Security Analysis of Constructions Combining FIL Random Oracles

Yannick Seurin and Thomas Peyrin

France Telecom R&D, 38-40 rue du Général Leclerc, F-92794 Issy-les-Moulineaux, France
Université de Versailles, 45 avenue des Etats-Unis, F-78035 Versailles, France
yannick.seurin@m4x.org, thomas.peyrin@orange-ftgroup.com

Abstract. We consider the security of compression functions built by combining smaller perfectly secure compression functions modeled as fixed input length random oracles. We give tight security bounds and generic attacks for various parameters of these constructions and apply our results to recent proposals of block cipher-based hash functions.

Key words: block ciphers, compression functions, hash functions, provable security, random oracle.

1 Introduction

Cryptographic hash functions are fundamental primitives in information security [17] used in a variety of applications such as message integrity, authentication schemes or digital signatures. Mathematically speaking, it is a function from $\{0,1\}^*$, the set of all finite length bit strings, to $\{0,1\}^l$ where l is the fixed size of the hash value. Ideally, a cryptographic hash function should possess the following properties:

- *collision resistance*: finding a pair $x \neq x' \in \{0,1\}^*$ such that $H(x) = H(x')$ should require $2^{l/2}$ operations
- *2nd preimage resistance*: for a given $x \in \{0,1\}^*$, finding a $x' \neq x$ such that $H(x) = H(x')$ should require 2^l operations
- *preimage resistance*: for a given $y \in \{0,1\}^l$, finding a $x \in \{0,1\}^*$ such that $H(x) = y$ should require 2^l operations.

All currently used hash functions are so-called *iterated* hash functions which are designed by iterating a *compression function* with a fixed-length input, say $h : \{0,1\}^{l+l'} \to \{0,1\}^l$. The iterated hash function H is then defined thanks to *domain extension* methods. The most popular one is the Merkle-Damgård method [5,18] which consists in first padding the input x so that the length of the padded message is a multiple of l' and outputing, for a padded message consisting of m l'-bit blocks $\mathrm{Pad}(x) = x_1\|\ldots\|x_m$, the value y_m defined by the recurrence $y_i = h(x_i\|y_{i-1})$, where y_0 is a fixed constant of $\{0,1\}^l$. The y_i's are called *chaining variables*. The popularity of the MD method comes from the fact that the hash function obtained is at least as resistant to collision attacks as the compression function. However, recent results have highlighted the intrinsic limitations of the MD approach [8,9] and motivated the study of other domain extension methods [1,4].

Most popular hash functions (e.g. MD5, SHA1) make use of compression functions build "from scratch", not appealing to any lower-level primitive. Another direction of research consists in trying to turn a block cipher into a compression function. This approach has been revived by the recent attacks on hash function using compression functions of dedicated design [28,27]. The question of how to turn a block cipher into a single block length (SBL) compression function (i.e. whose output length is the same as the block length of the block cipher) can be more or less considered as closed since the systematic study of Preneel *et al.* [23] and Black *et al.* [2]. However, the block length of the most trusted and standardized block ciphers such as DES and AES is too short to prevent collision attacks by the birthday paradox on SBL hash functions based on them. This is why there has been much effort in order to build a double block length (DBL) or more generally a multiple block length (MBL) compression function whose output is twice (or more) the block length of the block cipher. Most of the earlier proposals [3,15,16,22,24] turned out to have weaknesses [10,15]. Proofs of security for block cipher-based hash functions date back to Winternitz [29], who used the ideal cipher model of Shannon [25] to prove the security of the Davies-Meyer scheme against preimage attacks. Black *et al.* [2] used the same paradigm to study all the natural ways of building SBL compression functions, a work which had been initiated in [23]. Hirose [6,7] demonstrated the security of a family of DBL compression functions using two independent block ciphers with key length twice the block length, again in the ideal block cipher model. However, no secure DBL scheme using block ciphers with key length equal to the block length has been proposed so far. Nandi *et al.* [20] proposed DBL schemes with better rates than those of Hirose and claimed to have proved that an adversary must make $\Omega(2^{2n/3})$ oracle queries to get a collision and $\Omega(2^{4n/3})$ oracle queries to get a preimage. However, in light of the attacks presented in [11] (where a preimage attack requiring only $O(2^n)$ queries is described), we spotted a mistake in the security proof of [20]. One of the goal of this paper is to remedy the strategy they adopted.

At Asiacrypt '06 [21], Peyrin *et al.* presented a general framework to analyse how to combine secure compression functions in order to obtain compression functions with longer output. This approach had already been adopted in a series of papers [12,13,14] where partial answers were given thanks to error-correcting codes theory. Analysing two types of generic attacks, Peyrin *et al.* derived necessary conditions for the compression functions of their framework to be secure. Nevertheless, no security proofs were given. The aim of this paper is to analyse the constructions of the general framework introduced in [21] in a proof oriented manner. Though we will work in the fixed input length (FIL) random oracle model, this must be understood as a first step in the systematic study of MBL compression functions based on block ciphers.

The paper is organized as follows. In section 2 we establish the notations and some useful lemmas. In section 3 and 4 we carry out the security analysis for preimage resistance and collision resistance respectively. In section 5 we apply our results to previous proposals of block cipher-based hash functions and we draw our conclusions and propose future work in section 6.

2 Definitions and Notations

Basic Notations. In all the following, $\mathcal{I}_n$ will denote the set $\{0,1\}^n$, and $\mathcal{F}(a,b)$ the set of all functions from $\{0,1\}^a$ to $\{0,1\}^b$. We will often consider vectors of elements of $\mathcal{I}_n$ of various length which will be denoted by bold letters. For a binary vector $l = (l_1,\ldots,l_r) \in \{0,1\}^r$ and $\boldsymbol{X} = (X_1,\ldots,X_r) \in (\mathcal{I}_n)^r$, $\boldsymbol{X} \cdot l^T = l_1X_1 \oplus \cdots \oplus l_rX_r$. Similarly, for a binary matrix $L = [l_1^T,\ldots,l_s^T] \in \mathcal{M}_{r,s}(\{0,1\})$, $\boldsymbol{X} \cdot L$ is the vector $(\boldsymbol{X} \cdot l_1^T,\ldots,\boldsymbol{X} \cdot l_s^T)$. Given two vectors $\boldsymbol{X} = (X_1,\ldots,X_r)$ and $\boldsymbol{Y} = (Y_1,\ldots,Y_s)$, $\boldsymbol{X}\|\boldsymbol{Y}$ will denote the vector $(X_1,\ldots,X_r,Y_1,\ldots,Y_s)$. Finally, $\|\cdot\|_{\mathrm{H}}$ will denote the Hamming weight of a vector and $\mathbb{E}$ the expected value of a random variable.

Generic Constructions. The aim of this paper is to analyse the security of a very general class of compression functions build from smaller secure compression functions.

Namely, our building blocks will be t compression functions $f^{(1)}, \ldots, f^{(t)}$ taking each k n-bit blocks as input and outputing one n-bit block. For the security analysis, we will assume that these functions are independent random oracles. The larger compression function will take as input m n-bit message blocks and c n-bit chaining variable blocks, which will be denoted respectively $\boldsymbol{M} = (M_1, \ldots, M_m)$ and $\boldsymbol{H} = (H_1, \ldots, H_c)$. These blocks will be named *external* input blocks to distinguish them from the $t * k$ input blocks to the inner compression functions, which will be named *internal* input blocks. They are obtained as linear combinations of the external input blocks. Namely, for each $i \in [1..t]$, there is a binary matrix $\mathcal{A}_i \in \mathcal{M}_{(m+c,k)}(\{0,1\})$ such that the input to the i-th internal compression function is $\boldsymbol{M} \| \boldsymbol{H} \cdot \mathcal{A}_i$.

The output blocks of the internal compression functions

$$\boldsymbol{F} = (f^{(1)}(\boldsymbol{M} \| \boldsymbol{H} \cdot \mathcal{A}_1), \ldots, f^{(t)}(\boldsymbol{M} \| \boldsymbol{H} \cdot \mathcal{A}_t))$$

are then mixed by a linear output layer $\mathcal{B} \in \mathcal{M}_{(t,c)}(\{0,1\})$ to give the external output blocks $\boldsymbol{H'} = (H'_1, \ldots, H'_c)$ according to $\boldsymbol{H'} = \boldsymbol{F} \cdot \mathcal{B}$. In all the following, it will be assumed that $\mathcal{B}$ has full rank (otherwise the external output blocks are linearly dependent, which is clearly undesirable).

A *compression function construction* h is thus completely determined by the parameters (c, t, k, m) and the input and output layers $(\mathcal{A}_i)_{i \in [1..t]}$ and $\mathcal{B}$. The compression function obtained once the internal compression functions $f^{(1)}, \ldots, f^{(t)}$ are instantiated will be noted $h^{(f^{(1)}, \ldots, f^{(t)})}$. The construction can be summarized by the formula (see also Fig. 1)

$$h^{(f^{(1)}, \ldots, f^{(t)})}(\boldsymbol{M} \| \boldsymbol{H}) = \Big(f^{(1)}(\boldsymbol{M} \| \boldsymbol{H} \cdot \mathcal{A}_1), \ldots, f^{(t)}(\boldsymbol{M} \| \boldsymbol{H} \cdot \mathcal{A}_t)\Big) \cdot \mathcal{B}. \tag{1}$$

A more general framework could encompass a feedforward of the external input blocks,

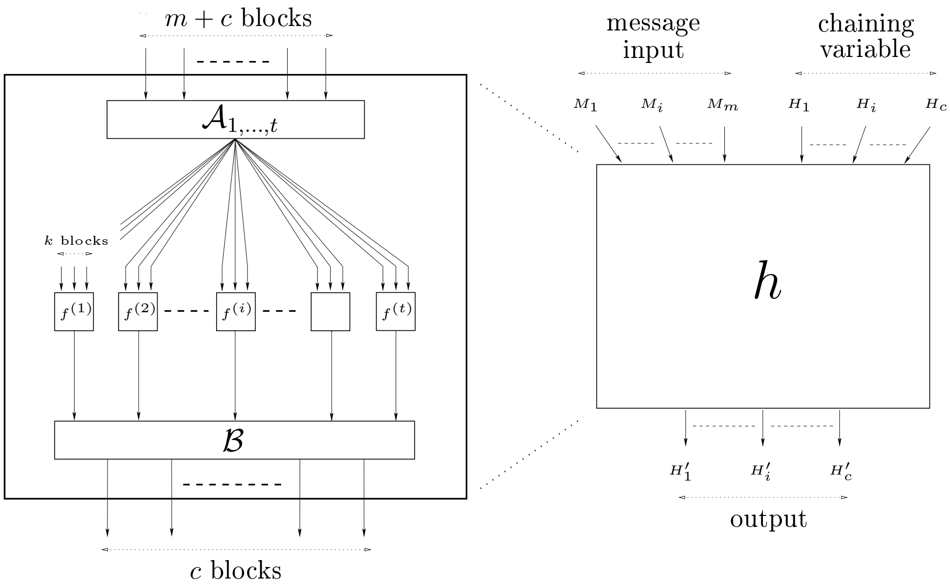

Fig. 1. The compression function h taking $(m + c)$ n-bit blocks in input and delivering c n-bit output blocks. It is build from t compression functions $f^{(i)}$ taking k n-bit input blocks and outputing one n-bit block.

i.e. allowing to xor some external input blocks with some internal output blocks. Though

it would definitely be useful in the ideal block cipher model, we do not believe this would strengthen in any way the constructions in the random oracle model, and thus we consider this feature as out of the scope of the article.
In the following, we will often consider linear combinations of the coordinates of the function h. For this, we will use the following notations. For $x \in \{0,1\}^c$, G_x will be the function

$$G_x : \boldsymbol{M}\|\boldsymbol{H} \mapsto h^{(f^{(1)},\dots,f^{(t)})}(\boldsymbol{M}\|\boldsymbol{H}) \cdot x^T.$$

Alternatively, G_x may be seen as a linear combination of the t functions $\boldsymbol{M}\|\boldsymbol{H} \mapsto f^{(i)}(\boldsymbol{M}\|\boldsymbol{H} \cdot \mathcal{A}_i)$. Indeed, writing $y = x \cdot \mathcal{B}^T \in \{0,1\}^t$, one has

$$G_x(\boldsymbol{M}\|\boldsymbol{H}) = (f^{(1)}(\boldsymbol{M}\|\boldsymbol{H} \cdot \mathcal{A}_1), \dots, f^{(t)}(\boldsymbol{M}\|\boldsymbol{H} \cdot \mathcal{A}_t)) \cdot y^T.$$

It will often be more convenient to define the set S_x of "active" inner compression functions, i.e. the set of integers $j \in [1..c]$ such that the j-th coordinate of $x \cdot \mathcal{B}^T$ is 1. Then G_x can be expressed by

$$G_x(\boldsymbol{M}\|\boldsymbol{H}) = \bigoplus_{j \in S_x} f^{(j)}(\boldsymbol{M}\|\boldsymbol{H} \cdot \mathcal{A}_j).$$

Security Model. In the following we will analyse the resistance of the compression functions we just described against preimage attacks and collision attacks. An *adversary* will be an algorithm with access to oracles for the inner compression functions $f^{(1)}, \dots, f^{(t)}$. Given a finite set S, $s \xleftarrow{\$} S$ denotes the operation of selecting s in the probability space S endowed with the uniform distribution. We will work in the random oracle model, meaning that the inner compression functions are uniformly and independently selected in the set $\mathcal{F}(kn, n)$. When these functions are asked queries from an algorithm, their output is uniform and independent from all other outputs, but consistent with answers to queries already asked. We now define the preimage and collision resistance of the compression function constructions.

Definition 1 (Preimage resistance of a compression function). *Let h be a (c,t,k,m)-compression function construction and let $\mathfrak{A}$ be an adversary. Then the advantage of $\mathfrak{A}$ in finding a preimage for h is the real number*

$$\boldsymbol{Adv}_h^{\mathrm{pre}}(\mathfrak{A}) = \Pr\Big[(f^{(1)}, \dots, f^{(t)}) \xleftarrow{\$} \mathcal{F}(kn,n)^t; \boldsymbol{H'} \xleftarrow{\$} (\mathcal{I}_n)^c; \\ \boldsymbol{M}\|\boldsymbol{H} \xleftarrow{\$} \mathfrak{A}(\boldsymbol{H'}) : h^{(f^{(1)},\dots,f^{(t)})}(\boldsymbol{M}\|\boldsymbol{H}) = \boldsymbol{H'}\Big].$$

We associate to each compression function construction h the insecurity measure

$$\boldsymbol{Adv}_h^{\mathrm{pre}}(q) = \max_{\mathfrak{A}}\{\boldsymbol{Adv}_h^{\mathrm{pre}}(\mathfrak{A})\}$$

where the maximum is taken over all adversaries making at most q oracle queries to each inner compression function $f^{(1)}, \dots, f^{(t)}$.

Definition 2 (Collision resistance of a compression function). *Let h be a (c,t,k,m)-compression function construction and let $\mathfrak{A}$ be an adversary. Then the advantage of $\mathfrak{A}$ in finding a collision for h is the real number*

$$\boldsymbol{Adv}_h^{\mathrm{coll}}(\mathfrak{A}) = \Pr\Big[(f^{(1)}, \dots, f^{(t)}) \xleftarrow{\$} \mathcal{F}(kn,n)^t; (\boldsymbol{M_1}\|\boldsymbol{H_1}, \boldsymbol{M_2}\|\boldsymbol{H_2}) \xleftarrow{\$} \mathfrak{A} : \\ \boldsymbol{M_1}\|\boldsymbol{H_1} \neq \boldsymbol{M_2}\|\boldsymbol{H_2} \wedge h^{(f^{(1)},\dots,f^{(t)})}(\boldsymbol{M_1}\|\boldsymbol{H_1}) = h^{(f^{(1)},\dots,f^{(t)})}(\boldsymbol{M_2}\|\boldsymbol{H_2})\Big].$$

We associate to each compression function construction h the insecurity measure

$$\boldsymbol{Adv}_h^{\mathrm{coll}}(q) = \max_{\mathfrak{A}}\{\boldsymbol{Adv}_h^{\mathrm{coll}}(\mathfrak{A})\}$$

where the maximum is taken over all adversaries making at most q oracle queries to each inner compression function $f^{(1)}, \dots, f^{(t)}$.

For the remainder of this paper we will make the following classical assumptions regarding the adversaries. First, they are computationally unbounded, in consequence of what we can restrain ourselves wlog to deterministic adversaries (so that we do not have to take into account any more the randomness coming from the random choices of the algorithm). Second, an adversary does not make the same oracle query more than once. Third, we will restrain ourselves to adversaries making exactly q queries to each inner compression function. These assumptions does not restrict the generality of the analysis in that for any adversary $\mathfrak{A}$ asking at most q queries there exists another adversary $\mathfrak{A}'$ verifying the assumptions that achieves at least the same advantage as $\mathfrak{A}$.

Type I Constructions. A natural requirement for the compression functions studied here would be that the image of two distinct inputs by any linear combination of the output blocks are independent. This is generally not the case, and compressions functions which does not possess this property are subject to devastating attack called DF attacks (*degrees of freedom*) in [21]. This feature is achieved by letting every external output block depend on all external input block, no matter which invertible transformations of the external inputs and outputs are used. Expressing it mathematically yields the following definition.

Definition 3 (Type I (for Independent) compression function construction). *A (c, t, k, m)-compression function construction will be said to be of type I iff for all $x \in \{0,1\}^c \setminus \{0\}$, $\bigcap_{j \in S_x} \ker \mathcal{A}_j = \{0\}$.*

For such constructions, one can prove the following property (the proof is given in Appendix A).

Lemma 1. *Let h be a (c, t, k, m) compression function construction of type I. Then for all $x \in \{0,1\}^c \setminus \{0\}$, and for all distinct $\boldsymbol{M_1}\|\boldsymbol{H_1}$ and $\boldsymbol{M_2}\|\boldsymbol{H_2}$, $G_x(\boldsymbol{M_1}\|\boldsymbol{H_1})$ and $G_x(\boldsymbol{M_2}\|\boldsymbol{H_2})$ are uniformly random and independent.*

Not all parameter sets permit to build type I compression function constructions. More precisely, one has the following necessary condition, which was proved in [21].

Lemma 2 ([21]). *Let h be a (c, t, k, m) compression function construction of type I. Then necessarily $\forall x \in \{0,1\}^c \setminus \{0\}$, $\|x \cdot \mathcal{B}^T\|_H \geq \frac{m+c}{k}$. In other words, $\mathcal{B}$ must have minimal distance at least $\lceil \frac{m+c}{k} \rceil$.*

Computable Inputs. In order to make our explanations more rigorous, we will need the following notions of *computability*, which are generalizations of concepts introduced in [20]. Informally speaking, once an adversary has made certain queries to the inner compression functions, we want to define for each $\boldsymbol{M}\|\boldsymbol{H}$ the number of coordinates of $h(\boldsymbol{M}\|\boldsymbol{H})$ the adversary is able to compute.

Definition 4 (G_x-computable input). *Let $\mathcal{Q}_1, \dots, \mathcal{Q}_t \subset (\mathcal{I}_n)^k$ be sets of queries to each of the inner compression functions. For $x \in \{0,1\}^c$, we will say that an external input $\boldsymbol{M}\|\boldsymbol{H} \in (\mathcal{I}_n)^{m+c}$ is G_x-*computable *with respect to these sets of queries if $\boldsymbol{M}\|\boldsymbol{H} \cdot \mathcal{A}_i \in \mathcal{Q}_i$, for each $i \in S_x$.*

It is easy to verify that given sets of queries $\mathcal{Q}_1, \ldots, \mathcal{Q}_t$ and $x_1, \ldots, x_r \in \{0,1\}^c$, $\boldsymbol{M}\|\boldsymbol{H}$ is G_{x_i}-computable for all $i \in [1..r]$ implies that $\boldsymbol{M}\|\boldsymbol{H}$ is G_x-computable for all $x \in \mathrm{Vec}(x_1, \ldots, x_r)$. It is thus natural to give the following definitions.

Definition 5 (V-computable input). *Let V be a subspace of $\{0,1\}^c$, $V \neq \emptyset$, let $\mathcal{Q}_1, \ldots, \mathcal{Q}_t \subset (\mathcal{I}_n)^k$ be the sets of queries to each of the inner compression functions. We will say that an external input $\boldsymbol{M}\|\boldsymbol{H} \in (\mathcal{I}_n)^{m+c}$ is* V-computable *with respect to these sets of queries if $\boldsymbol{M}\|\boldsymbol{H}$ is G_x-computable for all $x \in V$.*
Let $V_{\boldsymbol{M}\|\boldsymbol{H}}$ be the biggest subspace such that $\boldsymbol{M}\|\boldsymbol{H}$ is V-computable (possibly reduced to $\{0\}$). If r is the dimension of $V_{\boldsymbol{M}\|\boldsymbol{H}}$, we will say that $\boldsymbol{M}\|\boldsymbol{H}$ is r-computable. *We will also talk of* h-computable *input when $r = c$ and of* uncomputable *input when $r = 0$.*

Definition 6 (Maximal number of (at least) r-computable inputs with q queries). *Let h be a compression function construction and $q \geq 1$. We define the maximal number of (at least) r-computable inputs with q queries $\beta_r(q)$ as being*

$$\beta_r(q) = \max_{\mathcal{Q}_1,\ldots,\mathcal{Q}_t} \#\{\boldsymbol{M}\|\boldsymbol{H} \in (\mathcal{I}_n)^{m+c} \mid \boldsymbol{M}\|\boldsymbol{H} \textit{ is at least } r\textit{-computable}\}$$

where the maximum is taken over all the possible sets of q queries to the inner compression functions.
We will also need the following slightly different notion for $r = 1$:

$$\beta_1'(q) = \max_{x\in\{0,1\}^c\setminus\{0\}} \max_{\mathcal{Q}_1,\ldots,\mathcal{Q}_t} \#\{\boldsymbol{M}\|\boldsymbol{H} \in (\mathcal{I}_n)^{m+c} \mid \boldsymbol{M}\|\boldsymbol{H} \textit{ is } G_x\textit{-computable}\}$$

where the maximum is taken over all the non-zero linear combinations of output blocks and over all the possible sets of q queries to the inner compression functions.

The following proposition is rather obvious and given without proof.

Proposition 1.

$$q \leq \beta_c(q) \leq \beta_{c-1}(q) \leq \ldots \leq \beta_1(q).$$

$\beta_1(q)$ and $\beta_1'(q)$ capture approximately the same characteristic of the compression function for it is immediate to verify that

$$\beta_1'(q) \leq \beta_1(q) \leq 2^c \beta_1'(q). \tag{2}$$

In our security analysis, we will make an extensive use of the following lemma (see the proof in Appendix A).

Lemma 3 (Independency lemma). *Let $\mathcal{Q}_1, \ldots, \mathcal{Q}_t \subset (\mathcal{I}_n)^k$ be sets of queries to the inner compression functions such that $\boldsymbol{M}\|\boldsymbol{H}$ is r-computable. Let $(x_1, \ldots, x_r)$ be a basis of $V_{\boldsymbol{M}\|\boldsymbol{H}}$, and $X_1, \ldots, X_r, X \in \mathcal{I}_n$. Then $\forall x \in \{0,1\}^c \setminus V_{\boldsymbol{M}\|\boldsymbol{H}}$,*

$$\Pr[G_x(\boldsymbol{M}\|\boldsymbol{H}) = X \mid G_{x_1}(\boldsymbol{M}\|\boldsymbol{H}) = X_1, \ldots, G_{x_r}(\boldsymbol{M}\|\boldsymbol{H}) = X_r] = \frac{1}{2^n}.$$

More generally, if $\boldsymbol{H'}$ is such that for all $i \in [1..r]$, $\boldsymbol{H'} \cdot x_i^T = X_i$, then

$$\Pr[h(\boldsymbol{M}\|\boldsymbol{H}) = \boldsymbol{H'} \mid G_{x_1}(\boldsymbol{M}\|\boldsymbol{H}) = X_1, \ldots, G_{x_r}(\boldsymbol{M}\|\boldsymbol{H}) = X_r] = \frac{1}{2^{(c-r)n}}.$$

3 Security Analysis for Preimage Resistance

In this section we begin with providing a security bound to preimage attacks for the constructions of the general framework studied in this paper. Then we show that this security bound is tight by analysing an attack whose advantage is close to the security bound.

Theorem 1 (Security bound for preimage resistance). *Let h be a (c,t,k,m)-compression function construction (non necessarily of type I) with parameter $\beta_1(q)$ defined by definition 6. Then*

$$\boldsymbol{Adv}_h^{\mathrm{pre}}(q) \leq \frac{1}{2^n} + \frac{\beta_1(q)}{2^{cn}}.$$

Proof. Let $\mathfrak{A}$ be a preimage-finding adversary attacking the compression function h. We suppose wlog that the random input $\boldsymbol{H'}$ to the adversary is $\mathbf{0}$. We first define Preim as being the set of external inputs $\boldsymbol{M}\|\boldsymbol{H}$ which are h-computable with respect to the final sets of queries of $\mathfrak{A}$ and such that $h^{(f^{(1)},\dots,f^{(t)})}(\boldsymbol{M}\|\boldsymbol{H}) = \mathbf{0}$. First of all, if $\mathfrak{A}$ does not find any external input in this set, its probability of success is very low. Indeed, $\mathfrak{A}$ is bound to output an $\boldsymbol{M}\|\boldsymbol{H}$ which is not h-computable. According to Lemma 3, the probability for this output to be a good preimage is $\leq 1/2^n$. Therefore we have $\Pr[\mathfrak{A} \text{ wins}] \leq 1/2^n + \Pr[\mathrm{Preim} \neq \emptyset]$.
We now bound $\Pr[\mathrm{Preim} \neq \emptyset]$. For this, we analyse the behavior of $\mathfrak{A}$ in a sequential manner: $\mathfrak{A}$ makes its queries to the inner functions in a certain order. During this process, each external input $\boldsymbol{M}\|\boldsymbol{H}$ goes through successive states: either it is uncomputable, or it is r-computable and still a potential candidate to be mapped on to $\mathbf{0}$, or it is r-computable and discarded because there exists $x \in V_{\boldsymbol{M}\|\boldsymbol{H}}$ such that $G_x(\boldsymbol{M}\|\boldsymbol{H}) \neq 0$. More precisely, consider partial sets of queries $\mathcal{Q}'_1, \dots, \mathcal{Q}'_t \subset (\mathcal{I}_n)^k$. We will say that an external input $\boldsymbol{M}\|\boldsymbol{H}$ is *compatible* with these partial sets of queries if $\forall x \in V_{\boldsymbol{M}\|\boldsymbol{H}}, G_x(\boldsymbol{M}\|\boldsymbol{H}) = 0$. Note that an external input which is h-computable with respect to the final sets of queries of $\mathfrak{A}$ was necessarily 1-computable at some stage in the sequential queries of $\mathfrak{A}$. Said differently, an external input cannot "jump" from the state *uncomputable* to a state where it is r-computable for $r > 1$ with one single query because one single query never enables to compute more than one output block or linear combination of output blocks[1]. When it exists, we will note $G^1_{\boldsymbol{M}\|\boldsymbol{H}}$ the linear combination of output blocks associated with the first $x \in \{0,1\}^c \setminus \{0\}$ such that $\boldsymbol{M}\|\boldsymbol{H}$ is G_x-computable. Let us define the set Pot_1 as being the set of all $\boldsymbol{M}\|\boldsymbol{H}$ such that, at some stage in the sequential queries of $\mathfrak{A}$, $\boldsymbol{M}\|\boldsymbol{H}$ was 1-computable and compatible. Then one clearly has $\mathrm{Pot}_1 \supset \mathrm{Preim}$, so that

$$\Pr[\boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Preim}] = \Pr[\boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Preim} | \boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Pot}_1] \cdot \Pr[\boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Pot}_1].$$

The key point in the proof is the fact that according to Lemma 3, one has, for all $\boldsymbol{M}\|\boldsymbol{H}$,

$$\begin{aligned}\Pr[\boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Preim} | \boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Pot}_1] &\leq \Pr[h^{(f^{(1)},\dots,f^{(t)})}(\boldsymbol{M}\|\boldsymbol{H}) = \mathbf{0} | \boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Pot}_1] \\ &\leq \frac{1}{2^{(c-1)n}}.\end{aligned}$$

In consequence,

$$\Pr[\mathrm{Preim} \neq \emptyset] \leq \frac{1}{2^{(c-1)n}} \sum_{\boldsymbol{M}\|\boldsymbol{H}} \Pr[\boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Pot}_1].$$

[1] Suppose that one single query enables to compute both $G_{x_1}(\boldsymbol{M}\|\boldsymbol{H})$ and $G_{x_2}(\boldsymbol{M}\|\boldsymbol{H})$. This means that all the other queries necessary to compute them have been made previously. But this implies that $\boldsymbol{M}\|\boldsymbol{H}$ is already $G_{x_1 \oplus x_2}$-computable, so that in fact the computability of $\boldsymbol{M}\|\boldsymbol{H}$ has only been increased by 1.

Now we want to bound the sum $\sum_{\boldsymbol{M}\|\boldsymbol{H}} \Pr[\boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Pot}_1]$. Recall that by the definition of Pot_1, $\boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Pot}_1$ is the event that $\boldsymbol{M}\|\boldsymbol{H}$ is at least 1-computable with respect to the final sets of queries of $\mathfrak{A}$, and $G^1_{\boldsymbol{M}\|\boldsymbol{H}}(\boldsymbol{M}\|\boldsymbol{H}) = 0$. Now conditioning on the event that $\boldsymbol{M}\|\boldsymbol{H}$ is at least 1-computable with respect to the final sets of queries of $\mathfrak{A}$, the probability that $G^1_{\boldsymbol{M}\|\boldsymbol{H}}(\boldsymbol{M}\|\boldsymbol{H}) = 0$ is $1/2^n$. Summing up this reasoning with formulas yields

$$\begin{aligned}\sum_{\boldsymbol{M}\|\boldsymbol{H}} \Pr[\boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Pot}_1] &\leq \sum_{\boldsymbol{M}\|\boldsymbol{H}} \Pr[G^1_{\boldsymbol{M}\|\boldsymbol{H}}(\boldsymbol{M}\|\boldsymbol{H}) = 0 | \boldsymbol{M}\|\boldsymbol{H} \text{ is 1-computable}] \cdot \\ &\qquad\qquad \Pr[\boldsymbol{M}\|\boldsymbol{H} \text{ is 1-computable}] \\ &\leq \frac{1}{2^n} \sum_{\boldsymbol{M}\|\boldsymbol{H}} \Pr[\boldsymbol{M}\|\boldsymbol{H} \text{ is 1-computable}] \\ &\leq \frac{1}{2^n} \mathbb{E}\left(\#\{\boldsymbol{M}\|\boldsymbol{H} \mid \boldsymbol{M}\|\boldsymbol{H} \text{ is 1-computable}\}\right) \\ &\leq \frac{\beta_1(q)}{2^n}.\end{aligned}$$

The theorem follows immediately. □

Remark 1. The reasoning used in [20] concludes that preimage resistance is $O(\beta_c(q)/2^{cn})$, which cannot be in view of the generic attack presented hereafter. We reproduce this faulty reasoning and point out the mistake in Appendix C.

Theorem 2 (Preimage attack matching the security bound). *Let h be a (c, t, k, m)-compression function construction of type I with parameter $\beta'_1(q)$ defined by definition 6. Then $\beta'_1(q) = \Omega(2^{cn})$ and $q = \Omega(2^{(c-1)n})$ implies that $\boldsymbol{Adv}_h^{\mathrm{pre}}(q) = \Omega(1)$.*

Proof. Once again we suppose wlog that the random input $\boldsymbol{H'}$ to the adversary is $\boldsymbol{0}$. Consider the following adversary: $\mathfrak{A}$ first identifies $x \in \{0,1\}^c \setminus \{0\}$ such that $\beta'_1(q)$ is reached and makes the q queries to the inner compression functions $f^{(i)}$ involved in the calculation of G_x (i.e. such that $i \in S_x$), thus obtaining $\beta'_1(q)$ images by G_x. Let N_{G_x} be the random variable counting among these $\beta'_1(q)$ G_x-computable inputs the number of them such that $G_x(\boldsymbol{M}\|\boldsymbol{H}) = 0$. The compression function construction considered being of type I, the $\beta'_1(q)$ images by G_x obtained are random and pairwise independent. As $\beta'_1(q) = \Omega(2^{cn})$, with overwhelming probability $N_{G_x} = \Omega(2^{(c-1)n})$. After this first step, $\mathfrak{A}$ selects $\min(N_{G_x}, q)$ $\boldsymbol{M}\|\boldsymbol{H}$ such that $G_x(\boldsymbol{M}\|\boldsymbol{H}) = 0$. If $N_{G_x} > q$, it selects them randomly. $\mathfrak{A}$ then queries the remaining compression functions in order to obtain the full image of the selected external inputs. Restricting the number of selected external inputs to q ensures that $\mathfrak{A}$ is always able to obtain their image by h. The probability for one of these external inputs to be a good preimage is $1/2^{(c-1)n}$. As $q = \Omega(2^{(c-1)n})$ by hypothesis, the adversary finds a preimage of $\boldsymbol{0}$ with non-negligible probability. Hence the result. □

Conclusion for Preimage Resistance. The results of this section show that preimage resistance of a (c, t, k, m)-compression function construction of type I is governed by the parameter $\beta_1(q)$. Combining Theorems 1 and 2, and recalling inequality (2) proves that, at least for constructions such that one may have $\beta'_1(q) = \Omega(2^{cn})$ and $q = \Omega(2^{(c-1)n})$ at the same time, preimage resistance is $\Theta(\beta_1(q)/2^{cn})$.

4 Security Analysis for Collision Resistance

Theorem 3 (Security bound for collision resistance). *Let h be a (c, t, k, m) compression function construction (non necessarily of type I) with parameter $\beta_1(q)$ defined by* *definition 6. Then*

$$\boldsymbol{Adv}_h^{\mathrm{coll}}(q) \leq \frac{1}{2^n} + \frac{\beta_1(q)^2}{2 \cdot 2^{cn}}.$$

The proof of this theorem is very similar to the proof of Theorem 1 and is given in Appendix B. We now exhibit two collision attacks matching sometimes the security bound.

First Collision Attack. The first attack presented here is very simple and meets the security bound in some cases. It simply consists in computing the image by h of $\beta_c(q)$ external inputs. For a type I construction, they are random and independent, and a classical calculus tells us that the probability to obtain a collision is $1-\prod_{i=1}^{\beta_c(q)-1}(1-i/2^{cn})$. As a consequence we have the following result:

Theorem 4 (First collision attack.). *Let h be a (c,t,k,m)-compression function construction of type I with parameter $\beta_c(q)$ defined by definition 6. Then $\boldsymbol{Adv}_h^{\mathrm{coll}}(q) \geq 0.6\frac{\beta_c(q)(\beta_c(q)-1)}{2 \cdot 2^{cn}}$.*
In consequence, for constructions such that $\beta_c(q) \sim \beta_1(q)$ when $q \to \infty$, the security bound given in Theorem 3 is tight.

Proof. We have the following inequalities:

$$\begin{aligned}
\mathbf{Adv}_h^{\mathrm{coll}}(q) &\geq 1 - \prod_{i=1}^{\beta_c(q)-1}(1 - \frac{i}{2^{cn}}) \\
&\geq 1 - \exp\left(-\sum_{i=1}^{\beta_c(q)-1}\frac{i}{2^{cn}}\right) \\
&= 1 - \exp\left(-\frac{\beta_c(q)(\beta_c(q)-1)}{2 \cdot 2^{cn}}\right) \\
&\geq \left(1 - \frac{1}{e}\right)\frac{\beta_c(q)(\beta_c(q)-1)}{2 \cdot 2^{cn}}.
\end{aligned}$$

The inequality $(1-e^{-1}) > 0.6$ completes the proof. □

Second Collision Attack. The second attack is more similar to the preimage attack presented previously and may achieve or not a better advantage than the first one depending on the input and output mappings. The adversary proceeds as follows. $\mathfrak{A}$ first identifies $x \in \{0,1\}^c \setminus \{0\}$ such that $\beta'_1(q)$ is reached and makes the q queries to the inner compression functions involved in the calculation of G_x, thus obtaining $\beta'_1(q)$ images by G_x. $\mathfrak{A}$ quotients the set of the external inputs which are G_x-computable at this stage by the equivalence relation $G_x(\boldsymbol{M_1}\|\boldsymbol{H_1}) = G_x(\boldsymbol{M_2}\|\boldsymbol{H_2})$ and orders the quotient classes by decreasing cardinal. It then calculates the full image by h of the elements of the quotient classes, looking for a collision on the $(c-1)$ remaining output blocks, and beginning with the quotient class of larger cardinal in order to maximize its probability of success. $\mathfrak{A}$ is able to calculate at least q images. Analysing this adversary enables to enunciate the following result.

Theorem 5 (Second collision attack). *Let h be a (c,t,k,m)-compression function construction of type I with parameter $\beta'_1(q)$ defined by definition 6. Then $q\beta'_1(q) = \Omega(2^{cn})$ and $\beta'_1(q) = \Omega(n2^n)$ implies $\boldsymbol{Adv}_h^{\mathrm{coll}}(q) = \Omega(1)$.*

Proof. Let us analyse the probability of success of the adversary we just described. The fact that $\beta'_1(q) = \Omega(n2^n)$ implies that with probability $1 - O(1)$, $\mathfrak{A}$ obtains 2^n quotient classes containing $\Theta(\beta'_1(q)/2^n)$ elements each (this is a classical "balls and bins" result, see for example [19]). Though $\mathfrak{A}$ will not always be able to obtain the image by h of all the $\beta'_1(q)$ inputs, we can ensure that it will be able to do so for at least q inputs. So the number C of quotient classes in which it will be able to look for a full collision under h is such that $C \cdot \frac{\beta'_1(q)}{2^n} = q$, i.e. $C = \frac{q2^n}{\beta'_1(q)}$. The events "finding a collision in quotient class i", $i \in [1..C]$ are independent and their probability p_i verify (the proof is analog to the proof of Theorem 4)

$$p_i \geq 0.6 \frac{\#\mathcal{C}_i(\#\mathcal{C}_i - 1)}{2^{(c-1)n}}$$

where $\#\mathcal{C}_i$ is the cardinal of the quotient class being explored for a full collision. As $\#\mathcal{C}_i = \Theta(\beta'_1(q)/2^n))$ with overwhelming probability, we have that the total probability to find a collision is

$$\Omega\left(C \cdot \frac{\left(\frac{\beta'_1(q)}{2^n}\right)^2}{2^{(c-1)n}}\right) = \Omega\left(\frac{q\beta'_1(q)}{2^{cn}}\right).$$

Consequently $q\beta'_1(q) = \Omega(2^{cn})$ implies that the probability of success of the adversary is $\Omega(1)$. This concludes the proof. $\square$

Conclusion for Collision Resistance. The security analysis of (c, t, k, m)-compression functions for collision resistance is not as tight as for preimage resistance. We proved in this section that collision resistance is $O(\beta_1(q)^2/2^{cn})$, while the attacks we described show that a lower bound for collision resistance is $\Omega(\max(\beta_c(q)^2, q\beta_1(q))/2^{cn})$.

5 Application to Previously Proposed Schemes

Hirose Schemes. We call Hirose schemes the (c, t, k, m)-compression function constructions where $k = m + c$. In this case, using only $t = c$ inner compression functions, setting $\boldsymbol{M}\|\boldsymbol{H} \cdot \mathcal{A}_i = \boldsymbol{M}\|\boldsymbol{H}$ for all $i \in [0, t]$ and taking for $\mathcal{B}$ the $c \times c$ identity matrix yields a compression function such that $\beta_1(q) = \beta_c(q) = q$, so that its preimage resistance is $\Theta(\frac{q}{2^{cn}})$ and its collision resistance is $\Theta(\frac{q^2}{2^{cn}})$, which is optimal. This is not a surprising result since it is easy to see that in the random oracle model, the compression function obtained is itself a random function from $\mathcal{F}((m + c)n, cn)$. These type of schemes have been studied by Hirose in [6,7], where it is shown how to construct such an optimally resistant compression function with one ideal block cipher when $c = 2$.

Nandi *et al.* Schemes. Nandi *et al.* proposed two schemes in [20] which are depicted in Fig. 2. For these schemes, it was shown in [20] that $\beta_2(q) \leq q^{3/2}$ and it is not difficult to convince oneself that $\beta'_1(q) \leq q^2$. Conversely, $\beta_2(q) \geq \lfloor q^{1/2} \rfloor^3$ and, for $q \leq 2^n$, $\beta_1(q) \geq q^2$. Consequently their preimage resistance is $\Theta(q^2/2^{2n})$, and an attack requiring $\Theta(2^n)$ operations was described in [11]. For the collision resistance, our security proof shows that it is $O(q^4/2^{2n})$ while the two collision attacks we described achieve advantage $\Omega(q^3/2^{2n})$. The authors of [20] claimed to have proved that collision resistance for their schemes is $O(q^3/2^{2n})$, however we explain in Appendix C why their reasoning is incorrect. Nevertheless, we conjecture that our security proof can be enhanced to prove that collision resistance is indeed $O(q^3/2^{2n})$. But this must not discourage to look in the direction of finding better collision attacks than the one described in [11], which needs $\Theta(2^{2n/3})$ oracle queries.

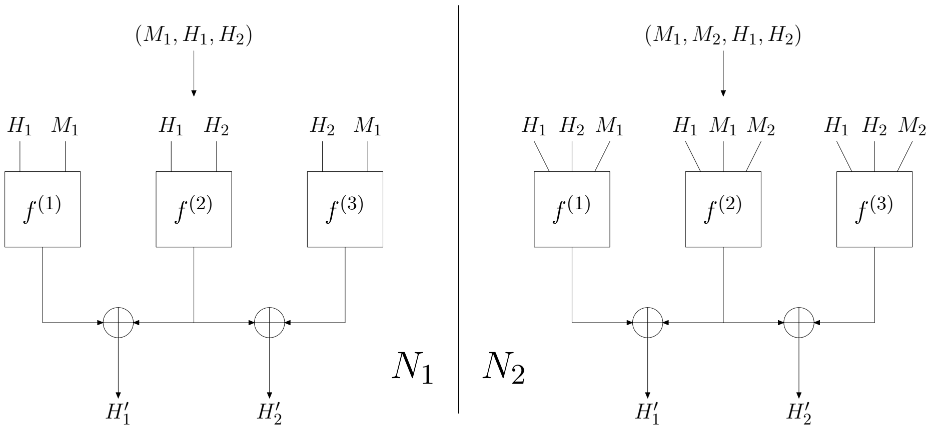

Fig. 2. Nandi *et al.* schemes [20].

Peyrin *et al.* Schemes. Peyrin *et al.* proposed two schemes in [21] which verified the necessary conditions they established for a scheme to be secure. They are depicted in Fig. 3. For the first scheme, one can prove with techniques similar to the ones used for Nandi *et al.* schemes that $\beta_1(q) = \Theta(q^{3/2})$ and $\beta_2(q) = \Theta(q^{3/2})$, so that the security analysis is tight in the collision case (collision resistance is $\Theta(q^3/2^{2n})$ as well as in the preimage case (preimage resistance is $\Theta(q^{3/2}/2^{2n})$). For the second scheme, $\beta_1(q) = \Theta(q^{3/2})$ and $\beta_2(q) = \Theta(q^{4/3})$. Here preimage resistance is $\Theta(q^{3/2}/2^{2n})$, and collision resistance is $O(q^3/2^{2n})$, while the first collision attack achieves advantage $\Omega(q^{8/3}/2^{2n})$. Here again it is an open question to close the gap between the security proof and the attack.

Related Algorithmic Problems. We want to emphasize that one must make a clear distinction between security analysis in terms of number of *oracle queries* and number of *operations*. While the number of queries is an obvious lower bound for the number of operations, it is not always clear how an attacker will be able to reach this lower complexity bound. For example for the scheme N1 of Fig. 2, the preimage resistance is $\Theta(q^2/2^{2n})$ so that an adversary must make $\Theta(2^n)$ oracle queries to find a preimage with non negligible probability. The authors of [11] presented an attack also requiring $\Theta(2^n)$ operations. Fundamentally, this is achievable thanks to an efficient algorithm for solving the so-called 2-sum problem which consists in finding, in two lists L_1 and L_2, two elements $x_1 \in L_1$ and $x_2 \in L_2$ such that $x_1 \oplus x_2 = 0$. The generalization to k lists was thoroughly studied by Wagner [26]. In the same way as the (in)security of the schemes of Nandi *et al.* is linked to efficient ways of solving the 2-sum problem [26], we conjecture that the security in terms of operations of the schemes of Peyrin *et al.* is related to the 3-sum problem, for which no good algorithm is known. Giving a reductionist security proof linking the security of these schemes to a 3-sum hard problem would be an elegant result.

6 Concluding Remarks

In this paper we conducted the security analysis in terms of oracle queries of very general constructions combining compression functions modeled as independent FIL random oracles to obtain a compression function with longer output. Using the concept of computable input, we gave a security bound for preimage resistance and collision resistance which is tight for some constructions.

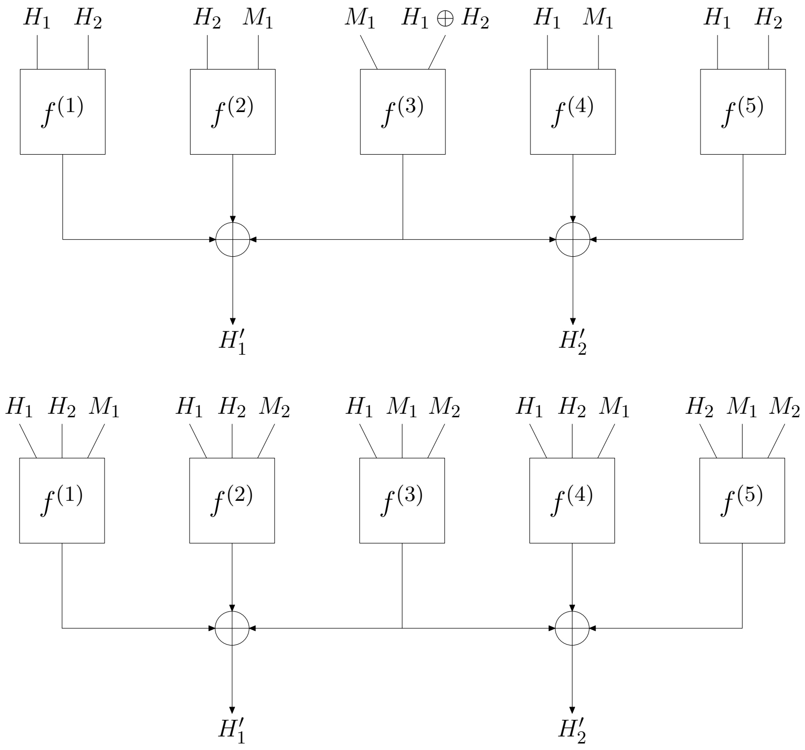

Fig. 3. Peyrin *et al.* schemes [21].

Future work includes carrying the security analysis in the ideal block cipher model as it was done for Hirose schemes [6,7], a more systematic study of the parameters $\beta_i(q)$, and closing in the general case the security gap, especially for collision resistance.

Acknowledgements

The authors are grateful to Henri Gilbert for his helpful comments.

References

1. Mihir Bellare and Thomas Ristenpart, *Multi-property-preserving hash domain extension and the EMD transform*, Advances in Cryptology – ASIACRYPT '06 (Xuejia Lai and Kefei Chen, eds.), Lecture Notes in Computer Science, vol. 4284, Springer-Verlag, 2006, pp. 299–314.
2. John R. Black, Phillip Rogaway, and Thomas Shrimpton, *Black-box analysis of the block-cipher-based hash-function constructions from PGV*, Advances in Cryptology – CRYPTO 2002 (Moti Yung, ed.), Lecture Notes in Computer Science, vol. 2442, Springer-Verlag, 2002, pp. 320–335.
3. D. Coppersmith, S. Pilpel, C.H. Meyer, S.M. Matyas, M.M. Hyden, J. Oseas, B. Brachtl, and M. Schilling, *Data authentication using modification dectection codes based on a public one way encryption function*, U.S. Patent No. 4,908,861, March 13, 1990.
4. Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya, *Merkle-Damgård revisited: How to construct a hash function*, Advances in Cryptology – CRYPTO 2005 (Victor Shoup, ed.), Lecture Notes in Computer Science, vol. 3621, Springer-Verlag, 2005, pp. 430–448.

5. Ivan Damgård, *A design principle for hash functions*, Advances in Cryptology – CRYPTO '89 (Gilles Brassard, ed.), Lecture Notes in Computer Science, vol. 435, Springer-Verlag, 1989, pp. 416–427.
6. Shoichi Hirose, *Provably secure double-block-length hash functions in a black-box model*, Information Security and Cryptology – ICISC 2004 (C. Park and S. Chee, eds.), Lecture Notes in Computer Science, vol. 3506, Springer-Verlag, 2004, pp. 330–342.
7. ______, *Some plausible constructions of double-block-length hash functions*, Fast Software Encryption – FSE 2006 (M.J.B. Robshaw, ed.), Lecture Notes in Computer Science, vol. 4047, Springer-Verlag, 2006.
8. Antoine Joux, *Multicollisions in iterated hash functions. application to cascaded constructions*, Advances in Cryptology – CRYPTO 2004 (Matthew K. Franklin, ed.), Lecture Notes in Computer Science, vol. 3152, Springer-Verlag, 2004, pp. 306–316.
9. John Kelsey and Bruce Schneier, *Second preimages on n-bit hash functions for much less than 2^n work*, Advances in Cryptology – EUROCRYPT 2005 (Ronald Cramer, ed.), Lecture Notes in Computer Science, vol. 3494, Springer-Verlag, 2005, pp. 474–490.
10. Lars R. Knudsen and Xuejia Lai, *New attacks on all double block length hash functions of hash rate 1, including the parallel-DM*, Advances in Cryptology – EUROCRYPT '94 (Alfredo De Santis, ed.), Lecture Notes in Computer Science, vol. 950, Springer-Verlag, 1994, pp. 410–418.
11. Lars R. Knudsen and Frederic Muller, *Some attacks against a double length hash proposal*, Advances in Cryptology – ASIACRYPT '05 (B. Roy, ed.), Lecture Notes in Computer Science, vol. 3788, Springer-Verlag, 2005, pp. 462–473.
12. Lars R. Knudsen and Bart Preneel, *Hash functions based on block ciphers and quaternary codes*, Advances in Cryptology – ASIACRYPT '96 (Kwangjo Kim and Tsutomu Matsumoto, eds.), Lecture Notes in Computer Science, vol. 1163, Springer-Verlag, 1996, pp. 77–90.
13. ______, *Fast and secure hashing based on codes*, Advances in Cryptology – CRYPTO '97 (Burton S. Jr. Kaliski, ed.), Lecture Notes in Computer Science, vol. 1294, Springer-Verlag, 1997, pp. 485–498.
14. ______, *Construction of secure and fast hash functions using nonbinary error-correcting codes*, IEEE Transactions on Information Theory **48** (2002), no. 9, 2524–2539.
15. Xuejia Lai and James L. Massey, *Hash function based on block ciphers*, Advances in Cryptology – EUROCRYPT '92 (Rainer A. Rueppel, ed.), Lecture Notes in Computer Science, vol. 658, Springer-Verlag, 1992, pp. 55–70.
16. Xuejia Lai, Christian Waldvogel, Walter Hohl, and Thomas Meier, *Security of iterated hash functions based on block ciphers*, Advances in Cryptology – CRYPTO '93 (Douglas Robert Stinson, ed.), Lecture Notes in Computer Science, vol. 773, Springer-Verlag, 1993, pp. 379–390.
17. Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
18. Ralph C. Merkle, *One way hash functions and DES*, Advances in Cryptology – CRYPTO '89 (Gilles Brassard, ed.), Lecture Notes in Computer Science, vol. 435, Springer-Verlag, 1989, pp. 428–446.
19. Rajeev Motwani and Prabhakar Raghavan, *Randomized algorithms*, Cambridge University Press, 1995.
20. Mridul Nandi, Wonil Lee, Kouichi Sakurai, and Sangjin Lee, *Security analysis of a 2/3-rate double length compression function in black-box model*, Fast Software Encryption – FSE 2005 (Henri Gilbert and Helena Handschuh, eds.), Lecture Notes in Computer Science, vol. 3557, Springer-Verlag, 2005, pp. 243–254.
21. Thomas Peyrin, Henri Gilbert, Frédéric Muller, and Matthew J. B. Robshaw, *Combining compression functions and block cipher-based hash functions*, Advances in Cryptology – ASIACRYPT '06 (Xuejia Lai and Kefei Chen, eds.), Lecture Notes in Computer Science, vol. 4284, Springer-Verlag, 2006, pp. 315–331.
22. Bart Preneel, Antoon Bosselaers, René Govaerts, and Joos Vandewalle, *Collision-free hash functions based on block cipher algorithms*, Proceedings 1989 International Carnahan Conference on Security Technology, IEEE, 1989, IEEE catalog number 89CH2774-8, pp. 203–210.
23. Bart Preneel, René Govaerts, and Joos Vandewalle, *Hash functions based on block ciphers: A synthetic approach*, Advances in Cryptology – CRYPTO '93 (Douglas Robert Stinson, ed.), Lecture Notes in Computer Science, vol. 773, Springer-Verlag, 1993, pp. 368–378.

24. Jean-Jacques Quisquater and Marc Girault, *2n-bit hash-functions using n-bit symmetric block cipher algorithms*, Advances in Cryptology – EUROCRYPT '89 (Jean-Jacques Quisquater and Joos Vandewalle, eds.), Lecture Notes in Computer Science, vol. 434, Springer-Verlag, 1989, pp. 102–109.
25. Claude Shannon, *Communication theory of secrecy systems*, Bell System Technical Journal **28** (1949), no. 4, 656–715.
26. David Wagner, *A generalized birthday problem*, Advances in Cryptology – CRYPTO 2002 (Moti Yung, ed.), Lecture Notes in Computer Science, vol. 2442, Springer-Verlag, 2002, pp. 288–303.
27. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu, *Finding collisions in the full SHA-1*, Advances in Cryptology – CRYPTO 2005 (Victor Shoup, ed.), Lecture Notes in Computer Science, vol. 3621, Springer-Verlag, 2005, pp. 17–36.
28. Xiaoyun Wang and Hongbo Yu, *How to break MD5 and other hash functions*, Advances in Cryptology – EUROCRYPT 2005 (Ronald Cramer, ed.), Lecture Notes in Computer Science, vol. 3494, Springer-Verlag, 2005, pp. 19–35.
29. Robert S. Winternitz, *A secure one-way hash function built from DES*, IEEE Symposium on Security and Privacy, 1984, pp. 88–90.

A Proof of Lemmata

Proof of Lemma 1. Consider $x \in \{0,1\}^c \setminus \{0\}$ and two distinct inputs $\boldsymbol{M_1}\|\boldsymbol{H_1}$ and $\boldsymbol{M_2}\|\boldsymbol{H_2}$. By definition of S_x, we have that

$$G_x(\boldsymbol{M}\|\boldsymbol{H}) = \bigoplus_{j \in S_x} f^{(j)}(\boldsymbol{M}\|\boldsymbol{H} \cdot \mathcal{A}_j).$$

The fact that $G_x(\boldsymbol{M_1}\|\boldsymbol{H_1})$ and $G_x(\boldsymbol{M_2}\|\boldsymbol{H_2})$ are uniformly random is obvious because they are linear combination of the uniformly random outputs of the $f^{(j)}$'s. Moreover the internal input blocks of the $f^{(j)}$'s differ in at least one bit for $\boldsymbol{M_1}\|\boldsymbol{H_1}$ and $\boldsymbol{M_2}\|\boldsymbol{H_2}$, otherwise, as $\boldsymbol{M_1}\|\boldsymbol{H_1} \neq \boldsymbol{M_2}\|\boldsymbol{H_2}$ it would be possible to construct a non zero element in $\bigcap_{j \in S_x} \ker \mathcal{A}_j$, which is $\{0\}$ by hypothesis. As the output of the $f^{(j)}$'s are random and independent, $G_x(\boldsymbol{M_1}\|\boldsymbol{H_1})$ and $G_x(\boldsymbol{M_2}\|\boldsymbol{H_2})$ are also independent.

Proof of Lemma 3. As $x \in \{0,1\}^c \setminus \mathrm{Vec}(x_1, \ldots, x_r)$, and as $\mathcal{B}$ has full rank, then necessarily $x \cdot \mathcal{B}^T$ is not a linear combination of the $(x_i \cdot \mathcal{B}^T)_{i \in [1..r]}$. Consequently, there exists $j \in [1..t]$ such that $f^{(j)}$ intervenes in G_x but in none of the $(G_{x_i})_{i \in [1..r]}$. As the outputs of the inner compression functions are independent, $G_x(\boldsymbol{M}\|\boldsymbol{H})$ is independent from $(G_{x_i}(\boldsymbol{M}\|\boldsymbol{H}))_{i \in [1..r]}$. The generalization follows easily by induction.

B Proof of Theorem 3

Let $\mathfrak{A}$ be a collision-finding adversary attacking the compression function h. Instead of working on single external inputs as for the proof of Theorem 1, we will work on pairs of distinct external inputs, but the reasoning will be quite similar and readers are recommended to read the preimage proof before this one. Let $\mathcal{P}_2$ be the set of all 2-elements subsets of $\{0,1\}^{m+c}$. External inputs will be noted $\boldsymbol{X}$ instead of $\boldsymbol{M}\|\boldsymbol{H}$ for concision. Let define Coll as being the set of pairs of distinct external inputs $\{\boldsymbol{X_1}, \boldsymbol{X_2}\}$ which are h-computable with respect to the final set of queries of $\mathfrak{A}$ and which collide under h. As for preimage, it is easy to see that $\Pr[\mathfrak{A} \text{ wins}] \leq 1/2^n + \Pr[\mathrm{Coll} \neq \emptyset]$.
We now bound $\Pr[\mathrm{Coll} \neq \emptyset]$. Given partial sets of queries $\mathcal{Q}'_1, \ldots, \mathcal{Q}'_t \subset (\mathcal{I}_n)^k$, we will say that the pair $\{\boldsymbol{X_1}, \boldsymbol{X_2}\} \in \mathcal{P}_2$ is *compatible* if $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ collide on $V_{\{\boldsymbol{X_1}, \boldsymbol{X_2}\}} = V_{\boldsymbol{X_1}} \cap V_{\boldsymbol{X_2}}$, meaning that for all $x \in V_{\{\boldsymbol{X_1}, \boldsymbol{X_2}\}}$, $G_x(\boldsymbol{X_1}) = G_x(\boldsymbol{X_2})$. Here also it is possible to show

that if $V_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}} \neq \{0\}$ with respect to the final sets of queries of $\mathfrak{A}$, then there is an unique $x \in \{0,1\}^c \setminus \{0\}$ such that $V_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}} = \mathrm{Vec}(x)$ when it becomes strictly bigger than $\{0\}$. We will note $G^1_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}}$ the linear combination of output blocks associated with this x. We define Pot_1 as being the set of all $\{\boldsymbol{X_1},\boldsymbol{X_2}\} \in \mathcal{P}_2$ such that $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ are at least 1-compatible with respect to the final sets of queries of $\mathfrak{A}$ and $G^1_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}}(\boldsymbol{X_1}) = G^1_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}}(\boldsymbol{X_2})$. It is now straightforward to follow the same reasoning as for the preimage proof, which we do without further justification:

$$\begin{aligned}
\Pr[\mathrm{Coll} \neq \emptyset] &\leq \sum_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}} \Pr[\{\boldsymbol{X_1},\boldsymbol{X_2}\} \in \mathrm{Coll}] \\
&\leq \sum_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}} \Pr[\{\boldsymbol{X_1},\boldsymbol{X_2}\} \in \mathrm{Coll} | \{\boldsymbol{X_1},\boldsymbol{X_2}\} \in \mathrm{Pot}_1] \cdot \\
&\qquad\qquad\qquad\qquad \Pr[\{\boldsymbol{X_1},\boldsymbol{X_2}\} \in \mathrm{Pot}_1] \\
&\leq \frac{1}{2^{(c-1)n}} \sum_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}} \Pr[\{\boldsymbol{X_1},\boldsymbol{X_2}\} \in \mathrm{Pot}_i] \\
&\leq \frac{1}{2^{(c-1)n}} \sum_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}} \Pr[G^1_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}}(\boldsymbol{X_1}) = G^1_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}}(\boldsymbol{X_2}) | \boldsymbol{X_1} \text{ and } \boldsymbol{X_2} \text{ are} \\
&\qquad\qquad 1\text{−computable}] \cdot \Pr[\boldsymbol{X_1} \text{ and } \boldsymbol{X_2} \text{ are } 1\text{−computable}] \\
&\leq \frac{1}{2^{(c-1)n}} \frac{1}{2^n} \sum_{\{\boldsymbol{X_1},\boldsymbol{X_2}\}} \Pr[\boldsymbol{X_1} \text{ and } \boldsymbol{X_2} \text{ are } 1\text{−computable}] \\
&\leq \frac{\beta_1(q)^2}{2 \cdot 2^{cn}}.
\end{aligned}$$

Hence the result.

C Why the Reasoning of [20] was Faulty

We'd like to emphasize why the following reasoning, which was the one used in [20] for their security proof for preimage attacks, is tempting but fallacious.
One can surely write (see the proof of Theorem 1 for the notations)

$$\begin{aligned}
\Pr[\mathrm{Preim} \neq \emptyset] &\leq \sum_{\boldsymbol{M}\|\boldsymbol{H}} \Pr[\boldsymbol{M}\|\boldsymbol{H} \in \mathrm{Preim}] \\
&\leq \sum_{\boldsymbol{M}\|\boldsymbol{H}} \Pr[\boldsymbol{M}\|\boldsymbol{H} \text{ is } h\text{-computable} \wedge h^{(f^{(1)},\ldots,f^{(t)})}(\boldsymbol{M}\|\boldsymbol{H}) = \boldsymbol{0}].
\end{aligned}$$

At this stage, it is tempting to claim that the events "$\boldsymbol{M}\|\boldsymbol{H}$ is h-computable" and "$h^{(f^{(1)},\ldots,f^{(t)})}(\boldsymbol{M}\|\boldsymbol{H}) = \boldsymbol{0}$" are independent, thus concluding that

$$\begin{aligned}
\Pr[\mathrm{Preim} \neq \emptyset] &\leq \frac{1}{2^{cn}} \sum_{\boldsymbol{M}\|\boldsymbol{H}} \Pr[\boldsymbol{M}\|\boldsymbol{H} \text{ is } h\text{-computable}] \\
&\leq \frac{1}{2^{cn}} \mathbb{E}(\#\{\boldsymbol{M}\|\boldsymbol{H} \mid \boldsymbol{M}\|\boldsymbol{H} \text{ is } h\text{-computable}\}) \\
&\leq \frac{\beta_c(q)}{2^{cn}}.
\end{aligned}$$

However, this is false because these two events are not independent. Indeed, one can intuitively argue that the fact that $h^{(f^{(1)},\ldots,f^{(t)})}(\boldsymbol{M}\|\boldsymbol{H}) = \boldsymbol{0}$, being detected on one of the

output blocks by the adversary, will increase the probability that $\mathfrak{A}$ makes the queries needed to compute $\boldsymbol{M}\|\boldsymbol{H}$ on other output blocks, thus increasing the probability for $\boldsymbol{M}\|\boldsymbol{H}$ to be h-computable.
The same type of problem arises for collision resistance, where an analogue but still hasty reasoning would conclude that $\Pr[\mathrm{Coll} \neq \emptyset] \leq \frac{\beta_c(q)^2}{2 \cdot 2^{cn}}$.