The Collision Security of MDC-4

No Author Given

No Institute Given

Abstract. There are four somewhat classical double length block cipher based compression functions known: MDC-2, MDC-4, Abreast-DM, and Tandem-DM. They all have been developed over 20 years ago. In recent years, cryptographic research has put a focus on block cipher based hashing and found collision security results for three of them (MDC-2, Abreast-DM, Tandem-DM). In this paper, we add MDC-4, which is part of the IBM CLiC cryptographic module[1], to that list by showing that – 'instantiated' using an ideal block cipher with 128 bit key/plaintext/ciphertext size – no adversary asking less than $2^{74.76}$ queries can find a collision with probability greater than 1/2. This is the first result on the collision security of the hash function MDC-4.
The compression function MDC-4 is created by interconnecting two MDC-2 compression functions but only hashing one message block with them instead of two. The developers aim for MDC-4 was to offer a higher security margin, when compared to MDC-2, but still being fast enough for practical purposes.
The MDC-2 collision security proof of Steinberger (EUROCRYPT 2007) cannot be directly applied to MDC-4 due to the structural differences. Although sharing many commonalities, our proof for MDC-4 is much shorter and we claim that our presentation is also easier to grasp.
Keywords: MDC-4, cryptographic hash function, block-cipher based, proof of security, double length, ideal cipher model.

1 Introduction

A cryptographic hash function is a function which maps an input of arbitrary length to an output of fixed length. It should satisfy at least collision-, preimage- and second-preimage resistance and is one of the most important primitives in cryptography [23]. In recent years, most of the functions in the widely used MD4-family (*e.g.*, MD4 [29], MD5 [30], RIPEMD [11], SHA-1 [27], SHA-2 [28]) have been successfully attacked in several ways [5, 10, 33, 34] which has stimulated researchers to look for alternatives. Block cipher based constructions seem promising since they are very well known – they even predate the MD4-approach [22]. One can easily create a hash function using, *e.g.*, the Davies-Meyer [35] mode of operation and the Merkle-Damgård transform [4, 24]. Also, many of the proposed SHA-3 designs like Skein [7], SHAvite-3 [1], and SIMD [21] use block cipher based instantiations. Another reason for the resurgence of interest in block cipher based hash functions is due to the rise of resource restricted devices such as RFID tags or smart cards. A hardware designer only needs to implement a block cipher in order to obtain an encryption function as well as a hash function. However, due to the short output length of most practical block ciphers, one is mainly interested in sound design principles for *double length* (DL) hash functions. Such double length hash functions use a block cipher with n-bit output as the building block by which it maps possibly long strings to $2n$-bit hash values. DL compression functions can be parted by the type of block cipher they need to operate: The first group, *(group-1)*, uses an internal block cipher with an n-bit plaintext/ciphertext/key, the second group, *(group-2)*, uses a block cipher with an n-bit plaintext/ciphertext and a k-bit key, $k > n$. DL compression functions in the first group are few. Currently, there are only three known candidates in literature: MDC-2, MDC-4 and a most recent variant of MDC-2: MJH

[1] FIPS 140-2 Security Policy for IBM CrytoLite in C, October 2003

Function	Security (Collision)	Attack (Collision)	Attack (Preimage)
MDC-2	$2^{74.91}$ [32]	2^{121} [13]	2^{2n} (time $\cdot$ space) [13, 16]
MDC-4	$2^{74.76}$ (this paper)	2^{96} [14] (only CF)	2^{224} [14]
MJH [18]	$2^{78.33}$	(no results known)	(no results known)

Table 1. List of known group-1 hash functions, values evaluated for an internal block cipher with 128 bit plaintext/ciphertext/key [Notation: CF = compression function]

[18]. Group-2 examples are Abreast-DM Tandem-DM, Cyclic-DM [16, 9], etc. The security of group-2 functions is relatively well understood.

MDC-4 is a acronym for Modification Detection Code with ratio 1/4, and was developed at IBM in the late eighties by Meyer and Schilling [25]. The ratio indicates the number of block cipher calls that are required to process a single message block. MDC-4 was originally specified for the 64-bit block cipher DES [26].

Our Contribution. In this paper, we give the first collision security bound for the hash function MDC-4, a block cipher based hash function that has been publicly known for more than 20 years. In our proof, we use many of the techniques that have been applied in the MDC-2 collision security proof [32]. Our proof is in the ideal cipher model, too. However, we consider MDC-4 using an ideal n-bit block cipher accepting n-bit keys. Furthermore, as in [32], we also ignore an additional *bit-fixing* step that was used back than as an additional security measure to avoid some DES specific key issues.

In this paper we show, assuming a hash output length of 256 bits, that any adversary asking less than $2^{74.76}$ queries to the block cipher cannot find a collision for the *hash function* MDC-4 with probability greater than 1/2. Note that the optimal security bound for collisions for 256 bit hash functions is about 2^{128}. For MDC-2 (ratio 1/2) and MJH (ratio 1/2), the trivial collision resistance bound is 2^{64}, since they both internally use a Davies-Meyer compression function. Although MDC-4 also uses Davies-Meyer type functions inside, even such a trivial bound is not so easy to see.

Related Work. For group-2 functions, there has been a lot of research in recent years, *e.g.* [8, 9, 15, 16, 17, 19, 20]. As a result, there are group-2 compression functions known that are 'provably optimal'. This is in stark contrast to the known results for group-1 functions which are summarized in Table 1.

Outline. The paper is organized as follows: Section 2 includes formal notations and definitions. In Section 3 we prove that an adversary asking less than $2^{74.76}$ oracle queries has the threshold probability 1/2 finding a collision for the MDC-4 hash function.

2 Preliminaries

2.1 General Notations

An n-bit block cipher is a keyed family of permutations consisting of two paired algorithms $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ and $E^{-1} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ both accepting a key of size n bits and an input block of size n bits for some $n > 0$. Let $\texttt{Block}(n)$ be the set of all n-bit block ciphers. For any $E \in \texttt{Block}(n)$ and any fixed key $K \in \{0,1\}^n$, decryption $E_K^{-1} := E^{-1}(K, \cdot)$

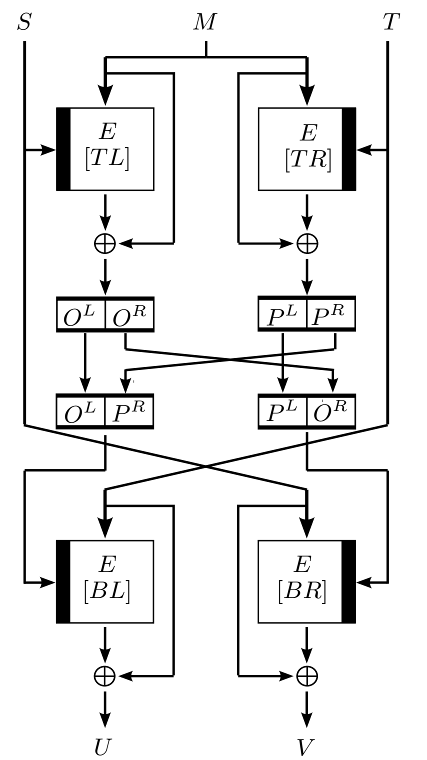

Figure 1. The double-length compression function H^E where E is an n-bit block cipher. The black bar inside the cipher indicates the key input.

is the inverse function of encryption $E_K := E(K, \cdot)$, so that $E_K^{-1}(E_K(X)) = X$ holds for any input $X \in \{0,1\}^n$. In the ideal cipher model E is modeled as a family of random permutations $\{E_K\}$ whereas the random permutations are chosen independently for each key K [2, 6, 12], *i.e.*, formally E is selected randomly from $\texttt{Block}(n)$. If $Y = E_K(X)$ we call the value $Z = X \oplus Y$ the XOR-output of a query (K, X, Y).

We use the convention to write oracles, that are provided to an algorithm, as superscripts. For example $\mathcal{A}^E$ is an algorithm $\mathcal{A}$ with oracle access to E to which $\mathcal{A}$ can request forward and backward queries. For ease of presentation, we identify the sets $\{0,1\}^{a+b}$ and $\{0,1\}^a \times \{0,1\}^b$. Similarly, for $A \in \{0,1\}^a$ and $B \in \{0,1\}^b$, the concatenation of these bit strings is denoted by $A||B \in \{0,1\}^{a+b} = \{0,1\}^a \times \{0,1\}^b$.

A compression function is a mapping $H : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^r$ for some $m, r > 0$. A block cipher-based compression function is a mapping $H^E : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^r$ that, given an r-bit state R and an m-bit message M, computes $H^E(M, R)$ using oracle access to some $E \in \texttt{Block}(n)$.

2.2 The MDC-4 Compression Function

The MDC-4 compression function H^E (cf. Figure 1) takes an n-bit message M, a $2n$-bit state (S, T) and outputs a new $2n$-bit state (U, V) as follows:

1. Compute $O = (O^L||O^R) = E_S(M) \oplus M$,
2. compute $P = (P^L||P^R) = E_T(M) \oplus M$,
3. compute $U = E_{O^L||P^R}(T) \oplus T$,

4. compute $V = E_{P^L||O^R}(S) \oplus S$,
5. output (U, V).

The superscript L denotes the left $n/2$ bits of an expression, and the superscript R denotes the right $n/2$ bits of an expression.

The original MDC-4 specification [25] swaps the right halves of U and V. But, since we are in the ideal cipher model, this operation does not change the distribution of the output and neither our collision security analysis. So, for ease of presentation, we omitted this additional step.

Our analysis is for the MDC-4 hash function $\mathcal{H}^E$ which is obtained by a simple iteration of the MDC-4 compression function H^E in the obvious manner: Given some $n \cdot \ell$-bit message $(M_1, \dots, M_\ell)$, $M_j \in \{0,1\}^n$ for $j = 1, \dots, \ell$ and an initial value $(S_0, T_0) \in \{0,1\}^{2n}$ it works by computing $(S_i, H_i) = H^E(M_i, S_{i-1}, T_{i-1})$ for $i = 1, \dots, \ell$. The hash value is (S_ℓ, T_ℓ).

2.3 Security of the MDC-4 compression function and the MDC-4 hash function

Generally, insecurity is quantified by the success probability of an optimal resource-bounded adversary. The resource is the number of backward and forward queries to the block cipher E. For a set C, let $Y \xleftarrow{\$} C$ represent random sampling from C under the uniform distribution. For a probabilistic algorithm $\mathcal{D}$, let $Y \xleftarrow{\$} \mathcal{D}$ mean that Y is an output of $\mathcal{D}$ and its distribution is based on the random choices of $\mathcal{D}$.

In our case, an adversary is a computationally unbounded collision-finding algorithm $\mathcal{A}^E$ with access to $E \in \mathtt{Block}(n)$. We assume that $\mathcal{A}^E$ is deterministic. The adversary may make a *forward* query $(K, X)_f$ to discover the corresponding value $Y = E_K(X)$, or the adversary may make a *backward* query $(K, Y)_b$, so as to learn the corresponding value $X = E_K^{-1}(Y)$ such that $E_K(X) = Y$. Either way, the result of the query is stored in a triple $(K_i, X_i, Y_i) := (K, X, Y)$ and the *query history* $\mathcal{Q}$ is the tuple $(Q_1, \dots, Q_q)$ where $Q_i = (K_i, X_i, Y_i)$ and q is the total number of queries made by the adversary.

Without loss of generality, we assume that $\mathcal{A}^E$ asks at most only once on a triplet of a key K_i, a plaintext X_i and a ciphertext Y_i obtained by a query and the corresponding reply.

Collision Resistance of the MDC-4 *compression function.* There is a very simple attack on the compression function which only requires about $2^{n/2}$ invocations of the E oracle: Let the adversary find values $K, K', M, M' \in \{0,1\}^n$ such that $E_K(M) = E_{K'}(M')$. This requires about $2^{n/2}$ E-oracle queries. Then, by

$$H^E(M, K, K) = H^E(M', K', K'),$$

a collision for the full MDC-4 compression function has been found. So our analysis will be for the MDC-4 compression function *in the iteration.* This attack is only possible if the the chaining values are equal.

3 Proof of Collision Resistance

3.1 Proof Model

Our analysis is for the MDC-4 hash function $\mathcal{H}^E$ assuming that the initial chaining values are different, *i.e.*, $S_0 \neq T_0$. The goal of the adversary is to output two messages $\mathcal{M}_1 \in \{0,1\}^{n \cdot \ell}$ and $\mathcal{M}_2 \in \{0,1\}^{n \cdot \ell'}$ such that $\mathcal{H}(\mathcal{M}_1) = \mathcal{H}(\mathcal{M}_2)$ for some non-zero integers ℓ, ℓ'.

In our analysis, we dispense the adversary from returning these two messages. Instead we upper bound his success probability by giving the attack to him if

(i) he has found an 'internal' collision, *i.e.*, (M, S, T) such that $(U, V) = H^E(M, S, T)$ with $U = V$ for some $U, V \in \{0,1\}^n$ or

(ii) case (i) is not `true` but he has either found a collision in the compression function H^E, *i.e.*, (M, S, T) and (M', S', T'), such that $H^E(M, S, T) = H^E(M', S', T')$ or

(iii) cases (i), (ii) are not `true` but he has found values (M, S, T) such that $(S_0, T_0) = H^E(M, S, T)$. *Note that this requirement essentially models the preimage resistance of the* MDC-4 *compression function.*

The proof is simple and straightforward. Assume a collision for $\mathcal{H}^E$ has been found using two not necessarily equal-length messages $\mathcal{M}$ and $\mathcal{M}'$, *i.e.*, $\mathcal{H}^E(\mathcal{M}) = \mathcal{H}^E(\mathcal{M}')$. Also assume that the collision is the earliest possible. Then the adversary has either found (i) or (ii). For case (iii), we also give the attack to the adversary, particularly for reasons already discussed in Section 2.3.

For our analysis, we impose the reasonable condition that the adversary must have made all queries necessary to compute the results. We determine whether the adversary has been successful or not by examining the query history $\mathcal{Q}$. Formally, we say that $\textsc{Coll}(\mathcal{Q})$ holds if there is such a collision and $\mathcal{Q}$ contains all the queries necessary to compute it.

We now define what we formally mean by a collision of the MDC-4 compression function.

Definition 1. ***(Collision resistance of the*** **MDC-4** ***compression function)*** *Let H^E be a* MDC-4 *compression function. Fix an adversary $\mathcal{A}$. Then the advantage of $\mathcal{A}$ in finding collisions for H^E is the real number*

$$\begin{aligned}\mathbf{Adv}_{H^E}^{\textsc{Coll}}(\mathcal{A}) = \Pr[&E \xleftarrow{\$} \mathit{Block}(n); ((M,S,T),(M',S',T')) \xleftarrow{\$} \mathcal{A}^{E,E^{-1}} : \\ &((M,S,T) \neq (M',S',T')) \wedge H^E(M,S,T) = H^E(M',S',T')].\end{aligned}$$

For $q \geq 1$ we write

$$\mathbf{Adv}_{H^E}^{\textsc{Coll}}(q) = \max_{\mathcal{A}}\{\mathbf{Adv}_{H^E}^{\textsc{Coll}}(\mathcal{A})\},$$

where the maximum is taken over all adversaries that ask at most q oracle queries, *i.e.*, forward and backward queries to E.

Since our analysis in the next sections is for $\mathcal{H}^E$, we informally say that the probability of a collision of $\mathcal{H}^E$ is upper bounded by using a union bound for the cases (i), (ii) and (iii). This is part of the formalization in Theorem 1.

3.2 Our Results

We now give our main result. Although having a substantial complexity on the first sight in its general form, we can easily evaluate it to numerical terms (cf. Corollary 1).

Theorem 1. *Fix some initial values $S_0, T_0 \in \{0,1\}^n$ with $S_0 \neq T_0$ and let $\mathcal{H}^E$ be the* MDC-4 *hash function as given in Section 2.2. Let α, β, γ be constants such that $eq2^{n/2}/(2^n - q) \leq \alpha$,*

q	$\mathbf{Adv}^{\text{Coll}}_{\mathcal{H}^E}(q) \le$	α	β	γ
2^{64}	$7.18 \cdot 10^{-7}$	42	4.0	$2 \cdot 10^6$
$2^{68.26}$	10^{-4}	126	4.0	$6 \cdot 10^6$
$2^{72.19}$	1/100	900	4.0	$1.3 \cdot 10^7$
$2^{73.84}$	1/10	2600	4.0	$1.4 \cdot 10^7$
$2^{74.40}$	1/4	3780	4.0	$1.5 \cdot 10^7$
$2^{74.76}$	1/2	4900	4.0	$1.5 \cdot 10^7$

Table 2. Upper bounds on $\mathbf{Adv}^{\text{Coll}}_{\mathcal{H}^E}(q)$ as given by Theorem 1

$eq/(2^n - q) \le \beta$ and let $\Pr[\text{Lucky}(\mathcal{Q})]$ *as in Proposition 5 (Appendix A). Then*

$$\begin{aligned}\mathbf{Adv}^{\text{Coll}}_{\mathcal{H}^E}(q) \le & q\left(\frac{\alpha^2+\gamma}{2^n-q} + \frac{\alpha\beta}{(2^n-q)(2^{n/2}-\alpha)} + \frac{\alpha}{2^n-2^{n/2}\alpha} + \frac{\beta^2+4}{2^n-q} + \frac{\beta}{2^n-q}\right)\\ & +2q\left(\frac{\alpha^2(\alpha^2+2\gamma+1)+\alpha\gamma+\alpha}{2^n-q} + \frac{\alpha^3+2\alpha^2+\alpha}{(2^{n/2}-\alpha)^2}\right)\\ & +q\left(\frac{\gamma\alpha^2+\gamma^2}{2^n-q} + \frac{2\alpha}{(2^{n/2}-\alpha)^2}\right) + \Pr[\text{Lucky}(\mathcal{Q})]. \end{aligned} \tag{1}$$

The proof of Theorem 1 is developed throughout the following discussion and explicitly stated in Section 3.5. As mentioned before, our bound is rather non-transparent, so we discuss it for $n = 128$. We evaluate the equation above such that the adversary's advantage is upper bounded by 1/2 – thereby maximizing the value of q by numerically optimizing the values of α, β and γ. Our result is the following corollary.

Corollary 1. *No adversary asking less than* $2^{74.76}$ *queries can find a collision for the* MDC-4 *hash function with probability greater than* 1/2*.*

An overview of the behavior of our upper bound is given in Table 2. Note that for other values of (α, β, γ) the bound stays *correct* but worsens numerically (as long as the conditions given in Theorem 1 hold).

3.3 Proof Preliminaries

Overview. Our discussion starts with case (ii). We analyze whether the list of oracle queries to E made by the adversary can be used for a collision of the MDC-4 compression function H^E. For a collision, there are eight – not necessarily distinct – block cipher queries necessary (cf. Figure 2).

To upper bound the probability of the adversary obtaining queries that can be used for a collision, we upper bound the probability of the adversary making a final query that can be used as the last query to complete such a collision. Let $\mathcal{Q}_i$ denote the set of the first i queries $(K_1, X_1, Y_1), ..., (K_i, X_i, Y_i)$ (either forward or backward) made by the adversary. Furthermore we denote by the term *last query* the latest query made by the adversary. This query has always index i. Therefore, for each i with $1 \le i \le q$, we upper bound the success probability of an adversary to use the i-th query to complete the collision.

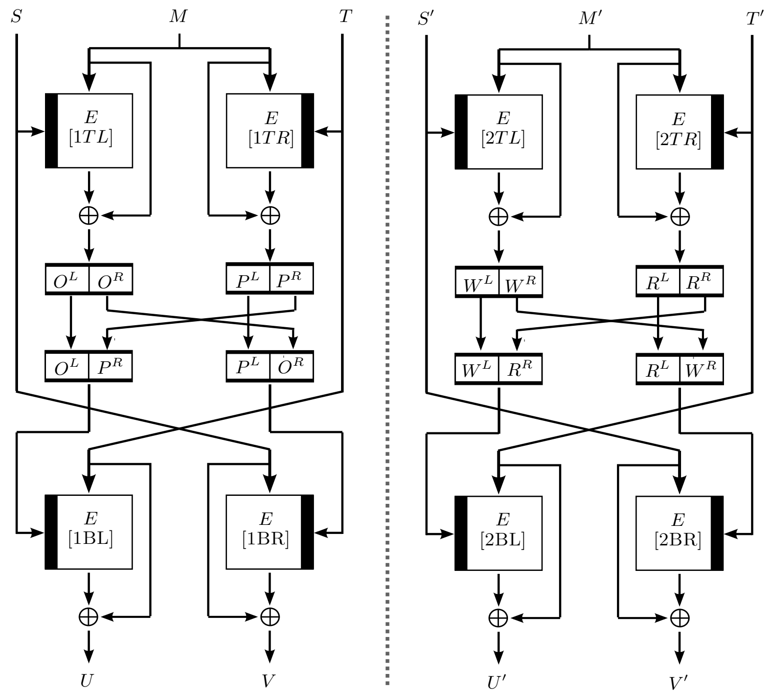

Figure 2. The double-length MDC-4 compression function H^E, where E is a (n, n)-block cipher. If $(S, M, T) \neq (S', M', T')$ but $(U, V) = (U', V')$ then the adversary has found a collision for H^E. The black beam inside the cipher indicates the key input. For later reference, the different positions a query can be used in are denoted by $1TL, 1TR, \ldots, 2\text{BR}$.

As the probability depends on the first $i - 1$ queries, we have to put some restrictions on these and also upper bound the probability that these restrictions are not met by an adversary. One example of such a restriction is to assume that, *e.g.*, the adversary has to find too many *collisions* for the underlying component function $E_K(X) \oplus X$.

Thus, our upper bound breaks down into two parts: an upper bound for the probability of an adversary not meeting our restrictions and the probability of an adversary ever making a successful i-th query, conditioned on the fact that the adversary does meet our restrictions and has not been successful by its $(i-1)$-th query. We use some notations that are given in Figure 2, *e.g.*, the statement $1\text{BL} \neq 2\text{BL}$ means that the query used in the bottom left of the 'left' side is not the same as the query used in the bottom left of the 'right' side.

3.4 Details.

We say $\textsc{Coll}(\mathcal{Q})$ if the adversary *wins*. Note that winning does not necessarily imply, that the adversary has found a collision. It might be that the adversary got lucky and does not meet our restrictions any more. But in the case of a collision $\textsc{Coll}(\mathcal{Q})$ always holds.
Proposition 1.

$$\begin{gathered}\textsc{Coll}(\mathcal{Q}) \Longrightarrow \\ \textsc{Lucky}(\mathcal{Q}) \vee \textsc{InternalColl}(\mathcal{Q}) \vee \textsc{CollTopRows}(\mathcal{Q}) \vee \textsc{CollLeftColumns}(\mathcal{Q}) \vee \\ \textsc{CollRightColumns}(\mathcal{Q}) \vee \textsc{CollBothColumns}(\mathcal{Q}) \vee \textsc{Preimage}(\mathcal{Q}).\end{gathered}$$

We now define the involved predicates of Proposition 1 and then give a proof. The predicates on the 'right' side are made mutually exclusive meaning that if the left side is true it follows that exactly one of the predicates on the right side is true. By upper bounding separately the probabilities of these predicates on the right side it is easy to see that the union bound can be used to upper bound the probability of $\textsc{Coll}(\mathcal{Q})$ as follows:

$$\begin{aligned}\Pr[\textsc{Coll}(\mathcal{Q})] \leq &\Pr[\textsc{Lucky}(\mathcal{Q})] + \Pr[\textsc{InternalColl}(\mathcal{Q})] + \Pr[\textsc{CollTopRows}(\mathcal{Q})] \\ &+ \Pr[\textsc{CollLeftColumns}(\mathcal{Q})] + \Pr[\textsc{CollRightColumns}(\mathcal{Q})] \\ &+ \Pr[\textsc{CollBothColumns}(\mathcal{Q})] + \Pr[\textsc{Preimage}(\mathcal{Q})].\end{aligned}$$

To state the predicate $\textsc{Lucky}(\mathcal{Q})$, we give some helper definitions that are also used as restrictions for the other predicates. Let $\texttt{NumEqual}(\mathcal{Q})$ be a function defined on the query set $\mathcal{Q}$, $|\mathcal{Q}| = q$ as follows:

$$\texttt{NumEqual}(\mathcal{Q}) = \max_{Z \in \{0,1\}^n} |\{i : E_{K_i}(X_i) \oplus X_i = Z\}|.$$

It is the maximum size of a set of queries in $\mathcal{Q}$ whose XOR-outputs are all the same. Similarly, we define $\texttt{NumEqualHalf}(\mathcal{Q})$ as the maximum size of a set of queries whose XOR-outputs either share the same left half or the same right half. Let

$$\begin{aligned}\texttt{NEH-L}(\mathcal{Q}) &= \max_{Z \in \{0,1\}^{n/2}} |\{i : (E_{K_i}(X_i) \oplus X_i)^L = Z\}|, \\ \texttt{NEH-R}(\mathcal{Q}) &= \max_{Z \in \{0,1\}^{n/2}} |\{i : (E_{K_i}(X_i) \oplus X_i)^R = Z\}|,\end{aligned}$$

then $\texttt{NumEqualHalf}(\mathcal{Q}) = \max(\texttt{NEH-L}(\mathcal{Q}), \texttt{NEH-R}(\mathcal{Q}))$. Let $\texttt{NumColl}(\mathcal{Q})$ be also defined on a query set $\mathcal{Q}$, $|\mathcal{Q}| = q$, as

$$\texttt{NumColl}(\mathcal{Q}) = |\{(i,j) \in \{1, \ldots, q\}^2 : i \neq j, E_{K_i}(X_i) \oplus X_i = E_{K_j}(X_j) \oplus X_j\}|.$$

It outputs the number of ordered pairs of distinct queries in $\mathcal{Q}$ which have the same XOR-outputs.

We now define the event $\textsc{Lucky}(\mathcal{Q})$ as

$$\textsc{Lucky}(\mathcal{Q}) = (\texttt{NumEqualHalf}(\mathcal{Q}) > \alpha) \ \vee \ (\texttt{NumEqual}(\mathcal{Q}) > \beta) \ \vee \ (\texttt{NumColl}(\mathcal{Q}) > \gamma),$$

where α, β and γ are the constants from Theorem 1. These constants are chosen depending on n and q by a simple numerical optimization process such that the upper bound of the advantage of an adversary is minimized for given values of n, q.

We now give the definitions of the other predicates.

FitInternalColl$(\mathcal{Q})$**.** The adversary has found four – not necessarily distinct – queries such that $H^E(M,S,T)$ can be computed and $H^E(M,S,T) = (U,U)$ holds for some arbitrary U with $S \neq T$.

FitCollLeftColumns$(\mathcal{Q})$**.** The adversary has found eight – not necessarily distinct – queries such that $(U,V) = H^E(M,S,T)$ and $(U',V') = H^E(M',S',T')$ can be computed with $U = U'$, 1BL $\neq$ 2BL and 1BR = 2BR.

FitCollRightColumns($\mathcal{Q}$)**.** The adversary has found eight – not necessarily distinct – queries such that $(U, V) = H^E(M, S, T)$ and $(U', V') = H^E(M', S', T')$ can be computed with $V = V'$, $1\mathrm{BR} \neq 2\mathrm{BR}$ and $1\mathrm{BL} = 2\mathrm{BL}$.

FitCollTopRows($\mathcal{Q}$)**.** The adversary has found four – not necessarily distinct – queries such that

$$(E_S(M) \oplus M, E_T(M) \oplus M) = (E_{S'}(M') \oplus M', E_{T'}(M') \oplus M')$$

for $S \neq T$, $S' \neq T'$, $1\mathrm{BL} = 2\mathrm{BL}$ and $1\mathrm{BR} = 2\mathrm{BR}$.

FitCollBothColumns($\mathcal{Q}$)**.** In this case we assume $\neg$FitCollLeftColumns($\mathcal{Q}$) and $\neg$FitCollRightColumns($\mathcal{Q}$). The adversary has found eight – not necessarily distinct – queries such that $(U, V) = H^E(M, S, T)$ and $(U', V') = H^E(M', S', T')$ can be computed with $U = U'$, $V = V'$, $1\mathrm{BL} \neq 2\mathrm{BL}$ and $1\mathrm{BR} \neq 2\mathrm{BR}$.

FitPreimage($\mathcal{Q}$)**.** This formalizes case (iii). The adversary has found four – not necessarily distinct – queries used in H^E in positions 1TL, 1TR, 1BL, 1BR such that the output of H^E is equal to (S_0, T_0), *i.e.*, the initial chaining values of the MDC-4 hash function.

For practical purposes we derive our predicates as follows.

$$\begin{aligned}
\textsc{InternalColl}(\mathcal{Q}) &= \neg\textsc{Lucky}(\mathcal{Q}) \wedge \textsc{FitInternalColl}(\mathcal{Q}),\\
\textsc{CollLeftColumns}(\mathcal{Q}) &= \neg(\textsc{Lucky}(\mathcal{Q}) \vee \textsc{FitInternalColl}(\mathcal{Q})) \wedge \textsc{FitCollLeftColumns}(\mathcal{Q}),\\
\textsc{CollRightColumns}(\mathcal{Q}) &= \neg(\textsc{Lucky}(\mathcal{Q}) \vee \textsc{FitInternalColl}(\mathcal{Q}) \vee \textsc{FitCollLeftColumns}(\mathcal{Q}))\\
&\quad \wedge \textsc{FitCollRightColumns}(\mathcal{Q}),\\
\textsc{CollTopRows}(\mathcal{Q}) &= \neg(\textsc{Lucky}(\mathcal{Q}) \vee \textsc{FitInternalColl}(\mathcal{Q}) \vee \textsc{FitCollLeftColumns}(\mathcal{Q})\\
&\qquad \vee \textsc{FitCollRightColumns}(\mathcal{Q})) \wedge \textsc{FitCollTopRows}(\mathcal{Q}),\\
\textsc{CollBothColumns}(\mathcal{Q}) &= \neg(\textsc{Lucky}(\mathcal{Q}) \vee \textsc{FitInternalColl}(\mathcal{Q})\\
&\quad \vee \textsc{FitCollLeftColumns}(\mathcal{Q}) \vee \textsc{FitCollRightColumns}(\mathcal{Q})\\
&\quad \vee \textsc{FitCollTopRows}(\mathcal{Q})) \wedge \textsc{FitCollBothColumns}(\mathcal{Q}),\\
\textsc{Preimage}(\mathcal{Q}) &= \neg(\textsc{Lucky}(\mathcal{Q}) \vee \textsc{FitInternalColl}(\mathcal{Q})\\
&\quad \vee \textsc{FitCollLeftColumns}(\mathcal{Q}) \vee \textsc{FitCollRightColumns}(\mathcal{Q})\\
&\quad \vee \textsc{FitCollTopRows}(\mathcal{Q}) \vee \textsc{FitCollBothColumns}(\mathcal{Q}))\\
&\quad \wedge \textsc{FitPreimage}(\mathcal{Q}).
\end{aligned}$$

Proof of Proposition 1. Assume that the adversary is not lucky, *i.e.*, $\neg\textsc{Lucky}(\mathcal{Q})$. Then it is easy to see that

$$\begin{gathered}
\textsc{FitInternalColl}(\mathcal{Q}) \vee \textsc{FitCollLeftColumns}(\mathcal{Q}) \vee \textsc{FitCollRightColumns}(\mathcal{Q}) \vee\\
\textsc{FitCollTopRows}(\mathcal{Q}) \vee \textsc{FitCollBothColumns}(\mathcal{Q}) \vee \textsc{FitPreimage}(\mathcal{Q})\\
\Longrightarrow\\
\textsc{InternalColl}(\mathcal{Q}) \vee \textsc{CollLeftColumns}(\mathcal{Q}) \vee \textsc{CollRightColumns}(\mathcal{Q}) \vee\\
\textsc{CollTopRows}(\mathcal{Q}) \vee \textsc{CollBothColumns}(\mathcal{Q}) \vee \textsc{Preimage}(\mathcal{Q})
\end{gathered}$$

holds. Therefore it is sufficient to show that

$$\begin{aligned}\textsc{Coll}(\mathcal{Q}) \qquad &\Longrightarrow \textsc{FitInternalColl}(\mathcal{Q}) \vee \textsc{FitCollLeftColumns}(\mathcal{Q}) \\ &\quad \vee \textsc{FitCollRightColumns}(\mathcal{Q}) \vee \textsc{FitCollTopRows}(\mathcal{Q}) \\ &\quad \vee \textsc{FitCollBothColumns}(\mathcal{Q}) \vee \textsc{FitPreimage}(\mathcal{Q}).\end{aligned}$$

To ensure that the chaining values are always different, we give the attack to the adversary if these values collide, *i.e.*, $U = V$ or $U' = V'$. Note that this is usually not a *real* collision, but we can exclude this case in our analysis. We call this $\textsc{InternalColl}(\mathcal{Q})$. This corresponds to case (i) in Section 3.1.

For the case (ii), we assume that a collision for the MDC-4 compression function H^E can be constructed from the queries in $\mathcal{Q}$. Then there are inputs $\mathcal{M}, \mathcal{M}' \in (\{0,1\}^n)^+$, $\mathcal{M} \neq \mathcal{M}'$ such that $\mathcal{H}(\mathcal{M}) = \mathcal{H}(\mathcal{M}')$. In particular, there are $M, M' \in \{0,1\}^n$ and $(S,T),(S',T') \in \{0,1\}^{2n}$, $(S,T,M) \neq (S',T',M')$, such that $H^E(S,T,M) = H^E(S',T',M')$.

For the following analysis we have $\neg\textsc{InternalColl}(\mathcal{Q})$, *i.e.*, $S \neq T$, $S' \neq T'$. Our case differentiation is based on the disposal of queries in the bottom row. First assume that 1BL = 2BL and 1BR = 2BR. Then $\textsc{CollTopRows}(\mathcal{Q})$. Now assume that 1BL = 2BL and 1BR $\neq$ 2BR. Then $\textsc{CollRightColumns}(\mathcal{Q})$. Conversely, if 1BL $\neq$ 2BL and 1BR = 2BR, we say $\textsc{CollLeftColumns}(\mathcal{Q})$. The only missing case, 1BL $\neq$ 2BL and 1BR $\neq$ 2BR, is denoted by $\textsc{CollBothColumns}(\mathcal{Q})$. $\textsc{Preimage}(\mathcal{Q})$ formalizes case (iii) of Section 3.1 and corresponds to $\textsc{FitPreimage}(\mathcal{Q})$. □

General Remarks. The strategy for the other predicates is to upper bound the probability of the last query being successful conditioned on the fact that the adversary has not yet been successful in previous queries. We say that the last query is successful if the output is such that `NumEqualHalf`$(\mathcal{Q}) < \alpha$, `NumEqual`$(\mathcal{Q}) < \beta$, `NumColl`$(\mathcal{Q}) < \gamma$ and that one of the predicates is *true*.

Proposition 2 (InternalColl($\mathcal{Q}$)).

$$\Pr[\textsc{InternalColl}(\mathcal{Q})] \leq q \left(\frac{\alpha^2 + \gamma}{2^n - q} + \frac{\alpha\beta}{(2^n - q)(2^{n/2} - \alpha)} + \frac{\alpha}{2^n - 2^{n/2}\alpha} \right)$$

Proof. The adversary can use the last query Q_i either once or twice. When Q_i is used three times or more then it must occur twice either in the top- or bottom row. But this would imply $S = T$.

In the case that the query is used once it can either be used in the top or bottom row. Due to the symmetric structure of MDC-4, we can assume WLOG that the last query Q_i is either used in position TL or BL[2]. The success probability is analyzed in Lemma 1.

In the case that Q_i is used twice, it must be used once in the top and once in the bottom row. We again assume that the last query is WLOG used in TL and BL or TL and BR. The success probability is analyzed in Lemma 2. □

Lemma 1. *Let $S \neq T$ and $\mathcal{Q}_{i-1}$ the query list not containing the last query Q_i. Assume that Q_i is used once in the* MDC-4 *compression function H^E. Then*

$$\Pr[(U,U) = H^E(S,T,M)] \leq q \left(\frac{\alpha^2 + \gamma}{2^n - q} \right).$$

[2] In this case we only consider the 'left' side of Figure 2 and denote 1TL by TL, 1TR by TR, 1BL by BL and 1BR by BR.

Proof.

Case 1: Assume first that $Q_i = (K_i^L || K_i^R, X_i, Y_i)$ is used in position BL. It follows that K_i^L must be equal to the XOR-output Z_{TL}^L of the query in TL. It follows that there are at most α different candidates for the query in TL in the query history $\mathcal{Q}_{i-i}$. Similarly, because K_i^R must be equal to the right half of the XOR-output of TR, Z_{TR}^R, there are at most α candidates for that can be used in TR. For the query in BR, there are at most α^2 possible key inputs, the ciphertext input of BR is determined by the query used in TL. So the probability that there is a query in $\mathcal{Q}_i$ such that $U = V$ is upper bounded by $\alpha^2/(2^n - q)$. For q queries, the total chance of success is $\leq q\alpha^2/(2^n - q)$.

Case 2: Now assume that Q_i is used in position TL. Since $S \neq T$ it follows that BL $\neq$ BR. So there are at most γ ordered pairs of queries that can be used in BL and BR such that their XOR-output collides. Fixing one of these, it fully determines the XOR-output TL. So, for q queries, Q_i has at most a chance of $q\gamma/(2^n - q)$. $\square$

Lemma 2. *Let $S \neq T$ and $\mathcal{Q}_{i-1}$ the query list not containing the last query Q_i. Assume that Q_i is used twice in the* MDC-4 *compression function H^E. Then*

$$\Pr[(U,U) = H^E(S,T,M)] \leq q\left(\frac{\alpha\beta}{(2^n - q)(2^{n/2} - \alpha)} + \frac{\alpha}{2^n - 2^{n/2}\alpha}\right).$$

Proof. By symmetry arguments, we assume WLOG that the last query Q_i is used in position TL. Since $S \neq T$, the last query can only appear a second time in position BL, or BR but not in TR.

Case 1: Assume Q_i is used in position TL and BL. This query can be used in these positions if the randomly determined left-side XOR-output Z_i^L is equal to the left-side of the key K_i^L. This event is called P_K and its probability of success can be upper bounded for Q_i by $\Pr[P_K] \leq \alpha/(2^{n/2} - \alpha)$. We now upper bound the number of queries that can be used in BR conditioned on the fact that P_K is successful. There are at most α queries that can be used in TR, since now the key input of BL is fixed. As the ciphertext input of BR is now also fixed by TL, there are at most β possibilities for BR. So the chance of success for the i-th query in this case is $\leq \frac{\beta}{2^n - q} \cdot \Pr[P_K]$. So for q queries the bound becomes $\frac{q\alpha\beta}{(2^n - q)(2^{n/2} - \alpha)}$.

Case 2: Assume Q_i is used in position TL and BR. Then, $K_i = X_i$. The query Q_i can be used in these two positions at the same time if the randomly determined right-half XOR-output Z_i^R is equal to the right-half of the key, $K_i^R = X_i^R$. This event is called O_K and its probability of success can be upper bounded for Q_i by $\Pr[O_K] \leq \frac{1}{2^{n/2}}$.
We now upper bound the number of queries that can be used in TR conditioned on the fact the O_K is successful. There are at most α queries that can be used in TR such that $Z_{TR}^L = K_i^L$ holds. Hence, there are at most α queries that can be used in BL. We denote the chance that $Z_{BL}^L = Z_i^L$ for the i-the query as $\Pr[Z_i^L]$. This event can thus be upper bounded by $\frac{\alpha}{2^{n/2} - \alpha} \cdot \Pr[O_K] \leq \frac{\alpha}{2^n - 2^{n/2}\alpha}$. For q queries we can upper bound this case by $\frac{q\alpha}{2^n - 2^{n/2}\alpha}$. $\square$

Proposition 3 (CollTopRows($\mathcal{Q}$)).

$$\Pr[\textsc{CollTopRows}(\mathcal{Q})] \leq \frac{q\beta}{2^n - q}$$

Proof. In this case we consider a collision in the top row, with 1BL = 2BL and 1BR = 2BR. This implies $S = S'$ and $T' = T$. Furthermore it implies $M \neq M'$, because otherwise we would have 1TL = 2TL and 1TR = 2TR. Regarding to this constraints we have to upper bound the probability that the i-th query can be used such that

$$(E_S(M) \oplus M, E_T(M) \oplus M) = (E_{S'}(M') \oplus M', E_{T'}(M') \oplus M').$$

Note, that no internal collision has happened before, *i.e.*, $\neg$InternalColl($\mathcal{Q}$), and therefore the chaining values are *always* different. First assume that the last query is used twice or more. In order to find a collision in the top-row, the last query must be used in the top-row or otherwise the success probability is zero. The last query cannot be used in 1TL and 2TL or else 1TL = 2TR and $M = M'$ would follow. The last query also cannot be used in 1TL and 2TR or else $S = T' = S' = T$ would follow.

Now assume that Q_i is used once, WLOG in 1TL. Then there are at most β pairs of queries for 1TR, 2TR that form a collision. So there are at most β queries that can be used in 2TL that may form a collision with the XOR-output of the last query used in 1TL. The success probability for q queries can therefore be upper bounded by $q\beta/(2^n - q)$. ◻

Proposition 4 (Preimage($\mathcal{Q}$)).

$$\Pr[\textsc{Preimage}(\mathcal{Q})] \leq \frac{q(4+\beta^2)}{2^n - q}$$

Proof. The adversary can use the last query either once or twice. If it is used twice, it is used at least once in the bottom row.

Case 1: Assume first, that the last query is used once and that it is used in the top row. Assume WLOG that it is used in 1TL. Since there are at most β queries that can be used in 1BL and also at most β queries for 1BR, the success probability is upper bounded for q queries by $q\beta^2/(2^n - q)$.
Now assume that the last query is used once and that it is used in the bottom row. Whether it is used in 1BL or 1BR, the success probability in each case for one query is $\leq 1/(2^n - q)$. So the total success probability for q queries for this case is upper bounded by $q(2+\beta^2)/(2^n - q)$.

Case 2: Now, assume that the last query is used twice. So it is used exactly once in the bottom row and the analysis of Case 1 (bottom row) gives an upper bound of $2q/(2^n - q)$.

◻

3.5 Proof of Theorem 1

The proof of Theorem 1 now follows with Proposition 1 by adding up the individual results from Propositions 2 - 4. Proposition 5 is given in Appendix A, Propositions 6 and 7 in Appendix B and Proposition 8 in Appendix C.

4 Conclusion

We have derived the first collision security bound for MDC-4, a double length block cipher based compression function which takes 4 calls to hashing a message block using a (n, n) block-cipher. Although MDC-4 is structurally quite different from MDC-2, it is somewhat surprising that the result given by Steinberger for MDC-2 ($2^{74.91}$) and our result for MDC-4 ($2^{74.76}$) are numerically quite similar – although we have applied much more economical proof techniques. This leads to open questions we have not been able to find satisfying answers for as, *e.g.*, *why are these results so similar?* One possible answer is, that MDC-2 and MDC-4 are security-wise very similar. This would lead to the conclusion that MDC-4 is totally dominated by MDC-2. Another answer might be that the limitations are due to the applied techniques in the proofs. Then it would be interesting and important to find new proof methods that help overcome these.

References

[1] Eli Biham and Orr Dunkelman. The SHAvite-3 Hash Function. Submission to NIST (Round 2), 2009.

[2] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 320–335. Springer, 2002.

[3] Gilles Brassard, editor. *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*. Springer, 1990.

[4] Ivan Damgård. A Design Principle for Hash Functions. In Brassard [3], pages 416–427.

[5] Bert den Boer and Antoon Bosselaers. Collisions for the Compressin Function of MD5. In *EUROCRYPT*, pages 293–304, 1993.

[6] Shimon Even and Yishay Mansour. A Construction of a Cipher From a Single Pseudorandom Permutation. In Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, editors, *ASIACRYPT*, volume 739 of *Lecture Notes in Computer Science*, pages 210–224. Springer, 1991.

[7] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The Skein Hash Function Family. Submission to NIST (Round 2), 2009.

[8] Ewan Fleischmann, Christian Forler, Michael Gorski, and Stefan Lucks. Collision Resistant Double-Length Hashing. In *ProvSec*, pages 102–118, 2010.

[9] Ewan Fleischmann, Michael Gorski, and Stefan Lucks. Security of Cyclic Double Block Length Hash Functions, booktitle = IMA Int. Conf. pages 153–175, 2009.

[10] H. Dobbertin. The status of MD5 after a recent attack, 1996.

[11] Hans Dobbertin, Anton Bosselaers, Bart Preneel. RIPEMD (RACE integrity primitives evaluation message digest), 1996.

[12] Joe Kilian and Phillip Rogaway. How to Protect DES Against Exhaustive Key Search. In Neal Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 252–267. Springer, 1996.

[13] Lars R. Knudsen, Florian Mendel, Christian Rechberger, and Søren S. Thomsen. Cryptanalysis of MDC-2. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 106–120. Springer, 2009.

[14] Lars R. Knudsen and Bart Preneel. Fast and Secure Hashing Based on Codes. In *CRYPTO*, pages 485–498, 1997.

[15] Matthias Krause, Frederik Armknecht, and Ewan Fleischmann. Preimage Resistance Beyond the Birthday Bound: Double-Length Hashing Revisited. Cryptology ePrint Archive, Report 2010/519, 2010. `http://eprint.iacr.org/`.

[16] Xuejia Lai and James L. Massey. Hash Function Based on Block Ciphers. In *EUROCRYPT*, pages 55–70, 1992.

[17] Jooyoung Lee and Daesung Kwon. The security of abreast-dm in the ideal cipher model. Cryptology ePrint Archive, Report 2009/225, 2009. `http://eprint.iacr.org/`.

[18] Jooyoung Lee and Martijn Stam. MJH: A Faster Alternative to MDC-2. In Aggelos Kiayias, editor, *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 213–236. Springer, 2011.

[19] Jooyoung Lee, Martijn Stam, and John Steinberger. The collision security of tandem-dm in the ideal cipher model. Cryptology ePrint Archive, Report 2010/409, 2010. `http://eprint.iacr.org/`.

[20] Jooyoung Lee, Martijn Stam, and John Steinberger. The preimage security of double-block-length compression functions. Cryptology ePrint Archive, Report 2011/210, 2011. `http://eprint.iacr.org/`.

[21] Gatan Leurent, Charles Bouillaguet, and Pierre-Alain Fouque. SIMD Is a Message Digest. Submission to NIST (Round 2), 2009.

[22] M. Rabin. Digitalized Signatures. *In R. DeMillo, D. Dobkin, A. Jones and R.Lipton, editors, Foundations of Secure Computation, Academic Press*, pages 155–168, 1978.

[23] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.
[24] Ralph C. Merkle. One Way Hash Functions and DES. In Brassard [3], pages 428–446.
[25] C.H. Meyer and M. Schilling. Secure program load with manipulation detection code. In *SECURICOM'88*, pages 111–130, France, 1988. Paris.
[26] National Bureau of Standards. *FIPS Publication 46-1: Data Encryption Standard*, January 1988.
[27] NIST National Institute of Standards and Technology. FIPS 180-1: Secure Hash Standard. April 1995. See http://csrc.nist.gov.
[28] NIST National Institute of Standards and Technology. FIPS 180-2: Secure Hash Standard. April 1995. See http://csrc.nist.gov.
[29] Ronald L. Rivest. The MD4 Message Digest Algorithm. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 303–311. Springer, 1990.
[30] Ronald L. Rivest. *RFC 1321: The MD5 Message-Digest Algorithm.* Internet Activities Board, April 1992.
[31] John P Steinberger. The Collision Intractability of MDC-2 in the Ideal Cipher Model. Cryptology ePrint Archive, Report 2006/294, 2006. `http://eprint.iacr.org/`.
[32] John P. Steinberger. The Collision Intractability of MDC-2 in the Ideal-Cipher Model. In *EUROCRYPT*, pages 34–51, 2007.
[33] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Cryptanalysis of the Hash Functions MD4 and RIPEMD. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2005.
[34] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.
[35] Robert S. Winternitz. A Secure One-Way Hash Function Built from DES. In *IEEE Symposium on Security and Privacy*, pages 88–90, 1984.

A Lucky($\mathcal{Q}$)

Proposition 5. *Let $n, q \in \mathbb{N}$, $n \geq q$. Let α, β, and γ be as in Theorem 1 with $eq2^{n/2}/(2^n - q) \leq \alpha$ and $eq/(2^n - q) \leq \beta$. Set $\tau = \frac{\alpha(2^n-q)}{q2^{n/2}}$ and $\nu = \frac{\beta(2^n-q)}{q}$. Then*

$$\Pr[\textsc{Lucky}(\mathcal{Q})] \leq \frac{q^2}{\gamma(2^n - q)} + 2q2^{n/2} e^{q2^{n/2}\tau(1-\ln\tau)/(2^n-q)} + q2^n e^{q2^n\nu(1-\ln\nu)/(2^n-q)}.$$

A proof can be found in [31, Appendix B].

B CollLeftColumns($\mathcal{Q}$) and CollRightColumns($\mathcal{Q}$)

Proposition 6 (CollLeftColumns($\mathcal{Q}$)).

$$\Pr[\textsc{CollLeftColumns}(\mathcal{Q})] \leq q\left(\frac{\alpha^2(\alpha^2 + 2\gamma + 1) + \alpha\gamma + \alpha}{2^n - q} + \frac{\alpha^3 + 2\alpha^2 + \alpha}{(2^{n/2} - \alpha)^2}\right)$$

Proof. We first assume that the adversary uses the last query once for a collision. So, by symmetry, Q_i is either used in 1TL, 1TR or 1BL, but not in 1BR since 1BR = 2BR. The success probability for this case is upper bounded by Lemma 3.

Now assume that the adversary uses the last query twice or more for a collision. By Lemma 4, we upper bound the two cases where Q_i is at least used in either 1TL and 2TL or in 1TR and 2TR.

Finally, Lemma 5 upper bounds the case where the last query is used in 1TL but not in 2TL and the case where the last query is used in 1TR but not in 2TR. Since there are no cases left, the union bound gives our claim. $\square$

Lemma 3. *Let $S \neq T$, $S' \neq T'$, $\mathcal{Q}_{i-1}$ the query list not containing the last query Q_i,* 1BL $\neq$ 2BL *and* 1BR $=$ 2BR. *Assume that the last query Q_i is used once in the* MDC-4 *compression function H^E. Then*

$$\Pr[H^E(S,T,M) = H^E(S',T',M')] \leq q\frac{\alpha^2(\alpha^2+2\gamma)}{2^n-q}$$

Proof. We can WLOG assume that the last query is used in word 1.

Case 1: The last query is used in position 1BL. We now upper bound the number of queries that can be used in 2BL.
The key K_i of $Q_i = (K_i, X_i, Y_i)$ uniquely determines O^L and P^R. There are at most α possible choices of O^R and at most α possible choices for P^L. Since $R^L = P^L$ and $Q^R = O^R$ we can use the same argument again and upper bound the number of possible values for Q^L by α^2 (since there are α possible values of Q^R) and state the same bound for the number of possible values for R^R. So there are at most α^4 queries that can be used in 2BL as the plaintext input is also fixed by the choice of the query 2TR. For q queries, the total probability of success is then upper bounded by $q\alpha^4/(2^n-q)$.

Case 2: The last query is used in position 1TL. Since the XOR-values of the queries 1BL and 2BL collide, there are most γ possible pairs of queries such that $U = U'$. So there are at most γ possible values for O^L, P^R. For each O^L, the values of R^R and Q^L are also uniquely determined. For a fixed query pair (1BL, 2BL), there are α possible queries for 2TL and α possible queries for 2TR and therefore α^2 possible queries for 2BR. Since 1BR = 2BR there are in total at most $\alpha^2 \cdot \gamma$ that can be used for $(O^L||O^R)$ and therefore, for q queries the total probability of success is upper bounded by $q\gamma\alpha^2/(2^n-q)$.

Case 3: The last query is used in position 1TR. The same arguments as in Case 2 can be applied delivering the same bound.

□

Lemma 4. *Let H^E be a* MDC-4 *compression function and assume* 1BR $=$ 2BR. *By $\mathcal{Q}_{i-1}$ we denote the query list not containing the last query Q_i. Further assume that Q_i is either used in position* 1TL *and* 2TL *or in position* 1TR *and* 2TR. *The query Q_i may be used in other positions as well. Then*

$$\Pr[H^E(S,T,M) = H^E(S',T',M')] \leq q\left(\frac{\alpha\gamma+\alpha^2}{2^n-q} + \frac{2\alpha^2}{(2^{n/2}-\alpha)^2}\right)$$

where the probability is measured over arbitrary inputs of H^E.

Proof. We only discuss the case where the last query is at least in $1\text{TL}, 2\text{TL}$ since the case where it is used in $1\text{TR}, 2\text{TR}$ is essentially the same. The bound is derived by doubling our result, *i.e.*, we apply the union bound.

Since 1TL $=$ 2TL, it follows 1TR $\neq$ 2TR. If there is such a collision, then the adversary must have found four queries previously for (i) $1\text{TR}, 1\text{BL}, 2\text{TR}, 2\text{BL}$ or (ii) $1\text{TL}, 1\text{BL}, 2\text{TL}, 2\text{BL}$ such that the XOR-output of 1BL and 2BL is equal, *i.e.*, a collision has been found in the left column. It suffices to upper bound (i) since - due to the symmetric structure of MDC-4 - (i) and (ii) are equivalent.

Case 1. The last query is used once. We WLOG assume that it is used in either 1TR or 1BL.

Subcase 1.1. The last query is used in 1TR. As the queries in the bottom row, 1BL and 2BL, collide, there are at most γ pairs of queries in $\mathcal{Q}_{i-1}$ that can be used. For any query in 2BL, there are at most α matching queries for 2TR. As the output of the last query is fully determined by 1BL and 2TR– since 1BR = 2BR and therefore their inputs are also equal – the last query has a chance of being successful $\leq \alpha\gamma/(2^n - q)$ and for q queries the total chance of success is $\leq q\alpha\gamma/(2^n - q)$.

Subcase 1.2. The last query is used at position 1BL. By the key input of this query, there are at most α queries that can be used in 1TR. For each query in 1TR, there are at most α possibilities for queries in 2TR as the queries in 1TR and 2TR share the left XOR-output. Since 1TL = 2TL, There are α^2 possible key inputs for 2BL. The plaintext input of 2BL is also uniquely determined by the chosen query 2TR. So for q queries the total chance of success is $\leq q\alpha^2/(2^n - q)$.

Case 2. The last query is used twice or more in $1\mathrm{TR}, 1\mathrm{BL}, 2\mathrm{TR}, 2\mathrm{BL}$. So the last query must be used exactly twice, otherwise we would have already had a collision, because in that case is S=S', M=M', and T=T' . We assume WLOG that it is either used in $1\mathrm{TR}, 1\mathrm{BL}$ or in $1\mathrm{TR}, 2\mathrm{BL}$.

Subcase 2.1: The last query is used in 1TR and 1BL. The right half of the query output must match the key input. The success is upper bounded by $\leq 1/(2^{n/2} - \alpha)$. If the right half is successful, then there are at most α queries in 2BL that have the same right half XOR-output as the last query. And for any query in 2BL there are again at most α queries that can be used for 2TR. Since the left half of the XOR-output of the last query must match the left half of the XOR-output of 2TR, the chance can upper bounded by $\alpha^2/(2^{n/2} - \alpha)$. So for q queries, the total chance of success is $\leq q\alpha^2/((2^{n/2} - \alpha)^2)$.

Subcase 2.2: The last query is used in 1TR and 2BL. Then there are at most α queries that can be used in 2TR given the key input of the last query. Since the left half of the XOR-output of 2TR must be equal to the left half of the XOR-output of 1TR, the chance of success in this case is upper bounded by $\alpha/(2^{n/2} - \alpha)$. If this half is successful, then there are at most α values that can be used in 1BL, so the right half of the XOR-output has a chance of success of $\leq \alpha/(2^{n/2} - \alpha)$. For q queries, the total chance of success is $\leq q\alpha^2/(2^{n/2} - \alpha)^2$. □

Lemma 5. *Let H^E be a* MDC-4 *compression function and assume* 1BR = 2BR *and* 1BL ≠ 2BL. *By $\mathcal{Q}_{i-1}$ we denote the query list not containing the last query Q_i. Further assume that Q_i is either used in position* 1TL *but not* 2TL *or in position* 1TR *but not* 2TR. *The query Q_i may be used in other positions as well. Then*

$$\Pr[H^E(S,T,M) = H^E(S',T',M')] \leq q\left(\frac{\alpha + \alpha^3}{(2^{n/2} - \alpha)^2} + \frac{\alpha}{2^n - q}\right)$$

where the probability is measured over arbitrary inputs of H^E.

Proof. We upper bound the case where the last query is used in 1TL but not in 2TL since the other case is equivalent. As always, we assume 1TL ≠ 1TR. Note that the last query cannot be used in 1BR and 2BR since the case would have been upper bounded by Lemma 4. By this Lemma we also have already upper bounded the case where the last query is only used in 1TL. So the following three cases remain to be upper bounded.

Case 1: The last query also appears in position 1BL. The success chance that the left half of the XOR-output matches the left half of the key input is $\leq 1/(2^{n/2} - \alpha)$. Now assume that

that this 'match' is successful, *i.e.*, our discussion now is based on this fact. Since the XOR-outputs of 1BL and 2BL must be equal, we already know the left half of the XOR-output of 2BL. There are at most α matching queries that can be used in 2BL in order to form a collision. So (assuming our fact) the probability of a collision is $\leq \alpha/(2^{n/2}-\alpha)$. For q queries, the total chance of success for an adversary is $\leq q\alpha/(2^{n/2}-\alpha)^2$.

Case 2: The last query also appears in position 2BL. Given the key input of 2BL, there are at most α different queries for position 2TL and α different queries for 2TR. So the number of queries for 2BR is upper bounded by α^2. Since 1BR = 2BR the right half of the XOR-output of 1BL has a chance of success of $\alpha^2/(2^{n/2}-\alpha)$. Conditioned on the fact that this right half is successful we have also fixed the right half of the XOR-output of 2BL. So there are at most α possible queries for 1BR and the left half of the output of 1TL has a chance of success of $\leq \alpha/(2^{n/2}-\alpha)$. For q queries our bound is therefore $\leq q\alpha^3/(2^{n/2}-\alpha)^2$.

Case 3: The last query also appears in position 2TR, and does not appear in position 1BL or 2BL, else we can revert to Case 1 or Case 2. Then there are at most α possibilities for position 1BL and 2BL. So the last query has chance of succeeding of $\leq \alpha/(2^n-q)$ and for q queries $\leq q\alpha/(2^n-q)$.

Proposition 7 (CollRightColumns($\mathcal{Q}$)).

$$\Pr[\textsc{CollRightColumns}(\mathcal{Q})] \leq q\left(\frac{\alpha^2(\alpha^2+2\gamma+1)+\alpha\gamma+\alpha}{2^n-q}+\frac{\alpha^3+2\alpha^2+1}{(2^{n/2}-\alpha)^2}\right)$$

Proof. Due to the symmetric structure of MDC-4 this proof is essentially the same as for proposition 6. □

C CollBothColumns($\mathcal{Q}$)

Proposition 8 (CollBothColumns($\mathcal{Q}$)).

$$\Pr[\textsc{CollBothColumns}(\mathcal{Q})] \leq q\left(\frac{\gamma\alpha^2+\gamma^2}{2^n-q}+\frac{2\alpha}{(2^{n/2}-\alpha)^2}\right)$$

Proof. In case 1, we discuss the implication if the last query is only used once, the cases 2-4 give bounds if the last query is used at least twice.

Case 1: The last query is used exactly once. We can WLOG assume the it is either used in 1TL or 1BL.

Subcase 1.1: The last query is used in position 1BL. Since 1BR = 2BR, there are at most γ pairs of queries in the query history that can be used for position 1BR, 2BR. Now, for any one query 2BR, there are at most α matching queries in position 2TL and at most α matching queries in 2TR. Since the queries in 2TL and 2TR uniquely determine the query 2BL, there are at most $\gamma\alpha^2$ queries that can be used for 2BL. Therefore the last query has a chance of being successful of $\leq \gamma\alpha^2/(2^n-q)$. For q queries, the total chance of success in this case is $\leq q\gamma\alpha^2/(2^n-q)$.

Subcase 1.2: The last query is used in position 1TL. There are at most γ possible pairs of queries that can be used for 1BL and 2BL and there are at most γ possible queries that can be used for 1BR and 2BR. We now upper bound the probability that the last query can be used in 1TL assuming a collision. There are at most γ^2 pairs of queries that can be used for 1BL and 1BR. Therefore the success probability of the last query can be upper bounded by $\leq \gamma^2/(2^n-q)$ and for q queries by $q\gamma^2/(2^n-q)$.

Case 2: The last query is only used in the bottom row. Then it is used exactly twice, WLOG in positions 1BL and 2BR. This would imply $U = V'$ which then – in the case of success – implies $\textsc{InternalColl}(\mathcal{Q})$.

Case 3: The last query is only used in the top row. We can WLOG assume it is used in 1TL. We can use the same reasoning as in Subcase 1.2 and therefore extend Subcase 1.2 to also handle this slightly more general situation here.

Case 4: The last query is used at least once in the bottom row and at least once in the top row. We can WLOG assume that it is used in position 1TL. Using the same argument as for Case 2, the last query must then appear exactly once in the bottom row. The following four subcases discuss the implications of the last query being also used in 1BL, 1BR, 2BL and 2BR. Note that the adversary may use it also a second time – apart from 1TL– in the top row but this does not change our bounds.

Subcase 4.1: The last query is also used in 1BL. The left half of the XOR-output of 1TL has a chance of being equal to its key input (*i.e.*, the key input of 1BL) of $\leq 1/(2^{n/2} - \alpha)$. The following analysis is now based on the fact the the left half of the XOR-output has matched the left half of the key input. Since we now also know the left half of the XOR-output of 2BL, there are at most α queries that can be used in 2BL. The chance that the right half of the XOR-output of 2BL matches the right half of the XOR-output of 1BL is therefore $\leq \alpha/(2^{n/2} - \alpha)$. So for q queries the total chance of success is $\leq q\alpha/(2^{n/2} - \alpha)^2$.

Subcase 4.2: The last query is also used in 1BR. The same arguing as for Subcase 4.1 can be used (apart from exchanging 'left' and 'right') and the bound for q queries is again $\leq \alpha/(2^{n/2} - \alpha)^2$.

Subcase 4.3 The last query is also used in position 2BL. There are at most γ possible pairs of query in the query history that can be used for the pair 1BR, 2BR that form a collision. The probability that the right half of the XOR-output of 1TL matches the right half of its key input (*i.e.*, for the last query being also used in 1BR) is $\leq 1/(2^{n/2} - \alpha)$. Conditioned on the fact that the right half of the XOR-output is now fixed there are at most α queries that can be used in 1BL such that the XOR-outputs of 1BL and 2BL collides. The probability that the left half of the XOR-output of 1TL is equal to the left half of the key of 1BL is therefore $\leq \alpha/(2^{n/2} - \alpha)$ and the total chance of success for q queries is $\leq q\alpha/(2^{n/2} - \alpha)^2$.

Subcase 4.4 The last query is also used in 2BR. The same arguing as for Subcase 4.3 can be used (apart from exchanging 'left' and 'right') and the bound for q queries is again $\leq q\alpha/(2^{n/2} - \alpha)^2$. □