Hash Functions Based on Three Permutations: A Generic Security Analysis

Bart Mennink and Bart Preneel

Dept. Electrical Engineering, ESAT/COSIC, KU Leuven, and IBBT, Belgium
bart.mennink@esat.kuleuven.be, bart.preneel@esat.kuleuven.be

Abstract. We consider the family of $2n$-to-n-bit compression functions that are solely based on at most three permutation executions and on XOR-operators, and analyze its collision and preimage security. Despite their elegance and simplicity, these designs are not covered by the results of Rogaway and Steinberger (CRYPTO 2008). By defining a carefully chosen equivalence relation on this family of compression functions, we obtain the following results. In the setting where the three permutations π_1, π_2, π_3 are selected independently and uniformly at random, there exist at most four equivalence classes that achieve optimal $2^{n/2}$ collision resistance. Under a certain extremal graph theory based conjecture, these classes are then proven optimally collision secure. Three of these classes allow for finding preimages in $2^{n/2}$ queries, and only one achieves optimal $2^{2n/3}$ preimage resistance (with respect to the bounds of Rogaway and Steinberger, EUROCRYPT 2008). Consequently, a compression function is optimally collision and preimage secure if and only if it is equivalent to $\mathsf{F}(x_1, x_2) = x_1 \oplus \pi_1(x_1) \oplus \pi_2(x_2) \oplus \pi_3(x_1 \oplus x_2 \oplus \pi_1(x_1))$. For compression functions that make three calls to the same permutation we obtain a surprising negative result, namely the impossibility of optimal $2^{n/2}$ collision security: for any scheme, collisions can be found with $2^{2n/5}$ queries. This result casts some doubt over the existence of any (larger) secure permutation-based compression function built only on XOR-operators and (multiple invocations of) a single permutation.

Keywords. Hash function, Permutation-based, Collision resistance, Preimage resistance.

1 Introduction

The traditional recipe for the design of a cryptographic hash function is to base it on one or more block ciphers. Since the late 70s, this methodology developed itself to become the dominating approach in the area of hash function design and plenty of hash functions have been constructed accordingly (either explicitly or implicitly) [3,4,7,8]. These designs are, however, characterized by the fact that the key input to the cipher depends on the input values; this implies that the key schedule has to be strong and that it needs to be executed for every encryption (or for every second encryption), which entails a substantial computational cost. An alternative approach is to fix one or more keys, and restrict the hash function

design to use the block cipher for these keys only. The usage of fixed-key block ciphers, or alternatively *permutations*, additionally causes gain that one does not need to implement an entire block cipher but only a limited number of instantiations of it.

Black, Cochran and Shrimpton [1] were the first to formally study this approach, demonstrating that a $2n$-to-n-bit compression function F using one n-bit permutation π cannot be secure. This result has been generalized by Rogaway and Steinberger [11], and refined by Stam [13] and Steinberger [14]. Consider any mn-to-rn-bit compression function using k n-bit permutations: if $2^{n(2m-2r-k+1)/(k+1)} \geq 17$, collisions can be found in at most $(2^n)^{1-(m-r+1)/(k+1)}$ queries to the underlying primitives, a bound proven by Steinberger in [14] but commonly known as "Stam's bound." Collisions and preimages can even be found in at most $(2^n)^{1-(m-r/2)/k}$ and $(2^n)^{1-(m-r)/k}$ queries respectively, provided the compression function satisfies the "uniformity assumption" [11]. Due to Stam's bound, a $2n$-to-n-bit compression function, which is the simplest case after all, achieves optimal $2^{n/2}$ collision resistance *only if* it employs at least three permutations. Yet, it cannot achieve optimal preimage resistance if it fulfills the uniformity assumption. These observations apply to the "multi-permutation setting", where each of the permutations is generated independently, as well as the "single-permutation setting" where the permutations are the same.

The construction of $2n$-to-n-bit compression functions (based on three permutations) that provably attain optimal collision security, has turned out to be a very challenging exercise. In [10], Rogaway and Steinberger formally proved a broad class of $2n$-to-n-bit compression functions using three distinct permutations and finite field scalar multiplications optimally collision and preimage secure (w.r.t. the bounds of [11]), provided the compression function satisfies a so-called "independence criterion" (a similar result for the single-permutation setting has been obtained by Lee and Kwon [5]). Unfortunately, this technical criterion rules out the most intuitive and elegant type of designs, namely compression functions that are (apart from the three permutations) solely based on XOR-operators. As the proof of [10] extensively relies on its independence criterion, the proof cannot be generalized to compression functions of this type. In [12], Shrimpton and Stam derived a XOR-based compression function, using three one-way functions rather than permutations: $\mathsf{F}(x_1, x_2) = f_1(x_1) \oplus f_3(f_1(x_1) \oplus f_2(x_2))$. This function is proven collision resistant up to $2^{n/2}$ queries (asymptotically), but preimages can be found with high probability after $2^{n/2}$ queries [12]. It has been demonstrated by an automated analysis of Rogaway and Steinberger [10] that the same results hold if f_1, f_2, f_3 are Davies-Meyer-like compression functions using permutations π_1, π_2, π_3, i.e. $f_i(x) = x \oplus \pi_i(x)$, but a formal security analysis has never been given. Since these works, a synthetic formal collision and preimage security analysis of XOR-based compression functions has remained an interesting and important theoretical open problem, because of their elegance and simplicity (the functions only employ XOR-operators) as well as their slight efficiency improvement (XOR-operators are slightly cheaper than finite field multiplications).

Our Contributions. We focus on the entire family of $2n$-to-n-bit compression functions constructed only of three isolated permutations and of XOR-operators, and analyze the security of these functions against information-theoretic adversaries. For each of the functions, we either provide a proof of optimal collision resistance or a collision attack faster than the birthday bound. We also analyze the preimage resistance of the schemes that have optimal collision security.

The approach followed in this work is based on defining an equivalence class on the set of compression functions, and is of independent interest: informally, two compression functions are equivalent if there exists a tight bi-directional preimage and collision security reduction (cf. Def. 3). Consequently, security results of one compression function hold for the entire class, and it suffices to analyze the security of one function per class. In this work we restrict to equivalence reductions that are easy to verify, such as interchanging the inputs to the compression function.

For the *multi-permutation* setting, where the three permutations π_1, π_2, π_3 are assumed to be selected independently and uniformly at random, the results are as follows. A compression function F is optimally collision secure (asymptotically) *if and only if* it is equivalent to one of the four compression functions $\mathsf{F}_1, \ldots, \mathsf{F}_4$:

$$\begin{aligned}
\mathsf{F}_1(x_1, x_2) &= x_2 \oplus \pi_2(x_2) \oplus && \pi_3(x_1 \oplus x_2 \oplus \pi_1(x_1))\,, \\
\mathsf{F}_2(x_1, x_2) &= x_1 \oplus \pi_1(x_1) \oplus \pi_2(x_2) \oplus && \pi_3(x_1 \oplus x_2 \oplus \pi_1(x_1))\,, \\
\mathsf{F}_3(x_1, x_2) &= x_1 \oplus \pi_1(x_1) \oplus && \pi_3(x_1 \oplus x_2 \oplus \pi_1(x_1) \oplus \pi_2(x_2))\,, \\
\mathsf{F}_4(x_1, x_2) &= x_1 \oplus x_2 \oplus \pi_1(x_1) \oplus && \pi_3(x_1 \oplus x_2 \oplus \pi_1(x_1) \oplus \pi_2(x_2))\,.
\end{aligned} \tag{1}$$

These compression functions are depicted in Fig. 1. Not surprisingly, the permutation-based variant of the Shrimpton-Stam compression function [12] is included, it equals F_3. For compression functions non-equivalent to any of $\mathsf{F}_1, \mathsf{F}_2, \mathsf{F}_3, \mathsf{F}_4$, collisions can be found faster than the birthday bound, namely in at most $2^{2n/5}$ queries. Compression functions equivalent to F_2 are proven optimally preimage secure up to $2^{2n/3}$ queries, and compression functions equivalent to $\mathsf{F}_1, \mathsf{F}_3$ or F_4 are additionally shown to achieve tight $2^{n/2}$ preimage security. Therefore, a compression function achieves optimal collision and preimage resistance (w.r.t. the bounds of [11]) if and only if it is equivalent to F_2. Particularly,

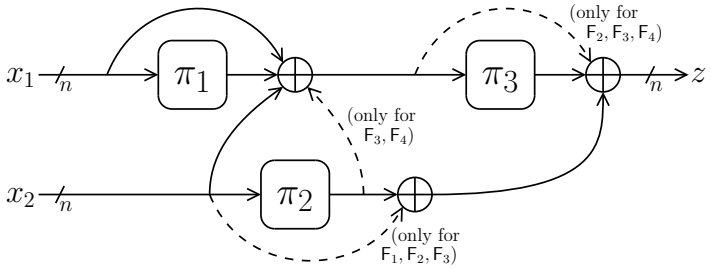

Fig. 1. A graphical representation of the compression functions $\mathsf{F}_1, \ldots, \mathsf{F}_4$ of (1).

this class of functions beats the Shrimpton-Stam compression function [12] with respect to preimage resistance. These results are summarized in Table 1.

A minor part of the results in the multi-permutation setting, more concretely the collision resistance of $\mathsf{F}_1, \mathsf{F}_2$ and F_4 and the preimage resistance of F_2, are based on an extremal graph theory based conjecture. Informally, this conjecture bounds the number of solutions $(x_1, x_2, x_3) \in X_1 \times X_2 \times X_3$ such that $x_2 \oplus x_3 = x_1 \oplus \pi_1(x_1)$, where X_1, X_2, X_3 are three sets of q elements. This conjecture is similar to (but more complex than) a problem posed by Zarankiewicz in 1951 (cf. [2, Ch. 6.2]), and is of independent interest. In the full version of this paper [6], we analyze our conjecture in more detail, provide it with a heuristic argument, and compare it with the conjecture of Zarankiewicz.

Table 1. The security results of this work for the multi-permutation setting. The functions $\mathsf{F}_1, \ldots, \mathsf{F}_4$ are given in (1) and Fig. 1. The equivalence relation is defined in Def. 3. For F_2, the obtained security results are optimal with respect to the bounds of Rogaway and Steinberger [11]. The proofs of the results with appended "[c]" fall back on Conjecture 1.

	collision		preimage	
F equivalent to:	security	attack	security	attack
$\mathsf{F}_1, \mathsf{F}_4$	$2^{n/2}$ [c]	$2^{n/2}$	$2^{n/2}$	$2^{n/2}$
F_2	$2^{n/2}$ [c]	$2^{n/2}$	$2^{2n/3}$ [c]	$2^{2n/3}$
F_3	$2^{n/2}$	$2^{n/2}$	$2^{n/2}$	$2^{n/2}$
none of these	?	$2^{2n/5}$	?	?

In the *single-permutation* setting, where the compression function makes three calls to the same random permutation π, *there does not exist any* compression function that achieves optimal collision resistance. In particular, for any possible function, collisions can be found in at most $2^{2n/5}$ queries, beating the desired birthday bound. This negative result is surprising, given the fair amount of secure functions we have found in the multi-permutation setting. The attacks mainly rely on the fact that the adversary can misuse the single-permutation property by introducing dependencies between the two input values x_1 and x_2. For instance, the function F_2 of (1) satisfies $\mathsf{F}_2(x_1, x_2) = \mathsf{F}_2(x_1, x_2 \oplus x_1 \oplus \pi(x_1))$ in the single-permutation setting. This result raises the interesting question whether (larger) compression functions exist based only on XOR-operators and (more than three invocations of) one single permutation.

Outline. In Sect. 2, we present some background information, and formally describe the set of permutation-based compression functions we have analyzed. In Sect. 3, the equivalence relation on the set of compression functions is formally defined. The main results are given in Sect. 4 for the multi-permutation setting

and in Sect. 5 for the single-permutation setting. We conclude the paper in Sect. 6.

2 Preliminaries

For an integer $n \in \mathbb{N}$, we denote by $\{0,1\}^n$ the set of bit strings of length n. For two bit strings x, y, we denote by $x\|y$ their concatenation and by $x \oplus y$ their bitwise XOR. If $\mathcal{X}$ is a set, by $x \xleftarrow{\$} \mathcal{X}$ we denote the uniformly random sampling of an element from $\mathcal{X}$. For two integers $m, n \in \mathbb{N}$, we denote by $\langle m \rangle_n$ the encoding of m as an n-bit string. By log we denote the logarithm function with respect to base 2. By P_n we denote the set of all permutations operating on n bits. Vectors are denoted as $\mathbf{x}$, and by $\|\mathbf{x}\| = \sum_i |x_i|$ we denote the 1-norm of $\mathbf{x}$. For a matrix A, by $a_{i,j}$ we denote its coefficient at the i^{th} row and j^{th} column. By $\mathbf{a}_{i,*}$ we denote the i^{th} row of A, and by $\mathbf{a}_{*,j}$ its j^{th} column.

2.1 Permutation Based Compression Functions

We consider the following type of $2n$-to-n-bit compression functions. Let π_1, π_2, $\pi_3 \in P_n$ be three permutations. For a binary 4×5 matrix A of the form

$$\mathrm{A} = \left(\begin{array}{cc|ccc} a_{11} & a_{12} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 \\ \hline a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{array}\right), \tag{2}$$

the compression function $\mathsf{F}_{\mathrm{A}} : \{0,1\}^{2n} \to \{0,1\}^n$ is defined as follows:

$$\begin{aligned} \mathsf{F}_{\mathrm{A}}(x_1, x_2) = z\,, \text{ where } \; & y_1 \leftarrow \pi_1(a_{11}x_1 \oplus a_{12}x_2)\,, \\ & y_2 \leftarrow \pi_2(a_{21}x_1 \oplus a_{22}x_2 \oplus a_{23}y_1)\,, \\ & y_3 \leftarrow \pi_3(a_{31}x_1 \oplus a_{32}x_2 \oplus a_{33}y_1 \oplus a_{34}y_2)\,, \\ & z \leftarrow a_{41}x_1 \oplus a_{42}x_2 \oplus a_{43}y_1 \oplus a_{44}y_2 \oplus a_{45}y_3\,. \end{aligned} \tag{3}$$

The function F_{A} is depicted in Fig. 2. If the three permutations are all different, we refer to it as the *multi-permutation* setting. If π_1, π_2, π_3 are equal to one permutation π, we are in the *single-permutation* setting. In total, we thus analyze $2 \cdot 2^{14}$ compression functions. Many of these, however, are trivially weak (cf. Sect. 2.3).

For the single-permutation setting, it is of interest to also consider the case where n-bit constants are added to the inputs to the permutations (e.g. $y_1 \leftarrow \pi_1(a_{11}x_1 \oplus a_{12}x_2 \oplus b_1)$ for $b_1 \in \{0,1\}^n$). This results in many more schemes, but requires a more complex analysis. Therefore, we present our main results on F_{A} of (3), and in App. A we generalize our findings on the single-permutation setting to cover any F_{A} where additional affine transformations on the permutation inputs are taken into account.

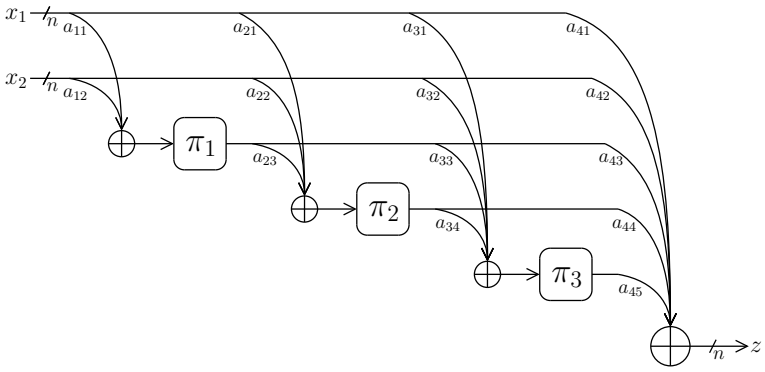

Fig. 2. The permutation-based compression function F_{A} of (3).

2.2 Security Notions

An adversary is a probabilistic algorithm with oracle access to the underlying permutations π_1, π_2, π_3. He can make forward and inverse queries to its oracles, and the queries are stored in a query history $\mathcal{Q}$. By $(x_k, y_k) \in \mathcal{Q}$, for $k \in \{1, 2, 3\}$, we denote that $y_k = \pi_k(x_k)$; the adversary either made a forward query x_k to obtain y_k or an inverse query y_k to obtain x_k. In the remainder, we assume that $\mathcal{Q}$ always contains the queries required for the attack, and we assume that the adversary does not make trivial queries, i.e. queries to which the adversary already knows the answer in advance. In this work we consider information-theoretic adversaries only. This type of adversary has unbounded computational power, and its complexity is measured by the number of queries made to its oracles.

Definition 1. *Let $\mathsf{F}_{\mathrm{A}} : \{0,1\}^{2n} \to \{0,1\}^n$ be a compression function defined by a matrix A of the form (2). Let $\mathcal{A}$ be a collision finding adversary for this compression function. The advantage of $\mathcal{A}$ is defined as*

$$\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}}}(\mathcal{A}) = \mathbf{Pr}\left(\pi_1, \pi_2, \pi_3 \xleftarrow{\$} P_n,\ x, x' \leftarrow \mathcal{A}^{\pi_i, \pi_i^{-1}} :\ x \neq x',\ \mathsf{F}^{\pi_i}_{\mathrm{A}}(x) = \mathsf{F}^{\pi_i}_{\mathrm{A}}(x')\right).$$

By $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}}}(q)$ we denote the maximum advantage, taken over all adversaries making q queries to each of their oracles.

Several definitions for preimage resistance are known, but we opt for everywhere preimage resistance [9], which intuitively guarantees preimage security for every range point.

Definition 2. *Let $\mathsf{F}_{\mathrm{A}} : \{0,1\}^{2n} \to \{0,1\}^n$ be a compression function defined by a matrix A of the form (2). Let $\mathcal{A}$ be an everywhere preimage finding adversary for this compression function. The advantage of $\mathcal{A}$ is defined as*

$$\mathbf{Adv}^{\mathsf{epre}}_{\mathsf{F}_{\mathrm{A}}}(\mathcal{A}) = \max_{z \in \{0,1\}^n} \mathbf{Pr}\left(\pi_1, \pi_2, \pi_3 \xleftarrow{\$} P_n,\ x \leftarrow \mathcal{A}^{\pi_i, \pi_i^{-1}}(z) :\ z = \mathsf{F}^{\pi_i}_{\mathrm{A}}(x)\right).$$

By $\mathbf{Adv}^{\mathsf{epre}}_{\mathsf{F_A}}(q)$ *we denote the maximum advantage, taken over all adversaries making q queries to each of their oracles.*

The security definitions for the single-permutation setting, where the compression function is built on one permutation π, are analogous.

2.3 Invalid Matrices

We will classify the set of optimally collision secure compression functions $\mathsf{F_A}$ of the form described in Sect. 2.1, but for some matrices A the induced compression function will clearly not fulfill the desired security requirements. For instance, if a compression function does not use one or more permutations, attacks faster than the birthday bound can easily be constructed. We introduce the notion of "valid" matrices, in order to rule out compression functions that trivially fail to achieve optimal collision resistance. A matrix A is called "valid" if it satisfies the following properties:

(1) For the j^{th} column ($j = 1, 2$), we have $a_{1j} + a_{2j} + a_{3j} \geq 1$. This requirement ensures that input x_j is used in the computation of at least one permutation. If this would not be the case, collisions can easily be constructed;
(2) For the j^{th} column ($j = 3, 4, 5$), we have $\|\mathbf{a}_{*,j}\| \geq 1$, and for the i^{th} row ($i = 1, 2, 3$), we have $\|\mathbf{a}_{i,*}\| \geq 1$. Notice that if the i^{th} row (resp. j^{th} column) would consist of zeroes only, it means that permutation π_i (resp. π_{j-2}) is not used in the computation, and collisions can be found in at most $2^{n/3}$ queries by Stam's bound [13,14].

In the remainder, we will consider valid matrices A only. By an extensive computation one can show that $2796 < 2^{12}$ out of 2^{14} matrices are valid (for both the single- and multi-permutation setting).

3 Equivalence Classes of Permutation Based Compression Functions

We define an equivalence relation on the set of compression functions $\mathsf{F_A}$. This equivalence relation intuitively describes classes of "equally secure" compression functions, and can be used to reduce the number of compression functions to be analyzed. Indeed, security properties of one compression function naturally convey to all compression functions in the same equivalence class. The equivalence relation is defined in Def. 3, and in Props. 1-4 we describe the four equivalence reductions that will be used in this work.

Definition 3. *Two compression functions* $\mathsf{F_A}$ *and* $\mathsf{F_{A'}}$ *are* equivalent *if for both collision and preimage security there exists a tight reduction from* $\mathsf{F_A}$ *to* $\mathsf{F_{A'}}$*, and vice versa.*

Proposition 1 (x-reduction). *Consider two matrices* $\mathrm{A} = \big(\mathbf{a}_{*,1}\,;\,\mathbf{a}_{*,2}\,;\,\mathbf{a}_{*,3}\,;\,\mathbf{a}_{*,4}\,;\,\mathbf{a}_{*,5}\big)$ *and* $\mathrm{A}' = \big(\mathbf{a}_{*,2}\,;\,\mathbf{a}_{*,1}\,;\,\mathbf{a}_{*,3}\,;\,\mathbf{a}_{*,4}\,;\,\mathbf{a}_{*,5}\big)$*. Then, the compression functions* F_{A} *and* $\mathsf{F}_{\mathrm{A}'}$ *are equivalent. Intuitively, this reduction corresponds to swapping* x_1 *and* x_2*.*

Proposition 2 (XOR-reduction). *Consider a matrix* $\mathrm{A} = \big(\mathbf{a}_{*,1}\,;\,\mathbf{a}_{*,2}\,;\,\mathbf{a}_{*,3}\,;\,\mathbf{a}_{*,4}\,;\,\mathbf{a}_{*,5}\big)$*, and let* $k = \min\{\ i \mid a_{i,2} \neq 0\ \}$ *(notice that* $k \in \{1,2,3\}$ *as* A *is valid). Let* $c_0, \ldots, c_2 \in \{0,1\}$*. Consider the matrix* $\mathrm{A}' = \mathrm{A} \oplus \big(c_0\mathbf{a}_{*,2}\,;\,\mathbf{0}\,;\,[k \geq 2]c_1\mathbf{a}_{*,2}\,;\,[k \geq 3]c_2\mathbf{a}_{*,2}\,;\,\mathbf{0}\big)$*, where* $[X] = 1$ *if* X *holds and* 0 *otherwise. Then, the compression functions* F_{A} *and* $\mathsf{F}_{\mathrm{A}'}$ *are equivalent. Intuitively,* π_k *is the first permutation that incorporates* x_2*, and this reduction represents replacing* x_2 *by* $x_2 \oplus c_0x_1 \oplus \sum_{i=1}^{k-1} c_iy_i$*, where* y_i *is the outcome of the* i^{th} *permutation. Using Prop. 1, the same reduction holds for* x_1*.*

Proposition 3 (π-swap-reduction). *Let* $i \in \{1,2\}$*, and consider a matrix* A *with* $a_{i+1,i+2} = 0$*. Consider the matrix* A' *obtained from* A *by swapping rows* $\mathbf{a}_{i,*}$ *and* $\mathbf{a}_{i+1,*}$ *and consequently swapping columns* $\mathbf{a}_{*,i+2}$ *and* $\mathbf{a}_{*,i+3}$*. Then, the compression functions* F_{A} *and* $\mathsf{F}_{\mathrm{A}'}$ *are equivalent. Intuitively, this reduction corresponds to swapping* π_i *and* π_{i+1}*, which is only possible if the outcome of* π_i *is not used as input of* π_{i+1} *(i.e. if* $a_{i+1,i+2} = 0$*).*

Proposition 4 (π-inverse-reduction). *Consider a matrix* A *with* $(a_{11}, a_{12}) = (1,0)$*. Consider the matrix* A' *obtained from* A *by swapping* (a_{21}, a_{31}, a_{41}) *and* (a_{23}, a_{33}, a_{43})*. Then, the compression functions* F_{A} *and* $\mathsf{F}_{\mathrm{A}'}$ *are equivalent. Intuitively, this reduction corresponds to replacing* π_1 *by* π_1^{-1}*. Using Prop. 1 and Prop. 3 on* $i = 1$*, the same reduction holds for* π_2*.*

Proof (Proof of Props. 1-4). Let F_{A} and $\mathsf{F}_{\mathrm{A}'}$ be two compression functions defined as in either of the propositions. For simplicity, in case of Prop. 2 we only consider $k = 2$ (so $a_{12} = 0$, $a_{22} = 1$ and $c_2 = 0$), for Prop. 3 we only consider $i = 1$ (so $a_{23} = 0$). By construction, the compression functions F_{A} and $\mathsf{F}_{\mathrm{A}'}$ satisfy the following properties:

$$\mathsf{F}_{\mathrm{A}}^{\pi_1,\pi_2,\pi_3}(x_1,x_2) = \begin{cases} \mathsf{F}_{\mathrm{A}'}^{\pi_1,\pi_2,\pi_3}(x_2,x_1) & \text{for Prop. 1,} \\ \mathsf{F}_{\mathrm{A}'}^{\pi_1,\pi_2,\pi_3}(x_1, x_2 \oplus c_0x_1 \oplus c_1\pi_1(a_{11}x_1)) & \text{for Prop. 2,} \\ \mathsf{F}_{\mathrm{A}'}^{\pi_2,\pi_1,\pi_3}(x_1,x_2) & \text{for Prop. 3,} \\ \mathsf{F}_{\mathrm{A}'}^{\pi_1^{-1},\pi_2,\pi_3}(\pi_1(x_1),x_2) & \text{for Prop. 4.} \end{cases} \tag{4}$$

We need to provide a bi-directional collision and preimage security reduction. For conciseness, we will provide only the collision security reduction; the case of preimage resistance is similar and is therefore omitted. Let $\mathcal{A}$ be a collision finding adversary for the compression function F_{A}, that on input of $\pi_1, \pi_2, \pi_3 \xleftarrow{\$} P_n$, outputs two tuples (x_1, x_2), (x_1', x_2') such that $\mathsf{F}_{\mathrm{A}}^{\pi_i}(x_1,x_2) = \mathsf{F}_{\mathrm{A}}^{\pi_i}(x_1',x_2')$. We construct a collision finding adversary $\mathcal{A}'$ for $\mathsf{F}_{\mathrm{A}'}$ that uses $\mathcal{A}$ as a subroutine and on input of $\pi_1', \pi_2', \pi_3' \xleftarrow{\$} P_n$ outputs a collision for $\mathsf{F}_{\mathrm{A}'}^{\pi_i'}$. Adversary $\mathcal{A}'$ operates as follows:

1. In Props. 1 and 2, the adversary $\mathcal{A}'$ sends $(\pi_1, \pi_2, \pi_3) \leftarrow (\pi'_1, \pi'_2, \pi'_3)$ to $\mathcal{A}$. In Prop. 3, the adversary $\mathcal{A}'$ sends $(\pi_1, \pi_2, \pi_3) \leftarrow (\pi'_2, \pi'_1, \pi'_3)$ to $\mathcal{A}$. In Prop. 4, the adversary $\mathcal{A}'$ sends $(\pi_1, \pi_2, \pi_3) \leftarrow ((\pi'_1)^{-1}, \pi'_2, \pi'_3)$ to $\mathcal{A}$;
2. $\mathcal{A}$ outputs two tuples (x_1, x_2), (x'_1, x'_2) such that $\mathsf{F}_{\mathrm{A}}^{\pi_i}(x_1, x_2) = \mathsf{F}_{\mathrm{A}}^{\pi_i}(x'_1, x'_2)$;
3. In Prop. 1, $\mathcal{A}'$ outputs collision (x_2, x_1) and (x'_2, x'_1). In Prop. 2, $\mathcal{A}'$ outputs $(x_1, x_2 \oplus c_0 x_1 \oplus c_1 \pi_1(a_{11} x_1))$ and $(x'_1, x'_2 \oplus c_0 x'_1 \oplus c_1 \pi_1(a_{11} x'_1))$. In Prop. 3, $\mathcal{A}'$ outputs (x_1, x_2) and (x'_1, x'_2). In Prop. 4, $\mathcal{A}'$ outputs $((\pi'_1)^{-1}(x_1), x_2)$ and $((\pi'_1)^{-1}(x'_1), x'_2)$.

Notice that in step one, (π_1, π_2, π_3) are clearly randomly and independently distributed as (π'_1, π'_2, π'_3) are, and therefore $\mathcal{A}$ can output (x_1, x_2), (x'_1, x'_2) such that $\mathsf{F}_{\mathrm{A}}^{\pi_1,\pi_2,\pi_3}(x_1, x_2) = \mathsf{F}_{\mathrm{A}}^{\pi_1,\pi_2,\pi_3}(x'_1, x'_2)$ with probability $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}}}(\mathcal{A})$. For $\mathsf{F}_{\mathrm{A}'}$ of Prop. 3, these tuples indeed render a collision as given in step 3:

$$\begin{aligned}
\mathsf{F}_{\mathrm{A}'}^{\pi'_1,\pi'_2,\pi'_3}(x_1, x_2) &= \mathsf{F}_{\mathrm{A}}^{\pi'_2,\pi'_1,\pi'_3}(x_1, x_2) && \text{by (4)},\\
&= \mathsf{F}_{\mathrm{A}}^{\pi'_2,\pi'_1,\pi'_3}(x'_1, x'_2) && \text{by collision for } \mathsf{F}_{\mathrm{A}},\\
&= \mathsf{F}_{\mathrm{A}'}^{\pi'_1,\pi'_2,\pi'_3}(x'_1, x'_2) && \text{by (4)}.
\end{aligned}$$

The same argument applies to the other propositions. In any case, $\mathcal{A}'$ needs at most four queries more than $\mathcal{A}$, and thus we obtain $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}}}(q) \leq \mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}'}}(q+4)$. The reductions in the other direction (from $\mathsf{F}_{\mathrm{A}'}$ to F_{A}) are identical due to symmetry. □

Except for Prop. 4, the reductions also hold in the single-permutation setting. We remark that these reductions are not only restricted to binary matrices, but apply to general matrices A. In particular, the independence criterion of [10] can be derived using the given reductions. Also, we note that the reductions can easily be represented by linear matrix operations.

4 Main Result for Multi-Permutation Setting

We classify the set of permutation-based compression functions of the form (3) that achieve optimal collision resistance. Theorem 1 shows that the set of (asymptotically) secure functions is fully covered by four equivalence classes; for any other compression function collisions can be found faster than the birthday bound. One of these four classes – defined by $\mathsf{F}_{\mathrm{A}_2}$ below – provides optimal (asymptotic) $2^{2n/3}$ preimage security, for the other three classes preimages can be found significantly faster.

Theorem 1. *Consider the multi-permutation setting. Let* F_{A} *be any compression function defined by a binary matrix* A *of the form (2). Let* $\mathsf{F}_{\mathrm{A}_k}$ *for* $k =$

$1, 2, 3, 4$ *be the compression functions defined by matrices*

$$\mathrm{A}_1 = \left(\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 1 \end{array}\right), \qquad \mathrm{A}_2 = \left(\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 1 & 1 \end{array}\right),$$
$$\mathrm{A}_3 = \left(\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 & 1 \end{array}\right), \qquad \mathrm{A}_4 = \left(\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 0 & 1 \end{array}\right). \tag{5}$$

Let $\varepsilon > 0$.

(i) If F_{A} *is equivalent to* $\mathsf{F}_{\mathrm{A}_k}$ *for* $k \in \{1, 2, 3, 4\}$*, it satisfies* $\lim_{n\to\infty} \mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}}}(2^{n/2(1-\varepsilon)}) = 0$*. Otherwise, it satisfies* $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}}}(q) = \Omega(q^5/2^{2n})$*;*

(ii) If F_{A} *is equivalent to* $\mathsf{F}_{\mathrm{A}_2}$*, it satisfies* $\lim_{n\to\infty} \mathbf{Adv}^{\mathsf{epre}}_{\mathsf{F}_{\mathrm{A}}}(2^{2n/3(1-\varepsilon)}) = 0$*;*

(iii) If F_{A} *is equivalent to* $\mathsf{F}_{\mathrm{A}_k}$ *for* $k \in \{1, 3, 4\}$*, it satisfies* $\mathbf{Adv}^{\mathsf{epre}}_{\mathsf{F}_{\mathrm{A}}}(q) = \Theta(q^2/2^n)$.

In other words, a compression function offers optimal collision resistance *if and only if* it is equivalent to either of $\mathsf{F}_{\mathrm{A}_1}, \mathsf{F}_{\mathrm{A}_2}, \mathsf{F}_{\mathrm{A}_3}, \mathsf{F}_{\mathrm{A}_4}$, and additionally achieves optimal preimage resistance (with respect to the bounds of [11]) *if and only if* it is equivalent to $\mathsf{F}_{\mathrm{A}_2}$.

In order to prove Thm. 1, more specifically part (i) for $k = 1, 2, 4$ and part (ii), we pose the following conjecture. This conjecture relates to the area of extremal graph theory and is of independent interest. In particular, it can be shown to be similar to (but more complex than) a longstanding problem of Zarankiewicz from 1951 [2, Ch. 6.2].

Conjecture 1. Let $q \leq 2^n$, and let Z be a set of q elements taken uniformly at random from $\{0,1\}^n$. Let β denote the maximum number of tuples $(x_1, x_2, z) \in X_1 \times X_2 \times Z$ such that $x_1 \oplus x_2 = z$, where X_1, X_2 are any two subsets of $\{0,1\}^n$ of size q. Formally:

$$\beta := \max_{\substack{X_1, X_2 \subseteq \{0,1\}^n \\ |X_1| = |X_2| = q}} \left| \{(x_1, x_2, z) \in X_1 \times X_2 \times Z \;\mid\; x_1 \oplus x_2 = z\} \right|. \tag{6}$$

There exists a constant d_1 such that $\mathbf{Pr}\left(\beta > d_1 q \log q\right) \to 0$ for $n \to \infty$ and $q < 2^{n/2}$. Similarly, there exists a constant d_2 such that $\mathbf{Pr}\left(\beta > d_2 q^{3/2}\right) \to 0$ for $n \to \infty$ and $q < 2^{2n/3}$.

The first bound is used in the proof Thm. 1(i) for $k = 1, 2, 4$, and the second bound in the proof Thm. 1(ii). A detailed heuristic for Conj. 1 is given in [6], together with a comparison with Zarankiewicz's conjecture, but we leave a full proof of Conj. 1 as an open problem.

4.1 Proof of Theorem 1

The proof of Thm. 1 is structured as follows. Firstly, in Lem. 1 we show that any compression function F_{A} can be reduced either to an invalid compression function or to a compression function $\mathsf{F}_{\mathrm{A}'}$ defined by a matrix A' with first two rows $10000, 01000$. By construction (see Sect. 3), the security properties of one compression function are valid for the whole equivalence class. Secondly, in Lem. 2 several collision attacks are described that invalidate the security of each of the remaining compression functions, except for the classes defined by $\mathsf{F}_{\mathrm{A}_k}$ ($k \in \{1,2,3,4\}$) for A_k as in (5). Thirdly, the collision and preimage resistance of the remaining four compression functions are analyzed in Lem. 3, which completes the proof of Thm. 1.

Lemma 1. *Any compression function* F_{A}*, for valid* A*, is equivalent to a compression function* $\mathsf{F}_{\mathrm{A}'}$*, where either* A' *is invalid or the first two rows of* A' *equal* $10000, 01000$*.*

Proof. The proof is constructive. Several reductions are used, but for ease of notation apostrophes are omitted. Let F_{A} be a compression function defined by some valid matrix A. As A is valid, we have $a_{11} + a_{12} \geq 1$. If $a_{11} + a_{12} = 2$, we can apply Prop. 2 on $c_0 = 1$ to obtain $a_{11} + a_{12} = 1$. Now, by Prop. 1 we can assume that $(a_{11}, a_{12}) = (1, 0)$.

Considering the second row of A, we distinguish between $a_{22} = 1$ and $a_{22} = 0$. In the former case, a XOR-reduction (Prop. 2) on $(c_0, c_1) = (a_{21}, a_{23})$ reduces the scheme to the required form. In the latter case, where $a_{22} = 0$, we proceed as follows. If $a_{32} = 0$, A is equivalent to an invalid matrix. Otherwise, by applying Prop. 2 with $(c_0, c_1, c_2) = (a_{31}, a_{33}, a_{34})$ we obtain that F_{A} is equivalent to a compression function $\mathsf{F}_{\mathrm{A}'}$, for some matrix A' with rows $(10000, a'_{21}0a'_{23}00, 01000, a'_{41}a'_{42}a'_{43}a'_{44}a'_{45})$. The result is now obtained by swapping π_2 and π_3 (Prop. 3 for $i = 2$). □

As a direct consequence of Lem. 1, it suffices to consider compression functions F_{A}, where

$$\mathrm{A} = \left(\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 \\ \hline a_{41} & a_{42} & a_{43} & a_{44} & 1 \end{array}\right) \tag{7}$$

for some binary values $a_{31}, \ldots, a_{44}$. Notice that $a_{45} = 1$ because of the validity of the matrix. We describe a couple of collision attacks that apply to compression functions of this form. We note that similar results also hold for preimage resistance.

Lemma 2. *Let* F_{A} *be a compression function defined by a valid matrix* A *of the form (7).*

(i) If A *satisfies* $(a_{31} + a_{33})(a_{32} + a_{34}) = 0$*, then* $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}}}(q) = \Omega(q^4/2^n)$*;*
(ii) If A *satisfies* $\bigvee_{j=1}^{4} a_{3j} = a_{4j} = 0$*, then* $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}}}(q) = \Omega(q^3/2^n)$*;*

(iii) If A *satisfies* $\bigwedge_{j=1}^{2} a_{3j}a_{4,j+2} \neq a_{3,j+2}a_{4j}$*, then* $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_\mathrm{A}}(q) = \Omega(q^3/2^n)$*;*
(iv) If A *satisfies* $a_{41} + a_{42} + a_{43} + a_{44} = 1$*, then* $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_\mathrm{A}}(q) = \Omega(q^5/2^{2n})$*.*

For clarity, the proofs of results (i), (ii), (iii) and (iv) will be given separately.

Proof (Proof of Lem. 2(i)). Without loss of generality, we assume $a_{32} + a_{34} = 0$, i.e. $a_{32} = a_{34} = 0$. Hence, we consider matrices A with $\left(\begin{smallmatrix} a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{smallmatrix}\right) = \left(\begin{smallmatrix} a_{31} & 0 & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{smallmatrix}\right)$, where $a_{31} + a_{33} \geq 1$, by validity of A. This matrix defines the compression function:

$$\mathsf{F}_\mathrm{A}(x_1, x_2) = a_{41}x_1 \oplus a_{42}x_2 \oplus a_{43}\pi_1(x_1) \oplus \pi_2(x_2) \oplus \pi_3(a_{31}x_1 \oplus a_{33}\pi_1(x_1))\,.$$

Define the functions $f_1(x) = a_{41}x \oplus a_{43}\pi_1(x) \oplus \pi_3(a_{31}x \oplus a_{33}\pi_1(x))$ and $f_2(x) = a_{42}x \oplus \pi_2(x)$. Notice that $\mathsf{F}_\mathrm{A}(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$. A collision-finding adversary $\mathcal{A}$ for F_A proceeds as follows. He sets up two lists of q random elements $X_1 := \{x_1^{(1)}, \ldots, x_1^{(q)}\}$ and $X_2 := \{x_2^{(1)}, \ldots, x_2^{(q)}\}$, and computes the corresponding values $f_1(x_1^{(k)})$ and $f_2(x_2^{(k)})$ (for $k = 1, \ldots, q$). Thus, in total $\mathcal{A}$ makes q queries to each of his random oracles. Given one of the $\binom{q}{2}^2$ combinations $x_1, x_1' \in X_1$, $x_2, x_2' \in X_2$, this combination yields a collision for F_A with probability $\Theta(2^{-n})$. Concluding, $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_\mathrm{A}}(q) = \Omega(q^4/2^n)$. □

Proof (Proof of Lem. 2(ii)). For the cases $j \in \{3, 4\}$ as explained in Sect. 2.3 (these cases are in fact redundant due to the validity of A), collisions can be found in at most $2^{n/3}$ queries due to Stam's bound [13,14]. We consider a matrix A with $a_{32} = a_{42} = 0$ (the case $j = 2$), a similar analysis holds for $j = 1$. Note that F_A satisfies $\mathsf{F}_\mathrm{A}(x_1, x_2) = \mathsf{F}_{\mathrm{A}'}(x_1, \pi_2(x_2))$, where A′ has third and fourth rows $(a_{31}a_{34}a_{33}00, a_{41}a_{44}a_{43}01)$. The compression function $\mathsf{F}_{\mathrm{A}'}$ satisfies the condition of this lemma for $j = 4$, and invertibility of π_2 guarantees a collision for F_A in the same amount of queries plus 2. We note that the result also follows from Prop. 4, but as we will use Lem. 2(ii) in the single-permutation setting as well, we here consider a more robust reduction. □

Proof (Proof of Lem. 2(iii)). The idea of the attack is to focus on collisions $(x_1, x_2) \neq (x_1', x_2')$ for which the input to the third permutation π_3 is the same. We first consider the case of matrices A with $\left(\begin{smallmatrix} a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{smallmatrix}\right) = \left(\begin{smallmatrix} 1 & 1 & 0 & 0 \\ a_{41} & a_{42} & 1 & 1 \end{smallmatrix}\right)$, the general case is discussed afterwards. The matrix defines compression function

$$\mathsf{F}_\mathrm{A}(x_1, x_2) = a_{41}x_1 \oplus a_{42}x_2 \oplus \pi_1(x_1) \oplus \pi_2(x_2) \oplus \pi_3(x_1 \oplus x_2)\,.$$

We construct an adversary $\mathcal{A}$ that aims at finding a collision $(x_1, x_2) \neq (x_1', x_2')$ such that

$$x_1 \oplus x_2 = x_1' \oplus x_2'\,, \tag{8a}$$

$$a_{41}x_1 \oplus a_{42}x_2 \oplus \pi_1(x_1) \oplus \pi_2(x_2) = a_{41}x_1' \oplus a_{42}x_2' \oplus \pi_1(x_1') \oplus \pi_2(x_2')\,. \tag{8b}$$

The adversary sets up two lists of $q = 2^\alpha$ elements $X_1 := \{x_1^{(1)}, \ldots, x_1^{(q)}\}$ and $X_2 := \{x_2^{(1)}, \ldots, x_2^{(q)}\}$, where $x_1^{(k)} = x_2^{(k)} = 0^{n-\alpha} \| \langle k-1 \rangle_\alpha$ for $k = 1, \ldots, q$.

He computes the corresponding values $\pi_1(x_1^{(k)})$ and $\pi_2(x_2^{(k)})$ (for $k = 1, \ldots, q$). Fix any x_1, x_2, x_1' such that $x_1 \neq x_1'$. Then, there is exactly one x_2' such that (8a) is satisfied. For any of these $q\binom{q}{2}$ options, (8b) is satisfied with probability $\Theta(2^{-n})$. For any of such succeeding tuples, the adversary additionally queries $\pi_3(x_1 \oplus x_2) = \pi_3(x_1' \oplus x_2')$ in order to get a collision. Concluding, $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F_A}}(q) = \Omega(q^3/2^n)$.

The described attack relies on the key property that the set of equations

$$\begin{pmatrix} a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} (x_1 \oplus x_1',\ x_2 \oplus x_2',\ \pi_1(x_1) \oplus \pi_1(x_1'),\ \pi_2(x_2) \oplus \pi_2(x_2'))^\top = 0$$

contains an equation in which x_1, x_2, x_1', x_2' occur exactly once. By the requirement of A, $\left(\begin{smallmatrix} a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{smallmatrix}\right)$ contains at least two zeroes. If two zeroes are located in the same row, this key property is satisfied and the attack succeeds. On the other hand, if both rows contain exactly one zero, one can XOR the first equation to the second one to return to the first case. □

Proof (Proof of Lem. 2(iv)). Without loss of generality, we assume $a_{41} = 1$. By Lem. 2(ii), we can consider $a_{32} = a_{33} = a_{34} = 1$. The matrix defines compression function

$$\mathsf{F_A}(x_1, x_2) = x_1 \oplus \pi_3(a_{31}x_1 \oplus x_2 \oplus \pi_1(x_1) \oplus \pi_2(x_2))\,.$$

We construct a collision adversary $\mathcal{A}$ for $\mathsf{F_A}$. The adversary sets up a list of $q = 2^\alpha$ random elements $X_2 := \{x_2^{(1)}, \ldots, x_2^{(q)}\}$, and computes the corresponding values $y_2^{(k)} = \pi_2(x_2^{(k)})$ (for $k = 1, \ldots, q$). Additionally, the adversary sets up two lists $X_1 := \{x_1^{(1)}, \ldots, x_1^{(q)}\}$ and $Y_3 := \{y_3^{(1)}, \ldots, y_3^{(q)}\}$, where $x_1^{(k)} = y_3^{(k)} = 0^{n-\alpha} \| \langle k-1 \rangle_\alpha$ for $k = 1, \ldots, q$. He computes the corresponding values $y_1^{(k)} = \pi_1(x_1^{(k)})$ and $x_3^{(k)} = \pi_3^{-1}(y_3^{(k)})$ (for $k = 1, \ldots, q$). Fix any x_1, y_3, x_1' such that $x_1 \neq x_1'$. Then, there is exactly one y_3' such that $x_1 \oplus y_3 = x_1' \oplus y_3'$. The adversary obtains a collision for $\mathsf{F_A}$ if X_2 contains two elements x_2, x_2' such that $x_2 \oplus y_2 = a_{31}x_1 \oplus y_1 \oplus x_3$ and $x_2' \oplus y_2' = a_{31}x_1' \oplus y_1' \oplus x_3'$. Two such x_2, x_2' exist with probability $\Omega(\binom{q}{2}/2^{2n})$. As the adversary needs to succeed for only one of the $q\binom{q}{2}$ choices of x_1, y_3, x_1', he finds a collision for $\mathsf{F_A}$ with probability $\Omega(q^5/2^{2n})$. □

Next, the compression functions evolved from Lem. 1 are analyzed with respect to the attacks of Lem. 2. Before proceeding, we remark that for the multi-permutation setting, the following reductions apply to the compression function classes evolved from Lem. 1. We refer to these reductions as the "M- and N-reduction".

M-reduction: Applying Prop. 1, and Prop. 3 on $i = 1$ corresponds to mutually swapping $\left(\begin{smallmatrix} a_{31} \\ a_{41} \end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix} a_{32} \\ a_{42} \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} a_{33} \\ a_{43} \end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix} a_{34} \\ a_{44} \end{smallmatrix}\right)$;

N-reduction: Prop. 4 reduces to swapping $\left(\begin{smallmatrix} a_{3j} \\ a_{4j} \end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix} a_{3,j+2} \\ a_{4,j+2} \end{smallmatrix}\right)$ for $j \in \{1, 2\}$.

We now continue evaluating the matrices A of the form (7), and consider the different values of $\|\mathbf{a}_{3,*}\|$.

$\|\mathbf{a}_{3,*}\| = \mathbf{0}$. The matrix is invalid and excluded by definition;
$\|\mathbf{a}_{3,*}\| = \mathbf{1}$. The matrix is vulnerable to the attack of Lem. 2(i);
$\|\mathbf{a}_{3,*}\| = \mathbf{2}$. The matrix contradicts either one of the requirements of Lem. 2. Technically, if $(a_{31} + a_{33})(a_{32} + a_{34}) = 0$ it violates Lem. 2(i), and otherwise the values $a_{41}, \ldots, a_{44}$ will violate either the requirement of Lem. 2(ii) or of Lem. 2(iii);
$\|\mathbf{a}_{3,*}\| = \mathbf{3}$. Due to M- and N-reductions, it suffices to consider $a_{31}a_{32}a_{33}a_{34} = 1110$, and consequently $a_{44} = 1$ by Lem. 2(ii). Lemma 2(iii) now states that we require $a_{41} = a_{43}$, which gives the following four options for $a_{41}a_{42}a_{43}$: 000, 010, 101 and 111. The first one is vulnerable to the attack of Lem. 2(iv), and the fourth matrix is equivalent to the second (by consequently applying Prop. 2 on $(c_0, c_1) = (1, 1)$, and Prop. 3 for $i = 2$). We are left with A_1 and A_2 of (5);
$\|\mathbf{a}_{3,*}\| = \mathbf{4}$. Due to M- and N-reductions, it suffices to consider $a_{41}a_{42}a_{43}a_{44} \in \{0000, 1000, 1010, 1100, 1110, 1111\}$. The cases 1000 and 1100 are vulnerable to the attacks of Lems. 2(iv) and 2(iii), respectively. For the cases 0000 and 1111, finding collisions is as hard as finding collisions for $\mathsf{F}(x_1, x_2) = x_1 \oplus x_2 \oplus \pi_1(x_1) \oplus \pi_2(x_2)$ (for which collisions are found in at most $2^{n/3}$ queries, due to Stam's bound [13,14]). We are left with A_3 and A_4 of (5).

It remains to analyze collision and preimage security of the four compression functions defined by the matrices of (5). This is covered by following lemma, which is proven in [6]. Particularly, Lem. 3 completes the proof of Thm. 1.

Lemma 3. *Let $\varepsilon > 0$. Then:*

(i) $\lim_{n\to\infty} \mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}_k}}(2^{n/2(1-\varepsilon)}) = 0$ *for* $k = 1, 2, 3, 4$*;*
(ii) $\lim_{n\to\infty} \mathbf{Adv}^{\mathsf{epre}}_{\mathsf{F}_{\mathrm{A}_2}}(2^{2n/3(1-\varepsilon)}) = 0$*, and* $\mathbf{Adv}^{\mathsf{epre}}_{\mathsf{F}_{\mathrm{A}_k}}(q) = \Theta(q^2/2^n)$ *for* $k = 1, 3, 4$.

5 Main Result for Single-Permutation Setting

In a similar fashion as in Sect. 4, we analyze the security of compression functions based on three calls to the same permutations, the single-permutation setting. It turns out that *there does not exist any* compression function of the form (3) that achieves optimal collision resistance. We note that this result does not rely on Conj. 1. In App. A we show how the results of this section can be generalized to cover any single-permutation compression function where additional affine transformations on the permutation inputs are taken into account.

Theorem 2. *Consider the single-permutation setting, where $\pi_1 = \pi_2 = \pi_3 =: \pi$. Any compression function F_{A} defined by a binary matrix* A *of the form (2) satisfies* $\mathbf{Adv}^{\mathsf{col}}_{\mathsf{F}_{\mathrm{A}}}(q) = \Omega(q^5/2^{2n})$.

Proof. The proof of Thm. 2 is similar to the proof of Thm. 1, and we highlight the differences. Lemmas 1 and 2 still apply, and additionally the M-reduction also holds in the single-permutation setting. Notice that the N-reduction *does not hold* as it incorporates Prop. 4. Similar to before, we will evaluate the matrices A of the form (7). The case $\|\mathbf{a}_{3,*}\| \leq 2$ is the same as before.

$\|\mathbf{a}_{3,*}\| = 3$. Due to M-reductions, it suffices to consider $a_{31}a_{32}a_{33}a_{34} \in \{1110, 0111\}$.

- $a_{31}a_{32}a_{33}a_{34} = 1110$. The same analysis as in Sect. 4.1 applies, leaving the matrices A_1 and A_2 of (5). In the single-permutation setting, the two corresponding compression functions satisfy $\mathsf{F}_{\mathrm{A}_1}(x_1, \pi(x_1)) = \pi^2(x_1)$ and $\mathsf{F}_{\mathrm{A}_2}(x_1, x_2) = \mathsf{F}_{\mathrm{A}_2}(x_1, x_1 \oplus x_2 \oplus \pi(x_1))$ for any x_1, x_2. Collisions can thus be trivially found;
- $a_{31}a_{32}a_{33}a_{34} = 0111$. By Lem. 2(ii), we have $a_{41} = 1$. Lemma 2(iii) now states that we require $a_{42} = a_{44}$, which gives the following four options for $a_{42}a_{43}a_{44}$: 000, 010, 101 and 111. The first one is vulnerable to the attack of Lem. 2(iv), the second, third and fourth matrix satisfy $\mathsf{F}_{\mathrm{A}}(x_1, x_1) = x_1$, $\mathsf{F}_{\mathrm{A}}(x_1, x_1) = 0$ and $\mathsf{F}_{\mathrm{A}}(x_1, x_1) = \pi(x_1)$, respectively, for any x_1. Collisions can thus be trivially found;

$\|\mathbf{a}_{3,*}\| = 4$. Except for $a_{41}a_{42}a_{43}a_{44} \in \{1010, 1001, 0110, 0101\}$, all induced compression functions satisfy $\mathsf{F}_{\mathrm{A}}(x_1, x_1) \oplus \pi(0) \in \{0, x_1, \pi(x_1)\}$ for any x_1, for which collisions can be trivially found. The cases $1001, 0110$ are vulnerable to Lem. 2(iii). The remaining two cases, which are equivalent by M-reduction, allow for trivial collisions as well: the compression function induced by $(a_{41}a_{42}a_{43}a_{44}) = (1010)$ satisfies $\mathsf{F}_{\mathrm{A}}(x_1, \pi^{-1}(x_1 \oplus \pi(x_1))) = 0$ for any x_1 (cf. [10]).

Hence, the analyzed compression functions either allow for trivial collision or are vulnerable to Lem. 2, therewith allowing for collisions in at most $2^{2n/5}$ queries. □

Concluding, for any compression function F_{A} of the form (3), where the three permutations are equal to one single permutation π, collisions can be found in at most $2^{2n/5}$ queries, hence considerably faster than in $2^{n/2}$ queries.

6 Conclusions

We provided a full security classification of $2n$-to-n-bit compression functions that are solely built of XOR-operators and of three permutations. Therewith, we have analyzed compression functions that are not included in the analysis of Rogaway and Steinberger [10], but yet are interesting because of their elegance (they only employ XOR-operators) and efficiency (XOR-operators are slightly cheaper than finite field multiplications by constants). For any of the 2^{15} compression functions of the described form, we either provide a formal collision and preimage security proof or a collision attack more efficient than the birthday bound.

For the multi-permutation setting, where the three permutations are different, there are exactly four equivalence classes of functions that allow for optimal collision resistance, one class of which the compression functions achieve optimal preimage resistance w.r.t. the bounds of [11]. A summary of these results is given in Table 1. Regarding the absolute number of collision/preimage secure compression functions, by ways of an extensive computation one finds 96 functions equivalent to $\mathsf{F}_{\mathrm{A}_1}$ (including the $\mathsf{F}_{\mathrm{A}_1}$ itself), 48 functions in each of the classes defined by $\mathsf{F}_{\mathrm{A}_2}$ and $\mathsf{F}_{\mathrm{A}_4}$, and 24 functions equivalent to $\mathsf{F}_{\mathrm{A}_3}$. In total, we have thus proven 216 compression functions optimally collision secure, 48 of which we have proven optimally preimage secure. A small part of the results for the multi-permutation setting relies on an extremal graph theory based conjecture, Conj. 1, which we supported by an extensive and detailed heuristic. We leave the full analysis of Conj. 1 as an open problem.

For the single-permutation setting, where the three permutations are the same, we show that it is not possible to construct a $2n$-to-n-bit compression function that achieves optimal collision resistance. In light of the amount of optimally secure compression functions we have found in the multi-permutation setting, this observation is not as expected. This negative result casts doubts over the existence of any (larger) permutation-based XOR-based compression function built on (multiple invocations of) one single permutation. We leave this question as an open problem.

The results in this work are derived in the permutation setting. Different results may be obtained if we consider three underlying primitives to be one-way functions: in particular, the π-inverse-reduction (Prop. 4) and Lem. 2 rely on the invertibility of these primitives. Further research questions include the applicability of the approach followed in this work to different classes of compression functions, for instance with larger domain and range, with more permutations or random functions instead, or defined over different fields.

Acknowledgments. This work has been funded in part by the IAP Program P6/26 BCRYPT of the Belgian State (Belgian Science Policy), in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II, and in part by the Research Council K.U.Leuven: GOA TENSE. The first author is supported by a Ph.D. Fellowship from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

1. Black, J., Cochran, M., Shrimpton, T.: On the impossibility of highly-efficient blockcipher-based hash functions. In: Advances in Cryptology - EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 526–541. Springer-Verlag, Berlin (2005)
2. Bollobás, B.: Extremal Graph Theory. Academic Press (1978)
3. Hirose, S.: Some plausible constructions of double-block-length hash functions. In: Fast Software Encryption '06. Lecture Notes in Computer Science, vol. 4047, pp. 210–225. Springer-Verlag, Berlin (2006)

4. Lai, X., Massey, J.: Hash function based on block ciphers. In: Advances in Cryptology - EUROCRYPT '92. Lecture Notes in Computer Science, vol. 658, pp. 55–70. Springer-Verlag, Berlin (1992)
5. Lee, J., Kwon, D.: Security of single-permutation-based compression functions. Cryptology ePrint Archive, Report 2009/145 (2009)
6. Mennink, B., Preneel, B.: Hash functions based on three permutations: A generic security analysis. Cryptology ePrint Archive, Report 2011/532 (2011), full version of this paper
7. Preneel, B., Govaerts, R., Vandewalle, J.: Hash functions based on block ciphers: A synthetic approach. In: Advances in Cryptology - CRYPTO '93. Lecture Notes in Computer Science, vol. 773, pp. 368–378. Springer-Verlag, Berlin (1993)
8. Rabin, M.: Digitalized signatures. In: Foundations of Secure Computation '78. pp. 155–166. Academic Press, New York (1978)
9. Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: Fast Software Encryption 2004. Lecture Notes in Computer Science, vol. 3017, pp. 371–388. Springer-Verlag, Berlin (2004)
10. Rogaway, P., Steinberger, J.: Constructing cryptographic hash functions from fixed-key blockciphers. In: Advances in Cryptology - CRYPTO 2008. Lecture Notes in Computer Science, vol. 5157, pp. 433–450. Springer-Verlag, Berlin (2008)
11. Rogaway, P., Steinberger, J.: Security/efficiency tradeoffs for permutation-based hashing. In: Advances in Cryptology - EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 220–236. Springer-Verlag, Berlin (2008)
12. Shrimpton, T., Stam, M.: Building a collision-resistant compression function from non-compressing primitives. In: International Colloquium on Automata, Languages and Programming - ICALP (2) 2008. Lecture Notes in Computer Science, vol. 5126, pp. 643–654. Springer-Verlag, Berlin (2008)
13. Stam, M.: Beyond uniformity: Better security/efficiency tradeoffs for compression functions. In: Advances in Cryptology - CRYPTO 2008. Lecture Notes in Computer Science, vol. 5157, pp. 397–412. Springer-Verlag, Berlin (2008)
14. Steinberger, J.: Stam's collision resistance conjecture. In: Advances in Cryptology - EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 597–615. Springer-Verlag, Berlin (2010)

A Generalization of Theorem 2

We generalize our findings on the single-permutation setting to cover *any* function, where affine transformations on the inputs to the permutations are taken into account. This generalization is straightforward, but technical and more elaborate. For a matrix $\mathrm{B} = (b_1,\ b_2,\ b_3,\ b_4)^\top$ with elements in $\{0,1\}^n$, we define the compression function F_{AB} as follows:

$$\begin{aligned}\mathsf{F}_{\mathrm{AB}}(x_1, x_2) = z\,,\ \text{where}\ \ & y_1 \leftarrow \pi_1(a_{11}x_1 \oplus a_{12}x_2 \oplus b_1)\,,\\ & y_2 \leftarrow \pi_2(a_{21}x_1 \oplus a_{22}x_2 \oplus a_{23}y_1 \oplus b_2)\,,\\ & y_3 \leftarrow \pi_3(a_{31}x_1 \oplus a_{32}x_2 \oplus a_{33}y_1 \oplus a_{34}y_2 \oplus b_3)\,,\\ & z \leftarrow a_{41}x_1 \oplus a_{42}x_2 \oplus a_{43}y_1 \oplus a_{44}y_2 \oplus a_{45}y_3 \oplus b_4\,.\end{aligned} \tag{9}$$

where A is as in Sect. 2.1. We note that for the multi-permutation setting, this generalization is of no added value, as the permutations are independently distributed anyway. Adding constants is, however, a customary approach to obtain "different" permutations from a single one (e.g. $\pi_i(x) = \pi(b_i \oplus x)$ for $i = 1, 2, 3$), but as we will show, the findings of Thm. 2 also apply to this extended setting.

We reformulate Props. 1-3 to the case of F_{AB} (recall that Prop. 4 did not apply to the single-permutation setting in the first place). Propositions 1 and 2 apply to any F_{AB} and $\mathsf{F}_{\mathrm{A'B'}}$ with $\mathrm{B} = \mathrm{B}'$ and Prop. 3 holds for any B and B′ with $(b'_i, b'_{i+1}) = (b_{i+1}, b_i)$. Given this, the proof of Thm. 2 almost carries over. Lemmas 1 and 2 apply with straightforward generalization. It remains to evaluate the matrices A of the form (7) *for any* $\mathrm{B} \in (\{0,1\}^n)^{4\times 1}$. The case $\|\mathbf{a}_{3,*}\| \leq 2$ is the same as in the proof of Thm. 2.

$\|\mathbf{a}_{3,*}\| = 3$. Due to M-reductions, it suffices to consider $a_{31}a_{32}a_{33}a_{34} \in \{1110, 0111\}$.

- $a_{31}a_{32}a_{33}a_{34} = 1110$. The same analysis as in Sect. 4.1 applies, leaving the matrices A_1 and A_2 of (5). In the extended single-permutation setting, the two corresponding compression functions satisfy $\mathsf{F}_{\mathrm{A}_1\mathrm{B}}(x_1 \oplus b_1, \pi(x_1) \oplus b_1 \oplus b_3) = \pi(\pi(x_1) \oplus b_1 \oplus b_2 \oplus b_3) \oplus b_1 \oplus b_3 \oplus b_4$ and $\mathsf{F}_{\mathrm{A}_2\mathrm{B}}(x_1, x_2) = \mathsf{F}_{\mathrm{A}_2\mathrm{B}}(x_1, x_1 \oplus x_2 \oplus \pi(x_1 \oplus b_1) \oplus b_2 \oplus b_3)$ for any x_1, x_2. Collisions can thus be trivially found;
- $a_{31}a_{32}a_{33}a_{34} = 0111$. By Lem. 2(ii), we have $a_{41} = 1$. Lemma 2(iii) now states that we require $a_{42} = a_{44}$, which gives the following four options for $a_{42}a_{43}a_{44}$: 000, 010, 101 and 111. The first one is vulnerable to the attack of Lem. 2(iv), the second, third and fourth matrix satisfy $\mathsf{F}_{\mathrm{AB}}(x_1, \pi^{-1}(\pi(x_1 \oplus b_1) \oplus b_2 \oplus b_3) \oplus b_2) = x_1 \oplus b_2 \oplus b_3 \oplus b_4$, $\mathsf{F}_{\mathrm{AB}}(x_1 \oplus b_1, x_1 \oplus b_2) = \mathsf{F}_{\mathrm{AB}}(x_1 \oplus b_1 \oplus b_2 \oplus b_3, x_1 \oplus b_3)$ and $\mathsf{F}_{\mathrm{AB}}(x_1 \oplus b_1, x_1 \oplus b_2) = \pi(x_1 \oplus b_2 \oplus b_3) \oplus b_1 \oplus b_2 \oplus b_4$, respectively, for any x_1. Collisions can thus be trivially found;

$\|\mathbf{a}_{3,*}\| = 4$. Except for $a_{41}a_{42}a_{43}a_{44} \in \{1010, 1001, 0110, 0101\}$, all induced compression functions satisfy $\mathsf{F}_{\mathrm{AB}}(x_1 \oplus b_1, x_1 \oplus b_2) \oplus \pi(b_1 \oplus b_2 \oplus b_3) \oplus a_{41}b_1 \oplus a_{42}b_2 \oplus b_4 \in \{0, x_1, \pi(x_1)\}$ for any x_1, for which collisions can be trivially found. The cases $1001, 0110$ are vulnerable to Lem. 2(iii). The remaining two cases, which are equivalent by M-reduction, allow for trivial collisions as well: the compression function induced by $(a_{41}a_{42}a_{43}a_{44}) = (1010)$ satisfies $\mathsf{F}_{\mathrm{AB}}(x_1, \pi^{-1}(x_1 \oplus \pi(x_1 \oplus b_1) \oplus b_2 \oplus b_3) \oplus b_2) = b_2 \oplus b_3 \oplus b_4$ for any x_1.

Hence, any of the analyzed compression functions either allows for trivial collision or is vulnerable to Lem. 2, therewith allowing for collisions in at most $2^{2n/5}$ queries.

Concluding, for any compression function F_{AB} of the generalized form (9), collisions can be found in at most $2^{2n/5}$ queries, hence considerably faster than in $2^{n/2}$ queries.