Counter-bDM: A Provably Secure Family of Multi-Block-Length Compression Functions

Farzaneh Abed, Christian Forler, Eik List, Stefan Lucks, and Jakob Wenzel

Bauhaus-Universität Weimar, Germany
{farzaneh.abed,christian.forler,eik.list,stefan.lucks,
jakob.wenzel}@uni-weimar.de

Abstract. Block-cipher-based compression functions serve an important purpose in cryptography since they allow to turn a given block cipher into a one-way hash function. While there are a number of secure double-block-length compression functions, there is little research on generalized constructions. This paper introduces the COUNTER-bDM family of multi-block-length compression functions, which, to the best of our knowledge, is the first provably secure block-cipher-based compression function with freely scalable output size. We present generic collision- and preimage-security proofs for it, and compare our results with those of existing double-block-length constructions. Our security bounds show that our construction is competitive with the best collision- and equal to the best preimage-security bound of existing double-block-length constructions.

Keywords: block cipher, compression function, hash function, provable security.

1 Introduction

While the SHA-3 competition has encouraged many new interesting ideas for designing hash and compression functions (e.g., the sponge framework [3]), one of the most popular approaches is to use a given block cipher and turn it into a one-way function. While the roots to this simple principle can be tracked back to Rabin [33] at the end of the 70s, the knowledge about it is still highly relevant today. For instance, the standardized SHA-1 and SHA-2 hash function families base on the SHACAL-1/2 ciphers. But also many submissions for the SHA-3 contest, such as – Blake [2], Skein [37], or SHAVITE-3 [4] – are built on block ciphers. The advantages are obvious: not only can compression-function designers profit from the pseudo-randomness of an IND-CCA-secure cipher, but also do they require only a single primitive to obtain both encryption and hashing – an important matter when designing hardware for resource-constrained devices.

The best understood principle for block-cipher-based compression functions are so-called *single-block-length* constructions, which compress a $2n$-bit input to an n-bit output, where n is the state size of the cipher. However, the state size of the AES is 128 bits, which yields a 64-bit collision security, which is

D. Pointcheval and D. Vergnaud (Eds.): AFRICACRYPT 2014, LNCS 8469, pp. 440–458, 2014.

insufficient for many applications. As a consequence, one is usually interested in double-block-length or, more generally, multi-block-length block-cipher-based hash functions, which take an (an)-bit input and produce a (bn)-bit output, for $a > b \geq 2$.

Related Work. The idea of double-block-length hashing can be attributed to Meyer and Schilling and their proposal of the rate-1/2 and rate-1/4 hash functions MDC-2 and MDC-4 [6] in 1988. Together with the Davies-Meyer-like schemes ABREAST-DM and TANDEM-DM from Lai and Massey [24], these four are commonly known as *classical* constructions. A number of further double-block-length functions have been proposed recently. According to Mennink [34], these can be ordered into the classes DBL^{2n} – which employ a cipher with a $2n$-bit key – and DBL^{n} – which use a cipher with an n-bit key (see [41] for example). The former class contains ABREAST-DM, the variants by Lee and Kwon [27], TANDEM-DM, HIROSE-DM [17], Stam's supercharged Type-I compression function [30,43,44], as well as the generalizations by Özen and Stam [38] and by Hirose [16].

Moreover, Fleischmann et al. generalized several classes of Davies-Meyer designs and proposed a class of cyclic constructions that contains the compression functions WEIMAR-DM, ADD-K-DM, and CUBE-DM [12,14]. A more detailed review of related work is provided in Appendix A. All of the mentioned provide a birthday- type collision security; in addition, there are security proofs for WEIMAR-DM, HIROSE-DM, TANDEM-DM, and ABREAST-DM are given in [12,17,26,27,29].

While double-block-length hashing can offer an acceptable collision security, a variety of applications demand secure multi-block-length functions with a freely scalable output of the compression function. For instance, public-key signature schemes expect inputs of the exact length of the signing key. Moreover, in the era of SHA-3, hash values with a length of ≥ 256 bits are standard. But it is still an open research question how to create provably secure b-block-length compression functions for $b > 2$.

Contribution. First, we define the class MBL^{bn} for multi-block-length compression functions that employ a (bn, n)-bit keyed block cipher $E : \{0,1\}^{bn} \times \{0,1\}^{n} \to \{0,1\}^{n}$, and produce a bn-bit chaining value. Then, we present a freely scalable multi-block-length compression function, called COUNTER-bDM, which, to the best of our knowledge, is the first provably secure multi-block-length compression function for $b > 2$. It is a generalization of the double-block-length compression function HIROSE-DM [18]. For the generic COUNTER-bDM, we present a detailed security analysis for proofs of collision and preimage security, which employs the idea of super queries by Armknecht et al. [1]. Similar approaches were presented by Mennink [34] and Lee [25].

For $b = 2$ our resulting collision-security bound shows that every adversary that wants to find a collision with advantage 1/2 requires $2^{125.18}$ queries, which is comparable to the currently best collision- security bound of WEIMAR-DM [12]. Concerning preimage security, we obtain a near-optimal bound of 2^{251} queries,

Table 1. Comparison of security results on double-block-length compression functions, evaluated for $n = 128$ bits and a success probability of 1/2. For Cyclic-DM, $k > 1$; for Add-k-DM $k' \geq 2$.

Compression function	Collision bound		Preimage bound	
Abreast-DM [24]	$2^{124.42}$	[14,26]	2^{246}	[1]
Add-k-DM [14]	$2^{127-k'}$	[14]	$\approx 2^{128}$	[14,26]
Counter-2DM [Sec. 3]	$\mathbf{2^{125.18}}$	[Sec. 5]	$\mathbf{2^{251}}$	[Sec. 6]
Cube-DM [14]	$2^{125.41}$	[14]	$\approx 2^{128}$	[14,26]
Cyclic-DM (cycle length > 2) [14]	2^{127-k}	[14]	$\approx 2^{128}$	[14,26]
Cyclic-DM (cycle length 2) [14]	$2^{124.55}$	[14]	$\approx 2^{128}$	[14,26]
Hirose-DM [17]	$2^{125.23}$	[13]	2^{251}	[1]
Lee/Kwon [27]	$2^{125.0}$	[26]	$\approx 2^{128}$	[14,26]
Tandem-DM [24]	$2^{120.87}$	[29]	2^{246}	[1]
Weimar-DM [12]	$2^{126.73}$	[9]	2^{251}	[12]

which is equivalent to the currently best bound of Weimar-DM. Table 1 compares our bounds with that of previously published double-block-length compression functions.

Outline. In what remains, Section 2 revisits the basic notions concerning block-cipher-based compression functions. Section 3 introduces Counter-bDM. Section 4 summarizes the formal security definitions that are essential for our analysis. In Section 5 we present the proof for the collision security of Counter-bDM. Section 6 then derives the preimage-security bound. Finally, Section 7 concludes the paper.

2 Basic Notions

This section recaps the relevant basic notions. We borrow the description of block-cipher-based compression functions from [12]:

Definition 1 (Block Cipher). *Let $k, n \geq 1$ be integers. We define a (k, n)-bit block cipher as a keyed family of permutations, which consists of an encryption function $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, and its inverse (decryption) function $D = E^{-1} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. Both take a k-bit key K and an n-bit input block X, and produce an n-bit output Y, where $D_K(E_K(X)) = X$, for all $X \in \{0,1\}^n, K \in \{0,1\}^k$. We denote by $\texttt{Block}(k, n)$ the set of all (k, n)-bit block ciphers.*

Definition 2 (Single-Block-Length Compression Function). *Let $n \geq 1$ be an integer. A single-block-length (SBL) block-cipher-based compression function is a function $H^{SBL} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ which uses a block cipher from $\texttt{Block}(n, n)$.*

The idea was discussed in the literature first by Rabin [33]. Most SBL functions use a block cipher from $\texttt{Block}(n, n)$ and compress a $2n$-bit string to an n-bit string. A popular example is the Davies-Meyer (DM) [46] mode:

$$H^{DM}(M, U) = E_M(U) \oplus U,$$

which is essentially used twice inside HIROSE-DM and b times, in slightly modified fashion, inside COUNTER-bDM.

Definition 3 (Multi-Block-Length Compression Function). *Let $b, n \geq 1$ be integers. A multi-block-length (MBL) block-cipher-based compression function is a function $H^{MBL} : \{0,1\}^{bn} \times \{0,1\}^n \to \{0,1\}^{bn}$, which takes an n-bit message and a bn-bit chaining value, and outputs a new bn-bit chaining value.*

Independent Ciphers. The sophisticated task of proving the security for a multi-block-length compression function simplifies greatly if one can ensure that the b outputs of the individual block-cipher calls in one invocation of the compression function are independent and distinct from each other. Previous double-block-length constructions achieve this requirement by either...

Distinct Permutations: ... using b independent permutations in the compression function. This approach is used, e.g., by the early construction of Hirose [16] or those by Rogaway and Steinberger [41].

Distinct Keys: ... guaranteeing that all key inputs K_i used for the block-cipher calls inside one compression-function call are different: $K_i \neq K_j,\ 1 \leq i < j \leq b$, which results in having de facto different permutations. This approach is used, e.g., by WEIMAR-DM [12].

Distinct Plaintexts: ... guaranteeing that all b plaintext inputs X_i used as inputs to the block cipher in one compression-function call are different: $X_i \neq X_j,\ 1 \leq i < j \leq b$. This approach is used, e.g., by CUBE-DM [14] or HIROSE-DM [18].

The first approach renders unpractical in practice since it requires multiple permutation implementations of the class MBL^{bn}. The further two approaches are similar. However, using a different key in every block-cipher call implies the potential need of running the key schedule of the underlying block cipher multiple times. Therefore, we employ the latter strategy function for COUNTER-bDM, i.e., we ensure that all plaintext inputs to the block-cipher calls are different.

3 Counter-bDM

This section defines the COUNTER-bDM family of multi-block-length compression functions. Note that we use H^{CbDM} as short notion of COUNTER-bDM.

Definition 4 (Counter-bDM). *Let E be a block cipher from $\texttt{Block}(bn,\ n)$. The compression function $H^{CbDM} : \{0,1\}^{bn} \times \{0,1\}^n \to \{0,1\}^{bn}$ is defined by*

$$H^{CbDM}(M, U_1, \ldots, U_b) = (V_1, \ldots, V_b),$$

where the outputs V_i *are given by* $V_i = E_K(U_1 \oplus (i-1)) \oplus U_1$*, with* $K = U_2 \,||\, \ldots \,||\, U_b \,||\, M$.

Two concrete examples of our multi-block-length compression-function family, Counter-3DM (left) and Counter-4DM (right), are illustrated in Figure 1. However, in our security analysis in Sections 5 and 6 we consider the generic version Counter-bDM.

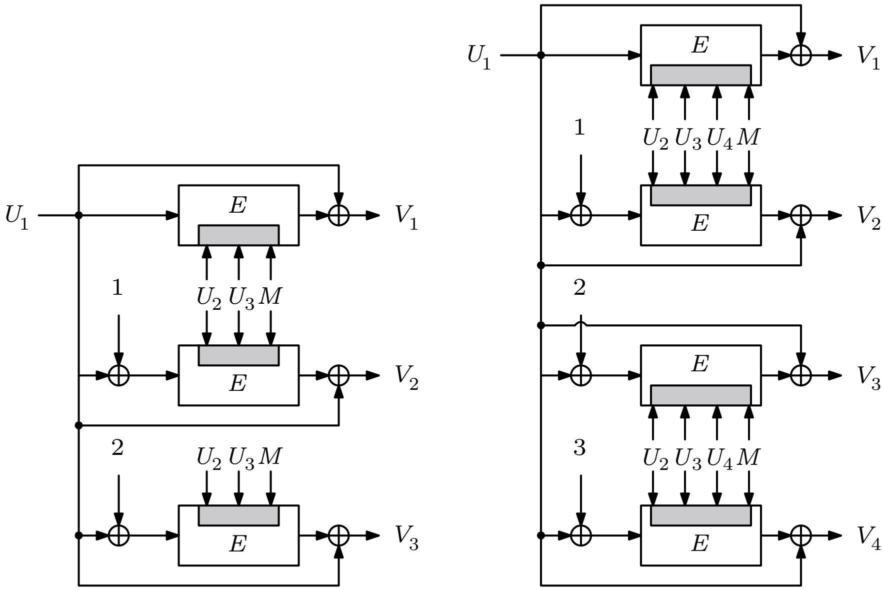

Fig. 1. Two examplary compression functions H^{C3DM} (left) and H^{C4DM} (right) from the family of compression functions H^{CbDM}

It is easy to see that, due to the XOR with the counter $i-1$, all plaintext inputs X_i to the block-cipher calls are pair-wise distinct. Additionally, since all values $i-1$ are in the range of $[0, \ldots, b-1]$, the counter values affect only the least significant $\lceil log_2(b) \rceil$ bits of the plaintexts. We call the most significant $n - \lceil log_2(b) \rceil$ bits of the plaintexts a *common prefix*.

Definition 5 (Common-Prefix Property). *Let* $X = X_{pre} \,||\, X_{post}$*,* $X \in \{0,1\}^n$ *be an n-bit integer, where* X_{pre} *denotes the* $n - \lceil log_2(b) \rceil$ *most significant bits, and* X_{post} *the* $\lceil log_2(b) \rceil$ *least significant bits of* X*. Further, let* $X_i = X \oplus (i-1)$ *(with* $1 \leq i \leq b$*) denote the values which are used as plaintext inputs to the block-cipher calls in one invocation of* H^{CbDM}*. Then, all values* X_i *share the same common prefix* $X_{pre} \in \{0,1\}^{n - \lceil log_2(b) \rceil}$.

Remark 1. For the remainder of this paper, we denote by $c = 2^{\lceil log_2(b) \rceil} \geq b$ the maximal number of plaintexts $X = X_{pre} \,||\, X_{post}$ which can share the same prefix X_{pre}.

We will see later that both the pair-wise distinct plaintexts and the common-prefix property will be beneficial for an easy-to-grasp security analysis of COUNTER-bDM.

4 Proof Preliminaries

This section formally describes the notions and properties that are relevant for our security analysis of COUNTER-bDM.

4.1 Proof Model

The security of a block-cipher-based compression function should depend only on the security of the construction, and not on that of the (potentially insecure) chosen block cipher inside. Thus, one usually considers the *ideal-cipher model* , wherein a block cipher is modeled as a family of random n-bit random permutations $\{E_K\}$. The permutation E that is used in the compression function is chosen at random from $\texttt{Block}(k, n)$: $E \xleftarrow{\$} \texttt{Block}(k, n)$. Thus, we follow the notions by Black et al. [5].

An adversary $\mathcal{A}$ is defined as a probabilistic, computationally unbounded algorithm that is limited only by a number of q queries it can ask to an oracle E. For any of its queries, the adversary is allowed to ask either a forward (encryption) query $E_K(X) = Y$, or a backward (decryption) query $X = D_K(Y)$, where $X, Y \in \{0,1\}^n$ and $\forall X : D_K(E_K(X)) = X$. Each query Q^i is stored as a 3-tuple (X_i, Y_i, K_i) in a query history $\mathcal{Q}$, where we denote by $\mathcal{Q}_i$ the state of the query history after i queries have been asked by the adversary, for $1 \leq i \leq q$. We further borrow two usual assumptions about $\mathcal{A}$ from [12]:

1. If $\mathcal{A}$ has successfully found a collision or a preimage for H^{CbDM}, it has obtained the necessary encryption or decryption results only by making queries to the oracle E.
2. $\mathcal{A}$ does not ask queries to which it already knows the answer, e.g., if $\mathcal{A}$ already knows the answer to a forward query $Y = E_K(X)$, it will not ask $D_K(Y)$ – which must return X – and vice versa.

4.2 Collision-Security

We define the collision security of our compression function H^{CbDM} by the advantage of an adversary $\mathcal{A}$ to win Experiment 1.

Experiment 1 (Collision-Finding Experiment $\texttt{Exp-Coll}_{\mathcal{A},\, H^{CbDM}}(bn)$)

1. *An adversary $\mathcal{A}$ is given oracle access to a block cipher $E \in \texttt{Block}(bn, n)$.*
2. *After asking at most q queries (X_i, Y_i, K_i) for $1 \leq i \leq q$, it outputs a pair $(M, U_1, \ldots, U_b), (M', U'_1, \ldots, U'_b) \in \{0,1\}^{(b+1)n} \times \{0,1\}^{(b+1)n}$.*

3. *The adversary wins the experiment iff its output is a valid collision for H^{CbDM}, i.e.,*

$$H^{CbDM}(M, U_1, \ldots, U_b) = H^{CbDM}(M', U_1', \ldots, U_b') \text{ and}$$
$$(M, U_1, \ldots, U_b) \neq (M', U_1', \ldots, U_b').$$

Otherwise, $\mathcal{A}$ loses the experiment.

The advantage of an adversary $\mathcal{A}$ to find such a collision for H^{CbDM} is given by the probability that $\mathcal{A}$ can win Experiment 1, or formally written, by

$$\mathbf{Adv}_{H^{CbDM}}^{COLL}(\mathcal{A}) = \Pr\left[\texttt{Exp-Coll}_{\mathcal{A},\, H^{CbDM}}(bn) = 1\right]$$

Since we only limit the adversary by the number of queries, it is allows to ask to E, we write

$$\mathbf{Adv}_{H^{CbDM}}^{COLL}(q) := \max_{\mathcal{A}} \left\{ \mathbf{Adv}_{H^{CbDM}}^{COLL}(\mathcal{A}) \right\},$$

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

4.3 Preimage Security

There are various notions considering preimage security (see [40] for example). We adapt that of *everywhere preimage security* (EPRE), which was introduced by Rogaway and Shrimpton in [40]. There, the adversary commits to a hash value before it makes any queries to the oracle. The preimage security of our compression function H^{CbDM} is therefore defined by the advantage that an adversary $\mathcal{A}$ wins Experiment 2.

Experiment 2 (Preimage-Finding Experiment $\texttt{Exp-ePre}_{\mathcal{A},\, H^{CbDM}}(bn)$)

1. *An adversary $\mathcal{A}$ is given oracle access to a block cipher $E \in \texttt{Block}(bn, n)$. Before it makes any queries, it announces a hash value $(V_1, \ldots, V_b) \in \{0,1\}^{bn}$.*
2. *After asking at most q queries (X_i, Y_i, K_i) for $1 \leq i \leq q$, it outputs a $(b+1)$-tuple $(M, U_1, \ldots, U_b) \in \{0,1\}^{(b+1)n}$.*
3. *The adversary wins the experiment iff its output is a valid preimage for $(V_1, \ldots, V_b)$ and H^{CbDM}, i.e.,*

$$H^{CbDM}(M, U_1, \ldots, U_b) \;=\; (V_1, \ldots, V_b).$$

Otherwise, $\mathcal{A}$ loses the experiment.

We let $\mathbf{Adv}_{H^{CbDM}}^{EPRE}(\mathcal{A})$ be true iff $\texttt{Exp-ePre}_{\mathcal{A},\, H^{CbDM}}(bn)$ returns 1. The precommitted hash value $(V_1, \ldots, V_b)$ is an omitted parameter of $\mathbf{Adv}_{H^{CbDM}}^{EPRE}(\mathcal{A})$. We define

$$\mathbf{Adv}_{H^{CbDM}}^{EPRE}(q) := \max_{\mathcal{A}} \left\{ \mathbf{Adv}_{H^{CbDM}}^{EPRE}(\mathcal{A}) \right\},$$

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

5 Collision-Security Analysis of Counter-bDM

Let $\mathcal{A}$ be a collision-finding adversary for H^{CbDM} that can ask queries to an oracle E. In between $\mathcal{A}$ and E, we construct another adversary $\mathcal{A}'$ which simulates $\mathcal{A}$, but sometimes is allowed to make additional queries to E that are not taken into account. Since $\mathcal{A}'$ is more powerful than $\mathcal{A}$, it is easy to see that it suffices for us to upper bound the success probability of $\mathcal{A}'$. Thereby, we say that an adversary $\mathcal{A}$ (or $\mathcal{A}'$, respectively) is *successful* if its query history contains the means of computing a collision for H^{CbDM}.

Attack Setting. During the attack, $\mathcal{A}$ maintains a query history $\mathcal{Q}$ wherein it stores all queries it poses to E. An entry in the query history of $\mathcal{A}$ is a tuple (K, X, Y), where $Y = E_K(X)$. Simultaneously, $\mathcal{A}'$ maintains a query list $\mathcal{L}$ which contains all input/output pairs to the compression function H^{CbDM} that can be computed by $\mathcal{A}$. An entry $L \in \mathcal{L}$ is a tuple $(K, X, Y_1, \ldots, Y_c) \in \{0,1\}^{(b+1+c)n}$, where $K \in \{0,1\}^{bn}$, $X \in \{0,1\}^n$ is the input to the compression function H^{CbDM}, and $c = 2^{\lceil log_2(b) \rceil}$ (see Remark 1). The values $Y_i \in \{0,1\}^n$ are given as the results of the forward queries $Y_i = E_K(X \oplus (i-1))$, for $1 \le i \le c$. Moreover, we define $\mathcal{L}_j$ to denote the state of $\mathcal{L}$, which contains the first j queries of $\mathcal{A}'$, with $j \geq 1$.

Collision Events. When E is modeled as an ideal cipher, we run into problems when $\mathcal{A}$ asks close to or even more than $q = 2^n$ queries. In the case when $\mathcal{A}$ asks q queries under the same key to E and q reaches $2^n - 1$, E loses its randomness. As a remedy to this problem, Armknecht et al. proposed the idea of *super queries* [1]; given some key K, $\mathcal{A}'$ can pose regular queries to E or D until $N/2$ queries with the same key K have been added to its query list $\mathcal{L}$, where $N = 2^n$.

If $\mathcal{L}$ contains $N/2$ queries for a key K and $\mathcal{A}$ requests another query for the key K from $\mathcal{A}'$, then, $\mathcal{A}'$ poses all remaining queries $(K, *, *)$ under this key to E at once. In this case, we say that a *super query* occurred. All queries that are part of a super query *are not taken into account*, i.e., they do not add to q, the number of queries $\mathcal{A}$ is allowed to ask. Since these free queries are asked at once, one no longer has to consider the success probability of a single query; instead, one can consider the event that $\mathcal{A}'$ is successful with any of the contained queries. Thus, E does not lose its randomness. In the following, we define three mutually exclusive events which cover all case when $\mathcal{A}'$ can be successful.

NormalQueryWin($\mathcal{L}$). This describes the case when $\mathcal{A}'$ finds a collision with its current query L^j and a query $L^r \in \mathcal{L}_{j-1}$, where L^j was a normal query.

SuperQueryWin($\mathcal{L}$). This describes the case when $\mathcal{A}'$ finds a collision with its current query L^j and a query $L^r \in \mathcal{L}_{j-1}$, where L^j was part of a super query.

SameQueryWin($\mathcal{L}$). This describes the case when $\mathcal{A}'$ finds a collision within the same entry $L^j \in \mathcal{L}$.

Since the adversary can only win if it finds a collision using either one of the mentioned events, it is sufficient for us to upper bound the sum of the probabilities. Thus, it holds that

$$\mathbf{Adv}^{COLL}_{H^{CbDM}}(q) \leq \Pr[\mathsf{NormalQueryWin}(\mathcal{L})] + \Pr[\mathsf{SuperQueryWin}(\mathcal{L})] \qquad (1)$$
$$+ \Pr[\mathsf{SameQueryWin}(\mathcal{L})].$$

Remark 2. Note that a tuple $L \in \mathcal{L}$ consists of $c = 2^{\lceil log_2(b) \rceil}$ query results. Since c always divides $N/2$, i.e., $c \mid N/2$, each tuple L is either part of a normal query or a super query, but never both.

Before we present our bound, we describe more precisely what we mean by $\mathcal{A}'$ has found a collision for H^{CbDM}. Let $L^r = (K^r, X^r, Y_1^r, \ldots, Y_c^r)$ represent the r-th entry in $\mathcal{L}$, and $L^j = (K^j, X^j, Y_1^j, \ldots, Y_c^j)$ the j-th entry in $\mathcal{L}$, where $1 \leq r < j \leq q$. We say that L^r and L^j provide the means for computing a collision if $\exists\, \ell, m \in \{0, \ldots, c-1\}$ so that b equations of the following form hold:

$$\begin{aligned}
E_{K^r}(X^r \oplus \ell \oplus 0) \oplus X^r &= E_{K^j}(X^j \oplus m \oplus 0) \oplus X^j,\\
E_{K^r}(X^r \oplus \ell \oplus 1) \oplus X^r &= E_{K^j}(X^j \oplus m \oplus 1) \oplus X^j,\\
&\vdots\\
E_{K^r}(X^r \oplus \ell \oplus (b-1)) \oplus X^r &= E_{K^j}(X^j \oplus m \oplus (b-1)) \oplus X^j.
\end{aligned}$$

Theorem 3. *Let $N = 2^n$. Then, it applies that*

$$\boldsymbol{Adv}^{COLL}_{H^{CbDM}}(q) \leq \frac{c^2 \cdot 2^b \cdot q^2}{N^b} + \frac{c^3 \cdot 2^{b+2} \cdot q^2}{N^{b+1}}.$$

Proof. After $\mathcal{A}$ has asked a (normal) forward query $Y^j = E_{K^j}(X^j)$ or a (normal) backward query $X^j = D_{K^j}(Y^j)$, $\mathcal{A}'$ checks if $\mathcal{L}_{j-1}$ already contains an entry $L^r = (K^j, X^j_{pre} \mid\mid *, *, \ldots, *)$, where X^j_{pre} denotes the prefix of X^j (see Definition 5) and $*$ denotes arbitrary values. In the following, we analyze two possible cases.

***Case 1:* L^r is not in $\mathcal{L}_{j-1}$.** In this case, $\mathcal{A}'$ labels Y^j as Y_1^j and asks $(c-1)$ further queries to E that are not taken into account:

$$\forall i \in \{2, \ldots, c\}: \quad Y_i^j = E_{K^j}(X^j \oplus (i-1)).$$

$\mathcal{A}'$ creates the tuple $L^j = (K^j, X^j, Y^j, \ldots, Y_c^j)$ and appends it to its query list, i.e., $\mathcal{L}_j = \mathcal{L}_{j-1} \cup \{L^j\}$. Now, we have to upper bound the success probability of $\mathcal{A}'$ to find a collision for H^{CbDM}, i.e., the success probabilities for the events mentioned above.

***Subcase 1.1:* NormalQueryWin($\mathcal{L}$).** In this case, the adversary finds a collision using a normal query L^j and a query L^r that was already contained in $\mathcal{L}$. While super queries may have occurred for different keys before, the query history of $\mathcal{A}'$ may contain at most $N/2 - c$ plaintext-ciphertext pairs for the current key K^j. So, our random permutation E samples the query responses $Y_1^j, \ldots, Y_c^j$ for the current query at random from a set of size of at least $N/2 + c \geq N/2$ elements.

Hence, the probability that one equation from above holds for some fixed ℓ and m can be upper bounded by $1/(N/2)$; and the probability for b equations to hold is then given by

$$\frac{1}{(N/2)^b} = \frac{2^b}{N^b}.$$

There are c^2 possible combinations for ℓ and m, s.t. b values V_i^j can form a valid collision with b values V_i^r, with $i \in \{0, \ldots, b-1\}$. Thus, $\mathcal{A}'$ has a success probability for finding a collision for H^{CbDM} for two fixed queries L^j and L^r is at most

$$\frac{c^2}{(N/2)^b} = \frac{c^2 \cdot 2^b}{N^b}.$$

Since the j-th query can form a collision with any of the previous entries in $\mathcal{L}_{j-1}$, we have to determine the maximum number of queries in $\mathcal{L}_{j-1}$. If $\mathcal{A}'$ obtained a super query for each key it queried before, $\mathcal{L}_{j-1}$ may contain up to $2(j-1)$ entries. Since the winning query has to be a normal query in this case, $\mathcal{L}$ can contain at most q normal queries and up to $(q-1)$ queries (without the current one) resulting from super queries in the history. This would imply that one had to sum up the probabilities up to $2q-1$:

$$\sum_{j=1}^{2q-1} \frac{2(j-1) \cdot c^2 \cdot 2^b}{N^b}.$$

However, we can do better. In the $\mathsf{NormalQueryWin}(\mathcal{L})$ case, $\mathcal{A}'$ will not win if its last (winning) query was part of a super query. Hence, we do not need to test if any of the super queries will produce a collision with any of their respective previous queries, and we have to test only possible collisions with the (at most q) normal queries. Nevertheless, $\mathcal{A}'$ still has to test each of the q normal queries if they collide with any of the at most $2q$ previous queries (including those which were part of a super query). Therefore, the success probability of $\mathcal{A}'$ to find a collision for H^{CbDM} can be upper bounded by

$$\Pr[\mathsf{NormalQueryWin}(\mathcal{L})] \leq \sum_{j=1}^{q} \frac{2(j-1) \cdot c^2 \cdot 2^b}{N^b} \leq \frac{q^2 \cdot c^2 \cdot 2^b}{N^b}. \qquad (2)$$

Subcase 1.2: **$\mathsf{SuperQueryWin}(\mathcal{L})$.** In this case, $\mathcal{A}'$ wins with a super query, i.e., it has asked the $(N/2+1)$-th query for K^j, triggering a super query to occur. We can reuse the argument from Subcase 1.1 that the success probability of $\mathcal{A}'$ to obtain b colliding equations for two fixed queries L^r, L^j can be upper bounded by

$$\frac{c^2}{(N/2)^b}.$$

Here, the query history $\mathcal{L}_q$ contains at most $2q$ queries. But this time, we do not have to test if any of the q normal queries produces a collision with any of their respective predecessors. Hence, we can upper bound the success probability of $\mathcal{A}'$ to find a collision for H^{CbDM} with one super query by

$$\frac{2q \cdot c^2 \cdot 2^b}{N^b}.$$

For a super query to occur, $\mathcal{A}$ has to ask at least $N/(2c)$ regular queries. Thus, there can be at most $q/(N/2c)$ super queries in $\mathcal{L}$ and we obtain

$$\Pr[\mathsf{SuperQueryWin}(\mathcal{L})] \;\leq\; \frac{2q \cdot c^2 \cdot 2^b}{N^b} \cdot \frac{q}{N/2c} \;=\; \frac{c^3 \cdot 2^{b+2} \cdot q^2}{N^{b+1}}. \tag{3}$$

***Subcase 1.3:* SameQueryWin($\mathcal{L}$).** In this case, $\mathcal{A}'$ wins if it finds two integers $\ell, m \in \{0, \ldots, c-1\}$ with $\ell \neq m$ s.t.:

$$\begin{aligned}
E_{K^j}(X^j \oplus \ell \oplus 0) \oplus X^j &= E_{K^j}(X^j \oplus m \oplus 0) \oplus X^j, \\
E_{K^j}(X^j \oplus \ell \oplus 1) \oplus X^j &= E_{K^j}(X^j \oplus m \oplus 1) \oplus X^j, \\
&\vdots \\
E_{K^j}(X^r \oplus \ell \oplus (b-1)) \oplus X^j &= E_{K^j}(X^j \oplus m \oplus (b-1)) \oplus X^j.
\end{aligned}$$

However, due to the XOR with the distinct values $i-1$, all plaintext inputs $X^j \oplus (i-1)$ in one compression-function call differ from each other. Furthermore, since all plaintext inputs are encrypted under the same key K^j and E is an ideal block cipher, their corresponding outputs Y_i^j are all different and uniformly distributed, and so are the values $Y_i^j \oplus X^j$ after the feed-forward operation. Hence, it is not possible for $\mathcal{A}'$ to find a collision for H^{CbDM} among the values $Y_i^j \oplus X^j$:

$$\Pr[\mathsf{SameQueryWin}(\mathcal{L})] \;=\; 0. \tag{4}$$

***Case 2:* L^r is in $\mathcal{L}_{j-1}$.** In this case, the key K^j and the plaintext prefix X_{pre}^j of $\mathcal{A}$'s current query $(K^j, X_{pre}^j \mid\mid X_{post'}^j)$ are already stored in some entry $L^r \in \mathcal{L}_{j-1}$, where $L^r = (K^r, X_{pre}^r \mid\mid X_{post}^r, Y_1^r, \ldots, Y_c^r)$. $\mathcal{A}'$ just extracts $Y^r_{(X_{post}^r \oplus X_{post'}^j)+1}$ from L^r, and passes it to $\mathcal{A}$. This implies that $\mathcal{A}$ can learn only information which $\mathcal{A}'$ already possesses. Thus,

$$\mathbf{Adv}_{H^{CbDM}}^{COLL}(\mathcal{A}) \leq \mathbf{Adv}_{H^{CbDM}}^{COLL}(\mathcal{A}').$$

Our claim is given by summing up equations (2), (3), and (4). □

Table 2 shows the minimal number of queries q an adversary has to ask in order to obtain an advantage of $\mathbf{Adv}_{H^{CbDM}}^{COLL}(q) = 1/2$ for the most practical block lengths $n \in \{64, 128\}$ and depending on b.

Table 2. Minimum number of block-cipher queries q that an adversary must ask in order to find a collision for H^{CbDM} with advantage 1/2

n = 64			**n = 128**		
#blocks b	**#queries** q	**optimal bound** $2^{bn/2}$	**#blocks** b	**#queries** q	**optimal bound** $2^{bn/2}$
2	$2^{61.50}$	2^{64}	2	$2^{125.50}$	2^{128}
4	$2^{123.50}$	2^{128}	4	$2^{251.50}$	2^{256}
8	$2^{248.50}$	2^{256}	8	$2^{504.50}$	2^{512}

6 Preimage-Security Analysis of Counter-bDM

Attack Setting. Let $(V_1, \ldots, V_b) \in \{0,1\}^{bn}$ be the point to invert (see Definition 4), chosen by an adversary $\mathcal{A}$ before it makes any query to E. We define that $\mathcal{A}$ has the goal to find a preimage for $(V_1, \ldots, V_b)$ as described in Experiment 2. For our preimage-security analysis, we adapt the procedure from our collision analysis, i.e., we construct another adversary $\mathcal{A}'$, which simulates $\mathcal{A}$, but sometimes is allowed to make additional queries to E that are not taken into account. Again, since $\mathcal{A}'$ is more powerful than $\mathcal{A}$, it suffices to upper bound the success probability of $\mathcal{A}'$. Here, we say that $\mathcal{A}'$ is *successful* if its query history $\mathcal{Q}$ contains the means of computing a preimage for $(V_1, \ldots, V_b)$.

The procedures of $\mathcal{A}$ and $\mathcal{A}'$ asking queries to the oracle E and building the query histories $\mathcal{Q}$ and $\mathcal{L}$ are the same as that described in our collision-security proof. Furthermore, we adopt the events $\mathsf{NormalQueryWin}(\mathcal{L})$ and $\mathsf{SuperQueryWin}(\mathcal{L})$ from there, which in this context, cover all possible winning events for $\mathcal{A}'$. Thus, it holds that

$$\mathbf{Adv}^{EPRE}_{H^{CbDM}}(q) \leq \Pr[\mathsf{NormalQueryWin}(\mathcal{L})] + \Pr[\mathsf{SuperQueryWin}(\mathcal{L})]. \quad (5)$$

Before we present our bound, we describe more precisely what is meant by $\mathcal{A}'$ has found a preimage for H^{CbDM}. Let $L^j = (K^j, X^j, Y^j_1, \ldots, Y^j_c)$ represent the j-th entry in $\mathcal{L}$. We say that L^j contains the means of computing a preimage if $\exists \ell \in \{0, \ldots, c-1\}$, so that the following b equations hold:

$$\begin{aligned} E_{K^j}(X^j \oplus \ell) \oplus X^j &= V_1 \\ E_{K^j}(X^j \oplus \ell \oplus 1) \oplus X^j &= V_2 \\ &\vdots \\ E_{K^j}(X^j \oplus \ell \oplus (b-1)) \oplus X^j &= V_b. \end{aligned}$$

Theorem 4. *Let $N = 2^n$. Then, it applies that*

$$\boldsymbol{Adv}^{EPRE}_{H^{CbDM}}(q) \leq \frac{c \cdot 2^{b+1} \cdot q}{N^b}.$$

Proof. After $\mathcal{A}$ has asked a (normal) forward query $Y^j = E_{K^j}(X^j)$ or a (normal) backward query $X^j = D_{K^j}(Y^j)$, $\mathcal{A}'$ checks if $\mathcal{L}_{j-1}$ already contains an entry $L^r = (K^j, X^j_{pre} \,||\, *, *, \ldots, *)$, where X^j_{pre} denotes the prefix of X^j. In the following, we analyze the possible cases and upper bound their success probabilities separately.

***Case 1:* L^r is not in $\mathcal{L}_{j-1}$.** In this case, $\mathcal{A}'$ labels Y as Y_1^j and asks $c-1$ further queries to E that are not taken into account:

$$\forall i \in \{2, \ldots, c\}: \quad Y_i^j \;=\; E_{K^j}(X^j \oplus (i-1)).$$

Then, $\mathcal{A}'$ creates the tuple $L^j = (K^j, X^j, Y_1^j, \ldots, Y_c^j)$ and appends it to its query list, i.e., $\mathcal{L}_j = \mathcal{L}_{j-1} \cup \{L^j\}$. Note that due to the XOR with $i-1$, all plaintexts X_i^j, with $i \leq i \leq c$, are pair-wise distinct. Thus, all ciphertexts Y_i^j, and the results of all feed-forward operations $(Y_i^j \oplus X^j)$ are always uniformly distributed.

In the following, we have to upper bound the success probability of $\mathcal{A}'$ to find a preimage for H^{CbDM} using either a normal query or a super query.

***Subcase 1.1:* NormalQueryWin($\mathcal{L}$).** Since we assume that the winning query is a normal one, $\mathcal{A}'$ can have collected at most $N/2 - c$ queries for the current key K^j. Thus, E samples the query responses $Y_1^j, \ldots, Y_c^j$ at random from a set of size of at least $N/2 + c \geq N/2$ elements. From the c values Y_i of L^j, the probability that one equation $E_{K^j}(X^j \oplus \ell) \oplus (X^j \oplus \ell) \;=\; V_i$ from above holds for some fixed value of ℓ, can be upper bounded by $1/(N/2)$. The probability that b equations from above hold for a fixed ℓ can be upper bounded by $1/(N/2)^b$. Since there are c possible values for ℓ, the probability to obtain a preimage with the j-th query is given by

$$\frac{c}{(N/2)^b} \;=\; \frac{c \cdot 2^b}{N^b}.$$

Since $\mathcal{A}'$ is allowed to ask at most q queries, it applies that

$$\Pr[\mathsf{NormalQueryWin}(\mathcal{L})] \;\leq\; \frac{c \cdot 2^b \cdot q}{N^b}. \tag{6}$$

***Subcase 1.2:* SuperQueryWin($\mathcal{L}$).** In this case, $\mathcal{A}'$ has already posed and stored $N/2c$ queries for the key K^j of its winning query. From the super query, it obtains the remaining $N/2c$ queries for K^j. We denote the latter set of queries by $\mathcal{SQ}$. From above, we already know that the probability that one point $L^j \in \mathcal{SQ}$ satisfies the preimage property can be upper bounded by

$$\frac{c}{(N/2)^b} \;=\; \frac{c \cdot 2^b}{N^b}.$$

Since the adversary obtains $N/2c$ points from the super query, the success probability that one of them yields a preimage for the given point is given by

$$\frac{N}{2c} \cdot \frac{c \cdot 2^b}{N^b} \;=\; \frac{2^{b-1}}{N^{b-1}}.$$

Table 3. Minimum number of block-cipher queries q that an adversary must ask in order to find a preimage for H^{CbDM} with advantage 1/2

n = 64			**n = 128**		
#blocks b	**#queries** q	**optimal bound** 2^{bn}	**#blocks** b	**#queries** q	**optimal bound** 2^{bn}
2	2^{123}	2^{128}	2	2^{251}	2^{256}
4	2^{248}	2^{256}	4	2^{504}	2^{512}
8	2^{499}	2^{512}	8	2^{1011}	2^{1024}

For every super query to occur, $\mathcal{A}'$ has to collect $N/2c$ queries in advance. Thus, there are at most $q/(N/2c)$ super queries and we obtain

$$\Pr[\mathsf{SuperQueryWin}(\mathcal{L})] \leq \frac{q}{N/2c} \cdot \frac{2^{b-1}}{N^{b-1}} = \frac{c \cdot 2^b \cdot q}{N^b}. \tag{7}$$

Case 2: **L^r is in $\mathcal{L}_{j-1}$.** Like in the Case 2 of our collision-security proof, the key K^j and the plaintext prefix X^j_{pre} of $\mathcal{A}$'s current query $(K^j, X^j_{pre} \mid\mid X^j_{post'})$ are already stored in some entry $L^r \in \mathcal{L}_{j-1}$, where $L^r = (K^j, X^j_{pre} \mid\mid X^j_{post}, Y^r_1, \ldots, Y^r_c)$. Again, $\mathcal{A}'$ extracts $Y^r_{(X^r_{post} \oplus X^j_{post'})+1}$ from L^r, and passes it to $\mathcal{A}$. This implies that $\mathcal{A}$ can learn only information that $\mathcal{A}'$ already possesses and

$$\mathbf{Adv}^{COLL}_{H^{CbDM}}(\mathcal{A}) \leq \mathbf{Adv}^{COLL}_{H^{CbDM}}(\mathcal{A}').$$

Our claim is given by summing up equations (6) and (7). □

For $n = 128$ and $\mathbf{Adv}^{EPRE}_{H^{CbDM}}(q) = 1/2$, we list in Table 3 the amounts of queries q an adversary has to make, depending on the value of b.

7 Conclusion and Outlook

This paper introduced COUNTER-bDM – the first provably secure family of multi-block-length compression functions, that maps $(b+1)n$-bit inputs to bn-bit outputs for arbitrary $b \geq 2$. With COUNTER-bDM, we propose a simple, though, very neat design, that not only avoids costly requirements such as the need of having independent ciphers, or having to run the key schedule multiple times, but also simplifies the analysis greatly. In our collision- and preimage-security analysis we provided proofs for arbitrary block lengths $b > 2$. It remains an open research topic to find a multi-block-length hash function with arbitrary output size employing an n-bit or at most $2n$-bit keyed block cipher.

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. Aumasson, J.-P., Henzen, L., Meier, W., Phan, R.C.-W.: SHA-3 proposal BLAKE. Submission to NIST, Round 3 (2010)
3. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. Ecrypt Hash Workshop (May 2007)
4. Biham, E., Dunkelman, O.: The SHAvite-3 Hash Function. Submission to NIST, Round 2 (2009)
5. 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)
6. Meyer, C., Matyas, S.: Secure Program Load With Manipulation Detection Code (1988)
7. Chang, D., Nandi, M., Lee, J., Sung, J., Hong, S., Lim, J., Park, H., Chun, K.: Compression Function Design Principles Supporting Variable Output Lengths from a Single Small Function. IEICE Transactions 91-A(9), 2607–2614 (2008)
8. Coppersmith, D., Pilpel, S., Meyer, C.H., Matyas, S.M., Hyden, M.M., Oseas, J., Brachtl, B., Schilling, M.: Data Authentication Using Modification Dectection Codes Based on a Public One-Way Encryption Function. U.S. Patent No. 4,908,861 (March 13, 1990)
9. Ewan Fleischmann. Analysis and Design of Blockcipher Based Cryptographic Algorithms. PhD thesis, Bauhaus-Universität Weimar (2013)
10. Fleischmann, E., Forler, C., Gorski, M., Lucks, S.: Collision-Resistant Double-Length Hashing. In: Heng, S.-H., Kurosawa, K. (eds.) ProvSec 2010. LNCS, vol. 6402, pp. 102–118. Springer, Heidelberg (2010)
11. Fleischmann, E., Forler, C., Lucks, S.: The Collision Security of MDC-4. In: Mitrokotsa, A., Vaudenay, S. (eds.) AFRICACRYPT 2012. LNCS, vol. 7374, pp. 252–269. Springer, Heidelberg (2012)
12. Fleischmann, E., Forler, C., Lucks, S., Wenzel, J.: Weimar-DM: A Highly Secure Double-Length Compression Function. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 152–165. Springer, Heidelberg (2012)
13. 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)
14. 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)
15. 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)
16. 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)
17. 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)

18. 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)
19. Hohl, W., Lai, X., Meier, T., Waldvogel, C.: Security of Iterated Hash Functions Based on Block Ciphers. In: Stinson, D.R. (ed.) Advances in Cryptology - CRYPTO 1993. LNCS, vol. 773, pp. 379–390. Springer, Heidelberg (1994)
20. 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)
21. Knudsen, L.R., Lai, X., Preneel, B.: Attacks on Fast Double Block Length Hash Functions. J. Cryptology 11(1), 59–72 (1998)
22. 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)
23. Krause, M., Armknecht, F., Fleischmann, E.: Preimage Resistance Beyond the Birthday Bound: Double-Length Hashing Revisited. IACR Cryptology ePrint Archive 2010, 519 (2010)
24. Lai, X., Massey, J.L.: Hash Functions Based on Block Ciphers. In: Rueppel, R.A. (ed.) Advances in Cryptology - EUROCRYPT1992. LNCS, vol. 658, pp. 55–70. Springer, Heidelberg (1993)
25. Lee, J.: Provable Security of the Knudsen-Preneel Compression Functions. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 504–525. Springer, Heidelberg (2012)
26. 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/`
27. Lee, J., Kwon, D.: The Security of Abreast-DM in the Ideal Cipher Model. IEICE Transactions 94-A(1), 104–109 (2011)
28. 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)
29. 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)
30. Lee, J., Steinberger, J.P.: Multiproperty-Preserving Domain Extension Using Polynomial-Based Modes of Operation. IEEE Transactions on Information Theory 58(9), 6165–6182 (2012)
31. Lucks, S.: A Collision-Resistant Rate-1 Double-Block-Length Hash Function. In: Symmetric Cryptography (2007)
32. Luo, Y., Lai, X.: Attacks On a Double Length Blockcipher-based Hash Proposal. IACR Cryptology ePrint Archive 2011, 238 (2011)
33. Rabin, M.: Digitalized Signatures. In: De Millo, R., Dobkin, D., Jones, A., Lipton, R. (eds.) Foundations of Secure Computation, pp. 155–168. Academic Press (1978)
34. Mennink, B.: Optimal Collision Security in Double Block Length Hashing with Single Length Key. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 526–543. Springer, Heidelberg (2012)
35. Merkle, R.C.: One Way Hash Functions and DES. In: Brassard, G. (ed.) Advances in Cryptology - CRYPT0 1989. LNCS, vol. 435, pp. 428–446. Springer, Heidelberg (1990)
36. Nandi, M., Lee, W.I., Sakurai, K., Lee, S.-J.: 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)

37. Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., Walker, J.: Skein Source Code and Test Vectors, `http://www.skein-hash.info/downloads`
38. Özen, O., Stam, M.: Another Glance at Double-Length Hashing. In: Parker, M.G. (ed.) Cryptography and Coding 2009. LNCS, vol. 5921, pp. 176–201. Springer, Heidelberg (2009)
39. Peyrin, T., Gilbert, H., Muller, F., Robshaw, M.J.B.: Combining Compression Functions and Block Cipher-Based Hash Functions. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 315–331. Springer, Heidelberg (2006)
40. 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)
41. 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)
42. Satoh, T., Haga, M., Kurosawa, K.: Towards Secure and Fast Hash Functions. TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems (1999)
43. Stam, M.: Beyond Uniformity: Better Security/Efficiency Tradeoffs for Compression Functions. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 397–412. Springer, Heidelberg (2008)
44. Stam, M.: Blockcipher-Based Hashing Revisited. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 67–83. Springer, Heidelberg (2009)
45. 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)
46. Robert, S., Winternitz: A Secure One-Way Hash Function Built from DES. In: IEEE Symposium on Security and Privacy, pp. 88–90 (1984)

A Related Work

This part summarizes related work regarding to single- and double-block-length hash functions.

Double-Block-Length Schemes. The essentially first double-block-length hash functions were presented by Merkle [35], who proposed three constructions on the basis of DES. Today, there are four so-called "classical" double-block-length constructions, which were introduced in the early 1990s: MDC-2, MDC-4, Abreast-DM, and Tandem-DM. MDC-2 and MDC-4 [8,20] are (n, n)-bit double-block-length hash functions with rates 1/2 and 1/4, respectively. For MDC-2, Steinberger [45] proved in 2006 that no adversary asking less than $2^{74.9}$ queries will obtain a significant advantage at finding a collision. In a sophisticated proof, it was shown by Fleischmann, Forler, and Lucks [11] in 2012, that for MDC-4 an adversary requires at least $2^{74.7}$ queries to find a collision with an advantage of 1/2.

Concerning rate-1 double-block-length hash functions, Lucks [31] presented a first construction at Dagstuhl'07. Stam [44] also proposed a rate-1 single-call double-block-length function, for which he showed an almost-optimal collision-resistance, up to a logarithmic factor. However, while Lucks and Stam claimed a rate-1 property for their constructions, those are actually much slower, as pointed out by Luo and Lai [32]. At CRYPTO'93, Hohl et al. [19] analyzed the security of compression functions of rate-1/2 double-block-length hash functions. In 1998, Knudsen, Lai, and Preneel [21] discussed the security of rate-1 double-block-length hash functions. In 1999, Satoh, Haga, and Kurosawa [42] as well as Hattori, Hirose, and Yoshida [15] in 2003 attacked rate-1 double-block-length hash functions. At FSE'05, Nandi et al. [36] presented a rate-2/3 compression function, which was later analyzed by Knudsen and Muller at ASIACRYPT'05 [22]. At CT-RSA'11, Lee and Stam [28] presented a faster alternative to MDC-2, called MJH.

Double-Block-Length Schemes with Birthday-Type Collision Security.

ABREAST-DM and TANDEM-DM base on the famous Davies-Meyer scheme, and have been presented by Lai and Massey [24] at EUROCRYPT'92. In 2004, Hirose added a large class of rate-1/2 double-block-length hash functions, composed of two independent $(2n, n)$-bit block ciphers, with $2n$ being the key and n the block size [16] . At FSE'06, he proposed a new scheme called HIROSE-DM [17], which dropped the requirement of independent ciphers, and for which he provided a collision-security proof in the ideal-cipher model, stating that no adversary asking less than $2^{124.55}$ queries can find a collision with probability $\geq 1/2$.

In [39], Peyrin et al. analyzed techniques to construct larger compression functions by combining smaller ones. The authors proposed $3n$-to-$2n$-bit and $4n$-to-$2n$-bit constructions composed of five public functions, yet they did not show proofs for their concepts.

In 2008, Chang et al. introduced a generic framework for purf-based multi block length constructions [7], where purf denotes a public random function.

Considering TANDEM-DM, Fleischmann, Gorski, and Lucks [13] gave a collision-security proof at FSE'09, showing that no adversary can obtain a significant advantage without making at least $2^{120.4}$ queries. In 2010, Lee, Stam, and Steinberger [29] have shown that the proof of Fleischmann et al. has several non-trivial flaws. Further, they provided a bound of $2^{120.87}$ queries for a collision adversary.

For ABREAST-DM, Fleischmann, Gorski, and Lucks [14] as well as Lee and Kwon presented, independent from each other, collision-security bound of $2^{124.42}$ queries. More general, [14] introduced the class notion of CYCLIC-DL, which included the constructions ABREAST-DM, CYCLIC-DM, ADD-K-DM, and CUBE-DM, and applied similar proofs for these. At IMA'09, Özen and Stam [38] proposed a framework for double-block-length hash functions by extending the generalized framework by Stam at FSE'09 for single-call hash functions. Still, their framework based on the usage of two independent block ciphers. At ProvSec'10, Fleischmann et al. [10] extended their general classification of double-block-length hash functions by the classes GENERIC-DL, SERIAL-DL, and PARALLEL-DL. For the framework by Özen and Stam, they relaxed the requirement of distinct independent block ciphers and gave collision bounds for TANDEM-DM and CYCLIC-DM. In [23], Krause, Armknecht, and Fleischmann provided techniques for proving asymptotically-optimal preimage-resistance bounds for block-cipher-based double-length, double-call hash functions. They introduced a new Davies-Meyer double-block-length hash function for which they proved that no adversary asking less than 2^{2n-5} queries can find a preimage with probability $\geq 1/2$. At ACISP'12, Fleischmann et al. [12] showed a very similar Davies-Meyer construction – called WEIMAR-DM– for which they could prove the currently best collision-security bound of $2^{126.23}$ queries, and the currently best preimage-security bound among the previously known double-block-length hash function.