A Collision-Resistant Rate-1 Double-Block-Length Hash Function

Stefan Lucks

University of Mannheim, Germany
http://th.informatik.uni-mannheim.de/people/lucks/
(on the leave to Bauhaus-University Weimar, Germany)

Abstract. This paper proposes a construction for collision resistant $2n$-bit hash functions, based on n-bit block ciphers with $2n$-bit keys. The construction is analysed in the ideal cipher model; for $n = 128$ an adversary would need roughly 2^{122} units of time to find a collision. The construction employs "combinatorial" hashing as an underlying building block (like Universal Hashing for cryptographic message authentication by Wegman and Carter). The construction runs at rate 1, thus improving on a similar rate 1/2 approach by Hirose (FSE 2006).

1 Introduction

The design of block cipher based hash functions has a long tradition in cryptography, see, e.g., [22] and references therein. Recently, many dedicated hash functions, including those most common in practical applications, have been broken [3, 25–28]). Hash functions based on a well-understood "unbroken" block cipher such as the AES are a possible alternative to dedicated hash functions, quickly available and easy to implement. Unfortunately, most block cipher based hash functions are single-block-length constructions, i.e., block cipher and hash size are the same. Good block cipher candidates with a block size of more than 128 bit are still missing. Nowadays, attacking any 128-bit hash is demanding but feasible. Therefore, we deal with double-block-length (in short: DBL) hash functions. Like Hirose [11], we employ an n-bit block cipher with a $2n$-bit key to define a $2n$-bit hash function. Our hash function runs at rate 1, instead of rate 1/2, as the hash function from [11].

For block cipher based hashing, there is no secret key under which the block cipher could operate. Instead the adversary has full control over all block cipher inputs, including the key. The common approach to deal with this (cf. [5]), which we follow in the current paper, is to model the block cipher as an ideal one. If we instantiate the hash function with some real block cipher and an attack occurs, this should be regarded as a block cipher weakness. [1]

Plain Iterated Hash Functions (PIHFs). Standard security requirements for cryptographic hashing are (2nd) preimage and collision resistance. Nowadays, most practical

[1] Black [6] designed an artificial hash function provably secure in the ideal cipher model, but insecure when instantiated by any real block cipher. Nevertheless, proofs of security in the ideal cipher model are still a strong indication for secure block cipher based hashing: When looking at the hash function in [6], it is obvious that it has been designed for the purpose of being insecure. And "no scheme proven secure in the ideal-cipher model has been broken after instantiation, unless this was the goal from the start" [6].

Dagstuhl Seminar Proceedings 07021
Symmetric Cryptography
http://drops.dagstuhl.de/opus/volltexte/2007/1017

hash functions are *plain iterated hash functions* (PIHFs), iterating an underlying *compression function* with fixed-length inputs. We will describe PIHFs in Section 2. As shown by Merkle and Damgård [20, 9], PIHFs are secure in the classical sense if the compression function is secure in the same sense. On the other hand, PIHFs still have undesirable properties, such as length-extendibility: Given the hash $h(M)$ of an unknown message M (but knowing the length of M), it is easy to choose some X and compute $h(M||X)$. Recently, further advanced attacks against the plain iterated hash structure have been proposed [12, 14, 13].

The Merkle-Damgård result indicated that the property of being collision resistant is preserved by the PIHF construction. On the other hand, even if we model the underlying compression function as a random oracle (for fixed-sized inputs), its plain iteration does not behave like a random oracle (for variably-sized inputs). E.g., a random oracle would not suffer from length-extendibility.

Recent Modifications of the PIHF Construction. For many applications, the structure of PIHFs appears to be insufficient. In 2005, Coron et al [8] proposed modified iterated constructions preserving the property of behaving like an ideal primitive. I.e., if we model the underlying primitive as a fixed-size random oracle or as an ideal block cipher, the constructions proposed in [8] behave like a random oracle for arbitrary input sizes. The same year, Lucks [17] proposed variations of the PIHF construction with some fall-back property to defend against compression function weaknesses. Lucks' "failure-friendly" variations of PIHFs do not suffer from length-extendibility. Further, if the compression function fails collision resistance, without failing too badly, the constructions from [17] still defends against advanced attacks and exploits.

In 2006, Liskov [16] proposed the "zipper hash", an extension of the PIHF designed to be both failure-friendly and to behave like a random oracle if the underlying compression function is modelled as a random oracle, thus uniting ideas from [8] and [17]. In a similar spirit, Bellare and Ristenpart [2] proposed the "Enveloped Merkle-Damgård" (EMD) structure – another variation of the PIHF scheme. If we model the compression function as a random oracle, EMD behaves like a random oracle. If the compression function fails to be a random oracle, but still is collision resistant, the the EMD hash function may fail to behave like a random oracle, but falls back to still being collision resistant. (As demonstrated in [2], none of the constructions from [8] provides such a fall-back property.)

Motivation and Contribution. There appears to be a consensus among researchers, that good general-purpose hash functions should provide more than just collision resistance, and that PIHFs by themselves are insufficient for the next generation of general-purpose hash functions. On the other hand, many proposals for advanced hash functions with an advanced security assurance are built upon PIHFs, employing and extending the construction, not abandoning it. Thus, collision resistant PIHFs seem to remain in the toolbox of secret-key cryptography, as special-purpose hash functions and as a building block for advanced hash functions. The current paper proposes block cipher based PIHFs, and analyses their security against classical attacks, with a focus on collision resistance. We present a generic

construction for a $2n$-bit hash function, using any n-bit block cipher with $2n$-bit keys. In the ideal cipher model, we prove the security of our hash function, similarly to Black, Rogaway and Shrimpton [5] for several single-block-length hash functions.

The so far most advanced construction for double-block-length (DBL) hash functions, employing an n-bit block cipher with $2n$-bit keys, has been proposed by Hirose [11]. (See also [11] for further references on DBL hash functions.) His hash function runs at rate 1/2, i.e., for hashing some nR-bit message, one has to call the underlying block cipher $2R$ times. Up to a small factor, his construction is as collision resistant as an ideal hash function (see Section 5.3). This leaves an obvious open problem: *Is it possible to construct such a hash function, running at rate 1?* The current paper give a positive answer.[2]

We also continue a line of research to employ combinatorial hash functions for secret-key cryptography, e.g. in some rounds of a Luby-Rackoff cipher [18, 21]. The Universal Hashing approach from Wegman and Carter [29] initialised an evolution of message authentication codes (MACs) which nowadays are much more efficient in software (when running on current desktop PCs and servers) than their block cipher based counterparts [4, 10, 7, 1]. In a similar spirit, the CWC authenticated encryption scheme combines block cipher based encryption with Carter-Wegman based authentication [15], optimised for good efficiency on dedicated hardware. But note that all the abovely mentioned constructions deal with cryptosystems operating under a secret key, unknown to the adversary. Hash functions are different: There is no secret key. Instead, the adversary controls *all* the inputs. To this end, our construction extends this line of research. (Internally, we use the output of a combinatorial hash function as a block cipher key, and the output of a block cipher call as the "key" for a combinatorial hash function.) We anticipate our paper to inspire further research on software-efficient cryptographic hash functions, perhaps like the evolution of highly efficient MACs.

2 Iterated Hash Functions and Standard Attacks

Standard Attacks. A hash function $h : \{0,1\}^* \to \{0,1\}^n$ should be *collision resistant*: finding messages $M \neq M'$ with $h(M) = h(M')$ should be infeasible. Similarly, both *preimage* attacks (given an image $T \in \{0,1\}^n$, find a preimage M with $h(M) = T$) and *2nd preimage* attacks (given a 1st preimage M', find a 2nd preimage $M'' \neq M'$ with $h(M'') = h(M')$) should be infeasible. As there are more preimages than images, collisions always exist. Thus, for any given h one can efficiently output a "hardwired" collision. Current cryptographic theory thus considers *families* of hash functions $\{0,1\}^* \to \{0,1\}^n$. The "key" (a pointer into the family to uniquely identify the hash function) is part of the adversary's input. This mismatches cryptographic practice, with its fixed hash functions. Recently, Rogaway proposed a more practical "unkeyed" formal model for hash functions – assuming "human ignorance", i.e., no adversary being able to provide a collision [23].

[2] Theoretically, we could claim our hash function to be twice as fast as Hirose's. In practice, the performance benefit of our construction will be a bit smaller. Firstly, Hirose's scheme always makes two block cipher calls under the same key, thus allowing an optimised implementation to skip every second key schedule. Secondly, apart from calling an n-bit block cipher, the instantiations we propose need to multiply over $\mathrm{GF}(2^n)$. This can be done very efficiently in dedicated hardware, but may be time-consuming in software.

Our setting is "unkeyed", as well. In contrast to Rogaway's standard model based approach, we work in the ideal cipher model, where the adversary is initially ignorant about the underlying block cipher and step-wise gains some knowledge about it by querying an oracle. Accordingly, we will prove that *finding a collision* (or a preimage or a 2nd preimage) *for our hash function would require making an overwhelming number of oracle queries.*

Plain Iterated Hash Functions (PIHFs). Write $\langle \text{CHN} \rangle$ for the set of intermediate and final hash values, and $\langle \text{BLK} \rangle$ for the set of message blocks. A message $M = (M_1, M_2 \ldots) \in \langle \text{BLK} \rangle^+$ consists of any nonzero number of blocks $M_i \in \langle \text{BLK} \rangle$. For hashing messages in $\{0,1\}^*$, we assume a non-ambiguous padding method [19, Section 9.3.3], without caring about details.

Consider a fixed-size hash function $c : \langle \text{CHN} \rangle \times \langle \text{BLK} \rangle \to \langle \text{CHN} \rangle$ (a "compression function"). By fixing an "initial value" $H_0 \in \langle \text{CHN} \rangle$ and iterating c to compute "chaining values" $H_i = c(H_{i-1}, M_i)$, we define a PIHF $c^*_{H_0} : \langle \text{BLK} \rangle^+ \to \langle \text{CHN} \rangle$:

$$c^*_{H_0}(M_1, \ldots, M_L) = H_L = c(\ldots c(c(H_0, M_1), M_2), \ldots, M_L).$$

The Merkle-Damgård Result [20, 9]. Assume no message being the postfix of another (longer) message. (E.g., apply the "Merkle-Damgård strengthening" [19, Remark 9.32].) If c is collision resistant, then so is the PIHF $c^*_{H_0}$, for all initial values H_0.

Block Ciphers and the Ideal Cipher Model. We consider a function $E : \langle \text{CHN} \rangle \times \langle \text{BLK} \rangle \to \langle \text{BLK} \rangle$ as a block cipher with key space $\langle \text{CHN} \rangle$, if, for all $K \in \langle \text{CHN} \rangle$, the function $E_K(\cdot) = E(K, \cdot)$ permutes over $\langle \text{BLK} \rangle$, and if additionally both E_K and its inverse E_K^{-1} can be computed efficiently. In the *ideal cipher model*, the E_K are independent uniformly distributed random permutations, and both the E and E^{-1} are simulated by oracles.

Set $N = |\langle \text{BLK} \rangle|$. As this paper deals with "double block length (DBL) hash functions", we focus on $|\langle \text{CHN} \rangle| \approx N^2$.

Assumptions on the Adversary. As the hash functions in this paper are block cipher based, we focus on counting the number of oracle queries as a lower bound for the adversarial running time. We make the following assumption on the adversarial behaviour:

- No redundant queries: After asking for $y = E_K(x)$, the adversary will neither ask for $x = E_K^{-1}(y)$, nor will it again ask for $y = E_K(x)$. Similarly after asking for $x = E_K^{-1}(y)$.
- Query completeness: Before writing its output, all non-redundant oracle queries are made, to actually compute $H(M)$ (and $H(M')$) and all the chaining values in between.

Neither assumption leads to unreasonable results on the adversary's running time. Any adversary can avoid asking any redundant queries without making any additional (non-redundant) queries. And standard adversaries are about finding collisions or (2nd) preimages, thus outputting either one preimage $M = (M_1, \ldots, M_R)$ or two colliding messages

$M = (M_1, \ldots, M_R)$ and $M = (M'_1, \ldots, M'_{R'})$. Therefore, any standard adversary can be transformed to respect query completeness by asking at most R (or $R + R'$) extra queries. This is proportional to the time for writing the output.

3 A PIHF Secure Against Standard Attacks

In this section, we present a compression function $b = b_{f,E}$ and analyse the security of its plain iteration $b^*_{H_0}$ against standard attacks.

3.1 The Compression Function $b = b_{f,E}$

Consider a function $f : \langle \text{CHN} \rangle \times \langle \text{BLK} \rangle \to \langle \text{CHN} \rangle$. For $x \in \langle \text{BLK} \rangle$ we write $f_x(\cdot) = f(\cdot, x)$. This will be our "combinatorial hash function", satisfying the following properties:

Invertibility: For all $x \in \langle \text{BLK} \rangle$, the function $f_x : \langle \text{CHN} \rangle \to \langle \text{CHN} \rangle$ is invertible with the inversion f_x^{-1}, i.e., $f_x^{-1}(f_x(H)) = H$ for all $H \in \langle \text{CHN} \rangle$.
Uniqueness: For all $(S, C) \in \langle \text{CHN} \rangle^2$, there exists at most one $y \in \langle \text{BLK} \rangle$ with $f_y(S) = C$.
Collision universality (with parameter β)**:** For all pairs $(S, S') \in \langle \text{CHN} \rangle^2$ with $S \neq S'$, there exist at most β pairs $(y, y') \in \langle \text{BLK} \rangle^2$ with $f_y(S) = f_{y'}(S')$.

In Section 5, we propose efficient instantiations of f, satisfying these requirements without making any unproven assumptions. For all x, the function f_x^{-1} must exist, but we do not need to compute it. For our proposed instantiations of f, computing f_x^{-1} actually is feasible. But, given an efficient instantiation of f for which computing f_x^{-1} is infeasible, one could likely improve some of our security results.

Let $E : \langle \text{CHN} \rangle \times \langle \text{BLK} \rangle \to \langle \text{BLK} \rangle$ be a block cipher with message space $\langle \text{BLK} \rangle$ and key space $\langle \text{CHN} \rangle$. For $S \in \langle \text{CHN} \rangle$ and $z \in \langle \text{BLK} \rangle$ we write $E_S(z)$ for the encryption of z and $E_S^{-1}(y)$ for its decryption. Our *compression function* (see also Figure 1) is

$$b_{f,E} : \langle \text{CHN} \rangle \times \langle \text{BLK} \rangle \to \langle \text{CHN} \rangle : \quad b_{f,E}(S, y) = f(\overbrace{f(H_{i-1}, x_i)}^{S_i}, \underbrace{E_{\overbrace{f(H_{i-1}, x_i)}^{S_i}}(x_i)}_{y_i}).$$

Computation of $b_{f,E}$:

0. Input: $H_{i-1} \in \langle \text{CHN} \rangle$ and message bock $x_i \in \langle \text{BLK} \rangle$.
1. $S_i := f(H_{i-1}, x_i)$.
2. $y_i := E_{S_i}(x_i)$.
3. Output: $H_i := f(S_i, y_i) \in \langle \text{CHN} \rangle$.

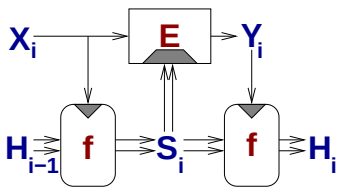

Fig. 1. The Basic Compression function $b_{f,E}$

If f and E are obvious from context, we write b instead of $b_{f,E}$: Observe that b is neither preimage nor 2nd preimage resistant (and thus not collision resistant, either): Let a target $H' \in \langle \text{CHN} \rangle$ be given. Recall that we did not assume computing f^{-1} to be infeasible. Computing E^{-1} is feasible, anyway. Thus we can run b backwards. *Choose any* $y \in \langle \text{BLK} \rangle$, *compute* $S = f_y^{-1}(C)$, $x = E_S^{-1}(y)$ *and output* $H = f_x^{-1}(S)$. *Now* $b(H, x) = H'$.

3.2 The Plain Iteration $b^*_{H_0}$ of b and its Security

As b itself lacks collision resistance, we cannot use the Merkle-Damgård result to show the collision resistance of $b^*_{H_0}$. Our approach to prove the collision resistance of $b^*_{H_0}$ resembles [5], who also prove the iteration of certain invertible compression functions to be collison resistant. (Note that [5] only deal with single block length hash functions, though.)

Theorem 1 (Proof deferred to Section 4). *Choose a random* H_0 *in* $\langle \text{CHN} \rangle$, *and model* E *as an ideal cipher. Let* f *be invertible, unique and collision universal with parameter* β, *as defined above. Set* $b = b_{f,E}$ *and consider the plain iteration* $b^*_{H_0}$ *of* b.

For all adversaries making at most N/K *oracle queries and all natural numbers* L, *the probability* $\Pr[\mathit{coll}(b^*_{H_0})]$ *of finding a collision for* $b^*_{H_0}$ *is*

$$\Pr[\mathit{coll}(b^*_{H_0})] < \frac{|\langle \text{CHN} \rangle| \cdot \beta^L}{L! \cdot (K-1)^L} + \frac{L}{K-1}. \tag{1}$$

Discussion. Our bound on the security of $b^*_{H_0}$ is not very revealing. There is an all-quantified natural number L, and there is a function f for which we only demand certain combinatorial properties.

So how do we choose f? How do we choose L to minimise K (i.e., to maximise the number of oracle queries allowed)? In Section 5, we will assume a 128-bit block cipher E with 256-bit keys (i.e., $N = 2^{128}$). Finding a collision for this 256-bit hash function needs an expected number of more than $2^{121.9}$ oracle queries. This is reasonably close to the workload of about 2^{128} queries to generate a collision for an ideal 256-bit hash.

Preimage and 2nd Preimage attacks. As in the case of the hash functions from [5], our hash function is vulnerable to meet-in-the-middle attacks. Thus, it is possible to compute preimages and 2nd preimages in time $O(\sqrt{|\langle \text{CHN} \rangle|})$. This upper bound on the complexity of preimage and 2nd preimage attacks is close to the lower bound on the complexity of collision attacks from Theorem 1 – preimage and second preimage attacks are not much slower than collision attacks. They are not much faster, either: A 2nd preimage trivially implies a collision. And an adversary who, given some $T \in \langle \text{CHN} \rangle$ finds some M with $h^*_{H_0}(M) = T$, can be turned into a probabilistic collision adversary – if we allow to choose $T = h^*_{H_0}(M')$ for a long secret random M'.

For this reason, we omit a formal treatment of security against preimage and 2nd preimage attacks, and we concentrate on the security against collision attacks.

4 Proof of Theorem 1 – the Collision Resistance of $b^*_{H_0}$

Each oracle query for $y = E_S(x)$ or $x = E_S^{-1}(y)$ implies two chaining values $H = f_x^{-1}(S)$ and $H' = f_y(S)$ (since $b(H, x) = H'$). As we assume no redundant queries, one of these chaining values is adversarially fixed, one isn't:

- If the query is for $y_i = E_S(x)$, then $H = f_x^{-1}(S)$ is adversarially fixed, while $H' = f_y(S)$ depends on the oracle's response y.
- Else, H' is fixed by the adversary, while H is oracle dependent.

We represent the implied chaining values and their relationship as a directed multi-graph (DMG) with edge-labels. The vertices are the chaining values H and H'. Each oracle query implies an arrow (i.e., a directed edge) pointing to the oracle dependent chaining value:

- If the adversary chooses x and S and asks for $y = E_S(x)$, the arrow points to H'.
- Else, the adversary chooses y and S and asks for $x = E_S^{-1}(y)$. The arrow points to H.

In both cases, the arrow is labelled by x.

Note that H and H' with $x \neq x'$ with $b(H, x) = b(H, x') = H'$ can exist. In the graph, the vertices H and H' are connected by two edges with labels x and x'. Depending on the oracle queries, the subgraph with these two edges and the vertices H and H' can be of either three shapes: two *different* arrows from H to H', or two *different* arrows from H' to H, or one arrow in each direction. Thus, our structure is a directed *multi*-graph (with edge-labels), but may fail to be a proper directed graph.

As it is easy to generate collisions or preimages for *adversarially fixed* chaining values, we concentrate on *oracle-dependent* chaining values.

- An adversary finds a **natural collision**, if any two arrows point to the same vertex.
- It finds a **natural preimage** of some image $H^* \in \langle \text{CHN} \rangle$, if any arrow points to H^*.

Next, we will show that finding a collision for $b^*_{H_0}$ implies finding either a natural collision or a natural preimage of H_0. We conclude the current proof with lower bounds for the number of oracle queries needed for finding natural collisions and finding natural preimages.

Lemma 2 *Model E as an ideal block cipher. Fix an initial value $H_0 \in \langle \text{CHN} \rangle$. Any adversary finding a collision for the iteration $b^*_{H_0}$ of $b = b_{f,E}$ either also finds a natural preimage of H_0 for b or a natural collision for b.*

Proof. Let the adversary output two messages $M = (M_1, \ldots, M_R) \neq (M'_1, \ldots, M'_{R'}) = M'$ (w.l.o.g. $R \geq R'$). Write $H_1, \ldots H_R$ and $H'_1, \ldots H'_{R'}$ for the chaining values. Assume the adversary neither found a natural collision nor a natural preimage of H_0, and observe the implications for our DMG: *Without a natural collision, there are no two edges pointing to the same vertex. Without a natural preimage of H_0, no vertex directs to H_0.*

Thus, our DMG actually is a loop-free proper graph – and even a very special one: Either M' is a prefix of M, in which case the DMG is a sequential list from H_0 to H_R, with $H_{R'} = H'_{R'} \neq H_R$ somewhere on the path. Or M' is not a prefix of M. The 2nd case implies

two sequential paths, one from H_0 to H_R, and another from H_0 to $H'_{R'}$. These fork at the first (i.e. smallest) j with $M_j \neq M'_j$ and never join again. See also Figure 2. (There is no third case – recall that $M' \neq M$ and M' is at most as long as M.) In both cases, $H'_{R'} \neq H_R$, i.e., M and M' don't collide. □

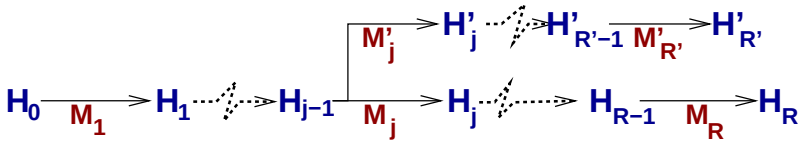

Fig. 2. The directed multi-graph (DMG) induced by hashing M and M', with edge-labels M_i and M'_i.

The following lemma quantifies how 'random' oracle-determined values are.

Lemma 3 *Allow the adversary to make up to N/K oracle queries. The probability to predict the response to the next (non-redundant) query is*

$$\textit{at most } \frac{1}{N-(N/K)} = \frac{K}{NK-N} = \frac{K}{N(K-1)}.$$

Proving Lemma 3 is trivial and omitted. Lemma 4 bounds the chances of finding a natural preimage of a given target, Lemma 5 bounds probability of finding a natural collision.

Lemma 4 *Let a target $T \in Chain$ for a natural preimage attack be given. Allow the adversary to make up to N/K oracle queries. Any such adversary succeeds in finding a preimage of T with at most the probability $1/(K-1)$.*

Proof. Consider a query for $y = E_S(x)$. As f is unique, there is (at most) one $y \in \langle \text{BLK} \rangle$ with $f_y(S) = T$, and, by Lemma 3, the oracle's response equals y with a probability $\leq K/(N(K-1))$. Similarly, if the oracle asked for $x = E_S^{-1}(y)$, the chance to respond the sole x with $f_x^{-1}(S) = T$ also is $\leq K/(N(K-1))$.

As we allow for N/K queries, the adversarial probability of success is no more than $(N/K) * (K/(N(K-1))) = 1/(K-1)$, as claimed. □

Lemma 5 *Allow the adversary to make N/K oracle queries. For all natural numbers L, the adversary's probability of finding a natural collision is*

$$\textit{less than } \frac{|\langle \text{CHN} \rangle| \cdot \beta^L}{L! \cdot (K-1)^L} + \frac{L-1}{K-1}.$$

Proof. For $S \in \langle \text{CHN} \rangle$ and an oracle query for $y = E_S(x)$, we define the set of chaining values which would allow a collision:

$$f_{(\cdot)}(S) := \{H \in \langle \text{CHN} \rangle \mid \exists y \in \langle \text{BLK} \rangle : \ f_y(S) = H\}.$$

Similarly if the query was for $x = E_S^{-1}(y)$:

$$f_{(\cdot)}(S) := \left\{ H \in \langle \textsc{chn} \rangle \mid \exists x \in \langle \textsc{blk} \rangle : \; f_x^{-1}(S) = H \right\}.$$

In the case of a natural collision two queries have been made, say the i-th query and the j-th query, w.l.o.g. $i < j$, whose oracle-dependent chaining values H_i and H_j match, i.e. $H_i = H_j$. This boils down to a sequence of two events:

1. $H_i \in f_{(\cdot)}(S_j)$, where S_j is fixed later.
2. $H_j = H_i$.

The larger $f_{(\cdot)}(S_j)$, the larger is the probability that some $i < j$ with $H_j = H_i$ exists. To prove an upper bound on the adversary's chance of success, we assume the adversary to choose S_j to maximise $|f_{(\cdot)}(S_j)|$, thus defining

$$\begin{aligned} |f_{(\cdot)}(S)|_j &:= \left| \left\{ H_i \in f_{(\cdot)}(S) \mid i < j \right\} \right| \\ \text{and} \quad |f_{(\cdot)}(\cdot)|_j &:= \max_{S \in \langle \textsc{chn} \rangle} \left\{ \left| \left\{ H_i \in f_{(\cdot)}(S) \mid i < j \right\} \right| \right\} = \max_{S \in \langle \textsc{chn} \rangle} \left\{ |f_{(\cdot)}(S)|_j \right\}. \end{aligned}$$

Two main claims for the proof of Lemma 5. Consider an arbitrary threshold L on $|f_{(\cdot)}(\cdot)|$.

1. Prove it to be unlikely that any S_j exists with L or more different chaining values $H_i \in f_{(\cdot)}(S_j)$ (with $i < j$). We denote this probability by $p_1(L)$ and claim

$$p_1(L) := \Pr\left[|f_{(\cdot)}(\cdot)|_j \geq L \right] \leq \frac{|\langle \textsc{chn} \rangle| \cdot \beta^L}{L! \cdot (K-1)^L}. \tag{2}$$

2. Prove that if there are less than L chaining values $H_i \in f_{(\cdot)}(S_j)$ (with $i < j$), then the probability of generating a natural collision is low. We denote this conditional probability by $p_2(L)$ and claim

$$p_2(L) < \frac{L-1}{K-1}. \tag{3}$$

3. To conclude, the probability of generating any natural collision is at most

$$p_1(L) + p_2(L) < \frac{|\langle \textsc{chn} \rangle| \cdot \beta^L}{L! \cdot (K-1)^L} + \frac{L-1}{K-1}.$$

Verifying the first claim (bound 2) for $p_1(L)$. We start with assuming some fixed $S \in \langle \textsc{chn} \rangle$ and showing a bound for $|f_{(\cdot)}(S)|_j$. Observe that $|f_{(\cdot)}(S)|_j \geq L$ if and only if there is at least *one* L-tuple $(H_{\sigma_1}, \ldots, H_{\sigma_L})$ of different oracle-dependent chaining values H_{σ_i} with

$$H_{\sigma_1} \in f_{(\cdot)}(S) \wedge \cdots \wedge H_{\sigma_L} \in f_{(\cdot)}(S).$$

By the collision universality of f (with parameter β), there are at most β oracle responses y_i that allow for $H_i \in f_{(\cdot)}(S)$. Using Lemma 3, we conclude that for all i $\Pr\left[H_i \in f_{(\cdot)}(S) \right] \leq$

$\frac{\beta K}{N(K-1)}$. Thus, for an L-tuple $(H_{\sigma_1}, \ldots, H_{\sigma_L})$ of pairwise different oracle-dependent chaining values we get

$$\Pr\left[H_{\sigma_1} \in f_{(\cdot)}(S) \wedge \ldots \wedge H_{\sigma_L} \in f_{(\cdot)}(S)\right] \leq \Pr\left[H_{\sigma_1} \in f_{(\cdot)}(S)\right] \cdot \ldots \cdot \Pr\left[H_{\sigma_L} \in f_{(\cdot)}(S)\right] \leq \left(\frac{\beta K}{N(K-1)}\right)^L \leq \frac{\beta^L K^L}{N^L (K-1)^L}.$$

The number of L-tuples of oracle queries is $\binom{N/K}{L}$. Thus

$$\Pr\left[|f_{(\cdot)}(S)|_j \geq L\right] \leq \binom{N/K}{L} \left(\frac{\beta K}{N(K-1)}\right)^L \leq \frac{N^L}{K^L \cdot L!} \cdot \frac{\beta^L K^L}{N^L (K-1)^L} \leq \frac{\beta^L}{L! \cdot (K-1)^L}.$$

As there are $|\langle \text{CHN} \rangle|$ choices for S, we get the claimed bound for $p_1(L)$:

$$p_1(L) = \Pr\left[\max_{S \in \langle \text{CHN} \rangle} \{|f_{(\cdot)}(S)|_j\} \geq L\right] \leq |\langle \text{CHN} \rangle| \cdot \Pr\left[|f_{(\cdot)}(S)|_j \geq L\right] \leq \frac{|\langle \text{CHN} \rangle| \cdot \beta^L}{L! \cdot (K-1)^L}.$$

Verifying the 2nd claim (bound 3), namely $p_2(L) \leq \frac{L-1}{K-1}$. Recall that $p_2(L)$ is under the condition that $|f_{(\cdot)}(\cdot)|_j < L$. Thus, for all S_j there are at most $L-1$ different chaining values $H_{\sigma_1}, \ldots, H_{\sigma_{L-1}} \in f_{(\cdot)}(S_j)$. Using Lemma 3 we conclude that the probability for the oracle-dependent chaining value H_j of the j-th query to equal a previous chaining value H_i (with $i < j$) is at most $(L-1)K/N(K-1)$.

The probability $p_2(L)$ that for $1 \leq j \leq N/K$ any $i < j$ with $H_j = H_i$ exists, is

$$p_2(L) < \frac{N}{K} \cdot \frac{(L-1)K}{N(K-1)} = \frac{L-1}{K-1}.$$

(There are several reasons why we can write "$<$" instead of "$\leq$". A sufficient one is that for $j = 1$ no such $i < j$ exists.) □

Proof (of Theorem 1). Due to Lemma 2, the probability of finding a collision for $b^*_{H_0}$ is bounded the probability of finding a natural collision plus the probability of finding a natural preimage of H_0:

$$\Pr[\text{coll}(b^*_{H_0})] < \overbrace{\frac{1}{K-1}}^{\text{Lemma 4}} + \overbrace{\frac{|\langle \text{CHN} \rangle| \cdot \beta^L}{L! \cdot (K-1)^L} + \frac{L-1}{K-1}}^{\text{Lemma 5}} = \frac{|\langle \text{CHN} \rangle| \cdot \beta^L}{L! \cdot (K-1)^L} + \frac{L}{K-1}.$$

□

5 Concrete Instantiations for f and Practical Consequences

In this section, we present practical choices for f satisfying the combinatorial properties defined in Section 3.1. For each choice of f, we will describe AES-based instantiations of E and evaluate the security level for practical parameter choices. (One can easily generalise this approach to one using any n-bit block cipher with $2n$-bit keys.)

5.1 Simple Instantiation (Collision Universality Parameter $\beta = 1$)

Let $\mathcal{F}$ be a field, $\mathcal{F}^* = \mathcal{F} - \{0\}$ its multiplicative subgroup, and set $\langle \text{CHN} \rangle := \mathcal{F} \times \mathcal{F}^*$ and $\langle \text{BLK} \rangle := \mathcal{F}^*$. We define

$$g : \overbrace{\mathcal{F} \times \mathcal{F}^*}^{\langle \text{CHN} \rangle} \times \overbrace{\mathcal{F}^*}^{\langle \text{BLK} \rangle} \to \overbrace{\mathcal{F} \times \mathcal{F}^*}^{\langle \text{CHN} \rangle} : \quad g(S, T, x) = g_x(S, T) = (S + x, T * x).$$

Now we show that our combinatorial properties hold:

Invertibility: For all $x \in \mathcal{F}^*$, the inverse of g_x is $g_x^{-1}(C, D) = (C - x, D/x)$.

Uniqueness: For all pairs $((S, T), (C, D)) \in \langle \text{CHN} \rangle^2$, there exists at most one $y \in \langle \text{BLK} \rangle$ with $g_y(S, T) = (C, D)$, namely $y = C - S$ if $C - S = D/T$, and none if $C - S \neq D/T$.

Collision universality (with parameter $\beta = 1$)**:** For all pairs $((S, T), (S', T')) \in \langle \text{CHN} \rangle^2$ with $(S, T) \neq (S', T')$, there exist at most one pair $(y, y') \in \langle \text{BLK} \rangle^2$ with $g_y(S, T) = g_{y'}(S', T')$:

- First, $g_y(S, T) = g_{y'}(S', T') \Leftrightarrow (S + y = S' + y') \wedge (T * y = T' y')$, thus $y' = S + y - S'$.
- Second, if $T = T'$ then $y = y'$ and thus $S = S'$, contradicting $(S, T) \neq (S', T')$.
- Third, if $T \neq T'$ then $T * y = T' * y' = T'S + T'y - T'S'$, and thus $y = \frac{T'(S - S')}{T - T'}$.
- Conclusion: if $T = T'$, there is no solution, else both y and y' are uniquely defined.

For **an AES-based instantiation** of the block cipher E, we set $\mathcal{F} = \text{GF}(2^{128})$. The key space of E is $\text{GF}(2^{128}) \times (\text{GF}(2^{128}) - \{0\})$ (where "0" is the neutral element of the addition). Thus, a tiny fraction of AES-256-keys are invalid for E. For all valid keys K, we define

$$E_K(X) = \begin{cases} \text{AES-256}_K(X) & \text{if AES-256}_K(X) \neq 0 \\ \text{AES-256}_K(\text{AES-256}_K(X)) & \text{else.} \end{cases}$$

As AES-256_K permutes over $\mathcal{F} = \text{GF}(2^{128})$, E permutes over $\mathcal{F}^*$. Now, we set $f := g$, implying $\beta = 1$. In order to minimise K under the condition $\Pr[\text{coll}(b^*_{H_0})] < 1/2$, we set $L := 28$ and $K := 59$. With these parameters, Equation 1 ensures $\Pr[\text{coll}(b^*_{H_0})] < 0.4988$. Even an adversary making up to $2^{128}/59 \approx 2^{122.1}$ oracle queries would succeed with at most the probability $\Pr[\text{coll}(b^*_{H_0})] < 0.5$.

5.2 Alternative Instantiation (Collision Universality Parameter $\beta = 3$)

The AES-based instantiation above prohibits the "0" as a message block, and, under very special circumstances, we have to call the AES (or whatever underlying block cipher) twice. This is unsatisfactory, as we may have to recode the message (to avoid "0"-blocks), and when hashing confidential messages, calling the AES twice may lead to side-channel attacks. For this reason, we propose an alternative construction, allowing all message blocks and always calling the AES only once. The disadvantage is that our collision universality parameter β increases to 3, marginally worsening the security level.

Set $\langle \text{CHN} \rangle := \mathcal{F} \times \mathcal{F}^*$ (as before), but $\langle \text{BLK} \rangle := \mathcal{F}$ (instead of $\mathcal{F}^*$) and define

$$g' : \overbrace{\mathcal{F} \times \mathcal{F}^*}^{\langle \text{CHN} \rangle} \times \overbrace{\mathcal{F}}^{\langle \text{BLK} \rangle} \to \overbrace{\mathcal{F} \times \mathcal{F}^*}^{\langle \text{CHN} \rangle} : \quad g'(S,T,x) = g'_x(S,T) = \begin{cases} (S + x, T * x) & \text{if } x \neq 0 \\ (S,T) & \text{if } x = 0. \end{cases}$$

Thus, $g'_x(S,T) = g_x(S,T)$ if $x \neq 0$ and $g'_0(S,T) = (S,T)$ else. This satisfies:

Invertibility: For $x \neq 0$, $g'_x = g_x$ is invertible. The identity g_0 is invertible, anyway.

Uniqueness: For all pairs $((S,T),(C,D)) \in \langle \text{CHN} \rangle^2$, there exists at most one $y \in \langle \text{BLK} \rangle$ with $g'_y(S,T) = (C,D)$, namely $y = C - S \neq 0$ if $C - S = D/T$ and $y = 0$ if $(S,T) = (C,D)$, and none if else.

Collision universality (with parameter $\beta = 3$)**:** From above, we know that for all pairs $((S,T),(S',T')) \in \langle \text{CHN} \rangle^2$ with $(S,T) \neq (S',T')$, there exists at most one pair (y,y') with $y \neq 0$, $y' \neq 0$ satisfying $g'_y(S,T) = g'_{y'}(S',T')$. Allowing $y = 0$ and $y' = 0$ may create at most two additional such pairs satisfying this equation.

This time, the **AES-based choice** for E is trivial: Under the same marginal key space restriction as above, set $E :=$AES-256 and $f := g'$, which implies $\beta = 3$. Minimise K, requiring $\Pr[\text{coll}(b^*_{H_0})] < 1$: $L := 33$, $K := 68$ and, by Equation 1, $\Pr[\text{coll}(b^*_{H_0})] < 0.493$. Thus, even if the adversary can make up to $2^{128}/68 \approx 2^{121.9}$ oracle queries, we have $\Pr[\text{coll}(b^*_{H_0})] < 0.5$.

5.3 A Practical Comparison of Security

Table 1 compares an ideal 256-bit hash function and three DBL constructions: the rate 1/2 hash from Hirose and two instantiations of our rate 1 hash. To measure the security, we require a collision probability below 1/2 and count the number of adversarial queries we can still allow. As it turns out, the security of Hirose's construction is less than ideal, but close. And the security of our constructions is less than that of Hirose's, but close again. The effect of changing the parameter β from 1 to 3 is practically negligible.

hash function H	number of queries allowed while $\Pr[\text{coll}(H)] \leq 0.5$
ideal 256-bit hash function	$\approx 2^{128}$
Hirose [11, Theorem 4]	$\leq 2^{125.7}$
this paper (with $\beta = 1$)	$\leq 2^{122.1}$
this paper (with $\beta = 3$)	$\leq 2^{121.9}$

Table 1. Comparing an ideal 256-bit hash function with 256-bit DBL hash functions based on 128-bit block ciphers, namely Hirose's construction and two possible instantiations of our construction. The table counts the number of queries allowed while preserving a collision probability below 1/2.

6 Conclusion and Further Work

In this paper, we proposed and analysed a rate 1 DBL hash function. In the ideal cipher model, we proved our construction collision resistant.

This work leaves some interesting open problems. Can one extend our construction to satisfy advanced security features for good general-purpose hash functions? Can one, e.g., apply the zipper or EMD construction [16, 2]? How should we design a combinatorial hash function f being very efficient in software? Is there a tradeoff between security and efficiency, i.e., can we improve efficiency by allowing a larger universal collision parameter β? How resistant are hash functions like our's to side-channel attacks, if we are hashing confidential messages?

References

1. D. Bernstein. The poly1305-AES message-authentication code. FSE 2005.
2. M. Bellare, T. Ristenpart. Multi-property-preserving hash domain extension: The EMD transform. Asiacrypt 2006.
3. E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet, W. Jalby. Collisions of SHA-0 and reduced SHA-1. Eurocrypt 2005.
4. J. Black, S. Halevi, H. Krawczyk, T. Krovetz, P. Rogaway. UMAC: fast and secure message authentication. Crypto '99.
5. J. Black, P. Rogaway, T. Shrimpton. Black-box analysis of the block-cipher based hash-function construction from PGV. Crypto 02.
6. J. Black. The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. FSE 2006.
7. M. Boesgaard, T. Christensen, E. Zenner. Badger – a fast and provably secure MAC. Applied Cryptography and Network Security, ACNS 2005.
8. J. Coron, Y. Dodis, C. Malinaud, P. Punyia. Merkle-Damgård revisited: how to construct a hash function. Crypto 2005.
9. I. Damgård. A design principle for hash functions. Crypto 89.
10. M. Etzel, S. Patel, Z. Ramzan. Square hash: fast message authentication via optimized universal hash functions. Crypto '99.
11. S. Hirose. Some Plausible Constructions of Double-Block-Length Hash Functions. FSE 2006.
12. A. Joux. Multicollisions in iterated hash functions, application to cascaded constructions. Crypto 04.
13. J. Kelsey, T. Kohno. Herding Hash Functions and the Nostradamus Attack. Eurocrypt 2006.
14. J. Kelsey. B. Schneier. Second preimages on n-bit hash functions in much less than 2^n work. Eurocrypt 2005.
15. T. Kohno, J. Viega, D. Whiting. CWC: a high performance conventional authenticared encryption mode. FSE 04.
16. M. Liskov. Constructing an Ideal Hash Function from Weak Ideal Compression Functions. SAC 2006.
17. S. Lucks. A Failure-Friendly Design Principle for Hash Functions. Asiacrypt 2005.
18. S. Lucks. Faster Luby-Rackoff Ciphers. FSE 1996.
19. A. Menezes, P. van Oorschot, S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
20. R. Merkle. One-way hash functions and DES. Crypto 89.
21. M. Naor, O. Reingold. On the Construction of Pseudorandom Permutations: Luby-Rackoff Revisited. *J. Cryptology 12(1)* 29-66 (1999).
22. B. Preneel, R. Govaerts, R. Vanderwalle. Hash functions based on block ciphers: A syntetic approach. Crypto 93.
23. P. Rogaway. Formalizing Human Ignorance: Collision-Resistant Hashing without the Keys. `http://eprint.iacr.org/2006/281`. (Also in Vietcrypt 2006.)

24. P. Rogaway and T. Shrimpton. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision-Resistance. FSE 2004.
25. X. Wang, X. Lai, D. Feng, H. Cheng, X. Yu. Cryptoanalyisis of the hash functions MD4 and RIPEMD. Eurocrypt 2005.
26. X. Wang, H. Yu. How to break MD5 and other hash functions. Eurocrypt 2005.
27. X. Wang, H. Yu, Y. L. Yin. Efficient collision search attacks on SHA0. Crypto 2005.
28. X. Wang, Y. L. Yin, H. Yu. Finding collisions in the full SHA1. Crypto 2005.
29. M. Wegman, J. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, Vol. 22, 1981, 265–279.