leftarrow a_{11} v_1 + a_{12} v_2$
$y_1 \leftarrow \pi_1(x_1)$
$x_2 \leftarrow a_{21} v_1 + a_{22} v_2 + a_{23} y_1$
$y_2 \leftarrow \pi_2(x_2)$
$x_3 \leftarrow a_{31} v_1 + a_{32} v_2 + a_{33} y_1 + a_{34} y_2$
$y_3 \leftarrow \pi_3(x_3)$
$w \leftarrow a_{41} v_1 + a_{42} v_2 + a_{43} y_1 + a_{44} y_2 + a_{45} y_3$
return w

(b) $H[A]$ for $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(2,3,1)$

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}, $i = 1, \ldots, k$, we execute the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{coll}}$ 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 $\mathsf{Map}_H(\mathcal{Q}) \subset I_n^m \times I_n^r$ is defined to be the set

of pairs (v, w) such that there exist evaluations $(i, x_i, y_i) \in \mathcal{Q}$, $i = 1, \ldots, k$, satisfying the following equations.

$$x_i = \sum_{j=1}^{m} a_{ij} v_j + \sum_{j=1}^{i-1} a_{i(m+j)} y_j, \qquad i = 1, \ldots, k,$$

$$w_i = \sum_{j=1}^{m} a_{(k+i)j} v_j + \sum_{j=1}^{k} a_{(k+i)(m+j)} y_j, \qquad i = 1, \ldots, r, \tag{2}$$

for $v = (v_1, \ldots, v_m)$ and $w = (w_1, \ldots, w_r)$. Informally, $\mathsf{Map}_H(\mathcal{Q})$ 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

$$\mathbf{Adv}_H^{\mathsf{coll}}(\mathcal{A}) = \mathsf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{coll}} = 1\right]. \tag{3}$$

The probability is taken over the random permutations $\pi_1, \ldots, \pi_k$, and $\mathcal{A}$'s coins (if any). For $q > 0$, we define $\mathbf{Adv}_H^{\mathsf{coll}}(q)$ as the maximum of $\mathbf{Adv}_H^{\mathsf{coll}}(\mathcal{A})$ over all adversaries $\mathcal{A}$ making at most q queries.

The preimage resistance of H is quantified similarly using the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{pre}}$ described in Figure 2(b). The adversary $\mathcal{A}$ takes as input a random $w \in I_n$ before it begins making queries to $\pi_i^{\pm 1}$, $i = 1, \ldots, k$. The *preimage-finding advantage* of $\mathcal{A}$ is defined to be

$$\mathbf{Adv}_H^{\mathsf{pre}}(\mathcal{A}) = \mathsf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{pre}} = 1\right]. \tag{4}$$

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

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{coll}}$

$\pi_1, \ldots, \pi_k \xleftarrow{\$} \Pi_n$
$\mathcal{A}^{\pi_1^{\pm 1}, \ldots, \pi_k^{\pm 1}}$ updates $\mathcal{Q}$
if $\exists\, v \neq v'$, w s.t. (v, w), $(v', w) \in \mathsf{Map}_H(\mathcal{Q})$ **then**
  output 1
else
  output 0

(a) Quantification of collision resistance

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{pre}}$

$\pi_1, \ldots, \pi_k \xleftarrow{\$} \Pi_n$
$w \xleftarrow{\$} I_n$
$\mathcal{A}(w)^{\pi_1^{\pm 1}, \ldots, \pi_k^{\pm 1}}$ updates $\mathcal{Q}$
if $\exists\, v$ such that $(v, w) \in \mathsf{Map}_H(\mathcal{Q})$ **then**
  output 1
else
  output 0

(b) Quantification of preimage resistance

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

Indifferentiability The indifferentiability framework was introduced by Maurer et al. in [14] 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 indifferentiable from $\mathcal{P}$ if the existence of such a simulator is proved. The indifferentiability 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 indifferentiability. For more comprehensive introduction of the indifferentiability framework, we refer to [7, 14].

Definition 1. *A Turing machine $\mathcal{C}$ with oracle access to an ideal primitive $\mathcal{F}$ is said to be (q, ϵ)-indifferentiable 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*

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

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 $\pi_i^{\pm 1}$, $i = 1, \ldots, k$, the adaptive preimage resistance of H is quantified by the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{a-pre}}$ described in Figure 3. At any point during the experiment, the adversary $\mathcal{A}$ can choose a "commitment" point $w \in I_n^r \backslash \mathsf{Range}_H(\mathcal{Q})$, where

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

Then the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{a-pre}}$ 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

$$\mathbf{Adv}_H^{\mathsf{a-pre}}(\mathcal{A}) = \mathsf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{a-pre}} = 1\right]. \tag{5}$$

For $q > 0$, we define $\mathbf{Adv}_H^{\mathsf{a-pre}}(q)$ as the maximum of $\mathbf{Adv}_H^{\mathsf{a-pre}}(\mathcal{A})$ over all adversaries $\mathcal{A}$ such that the total number of queries and commitments is not greater than q. From the definition, it is easy to prove that any adaptive preimage resistant function is preimage resistant. That is, for any compression function $H = H[A]$, $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(m, k, r)$, it follows that

$$\mathbf{Adv}_H^{\mathsf{pre}}(q) \leq \mathbf{Adv}_H^{\mathsf{a-pre}}(q + 1). \tag{6}$$

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{a-pre}}$

$\pi_1, \ldots, \pi_k \xleftarrow{\$} \Pi_n$
$\mathcal{A}^{\pi_1^{\pm 1}, \ldots, \pi_k^{\pm 1}}$ updates $\mathcal{Q}$ and $\mathcal{L}$ (in an arbitrarily interleaved order)
if $\exists\, v$ such that $(v, w) \in \mathsf{Map}_H(\mathcal{Q})$ for some $w \in \mathcal{L}$ **then**
  output 1
else
  output 0

Fig. 3. Experiment for quantification of adaptive preimage resistance

3.2 Construction of Hash Functions

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

Function $MD^H(p_1, \ldots, p_l)$

$x_0 \leftarrow IV$
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 $H : I_s \times I_t \to I_s$ be a compression function, and let $MD^H : I_t^* \to I_s$ be the Merkle-Damgård transform based on the function H. Then it holds that*

$$\mathbf{Adv}^{\mathsf{a-pre}}_{MD^H}(q) \leq \mathbf{Adv}^{\mathsf{coll}}_{H}(q) + \mathbf{Adv}^{\mathsf{a-pre}}_{H}(q).$$

Proof. Let $\mathcal{A}$ be an adaptive preimage-finding adversary for MD^H such that

$$\epsilon = \mathbf{Adv}^{\mathsf{a-pre}}_{MD^H}(q) = \mathbf{Adv}^{\mathsf{a-pre}}_{MD^H}(\mathcal{A}).$$

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 $\mathsf{Map}_H(\mathcal{Q})$. $\mathcal{A}^*$ also grows a directed graph $\mathcal{T}$ on I_s, where $((c_1, p), c_2) \in \mathsf{Map}_H(\mathcal{Q})$ if and only if $\overrightarrow{c_1c_2} \in \mathcal{T}$ with label p.
- When $\mathcal{A}$ chooses a commitment point c, $\mathcal{A}^*$ finds an element $\theta(c) = c_0$ such that there exists a path $\overrightarrow{c_0c_1}, \overrightarrow{c_1c_2} \ldots, \overrightarrow{c_{l-1}c_l} \in \mathcal{T}$ 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. □

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

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

$$\epsilon = \mathbf{Adv}^{\mathsf{coll}}_{H}(q) + \mathbf{Adv}^{\mathsf{a-pre}}_{H}(q).$$

Proof. For simplicity of notations, we provide a proof for linearly-determined permutation-based compression functions $H = H[A]$, $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(m, k, r)$, with oracle access to public random permutations $\pi_1, \ldots, \pi_k$.

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

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

for $w \in I_n^r$. The simulator $\mathcal{S}^G$ faithfully responds to the queries $\pi_i^{\pm 1}(\cdot)$, $i = 1 \ldots, k$. 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 $\mathsf{Inv}_{H,w}(\mathcal{Q})$ 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 $\mathsf{Map}_H(\mathcal{Q})$ whenever it updates the query history $\mathcal{Q}$. The update of $\mathsf{Map}_H(\mathcal{Q})$ requires $O(q^{k-1})$ steps. On a query $F(w)$, $\hat{\mathcal{S}}^G$ chooses an arbitrary element in $\mathsf{Inv}_{H,w}(\mathcal{Q})$ by searching the list $\mathsf{Map}_H(\mathcal{Q})$. It does not need to find every element of $\mathsf{Inv}_{H,w}(\mathcal{Q})$.

A distinguisher interacts with one of two systems: one consists of a set $\mathcal{F} = \{\pi_1, \ldots, \pi_k, F\}$ 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.

Simulator $\mathcal{S}^G$

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

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

- 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.

Now the rigorous proof of the theorem 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 $\pi_1, \ldots, \pi_k$, that are required to compute $H[A](v)$.
2. Each game records a query history $\mathcal{Q}$ for $\pi_1, \ldots, \pi_k$. 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 $\mathsf{Sample}\text{-}G$ is called only within the subroutine $\mathsf{Sample}\text{-}F$ in game $\mathcal{G}$. Therefore, $\mathsf{Sample}\text{-}G$ can be merged into $\mathsf{Sample}\text{-}F$, 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 $\mathsf{Inv}_{H,w}(\mathcal{Q}) = \{v\}$. Assuming $F(w) = \perp$, the subroutine $\mathsf{Sample}\text{-}F$ returns $\mathsf{Sample}\text{-}G(v)$. On the contrary, suppose that $F(w)$ is already defined, say at the j-th query. Then, it should be the case $\mathsf{Inv}_{H,w}(\mathcal{Q}) = \{v\}$ at the point where the subroutine $\mathsf{Sample}\text{-}F(w)$ is called for the first time; if $\mathsf{Inv}_{H,w}(\mathcal{Q}) = \emptyset$, 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 $\mathsf{Inv}_{H,w}(\mathcal{Q})$ 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 $\mathsf{Sample}\text{-}F$ returns $\mathsf{Sample}\text{-}G(v)$. 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 $\mathsf{Pr}\left(\mathcal{D}^{\mathcal{G}} \text{ sets } \mathsf{bad}_1\right) \leq \mathbf{Adv}_H^{\mathsf{coll}}(\mathcal{A})$. Also, $\mathcal{D}$ can be transformed into an adaptive preimage-finding adversary $\mathcal{B}$ such that $\mathsf{Pr}\left(\mathcal{D}^{\mathcal{G}} \text{ sets } \mathsf{bad}_2\right) \leq \mathbf{Adv}_H^{\mathsf{a-pre}}(\mathcal{B})$. Such an adversary $\mathcal{B}$ runs $\mathcal{D}$ as a subroutine and, when $\mathcal{D}$ makes a query $F(w)$ such that $\mathsf{Inv}_{H,w}(\mathcal{Q}) = \emptyset$, chooses the

Games $\mathcal{G}$

Initialize

$\pi_1, \ldots, \pi_k \xleftarrow{\$} \Pi_n$

Interface $\pi_i(x)$ $(i = 1, \ldots, k)$

if $\pi_i(x)$ makes a collision for H **then**
  $\mathsf{bad}_1 \leftarrow$ **true**
else if $\pi_i(x)$ determines $w = H(v)$ s.t. $F(w) \neq \perp$ **then**
  $\mathsf{bad}_2 \leftarrow$ **true**
return $\pi_i(x)$

Interface $\pi_i^{-1}(y)$ $(i = 1, \ldots, k)$

if $\pi_i^{-1}(y)$ makes a collision for H **then**
  $\mathsf{bad}_1 \leftarrow$ **true**
else if $\pi_i^{-1}(y)$ determines $w = H(v)$ s.t. $F(w) \neq \perp$ **then**
  $\mathsf{bad}_2 \leftarrow$ **true**
return $\pi_i^{-1}(y)$

Interface $F(w)$

return Sample-$F(w)$

Interface $G(v)$

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 Sample-$F(w)$

Subroutine Sample-$F(w)$

if $F(w) = \perp$ **then**
  if $\mathsf{Inv}_{H,w}(\mathcal{Q}) = \emptyset$ **then**
    $F(w) \xleftarrow{\$} I_n$
  else
    $v \xleftarrow{\$} \mathsf{Inv}_{H,w}(\mathcal{Q})$
    $F(w) \leftarrow$ Sample-$G(v)$
return $F(w)$

Subroutine Sample-$G(v)$

if $G(v) = \perp$ **then**
  $G(v) \xleftarrow{\$} I_n$
return $G(v)$

Game $\mathcal{H}$

Initialize

$\pi_1, \ldots, \pi_k \xleftarrow{\$} \Pi_n$

Interface $\pi_i(x)$ $(i = 1, \ldots, k)$

return $\pi_i(x)$

Interface $\pi_i^{-1}(y)$ $(i = 1, \ldots, k)$

return $\pi_i^{-1}(y)$

Interface $F(w)$

return Sample-$F(w)$

Interface $G(v)$

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 Sample-$F(w)$

Subroutine Sample-$F(w)$

if $F(w) = \perp$ **then**
  $F(w) \xleftarrow{\$} I_n$
return $F(w)$

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

element w as a commitment. Therefore we conclude that

$$\left|\mathsf{Pr}\left[\mathcal{D}^{F\circ H[A],\mathcal{F}}=1\right]-\mathsf{Pr}\left[\mathcal{D}^{G,\mathcal{S}^G}=1\right]\right| \le \mathsf{Pr}\left(\mathcal{D}^{\mathcal{G}} \text{ sets } \mathsf{bad}_1\right)+\mathsf{Pr}\left(\mathcal{D}^{\mathcal{G}} \text{ sets } \mathsf{bad}_2\right)$$
$$\le \mathbf{Adv}_H^{\mathsf{coll}}(q)+\mathbf{Adv}_H^{\mathsf{a-pre}}(q), \tag{7}$$

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

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 in a simpler way than the one given in [17]. 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.

$$\begin{aligned} x_1 &= a_{11}v_1 + a_{12}v_2 \\ x_2 &= a_{21}v_1 + a_{22}v_2 + a_{23}y_1 \\ x_3 &= a_{31}v_1 + a_{32}v_2 + a_{33}y_1 + a_{34}y_2 \\ w &= a_{41}v_1 + a_{42}v_2 + a_{43}y_1 + a_{44}y_2 + a_{45}y_3. \end{aligned} \tag{8}$$

If we regard every variable in the system (8) as constant except four variables v_1, v_2, x_1 and y_1, then the system (8) is rewritten as the following system of equations in the four variables.

$$M_1(A)\begin{bmatrix} v_1 \\ v_2 \\ x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ x_2 \\ x_3 + a_{34}y_2 \\ a_{44}y_2 + a_{45}y_3 + w \end{bmatrix}, \quad M_1(A) = \begin{bmatrix} a_{11} & a_{12} & 1 & 0 \\ a_{21} & a_{22} & 0 & a_{23} \\ a_{31} & a_{32} & 0 & a_{33} \\ a_{41} & a_{42} & 0 & a_{43} \end{bmatrix}. \tag{9}$$

Similarly, from the system (8) of equations, we obtain a system of equations in variables v_1, v_2, x_2 and y_2,

$$M_2(A)\begin{bmatrix} v_1 \\ v_2 \\ x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ a_{23}y_1 \\ x_3 + a_{33}y_1 \\ a_{43}y_1 + a_{45}y_3 + w \end{bmatrix}, \quad M_2(A) = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 1 & 0 \\ a_{31} & a_{32} & 0 & a_{34} \\ a_{41} & a_{42} & 0 & a_{44} \end{bmatrix}, \tag{10}$$

and a system of equations in variables v_1, v_2, x_3 and y_3,

$$M_3(A)\begin{bmatrix} v_1 \\ v_2 \\ x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 + a_{23}y_1 \\ a_{33}y_1 + a_{34}y_2 \\ a_{43}y_1 + a_{44}y_2 + w \end{bmatrix}, \quad M_3(A) = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & 1 & 0 \\ a_{41} & a_{42} & 0 & a_{45} \end{bmatrix}. \tag{11}$$

If the matrices $M_1(A)$, $M_2(A)$ and $M_3(A)$ are all invertible, then we can solve the system (9), (10) and (11) to obtain equations of the following form.

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

where the coefficients $\alpha_{ij} \in \mathbb{F}_{2^n}$ and $\beta_{ij} \in \mathbb{F}_{2^n}$ are determined by the matrix A for $1 \le i \le 3$ and $1 \le j \le 5$. Let

$$\mathcal{M}^*_{\mathbb{F}_{2^n}}(2,3,1) = \{A \in \mathcal{M}_{\mathbb{F}_{2^n}}(2,3,1) : M_i(A) \text{ is invertible for } 1 \le i \le 3, \\ \alpha_{ij} \neq 0 \text{ and } \beta_{ij} \neq 0 \text{ for } 1 \le i \le 3 \text{ and } 1 \le j \le 4\}. \quad (12)$$

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 $\mathbb{F}_2$. Then

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

is contained in $\mathcal{M}^*_{\mathbb{F}_{2^{128}}}(2,3,1)$.

Remark 1. Our condition that consists of 27 inequalities is not equivalent to the "independence criterion" described in [17]. The independence criterion is the set of 16 inequalities on the coefficients $a_{ij}'s$. Nevertheless, most choices of the coefficients would satisfy both conditions.

4.2 Security Bounds of $H[A]$

In this section, we analyze adaptive preimage 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 two lemmas.

Lemma 1. *Suppose that an adversary $\mathcal{A}$ makes q $(\le 2^{n-1})$ 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 α_1, $\alpha_2 \in \mathbb{F}_{2^n}$. Then, for $l > 0$, it holds that

$$\mathsf{prob}_1(l) = \mathsf{Pr}[\mathcal{A} \textit{ sets } \mathsf{mult}_U(c) \ge l \textit{ for some } c \in \mathbb{F}_{2^n}] \le 2^n \binom{q}{l} \left(\frac{1}{2^{n-1}}\right)^l.$$

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 "$\mathsf{mult}_U(c^*) \ge l$" 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

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

and

$$\mathsf{Pr}[\mathcal{A} \text{ sets } \mathsf{mult}_U(c) \ge l \text{ for some } c \in \mathbb{F}_{2^n}] \le \sum_{c^* \in \mathbb{F}_{2^n}} \mathsf{Pr}[\mathsf{mult}_U(c^*) \ge l],$$

it follows that

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

□

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

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

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 $l > 1$, it holds that

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

Proof. First, we define multisets

$$U_1 = \{\alpha_{11}x + \alpha_{12}y : (1, x, y) \in \mathcal{Q}\} \text{ and } U_2 = \{\alpha_{21}x + \alpha_{22}y : (2, x, y) \in \mathcal{Q}\}.$$

Note that $U = U_1 + U_2$. Given a query history $\mathcal{Q}$, let

$$V_1 = \{c \in \mathbb{F}_{2^n} : \mathsf{mult}_{U_1}(c) = l - 1\} \text{ and } V_2 = \{c \in \mathbb{F}_{2^n} : \mathsf{mult}_{U_2}(c) = l - 1\}.$$

Then we define games $\mathcal{G}$ and $\mathcal{H}$ as described in Figure 6. In the description of the games $\mathcal{G}$ and $\mathcal{H}$, we use the following notations for $i = 1, 2$.

- $\mathsf{Domain}_{\pi_i}(\mathcal{Q}) = \{x \in I_n : (i, x, y) \in \mathcal{Q} \text{ for some } y \in I_n\}$.
- $\mathsf{Range}_{\pi_i}(\mathcal{Q}) = \{y \in I_n : (i, x, y) \in \mathcal{Q} \text{ for some } x \in I_n\}$.

Game $\mathcal{G}$ $\boxed{\textbf{Game } \mathcal{H}}$

<u>Initialize</u>

$\mathcal{Q} \leftarrow \emptyset$

<u>Interface $\pi_i(x)$</u> $(i = 1, 2)$

$\pi_i(x) \xleftarrow{\$} \overline{\mathsf{Range}}_{\pi_i}(\mathcal{Q})$
if $\alpha_{i1}x + \alpha_{i2}\pi_i(x) \in V_i$ **then**
  $\mathsf{bad}_1 \leftarrow$ **true**
  $\boxed{\pi_i(x) \xleftarrow{\$} \overline{\mathsf{Range}}_{\pi_i}(\mathcal{Q}) \cap \overline{\alpha_{i2}^{-1}(V_i + \alpha_{i1}x)}}$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(i, x, \pi_i(x))\}$
return $\pi_i(x)$

<u>Interface $\pi_i^{-1}(y)$</u> $(i = 1, 2)$

$\pi_i^{-1}(y) \xleftarrow{\$} \overline{\mathsf{Domain}}_{\pi_i}(\mathcal{Q})$
if $\alpha_{i1}\pi_i^{-1}(y) + \alpha_{i2}y \in V_i$ **then**
  $\mathsf{bad}_1 \leftarrow$ **true**
  $\boxed{\pi_i^{-1}(y) \xleftarrow{\$} \overline{\mathsf{Domain}}_{\pi_i}(\mathcal{Q}) \cap \overline{\alpha_{i1}^{-1}(V_i + \alpha_{i2}y)}}$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(i, \pi_i^{-1}(y), y)\}$
return $\pi_i^{-1}(y)$

<u>Finalize</u>

if $\mathsf{mult}_U(c) \geq L$ **then** $\mathsf{bad}_2(c) \leftarrow$ **true**

Fig. 6. Games $\mathcal{G}$ and $\mathcal{H}$. $\mathcal{H}$ includes the boxed statements

Suppose that the adversary $\mathcal{A}$ interacts with games $\mathcal{G}$ and $\mathcal{H}$. Then it follows that

$$\begin{aligned} \mathsf{prob}_2(L) &= \mathsf{Pr}\left[\mathcal{G}^{\mathcal{A}} \text{ sets } \mathsf{bad}_2(c) \text{ for some } c \in \mathbb{F}_{2^n}\right] \\ &\leq \mathsf{Pr}\left[\mathcal{G}^{\mathcal{A}} \text{ sets } \mathsf{bad}_1\right] + \mathsf{Pr}\left[\mathcal{H}^{\mathcal{A}} \text{ sets } \mathsf{bad}_2(c) \text{ for some } c \in \mathbb{F}_{2^n}\right]. \end{aligned} \tag{13}$$

If the event "$\alpha_{i1}x^* + \alpha_{i2}\pi_i(x^*) \in V_i$" occurs for a query $\pi_i(x^*)$ or the event "$\alpha_{i1}\pi_i^{-1}(y^*) + \alpha_{i2}y^* \in V_i$" occurs for a query $\pi_i^{-1}(y^*)$ during interaction with game $\mathcal{G}$, then $\mathcal{A}$ obtains $c \in \mathbb{F}_{2^n}$ such that $\mathsf{mult}_{U_i}(c) = l$, where $i = 1, 2$. Therefore, by using Lemma 1, we obtain the following estimation.

$$\mathsf{Pr}\left[\mathcal{G}^{\mathcal{A}} \text{ sets } \mathsf{bad}_1\right] \leq 2\mathsf{prob}_1(l) \leq 2^{n+1}\binom{q}{l}\left(\frac{1}{2^{n-1}}\right)^l. \tag{14}$$

Now we estimate the probability

$$\mathsf{Pr}\left[\mathcal{H}^{\mathcal{A}} \text{ sets } \mathsf{bad}_2(c^*)\right] = \mathsf{Pr}\left[\mathcal{A} \text{ sets } \mathsf{mult}_U(c^*) \geq L\right], \tag{15}$$

for a fixed $c^* \in \mathbb{F}_{2^n}$. Since it holds that $\mathsf{mult}_{U_1}(c) \leq l-1$ and $\mathsf{mult}_{U_2}(c) \leq l-1$ for any $c \in \mathbb{F}_{2^n}$ during interaction with game $\mathcal{H}$, $\mathsf{mult}_U(c^*)$ increases at most by $(l-1)$ at each query. Therefore, if it holds that $\mathsf{mult}_U(c^*) \geq L$ at the end of the interaction, there should be at least $d = \lceil L/(l-1) \rceil$ queries that increase $\mathsf{mult}_U(c^*)$ at least by one. Since, at the j-th query,

$$\left|\overline{\mathsf{Range}}_{\pi_i}(\mathcal{Q}) \cap \overline{\alpha_{i2}^{-1}\left(V_i + \alpha_{i1}x\right)}\right| \geq 2^n - 2(j-1), \tag{16}$$

and

$$\left|\overline{\mathsf{Domain}}_{\pi_i}(\mathcal{Q}) \cap \overline{\alpha_{i1}^{-1}\left(V_i + \alpha_{i2}y\right)}\right| \geq 2^n - 2(j-1), \tag{17}$$

for $i = 1, 2$, the probability that $\mathsf{mult}_U(c^*)$ increases at least by one is not greater than $(j-1)/(2^n - 2(j-1)) \leq q/2^{n-1}$. Therefore, we conclude that

$$\begin{aligned}\mathsf{Pr}\left[\mathcal{H}^{\mathcal{A}} \text{ sets } \mathsf{bad}_2(c) \text{ for some } c \in \mathbb{F}_{2^n}\right] &\leq \sum_{c^* \in \mathbb{F}_{2^n}} \mathsf{Pr}\left[\mathcal{H}^{\mathcal{A}} \text{ sets } \mathsf{bad}_2(c^*)\right] \\ &\leq 2^n \binom{q}{d} \left(\frac{q}{2^{n-1}}\right)^d. \end{aligned} \tag{18}$$

The lemma is followed from (13), (14) and (18). □

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 updates a query history $\mathcal{Q}$. Then, for $L > 0$ and $l > 1$, it holds that*

$$\mathbf{Adv}_H^{\mathsf{a-pre}}(q) \leq \frac{q^2 L}{2^{n-1}} + 6\epsilon, \tag{19}$$

as well as,

$$\mathbf{Adv}_H^{\mathsf{coll}}(q) \leq \frac{q^2 L^2}{2^{n-1}} + 6\epsilon, \quad \mathbf{Adv}_H^{\mathsf{pre}}(q) \leq \frac{qL}{2^{n-1}} + 6\epsilon, \tag{20}$$

where

$$\epsilon = 2^{n+1}\binom{q}{l}\left(\frac{1}{2^{n-1}}\right)^l + 2^n\binom{q}{\lceil L/(l-1)\rceil}\left(\frac{q}{2^{n-1}}\right)^{\lceil L/(l-1)\rceil}. \tag{21}$$

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

$$\Gamma_{\Phi_1}(x^*, w^*) = \{((2, x_2, y_2), (3, x_3, y_3)) \in \mathcal{Q}^2 : x^* = \Phi_1(x_2, y_2, x_3, y_3, w^*)\},$$

and

$$\Gamma_{\Psi_1}(y^*, w^*) = \{((2, x_2, y_2), (3, x_3, y_3)) \in \mathcal{Q}^2 : y^* = \Psi_1(x_2, y_2, x_3, y_3, w^*)\}.$$

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

$$|\Gamma_{\Phi_i}(x^*, w^*)| < L, \ \forall\, x^*, w^* \in \mathbb{F}_{2^n}, \tag{22}$$

except with probability ϵ. For example, suppose that $|\Gamma_{\Phi_1}(x^*, w^*)| \geq L$ for some $x^*, w^* \in \mathbb{F}_{2^n}$. It means that $\mathsf{mult}_U(x^* + \alpha_{15}w^*) \geq L$, where

$$U = \{\alpha_{11}x_2 + \alpha_{12}y_2 + \alpha_{13}x_3 + \alpha_{14}y_3 : (2, x_2, y_2), (3, x_3, y_3) \in \mathcal{Q}\}.$$

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

$$|\Gamma_{\Psi_i}(y^*, w^*)| < L, \ \forall\, y^*, w^* \in \mathbb{F}_{2^n} \tag{23}$$

is not greater than ϵ. Let $\mathcal{E}_{1i}$ and $\mathcal{E}_{2i}$, respectively, denote the event (22) and (23) 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 $\mathsf{Pr}[\mathcal{E}] \leq 6\epsilon$.

Adaptive Preimage Resistance. For the proof of (19), it is sufficient to show that

$$\mathsf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{a-pre}} = 1 \mid \mathcal{E}\right] \leq \frac{q^2 L}{2^{n-1}}, \tag{24}$$

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 $(v, w^*) \in \mathsf{Map}_H(\mathcal{Q})$ 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 $L/\left(2^n - (j-1)\right) \leq L/2^{n-1}$, conditioned on $\mathcal{E}$, since $|\Gamma_{\Phi_1}(x^*, w^*)| \leq L$. Now the inequality (24) is proved since $|\mathcal{Q}| \leq q$ and $|\mathcal{L}| \leq q$.

Preimage Resistance. Let $\mathcal{A}$ be a preimage-finding adversary, and let w^* be the random point that the environment of $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{pre}}$ has chosen at the beginning of the experiment. Then the probability, conditioned on $\mathcal{E}$, that the j-th query determines an evaluation (v, w^*) in $\mathsf{Map}_H(\mathcal{Q})$ for some v is not greater than $L/2^{n-1}$. Therefore, we have

$$\mathsf{Pr}\left[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{pre}} = 1 \mid \mathcal{E}\right] \leq \frac{qL}{2^{n-1}}, \tag{25}$$

and proves the inequality for $\mathbf{Adv}_H^{\mathsf{pre}}(q)$. □

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 $\lim_{n\to\infty} f(n) = 0$, then $\lim_{n\to\infty} \mathbf{Adv}_{H_n}^{\mathsf{pre}}(q) = 0$.*

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

$$\mathbf{Adv}_H^{\mathsf{pre}}(q) \leq \frac{qL}{2^{n-1}} + 6\epsilon. \tag{26}$$

For the first term on the right-hand side of the bound (26), we have

$$\lim_{n\to\infty} \frac{qL}{2^{n-1}} = \lim_{n\to\infty} (4f(n)) = 0. \tag{27}$$

Since

$$\begin{aligned}
\log_2\left(2^{n+1}\binom{q}{l}\left(\frac{1}{2^{n-1}}\right)^l\right) &\leq \log_2\left(2^{n+1}\left(\frac{qe}{l}\right)^l\left(\frac{1}{2^{n-1}}\right)^l\right) \\
&\leq n + 1 + 3\left(\frac{2n}{3} + \log_2 f(n) - n + C\right) \\
&= 3\log_2 f(n) + 3C + 1 \longrightarrow -\infty,
\end{aligned} \tag{28}$$

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

$$\begin{aligned}
\log_2\left(2^n\binom{q}{d}\left(\frac{q}{2^{n-1}}\right)^d\right) &\leq \log_2\left(2^n\left(\frac{qe}{d}\right)^d\left(\frac{q}{2^{n-1}}\right)^d\right) \\
&\leq n + 2^{\frac{n}{3}}\left(2\left(\frac{2n}{3} + \log_2 f(n)\right) - \frac{4n}{3} + C'\right) \\
&= n + 2^{\frac{n}{3}}\left(2\log_2 f(n) + C'\right) \longrightarrow -\infty,
\end{aligned} \tag{29}$$

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

$$\lim_{n\to\infty} \epsilon(n) = 0. \tag{30}$$

Now (26), (27) and (30) complete the proof. □

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 $\lim_{n\to\infty} \mathbf{Adv}_{H_n}^{\mathsf{a-pre}}(q) = 0$.*

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

5 Discussion

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

1. Construct a compression function H that is collision resistant and adaptive preimage resistant based on a small number of public random permutations.
2. Apply the Merkle-Damgård transform to the compression function H. Also, use an appropriate padding scheme so that MD^H is both collision resistant and adaptive preimage resistant.
3. Construct a public random function F based on a small number of public random permutations.
4. Define the full hash function to be the composite $F \circ MD^H$.

As for the third step, it seems to be challenging to build a non-compressing random function with reasonable security and efficiency from public random permutations of the same size. Instead, we might apply the Sponge construction to the (fixed-size) output of the permutation-based MD hash function. Suppose that we choose $H[A]$, $A \in \mathcal{M}_{\mathbb{F}_{2^n}}(2,3,1)$, as the underlying compression function, and the Sponge construction using an independent permutation of size n as the filtering function. Then the security of the resulting hash function is dominated by that of the filtering function, and the rate is given by $1/3$ (ignoring the rate of the filtering function). On the other hand, the full Sponge construction of rate $1/3$ could not achieve security better than $O(2^{n/3})$.

We also might define $F(x) = E_x(IV)$ using an ideal cipher $E_{(\cdot)}(\cdot)$ with a fixed constant IV. Combined with an MD hash function using LP231, we obtain a hash function of rate $1/3$ that achieves the indifferentiability from a random oracle up to $O\left(2^{n/2}/n\right)$ queries. (The collision resistance of LP231 can be improved up to $O\left(2^{n/2}/n\right)$ queries [22].) We cannot say that such construction is fully permutation-based, while this might be acceptable in terms of efficiency since the encryption is called only once per message.

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* proposed and studied in [9]. As another future work, it would be interesting to study whether our approach given in this paper can be extended to a wider class of LP compression functions to give a simple proof of their collision resistance and adaptive preimage resistance, which is not computer-aided.

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 C. Puniya. Merkle-Damgård revisited: How to construct a hash function. Crypto 2005, LNCS 3621, pp. 430–448, Springer-Verlag, 2005.
8. Y. Dodis, L. Reyzin, R. L. Rivest and E. Shen. Indifferentiability of permutation-based compression functions and tree-based modes of operation, with applications to MD6. FSE 2009, To appear. Available at http://www.cs.nyu/ dodis.
9. Y. Dodis, T. Ristenpart and T. Shrimpton. Salvaging Merkle-Damgård for practical applications. Eurocrypt 2009, To appear. Available at http://www.cs.nyu/ dodis.

10. S. Hirose. Provably secure double-block-length hash functions in a black-box model. ICISC 2004, LNCS 3506, pp. 330–342, Springer-Verlag, 2005.
11. S. Hirose. Some plausible construction of double-block-length hash functions. FSE 2006, LNCS 4047, pp. 210–225, Springer-Verlag, 2006.
12. H. Krawczyk. HMQV: 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.
13. 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.
14. U. Maurer, R. Renner and R. Holenstein. Indifferentiability, impossibility results on reductions, and apllications to the random oracle methodology. TCC 2004, LNCS 2951, pp. 21–39, Springer-Verlag, 2008.
15. 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.
16. 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.
17. P. Rogaway and J. Steinberger. Constructing cryptographic hash functions from fixed-key blockciphers. Crypto 2008, LNCS 5157, pp. 433–450, Springer-Verlag, 2008.
18. P. Rogaway and J. Steinberger. Security/efficiency tradeoffs fro permuation-based hashing. Eurocrypt 2008, LNCS 4965, pp. 220–236, Springer-Verlag, 2008.
19. T. Shrimpton and M. Stam. Building a collision-resistant function from non-compressing primitives. ICALP 2008, LNCS 5126, pp. 643–654, Springer-Verlag, 2008.
20. M. Stam. Beyond uniformity: Better security/efficiency tradeoffs for compression functions. Crypto 2008, LNCS 5157, pp. 397–412, Springer-Verlag, 2008.
21. J. Steinberger. The collision intractability of MDC-2 in the ideal-cipher model. Eurocrypt 2007, LNCS 4515, pp. 34–51, Springer-Verlag, 2008.
22. J. Steinberger. private communication.
23. R. Winternitz. A secure one-way hash function built from DES. IEEE Symposium on Information Security and Privacy, pp. 88–90, IEEE Press, 1984.