Weimar-DM: A Highly Secure Double-Length Compression Function

Ewan Fleischmann, Christian Forler, Stefan Lucks, and Jakob Wenzel

Bauhaus-Universität Weimar, Germany
{ewan.fleischmann,christian.forler,stefan.lucks,
jakob.wenzel}@uni-weimar.de

Abstract. We present Weimar-DM, a double length compression function using two calls to a block cipher with $2n$-bit key and n-bit block size to compress a $3n$-bit string to a $2n$-bit one. For Weimar-DM, we show that for $n = 128$, no adversary asking less than $2^{n-1.77} = 2^{126.23}$ queries can find a collision with probability greater than $1/2$. This is the highest collision security bound ever shown for such a compression function. Even more important, our security analysis is much simpler than that for comparable functions as, *e.g.*, Tandem-DM, Abreast-DM or Hirose-DM. We also give a preimage security analysis of Weimar-DM showing a near-optimal bound of $2^{2n-5} = 2^{251}$ queries. Our security bounds are asymptotically optimal.

Keywords: double length compression function, block cipher based, ideal cipher model, collision security, preimage security.

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 [26].

Block Cipher-Based Hash Functions. Since their initial design by Rivest, MD4-family hash functions (*e.g.*, MD4, MD5, RIPEMD, SHA-1, SHA2 [4,29,30,32,33]) have dominated cryptographic practice. But in recent years, a sequence of attacks on these type of functions [8,12,41,42] has led to a generalized sense of concern about the MD4-approach. The most natural place to look for an alternative is in block cipher-based constructions, which in fact predate the MD4-approach [25]. Another reason for the resurgence of interest in block cipher-based hash functions is due to the rise of size restricted devices such as RFID tags or smart cards: A hardware designer has to implement only a block cipher in order to obtain an encryption function as well as a hash function.

But since the output length of most practical encryption functions is far too short for a collision resistant hash function, *e.g.*, 128-bit for AES, one is mainly interested in sound design principles for *double block length* (DL) hash functions

W. Susilo, Y. Mu, and J. Seberry (Eds.): ACISP 2012, LNCS 7372, pp. 152–165, 2012.

Table 1. Comparison of double length compression function security results evaluated for $n = 128$ and a success probability of 1/2; for CYCLIC-DM $k > 1$, *i.e.*, the cycle length > 2 the value of k' is ≥ 2

compression function	collision bound	preimage bound
WEIMAR-DM	$2^{126.23}$ (this paper)	$2^{252.5}$ (this paper)
ABREAST-DM	$2^{124.42}$ [11,22]	2^{246} [1]
HIROSE-DM	$2^{124.55}$ [15]	2^{251} [1]
TANDEM-DM	$2^{120.87}$ [24]	2^{246} [1]
CYCLIC-DM (cycle length > 2)	2^{127-k} [11]	$\approx 2^{128}$ [11,22]
CYCLIC-DM (cycle length 2)	$2^{124.55}$ [11]	$\approx 2^{128}$ [11,22]
CUBE-DM	$2^{125.56}$ [11]	$\approx 2^{128}$ [11,22]
ADD/K-DM	$2^{127-k'}$ [11]	$\approx 2^{128}$ [11,22]
Lee/Kwon	$2^{125.0}$ [22]	$\approx 2^{128}$ [11,22]

[2]. A DL hash function uses a block cipher with n-bit output as the building block by which it maps possibly long strings to $2n$-bit ones. Usually, hash functions are built using compression functions only being able to compress a fixed length input into a (smaller) fixed-length output. These compression functions are iterated, *e.g.*, using the Merkle-Damgård [7,27] transform, in order to get a full-fledged hash function. Since these transforms are property preserving, this article focuses only on the compression function.

WEIMAR-DM. We define a new double length double call compression function as follows (cf. Figure 1).

Definition 1. *Let E be a block cipher taking an $2n$-bit key and an n-bit block size. The compression function $H^{\mathrm{WDM}} : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ is defined as (cf. Figure 1)*

$$H^{\mathrm{WDM}}(M, U, \widehat{U}) = \left(E_{M\|U}(\widehat{U}) \oplus \widehat{U}, E_{\overline{M\|U}}(\widehat{U}) \oplus \widehat{U} \right),$$

where $\overline{M\|U}$ denotes the bit-by-bit complement of the bit-string $M\|U$.

In this paper we give very tight collision security and preimage security bounds for WEIMAR-DM. Table 1 gives an overview on known double length compression function designs using two calls to a block cipher with $2n$-bit key and n-bit block size inside. The results obtained in this paper for WEIMAR-DM have also been included.

Our Contribution. We present a new and surprisingly simple design of a double length double call compression function (WEIMAR-DM) and give a collision security bound as well as a preimage security bound. It has the best collision

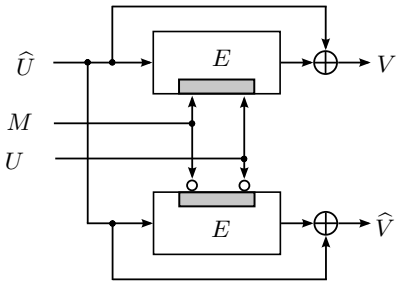

Fig. 1. WEIMAR-DM compression function H^{WDM}; the small circle '$\circ$' denotes a bit-by-bit complement

security bound of all known double length double call compression functions using a block cipher with $2n$-bit key and n-bit block size. Also, no compression function has a tighter preimage security bound, only for HIROSE-DM a comparable one is known. The collision security proof not only delivers an ultra-tight bound, but is also very short.

Outline. The paper is organized as follows: Section 2 gives formal notations and definitions. In Section 3, we prove that any adversary asking less than $2^{126.23}$ oracle queries has negligible advantage in finding a collision for the WEIMAR-DM compression function. Section 4 derives a near-optimal preimage bound for WEIMAR-DM. In Section 5 we discuss our results and conclude. Directly related publications have been mentioned in Table 1, a broader overview on block-cipher based hashing is provided in Appendix A.

2 Preliminaries

2.1 Basic Notions

Ideal Cipher Model. A (k, n) *block cipher* is a keyed family of permutations consisting of two paired algorithms $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ and $E^{-1} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, both accepting a key of size k bits and an input block of size n bits for some $k, n > 0$. For positive k, n, $\textsc{Block}(k, n)$ is the set of all (k, n) block ciphers. For any $E \in \textsc{Block}(k, n)$ and any fixed key $K \in \{0,1\}^k$, decryption $E_K^{-1} := E^{-1}(K, \cdot)$ is the inverse function of encryption $E_K := E(K, \cdot)$, so that $E_K^{-1}(E_K(X)) = X$ holds for any admissible input $X \in \{0,1\}^n$.

Most of the attacks on hash functions based on block ciphers do not utilize the internal structure of the block ciphers. The security of such hash functions is usually analyzed in the *ideal cipher model* [2,9,18]. In this model, the underlying primitive, the block cipher E, is modeled as a family of random permutations $\{E_K\}$ whereas the random permutations are chosen independently for each key K, *i.e.* formally E is selected randomly from $\mathrm{BC}(\mathcal{X}, \mathcal{K})$.

Block Cipher Based Compression Functions. Generally speaking, a *single length* (SL) block cipher based compression function is a compression function $H^{\mathrm{SL}} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ using a block cipher with n-bit block size inside. The idea was first discussed in literature by Rabin [25]. Most SL functions use a block cipher from $\textsc{Block}(n,n)$ and compress a $2n$ bit string to an n bit string. A popular example is the Davies-Meyer (DM) [43] mode

$$H(M,U) = E_M(U) \oplus U,$$

which is essentially used twice inside WEIMAR-DM. The $\oplus$ operation is usually called *feed-forward*. A double (block) length (DL) compression function is a compression function $H^{\mathrm{DL}} : \{0,1\}^{k-n} \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ taking a $(k-n)$-bit message and a $2n$-bit chaining value and outputs a new $2n$-bit chaining value. It also uses a block cipher from $\textsc{Block}(k,n)$ inside. WEIMAR-DM as given in Definition 1 is an example of a double length compression function using exactly two calls to a block cipher from $\textsc{Block}(2n,n)$ in order to compute its output value.

2.2 Security Notions for Double Length Compression Functions

Security is quantified by the success probability of an optimal resource-bounded adversary. An adversary is a computationally unbounded but always-halting collision-finding algorithm $\mathcal{A}$ with resource-bounded access to an oracle $E \in \textsc{Block}(2n,n)$. We can assume (by standard arguments) that $\mathcal{A}$ is deterministic. The adversary may make *forward* queries $(X,K,?)_{fwd}$ to discover the corresponding value $Y = E_K(X)$, or the adversary may make *backward* queries $(?,K,Y)_{bwd}$, so as to learn the corresponding value $X = E_K^{-1}(Y)$ for which $E_K(X) = Y$. Either way the result of the query is stored in a triple (X_i, K_i, Y_i). The *query history*, denoted by $\mathcal{Q}$, is the tuple $(Q_1, \ldots, Q_q)$ where $Q_i = (X_i, K_i, Y_i)$ is the result of the i-th query made by the adversary and where q is the total number of queries made by the adversary. Without loss of generality, it is assumed that $\mathcal{A}$ 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.

As usual, we define the collision security of a hash function $\mathcal{H}$ by an experiment of an adversary $\mathcal{A}$ with a security parameter of $2n$, *i.e.* equal to the output bit-length of the compression function.

Experiment 1 (Collision-Finding Experiment $\texttt{Exp-Coll}_{\mathcal{A},H^{\mathrm{DL}}}(2n)$)

1. *The adversary $\mathcal{A}$ is given oracle access to a block cipher $E \in \textsc{Block}(k,n)$ and returns values $(M,U,\widehat{U}), (M',U',\widehat{U}') \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$.*
2. *The output of the experiment is defined to be 1 iff $(M,U,\widehat{U}) \neq (M',U',\widehat{U}')$ and $H^{\mathrm{DL}}(M,U,\widehat{U}) = H^{\mathrm{DL}}(M',U',\widehat{U}')$. In such a case we say that $\mathcal{A}$ has found a collision for H^{DL}.*

The advantage of an adversary $\mathcal{A}$ finding such a collision of H^{DL} is given in the following definition.

Definition 2. $\mathbf{Adv}^{\mathrm{COLL}}_{H^{\mathrm{DL}}}(\mathcal{A}) = \Pr\left[\texttt{Exp-Coll}_{\mathcal{A},H^{\mathrm{DL}}}(2n) = 1\right]$.

Since we only limit the adversary by the number of queries it is allowed to ask to the E oracle, *i.e.* it is explicitly given 'unlimited computing power', we write

$$\mathbf{Adv}^{\mathrm{COLL}}_{H^{\mathrm{DL}}}(q) := \max_{\mathcal{A}}\{\mathbf{Adv}^{\mathrm{COLL}}_{H^{\mathrm{DL}}}(\mathcal{A})\},$$

where the maximum is taken over all adversaries that ask at most q oracle queries in total.

There are several notions known that formalize *preimage security* [34]. We adopt *everywhere preimage resistance* (EPRE) in the information theoretic setting which essentially lets the adversary pre-commit to the hash value it likes to be challenged on *before* submitting any queries to the oracle. The corresponding preimage finding experiment is definied as follows.

Experiment 2 (Preimage-Finding Experiment $\texttt{Exp-Epre}_{\mathcal{A},H^{\mathrm{DL}}}(2n)$)

1. *The adversary $\mathcal{A}$ is given oracle access to a block cipher $E \in \textsc{Block}(k,n)$. $\mathcal{A}$ selects and announces a value $(V,\widehat{V}) \in \{0,1\}^n \times \{0,1\}^n$ before making any oracle queries. It outputs a value $(M,U,\widehat{U}) \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$.*
2. *The output of the experiment is defined to be 1 iff $H^{\mathrm{DL}}(M,U,\widehat{U}) = (V,\widehat{V})$. In such a case we say that $\mathcal{A}$ has found a preimage of H^{DL}.*

Now, we let $\mathbf{Adv}^{\mathrm{EPRE}}_{H^{\mathrm{DL}}}(\mathcal{A})$ be the predicate that is true iff '1' is returned by the experiment $\texttt{Exp-Epre}_{\mathcal{A},H^{\mathrm{DL}}}(2n)$. The pre-committed value $(V,\widehat{V})$ is an omitted parameter of $\mathbf{Adv}^{\mathrm{EPRE}}_{H^{\mathrm{DL}}}(\mathcal{A})$. Again, we define

$$\mathbf{Adv}^{\mathrm{EPRE}}_{H^{\mathrm{DL}}}(q) := \max_{\mathcal{A}}\{\mathbf{Adv}^{\mathrm{EPRE}}_{H^{\mathrm{DL}}}(\mathcal{A})\},$$

where the maximum is taken over all adversaries that ask at most q oracle queries in total.

3 Collision Security Analysis of Weimar-DM

3.1 Security Results

It is easy to see that H^{WDM} is of type Cyclic-DM with a cycle length of 2, *i.e.*, we directly have a collision security bound of $2^{124.55}$ (cf. Table 1). So we are done with our analysis. But we do not use this generic proof technique but rather use a specialized one delivering us a number of benefits. First, our proof is way simpler than the generic proof for Cyclic-DM. And, second, our new collision security bound is much better by virtually halving the gap between the theoretically optimal bound known before (via Cyclic-DM) and the best bound theoretically possible ($\approx 2^{127}$). Our main collision security result is stated in the following theorem.

Theorem 1. *Let* $N = 2^n$. *Then,* $\mathbf{Adv}^{\mathrm{COLL}}_{H^{\mathrm{WDM}}}(q) \leq \frac{q(q+1)}{(N-2q)^2}$.

In numerical terms, *e.g.*, for $n = 128$ and $\mathbf{Adv}^{\mathrm{COLL}}_{H^{\mathrm{WDM}}}(q) = 1/2$, we have $q = 2^{126.23}$. Using simple calculus, it is easy to see that for $\alpha = N(1 - \frac{1}{\sqrt{2}}) = 2^{n-1.77}$ we have

$$\mathbf{Adv}^{\mathrm{COLL}}_{H^{\mathrm{WDM}}}(\alpha) = \frac{1}{2} + o(1),$$

where the term $o(1) \to 0$ for $n \to \infty$. Neglecting constant factors, our security bound reads as an asymptotically optimal bound of $O(2^n)$ for a compression function with $2n$-bit output.

3.2 Proof of Theorem 1

We assume that the adversary has made any relevant query to E to come up with a collision – which is reasonable in the ideal cipher model. Another standard assumption made in ideal cipher proofs is that "the adversary never makes a query to which it already knows the answer". By this it is meant, for example, that one can assume that the adversary never makes a query $E_K(X)$, obtaining an answer Y, and then makes the query $E_K^{-1}(Y)$ (which will necessarily be answered by X). We start by considering an arbitrary q-query collision finding adversary $\mathcal{A}$. We then construct an adversary $\mathcal{A}'$ which simulates $\mathcal{A}$ but does sometimes ask an additional query to the E oracle under certain circumstances.

Since $\mathcal{A}'$ is more powerful than $\mathcal{A}$, it suffices to upper bound the success probability of A'. We now give a detailed description of $\mathcal{A}'$ by simultaneously upper bounding its chances of success. We say that an adversary is *successful* if its query history $\mathcal{Q}$ contains the means of computing a collision. This is discussed more thoroughly in the following case analysis.

Description of A'. The adversary $\mathcal{A}'$ maintains an initially empty list $\mathcal{L}$ representing any possible input/output of the compression function H^{WDM} that can be computed by the adversary $\mathcal{A}$. An entry $L \in \mathcal{L}$ is a 4-tuple $(K, X, Y, Y') \in \{0,1\}^{5n}$ where $K \in \{0,1\}^{2n}$, $X \in \{0,1\}^n$ is the $3n$-bit input to the compression function such that $(M, U) = K$ and $\widehat{U} = X$. The n-bit values Y, Y' are given by $Y = E_K(X)$, $Y' = E_{\overline{K}}(X)$.

The list is now built as follows. Say that the adversary $\mathcal{A}$ mounts its i-th query to E or E^{-1}, $1 \leq i \leq q$. In the case of a forward query, the adversary gets hold of the tuple (K, X, Y) where $Y = E_K(X)$. In the case of a backward query, the adversary gets also hold of the tuple (K, X, Y), but in this case $X = E_K^{-1}(Y)$. In either case, the value $X \oplus Y$ is randomly determined by the output of the query.

Now, $\mathcal{A}'$ checks if an entry $L = (K, X, *, *)$ or $L' = (\overline{K}, X, *, *)$ is contained in $\mathcal{L}$ where $'*'$ denotes an arbitrary value. We now analyze the two possible cases $\mathcal{A}'$ might be confronted with and upper bound their success probabilities separately.

Case 1. Neither L nor L' are in $\mathcal{L}$. Then $\mathcal{A}'$ mounts an additional forward query $Y' = E_{\overline{K}}(X)$. Note that $Y' \oplus X$, the result of the 'bottom row' of the compression function, is always uniformly distributed since $K \neq \overline{K}$ always, *i.e.*, the results of the first query asked by the adversary $\mathcal{A}$ and the second query asked additionally by the adversary $\mathcal{A}'$ are always independently distributed. Set $L_i := (K, X, Y, Y')$. We append L_i to the list $\mathcal{L}$.

We now define what we mean by a collision in the list. Fix two integers r, s with $r \neq s$, such that $L_r = (K_r, X_r, Y_r, Y'_r)$ represents the r-th entry in $\mathcal{L}$ and $L_s = (K_s, X_s, Y_s, Y'_s)$ the s-th entry in $\mathcal{L}$ and both entries exist. We say that L_s and L_r collide if a collision of the compression functions occurs that can be computed using the query results given in L_r and L_s. This is the case if at least one of the following two conditions is met:

1. $Y_r \oplus X_r = Y_s \oplus X_s$ and $Y'_r \oplus X_r = Y'_s \oplus X_s$ or
2. $Y_r \oplus X_r = Y'_s \oplus X_s$ and $Y'_r \oplus X_r = Y_s \oplus X_s$.

So for the i-th query, there are at most $i - 1$ entries in the list $\mathcal{L}$ that might collide with L_i. We can upper bound the probability of success of the i-th query by

$$\sum_{j=1}^{i-1} \frac{2}{(N-2q)(N-2q)} \leq \frac{2i}{(N-2q)(N-2q)}$$

As the adversary can ask at most q queries, the list $\mathcal{L}$ cannot contain more than q entries since for any adversary query at most one additional entry is added to the list $\mathcal{L}$ of $\mathcal{A}'$. So the total chance of success for q queries is

$$\leq \sum_{i=1}^{q} \frac{2i}{(N-2q)(N-2q)} = \frac{q(q+1)}{(N-2q)^2}.$$

In case of a collision in $\mathcal{L}$ we give the attack to the adversary.

Case 2. It is clear that, by design, it cannot happen that exactly one of the values L or L' is already in $\mathcal{L}$. So now assume that both values L, L' are already in $\mathcal{L}$. Then $\mathcal{A}'$ ignores this query, since we know that $\mathcal{A}$ has zero chance of winning since otherwise we would have given the attack to the adversary before. □

4 Preimage Security Analysis of Weimar-DM

4.1 Security Results

Preimage security results for double length compression function have 'historically' been limited by the birthday bound, mainly due to technical reasons. At Asiacrypt 2011 a new breakthrough result by Armknecht *et al.* [1] gave new techniques that enable preimage security results for double length compression function way beyond the birthday-barrier. For our preimage security proof of Weimar-DM, we adopt these methods. More precise, we show the following Theorem.

Theorem 2. *Let* $N = 2^n$. *Then,* $\mathbf{Adv}^{\mathrm{EPRE}}_{H^{\mathrm{WDM}}}(q) \leq 16q/N^2$.

It is easy to see that $\mathbf{Adv}^{\mathrm{EPRE}}_{H^{\mathrm{WDM}}}(2^{2n-5}) = 1/2$ and therefore our bound is asymptotically optimal for a $2n$-bit compression function.

4.2 Proof of Theorem 2

Parts of the proof closely follow the proofs of [1, Theorems 1 and 2]. Our security proof uses the notion of *free* queries. Formally, these can be modeled as queries which the adversary is *forced* to query (under certain conditions), but for which the adversary is not charged: they do not count towards the maximum of q queries which the adversary is allowed. However, these queries become part of the adversary's query history, just like other queries. In particular, the adversary is not allowed, later, to remake these queries "on its own" (due to the previously discussed assumption that the adversary never makes a query which it already owns).

Similar to our collision security analysis, we say the attacker *succeeds* or *finds a preimage* if its query history $\mathcal{Q}$ contains the means of computing a preimage of C, in the sense that there exist values $B \in \{0,1\}^{3n}$, $K_1, K_2 \in \{0,1\}^{2n}$ and $X_1, X_2, Y_1, Y_2 \in \{0,1\}^n$ such that both (X_1, K_1, Y_1) and (X_2, K_2, Y_2) are in the query history $\mathcal{Q}$, $H^{\mathrm{WDM}}(B) = C$ and the two queries used to evaluate $H^{\mathrm{WDM}}(B)$ are precisely $E_{K_1}(X_1)$ and $E_{K_2}(X_2)$. In this case, we also say $\mathcal{Q}$ *contains a preimage* of C. In the current context, where we consider adversaries making 2^n queries or more, the assumption that the adversary never makes a query where it knows the answer to, should be more precisely restated as "the adversary never makes a query that will result in a triple (X, K, Y) which is already present in the query history". (This latter assumption can be made without loss of generality using the fact that $E_K(\cdot)$ is a permutation.) Indeed, if an adversary has made $2^n - 1$ queries under a key K, the result of the last query under that key is predetermined, and thus the adversary "already knows" the answer to this query. However, one should not forbid the adversary from making this query, since the query may be necessary to complete the attack.

Let $(V, \widehat{V}) \in \{0,1\}^n \times \{0,1\}^n$ be the point to invert (chosen by the adversary before it makes any queries to E). We upper bound the probability that, in q queries, the adversary finds a point $(M, U, \widehat{U}) \in (\{0,1\}^n)^3$ such that $H^{\mathrm{WDM}}(M, U, \widehat{U}) = (V, \widehat{V})$.

When the adversary makes a (normal) *forward query* $E_{M\|U}(\widehat{U})$ we give it for free, also, the answer to the query $E_{\overline{M\|U}}(\widehat{U})$. Moreover when the adversary makes a (normal) *backward query* $E^{-1}_{M\|U}(R)$, resulting in an answer $\widehat{U} = E^{-1}_{M\|U}(R)$, we give it for free the answer to the forward query $E_{\overline{M\|U}}(\widehat{U})$. As discussed, we assume that the adversary never makes a query to which it knows the answer. Thus the elements of the adversary's query history $\mathcal{Q}$ can be paired into adjacent pairs of the form $(M\|U, \widehat{U}, R), (\overline{M\|U}, \widehat{U}, S)$. We call such a pair an *adjacent query pair*.

We now give further free queries to the adversary, in the fashion described next. After each adjacent query pair has been completed (namely, after the adversary has received the response to both its query and its associated free query, and after these have been placed in the query history), we check whether the key prefix used for the latest query is such that the (current) query history contains exactly $N/2$ adjacent query pairs with this key prefix. If so, we give *all* remaining adjacent query pairs under this key for free to the adversary. There will be exactly $N/2$ such query pairs. We insert these $N/2$ free query pairs into the query history pair-by-pair (to maintain, mostly for conceptual simplicity, the adjacent pair structure of the query history). We note that, after these free queries have been inserted into the query history, the adversary cannot make any more queries under this key prefix, since, the adversary is assumed never to make a query to which it knows the answer. When $N/2$ free query pairs are given to the adversary in the fashion just described, we say that a *super query* occurs. This can be summed up as follows.

Super Query. Given $N/2$ adjacent query pairs to E all using the same key $K \in \{0,1\}^{2n}$, all the remaining $N/2$ queries using the same key K and the remaining $N/2$ queries using key $\overline{K}$ are given for free.

We say that an adjacent query pair $(M\|U, \widehat{U}, R), (\overline{M\|U}, \widehat{U}, S)$ is *successful*, if $\widehat{U} \oplus R = V$ and $\widehat{U} \oplus S = \widehat{V}$, or if $\widehat{U} \oplus R = \widehat{V}$ and $\widehat{U} \oplus S = V$. Thus the adversary obtains a preimage of $(V, \widehat{V})$ precisely if it obtains a successful adjacent query pair. This can occur in one of two ways: either the winning query pair is part of a super query, or not. We let $\mathsf{SuperQueryWin}(\mathcal{Q})$ denote the event that the adversary obtains a winning query pair that is part of a super query, and $\mathsf{NormalQueryWin}(\mathcal{Q})$ the event that the adversary obtains a winning query pair of normal queries (either forward or backward). It thus suffices to upper bound

$$\Pr[\mathsf{SuperQueryWin}(\mathcal{Q})] + \Pr[\mathsf{NormalQueryWin}(\mathcal{Q})].$$

Here probabilities are taken (as usual) over the adversary's randomness (if any) and over the randomness of the ideal cipher.

We first upper bound $\Pr[\mathsf{NormalQueryWin}(\mathcal{Q})]$. Note that when the adversary makes, say, a forward query $E_{M\|U}(\widehat{U})$, at most $N/2 - 2$ queries (counting free queries) have been previously answered with the key $M\|U$, since otherwise a super query for the key $M\|U$ would have occurred. Thus the value $R = E_{M\|U}(\widehat{U})$ comes uniformly at random from a set of size at least $N/2 + 2 \geq N/2$, and there is chance at most $2/(N/2) = 4/N$ that either $\widehat{U} \oplus R = V$ or $\widehat{U} \oplus R = \widehat{V}$ (this is also true if $V = \widehat{V}$). If, say, $\widehat{U} \oplus R = V$, there is further chance at most $1/(N/2) = 2/N$ that the free query $E_{\overline{M\|U}}(\widehat{U})$ returns $\widehat{U} \oplus \widehat{V}$, since the answer to the free query comes uniformly at random from a set of size at least $N/2+1 \leq N/2$. Other cases (*e.g.* when $\widehat{U} \oplus R = \widehat{V}$, and when the adversary makes

a backward query $E^{-1}_{M\|U}(R)$) are similarly analyzed, showing that the adversary's chance of triggering the event $\mathsf{NormalQueryWin}(\mathcal{Q})$ at any given query is at most $(4/N)(2/N) = 8/N^2$. Since the adversary makes q queries total, we therefore have

$$\Pr[\mathsf{NormalQueryWin}(\mathcal{Q})] \leq 8q/N^2. \tag{1}$$

We now bound $\Pr[\mathsf{SuperQueryWin}(\mathcal{Q})]$. Assume that a super query is about to occur on keys $M\|U$ and $\overline{M\|U}$ meaning that the value of $E_{M\|U}(\cdot)$ and $E_{\overline{M\|U}}(\cdot)$ are already known on exactly $N/2$ points. Let us denote this set of points by $\mathcal{X}$ and let $\mathcal{Y} = E_{M\|U}(\mathcal{X})$ and $\mathcal{Y}' = E_{\overline{M\|U}}(\mathcal{X})$. Further let $\mathcal{R} = \{0,1\}^n \backslash \mathcal{X}$, $\mathcal{S} = \{0,1\}^n \backslash \mathcal{Y}$ and $\mathcal{S}' = \{0,1\}^n \backslash \mathcal{Y}'$. Clearly, $|\mathcal{X}| = |\mathcal{Y}| = |\mathcal{Y}'| = |\mathcal{R}| = |\mathcal{S}| = |\mathcal{S}'|$.

Now fix a point $R \in \mathcal{R}$ in the domain of the super query. We now estimate the probability that this point R induces a successful pair. This can only be the case if

1. $R \oplus V \in \mathcal{S}$ and $R \oplus \widehat{V} \in \mathcal{S}'$ or
2. $R \oplus \widehat{V} \in \mathcal{S}$ and $R \oplus \mathcal{V} \in \mathcal{S}'$.

The probability that $E_{M\|U}(R) = R \oplus V$ and $E_{M\|U}(R) = R \oplus \widehat{V}$ equals $1/(N/2)^2$. The same is true for the probability that $E_{M\|U}(R) = R \oplus \widehat{V}$ and $E_{M\|U}(R) = R \oplus V$. Thus the total probability to be successful in a super query is at most

$$2 \cdot N/2 \cdot \left(\frac{1}{N/2}\right)^2 = \frac{2}{N/2}.$$

Since at most $q/(N/2)$ super queries can ever occur, we have

$$\Pr[\mathsf{SuperQueryWin}(\mathcal{Q})] \leq 8q/N^2. \tag{2}$$

The sum of (1) and (2) gives our claim. □

5 Discussion and Conclusion

In this paper, we have presented WEIMAR-DM, a double length compression function. We have shown very tight collision security bounds and preimage security bounds. The collision security bound is currently the best known bound for any such compression functions known in literature. Also, no compression function with a tighter preimage security bound is known – only HIROSE-DM has a numerically similar bound. For our security benefits, we have to pay the price of two key-scheduler runs per compression function.

Although a lot of progress has been made in recent years in the field of double length hashing, a lot of open questions remain. Related to our analysis, it would be interesting to investigate if our techniques in the collision security proof can be generalized, *e.g.*, to a subclass of CYCLIC-DM. Another open problem is the design of conveniently secure compression functions only using a block cipher from $\textsc{Block}(n, n)$.

References

1. Armknecht, F., Fleischmann, E., Krause, M., Lee, J., Stam, M., Steinberger, J.: The Preimage Security of Double-Block-Length Compression Functions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 233–251. Springer, Heidelberg (2011)
2. Black, J.A., Rogaway, P., Shrimpton, T.: Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 320–335. Springer, Heidelberg (2002)
3. Bos, J.W., Özen, O., Stam, M.: Efficient Hashing Using the AES Instruction Set. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 507–522. Springer, Heidelberg (2011)
4. Bosselaers, A., Preneel, B. (eds.): RIPE 1992. LNCS, vol. 1007. Springer, Heidelberg (1995)
5. Brachtl, B., Coppersmith, D., Hyden, M.M., Meyer, C.H., Matyas, S.M., Oseas, J., Pilpel, S., Schilling, M.: Data authentication using modification detection codes based on a public one way encryption function. U.S. Patent No. 4,908,861, March 13 (1990)
6. Brassard, G. (ed.): CRYPTO 1989. LNCS, vol. 435. Springer, Heidelberg (1990)
7. Damgård, I.: A design principle for hash functions. In: Brassard [6], pp. 416–427
8. den Boer, B., Bosselaers, A.: Collisions for the Compression Function of MD-5. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 293–304. Springer, Heidelberg (1994)
9. Even, S., Mansour, Y.: A Construction of a Cipher From a Single Pseudorandom Permutation. In: Matsumoto, T., Imai, H., Rivest, R.L. (eds.) ASIACRYPT 1991. LNCS, vol. 739, pp. 210–224. Springer, Heidelberg (1993)
10. Fleischmann, E., Gorski, M., Lucks, S.: On the Security of Tandem-DM. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 84–103. Springer, Heidelberg (2009)
11. Fleischmann, E., Gorski, M., Lucks, S.: Security of Cyclic Double Block Length Hash Functions. In: Parker, M.G. (ed.) Cryptography and Coding 2009. LNCS, vol. 5921, pp. 153–175. Springer, Heidelberg (2009)
12. Dobbertin, H.: The status of MD5 after a recent attack (1996)
13. Hattori, M., Hirose, S., Yoshida, S.: Analysis of Double Block Length Hash Functions. In: Paterson, K.G. (ed.) Cryptography and Coding 2003. LNCS, vol. 2898, pp. 290–302. Springer, Heidelberg (2003)
14. Hirose, S.: Provably Secure Double-Block-Length Hash Functions in a Black-Box Model. In: Park, C., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 330–342. Springer, Heidelberg (2005)
15. Hirose, S.: Some Plausible Constructions of Double-Block-Length Hash Functions. In: Robshaw, M. (ed.) FSE 2006. LNCS, vol. 4047, pp. 210–225. Springer, Heidelberg (2006)
16. Hohl, W., Lai, X., Meier, T., Waldvogel, C.: Security of Iterated Hash Functions Based on Block Ciphers. In: Stinson [40], pp. 379–390
17. ISO/IEC. ISO DIS 10118-2: Information technology - Security techniques - Hash-functions, Part 2: Hash-functions using an n-bit block cipher algorithm. First released in 1992 (2000)
18. Kilian, J., Rogaway, P.: How to Protect DES against Exhaustive Key Search. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 252–267. Springer, Heidelberg (1996)

19. Knudsen, L.R., Lai, X., Preneel, B.: Attacks on Fast Double Block Length Hash Functions. J. Cryptology 11(1), 59–72 (1998)
20. Knudsen, L.R., Muller, F.: Some Attacks Against a Double Length Hash Proposal. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 462–473. Springer, Heidelberg (2005)
21. Lee, J., Kwon, D.: The security of abreast-dm in the ideal cipher model. Cryptology ePrint Archive, Report 2009/225 (2009), http://eprint.iacr.org/
22. Lee, J., Kwon, D.: The Security of Abreast-DM in the Ideal Cipher Model. IACR Cryptology ePrint Archive, 2009, 225 (2009)
23. Lee, J., Stam, M.: MJH: A Faster Alternative to MDC-2. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 213–236. Springer, Heidelberg (2011)
24. Lee, J., Stam, M., Steinberger, J.: The Collision Security of Tandem-DM in the Ideal Cipher Model. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 561–577. Springer, Heidelberg (2011)
25. Rabin, M.: Digitalized Signatures (1978)
26. Menezes, A., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1996)
27. Merkle, R.C.: One Way Hash Functions and DES. In: Brassard [6], pp. 428–446
28. Nandi, M., Lee, W.I., Sakurai, K., Lee, S.: Security Analysis of a 2/3-Rate Double Length Compression Function in the Black-Box Model. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 243–254. Springer, Heidelberg (2005)
29. NIST National Institute of Standards and Technology. FIPS 180-1: Secure Hash Standard (April 1995), http://csrc.nist.gov
30. NIST National Institute of Standards and Technology. FIPS 180-2: Secure Hash Standard (April 1995), http://csrc.nist.gov
31. Preneel, B., Govaerts, R., Vandewalle, J.: Hash Functions Based on Block Ciphers: A Synthetic Approach. In: Stinson [40], pp. 368–378
32. Rivest, R.L.: RFC 1321: The MD5 Message-Digest Algorithm. Internet Activities Board (April 1992)
33. Rivest, R.L.: The MD4 Message Digest Algorithm. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 303–311. Springer, Heidelberg (1991)
34. Rogaway, P., Shrimpton, T.: Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 371–388. Springer, Heidelberg (2004)
35. Rogaway, P., Steinberger, J.P.: Constructing Cryptographic Hash Functions from Fixed-Key Blockciphers. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 433–450. Springer, Heidelberg (2008)
36. Rogaway, P., Steinberger, J.P.: Security/Efficiency Tradeoffs for Permutation-Based Hashing. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 220–236. Springer, Heidelberg (2008)
37. Satoh, Haga, Kurosawa: Towards secure and fast hash functions. TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems (1999)
38. Stam, M.: Blockcipher-Based Hashing Revisited. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 67–83. Springer, Heidelberg (2009)
39. Steinberger, J.P.: The Collision Intractability of MDC-2 in the Ideal-Cipher Model. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 34–51. Springer, Heidelberg (2007)
40. Stinson, D.R. (ed.): CRYPTO 1993. LNCS, vol. 773. Springer, Heidelberg (1994)

41. Wang, X., Lai, X., Feng, D., Chen, H., Yu, X.: Cryptanalysis of the Hash Functions MD4 and RIPEMD. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 1–18. Springer, Heidelberg (2005)
42. Wang, X., Yin, Y.L., Yu, H.: Finding Collisions in the Full SHA-1. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 17–36. Springer, Heidelberg (2005)
43. Winternitz, R.S.: A Secure One-Way Hash Function Built from DES. In: IEEE Symposium on Security and Privacy, pp. 88–90 (1984)

A Related Work

Schemes with NonOoptimal or Unknown Collision Security. Preneel *et al.* [31] discussed the security of single (block)length hash functions against several generic attacks. They concluded that 12 out of 64 hash functions are secure against these attacks. However, formal proofs were first given by Black *et al.* [2] about 10 years later. Their most important result is that 20 hash functions – including the 12 mentioned above – are optimally collision resistant. Knudsen *et al.* [19] discussed the insecurity of DBL hash functions with rate 1 composed of (n, n) block ciphers. Hohl *et al.* [16] analyzed the security of DBL compression functions with rate 1 and 1/2. Satoh *et al.* [37] and Hattoris *et al.* [13] discussed DBL hash functions with rate 1 composed of $(2n, n)$ block ciphers. MDC-2 and MDC-4 [5,17] are (n, n) block cipher based DBL hash functions with rates 1/2 and 1/4, respectively. Steinberger [39] proved that for MDC-2 instantiated with, *e.g.*, AES-128 no adversary asking less than $2^{74.9}$ can usually find a collision. Nandi *et al.* [28] proposed a construction with rate 2/3 but it is not optimally collision resistant. In [20], Knudsen and Muller presented some attacks against it. At EUROCRYPT'08 and CRYPTO'08, Steinberger [35,36] proved some security bounds for fixed-key (n, n) block cipher based hash functions, *i.e.*, permutation based hash functions, that all have small rates and low security guarantees. None of these schemes/techniques mentioned so far are known to have birthday-type collision resistance. Lee and Stam [23] gave a scheme similar to MDC-2, called MJH. It uses finite field multiplications to offer a collision security bound in the iteration of $O(2^{2n/3-\log n})$.

Schemes with Birthday-Type Collision Security. Merkle [27] presented three DBL hash functions composed of DES with rates of at most 0.276. They are optimally collision resistant in the ideal cipher model. Hirose [14] presented a class of DBL hash functions with rate 1/2 which are composed of two different and independent $(2n, n)$ block ciphers that have birthday-type collision resistance. At FSE'06, Hirose [15] presented a rate 1/2 and $(2n, n)$ block cipher based DBL hash function that has birthday-type collision resistance. He stated that for his compression function, no adversary can find a collision with probability greater than 1/2 if no more than $2^{124.55}$ queries are asked (see [10, App. B] for details on this). For Tandem-DM, the best known collision security bound is $2^{120.87}$ queries [24]. Fleischmann *et al.* [11] as well as Lee and Kwon [21] independently provided a security bound for Abreast-DM of $2^{124.42}$. In [11] a lot of variants are also discussed, *e.g.*, Cyclic-DM, Cube-DM or Add/k-DM. Bos *et al.* [3]

provided practical performance figures for some double length hash functions using the AES-NI instruction set.

Preimage Security Results. For single length compression functions, tight security results are known [2,38]. For double length compression functions, some birthday-type preimage results are also known [22,24], essentially stating that any adversary asking less 2^n queries has only a negligible chance of finding a preimage. For ABREAST-DM, TANDEM-DM and HIROSE-DM there are better bounds known [1] (cf. also Table 1).