Security of Cyclic Double Block Length Hash Functions including Abreast-DM

Ewan Fleischmann, Michael Gorski, Stefan Lucks
{ewan.fleischmann,michael.gorski,stefan.lucks}@uni-weimar.de

Bauhaus-University Weimar, Germany

Abstract. We provide the first proof of security for Abreast-DM, one of the oldest and most well-known constructions for turning a block cipher with n-bit block length and $2n$-bit key length into a $2n$-bit cryptographic hash function. In particular, we prove that when Abreast-DM is instantiated with AES-256, *i.e.* a block cipher with 128-bit block length and 256-bit key length, any adversary that asks less than $2^{124.42}$ queries cannot find a collision with success probability greater than 1/2. Surprisingly, this about 15 years old construction is one of the few constructions that have the desirable feature of a near-optimal collision resistance guarantee.

We generalize our techniques used in the proof of Abreast-DM to a huge class of double block length (DBL) hash functions that we will call cyclic. Using this generalized theorem we are able to derive several DBL constructions that lead to compression functions that even have a higher security guarantee and are more efficient than Abreast-DM. Furthermore we give DBL constructions that have the highest security guarantee of all DBL compression functions currently known in literature. We also provide an analysis of preimage resistance for cyclic compression functions. Note that this work has been already presented at Dagstuhl '09.

Keywords: cryptographic hash function, block cipher based, proof of security, double-block length, ideal cipher model, Abreast-DM.

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 is one of the most important primitives in cryptography [23].

Block Cipher-Based Hash Functions. Since their initial design by Rivest, MD4-family hash functions (*e.g.* MD4, MD5, RIPEMD, SHA-1, SHA2 [3, 26, 27, 29, 30]) have dominated cryptographic practice. But in recent years, a sequence of attacks on these type of functions [8, 11, 38, 39] 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 [22]. 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* (DBL) hash functions [2]. A DBL 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.

Our Contribution. Four, somewhat 'classical' DBL hash functions are known: MDC-2, MDC-4, Abreast-DM and Tandem-DM [4, 5, 21]. At EUROCRYPT'07 and FSE'09 security bounds for MDC-2 and Tandem-DM had been shown [36, 10]. In this article, we will give the first security bound for Abreast-DM in terms of collision- and preimage resistance. Assuming the same hash output length of 256 bits our security bound will state that no adversary asking less than $2^{124.42}$ queries cannot find a collision with probability greater than 1/2. We will generalize our proof techniques to a huge class of DBL compression functions called Cyclic. By applying these methods, we are able to derive compression functions that have an even higher security guarantee than Abreast-DM. Since there are currently only two DBL compression functions known in literature that have a birthday-type security guarantee (for collision resistance), Hirose's FSE'06 construction [14] and Tandem-DM [21, 10], we not only add another compression function to this exclusive club, but also provide a technique for constructing such functions. Using this construction method, we are able to derive practical DBL compression functions that have the highest security guarantee currently known.

We will also prove an upper bound of success if an adversary is trying to find a (second-)preimage. This bound is rather weak as it essentially states, that the success probability of an adversary asking strictly less than 2^n queries is asymptotically negligible.

Outline. The paper is organized as follows: Section 2 includes formal notations and definitions as well as a review of related work. In Section 3, we proof that any adversary asking less than $2^{124.42}$ oracle queries has negligible advantage in finding a collision for the Abreast-DM compression function. Section 4 generalizes our techniques to a huge class of DBL compression functions and give security bounds in terms of collision resistance and preimage resistance. Section 5 discusses how we can use the results from the previous section to derive new DBL compression functions that have the highest security guarantee of all currently known DBL compression functions. In Section 6 we discuss our results and conclude.

2 Preliminaries

2.1 Iterated DBL Hash Function Based on block ciphers

Ideal Cipher Model. A block cipher is a keyed family of permutations consisting of two paired algorithms $E : \Omega \times \mathcal{K} \to \Omega$ and $E^{-1} : \Omega \times \mathcal{K} \to \Omega$ where Ω is the set of plaintexts/ciphertexts, and $\mathcal{K}$ the set of keys. If $\Omega = \{0,1\}^n$ and $\mathcal{K} = \{0,1\}^k$, we will call it an (n,k)-block cipher. Let $\mathrm{BC}(\Omega, \mathcal{K})$ be the set of all such block ciphers. Now, for any one fixed key $K \in \mathcal{K}$, decryption $E_K^{-1} = E^{-1}(\cdot, K)$ is the inverse function of encryption $E_K = E(\cdot, K)$, so that $E_K^{-1}(E_K(X)) = X$ holds for any input $X \in \Omega$.

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 the ideal cipher 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 $\mathrm{BC}(\mathcal{X}, \mathcal{K})$.

DBL Compression Functions. Iterated DBL hash functions with two block cipher calls in their compression function are discussed in this article. A hash function $H : \{0,1\}^* \to \mathcal{X}^2$ can be built by iterating a compression function $F : \Omega^2 \times \{0,1\}^b \to \Omega^2$ as follows: Split the padded message M

into b-bit blocks $M_1, \ldots, M_l$, fix (G_0, H_0), apply $(G_i, H_i) = F(G_{i-1}, H_{i-1}, M_i)$ for $i = 1, \ldots, l$ and finally set $H(M) := (G_l, H_l)$. Let the compression function F be such that

$$(G_i, H_i) = F(G_{i-1}, H_{i-1}, M_i),$$

where $G_{i-1}, H_{i-1}, G_i, H_i \in \Omega$ and $M_i \in \{0,1\}^b$. We assume that the compression function F consists of F_T, the top row, and F_B, the bottom row. Each of the component functions F_B and F_T performs exactly *one* call to the block cipher and can be defined as follows:

$$\begin{aligned} G_i &= F_T(G_{i-1}, H_{i-1}, M_i) = E(X_T, K_T) \oplus Z_T, \\ H_i &= F_B(G_{i-1}, H_{i-1}, M_i) = E(X_B, K_B) \oplus Z_B, \end{aligned}$$

where X_T, K_T, Z_T and X_B, K_B, Z_B are uniquely determined by G_{i-1}, H_{i-1}, M_i.
We define the rate r of a block cipher based compression/hash function F by

$$r = \frac{|M_i|}{(\text{number of block cipher calls in F}) \times n}.$$

The key scheduler rate r_{key} is defined as

$$r_{key} = \frac{1}{\text{number of key scheduler operations per compression function}}.$$

It follows that $r_{key} = 1$ if $K_T = K_B$ and $r_{key} = 1/2$ otherwise. They both are a measure of efficiency for such block cipher based constructions. Note that there is currently a discussion in literature on how to measure this efficiency 'correctly'.

2.2 Defining Security – Collision Resistance of a Compression Function

Insecurity is quantified by the success probability of an optimal resource-bounded adversary. The resource is the number of queries to the ideal cipher oracles E or E^{-1}. For a set S, let $z \overset{R}{\leftarrow} S$ represent random sampling from S under the uniform distribution. For a probabilistic algorithm $\mathcal{M}$, let $z \overset{R}{\leftarrow} \mathcal{M}$ mean that z is an output of $\mathcal{M}$ and its distribution is based on the random choices of $\mathcal{M}$.

An adversary is a computationally unbounded but always-halting collision-finding algorithm $\mathcal{A}$ with access to an oracle $E \in \mathrm{BC}(\mathcal{X}, \mathcal{K})$. We can assume (by standard arguments) that $\mathcal{A}$ is deterministic. The adversary may make a *forward* query $(X, K, ?)_{fwd}$ to discover the corresponding value $Y = E_K(X)$, or the adversary may make a *backward* query $(?, K, Y)_{bwd}$, 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) and the *query history*, denoted $\mathcal{Q}$, 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.

The adversary's goal is to output two different triplets (G, H, M) and (G', H', M') such that $F(G, H, M) = F(G', H', M')$. Since E is assumed to be an ideal cipher, we impose the reasonable condition that the adversary must have made all queries necessary to compute $F(G, H, M)$ and

$F(G', H', M')$. We will in fact dispense the adversary from having to output these two triplets, and simply determine whether the adversary has been successful or not by examining its 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.

Definition 1. ***(Collision resistance of a compression function)*** *Let F be a blockcipher based compression function, $F : \Omega^2 \times \{0,1\}^b \to \Omega^2$. Fix an adversary $\mathcal{A}$. Then the advantage of $\mathcal{A}$ in finding collisions in F is the real number*

$$\begin{aligned}\mathbf{Adv}_F^{\textsc{Coll}}(\mathcal{A}) = \Pr[&E \overset{R}{\leftarrow} \mathrm{BC}(\mathcal{X}, \mathcal{K}); ((G,H,M),(G',H',M')) \overset{R}{\leftarrow} \mathcal{A}^{E,E^{-1}} : \\ &((G,H,M) \neq (G',H',M')) \wedge F(G,H,M) = F(G',H',M')].\end{aligned}$$

For $q \geq 1$ we write

$$\mathbf{Adv}_F^{\textsc{Coll}}(q) = \max_{\mathcal{A}}\{\mathbf{Adv}_F^{\textsc{Coll}}(\mathcal{A})\}$$

where the maximum is taken over all adversaries that ask at most q oracle queries (*i.e.* E and E^{-1} queries).

2.3 Related Work

Schemes with non-optimal or unknown collision resistance. Preneel *et al.* [28] discussed the security of SBL hash functions against several generic attacks. They concluded that 12 out of 64 hash functions are secure against the 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.* [15] analyzed the security of DBL compression functions with rate 1 and 1/2. Satoh *et al.* [34] and Hattoris *et al.* [12] discussed DBL hash functions with rate 1 composed of $(n, 2n)$-block ciphers. MDC-2 and MDC-4 [16, 1, 5] are (n, n)-block cipher based DBL hash functions with rates 1/2 and 1/4, respectively. Steinberger [36] 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.* [25] 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 [32, 33] 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.

Schemes with Birthday-Type Collision Resistance. Merkle [24] 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 [13] presented a class of DBL hash functions with rate 1/2 which are composed of two different and independent $(n, 2n)$-block ciphers that have birthday-type collision resistance. At FSE'06, Hirose [14] presented a rate 1/2 and $(n, 2n)$-block cipher based DBL hash function that has birthday-type collision resistance. He essentially 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). At FSE'09, Fleischmann et. al. [10] that for $\textsc{Tandem}$-DM, no adversary asking less than $2^{120.4}$ queries can find a collision with probabilty greater than 1/2.

3 Security of Abreast-DM

3.1 Compression Function

The Abreast-DM hash function was proposed at EUROCRYPT '92 by Xuejia Lai and James L. Massey [21]. It incorporates two Davies-Meyer (DM) single block length compression functions [23] which are used side-by-side. The compression function is illustrated in Figure 1 and is formally given in Definition 2.

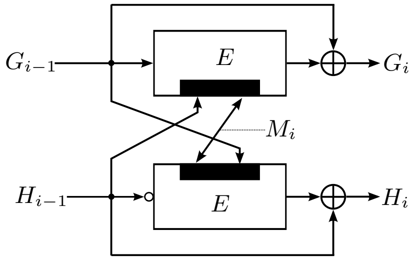


Figure 1. The compression function F^{ADM} of Abreast-DM, the small circle '$\circ$' denotes a bit-by-bit complement

Definition 2. *Let* $\mathrm{F}^{\text{ADM}} : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^{2n}$ *be a compression function such that* $(G_i, H_i) = F^{\text{ADM}}(G_{i-1}, H_{i-1}, M_i)$ *where* $G_i, H_i, M_i, G_{i-1}, H_{i-1} \in \{0,1\}^n$. F^{ADM} *consists of a* $(n, 2n)$*-block cipher* E *as follows:*

$$G_i = G_{i-1} \oplus E_{H_{i-1}|M_i}(G_{i-1})$$
$$H_i = H_{i-1} \oplus E_{M_i|G_{i-1}}(\overline{H}_{i-1}),$$

where $\overline{H}$ *denotes the bit-by-bit complement of* H.

The compression function F^{ADM} requires two invocations of the block cipher E to produce an output. Note that these two block cipher invocations can be computed in parallel. Normally, E would be assumed to be AES-256 and therefore $n = 128$.

3.2 Security Results

Our discussion will result in proofs for the following bounds as given by Theorems 1 and 2.

Theorem 1. *(Collision Resistance) Let* $F := F^{\text{ADM}}$ *as in Definition 2 and* n, q *be natural numbers with* $q < 2^{n-2.58}$. *Then*

$$\mathbf{Adv}_F^{\text{Coll}}(q) \leq 18 \left(\frac{q}{2^{n-1}}\right)^2.$$

The following corollary explicitly states what this Theorem means for $n = 128$.

Corollary 1. *For the compression function* Abreast-DM*, instantiated with AES-*256[1] *any adversary asking less than* $q = 2^{124.42}$ *(backward or forward) oracle queries cannot usually find a collision.*

[1] Formally, we model the AES-256 block cipher as an ideal block cipher.

Theorem 2. *(Preimage Resistance) Let $F := F^{\text{ADM}}$ be as in Definition 2. For every $N' = 2^n - q$ and $q > 1$*

$$\mathbf{Adv}_F^{\text{INV}}(q) \leq 2q/(N')^2.$$

The proof of for Theorem 1 is given in Section 3.3, the proof of Theorem 2 is a simple corollary of Theorem 5 and will be omitted. Using Theorem 1 and simple calculus, it is easy to see that the compression function is asymptotically optimal for $n \to \infty$ since

$$\lim_{n\to\infty} \mathbf{Adv}_F^{\text{COLL}}(q) = \frac{q^2}{2^{2n}}.$$

3.3 Collision Resistance – Proof of Theorem 1

Analysis Overview. We will analyze if the queries made by the adversary contain the means for constructing a collision of the compression function F^{ADM}. Two queries to the oracles E, E^{-1} in total are required to compute the output (G_i, H_i) of F^{ADM} for any given input (G_{i-1}, H_{i-1}, M_i). It is easy to see, that one oracle query uniquely determines the other query. Effectively, we look to see whether there exist four (not necessarily distinct) queries that form a collision (see Figure 2).

To upper bound the probability of the adversary obtaining queries than can be used to construct a collision, we upper bound the probability of the adversary making a query that can be used as the final query to complete such a collision. Namely, for each i, $1 \leq i \leq q$, we upper bound the probability that the answer to the adversary's i-th query $(X_i, K_i, ?)_{fwd}$ or $(?, K_i, Y_i)_{bwd}$ will allow the adversary to use the i-th query to complete the collision. In the latter case, we say that the i-th query is 'successful' and we give the attack to the adversary.

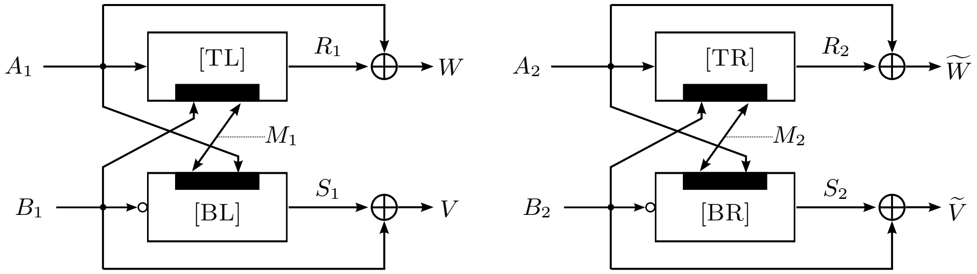


Figure 2. Notations used for a collision of Abreast-DM; $\text{Coll}^{ADM}(\mathcal{Q})$, in this case $W = \widetilde{W}$ and $V = \widetilde{V}$.

Naturally, the computation of any single compression function depends on *two* block cipher calls – assuming that the construction does not allow to use one and the same query in the top- and the bottom row of the compression function as in the case of Abreast-DM. In order to upper bound the success probability for any single query mounted by the adversary, we have to upper bound the maximal number of compression functions the adversary can complete with this single query result. At a first glance any single query can be used in the *top row* or in the *bottom row* of a compression function. Say, *e.g.*, the query is $(A, B|M, R)$ and the adversary intends to use it in the top row. As it is practically impossible to track, whether the adversary has mounted the corresponding bottom row query $(\overline{B}, M|A, S)$ in the past, we have to assume that the adversary has access to this query.

Formally, we give the adversary this query *for free*. In general, just this *free* query that is intended for use in the bottom row, can be used also in the top row by the adversary in order to start the computation of a new compression function. And, again, as we cannot say wheter the adversary has access to the corresponding bottom row query, we have to assume that the adversary has access to it (*i.e.* we will give it for free to him). Pursuing this process seems to result in a practically infinite action. But it does not for Abreast-DM (and for all Cyclic compression functions, see Section 4) as will will discuss now.

On Abreast-DM*'s Cycle.* Assume that the adversary mounts his i-th query denoted by $Q_{6i} = (A, B|M, R)$. For the ease of presentation, we will give the adversry's 'first' query index zero and therefore $i \in \{0, 1, \dots, q-1\}$ assuming that the adversary mounts q queries in total. Also we give the adversary's i-th query index $6i$ for reasons that will become clear later – in short, this is due to the 5 'free' queries the adversary is given for any single mounted query. First assume that the query Q_{6i} is used in the top row. The adversary is given for free the corresponding query of the bottom row $Q_{6i+1} := (\overline{B}, M|A, S)$. Using Q_{6i} and Q_{6i+1}, the adversary is able to compute one result of a compression function

$$(W_1, V_1) := F^{\text{ADM}}(A, B, M) = (E_{B|M}(A) \oplus A, E_{M,A}(\overline{B}) \oplus B).$$

As the adversary can (re)use the free query Q_{6i+1} in the top row, we give the adversary for free the corresponding query $Q_{6i+2} = (\overline{M}, A|\overline{B}, S)$ for the bottom row. After this free query, the adversary can compute

$$(W_2, V_2) := F^{\text{ADM}}(\overline{B}, M, A) = (E_{M|A}(\overline{B}) \oplus \overline{B}, E_{A|\overline{B}}(\overline{M}) \oplus M)$$

using the queries Q_{6i+1} and Q_{6i+2}. The continuation of this process is summarized in Table 1. Likewise, the adversary is given for free the (forward) queries $Q_{6i+3}, Q_{6i+4}, Q_{6i+5}$. The query Q_{6i+6} is equal to the initial query of the adversary Q_{6i} and this process comes to an end.

Note that the query numbers in parentheses denote a reuse of a previous query, *e.g.* (6i+1) denotes the reuse of query Q_{6i+1} in another position (top/bottom).

As indicated by Table 1, *any* single query is used in the bottom row as well as in the top row. Since any single query (used in the top- or bottom position) uniquely determines the corresponding query in the bottom- or top position it follows that the all queries can only be used for the compression functions mentioned in Table 1.

Analysis Details. Fix numbers n, q and an adversary $\mathcal{A}$ asking q forward or backward queries to its oracle E in total. Let $\text{Coll}^{\text{ADM}}(\mathcal{Q})$ be the event that the adversary can construct a collision of F^{ADM} using the queries in $\mathcal{Q}$. The term 'last query' means the latest query made by the adversary. It is always the i-th query of the adversary and it is always denoted as Q_{6i}. This is due to the fact that we will give the adversary, for any single query, 5 additional free queries. We examine the adversary's mounted forward queries $(X_{6i}, K^1_{6i}|K^2_{6i}, ?)_{fwd}$ or backward queries $(?, K^1_{6i}|K^2_{6i}, Y_{6i})_{bwd}$ one at a time as they come in. Similarly, the free queries are also examined one at a time as they are given to the adversary.

We say a query $Q_m = (X_m, K^1_m|K^2_m, Y_m)$ is successful if the output – Y_m for a forward query or X_m for a backward query – is such that the adversary can use this very query Q_m to form a collision. More precise there are are three queries Q_j, Q_k, Q_l in the query history $\mathcal{Q}$ such that

$F^{\text{ADM}}(\cdot)$	Query #	Plaintext	Key	Ciphertext	Chaining Value
(A, B, M)	$6i$ (*)	A	$B\vert M$	R	$W_1 = R \oplus A$
	$6i+1$	$\overline{B}$	$M\vert A$	S_1	$V_1 = S \oplus B$
$(\overline{B}, M, A)$	$(6i+1)$	$\overline{B}$	$M\vert A$	S_1	$W_2 = S_1 \oplus \overline{B}$
	$6i+2$	$\overline{M}$	$A\vert\overline{B}$	S_2	$V_2 = S_2 \oplus M$
$(\overline{M}, A, \overline{B})$	$(6i+2)$	$\overline{M}$	$A\vert\overline{B}$	S_2	$W_3 = S_2 \oplus \overline{M}$
	$6i+3$	$\overline{A}$	$\overline{B}\vert\overline{M}$	S_3	$V_3 = S_3 \oplus A$
$(\overline{A}, \overline{B}, \overline{M})$	$(6i+3)$	$\overline{A}$	$\overline{B}\vert\overline{M}$	S_3	$W_4 = S_3 \oplus \overline{A}$
	$6i+4$	B	$\overline{M}\vert\overline{A}$	S_4	$V_4 = S_4 \oplus \overline{B}$
$(B, \overline{M}, \overline{A})$	$(6i+4)$	B	$\overline{M}\vert\overline{A}$	S_4	$W_5 = S_4 \oplus B$
	$6i+5$	M	$\overline{A}\vert B$	S_5	$V_5 = S_5 \oplus \overline{M}$
$(M, \overline{A}, B)$	$(6i+5)$	M	$\overline{A}\vert B$	S_5	$W_6 = S_5 \oplus M$
	$(6i)$	A	$B\vert M$	R	$V_6 = R \oplus \overline{A}$

Table 1. Starting by the i-th query of the adversary, Q_{6i}, either $(A, B|M, ?)_{fwd}$ or $(?, B|M, R)_{bwd}$, the adversary is given 5 forward queries, query #'s $6i+1, 6i+2, 6i+3, 6i+4, 6i+5$, for free. In total, he is able to compute 6 complete compression functions F^{ADM} by using these 6 queries. (*) This is the only query the adversary has mounted.

the four (not necessarily pairwise different) queries Q_m, Q_j, Q_k, Q_l can be used for a collision (see Figure 2). The goal is thus to upper bound the adversary's chance of ever making a successful last query.

We now upper bound $\Pr[\textsc{Coll}^{\text{ADM}}(\mathcal{Q})]$ by exhibiting predicates $\textsc{Win}_0(\mathcal{Q}), \ldots, \textsc{Win}_{q-1}(\mathcal{Q})$ such that $\textsc{Coll}^{\text{ADM}} \Longrightarrow \textsc{Win}_1(\mathcal{Q}) \vee \ldots \vee \textsc{Win}_q(\mathcal{Q})$. Then, $\Pr[\textsc{Coll}^{\text{ADM}}(\mathcal{Q})] \leq \textsc{Win}_0(\mathcal{Q}) + \ldots + \textsc{Win}_{q-1}(\mathcal{Q})$.

Since the adversary mounts q queries in total, we informally say that $\textsc{Win}_i(\mathcal{Q})$, $0 \leq i \leq q-1$ holds if the adversary finds a collision after mounting the i-th query using at least one of the following queries $Q_{6i}, \ldots, Q_{6i+5}$ conditioned on the fact that the adversary has not been successful before.

Notation: Let $\mathcal{Q}_k$ denote the first k queries made by the adversary or the adversary had been given for free: $\mathcal{Q}_k = \cup_{0 \leq j \leq k} Q_j$ and $|\mathcal{Q}_k| = k+1$.

To formally define the predicates $\textsc{Win}_i(\mathcal{Q})$ the following Definitions are useful.

Definition 3. *We say that a pair of queries* (a, b) is successful in $\mathcal{Q}_c$*, if the query* Q_a *is used in the top row,* Q_b *in the bottom row in the computation of a compression function* F^{ADM} *and there exists a pair of queries* $Q_j, Q_k \in \mathcal{Q}_c$ *such that a collision for* F^{ADM} *can be computed:*

$$X_a \oplus Y_a = X_j \oplus Y_j \qquad \textit{and} \qquad \overline{X_b} \oplus Y_b = \overline{X_k} \oplus Y_k.$$

Definition 4. *Let* $d = 0, \ldots, 5$*,* $d' = d + 1 \ mod\, 6$*,* $\widetilde{d} = \max(d, d')$*. We say* $\textsc{CollFit}_i^d(\mathcal{Q})$ *if (i) the pair of queries* $(6i+d, 6i+d')$ *is successful in* $\mathcal{Q}_{6i+\widetilde{d}}$ *and (ii) the adversary had not been successful for* $0 \leq t \leq d-1$*:* $\neg\textsc{CollFit}_i^t(\mathcal{Q})$*.*

The predicates $\textsc{Win}_i(\mathcal{Q})$ are defined as follows:

Definition 5. *For* $0 \leq i \leq q-1$,

$$\textsc{Win}_i(\mathcal{Q}) = \neg \left(\bigvee_{0 \leq j \leq i-1} \textsc{Win}_j(\mathcal{Q}) \right) \wedge \left(\textsc{CollFit}_i^0(\mathcal{Q}) \vee \ldots \vee \textsc{CollFit}_i^5(\mathcal{Q}) \right).$$

We now show that our case analysis is complete.

Lemma 1. $\textsc{Coll}^{\text{ADM}}(\mathcal{Q}) \Longrightarrow \textsc{Win}_0(\mathcal{Q}) \vee \ldots \vee \textsc{Win}_{q-1}(\mathcal{Q})$.

Proof. Say $\textsc{Coll}^{\text{ADM}}(\mathcal{Q})$. Then a collision can be constructed from the queries $\mathcal{Q}$. That is, our query history $\mathcal{Q}$ contains queries Q_i, Q_j, Q_k, Q_l (see Figure 2) that can be used in positions TL, TR, BL and BR, TL $\neq$ TR, such that $V = \widetilde{V}$ and $W = \widetilde{W}$. Note that the condition TL $\neq$ TR suffices to ensure that a collision from two different inputs has occurred. It is easy to see that no query mounted directly by the adversary can be successful since any such query only can only serve for either a top- or bottom row position in the compression function F^{ADM}. Also, the corresponding query necessary to compute the complete compression function will be given to the adversary for free *after* he has mounted a query. So the adversary can only be successful in the phase where he is given the free queries one after another. Say the adversary is successful during the phase where he gets the free queries following his i-th query. We can safely assume that this is the first time the adversary has found such a collision and therefore i is minimal. Then $\neg\textsc{Win}_j(\mathcal{Q})$, $1 \leq j < i$, $\textsc{CollFit}_i^d(\mathcal{Q})$ such that $d \in \{0, 1, \ldots, 5\}$ is minimal and therefore $\textsc{Win}_i(\mathcal{Q})$. This proves our claim. (□)

Since $\Pr[\textsc{Coll}^{\text{ADM}}(\mathcal{Q})] \leq \sum_{j=0}^{q-1} \textsc{Win}_j(\mathcal{Q})$ it follows that

$$\Pr[\textsc{Coll}^{\text{ADM}}(\mathcal{Q})] \leq \sum_{i=1}^{q} \sum_{d=0}^{5} \textsc{CollFit}_i^d(\mathcal{Q}). \tag{1}$$

We will now upper bound the probability of $\textsc{CollFit}_i^d(\mathcal{Q})$.

Lemma 2. *Let* $1 \leq i \leq q$ *and* $0 \leq d \leq 5$. *Then*

$$\Pr[\textsc{CollFit}_i^d(\mathcal{Q})] \leq \frac{6i}{(2^n - 6i)^2}$$

Proof. Let $d' = d + 1 \ mod 6$. The output of the compression function F^{ADM}, (W, V), is uniquely determined by the queries $Q_{6i+d} = (X_{6i+d}, K_{6i+d}, Y_{6i+d})$ and $Q_{6i+d'} = (X_{6i+d'}, K_{6i+d'}, Y_{6i+d'})$,

$$W = Y_{6i+d} \oplus X_{6i+d} \qquad \text{and} \qquad V = Y_{6i+d'} \oplus \overline{X_{6i+d'}}.$$

Both W, V depend on the plaintext and the ciphertext of E. If Q_{6i+d} was received by a forward query, the key and the plaintext are fixed. As the result of the query, the ciphertext, is chosen uniformly random from the set $\{0,1\}^n$ (since we assume that E is an ideal cipher), it follows that W is randomly determined by the answer of the oracle. In the case of a backward query, the key and the ciphertext are fixed. The result of this query, the plaintext, is chosen uniformly random and it follows again that W is randomly determined by the answer of the oracle. Using the same arguments, it follows that V is also randomly determined by the answer of the oracle. Note that the bit-by-bit inversion in the bottom row of $X_{6i+d'}$ does *not* change any of our arguments.

To form a collision, two queries Q_j, Q_k are needed that can be chosen from at most $6(i+1)$ queries in $\mathcal{Q}_{6(i+1)-1}$. The adversary can use them to compute the output of $< 6(i+1)$ compression functions F^{ADM}. Therefore,

$$\Pr[\textsc{CollFit}_i^d(\mathcal{Q})] \leq \frac{6(i+1)}{(2^n - 6(i+1))^2}.$$

(□)

Using (1) we get the following upper bound for any $q < 2^{n-\log_2 6} = 2^{n-2.58}$

$$\begin{aligned}\Pr[\textsc{Coll}^{\text{ADM}}(\mathcal{Q})] &\leq \sum_{i=0}^{q-1}\sum_{d=0}^{5} \frac{6(i+1)}{(2^n - 6(i+1))^2} \leq \sum_{i=1}^{q}\sum_{d=0}^{5} \frac{6i}{(2^n - 6i)^2} \\ &\leq \sum_{i=1}^{q} \frac{36i}{(2^n - 6i)^2} \leq \frac{36 \cdot q^2 \cdot \frac{1}{2}}{(2^n - 6i)^2} \leq 18 \left(\frac{q}{2^{n-1}}\right)^2\end{aligned}$$

This completes our proof of Theorem 1. ■

Note that this bound is not meaningful even for $q \approx 2^{n-2.58}$ since $18q^2/2^{2n-2}$ would be larger than one. Bounds for q in the case $n = 128$ have been discussed in Section 3.2. Note that the power of the arguments stems from the fact that we can tightly upper bound the number of compression functions that an adversary can compute given an upper bound of queries mounted by an adversary.

4 Security of Cyclic Hash Functions

In this section, we will generalize the definitions and techniques of the previous section.

4.1 Cyclic Compression Functions

Definition 6. *Let $(\Omega, *)$ be a group, $N = |\Omega|$. Let $F^{\text{CYC}} : \Omega^2 \times \{0,1\}^b \longrightarrow \Omega^2$ be a compression function such that $(G_i, H_i) = F^{\text{CYC}}(G_{i-1}, H_{i-1}, M_i)$ where $G_{i-1}, H_{i-1}, G_i, H_i \in \Omega$ and $M_i \in \{0,1\}^b$, $b > 0$. Let $E \in BC(\Omega, \Omega \times \{0,1\}^b)$ be a block cipher; ρ and σ permutations on the set $\Omega^2 \times \{0,1\}^b$ and π^T, π^B permutations on Ω. Let $Z := (G_{i-1}, H_{i-1}, M_i) \in \Omega^2 \times \{0,1\}^b$. Then $X^T, X^B \in \Omega$, $K^T, K^B \in \Omega \times \{0,1\}^b$ such that $(X^T, K^T) = \rho(Z)$ and $(X^B, K^B) = \sigma(\rho(Z))$. Now F^{CYC} consists of a E as follows:*

$$\begin{cases} G_i = E_{K^T}(M^T) * \pi^T(X^T) \\ H_i = E_{K^B}(M^B) * \pi^B(X^B) \end{cases}$$

where the computation leading to G_i is informally called the 'top row', and for H_i called 'bottom row'.

The compression function F^{CYC} is visualized in Figure 3.

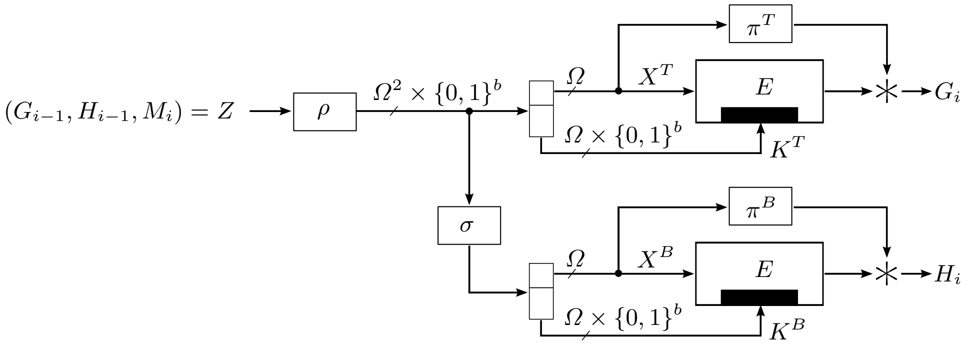

$(G_{i-1}, H_{i-1}, M_i) = Z$
ρ
$\Omega^2 \times \{0,1\}^b$
π^T
Ω
X^T
E
G_i
$\Omega \times \{0,1\}^b$
K^T
σ
π^B
Ω
X^B
E
H_i
$\Omega \times \{0,1\}^b$
K^B

Figure 3. Cyclic Compression Function $(G_i, H_i) = F^{\mathrm{CYC}}(Z)$, $Z = (G_{i-1}, H_{i-1}, M_i)$

Since the properties of the permutation σ are highly relevant for the proof, we will discuss them now. The following Definitions 7, 8 are in some way the heart of this discussion. They lay the groundwork for defining cyclic double block length compression functions in the first place and provide for a main notion that we will use, the *order of an element* and the *order of a mapping.*

Definition 7. *Let σ be a bijective mapping on a set $\mathcal{S}$ where $\mathcal{S} := \Omega^2 \times \{0,1\}^b$. Let* ID *be the identity mapping on $\mathcal{S}$. The function σ^k is defined as $\sigma^k := \sigma \circ \sigma^{k-1}$ for $k > 0$ and $\sigma^0 :=$* ID*.*

(i) Fix some element $s \in \mathcal{S}$. The order *of s is defined to be $|s| = \min_{r \geq 1}(\sigma^r(s) = s)$,* i.e. *$|s|$ is minimal (but > 0) such that $\sigma^{|s|}(s) = s$.*

(ii) If there is a $c \in \mathbb{N}_{\geq 1}$ such that $\forall \widetilde{s} \in \mathcal{S} : |\widetilde{s}| = c$, we say the oder of the mapping *σ, denoted by $|\sigma|$, is equal to c,* i.e. *$|\sigma| = c$. If there is no such c, then $|\sigma| := 0$. Note that, if $|\sigma| > 0$, the order of σ is equal to the order of* any *element chosen from $\mathcal{S}$.*

Definition 8. *Let F^{CYC}, ρ and σ be as in Definition 6. If $|\sigma| \geq 2$, then F^{CYC} is called a* cyclic *double block length (CDBL) compression function with* cycle length $|\sigma|$.

Properties of CDBL Compression Functions. Now we will discuss the main properties of F^{CYC}. It is easy to see that a CDBL compression function with cycle length 1 is not reasonable as this would essentially F^{CYC} render a *single block length* compression function. A cycle length of 1 would imply $\sigma =$ ID. The values of the initial vector (G_0, H_0) is nonetheless free to be different. Single block length hash functions have already been thoroughly analyzed in [2, 28, 35].

Lemma 3. *Let F^{CYC} be as in Definition 8.*

(i) *Any oracle query for the top- or bottom row of F^{CYC} uniquely determines the oracle query in the bottom- or top row.*

(ii) *The queries used in F^{CYC} for the bottom- and top row are always different,* i.e. *there are no fixed-points.*

Proof. The proof of (i) is trivial and (ii) is a consequence of $|\sigma| > 1$. (□)

(Counter)Examples. In the following, we will give some examples of known DBL constructions and discuss how they match Definition 6.

Abreast-DM. To match Definition 6, we choose $\Omega = \{0,1\}^n$, $b = n$, $\pi^T = \text{ID}$, $\pi^B(X) = \overline{X}$, $\rho(G,H,M) = \text{ID} = (G,H,M)$ and $\sigma(G,H,M) = (\overline{H},M,G)$. As discussed in Section 3.3, it is easy to see that Abreast-DM has a cycle length of $|\sigma| = c = 6$ using Table 1.

Hirose's FSE'06 Proposal. A description of F^{Hirose} is given in Appendix A. In this case we choose $\Omega = \{0,1\}^n$, $b = n$, $\pi^T = \pi^B = \text{ID}$, $\rho = \text{ID}$ and $\sigma(G_{i-1}, H_{i-1}, M_i) = (G_{i-1} \oplus const, H_{i-1}, M_i)$ in order to map with the definition of F^{CYC}. It is easy to see that $|\sigma| = 2$. In [10, Appendix B] it is shown that for F^{Hirose} no adversary asking less than $2^{124.55}$ queries cannot find a collision probability greater than 1/2 given that F^{Hirose} was instantiated with a $(128, 256)$ cipher as, *e.g.*, AES-256.

Tandem-DM. This compression function can be seen as a counter-example. A description of the compression function F^{TDM} is given in Appendix B. Its security was analyzed by Fleischmann et al. at FSE'09 [10] where it was shown that no adversary asking less than $2^{120.4}$ queries cannot find a collision with probability greater than 1/2, given that F^{TDM} was instantiated with a $(128, 256)$ cipher as, *e.g.*, AES-256. As the compression function feeds in the ciphertext of the top row into the bottom row, it cannot be represented as an instantiation of F^{CYC} since the definition does not allow any ciphertext feedback.

4.2 Security Results

Our discussion will result in proofs for the following bounds as given by Theorems 3, 4 and 5.

Theorem 3. *(Collision Resistance for $|\sigma| = 2$) Let $F := F^{\text{CYC}}$ be a cyclic compression function with cycle length $c = |\sigma| = 2$ as in Definition 8. If $\pi^T = \pi^B$, then $a = 1$, else $a = 2$. Then, for any $q > 1$ and $2q < N$,*

$$\mathbf{Adv}_F^{\text{Coll}}(q) \leq \frac{2aq^2}{(N-2q)^2} + \frac{2q}{N-2q}.$$

Theorem 4. *(Collision Resistance for $|\sigma| > 2$) Let $F := F^{\text{CYC}}$ be a cyclic compression function with cycle length $c = |\sigma| > 2$ as in Definition 8. Then, for any $q > 1$ and $cq < N$,*

$$\mathbf{Adv}_F^{\text{Coll}}(q) \leq \frac{c^2}{2}\left(\frac{q}{N-cq}\right)^2.$$

Theorem 5. *(Preimage Resistance) Let $F := F^{\text{CYC}}$ be a cyclic compression function as in Definition 8. Then, for any $q > 1$ and $q < N$,*

$$\mathbf{Adv}_F^{\text{Inv}}(q) \leq 2q/(N-q)^2.$$

Applications will be discussed in Section 5. The proof of Theorem 3 is given in Appendix C. The proof of Theorem 5 is essentially due to Fleischmann et. al. [10, Thm. 2] and can be found in Appendix D. The proof of Theorem 4 is given in Section 4.3.

4.3 Collision Resistance – Proof of Theorem 4

Analysis Overview. In this section we will omit some details that were already discussed in Section 3.3. Again, we will analyze if the queries made by the adversary contain the means for constructing a collision of the compression function F^{CYC}. Similarly as in the proof of Abreast-DM, we upper bound the probability of the adversary making a query that can be used as the final query to complete a collision.

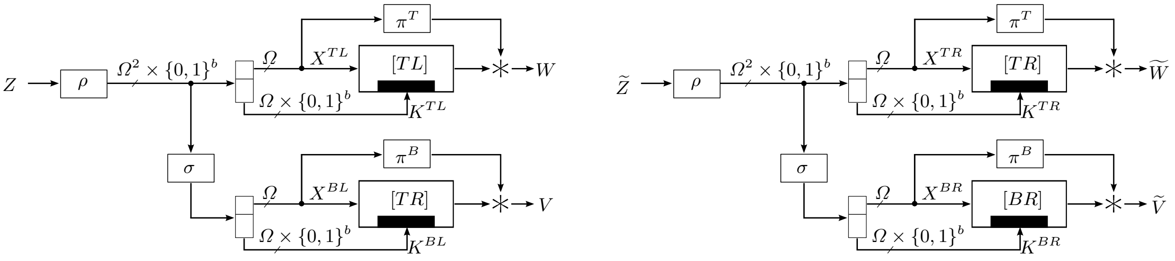


Figure 4. Notations used for a collision of Cyclic: $\text{Coll}^{\text{CYC}}(\mathcal{Q})$, in this case $W = \widetilde{W}$ and $V = \widetilde{V}$ but $Z \neq \widetilde{Z}$.

The cycle in Cyclic. Assume that the adversary mounts a query $Q_{ci} = (X_{ci}, K_{ci}, Y_{ci})$, where $X_{ci}, Y_{ci} \in \Omega$, $K_{ci} \in \Omega \times \{0,1\}^b$, $Y_{ci} = E_{K_{ci}}(X_{ci})$. The query index $c \cdot i$ for the i-th query of the adversary is – similar as in the case of Abreast-DM – due the the $c-1$ free queries the adversary is given for any mounted query. First assume that the query is used in the top row. Let $U_1 = (X_{ci}, K_{ci}) \in \Omega \times (\Omega \times \{0,1\}^b) = \Omega^2 \times \{0,1\}^b$ and $U_2 = (X_{ci+1}, K_{ci+1}) = \sigma(U_1)$ where $X_{ci+1} \in \Omega$ and $K_{ci+1} \in \Omega \times \{0,1\}^b$. The adversary is given for free the corresponding query in the bottom row $Q_{ci+1} = (X_{ci+1}, K_{ci+1}, Y_{ci+1})$, $Y_{ci+1} = E_{K_{ci+1}}(X_{ci+1})$. Given these two queries, the adversary is able to compute one output of the compression function $(W_1, V_1) = F^{\text{CYC}}(\rho^{-1}(U_1))$. The adversary can 'reuse' the query Q_{ci+1} in the top row as a starting point to compute a new result of F^{CYC}. We now give the adversary for free the corresponding bottom row query, Q_{ci+2}, assuming that Q_{ci+1} is used in the top row. For this query Q_{ci+2}, we have $U_3 = (X_{ci+2}, K_{ci+2}) = \sigma(U_2)$ where where $X_{ci+2} \in \Omega$ and $K_{ci+2} \in \Omega \times \{0,1\}^b$ and $Y_{ci+2} = E_{K_{ci+2}}(X_{ci+2})$. Our main observation is that $U_3 = \sigma(U_2) = \sigma^2(U_1)$.

Let $c = |\sigma|$ denote the cycle length of F^{CYC}. This process can be continued. The adversary is given for free the queries $Q_{ci+3}, \ldots, Q_{ci+c-1}$ as is shown in Table 2 in more detail. A cycle is formed since $U_c = \sigma(U_{c-1}) = \ldots = \sigma^c(U_1) = U_1$ and therefore $Q_{ci+c} = Q_{ci}$. The queries forming the cycle are visualized in Figure 5.

Analysis Details. Fix a set Ω, numbers b, q and an adversary $\mathcal{A}$ asking q backward and forward queries to its oracle in total. Let $\text{Coll}^{\text{CYC}}(\mathcal{Q})$ be the event that the adversary is able to construct a collision of F^{CYC} using the queries in $\mathcal{Q}$. The term 'last query' means the latest query made by the adversary and is always given index $c \cdot i$ and denoted as Q_{ci}. We will examine the adversary mounted queries ($d = 0$) and the free queries ($d = 1, 2, \ldots, c-1$), $(X_{ci+d}, K_{ci+d}, ?)_{fwd}$ or $(?, K_{ci+d}, Y_{ci+d})_{bwd}$ one at a time as the adversary gets hold of them. A query $Q_m = (X_m, K_m, Y_m)$ is successful, if it can be used to form a collision using other queries contained in the query history $\mathcal{Q}_m$ as indicated in Figure 4.

$F^{\text{CYC}}(\cdot)$	Query #	Plaintext	Key	Ciphertext	Chaining Value
$\rho^{-1}(U_1)$	ci (*)	X_{ci}	K_{ci}	Y_{ci}	$W_1 = Y_{ci} * \pi^T(X_{ci})$
	$ci+1$	X_{ci+1}	K_{ci+1}	Y_{ci+1}	$V_1 = Y_{ci+1} * \pi^B(X_{ci+1})$
$\rho^{-1}(\sigma(U_1))$	$(ci+1)$	X_{ci+1}	K_{ci+1}	Y_{ci+1}	$W_2 = Y_{ci+1} * \pi^T(X_{ci+1})$
	$ci+2$	X_{ci+2}	K_{ci+2}	Y_{ci+2}	$V_2 = Y_{ci+2} * \pi^B(X_{ci+2})$
$\rho^{-1}(\sigma^2(U_1))$	$(ci+2)$	X_{ci+2}	K_{ci+2}	Y_{ci+2}	$W_3 = Y_{ci+2} * \pi^T(X_{ci+2})$
	$ci+3$	X_{ci+3}	K_{ci+3}	Y_{ci+3}	$V_3 = Y_{ci+3} * \pi^B(X_{ci+3})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\rho^{-1}(\sigma^{c-2}(U_1))$	$(ci+c-2)$	X_{ci+c-2}	K_{ci+c-2}	Y_{ci+c-2}	$W_{c-1} = Y_{ci+c-2} * \pi^T(X_{ci+c-2})$
	$ci+c-1$	X_{ci+c-1}	K_{ci+c-1}	Y_{ci+c-1}	$V_{c-1} = Y_{ci+c-1} * \pi^B(X_{ci+c-1})$
$\rho^{-1}(\sigma^{c-1}(U_1))$	$(ci+c-1)$	X_{ci+c-1}	K_{ci+c-1}	Y_{ci+c-1}	$W_c = Y_{ci+c-1} * \pi^T(X_{ci+c-1})$
	(ci)	X_{ci}	K_{ci}	Y_{ci}	$V_c = Y_{ci} * \pi^B(X_{ci})$

Table 2. Starting with query ci, $(X_{ci}, K_{ci}, ?)_{fwd}$ or $(?, K_{ci}, Y_{ci})_{bwd}$, the adversary is given $c-1$ forward queries $ci+1, ci+2, \ldots, ci+c-1$ for free. In total, he is able to compute c results of F^{CYC} by using these c queries. The notations used in the table are given in the text.

We now upper bound $\Pr[\text{Coll}^{\text{CYC}}(\mathcal{Q})]$ by exhibiting predicates $\text{Win}_0(\mathcal{Q}), \ldots, \text{Win}_{q-1}(\mathcal{Q})$ such that $\text{Coll}^{\text{CYC}} \implies \text{Win}_0(\mathcal{Q}) \vee \ldots \vee \text{Win}_{q-1}(\mathcal{Q})$. Then, $\Pr[\text{Coll}^{\text{CYC}}(\mathcal{Q})] \leq \text{Win}_0(\mathcal{Q}) + \ldots + \text{Win}_{q-1}(\mathcal{Q})$. Since the adversary mounts q queries in total we informally say that $\text{Win}_i(\mathcal{Q})$, $0 \leq i \leq q-1$ holds if the adversary finds a collision after mounting the i-th query, $0 \leq i \leq q-1$, using at least two of the following queries $Q_{ci}, \ldots, Q_{ci+c-1}$ conditioned on the fact that the adversary has not been successful before. For simplicity, we assume again that the free queries are always given in 'ascending' order as given in Table 2.

Note that the following Definitions and Lemmas are generalizations of the Definitions and Lemmas given in Section 3.3.

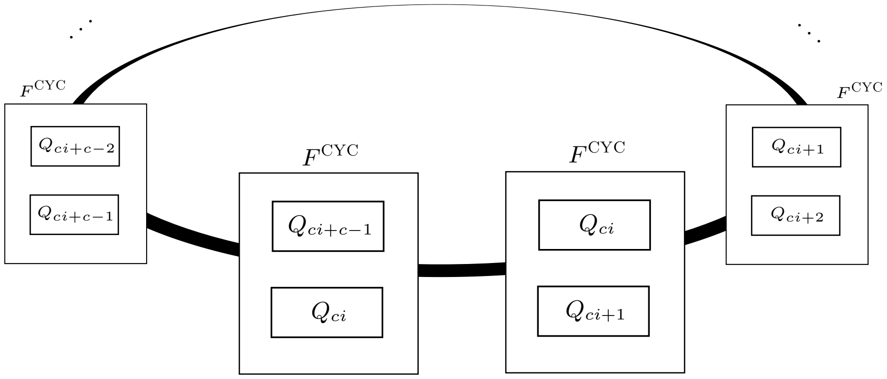


Figure 5. A Cycle: An adversary uses the c queries to compute the complete output of c compression functions F^{CYC}.

Definition 9. *We say that a pair of queries (a, b) is* successful *in $\mathcal{Q}_c$, if the query Q_a is used in the top row, Q_b in the bottom row in the computation of a compression function F^{CYC} and there exists a pair of queries $Q_j, Q_k \in \mathcal{Q}_c$ such that a collision of F^{CYC} can be computed:*

$$\pi^T(X_a) \oplus Y_a = \pi^T(X_j) \oplus Y_j \qquad \textit{and} \qquad \pi^B(X_b) \oplus Y_b = \pi^B(X_k) \oplus Y_k.$$

Definition 10. *Let $d = 0, \ldots, c-1$, $d' = d + 1 \mod c$, $\widetilde{d} = \max(d, d')$. We say $\text{CollFit}_i^d(\mathcal{Q})$ if (i) the pair of queries $(ci + d, ci + d')$ is successful in $\mathcal{Q}_{ci+\widetilde{d}}$ and (ii) the adversary had not been successful for $0 \le t \le d-1$: $\neg\text{CollFit}_i^t(\mathcal{Q})$.*

The predicates $\text{Win}_i(\mathcal{Q})$ are defined as follows:

Definition 11.

$$\text{Win}_i(\mathcal{Q}) = \neg\left(\bigvee_{0 \le j \le i-1} \text{Win}_j(\mathcal{Q})\right) \wedge \left(\text{CollFit}_i^1(\mathcal{Q}) \vee \ldots \vee \text{CollFit}_i^c(\mathcal{Q})\right)$$

We now show that our case analysis is complete.

Lemma 4. $\text{Coll}^{\text{CYC}}(\mathcal{Q}) \Longrightarrow \text{Win}_0(\mathcal{Q}) \vee \ldots \vee \text{Win}_{q-1}(\mathcal{Q})$.

This proof is omitted as it is essentially the same as the proof of Lemma 1, the only difference is that d is not chosen from the set $\{0, 1, \ldots, 5\}$ but from the set $\{0, 1, \ldots, c-1\}$. (□)

Since $\Pr[\text{Coll}^{\text{CYC}}(\mathcal{Q})] \le \sum_{j=0}^{q-1} \text{Win}_j(\mathcal{Q})$ it follows that

$$\Pr[\text{Coll}^{\text{CYC}}(\mathcal{Q})] \le \sum_{i=0}^{q-1} \sum_{d=0}^{c-1} \text{CollFit}_i^d(\mathcal{Q}). \tag{2}$$

We will now upper bound $\Pr[\text{CollFit}_i^d(\mathcal{Q})]$.

Lemma 5. *Let $0 \le i \le q-1$ and $0 \le d \le c-1$. Then*

$$\Pr[\text{CollFit}_i^d(\mathcal{Q})] \le \frac{ci}{(N - ci)^2}.$$

Proof. Let $d' = d + 1 \mod c$. The output of the compression function F^{CYC}, (W, V), is uniquely determined by the queries $Q_{ci+d} = (X_{ci+d}, K_{ci+d}, Y_{ci+d})$ and $Q_{6i+d'} = (X_{ci+d'}, K_{ci+d'}, Y_{ci+d'})$,

$$W = Y_{ci+d} * \pi^T(X_{ci+d}) \qquad \text{and} \qquad V = Y_{ci+d'} * \pi^B(X_{ci+d'}).$$

Using the same arguments as in the proof of Lemma 2 both W and V are randomly determined by the answer of the oracle. Not that the permutations π^T and π^B do not change these arguments.

To form a collision, two queries Q_j, Q_k are needed that can be chosen from at most $c(i + 1)$ queries in $\mathcal{Q}_{c(i+1)-1}$. The adversary can use them to compute the output of $< c(i + 1)$ compression functions F^{CYC}. Therefore,

$$\Pr[\text{CollFit}_i^d(\mathcal{Q})] \le \frac{c(i+1)}{(N - c(i+1))^2}.$$

(□)

Using (2) we get the following upper bound for any $q \geq 1$ and $N > cq$

$$\Pr[\textsc{Coll}^{\textsc{cyc}}(\mathcal{Q})] \leq \sum_{i=0}^{q-1}\sum_{d=0}^{c-1} \frac{c(i+1)}{(N-c(i+1))^2} \leq \sum_{i=1}^{q}\sum_{d=0}^{c-1} \frac{ci}{(N-ci)^2}$$
$$\leq \sum_{i=1}^{q} \frac{c^2 i}{(N-ci)^2} \leq \frac{c^2 \cdot q^2 \cdot \frac{1}{2}}{(N-cq)^2} \leq \frac{c^2}{2}\left(\frac{q}{N-cq}\right)^2.$$

This completes our proof of Theorem 4. ■

5 Building more Efficient and Secure DBL Compression Functions

The following list contains all efficient double block length compression functions known from literature that have provably birthday-type collision resistance. Except for Tandem-DM, they are all in the class of Cyclic. The threshold value 'α' gives the least amount of queries any adversary must ask in order to have more than a chance of 0.5 in finding a collision for the compression function assuming a plain-/ciphertext length of 128 bit of the block cipher.

Cycle length	Threshold α	Example(s)	Common Key	Parallel
2	$2^{124.55}$	Hirose FSE'06 [14]	yes	yes
		Add/1-DM, Section 5.1	yes	yes
3	$2^{125.42}$	Section 5.2	yes	yes
4	$2^{125.0}$	Add/2-DM, Section 5.1	yes	yes
6	$2^{124.42}$	Abreast-DM, Section 3.2	no	yes
2^k $(k \geq 2)$	2^{127-k}	Add/k-DM, Section 5.1	yes	yes
–	$2^{120.4}$	Tandem-DM, FSE'09 [10]	no	no

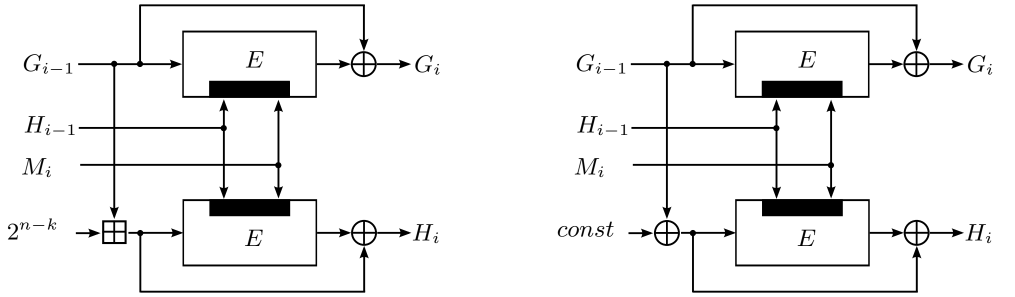
Table 3. List of all known efficient double block length compression functions. 'Common Key' indicates whether both block cipher calls use the same key for their encryption operations, 'Parallel' indicates whether both encryption operations are independent of each other and can therefore be computed in parallel.

5.1 Add/k-DM (cycle length 2^k)

Luckily, there does exist a very elegant method and efficient method to instantiate a compression function with cycle length $c = 2^k$ for any $k \geq 1$. This construction is very similar to Hirose's FSE'06 proposal. It is shown in Figure 6 and formally given in Definition 12.

Definition 12. *Let $F^{\textsc{Add/k}} : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^{2n}$ be a compression function such that $(G_i, H_i) = F^{\textsc{Add/k}}(G_{i-1}, H_{i-1}, M_i)$ where $G_i, H_i, M_i \in \{0,1\}^n$ and let $k \in \mathbb{N}$ such that $1 \leq k < n$. $F^{\textsc{Add/k}}$ is built upon a $(n, 2n)$-block cipher E as follows:*

$$G_i = E(G_{i-1}, H_{i-1}|M_i) \oplus G_{i-1}$$
$$H_i = E(G_{i-1} \boxplus 2^{n-k}, H_{i-1}|M_i) \oplus (G_{i-1} \boxplus 2^{n-k}),$$



ADD/K-DM, cycle-length 2^k

Hirose's FSE'06 proposal, cycle length 2, *const* $\neq 0$

Figure 6. Left: Cyclic Compression Function with cycle length 2^k, $k > 1$. Right: (for comparison) Hirose's FSE'06 proposal with a cycle length of 2.

where | represents concatenation. The symbol '$\boxplus$' denotes an addition modulo 2^n.

Lemma 6. *The compression function $F^{\text{ADD/K}}$ is in* CYCLIC *and has a cycle length of 2^k.*

Proof. To map with Definition 8 we let $\Omega = \{0,1\}^n$, $b = n$, $\pi^T = \pi^B = \text{ID}$, $\rho = \text{ID}$ and $\sigma : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^n \times \{0,1\}^{2n}$ is chosen as $\sigma(M,K) = (M \boxplus 2^{n-k}, K)$. The claim follows since

$$\underbrace{(\sigma \circ \ldots \circ \sigma)}_{2^k \text{ times}}(M,K) = (M \boxplus 2^k \cdot 2^{n-k}, K) = (M,K).$$

■

Therefore we can apply Theorem 3 for $k = 1$ or Theorem 4 if $k \geq 2$.

Corollary 2. *No adversary asking less than 2^{n-k-1} queries can have more than a chance of 0.5 in finding a collision for the compression function $F := F^{\text{ADD/K}}$ for any $1 < k < n$.*

Proof. This result can be obtained by using a simple calculation. As the cycle length c is equal to 2^k (Lemma 6), it follows using Theorem 4

$$\mathbf{Adv}_F^{\text{COLL}}(q) = \frac{2^{2k}}{2}\left(\frac{q}{2^{n-1}}\right)^2.$$

By applying $\mathbf{Adv}_F^{\text{COLL}}(q) = 0.5$ and solving after q one obtains $q(k) = \sqrt{2^{2n-2k-2}} = 2^{n-k-1}$. ■

Using $n = 128$, as for AES-256, we can derive without effort that no adversary asking less than 2^{122} queries can have more than a chance of 0.5 in finding a collision for the compression function $F^{\text{ADD}/5}$. The compression function $F^{\text{ADD}/5}$ has a cycle length of $2^5 = 32$.

5.2 Cube-DM (cycle length = 3)

The 'most optimal' result in terms of security – at least in the class CYCLIC – can be achieved by using a compression function that has a cycle length 3. The approach is slightly different compared to ADD/K-DM as neither additions modulo 2^n nor XOR can be used to create a permutation σ with $|\sigma| = 3$. The guiding idea to use a message space Ω such that $|\Omega|$ is evenly divisible by three. This construction is visualized in Figure 7 and given in Definition 14.

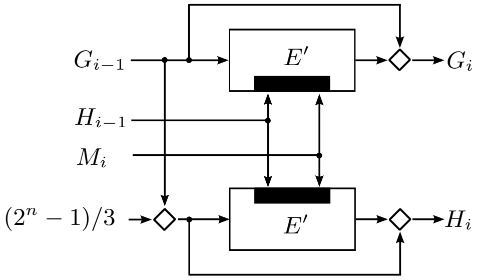


Figure 7. CUBE-DM, a compression function with cycle length $|\sigma| = 3$, the symbol $'\diamond'$ denotes an addition modulo $2^n - 1$.

Definition 13. *Let $E : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^n$ be a block cipher with n-bit plain-/ciphertext and $2n$-bit key. Let $\Omega = \{0,1\}^n - \{1^n\}$,* i.e. *$|\Omega| = 2^n - 1$. The block cipher $E' : \Omega \times (\Omega \times \{0,1\}^n) \to \Omega$, where $\Omega \times \{0,1\}^n$ is the key space is defined as*

$$E'_K(X) = \begin{cases} E_K(X), & \text{if } E_K(X) \neq 1^n, \\ E_K(E_K(X)), & \text{else.} \end{cases}$$

This definition of the block cipher E' ensures that that $E'_K(X) \in \Omega$ for any value of $X \in \Omega$: since E is a permutation, it follows that E' is a permutation. It is easy to see that, for n even, $|\Omega|$ is divisible by three since

$$|\Omega|\; mod\, 3 = 2^n - 1\; mod\, 3 = (2 \cdot 2)^{n'} - 1\; mod\, 3 = 0\; mod\, 3. \tag{3}$$

Definition 14. *Let $\Omega = \{0,1\}^n - \{1^n\}$, $N = |\Omega| = 2^n - 1$. Let $F^{\text{CUBE}} : \Omega^2 \times \{0,1\}^n \to \Omega^2$ be a compression function such that $(G_i, H_i) = F^{\text{CUBE}}(G_{i-1}, H_{i-1}, M_i)$ where $G_{i-1}, H_{i-1}, G_i, H_i \in \Omega$ and $M_i \in \{0,1\}^b$. Furthermore, let $const = (2^n - 1)/3$ and $'\diamond'$ be the addition modulo $2^n - 1$. Now F^{CUBE} is built upon a block cipher E' as in Definition 13:*

$$\begin{aligned} G_i &= E_{H_{i-1}|M_i}(G_{i-1}) \diamond G_{i-1} \\ H_i &= E_{H_{i-1}|M_i}(G_{i-1} \diamond const) \diamond (G_{i-1} \diamond const), \end{aligned}$$

where $'|'$ represents concatenation.

Lemma 7. *The compression function F^{CUBE} is in* CYCLIC *and has a cycle length of* 3.

Proof. To map with Definition 8, we choose $\rho = \text{ID}$, $\pi^T = \pi^B = \text{ID}$, $b = n$ and $\sigma : \Omega^2 \times \{0,1\}^n \to \Omega^2 \times \{0,1\}^n$ is chosen to be $\sigma(M, K) = (M \diamond (2^n - 1)/3, K)$. The claim follows using (3) and

$$(\sigma \circ \sigma \circ \sigma)(M, K) = (M \diamond 3 \cdot \frac{2^n - 1}{3}\; mod\, 2^n - 1, K) = (M, K).$$

■

The threshold value of $\alpha = 2^{125.42}$ as given in Table 3 follows with Theorem 4. Note that the operation '$\diamond$' is trivially efficient since a simple 'if' suffices to implement it. Also, the implementation of E' is not assumed to cost any measurable performance.

6 Discussion and Conclusion

In this paper, we have investigated the security of Abreast-DM, a long outstanding DBL compression function based on a $(n, 2n)$ block cipher that was presented at EUROCRYPT'92. In the ideal cipher model, we showed that this construction hast birthday type collision resistance: any adversary asking less than $2^{124.42}$ queries cannot find a collision with probability greater than 1/2. The proof technique was generalized to a class of double block length compression functions Cyclic and rigorous security bounds in terms of collision resistance and preimage resistance were given for this construction. The security of such constructions mainly depends on a parameter, the *cycle length*. Several new double block length compression functions were presented, some of them (Cube-DM and Add/4-DM) both have a higher security guarantee in terms of collision resistance than the best known DBL compression functions known in literature today.

Our work not only adds to the understanding of block cipher based compression functions but also introduces generic construction principles for such constructions. This is even more important as there are only two constructions known to have provably birthday type collision resistance (Hirose FSE'09 and Tandem-DM). Somewhat interestingly, one of the implicit results seems to be that, given the right construction, the security does *not* depend on whether the two block ciphers are fed in with different keys. This result alone renders constructions with different keys to be in the class of inefficient constructions as better ones always seem to be available.

Taking the long time Abreast-DM lacked a security proof, it is clear that there needs to be a lot of research done in the field of block cipher based hash functions. Still, there do not exist completely satisfying constructions and/or security proofs for, *e.g.*, MDC-2/4. More general, there has to be added a lot to our understanding, especially for constructions that are more efficient, *e.g.* have rate 1, or use other building blocks such as, *e.g.*, (n, n) block ciphers.

Acknowledgements

The authors wish to thank Thomas Shrimpton for pointing out the idea of Add/Cube-DM.

References

[1] ANSI. *ANSI X9.31:1998: Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)*. American National Standards Institute, pub-ANSI:adr, 1998.

[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] Antoon Bosselaers and Bart Preneel. Integrity Primitives for Secure Information Systems, Final Report of RACE Integrity Primitives Evaluation RIPE-RACE 1040. Springer, 1995, Lecture Notes in Computer Science, Volume 1007.

[4] C.Meyer and S.Matyas. Secure program load with manipulation detection code, 1988.

[5] D. Coppersmith, S. Pilpel, C. H. Meyer, S. M. Matyas, M. M. Hyden, J. Oseas, B. Brachtl, and M. Schilling. Data authentication using modification dectection codes based on a public one way encryption function. U.S. Patent No. 4,908,861, March 13, 1990.

[6] Ronald Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.

[7] Richard Drews Dean. *Formal aspects of mobile code security*. PhD thesis, Princeton, NJ, USA, 1999. Adviser-Andrew Appel.

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

[9] 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.

[10] Ewan Fleischmann, Michael Gorski, and Stefan Lucks. On the Security of Tandem-DM. In Robshaw [31], page ??

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

[12] Mitsuhiro Hattori, Shoichi Hirose, and Susumu Yoshida. Analysis of double block length hash functions. In Kenneth G. Paterson, editor, *IMA Int. Conf.*, volume 2898 of *Lecture Notes in Computer Science*, pages 290–302. Springer, 2003.

[13] Shoichi Hirose. Provably secure double-block-length hash functions in a black-box model. In Choonsik Park and Seongtaek Chee, editors, *ICISC*, volume 3506 of *Lecture Notes in Computer Science*, pages 330–342. Springer, 2004.

[14] Shoichi Hirose. Some Plausible Constructions of Double-Block-Length Hash Functions. In Matthew J. B. Robshaw, editor, *FSE*, volume 4047 of *Lecture Notes in Computer Science*, pages 210–225. Springer, 2006.

[15] Walter Hohl, Xuejia Lai, Thomas Meier, and Christian Waldvogel. Security of iterated hash functions based on block ciphers. In Stinson [37], pages 379–390.

[16] 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.

[17] John Kelsey and Bruce Schneier. Second Preimages on n-Bit Hash Functions for Much Less than 2^{n} Work. In Cramer [6], pages 474–490.

[18] 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.

[19] Lars R. Knudsen, Xuejia Lai, and Bart Preneel. Attacks on fast double block length hash functions. *J. Cryptology*, 11(1):59–72, 1998.

[20] Lars R. Knudsen and Frédéric Muller. Some attacks against a double length hash proposal. In Bimal K. Roy, editor, *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 462–473. Springer, 2005.

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

[22] M. Rabin. Digitalized Signatures, 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 Gilles Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.

[25] Nandi, Lee, Sakurai, and Lee. Security analysis of a 2/3-rate double length compression function in the black-box model. In *IWFSE: International Workshop on Fast Software Encryption, LNCS*, 2005.

[26] NIST National Institute of Standards and Technology. FIPS 180-1: Secure Hash Standard. April 1995. See http://csrc.nist.gov.

[27] NIST National Institute of Standards and Technology. FIPS 180-2: Secure Hash Standard. April 1995. See http://csrc.nist.gov.

[28] Bart Preneel, René Govaerts, and Joos Vandewalle. Hash functions based on block ciphers: A synthetic approach. In Stinson [37], pages 368–378.

[29] R. L. Rivest. *RFC 1321: The MD5 Message-Digest Algorithm*. Internet Activities Board, April 1992.

[30] 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.

[31] Matthew J. B. Robshaw, editor. *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 25-28, 2009, Revised Selected Papers*, volume 5??? of *Lecture Notes in Computer Science*. Springer, 2009.

[32] Phillip Rogaway and John P. Steinberger. Constructing cryptographic hash functions from fixed-key blockciphers. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 433–450. Springer, 2008.

[33] Phillip Rogaway and John P. Steinberger. Security/efficiency tradeoffs for permutation-based hashing. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 220–236. Springer, 2008.

[34] Satoh, Haga, and Kurosawa. Towards secure and fast hash functions. *TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems*, 1999.

[35] Martijn Stam. Blockcipher Based Hashing Revisited. In Robshaw [31], page ??

[36] John P. Steinberger. The collision intractability of mdc-2 in the ideal-cipher model. In Moni Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 34–51. Springer, 2007.

[37] Douglas R. Stinson, editor. *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*. Springer, 1994.

[38] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Cryptanalysis of the hash functions md4 and ripemd. In Cramer [6], pages 1–18.

[39] 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.

A Hirose's FSE'06 Proposal of a DBL Compression Function

At FSE'06, Hirose [14] proposed the DBL compression function F^{Hirose} (Definition 15 and Figure 8). He proved that when F^{Hirose} is employed in an iterated hash function H, then no adversary asking less than $2^{125.7}$ queries can have more than a chance of 0.5 in finding a collision for $n = 128$.

Definition 15. *Let $F^{Hirose} : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^{2n}$ be a compression function such that $(G_i, H_i) = F^{Hirose}(G_{i-1}, H_{i-1}, M_i)$ where $G_i, H_i, M_i \in \{0,1\}^n$. F^{Hirose} is built upon a $(n, 2n)$ block cipher E as follows:*

$$G_i = \mathrm{F}_T(G_{i-1}, H_{i-1}, M_i) = E(G_{i-1}, H_{i-1}|M_i) \oplus G_{i-1}$$
$$H_i = \mathrm{F}_B(G_{i-1}, H_{i-1}, M_i) = E(G_{i-1} \oplus C, H_{i-1}|M_i) \oplus G_{i-1} \oplus C,$$

where $'|'$ represents concatenation and $c \in \{0,1\}^n - \{0^n\}$ is a constant.

A visualization of this compression function is given in Figure 8.

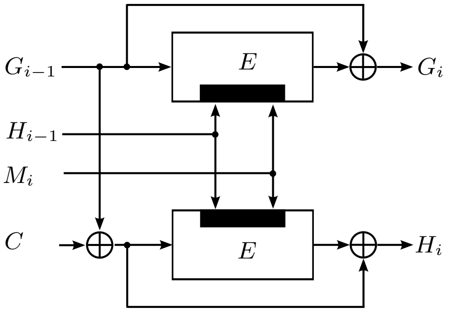


Figure 8. The compression function F^{Hirose}, E is an $(n, 2n)$ block cipher.

B A non-cyclic compression function: Tandem-DM

The TANDEM-DM compression function was proposed by Lai and Massey at EUROCRYPT'92 [21]. It uses two cascaded Davies-Meyer [2] schemes. The compression function is illustrated in Figure 9 and is formally given in Definition 16.

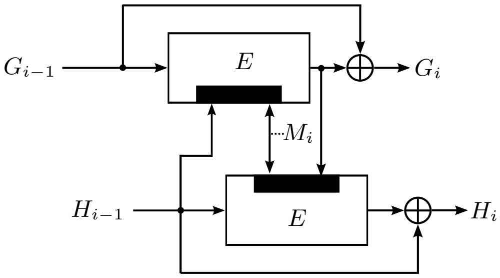


Figure 9. The compression function TANDEM-DM F^{TDM} where E is an $(n, 2n)$ block cipher, the black rectangle inside the cipher rectangle indicates the key input

Definition 16. *Let* $F^{TDM} : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^{2n}$ *be a compression function such that* $(G_i, H_i) = F^{TDM}(G_{i-1}, H_{i-1}, M_i)$ *where* $G_i, H_i, M_i \in \{0,1\}^n$. F^{TDM} *is built upon an* $(n, 2n)$ *block cipher* E *as follows:*

$$\begin{aligned} W_i &= E(G_{i-1}, H_{i-1}|M_i) \\ G_i &= \mathrm{F}_T(G_{i-1}, H_{i-1}, M_i) = W_i \oplus G_{i-1} \\ H_i &= \mathrm{F}_B(G_{i-1}, H_{i-1}, M_i) = E(H_{i-1}, M_i|W_i) \oplus H_{i-1}. \end{aligned}$$

C Collision Resistance – Proof of Theorem 3

Due to the special structure of the compression function in the case of $c = |\sigma| = 2$, the following definition is useful for the proof.

Definition 17. *A pair of distinct inputs* $(G_{i-1}, H_{i-1}, M_i), (G'_{i-1}, H'_{i-1}, M'_i)$ *to* F^{CYC} *is called a* matching *pair if* $(G'_{i-1}, H'_{i-1}, M'_i) = (\rho^{-1} \circ \sigma \circ \rho)(G_{i-1}, H_{i-1}, M_i)$. *Otherwise they are called a* non-matching *pair.*

Fix numbers n, q and an adversary $\mathcal{A}$ asking q backward and forward queries to its oracle E in total. Note that we will assume throughout this proof that the cycle length $c = |\sigma| = 2$. All queries to the oracle are saved in a query history $\mathcal{Q}$. Let $\text{COLL}^{\text{CYC}-2}$ be the event that the adversary is able to construct a collision of F^{CYC} in this case. We will examine the queries one at a time as they come in; the latest query made by the adversary, his i-th query, will always be given index $2i$, and is denoted as Q_{2i}. Say the query $Q_{2i} = (X_{2i}, K_{2i}, Y_{2i})$ is a forward or backward query mounted by the adversary and assume that Q_{2i} is used in the top row. As two queries are required for the computation of F^{CYC} we will give the adversary the bottom row query for free. This query is uniquely determined by its plaintext X_{2i+1} and key K_{2i+1} component as follows:

$$(X_{2i+1}, K_{2i+1}) = \mathcal{S}(\sigma(\mathcal{S}^{-1}(X_{2i}, K_{2i})))$$

and the adversary is given the ciphertext $Y_{2i+1} = E_{K_{2i+1}}(X_{2i+1})$. If the adversary uses the query Q_{2i} in the bottom row, we give him the top row query for free:

$$(X_{2i+1}, K_{2i+1}) = \mathcal{S}(\sigma^{-1}(\mathcal{S}^{-1}(X_{2i}, K_{2i})))$$

and the adversary is given the ciphertext $Y_{2i+1} = E_{K_{2i+1}}(X_{2i+1})$ in this case. Since $\sigma^2 = \text{ID}$ it follows that $\sigma = \sigma^{-1}$ it follows that in either case, the adversary is given the *same* free query, *i.e.* the input to the other query is always uniquely determined using one and the same computation.

Now assume for the simplicity of the following argument that the query Q_{2i} is used in the top row and Q_{2i+1} in the bottom row. As $G_i = Y_{2i} \oplus \pi^T(X_{2i})$ depends both on the plaintext and the ciphertext of E and one of them is fixed by query and the other is determined randomly by the oracle it follows that G_i is randomly determined by that answer. Using the same argument, $H_i = Y_{2i+1} \oplus \pi^T(X_{2i+1})$ is also randomly determined by the other answer.

For any $2 \le i \le q$ let C_i be the event that a colliding pair of non-matching inputs are found for F^{CYC} with the i-th pair of queries. Namely, it is the event that for some $i' < i$

$$F^{\text{CYC}}(\rho^{-1}(X_{2i}, K_{2i})) \in \{F^{\text{CYC}}(\rho^{-1}(X_{2i'}, K_{2i'})),\ F^{\text{CYC}}(\rho^{-1}(X_{2i'+1}, K_{2i'+1}))\}$$

or

$$F^{\text{CYC}}(\rho^{-1}(X_{2i+1}, K_{2i+1})) \in \{F^{\text{CYC}}(\rho^{-1}(X_{2i'}, K_{2i'})),\ F^{\text{CYC}}(\rho^{-1}(X_{2i'+1}, K_{2i'+1}))\}$$

This condition is equivalent to

$$(Y_{2i} * \pi^T(X_{2i}), Y_{2i+1} * \pi^B(X_{2i+1})) = (Y_{2i'} * \pi^T(X_{2i'}), Y_{2i+1} * \pi^B(X_{2i'+1})) \quad \text{or} \tag{4}$$
$$(Y_{2i} * \pi^T(X_{2i}), Y_{2i+1} * \pi^B(X_{2i+1})) = (Y_{2i'+1} * \pi^T(X_{2i'+1}), Y_{2i} * \pi^B(X_{2i'})) \quad \text{or} \tag{5}$$
$$(Y_{2i+1} * \pi^T(X_{2i+1}), Y_{2i} * \pi^B(X_{2i})) = (Y_{2i'} * \pi^T(X_{2i'}), Y_{2i+1} * \pi^B(X_{2i'+1})) \quad \text{or} \tag{6}$$
$$(Y_{2i+1} * \pi^T(X_{2i+1}), Y_{2i} * \pi^B(X_{2i})) = (Y_{2i'+1} * \pi^T(X_{2i'+1}), Y_{2i} * \pi^B(X_{2i'})). \tag{7}$$

Note that (4) is equal to (7) and (5) is equal to (6) if $\pi^T = \pi^B$. In this case, it follows that for $2q < N$

$$\Pr[\mathrm{C}_i] \le \frac{2(i-1)}{(N-(2i-2))(N-(2i-1))} \le \frac{2q}{(N-2q)^2}. \tag{8}$$

Assuming $\pi^T \neq \pi^B$ we obtain

$$\Pr[\mathrm{C}_i] \le \frac{4(i-1)}{(N-(2i-2))(N-(2i-1))} \le \frac{4q}{(N-2q)^2}. \tag{9}$$

For unifying the treatment of these two cases, we set $a = 1$ if $\pi^T = \pi^B$ and $a = 2$ otherwise. Let C be the event that a colliding pair of non-matching inputs are found for F^{CYC} with q (pairs) of queries. Then,

$$\Pr[\mathrm{C}] \le \sum_{i=2}^{q} \Pr[\mathrm{C}_j] \le \sum_{i=2}^{q} \frac{2q \cdot a}{(N-2q)^2} \le \frac{2aq^2}{(N-2q)^2}.$$

Now, let $\hat{\mathrm{C}}_i$ be the event that a colliding pair of matching inputs is found for F^{CYC}. It follows, that

$$\Pr[\hat{\mathrm{C}}_i] \leq \frac{2}{N-2q}.$$

Let $\hat{\mathrm{C}}$ be the event that a colliding pair of matching inputs are found for F^{CYC} with q (pairs) of queries. Then,

$$\Pr[\hat{\mathrm{C}}] \leq \sum_{i=2}^{q} \Pr[\hat{\mathrm{C}}_i] \leq \frac{2q}{N-2q}.$$

Since $\mathbf{Adv}_F^{\mathrm{COLL}}(q) = \Pr[\mathrm{C} \vee \hat{\mathrm{C}}] \leq \Pr[\mathrm{C}] + \Pr[\hat{\mathrm{C}}]$, the claim follows. ■

D Preimage resistance – Proof of Theorem 5

Although, the main focus is on collision resistance, we are also interested in the difficulty of inverting the compression function of F^{CYC}. Generally speaking, second-preimage resistance is a stronger security requirement than preimage resistance. A preimage may have some information of another preimage which produces the same output. However, in the ideal cipher model, for the compression function F^{CYC}, a second-preimage has no information useful to find another preimage. Thus, only preimage resistance is analyzed. Note, that there have be various results that discuss attacks on iterated hash functions in terms of pre- and second-preimage, *e.g.* long-message second-preimage attacks [7, 17], in such a way that the preimage-resistance level of a compression function cannot easily be transferred to an iterated hash function built on it.

The adversary's goal is to output a preimage (G, H, M) for a given ζ, where ζ is taken randomly from the output domain, such as $F^{\mathrm{CYC}}(G, H, M) = \zeta$. We will again dispense the adversary from having to output such a preimage. Instead, we will determine whether the adversary has been successful or not by examining its query history $\mathcal{Q}$. We say, that $\textsc{PreImg}(\mathcal{Q})$ holds if there is such a preimage and $\mathcal{Q}$ contains all the queries necessary to compute it.

Definition 18. ***(Inverting random points of a compression function)*** *Let F^{CYC} be as in Definition 6. Fix an adversary $\mathcal{A}$ that has access to oracles E, E^{-1}. The advantage of $\mathcal{A}$ of inverting $F := F^{\mathrm{CYC}}$ is the real number*

$$\mathbf{Adv}_F^{\mathrm{INV}}(\mathcal{A}) = \Pr[E \stackrel{R}{\leftarrow} \mathrm{BC}(n,k); \zeta \stackrel{R}{\leftarrow} \Omega^2; (G,H,M) \stackrel{R}{\leftarrow} \mathcal{A}^{E,E^{-1}}(\zeta) \mid F^{\mathrm{CYC}}(G,H,M) = \zeta].$$

Again, for $q \geq 1$, we write

$$\mathbf{Adv}_F^{\mathrm{INV}}(q) = \max_{\mathcal{A}}\{\mathbf{Adv}_F^{\mathrm{INV}}(\mathcal{A})\}$$

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

Note, that there has been a discussion on formalizations of preimage resistance. For details we refer to [2, Section 2, Appendix B].

The preimage resistance of the compression function F is given in the following Theorem.

Theorem 6. *Let $F := F^{\mathrm{CYC}}$ be as in Definition 6. For any $N' = N - q$ and $q > 1$*

$$\mathbf{Adv}_F^{\mathrm{INV}}(q) \leq 2q/(N')^2.$$

Proof. Fix $\zeta = (\zeta_1, \zeta_2) \in \Omega^2$ where $\sigma_1, \sigma_2 \in \Omega$ and an adversary $\mathcal{A}$ asking q queries to its oracles. We upper bound the probability that $\mathcal{A}$ finds a preimage for a given ζ by examining the oracle queries as they come in and upper bound the probability that the last query can be used to create a preimage, *i.e.* we upper bound $\Pr[\textsc{PreImg}(\mathcal{Q})]$. Let $\mathcal{Q}_i$ denote the first i queries made by the adversary. The term 'last query' means the latest query made by the adversary since we examine again the adversary's queries $(X_i, K_i)_{fwd}$ or $(K_i, Y_i)_{bwd}$ one at a time as they come in. The last query is always given index i.

Case 1: The last query (X_i, K_i, Y_i) is used in the top row. Either X_i or Y_i was randomly assigned by the oracle from a set of at least the size $N' := N - q$. The query is successful *in the top row* if $P_M^T(X_i) \oplus Y_i = \sigma_1$ and thus has a chance of success of $\leq 1/N'$. In $\mathcal{Q}_i$ there is at most one query Q_j, $j \leq i$ that can be used in the bottom row. This 'bottom' query is successful if such a query is in the query history $\mathcal{Q}$ and $P_M^B(X_j) \oplus Y_j = \sigma_2$ and therefore has a chance of success of $\leq 1/N'$. So the total chance of success is $\leq q/(N')^2$ as the adversary mounts at most q queries.

Case 2: The last query (X_i, K_i, Y_i) is used in the bottom row. The analysis is essentially the same as in case 1. The total chance of success is $\leq q/(N')^2$, too.

As any query can be either used in the top or the bottom row, the claim follows.