Beyond Uniformity: Better Security/Efficiency Tradeoffs for Compression Functions

Martijn Stam

martijn.stam@epfl.ch
EPFL, Switzerland

Abstract. Suppose we are given a perfect $n+c$-to-n bit compression function f and we want to construct a larger $m+s$-to-s bit compression function H instead. What level of security, in particular collision resistance, can we expect from H if it makes r calls to f? We conjecture that typically collisions can be found in $2^{(nr+cr-m)/(r+1)}$ queries. This bound is also relevant for building a $m+s$-to-s bit compression function based on a blockcipher with k-bit keys and n-bit blocks: simply set $c=k$, or $c=0$ in case of fixed keys.

We also exhibit a number of (conceptual) compression functions whose collision resistance is close to this bound. In particular, we consider the following four scenarios:

1. A $2n$-to-n bit compression function making two calls to an n-to-n bit primitive, providing collision resistance up to $2^{n/3}/n$ queries. This beats a recent bound by Rogaway and Steinberger that $2^{n/4}$ queries to the underlying random n-to-n bit function suffice to find collisions in any rate-1/2 compression function. In particular, this shows that Rogaway and Steinberger's recent bound of $2^{(nr-m-s/2)/r)}$ queries (for $c=0$) crucially relies upon a uniformity assumption; a blanket generalization to arbitrary compression functions would be incorrect.
2. A $3n$-to-$2n$ bit compression function making a single call to a $3n$-to-n bit primitive, providing collision resistance up to 2^n queries.
3. A $3n$-to-$2n$ bit compression function making two calls to a $2n$-to-n bit primitive, providing collision resistance up to 2^n queries.
4. A single call compression function with parameters satisfying $m \leq n+c, n \leq s, c \leq m$. This result provides a tradeoff between how many bits you can compress for what level of security given a single call to an $n+c$-to-n bit random function.

1 Introduction

Hash function design based on idealized primitives has recently undergone a surge in popularity. One of the earliest approaches is Merkle's use of the ideal cipher model to argue the collision resistance of his double length construction [10]. The use of the ideal cipher model has also been instrumental in proving security properties of single call blockcipher based compression functions by Black, Rogaway and Shrimpton [3]. These 1-call blockcipher based constructions have the disadvantage of rekeying every round, which is expensive.

An alternative is the use of a blockcipher with its key fixed or, slightly relaxed, simply a a random n-to-n bit function. Black, Cochran and Shrimpton [2] show that no compression function can exist making only a single call to a fixed key ideal cipher yet still achieving collision resistance. Indeed, two queries suffice to find a collision with certainty.

Rogaway and Steinberger [16] have recently generalized this result considerably. They consider a compression function that maps $m + s$ bits to s bits using r calls to n-to-n bit random functions.[1] Central to their results is the yield of an adversary, that is the number of compression function evaluations that can be made after q queries. It can be shown that if $q = 2^{(nr-n-s/2)/r}$, a greedy adversary can evaluate the compression function on at least $2^{s/2}$ different inputs. Assuming the corresponding evaluations are uniformly distributed implies a collision can be expected (birthday paradox). Consequently [16, Theorem 2], for any compression function satisfying the uniformity assumption, $2^{(nr-n-s/2)/r}$ queries suffice to find a collision with high probability.

One could argue that any good compression function ought to be 'collision-uniform'. But what happens if the compression function is somehow 'bad' and the assumption does not hold? In the case of standard birthday attacks on compression functions [1], deviation from uniformity only reduces collision resistance and it is tempting to generalize to the current scenario. In any case, in the original[2] interpretation of their results, Rogaway and Steinberger silently drop any mention of the uniformity assumption and seemingly claim that, for *any* compression function, an adversary will be able to find collisions with high probability after only $2^{(nr-n-s/2)/r}$ queries. In particular, this would imply that around $2^{n/4}$ queries would typically suffice to find collisions in a 2-call $2n$-to-n bit compression function (a result also alluded to by Shrimpton and Stam [19]).

We show that this interpretation is *incorrect*. In particular, we demonstrate a 2-call compression function that provably requires around $2^{n/3}/n$ queries to find a collision. A first impression why this might be possible is already contained in the bound on the number of queries required under the uniformity assumption. Indeed, if $2^{(nr-n-s/2)/r}$ queries were required, this would indicate that enlarging the state size s would actually *reduce* the collision resistance.[3] This is clearly incorrect, since one can always just expand the state by keeping part of it fixed, a measure that will not influence collision resistance. Nonetheless, the bound $2^{(nr-n-s/2)/r}$ is useful, since it provides us with a means to

[1] Our notation deviates from theirs; we emphasize the size s of the chaining variable, or state, and the size m of message material to be hashed when the compression function would be Merkle-Damgård iterated.

[2] In a response to an early manuscript of this paper, the phrasing is more accurate in an updated version [14].

[3] This problem is actually less clear from Rogaway and Steinberger's formulation of the bound.

identify the ideal state size for a given rate. Using an ordinary birthday attack would require $2^{s/2}$ queries, intuitively the optimal state size is that for which the yield-based bound coincides with the standard birthday bound. This crossover occurs for $s = 2(nr - m)/(r+1)$, heuristically yielding collision resistance up to $q = 2^{(nr-m)/(r+1)}$ queries, or $q = 2^{n(r-1)/(r+1)}$ assuming $m = n$.

For the aforementioned 2-call compression function, the optimal state size is $2n/3$, implying we could expect collision resistance up to $2^{n/3}$ for a 2-call $2n$-to-n bit compression function. We give a surprisingly simple compression function almost achieving this bound. For 3-call schemes, the optimal state size is n, yielding collision resistance $2^{n/2}$, coinciding with the Rogaway-Steinberger bound. Shrimpton and Stam [18] and Rogaway and Steinberger [15] already gave distinct 3-call $2n$-to-n bit compression functions achieving collision resistance up to almost $2^{n/2}$ queries. For 4-call schemes, the optimal state size is $6n/5$, yielding collision resistance $2^{3n/5}$. Thus a 4-call double length $3n$-to-$2n$ bit function can be expected to be broken within $2^{3n/5}$ queries, not the $2^{n/4}$ queries as reported by Rogaway and Steinberger. In particular, this indicates that one might already achieve more security with a 4-call double length function than what could be achieved with a single length function. However, we do not yet have a construction matching this bound.

One could object to compression functions that are not as uniform when it comes to their collision behaviour. We agree, but only up to a point. At the core of our 2-call $2n$-to-n bit construction is a 2-call $\frac{5}{3}n$-to-$\frac{2}{3}n$ bit compression function, also with collision resistance up to about $2^{n/3}$ queries. This smaller-state compression function is expected to behave collision-uniform. Thus, in this particular case, the choice really is between a collision-uniform compression function outputting n bits and being collision resistant up to $2^{n/4}$ queries on the one hand, and a collision-uniform compression function outputting only $\frac{2}{3}n$ bits yet being collision resistant up to $2^{n/3}$ queries. We believe the latter option is more desirable in practice.

We stress that our work does not contradict or invalidate [16, Theorem 2] in any way; we do show that by dropping uniformity, or rather state size, one can do better. We also point out that many of the bounds obtained by Rogaway and Steinberger do not have uniformity as a premise: for any compression function just slightly over $2^{(nr-m)/r}$ queries are guaranteed to give a collision [16, Theorem 1] and similarly for any hash function that makes on average r calls per message block, $n2^{(nr-n)/r}$ queries will suffice [16, Theorem 3].

We then ask ourselves the question what happens if the underlying primitive already compresses, that is, if we use idealized $n+c$-to-n bit functions instead of n-to-n bit ones as underlying primitive. This question is most interesting if the compression function to be constructed has a larger input size than the idealized primitive (i.e., $m + s > n + c$, cf. the examples above) or outputs more bits (i.e., $s > n$), which is relevant for instance for the construction of double length compression functions. We show that if we build a $m + s$-to-s bit compression function H making r calls to an $n + c$-to-n bit primitive, we can expect to find collision after $2^{(nr+cr-m)/(r+1)}$ queries. We believe this bound to be tight up to some pathological cases, namely when 2^{nr} or $2^{s/2}$ is smaller than said bound. We also prove an upper bound on indifferentiability.

Assuming our conjectured bound can be achieved has interesting implications for the construction of double-length hash functions, where $m = n$ and $s = 2n$. In particular, one call to a $3n$-to-n bit primitive or two calls to a $2n$-to-n bit primitive would suffice to obtain optimal collision resistance in the compression function. This contrasts starkly with earlier approaches, where either more calls needed to be made, or the collision resistance can only (partially) be proven in the iteration [5, 7, 8, 11–13, 17, 20]. For both scenarios we give a construction that we believe offers the required collision resistance, up to a small factor. Against non-adaptive adversaries the proof is surprisingly straightforward; against adaptive adversaries we need a reasonable assumption. Although our compression functions sport impressive collision resistance, they do have some obvious shortcomings when other properties are taken into account. As such, we consider them more a proof of concept—the setting of a bar—than actual proposals to be implemented and used as is. We leave open the problem of designing cryptographically satisfactory compression functions reaching the collision resistance bound provided in this paper.

Finally, we present a general single call construction for the case $m \leq n + c, n \leq s$, and $c \leq m$, achieving collision resistance up to $2^{(n+c-m)/2}$ queries. This provides a tradeoff of how much message bits you can hash and what collision resistance you can expect. It also fills the gap between the impossibility result of Black et al. [2] for $c = 0$ and $m = n$ and the trivial optimally collision resistant solution when $c = n$ and $m = n$.

Notwithstanding the emphasis in this paper on random $n + c$-to-n bit functions, the bounds are also indicative for ideal ciphers with k bit keys and n bit blocks, by setting $c = k$. Fixed-key ideal ciphers correspond to $c = 0$. No constructions in this scenario are presented, although our constructions with the public random functions replaced with ideal ciphers in Davies-Meyer mode are obvious candidates.

Our paper is organized as follows. In Section 2 we introduce notation and recall some relevant results. In Section 3, we discuss upper bounding the probability of finding collisions and, to a lesser extent, preimages in the compression function. Finally, Section 4 consists of four parts, each detailing a construction that (almost) meets the upper bound from its preceding section.

2 Background

2.1 General Notation

For a positive integer n, we write $\{0,1\}^n$ for the set of all bitstrings of length n. When X and Y are strings we write $X \,||\, Y$ to mean their concatenation and $X \oplus Y$ to mean their bitwise exclusive-or (xor). Unless specified otherwise, we will consider bitstrings as elements in the group $(\{0,1\}^n, \oplus)$.

For positive integers m and n, we let $\mathrm{Func}(m,n)$ denote the set of all functions mapping $\{0,1\}^m$ into $\{0,1\}^n$. We write $f \xleftarrow{\$} \mathrm{Func}(m,n)$ to denote random sampling from the set $\mathrm{Func}(m,n)$ and assignment to f. Unless otherwise specified, all finite sets are equipped with a uniform distribution.

2.2 Compression Functions

A *compression function* is a mapping from $\{0,1\}^m \times \{0,1\}^s$ to $\{0,1\}^s$ for some $m, s > 0$. For us, a compression function H must be given by a program that, given (M,V), computes $H^{f_1,\ldots,f_r}(M,V)$ via access to a finite number of specified oracles $f_1,\ldots,f_r$, where we use the convention to write oracles that are provided to an algorithm as superscripts.

Let $f_1,\ldots,f_r$ be random functions from $\{0,1\}^{n+c} \to \{0,1\}^n$. Let C_i : $\{0,1\}^s \times \{0,1\}^m \times (\{0,1\}^n)^i \to \{0,1\}^{n+c}$, for $i = 0,\ldots,r-1$, and C_r : $\{0,1\}^s \times \{0,1\}^m \times (\{0,1\}^n)^r \to \{0,1\}^s$, be processing functions. Compression of a message block then proceeds as follows: Given an s-bit state V and m bit message M, compute output $H = H^{f_1,\ldots,f_r}(M,V)$ by

$$\begin{aligned}
Y_1 &\leftarrow f_1(C_0(M,V)) \\
Y_2 &\leftarrow f_2(C_1(M,V,Y_1)) \\
&\vdots \\
Y_r &\leftarrow f_r(C_{r-1}(M,V,Y_1,\ldots,Y_{r-1})) \\
H &\leftarrow C_r(M,V,Y_1,\ldots,Y_r)
\end{aligned}$$

as illustrated by Figure 1. In particular, we assume the functions $f_1,\ldots,f_r$ are always called in the same, fixed order. Dynamic, input-dependent ordering can to some extent be modelled by only considering the combined function $\tilde{f}(i,x) = f_i(x)$, thus increasing c by $\lg r$.

Normally one defines the rate as the reciprocal of the number of calls made to the underlying primitive. For primitives have differing input sizes this could skew the comparison: we suggest to use $R = m/(rn + rc)$ as rate, that is the number of bits compressed divided by the total number of input bits taken by

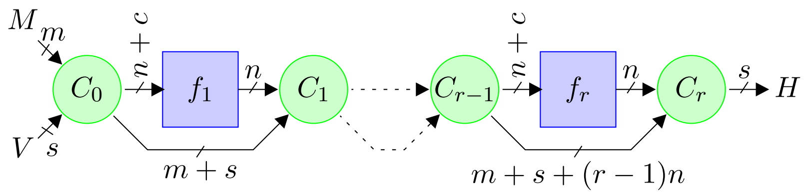

Fig. 1: General form of a $m+s$-to-s bit compression function based on r calls to underlying $n+c$-to-n bit primitive.

the underlying primitives. However, most of the time we will concentrate on the number of calls made, and simply talk of an r-call primitive.

A compression function can be made into a hash function by iterating it. We briefly recall the standard Merkle-Damgård iteration [4, 10], where we assume that there is already some injective padding from $\{0,1\}^* \to (\{0,1\}^m)^*$ in place. Given an initial vector $V_0 \in \{0,1\}^s$ define $\mathcal{H}^H : (\{0,1\}^m)^* \to \{0,1\}^s$ as follows for $\mathbf{M} = (M_1, \ldots, M_\ell)$:

1. Set $V_i \leftarrow H^{f_1,\ldots,f_r}(M_i, V_{i-1})$ for $i = 1, \ldots, \ell$.
2. Output $\mathcal{H}^H(\mathbf{M}) = V_\ell$.

In particular, the hash of the empty message $\mathbf{M} = \emptyset$ corresponds to $\ell = 0$, so $\mathcal{H}^H(\emptyset) = V_0$, the initial vector. Baring this iteration in mind, given a compression function $H : \{0,1\}^m \times \{0,1\}^s \to \{0,1\}^s$ we will refer to the $\{0,1\}^m$ part of the input as 'message' and the $\{0,1\}^s$ as the state. In particular, we refer to s as the state size; increasing the state size will reflect the size of both the input and output of the compression function.

A *collision-finding adversary* is an algorithm with access to one or more oracles, whose goal it is to find collisions in some specified compression or hash function. It is standard practice to consider information-theoretic adversaries only. Currently this seems to provide the only handle to get any provable results. Information-theoretic adversaries are computationally unbounded and their complexity is measured only by the number of queries made to their oracles. Without loss of generality, such adversaries are assumed not to repeat queries to oracles nor to query an oracle outside of its specified domain.

Definition 1 *Let $n, c, m, s > 0$ be integer parameters, and fix an integer $r > 0$. Let $H : \{0,1\}^m \times \{0,1\}^s \to \{0,1\}^s$ be a compression function taking r oracles $f_1, \ldots, f_r : \{0,1\}^{n+c} \to \{0,1\}^n$. Let $\mathcal{A}$ be a collision-finding adversary for H*

that takes r oracles. The collision-finding advantage of $\mathcal{A}$ is defined to be

$$\mathbf{Adv}^{\mathrm{coll}}_{H(n)}(\mathcal{A}) = \Pr\Big[f_1..f_r \xleftarrow{\$} \mathrm{Func}(n+c,n), (M,V),(M',V') \leftarrow \mathcal{A}^{f_1..f_r} : (M,V) \neq (M',V') \text{ and } H^{f_1..f_r}(M,V) = H^{f_1..f_r}(M',V')\Big]$$

Define $\mathbf{Adv}^{\mathrm{coll}}_{H(n)}(q)$ as the maximum advantage over all adversaries making at most q queries to each of their oracles.

Definition 2 *Let $n, c, m, s > 0$ be integer parameters, and fix an integer $r > 0$. Let $H\colon \{0,1\}^m \times \{0,1\}^s \to \{0,1\}^s$ be a compression function taking r oracles $f_1, \ldots, f_r : \{0,1\}^{n+c} \to \{0,1\}^n$. Let $\mathcal{A}$ be a preimage-finding adversary for H that takes r oracles. The preimage-finding advantage of $\mathcal{A}$ is defined to be*

$$\mathbf{Adv}^{\mathrm{preim}}_{H(n)}(\mathcal{A}) = \Pr\Big[f_1..f_r \xleftarrow{\$} \mathrm{Func}(n+c,n), (M,V) \xleftarrow{\$} \{0,1\}^{m+s}, H \leftarrow H^{f_1..f_r}(M,V), (M',V') \leftarrow \mathcal{A}^{f_1..f_r}(H) :\ H = H^{f_1..f_r}(M',V')\Big]$$

Define $\mathbf{Adv}^{\mathrm{preim}}_{H(n)}(q)$ as the maximum advantage over all adversaries making at most q queries in total to their oracles.

For future reference we offer the following little lemma, which basically states that one can increase the state size of a compression function by simply forwarding the extra state bits untouched, without aversely affecting preimage or collision resistance. The lemma is mainly of theoretical use, since simply outputting part of the input is counter to practical hash design.

Lemma 3 *Let m, s, s' be positive integers with $s' > s$. Let $H : \{0,1\}^m \times \{0,1\}^s \to \{0,1\}^s$ be a hash function. Define $H' : \{0,1\}^m \times \{0,1\}^{s'} \to \{0,1\}^{s'}$ by $H'(M, V||V') = (H(M,V)||V')$ where $V \in \{0,1\}^s$ and $V' \in \{0,1\}^{s'-s}$. Then H' inherits its collision resistance and preimage resistance from H, that is, for any adversary $\mathcal{A}'$ on H' there is an adversary $\mathcal{A}$ on H with essentially the same complexity and advantage.*

Proof: We first prove the statement for collision resistance. Let an adversary $\mathcal{A}'$ on the collision resistance of H' be given. Then $\mathcal{A}$ runs $\mathcal{A}'$ and, supposing $\mathcal{A}'$ outputs a collision (M, V, V') and $(\tilde{M}, \tilde{V}, \tilde{V}')$, outputs (M, V) and $(\tilde{M}, \tilde{V})$. Then $H(M, V) = H(\tilde{M}, \tilde{V})$ and $\tilde{V} = \tilde{V}'$. Because $(M, V, V') \neq (\tilde{M}, \tilde{V}, \tilde{V}')$ this implies that $(M, V) \neq (\tilde{M}, \tilde{V})$, making it a collision on H.

The proof for preimage resistance is similar. Let an adversary $\mathcal{A}'$ on the preimage resistance of H' be given and suppose $\mathcal{A}$ needs to find a preimage of

$Z \in \{0,1\}^s$, where Z is distributed by applying H to the uniform distribution over $\{0,1\}^{m+s}$. Then $\mathcal{A}$ randomly selects $V' \in \{0,1\}^{s'-s}$ and runs $\mathcal{A}'$ on input $Z' = (Z||V')$. By construction, Z' is distributed correctly (as if applying H' to the uniform distribution over $\{0,1\}^{m+s'}$), so suppose $\mathcal{A}'$ outputs a preimage (M, V, V'). Then $\mathcal{A}$ outputs (M, V) as preimage of Z under H. *Q.E.D.*

2.3 Collisions in Uniform Samples

With $(\{0,1\}^n)^q$ we denote the set of q-element vectors, or q-*vectors*, in which each element is an n-bit string. When $\mathbf{a} \in (\{0,1\}^n)^q$, we will write $\mathbf{a} = (a_1, \ldots, a_q)$ when we wish to stress its components. We will use U to denote the uniform distribution over $(\{0,1\}^n)^q$ (where n and q will often follow from the context). Thus U corresponds to sampling q strings from $\{0,1\}^n$ uniformly and independently *with* replacement.

If in a random sample some value appears *exactly* k times, we say there is a k-*way collision* in that sample. Let $M_U(k)$ be the random variable describing the number of k-way collisions when the samples are drawn according to the distribution U. We recall the following well known "urns and balls" result [6]. The expected number of k-way collisions is $\mathbb{E}\left[M_U(k)\right] = N\binom{q}{k}\left(\frac{1}{N}\right)^k\left(1 - \frac{1}{N}\right)^{q-k}$, where $N = 2^n$. Thus, $\mathbb{E}[M_U(k)]$ follows a (scaled) binomial distribution with parameters $1/N$ and q. Asymptotically, this would correspond to a scaled Poisson distribution with parameter q/N (provided q/N remains bounded).

The probability of finding any sort of collision is at most $\frac{q^2}{2N}$. We can also bound the probability of finding a k-way collision, see Lemma 4 below. In particular, for $q = 2^n/n$ and $k = n$ the probability $\Pr(M_U(n) > 0] < (2/n)^n$ tends to zero for increasing n and with a little bit more work $\sum_{k \geq n} \Pr[M_U(n) > 0] \leq 2(2/n)^n$, also tending to zero.

Lemma 4 *Let $q, n,$ and k be positive integers. Then* $\Pr[M_U(k) > 0] \leq 2^n(q/2^n)^k$.

3 Upper Bounding Collision and Preimage Resistance

3.1 Introduction

Let us consider an $m+s$-to-s bit compression function that uses one call to each of r independent $n+c$-to-n bit random functions $f_1, \ldots, f_r$. We are interested in what kind of collision respectively preimage resistance we can expect. Before we discuss the main line of attack, we mention two other attacks.

Firstly, since the compression function maps to s-bit strings, we know that collisions can be expected after $2^{s/2}$ queries, whereas preimages will be found after just under 2^s queries. Note that these complexities depend only on the size of the compression functions output and not on the dimensions of the underlying primitive, how often it is called, or how many bits are compressed.

Collisions (and in many cases preimages as well) can often also be found using 2^{nr} queries, essentially by guessing the output of the queries corresponding to a certain H-input. Consider C_r, the final function mapping $m + s + rn$ bits to s bits. To find a collision, evaluate the compression function for a random value. If C_r is balanced, every possible output has 2^{m+rn} preimages, each of $m+s+rn$ bits. Parse into $M \times V \times Y_1 \times \cdots \times Y_r$ and evaluate the compression function on input (M, V). With probability 2^{-nr} the Y_i values will correspond with that of the chosen C_r preimage, resulting in a collision. Since there are 2^{m+rn} preimages we can hope that these all have distinct (M, V), so a collision can be found in 2^{nr} queries. If n is relatively small compared to c, this attack might beat the other two.

For the final and main attack the adversary tries to maximize the number of compression function evaluations it can make given his queries. Shrimpton and Stam [18] call this the yield.

Definition 5 *Let $H^{f_1,\ldots,f_r}$ be a compression function based on a primitives $f_1, \ldots, f_r$. The yield of an adversary after a set of queries to $f_1, \ldots, f_r$, is the number of inputs to H for which he can compute $H^{f_1,\cdots,f_r}$ given the answers to his queries. With* $\mathrm{yield}(q)$ *we denote the maximum expected yield given q queries to each of the oracles $f_1, \ldots, f_r$.*

The central theorem is the following generalization of a result by Rogaway and Steinberger [16], who give the result for $c = 0$ only (Shrimpton and Stam [19] give the result for $c = 0$ and $r = 2$ only).

Theorem 6 *Let $H^{f_1,\ldots,f_r}$ be an $m + s$-to-s bit compression function making one call to each of the $n + c$-to-n bit primitives $f_1, \ldots, f_r$. Then* $\mathrm{yield}(q) \geq 2^{m+s}(q/2^{n+c})^r$.

Proof: Consider the following greedy adversary. Let $0 < i < r$. Suppose that after q queries to each of the oracles $f_1, \ldots, f_i$ the adversary can compute $Y_0, \ldots, Y_i$ for Q_i input pairs $(M, V) \in \{0, 1\}^m \times \{0, 1\}^s$. Since there are 2^{n+c} possible inputs to f_{i+1}, on average for each possible input to f_{i+1} there are $Q_i/2^{n+c}$ inputs to the compression function for which the adversary can compute all intermediate chaining values $Y_0, \ldots, Y_i$. If the adversary queries f_{i+1} on the q values for which he knows most intermediate chaining paths, he will be able to compute Y_{i+1} for at least $Q_i q/2^{n+c}$ values (by the pigeon hole principle). With finite induction and using that $Q_0 = 2^{m+s}$ it follows that this adversary can compute Y_r, and hence the compression output, for at least $2^{m+s}(q/2^{n+c})^r$ values. *Q.E.D.*

3.2 Rogaway and Steinberger's Bounds (Generalized)

Rogaway and Steinberger [16] observe that if $\mathrm{yield}(q) > 2^s$ a collision is guaranteed. Moreover, if $\mathrm{yield}(q) > 2^s$ one expects to be able to find preimages, provided the compression function has sufficiently uniform behaviour.[4] Similarly, if $\mathrm{yield}(q) > 2^{s/2}$ one would expect a collision, again provided sufficiently uniform behaviour. The following formulation captures the loosely stated uniformity assumption, taking into account preimages as well. See [14] for a fine-grained description (also of Theorem 8 with $c = 0$).

Assumption 7 *Let H^f be an $m + s$-to-s bit compression function making r calls to $n + c$-to-n bit primitive f. Let $\mathcal{A}$ be the adversary that optimizes its yield according to the proof of Theorem 6. Then the spread and occurence of collisions for the adversary's compression function evalutions behave as if these* $\mathrm{yield}(q)$ *elements were drawn at random.*

Theorem 8 *(Case $c = 0$ corresponds to [16, Theorem 2]) Let H^f be an $m + s$-to-s bit compression function making r calls to $n + c$-to-n bit primitive f.*

1. *If $q \geq 2^{(nr+cr-m-s/2)/r}$ then* $\mathrm{yield}(q) \geq 2^{s/2}$ *and, under Assumption 7 a collision in H^f can be found with high probability.*
2. *If $q \geq 2^{(nr+cr-m)/r}$ then* $\mathrm{yield}(q) \geq 2^s$ *and a collision in H^f can be found with certainty.*
3. *If $q \geq 2^{(nr+cr-m)/r}$ then* $\mathrm{yield}(q) \geq 2^s$ *and, under Assumption 7 preimages in H^f can be found with high probability.*

Proof: Given the lower bound on the yield (Theorem 6) it suffices to determine those q for which $2^{m+s}(q/2^{n+c})^r \geq 2^{s/2}$ respectively $2^{m+s}(q/2^{n+c})^r \geq 2^s$ holds. *Q.E.D.*

3.3 New Bounds

It is easy to see that Assumption 7 cannot be true for all possible H and examples, for which the greedy adversary does not find collisions within the required amount of queries suggested by Theorem 8, are easy to find (another more efficient adversary might exist). Moreover, upon closer inspection the bound of

[4] The preimage result erroneously omits a uniformity assumption [16, Theorem 4], corrected in [14, Theorem 4].

Lemma 8 has a very strange consequence: increasing the state size reduces the security! (Note that this problem does not exist for the bound on preimage resistance.) This is counterintuitive; indeed, given a $m+s$-to-s bit compression function one can easily increase the state size without affecting collision or preimage resistance by simply forwarding the extra state bits untouched (Lemma 3).

The solution to this problem presents itself naturally: first determine the optimal state size. For this we need to determine for which state size the direct yield-based bound and the generic birthday bound coincide. That is, for which s do we have $2^{s/2} = 2^{(nr+cr-m-s/2)/r}$. Taking logarithms and some simple formula manipulation leads to $s = 2(nr + cr - m)/(r + 1)$, corresponding to collision resistance $q = 2^{(nr+cr-m)/(r+1)}$. All in all this leads us to the following conjecture.

Conjecture 9 *Let H^f be an $m+s$-to-s bit compression function making r calls to $n+c$-to-n bit primitive f. Then collisions can be found for $q \leq 2^{(nr+cr-m)/(r+1)}$.*

The yield can also be used to obtain bounds on the indifferentiability of a construction (we refer to [9] for an introduction to hash function indifferentiability).

Theorem 10 *Let H^f be an $m + s$-to-s bit compression function making r calls to $n + c$-to-n bit primitive f. Then H^f is differentiable with high probability from a random $m + s$-to-s bit function after $q > 2^{n+c}(\frac{nr}{s}2^{n+c-m-s})^{1/(r-1)}$ queries.*

Proof: (Sketch) We claim that if $\mathrm{yield}(q) > nqr/s$ the construction cannot possibly be indifferentiable. Suppose the adversary is communicating with a real $m+s$-to-s bit public random function H and simulated $f_1, \ldots, f_r$. After q calls to each of $f_1, \ldots, f_r$, the adversary has received a total of qrn bits in answers. Yet he can now predict the outcome of H for $\mathrm{yield}(q)$ values, i.e., a total of $\mathrm{yield}(q)s$ completely random bits. The claim follows from incompressibility of completely random bitstrings.

Using Theorem 6 we get differentiability for $2^{m+s}(q/2^{n+c})^r > nqr/s$ or $q > 2^{n+c}(\frac{nr}{s}2^{n+c-m-s})^{1/(r-1)}$. *Q.E.D.*

3.4 Interpretation

In Tables 1 and 2 we look at the maximal attainable security under our conjecture and theorem. Both tables are for non-compressing n-to-n bit underlying public random functions (so $c = 0$), similar to the random permutation setting studied by Rogaway and Steinberger. Their bounds for compression functions

	Collision Conjecture 9 $2^{n(1-2/(r+1))}$	Collision [16, Theorem 2] $2^{n(1-3/(2r))}$	Preimages [16, Theorem 4] $2^{n(1-1/r)}$	Indifferentiable Theorem 10 $\approx 2^{n(1-1/(r-1))}$
$r=2$	$2^{n/3}$	$2^{n/4}$	$2^{n/2}$	2
$r=3$	$2^{n/2}$	$2^{n/2}$	$2^{2n/3}$	$2^{n/2}$
$r=4$			$2^{3n/4}$	$2^{2n/3}$
$r=5$			$2^{4n/5}$	$2^{3n/4}$

Table 1: Security Bounds for Single-Length Constructions ($s = n$). Listed are the approximate number of queries after which a certain property will be broken.

	Collision Conjecture 9 $2^{n(1-2/(r+1))}$	Collision [16, Theorem 2] $2^{n(1-2/r)}$	Preimages [16, Theorem 4] $2^{n(1-1/r)}$	Indifferentiable Theorem 10 $\approx 2^{n(1-2/(r-1))}$
$r=3$	$2^{n/2}$	$2^{n/3}$	$2^{2n/3}$	2
$r=4$	$2^{3n/5}$	$2^{n/2}$	$2^{3n/4}$	$2^{n/3}$
$r=5$	$2^{2n/3}$	$2^{3n/5}$	$2^{4n/5}$	$2^{n/2}$

Table 2: Security Bounds for Double-Length Constructions ($s = 2n$). Listed are the approximate number after which a certain property will be broken.

satisfying the uniformity assumption are included in the tables. Table 1 focuses on single length compression functions, that is $s = n$, and Table 2 focuses on double length compression functions, so $s = 2n$. In both cases $m = n$ bits of message are compressed.

In the interesting cases where our upper bound on collision resistance is higher than Rogaway and Steinbergers, notably $r = 2$ for Table 1 and all of Table 2 we suggest to actually reduce the state size to match the provided collision resistance (e.g., $s = 2n/3$ for $r = 2$ in Table 1). Increasing the state size would either reduce collision resistance or introduce questionable behaviour invalidating the uniformity assumption (cf. Lemma 3).

One can also look at the maximum number of message bits one can hope to compress given a targeted leved of collision resistance and number of calls to the underlying public random function. For a collision resistance level of $2^{n/2}$ queries, Conjecture 9 implies one can hash at most $m \leq (\frac{n}{2} + c)r - \frac{n}{2}$ message bits. For double-length constructions and corresponding target of 2^n queries the number of bits increases to $m \leq cr - n$.

Finally, for $c = 0$ and writing rate $R = m/nr$, the bound can be rewritten as $q \leq 2^{n(1-1/R)(r+1)/r}$ indicating that asymptotically (in r) one can get collision

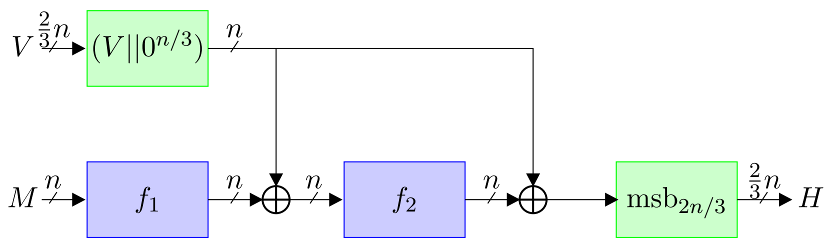

Fig. 2: The 2-call compression function H based on public random n-to-n bit functions f_1 and f_2.

resistance up to $2^{n(1-1/R)}$ queries. Up to constants this is the same as the bound by Rogaway and Steinberger [16, Theorem 3], but an important difference is that their bound is rigorously proven, whereas ours follows from a conjecture.

4 Matching Collision Resistant Constructions

4.1 Case I: A Rate-1/2 Single Length Compression Function

Our main result in this section is a compression function with state size $s = 2n/3$ and almost optimal collision resistance, making only 2 calls to a n-to-n bit public random function. Let $M \in \{0,1\}^n$ and $V \in \{0,1\}^{2n/3}$. Define (Figure 2)

$$H^{f_1,f_2}(M,V) = V \oplus \mathrm{msb}_{2n/3}(f_2((V||0^{n/3}) \oplus f_1(M)))$$

The state size can be expanded by forwarding the extra chaining bits untouched (Lemma 3), giving rise to a $2n$-to-n bit compression function beating the upper bound of $2^{n/4}$ queries for compression functions satisfying the uniformity assumption.

Theorem 11 *Let H^{f_1,f_2} be as given above. Then*

$$\mathbf{Adv}^{\mathrm{coll}}_{H(n)}(q) \le q^2/2^{n+1} + 2^{n/3}(q/2^{n/3})^n + q(q-1)n^2/2^{2n/3} \,.$$

Since the third term is initially dominant, an adversary needs to asks roughly $2^{n/3}/n$ queries for a reasonable advantage.

Proof: Let an adversary $\mathcal{A}$ be given. We need to bound its success probability. Suppose that, whenever $\mathcal{A}$ makes a query to f_1, we do not answer with f_1's answer, but with a randomly drawn value in $\{0,1\}^n$. From the adversary's point of view, this is the same distribution. What's more, we can even decide upon a list of answers to the adversary's f_1 queries ahead of time. Since the input to f_1 is not used elsewhere in the compression function and f_2 is independent of f_1, we could even give this list to the adversary in advance of any query.

Thus we can replace the q queries to f_1 by a list $\mathbf{a}$ of q values drawn at random with replacement from $\{0,1\}^n$, given to the adversary before any queries to f_2 are made. Let us consider the probability that the adversary finds a collision on the i-th query to f_2. Let X_i be the value queried by the adversary. This will allow the adversary to evaluate H for those $a \in \mathbf{a}$ whose first $n/3$ bits coincide with X_i. Let's call this number k_i. Moreover, unless two values $a, a' \in \mathbf{a}$ are identical, this cannot lead to a collision based on the i'th f_2-query alone. Note that collisions in $\mathbf{a}$ occur with probability $\leq q^2/2^{n+1}$.

Suppose that after $i-1$ queries the adversary can evaluate H for Q_{i-1} different values. Then the probability of a collision is on the i-th query is at most $k_i Q_{i-1}/2^{2n/3}$. Note that $Q_i = \sum_{j=1}^{i-1} k_j$. With probability at most $2^{n/3}(q/2^{n/3})^n$ some n-way collision occurs (Lemma 4), otherwise all $k_i < n$ and $Q_i < (i-1)n$. Thus the probability of a collision (provided no n-way collisions occur in the $n/3$ upper most bits of $\mathbf{a}$) is upper bounded by $\sum_{i=1}^{q}(i-1)n^2/2^{2n/3} = q(q-1)n^2/2^{2n/3}$. *Q.E.D.*

As an aside, our construction shares some of the disadvantages of the rate-1/3 construction by Shrimpton and Stam [19]. In particular, finding a collision in f_1 leads to many collisions in H (although the gap between finding a single collision in H and one in f_1 is significantly bigger this time). Implementing f_1 using a fixed key ideal cipher in Davies-Meyer mode does not affect the security (PPP Switching Lemma [18, Lemma 6]), but the effect of replacing f_2 with a fixed-key ideal cipher is less clear.

4.2 Case II: A Single-Call Double Length Compression Function

We will now consider a $3n$-to-$2n$ bit compression function based on a single call to a random $3n$-to-n bit function. We show that there exists a compression function for which the number of random function queries to find a collision is of the order 2^n for non-adaptive adversaries, that need to commit to all their queries in advance of receiving any answers. Subsequently we indicate why we expect the advantage not to drop significantly when taking into account adaptive adversaries and discuss a variant based on two random $2n$-to-n bit functions.

Let us first define the hash function. For ease of exposition, we consider the hash function to have three n-bit inputs U, V, and W. Moreover, we will interpret n-bit strings as elements in $\mathbb{F}_{2^n}$. The input U, V, W is then used to define a quadratic polynomial over $\mathbb{F}_{2^n}$. The hash consists of the output of f (on

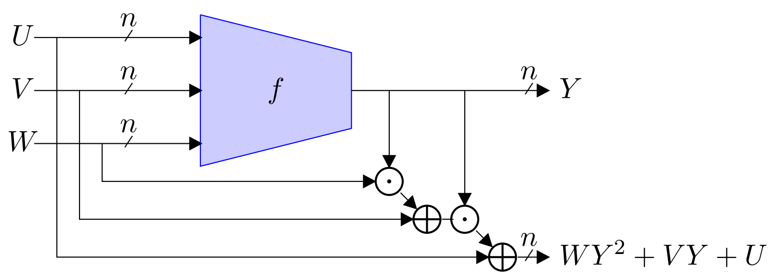

Fig. 3: A single call double length compression function with close to optimal collision resistance. Arithmetic over $\mathbb{F}_{2^n}$.

input U, V, W) and the polynomial evaluated in this output. In other words, to compute $H^f(U, V, W)$ do the folowing (and see Figure 3):

1. Set $Y \leftarrow f(U, V, W)$
2. Output $H^f(U, V, W) = (Y || WY^2 + VY + U)$.

Theorem 12 *Let H^f be as given above. Then*

$$\mathbf{Adv}^{\mathrm{coll}}_{H(n)}(q) \leq q(q-1)/2^{2n} ,$$

for non-adaptive adversaries. Hence a non-adaptive adversary needs to ask roughly 2^n queries for a reasonable advantage.

Proof: Suppose an adversary wants to find a collision (U, V, W) and (U', V', W'). Let $Y = f(U, V, W)$ and $Y' = f(U', V', W')$. Then we need that $Y = Y'$ and $(W - W')Y^2 + (V - V')Y + (U - U') = 0$. Since $(U, V, W) \neq (U', V', W')$ for a collision, this means that Y needs to be a root of a non-zero quadratic polynomial. There are at most two roots over $\mathbb{F}_{2^n}$. The probability that both $f(U, V, W)$ and $f(U', V', W')$ evaluate to the same root is at most $2/2^{2n}$. Since we assume a non-adaptive adversary, we can use a union bound over all pairs (U, V, W) and (U', V', W') to obtain an upper bound on the adversary finding a collision of $\binom{q}{2} 2/2^{2n} = q(q-1)/2^{2n}$. *Q.E.D.*

The question that remains to be answered is what happens if the adversary is adaptive. In this case, given a list of queries (U_j, V_j, W_j) with answers Y_j for $j = 1, \ldots, i-1$ with $i < q$, the adversary will already know when he makes a new query (U_i, V_i, W_i) for which $0 < j < i$ it holds that Y_j satisfies $(U_i - U_j)Y_j^2 + (V_i - V_j)Y_j + (W_i - W_j) = 0$. He finds a collision if Y_i

will equal one of those Y_j, so he optimizes his probability querying (U_i, V_i, W_i) that maximizes the number of $0 < j < i$ for which this query could result in a collision. Suppose there are several j's we target, let's say $j_1, \ldots, j_\ell$. Then one can express W_i as a function of $(U_j, V_j, W_j, Y_j), U_i$, and V_i for any of the j's. Because W_i is unique, one can subsequently express V_i in U_i and any two of the (U_j, V_j, W_j, Y_j)'s. Taking this one step further leads to U_i expressed in any triple of the (U_j, V_j, W_j, Y_j). For $\ell > 3$ the corresponding (U_j, V_j, W_j, Y_j) already need to satisfy an increasing (in ℓ) number of conditions. Since the Y_j are random, we believe one can upper bound the probability p_ℓ of an ℓ-tuple occurring satisfying these conditions, leading to an overall bound on the collision probability of $p_\ell + \sum_{i=1}^{q} \ell/2^n = p_\ell + q\ell/2^n$.

In the full version we show that if one instantiates the $3n$-to-n bit function f with a cascade of two random $2n$-to-n bit functions f_1 and f_2 (i.e., $f(U, V, W) = f_2(f_1(U, V), W)$) the construction remains secure. For any quasi-adaptive adversary that provides a list of queries to f_1 and, only after receiving the answers to this list, will prepare a list of queries to f_2, we bound the advantage

$$\mathbf{Adv}^{\mathrm{coll}}_{H(n)}(q) \leq n^2 q^2 / 2^{2n} + (2q/2^n)^n \ ,$$

so again $2^n/n$ queries are needed for a reasonable advantage.

4.3 Case III: Single Call Efficiency/Security Tradeoff

In this section we show that our conjectured upper bound can be achieved for single call schemes when $m \leq n + c$, $n \leq s$ and $c \leq m$, achieving collision resistance up to $2^{(n-c+m)/2}$ queries. This parameter set might seem a bit artificial at first, but it neatly fills the gap between Black et al.'s impossibility result of creating a $2n$-to-n bit compression function using a single n-to-n bit random function and the trivial construction of making a $2n$-to-n bit compression function based on a single call to a random $2n$-to-n bit function. Indeed, we have $n + c \leq m + s$ in this case, so unless the equality holds, we need to compress before calling f.

For the moment we concentrate on the case $s \leq n + c$, we deal with $s > n + c$ at the end of this section. Recall that the hash function $H^f(M, V)$ has two inputs, M and V and a single oracle f. Split $V = V_0 || V_1$ in two where $|V_0| = n + c - m$ and $|V_1| = m + s - n - c$. Split $M = M_0 || M_1$ in two where $|M_0| = n + c - s$ and $|M_1| = m + s - n - c$. Finally split f's output $F = F_0 || F_1$ in two where $|F_0| = n + c - m$ and $|F_1| = m - c$. Then define

$$C_0(M, V) = (M_0 || M_1 \oplus V_1 || V_0)$$
$$C_1(M, V, F) = (F_0 || (F_1 || 0^{s-n}) \oplus V_1)$$

Theorem 13 *Let H^f be as given above. Then*

$$\mathbf{Adv}^{\mathrm{coll}}_{H(n)}(q) \leq q^2/2^{n+c-m+1} .$$

Hence an adversary needs to ask roughly $2^{(n+c-m)/2}$ queries for a reasonable advantage.

Proof: Suppose an adversary wants to find a collision (M, V) and (M', V') on H^f. Let $X = C_0(M, V), F = f(X)$, and $H = C_1(M, V, F)$. Similar for X', F', and H'. A collision means $(M, V) \neq (M', V')$ yet $H = H'$. A priori, there are two possibilities in such a case. Either $X = X'$ or not. If $X = X'$ then also $F = F'$ and in particular $F_1 = F_1'$. Since $H = H'$ implies $F_1 \oplus V_1 = F_1' \oplus V_1'$, we get $V_1 = V_1'$. This combined with $X = X'$ would already imply $(M, V) = (M', V')$, hence no collision.

Thus we are left with the case $X \neq X'$. In that case $H = H'$ requires $F_0 = F_0'$. This is outside the adversary's control, his advantage after q queries follows the birthday bound based on F_0's length, so is upper bounded by $\frac{1}{2}q^2/|F_0| = q^2/2^{n+c+1-m}$, so roughly $|F_0|^{1/2}$ queries are needed to find a collision for F_0. *Q.E.D.*

Note that, in the proof above, once an F_0-collision has been obtained, one can pick V_1 freely and (uniquely) complete the collision given F_1, F_1', X and X'. Thus the adversary needs about $2^{(n+c-m)/2}$ queries to find his first collision, but it will immediately be a $2^{m+s-n-c}$-way collision.

If $s > n + c$ this multicollision behaviour can be changed slightly. In that case one can simply feed forward $s-n-c$ bits of the state to the next without any processing (and apply the construction above on the $s' = n + c$ remaining bits). Finding a collision will still take time $2^{(n+c-m)/2}$, but this time the adversary will find 2^{s-n-c} collisions that are 2^m-way.

5 Conclusion

Thanks to Phil Rogaway, John Steinberger, Stefano Tessaro and the anonymous Crypto'08 referees for their valuable feedback. Thanks to the folks in Bristol for their hospitality during a vital phase in the writing of this paper and special thanks to Tom Shrimpton for great discussions and feedback on this work from the initial stages to the end.

References

1. M. Bellare and T. Kohno. Hash function balance and its impact on birthday attacks. In *Advances in Cryptology – EUROCRYPT '04*, volume 3027 of *LNCS*, pages 401–418. Springer-Verlag, 2004.

2. J. Black, M. Cochran, and T. Shrimpton. On the impossibility of highly efficient blockcipher-based hash functions. In *Advances in Cryptology – EUROCRYPT '05*, volume 3494 of *LNCS*, pages 526–541. Springer-Verlag, 2005.
3. J. Black, P. Rogaway, and T. Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from PGV. In *Advances in Cryptology – CRYPTO '02*, volume 2442 of *LNCS*. Springer-Verlag, 2002.
4. I. Damgård. A design principle for hash functions. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89*, volume 435 of *LNCS*. Springer-Verlag, 1990.
5. S. Hirose. Some plausible constructions of double-length hash functions. In *FSE'06*, volume 4047 of *LNCS*, pages 210–225. Springer-Verlag, 2006.
6. N. L. Johnson and S. Kotz. *Urn Models and Their Applications*. John Wiley and Sons, Inc., 1977.
7. L. Knudsen and F. Muller. Some attacks against a double length hash proposal. In *Advances in Cryptology – ASIACRYPT'06*, volume 4284 of *LNCS*, pages 462–473. Springer-Verlag, 2006.
8. S. Lucks. A collision-resistant rate-1 double-block-length hash function. In E. Biham, H. Handschuh, S. Lucks, and V. Rijmen, editors, *Symmetric Cryptography*, number 07021 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
9. U. Maurer and S. Tessaro. Domain extension of public random functions: Beyond the birthday barrier. In *Advances in Cryptology – CRYPTO '07*, volume 4622 of *LNCS*, pages 187–204. Springer-Verlag, 2007.
10. R. Merkle. One way hash functions and DES. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89*, volume 435 of *LNCS*, pages 428–466. Springer-Verlag, 1990.
11. I. Mironov and A. Narayanan. Domain extension for random oracles: Beyond the birthday-paradox bound. ECRYPT Hash Workshop 2007, May 24–25, Barcelona, 2007.
12. M. Nandi, W. Lee, K. Sakurai, and S. Lee. Security analysis of a 2/3-rate double length compression function in black-box model. In *Fast Software Encryption – FSE'05*, volume 3557 of *LNCS*, pages 243–254. Springer-Verlag, 2005.
13. T. Peyrin, H. Gilbert, F. Muller, and M. Robshaw. Combining compression functions and block cipher-based hash functions. In *Advances in Cryptology – ASIACRYPT'06*, volume 4284 of *LNCS*, pages 315–331. Springer-Verlag, 2006.
14. P. Rogaway and J. Steinberger. Security/efficiency tradeoffs for permutation-based hashing. Full version of [16], available through authors' website.
15. P. Rogaway and J. Steinberger. Constructing cryptographic hash functions from fixed-key blockciphers. In *Advances in Cryptology – CRYPTO '08*, volume This Proceedings of *LNCS*. Springer-Verlag, 2008.
16. P. Rogaway and J. Steinberger. Security/efficiency tradeoffs for permutation-based hashing. In *Advances in Cryptology – EUROCRYPT '08*, volume 4965 of *LNCS*, pages 220–236. Springer-Verlag, 2008.
17. Y. Seurin and T. Peyrin. Security analysis of constructions combining FIL random oracles. In *Fast Software Encryption (FSE'07)*, volume 4593 of *LNCS*, pages 119–136. Springer-Verlag, 2007.
18. T. Shrimpton and M. Stam. Efficient collision-resistant hashing from fixed-length random oracles. ECRYPT Hash Workshop 2007, May 24–25, Barcelona, 2007.
19. T. Shrimpton and M. Stam. Building a collision-resistant compression function from non-compressing primitives. In *ICALP 2008, Part II*, volume 5126, pages 643–654. Springer-Verlag, 2008. Supersedes [18].
20. J. Steinberger. The collision intractability of MDC-2 in the ideal-cipher model. In *Advances in Cryptology – EUROCRYPT'07*, volume 4515 of *LNCS*, pages 34–51. Springer-Verlag, 2007.