# The preimage security of double-block-length compression functions

Frederik Armknecht[1], Ewan Fleischmann[2], Matthias Krause[1], Jooyoung Lee[3]*, Martijn Stam[4], and John Steinberger[5]†

[1] Arbeitsgruppe Theoretische Informatik und Datensicherheit, University of Mannheim, Germany, {armknecht,krause}@uni-mannheim.de

[2] Chair of Media Security, Bauhaus-University Weimar, Germany, ewan.fleischmann@uni-weimar.de

[3] Faculty of Mathematics and Statistics, Sejong University, Seoul, Korea, jlee05@sejong.ac.kr

[4] Dept. of Computer Science, University of Bristol, United Kingdom, m.stam@alumnus.tue.nl

[5] Institute of Theoretical Computer Science, Tsinghua University, Beijing, China, jpsteinb@gmail.com

**Abstract.** We present new techniques for deriving preimage resistance bounds for block cipher based double-block-length, double-call hash functions. We give improved bounds on the preimage security of the three "classical" double-block-length, double-call, block cipher-based compression functions, these being Abreast-DM, Tandem-DM and Hirose's scheme. For Hirose's scheme, we show that an adversary must make at least $2^{2n-5}$ block cipher queries to achieve chance 0.5 of inverting a randomly chosen point in the range. For Abreast-DM and Tandem-DM we show that at least $2^{2n-10}$ queries are necessary. These bounds improve upon the previous best bounds of $\Omega(2^n)$ queries, and are optimal up to a constant factor since the compression functions in question have range of size $2^{2n}$.

**Keywords:** Hash Function, Preimage Resistance, Block Cipher, Beyond Birthday Bound, Foundations

## 1 Introduction

Almost as soon as the idea of turning a block cipher into a hash function appeared [9], it became evident that, for typical block ciphers and security expectations, the hash function needs to output a digest that is considerably larger

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0013560).

† Supported by the National Natural Science Foundation of China Grant 61033001, 61061130540, 61073174, by the National Basic Research Program of China Grant 2007CB807900, 2007CB807901 and by NSF grand CNS 0904380.

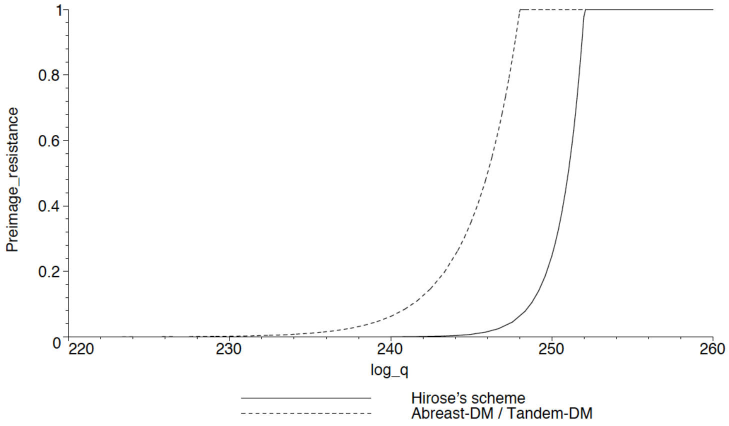

**Fig. 1.** Preimage bounds for the classical constructions.

than the block cipher's block size. Consequently, many proposals of double-block-length, or more generally multi-block-length, hash functions have appeared in the literature. In this article we focus on a subclass of double-block-length constructions, where a $3n$-bit to $2n$-bit compression function makes two calls to a block cipher of $2n$-bit key and $n$-bit block.

Recently, for all three well-known members of this class—those being Tandem-DM [5], Abreast-DM [5] and Hirose's construction [4]—collision resistance has been successfully resolved [2, 4, 6, 7]: for Abreast-DM and Hirose's scheme, $\Omega(2^n)$ queries to the underlying block cipher are needed to obtain a non-vanishing advantage in finding a collision. For Tandem-DM, $\Omega(2^{n-\log n})$ queries are needed, which is almost optimal ignoring log factors.

On the other hand, the corresponding situation for preimage resistance is far less satisfactory. Up to now, it has been an open problem to prove preimage resistance for values of $q$ higher than $2^n$ for either Abreast-DM, Tandem-DM or Hirose. This is not to say that no dedicated preimage security proofs have appeared in the literature. For instance, Lee, Stam and Steinberger [7] provide a preimage resistance bound for Tandem-DM that is a lot closer to $2^n$ than a straightforward implication [10] of their collision bound would give. However, a "natural barrier" occurs once $2^n$ queries are reached: namely, a block cipher "loses randomness" after being queried $\Omega(2^n)$ times on the same key (for example, when $2^n - 1$ queries have been made to a block cipher under a given key, the answer to the last query under that key is deterministic). Going beyond the $2^n$ barrier seemed to require either a very technical probabilistic analysis, or some brand new idea. In this paper, we show a new idea which delivers tight bounds in a quite pain-free and non-technical fashion.

Our contribution. In this paper, we prove that various compression functions that turn a block cipher of $2n$-bit key into a double-block-length hash function, have preimage resistance close to the optimal $2^{2n}$ in the ideal cipher model. Our analysis covers many practically relevant proposals, such as Abreast-DM, Hirose-DM and Tandem-DM. Bounds for the case $n = 128$ are depicted in Figure 1. At the heart of our result are so-called "super queries", a new technique to restrict the advantage of an adaptive preimage-finding adversary.

To build some intuition for our result, let us first consider the much easier problem of constructing a $3n$-bit to $2n$-bit compression function $H$ based on two $3n$-bit to $n$-bit smaller underlying primitives $f$ and $f'$. An obvious approach is simply to concatenate the outputs of $f$ and $f'$, that is let $H(B) = f(B)\|f'(B)$ for $B \in \{0,1\}^{3n}$. If $f$ and $f'$ are modeled as independent, ideal random functions, then it is not hard to see that $H$ behaves ideally as well. In particular, it is preimage resistant up to $2^{2n}$ queries (to $f$ and $f'$).

When switching to a block cipher-based scenario, it is natural to replace $f$ and $f'$ in the construction above by $E$, resp. $E'$, both run in Davies–Meyer mode. In other words, for block ciphers $E$ and $E'$ both with $2n$-bit keys and operating on $n$-bit blocks, define $H(A\|B) = (E_B(A) \oplus A)\|(E'_B(A) \oplus A)$ where $A \in \{0,1\}^n$ and $B \in \{0,1\}^{2n}$. While there is every reason to believe this construction maintains preimage resistance up to $2^{2n}$ queries, the standard proof technique against adaptive adversaries falls short significantly. Indeed, the usual argument goes that the $i$-th query an adversary makes to $E$ using key $K$ will return an answer uniform from a set of size at least $2^n - (i-1)$ and thus the probability of hitting a prespecified value is at most $1/(2^n - (i-1)) < 1/(2^n - q)$. Unfortunately, once $q$ approaches $2^n$, the denominator tends to zero (rendering the bound useless). As a result, one cannot hope to prove anything beyond $2^n$ queries using this method. This restriction holds even for a "typical" bound of type $q/(2^n - q)^2$.

When considering *non-adaptive* adversaries only, the situation is far less grim. Such adversaries need to commit to all queries in advance, which allows bounding the probability of each individual query hitting a prespecified value by $2^{-n}$. While obviously there are dependencies (in the answers), these can safely be ignored when a union bound is later used to combine the various individual queries. Since the $q$ offset has disappeared from the denominator, the typical bound $q/(2^n)^2$ *would* give the desired security.

Our solution, then, is to force an adaptive adversary to behave non-adaptively. As this might sound a bit cryptic, let us be more precise. Consider an adversary adaptively making queries to the block cipher, using the same key throughout. As soon as the number of queries *to this key* passes a certain threshold, we give the remaining queries to the block cipher using this very key *for free*. We will refer to this event as a *super query*. Since these free queries are all asked in one go, they can be dealt with non-adaptively, preempting the problems that occur (in standard proofs) due to adaptive queries. Nonetheless, for every super query we need to hand out a very large number of free queries, which can aid the adversary. Thus we need to limit the amount of super queries an adversary can make by setting the threshold that triggers a super query sufficiently high.

In fact, we set the threshold at exactly half[6] the total number of queries that can be made under a given key (i.e., it is set at $2^n/2$ queries). This effectively doubles the adversary's query budget, since for every query the adversary makes it can get another one later "for free" (if it keeps on making queries under the same key), but such a doubling of the number of queries does not lead to an unacceptable deterioration of the security bound.

With this new technique in hand, we can prove in Section 3 that the construction $H$ given above has indeed an asymptotically optimal preimage resistance bound. Afterwards, we revisit the proofs of preimage resistance of the three main double-block-length, double-call constructions: Hirose (Section 4), Abreast-DM (Section 5) and Tandem-DM (Section 6). An additional technical problem is that these compression functions each make two calls to the same block cipher, as opposed to using two calls to independent block ciphers. Ideally, to get a good bound, one would like to query the two calls necessary for a single compression function evaluation in conjunction (this would allow using the randomness of both calls simultaneously, potentially leading to a denominator $2^{2n}$ as desired for preimage resistance). For instance, in the context of collision resistance for Hirose-DM and Abreast-DM corresponding queries are grouped in cycles (of length 2 and 6, respectively) and all queries in a cycle are made simultaneously: if the adversary makes one query in a cycle, the remaining queries are handed out for free. Care has to be taken that these free queries and the free queries due to super queries do not reinforce each other to untenable levels.

For Hirose's scheme, there are no problems as the free queries introduced by a super query necessarily consist of full cycles only. The corresponding (upper) bound on the preimage finding advantage is $16q/2^{2n}$ which is as desired, up to a small factor. For Abreast-DM, however, the cyclic nature can no longer be exploited: any super query introduces many partial cycles, yet freely completing these might well trigger a new super query, etc.! Luckily, the original preimage proof for Tandem-DM [7] (which does not involve cycles) provides a way out of this conundrum. The downside however is that our preimage bound for Abreast-DM and Tandem-DM is slightly less tight than that for Hirose's scheme. Ignoring negligible terms, it grows roughly as $16\sqrt{q}/2^n$. Although this is faster than one might wish for (as can be seen in Figure 1), it does imply that $\Omega(2^{2n})$ queries are required to find a preimage with constant probability.

## 2 The model

A block cipher is a function $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ such that $E(K,\cdot)$ is a permutation of $\{0,1\}^n$ for each $K \in \{0,1\}^m$. We call $m$ the *key size* and $n$ the *block length* of the block cipher. It is customary to write $E_K(X)$ instead of $E(K,X)$ for $K \in \{0,1\}^m$, $X \in \{0,1\}^n$. The function $E_K^{-1}(\cdot)$ denotes the inverse of $E_K(\cdot)$ (as $E_K(\cdot)$ is a permutation). Henceforth, we will restrict to the case $m = 2n$ and we define $N = 2^n$.

[6] The "optimized" threshold turns out to be very near one half, but a bit less; we set the threshold at a half for simplicity in our proofs.

A compression function $H$ is block cipher-based if, in its execution, it has access to a block cipher. In this paper, we only discuss double-block-length, double-call constructions, meaning that $H$ is a function from $3n$-bits to $2n$-bits making two calls to some underlying block cipher $E$. (This definition will become more concrete in the next sections.)

As our preimage security notion for $H$, we adopt everywhere preimage resistance in the information theoretic setting [10]. In this preimage resistance experiment, a computationally unbounded adversary with oracle access to a uniformly sampled block cipher $E : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$ selects and announces a point $C \in \{0,1\}^{2n}$, before making queries to $E$. We allow the adversary to query both $E$ and $E^{-1}$. After $q$ queries to $E$, the *query history* of the attacker is the set of triples $\mathcal{Q} = \{(X_i, K_i, Y_i)\}_{i=1}^q$ such that $E_{K_i}(X_i) = Y_i$ and the attacker's $i$-th query is either $E_{K_i}(X_i)$ or $E_{K_i}^{-1}(Y_i)$ for $1 \leq i \leq q$. We say the attacker *succeeds* or *finds a preimage* if its query history $\mathcal{Q}$ contains the means of computing a preimage of $C$, in the sense that there exist values $B \in \{0,1\}^{3n}$, $K_1, K_2 \in \{0,1\}^{2n}$ and $X_1, X_2, Y_1, Y_2 \in \{0,1\}^n$ such that both $(X_1, K_1, Y_1)$ and $(X_2, K_2, Y_2)$ are in the query history $\mathcal{Q}$, $H(B) = C$ and the two queries used to evaluate $H(B)$ are precisely $E_{K_1}(X_1)$ and $E_{K_2}(X_2)$. In this case, we also say $\mathcal{Q}$ *contains a preimage* of $C$. We let $\mathsf{Preim}(\mathcal{Q})$ be the predicate that is true if and only if $\mathcal{Q}$ contains a preimage of $C$, where $C$ is an elided-but-understood parameter of the predicate. We define

$$\mathbf{Adv}_H^{\text{epre}}(q) = \max_A \Pr[\mathsf{Preim}(\mathcal{Q})]$$

where the maximum is taken over all adversaries making at most $q$ queries, and where the probability is taken over the randomness of $E$ as well as over the adversary's coins, if any.

For Tandem-DM, it turns out that the everywhere preimage resistance notion is slightly too strong, as there is one weak point (namely $0^{2n}$) in the range, for which finding preimages is a bit easier. A simple adaptation of the everywhere preimage resistance definition is to disallow the adversary to choose $C = 0^{2n}$ as the target point [7]; we denote the corresponding advantage as

$$\mathbf{Adv}_H^{\text{epre}\neq 0}(q) \ .$$

(We will still use the same predicate $\mathsf{Preim}(\mathcal{Q})$ though.)

A standard assumption made in ideal cipher proofs is that "the adversary never makes a query to which it already knows the answer". By this it is meant, for example, that one can assume the adversary never makes a query $E_K(X)$, obtaining an anwer $Y$, and then makes the query $E_K^{-1}(Y)$ (which will necessarily be answered by $X$). In the current context, where we consider adversaries making $2^n$ queries or more, this assumption should be more precisely restated as "the adversary never makes a query that will result in a triple $(X, K, Y)$ which is already present in the query history". (This latter assumption can be made without loss of generality using the fact that $E_K(\cdot)$ is a permutation.) Indeed, if an adversary has made $2^n - 1$ queries under a key $K$, the result of the last query under that key is predetermined, and thus the adversary “already knows” the answer to this query. However, one should not forbid the adversary from making this query, since the query may be necessary to complete a preimage.

Our security proofs also use the notion of “free” queries. Formally, these can be modelled as queries which the adversary is “forced” to query (under certain conditions), but for which the adversary is not charged: they do not count towards the maximum of $q$ queries which the adversary is allowed. However, these queries become part of the adversary’s query history, just like other queries. In particular, the adversary is not allowed, later, to remake these queries “on its own” (due to the previously discussed assumption that the adversary never makes a query which it already owns). Observe that “free” queries are a common tool for analyzing the security of hash functions, e.g., see [2, 3, 6].

## 3 An example case

Before we apply the new technique of super queries to the analysis of three well-known constructions that compress $3n$ bits to $2n$ bits and that each call the same block cipher twice, we demonstrate our technique on the following simplest possible example. We consider the construction $H_1$, compressing $3n-1$ bits to $2n$ bits that makes two block cipher calls. Given a block cipher $E$ of key length $m = 2n$ and block length $n$, an input block $X \in \{0,1\}^n$ and a key prefix $K \in \{0,1\}^{2n-1}$ we define

$$\mathrm{H}_1(K, X) = (E_{K\|0}(X) \oplus X, E_{K\|1}(X) \oplus X)$$

where $\|$ denotes concatenation. If we consider the ideal cipher model, the two block cipher calls are independent. $H_1$ can be seen as a simple special case of a scenario where two different block ciphers are called and which is closely connected with the more general framework introduced by Özen and Stam [8, 11] (with slightly different notation though).

**Theorem 1.** *Let* $\mathrm{H}_1 : \{0,1\}^{3n-1} \rightarrow \{0,1\}^{2n}$ *be the block cipher-based compression function defined as above. Then*

$$\mathbf{Adv}^{\mathrm{epre}}_{\mathrm{H}_1}(q) \leq 8q/N^2.$$

*In particular, to achieve an advantage of* $1/2$ *the adversary has to make at least* $2^{2n-4}$ *queries.*

*Proof.* Let $U\|V \in \{0,1\}^{2n}$ be the point to invert (chosen by the adversary before it makes any queries to $E$). We upper bound the probability that, in $q$ queries, the adversary finds a point $A \in \{0,1\}^n$ and a key prefix $K \in \{0,1\}^{2n-1}$ such that $\mathrm{H}_1(K\|A) = U\|V$. On top of the $q$ queries the adversary wants to make, we give it several queries for free, to ensure that the elements $(X, K\|0, Y)$ and $(X, K\|1, Y')$ are always added to the query history as a pair. We call such a pair an “adjacent query pair” with respect to the key prefix $K \in \{0,1\}^{2n-1}$. The involved free queries are as follows.

**Normal forward query.** If the adversary queries $E_{K\|0}(X)$ (resp. $E_{K\|1}(X)$) for some key prefix $K \in \{0,1\}^{2n-1}$ and $X \in \{0,1\}^n$, we also give it for free $E_{K\|1}(X)$ (resp. $E_{K\|0}(X)$).

**Normal inverse query.** If the adversary queries $E^{-1}_{K\|0}(Y)$ (resp. $E_{K\|1}(Y')$) for some key prefix $K \in \{0,1\}^{2n-1}$ and receives answer $X$, we also give it for free $E_{K\|1}(X)$ (resp. $E_{K\|0}(X)$).

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

**Super query.** When the query history contains $N/2$ adjacent query pairs all using the same key prefix $K \in \{0,1\}^{2n-1}$, all the remaining queries of the form $E_{K||0}(\cdot)$ and $E_{K||1}(\cdot)$ are given for free.

We say that an adjacent query pair $(X, K\|0, Y)$, $(X, K\|1, Y')$ is "winning", or "successful", if $X \oplus Y = U$ and $X \oplus Y' = V$. Thus the adversary obtains a preimage of $U\|V$ precisely if it obtains a winning adjacent query pair. This can occur in one of two ways: either the winning query pair is part of a super query, or not. We let $\mathsf{SuperQueryWin}(\mathcal{Q})$ denote the event that the adversary obtains a winning query pair that is part of a super query, and $\mathsf{NormalQueryWin}(\mathcal{Q})$ the event that the adversary obtains a winning query pair of normal queries. It thus suffices to upper bound

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

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

We first upper bound $\Pr[\mathsf{NormalQueryWin}(\mathcal{Q})]$. Note that when the adversary makes, say, a forward query $E_{K\|0}(X)$, at most $N/2-1$ queries have been previously answered to the key $K\|0$ and at most $N/2-1$ queries have been previously answered to the key $K\|1$, since otherwise a super query for the key prefix $K$ would have occurred. Thus the values $Y = E_{K\|0}(X)$ and $Y' = E_{K\|1}(X)$ come uniformly and independently at random from a set of size at least $N/2+1 \geq N/2$, and there is chance at most $(1/(N/2))^2 = 4/N^2$ that we obtain a winning pair of adjacent queries. The same is true if the adversary makes a forward query $E_{K\|1}(X)$, or an inverse query $E^{-1}_{K\|0}(Y)$, or an inverse query $E^{-1}_{K\|1}(Y')$. Since the adversary makes $q$ queries in total, we therefore have

$$\Pr[\mathsf{NormalQueryWin}(\mathcal{Q})] \leq 4q/N^2. \tag{1}$$

We now bound $\Pr[\mathsf{SuperQueryWin}(\mathcal{Q})]$. Say a super query is about to occur on key prefix $K \in \{0,1\}^{2n-1}$, meaning that the value of $E_{K\|0}(\cdot)$ and $E_{K\|1}(\cdot)$ is already known on exactly $N/2$ points. Let us denote this set of points by $\mathcal{X}$, and let $\mathcal{Y} = E_{K\|0}(\mathcal{X})$ and $\mathcal{Y}' = E_{K\|1}(\mathcal{X})$. Further let $\mathcal{A} = \{0,1\}^n \backslash \mathcal{X}$, $\mathcal{B} = \{0,1\}^n \backslash \mathcal{Y}$, and $\mathcal{B}' = \{0,1\}^n \setminus \mathcal{Y}'$. Note that $|\mathcal{X}| = |\mathcal{Y}| = |\mathcal{Y}'| = |\mathcal{A}| = |\mathcal{B}| = |\mathcal{B}'| = N/2$.

Now let a point $A \in \mathcal{A}$ in the domain of the super query be arbitrarily fixed, and let us estimate the probability that point $A$ induces a winning pair under $E$. If $A \oplus U \in \mathcal{Y}$ or if $A \oplus V \in \mathcal{Y}'$, this probability is zero. Consequently, let us suppose that $A \oplus U \in \mathcal{B}$ and $A \oplus V \in \mathcal{B}'$.

The probability (taken w.r.t. $E$) that $E_{K\|0}(A) = A \oplus U$ and $E_{K\|1}(A) = A \oplus V$ equals $\left(\frac{(N/2-1)!}{(N/2)!}\right)^2 = \left(\frac{1}{N/2}\right)^2$. Thus, by union bounding over $A$, we find that the probability of the super query producing a winning pair of adjacent queries is at most $N/2 \cdot \left(\frac{1}{N/2}\right)^2 = \frac{1}{N/2}$. We now observe that at most $q/(N/2)$ super queries can ever occur, since each super query requires a "setup" cost of $N/2$ queries. Thus

$$\Pr[\mathsf{SuperQueryWin}(\mathcal{Q})] \leq 4q/N^2. \tag{2}$$

Summing (1) and (2) completes the proof. □

## 4 Preimage security results for Hirose's scheme

Hirose [4] introduced his $3n$-bit to $2n$-bit compression function making two calls to a block cipher of $2n$-bit key over 10 years after Abreast-DM and Tandem-DM (see the next Sections). Hirose's construction (Figure 2) is simpler than either of its predecessors and it uses a single keying schedule for the top and bottom block ciphers. Moreover, Hirose himself already proved birthday-type collision resistance for his construction in the ideal cipher model, thereby predating similar collision resistance analyses for Abreast-DM and Tandem-DM. Previously, Lee and Kwon [6] have shown that $\mathbf{Adv}^{\mathrm{epre}}_{\mathrm{Hir}}(q) \leq 2q/(N-2q)^2$, which becomes void once $q > N/2$. We improve upon this bound considerably.

**Theorem 2.** *Let* $\mathrm{Hir} : \{0,1\}^{3n} \to \{0,1\}^{2n}$ *be the block cipher-based compression function depicted in Figure 2. Then*

$$\mathbf{Adv}^{\mathrm{epre}}_{\mathrm{Hir}}(q) \leq 8q/N^2 + 8q/N(N-2).$$

*In particular,* $\mathbf{Adv}^{\mathrm{epre}}_{\mathrm{Hir}}(q)$ *is upper bounded by approximately* $16q/N^2$.

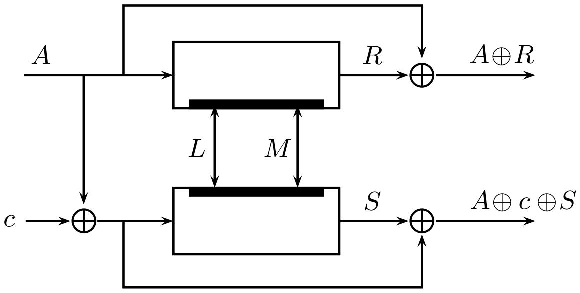

**Fig. 2.** Hirose's compression function. All wires carry $n$-bit values. The top and bottom block ciphers, which are the same block cipher, have $2n$-bit key and $n$-bit input/output. The wires $A, L, M$ are the inputs to the compression function. The bottom left-hand wire is not an input; it carries an arbitrary nonzero constant $c$.

*Proof.* Let $U\|V \in \{0,1\}^{2n}$ be the point to invert (chosen by the adversary before it makes any queries to $E$). We upper bound the probability that, in $q$ queries, the adversary finds a point $A\|L\|M \in \{0,1\}^{3n}$ such that $\mathrm{Hir}(A\|L\|M) = U\|V$.

When the adversary makes a *forward query* $E_{L\|M}(A)$ we give it for free, also, the answer to the query $E_{L\|M}(A \oplus c)$. Moreover when the adversary makes a *backward query* $E^{-1}_{L\|M}(R)$, resulting in an answer $A = E^{-1}_{L\|M}(R)$, we give it for free the answer to the forward query $E_{L\|M}(A \oplus c)$. Also, we assume that the adversary never makes a query to which it knows the answer (in the sense discussed in Section 2). Thus the elements of the adversary's query history $\mathcal{Q}$ can be paired into adjacent pairs of the form $(A, L\|M, R), (A \oplus c, L\|M, S)$. We call such a pair an "adjacent query pair". Furthermore, we define *super queries* analogously to the definition used in the proof of Theorem 1. More precisely, as soon as the (current) query history contains exactly $N/2$ queries with the same key, *all* remaining queries under this key are given for free to the adversary. (A minor difference with Theorem 1 is that it only takes $N/4$ queries to trigger a super query under a given key, instead of $N/2$.)

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

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

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

We first upper bound $\Pr[\mathsf{NormalQueryWin}(\mathcal{Q})]$. Note that when the adversary makes, say, a forward query $E_{L\|M}(A)$, at most $N/2 - 2$ queries (counting free queries) have been previously answered with the key $L\|M$, since otherwise a super query for the key $L\|M$ would have occured. Thus the value $R = E_{L\|M}(A)$ comes uniformly at random from a set of size at least $N/2+2 \geq N/2$, and there is chance at most $2/(N/2) = 4/N$ that either $A \oplus R = U$ or $A \oplus R = V$ (this is also true if $U = V$). If, say, $A \oplus R = U$, there is further chance at most $1/(N/2) = 2/N$ that the free query $E_{L\|M}(A \oplus c)$ returns $A \oplus c \oplus V$, since the answer to the free query comes uniformly at random from a set of size at least $N/2+1 \leq N/2$. Other cases (e.g. when $A \oplus R = V$, and when the adversary makes a backward query $E^{-1}_{L\|M}(R)$) are similarly analyzed, showing that the adversary's chance of triggering the event $\mathsf{NormalQueryWin}(\mathcal{Q})$ at any given query is at most $(4/N)(2/N) = 8/N^2$. Since the adversary makes $q$ queries total, we have

$$\Pr[\mathsf{NormalQueryWin}(\mathcal{Q})] \leq 8q/N^2. \tag{3}$$

We now bound $\Pr[\mathsf{SuperQueryWin}(\mathcal{Q})]$. Say a super query is about to occur on key $L\|M$, meaning that the value of $E_{L\|M}(\cdot)$ is already known on exactly $N/2$ points paired into $N/4$ query pairs. Let $A, A \oplus c$ be in the domain of the super query. (We say that a point $B \in \{0,1\}^n$ is "in the domain of the super query" if $E_{L\|M}(B)$ is not yet known, and will be queried as part of the super query; note that a point $A \in \{0,1\}^n$ is in the domain of the super query if and only if $A \oplus c$ is in the domain of the super query.) Then the probability that $E_{L\|M}(A) = U$ is either 0 if $U$ is not in the range of the super query (meaning there is a normal query $E_{L\|M}(B) = U$ already present in the query history when the super query is made), or else is exactly $2/N$, since the value of $E_{L\|M}(A)$ returned by the super query is uniform at random in a set of size $N/2$. Thus, by a similar argument on $V$, the probability that $E_{L\|M}(A) \in \{U, V\}$ is at most $4/N$. Conditioning on the event $E_{L\|M}(A) \in \{U, V\}$, the probability that $E_{L\|M}(A \oplus c) \in \{U, V\}$ is at most $1/(N/2-1)$, since $E_{L\|M}(A \oplus c)$ is sampled uniformly at random from a set of size $N/2-1$, once the value $E_{L\|M}(A)$ is known. Thus the probability that the super query returns values such that the adjacent query pair $(A, L\|M, \cdot)$, $(A \oplus c, L\|M, \cdot)$ is winning is at most $4/N(N/2-1)$. But $A, A \oplus c$ were two arbitrary paired domain points; taking a union bound over the $N/4$ such pairs in the domain of the super query, we find that the probability of the super query producing a winning pair of adjacent queries is at most

$$(N/4) \cdot (4/N(N/2-1)) = 1/(N/2-1).$$

We now observe that at most $q/(N/4)$ super queries can ever occur, since each super query requires a "setup" cost of $N/4$ queries. Thus

$$\Pr[\mathsf{SuperQueryWin}(\mathcal{Q})] \leq 4q/N(N/2-1). \tag{4}$$

Summing (3) and (4) completes the proof. □

## 5 Preimage security results for Abreast-DM

Abreast-DM, pictured in Figure 3, is one of the classical schemes for turning a $2n$-bit key block cipher into a $3n$-bit to $2n$-bit compression function. It was

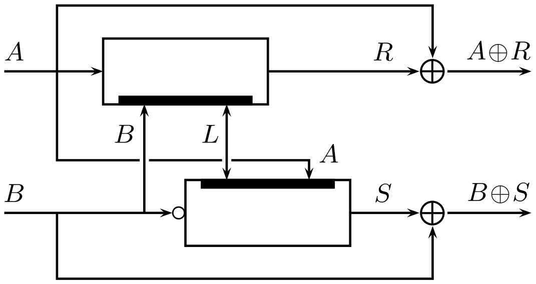

**Fig. 3.** The Abreast-DM compression function. The wires $A, B, L$ are the inputs to the compression function. The empty circle at the left side of the bottom block cipher denotes bit complementation.

proposed by Lai and Massey in the same paper as Tandem-DM [5]. The collision resistance of Abreast-DM was independently resolved by Fleischmann, Gorski and Lucks [2] and Lee and Kwon [6], who both showed birthday-type collision resistance for Abreast-DM. Previously, Hirose [3] had given a collision resistance analysis for a general class of compression functions that included Abreast-DM as a special case, but under the assumption that the top and bottom block ciphers of the diagram be distinct. This assumption considerably simplifies the analysis (see also the later generalization by Özen and Stam [8]).

Previously, Lee and Kwon [6] have shown that $\mathbf{Adv}^{\mathrm{epre}}_{\mathrm{Abr}}(q) \leq 6q/(2^n - 6q)^2$. Although our bound for Abreast-DM (Theorem 3) is not as tight as our bound for Hirose's scheme (Theorem 2), it is clear from Corollary 1 below that our result significantly improves this bound.

**Theorem 3.** *Let* Abr : $\{0,1\}^{3n} \to \{0,1\}^{2n}$ *be the block cipher-based compression function depicted in Figure 3. Let* $\alpha > 0$ *be an integer. Then*

$$\mathbf{Adv}^{\mathrm{epre}}_{\mathrm{Abr}}(q) \leq \frac{16\alpha}{N} + \frac{8q}{N^2(N-2)} + 2 \cdot \left(\frac{2eq}{\alpha N}\right)^{\alpha} + \frac{4q}{\alpha N}.$$

*Proof.* Let $U\|V$ be the point to invert, chosen by the adversary before any queries are made to $E$.

Unlike in the proof for Hirose's scheme, we do not give the adversary a free query after each query it makes. However, we still give the adversary "super queries" for free. More precisely, whenever the adversary has made $N/2$ queries under a given key $K\|L$, and after the $(N/2)$-th such query has been answered and placed in the query history, we give the remaining $N/2$ queries under the key $K\|L$ for free to the adversary, in any order. In this case, we say that a super query occurs; every query in the query history is either part of a super query, or not; in the latter case we call the query a "normal query". (Thus, in this theorem, normal queries are exactly the non-free queries.) Unlike in the proof of Theorem 2, there is no notion of an adjacent query pair. However, like in the proof of Theorem 2, we alert the reader to the fact that a "super query" consists of a set of $N/2$ queries, whereas a "normal query" is a single query.

We define an event $\mathsf{Lucky}(\mathcal{Q})$ on the query history; $\mathsf{Lucky}(\mathcal{Q})$ occurs if

$$|\{(X, K\|L, Y) \in \mathcal{Q} : X \oplus Y = U\}| > 2\alpha,$$

or if

$$|\{(X, K\|L, Y) \in \mathcal{Q} : X \oplus Y = V\}| > 2\alpha.$$

The adversary obtains a preimage of $U\|V$ precisely if it obtains queries of the form $(A, B\|L, R)$, $(\overline{B}, L\|A, S)$ such that $A \oplus R = U$ and $B \oplus S = V$, where $\overline{B}$ is bitwise complementation of $B$. It is easy to check that these two queries must be distinct, otherwise one obtains the contradiction $\overline{B} = A = L = B$. We call two such queries a "winning pair" of queries. Note, of course, that the queries in a winning pair need not be adjacent in the query history. We speak of the "first" and "second" query in a winning pair referring to the order in which they appear in the query history.

Let $\mathsf{WinNormal}(\mathcal{Q})$ be the event that the adversary obtains a winning pair in which the second query is a normal query. Let $\mathsf{WinSuper}_1(\mathcal{Q})$ be the event that the adversary obtains a winning pair in which the second query is part of a super query and the first is either normal or part of a super query, but is not part of the *same* super query as the second. Finally let $\mathsf{WinSuper}_2(\mathcal{Q})$ be the event that the adversary obtains a winning pair in which both queries of the pair are part of the same super query. It is then clear that if the adversary wins, one of the events

$$\mathsf{WinNormal}(\mathcal{Q}), \mathsf{WinSuper}_1(\mathcal{Q}) \text{ or } \mathsf{WinSuper}_2(\mathcal{Q})$$

occurs. In particular, thus, one of the four events

$$\mathsf{Lucky}(\mathcal{Q}), \mathsf{WinNormal}(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q}), \mathsf{WinSuper}_1(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q}),$$

$$\mathsf{WinSuper}_2(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})$$

must occur if the adversary wins. We upper bound the probability of each of these four events and sum the upper bounds in order to obtain an upper bound on the adversary's advantage.

We start by upper bounding $\Pr[\mathsf{Lucky}(\mathcal{Q})]$. For this we introduce two new events. Let $\mathcal{Q}_{\mathrm{n}}$ be the restriction of $\mathcal{Q}$ to normal queries, and let $\mathcal{Q}_{\mathrm{s}}$ be the restriction of $\mathcal{Q}$ to queries that are part of super queries. Let $\mathsf{Lucky}_{\mathrm{n}}(\mathcal{Q})$ be the event that either

$$|\{(X, K\|L, Y) \in \mathcal{Q}_{\mathrm{n}} : X \oplus Y = U\}| > \alpha,$$

or

$$|\{(X, K\|L, Y) \in \mathcal{Q}_{\mathrm{n}} : X \oplus Y = V\}| > \alpha.$$

The event $\mathsf{Lucky}_{\mathrm{s}}(\mathcal{Q})$ is likewise defined with respect to $\mathcal{Q}_{\mathrm{s}}$. Obviously, $\mathsf{Lucky}(\mathcal{Q}) \implies \mathsf{Lucky}_{\mathrm{n}}(\mathcal{Q}) \vee \mathsf{Lucky}_{\mathrm{s}}(\mathcal{Q})$, so it suffices to upper bound $\mathsf{Lucky}_{\mathrm{n}}(\mathcal{Q})$ and $\mathsf{Lucky}_{\mathrm{s}}(\mathcal{Q})$ and to sum these upper bounds.

Since every answer to a normal query, forward or backward, comes at random from a set of size at least $N/2$, and since at most $q$ normal queries are made, we have that

$$\Pr[\mathsf{Lucky}_{\mathrm{n}}(\mathcal{Q})] \leq 2 \cdot \binom{q}{\alpha} \left(\frac{2}{N}\right)^{\alpha} \leq 2 \cdot \left(\frac{2eq}{\alpha N}\right)^{\alpha}.$$

To upper bound $\Pr[\mathsf{Lucky}_{\mathrm{s}}(\mathcal{Q})]$, note that when a super query is made on key $K \| L$, the expected number of points $X \in \{0,1\}^n$ in the domain of the super query such that $X \oplus E_{K\|L}(X) = U$ is at most $(N/2) \cdot (2/N) = 1$, since for each individual such point the probability that $X \oplus E_{K\|L}(X) = U$ is either 0 (if $X \oplus U$ is not in the range of the super query) or $2/N$. Moreover there occur at most $q/(N/2) = 2q/N$ super queries, since it costs $N/2$ queries to setup a super query for a given key. Thus, the expectation of the random variable

$$|\{(X, K\|L, Y) \in \mathcal{Q}_{\mathrm{s}} : X \oplus Y = U\}|,$$

taken over the coin tosses of the adversary and the randomness of $E$, is at most $2q/N \cdot 1 = 2q/N$. It then follows by Markov's inequality that the probability that

$$|\{(X, K\|L, Y) \in \mathcal{Q}_{\mathrm{s}} : X \oplus Y = U\}| > \alpha$$

is at most $2q/\alpha N$. Then by a union bound and a symmetric argument (for $X \oplus Y = V$) , we obtain that $\Pr[\mathsf{Lucky}_{\mathrm{s}}(\mathcal{Q})] \leq 4q/\alpha N$. Summing the upper bounds for $\Pr[\mathsf{Lucky}_{\mathrm{n}}(\mathcal{Q})]$ and $\Pr[\mathsf{Lucky}_{\mathrm{s}}(\mathcal{Q})]$, we thus obtain that

$$\Pr[\mathsf{Lucky}(\mathcal{Q})] \leq 2 \cdot \left(\frac{2eq}{\alpha N}\right)^{\alpha} + \frac{4q}{\alpha N}. \tag{5}$$

We now upper bound $\Pr[\mathsf{WinNormal}(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})]$. For this we use a "wish list" argument similar to that of [7]. As the adversary makes queries, we maintain two sequences $\mathcal{W}_{\mathrm{T}}$ and $\mathcal{W}_{\mathrm{B}}$ called *wish lists*. These are initially empty. For each query $(X, K\|L, Y)$ added to the query history (whether normal or part of a super query) we update the wish lists as follows:

1. If $X \oplus Y = U$ then $(\overline{K}, L\|X, K \oplus V)$ is added to $\mathcal{W}_{\mathrm{B}}$.
2. If $X \oplus Y = V$ then $(L, \overline{X}\|K, L \oplus U)$ is added to $\mathcal{W}_{\mathrm{T}}$.

We emphasize that $\mathcal{W}_{\mathrm{B}}$ and $\mathcal{W}_{\mathrm{T}}$ are sequences, not sets. The following properties are easy to check: (i) a query never "adds itself" to a wish list (namely, the queries inserted into the wish lists—if any—as a result of query $(X, K\|L, Y)$ being added to the query history, are distinct from $(X, K\|L, Y)$ itself); (ii) the elements of $\mathcal{W}_{\mathrm{T}}$ are all distinct from one another, and the elements of $\mathcal{W}_{\mathrm{B}}$ are all distinct from one another—namely, the same triple is never added twice to a wish list; (iii) the adversary obtains a winning pair precisely if a query is ever added to its query history that is already a member of one of its wish lists before the updating of the wish lists for that query (by property (i), however, we could equally well say "after the updating of the wish lists for that query"). Moreover, as long as $\neg\mathsf{Lucky}(\mathcal{Q})$ holds, the wish lists never exceed length $2\alpha$.

Let $E_{K\|L}(X)$ be a query made to $E$ during the adversary's attack (either a normal query, or as part of a super query). If, at the moment when the query is being made, there is an element of the form $(X, K\|L, Y)$ in (at least) one of the wish lists for some $Y \in \{0,1\}^n$, then we say this wish list element is being "wished for" when the query $E_{K\|L}(X)$ is made. We similarly say the wish list element $(X, K\|L, Y)$ is being "wished for" if the query $E^{-1}_{K\|L}(Y)$ is made (note that in this case, the query $E^{-1}_{K\|L}(Y)$ is necessarily normal, since a super query is, by default, implemented by forward queries). We note, importantly, that any wish list element can only be wished for once, since $E_{K\|L}(\cdot)$ is a permutation.

Let $\mathsf{NormalWishGranted}_{\mathrm{T},i}$ be the event that a normal query $(X, K\|L, Y)$, when added to the query list, is equal to the $i$-th element of $\mathcal{W}_{\mathrm{T}}$ (presuming $\mathcal{W}_{\mathrm{T}}$ has length at least $i$ when the query is added). Likewise define $\mathsf{NormalWishGranted}_{\mathrm{B},i}$ with respect to the list $\mathcal{W}_{\mathrm{B}}$. Then by the above remarks

$$\begin{aligned}\mathsf{WinNormal}(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q}) \implies & \bigvee_{i=1}^{2\alpha} \mathsf{NormalWishGranted}_{\mathrm{T},i} \vee \\ & \bigvee_{i=1}^{2\alpha} \mathsf{NormalWishGranted}_{\mathrm{B},i}\end{aligned}$$

so by a union bound

$$\begin{aligned}\Pr[\mathsf{WinNormal}(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})] \le & \sum_{i=1}^{2\alpha} \Pr[\mathsf{NormalWishGranted}_{\mathrm{T},i}] + \\ & \sum_{i=1}^{2\alpha} \Pr[\mathsf{NormalWishGranted}_{\mathrm{B},i}].\end{aligned}$$

Because each wish list element can only be wished for once and because a normal query is answered at random uniformly from a set of size at least $N/2$, we have

$$\Pr[\mathsf{NormalWishGranted}_{\mathrm{T},i}] \le 2/N, \qquad \Pr[\mathsf{NormalWishGranted}_{\mathrm{B},i}] \le 2/N$$

and therefore

$$\Pr[\mathsf{WinNormal}(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})] \le 2 \cdot (4\alpha/N) = 8\alpha/N. \tag{6}$$

We now upper bound $\Pr[\mathsf{WinSuper}_1(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})]$. We keep the same definition of the wish lists $\mathcal{W}_{\mathrm{T}}$, $\mathcal{W}_{\mathrm{B}}$ as above. We let $\mathsf{SuperWishGranted}^1_{\mathrm{T},i}$ be the event that a query $(X, K\|L, Y)$ that is part of a super query is equal to the $i$-th element of $\mathcal{W}_{\mathrm{T}}$, where $\mathcal{W}_{\mathrm{T}}$ has length $\ge i$ before any of the super queries under key $K\|L$ have been made. The event $\mathsf{SuperWishGranted}^1_{\mathrm{B},i}$ is similarly defined. By the definition of $\mathsf{WinSuper}_1(\mathcal{Q})$ we have that

$$\begin{aligned}\Pr[\mathsf{WinSuper}_1(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})] \le & \sum_{i=1}^{2\alpha} \Pr[\mathsf{SuperWishGranted}^1_{\mathrm{T},i}] + \\ & \sum_{i=1}^{2\alpha} \Pr[\mathsf{SuperWishGranted}^1_{\mathrm{B},i}].\end{aligned}$$

Assume, for a given $i$, that the $i$-th element of $\mathcal{W}_{\mathrm{T}}$ (say) is $(X, K\|L, Y)$, and that a super query is about to be made for the key $K\|L$, and that $X$ is in the domain of the super query. Then the probability that $E_{K\|L}(X) = Y$ is at most $2/N$ (more precisely, it is exactly $2/N$ unless $Y$ is not in the super query's range, in which case it is 0). Thus, arguing similarly for the list $\mathcal{W}_{\mathrm{B}}$, we obtain that

$$\Pr[\mathsf{SuperWishGranted}^1_{\mathrm{T},i}] \leq 2/N, \qquad \Pr[\mathsf{SuperWishGranted}^1_{\mathrm{B},i}] \leq 2/N.$$

Therefore

$$\Pr[\mathsf{WinSuper}_1(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})] \leq 8\alpha/N. \tag{7}$$

We finally bound $\Pr[\mathsf{WinSuper}_2(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})]$. In fact we upper bound the value $\Pr[\mathsf{WinSuper}_2(\mathcal{Q})]$, and we do not use a wish list argument. Note the event $\mathsf{WinSuper}_2(\mathcal{Q})$ can only occur when a super query is made on a key of the form $L\|L$, and then occurs only if both $L$ and $\overline{L}$ are in the domain of the super query and if $E_{L\|L}(L) \oplus L = U$, $E_{L\|L}(\overline{L}) \oplus L = V$. It is easy to see that probability (when the super query is made) that these latter equalities hold is at most $(2/N) \cdot (1/(N/2 - 1))$. Since at most $q/(N/2)$ super queries are made, we therefore have

$$\Pr[\mathsf{WinSuper}_2(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})] \leq \Pr[\mathsf{WinSuper}_2(\mathcal{Q})] \leq 4q/N^2(N/2-1). \tag{8}$$

Finally, we obtain the theorem by summing (5), (6), (7) and (8). □

**Corollary 1.** *We have*

$$\mathbf{Adv}^{\mathrm{epre}}_{\mathrm{Abr}}(2^{2n-10}) \leq 1/2 + o(1)$$

*where the $o(1)$ term tends to 0 as $n \to \infty$.*

*Proof.* By setting $\alpha = q^{1/2}/2$ (note that $\alpha$ is allowed to depend on $q$), the bound from Theorem 3 simplifies to

$$\frac{16q^{1/2}}{N} + \frac{8q}{N^2(N-2)} + 2 \cdot \left(\frac{4eq^{1/2}}{N}\right)^{q^{1/2}/2}$$

Suppose that $q = (cN)^2$ for some $0 < c < 1$, then this bound can be rewritten as

$$16c + \frac{8c^2}{N-2} + 2 \cdot (4ec)^{cN/2}.$$

For $4ec < 1$ this tends $16c$, so setting $c = 1/32$ gives us the claimed result. □

## 6 Preimage security results for Tandem-DM

The Tandem-DM compression function, proposed by Lai and Massey in 1992 [5], is a $3n$-bit to $2n$-bit compression function based on two applications of a block

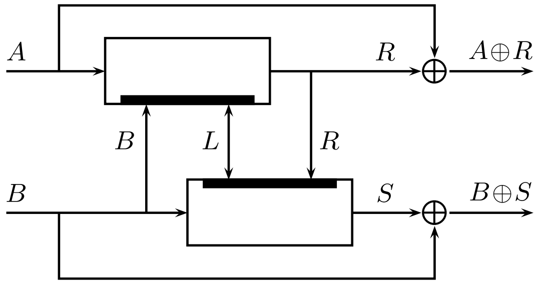

**Fig. 4.** The Tandem-DM compression function. The wires $A, B, L$ are the inputs to the compression function.

cipher of $2n$-bit key and $n$-bit word length (Figure 4). The first (flawed) proof of collision security for Tandem-DM (by Fleischmann, Gorski and Lucks [1]) did not appear until 2009. Later, Lee, Stam and Steinberger [7] gave a correct collision resistance analysis of Tandem-DM showing that indeed it has birthday-type collision security in the ideal cipher model (necessitating at least $2^{120.8}$ queries to break when the output length is $2n = 256$ bits). They also showed preimage resistance up to essentially $2^{128}$ queries (for $n = 128$), once $0^n \| 0^n$ is excluded as challenge digest. Our new bound is identical to the bound we gave for Abreast-DM, so in particular $2^{2n-10}$ queries are needed to obtain a preimage with probability $\sim 0.5$ (Corollary 2).

**Theorem 4.** *Let* Tan : $\{0,1\}^{3n} \to \{0,1\}^{2n}$ *be the block cipher-based compression function depicted in Figure 4. Let* $\alpha > 0$ *be an integer. Then*

$$\mathbf{Adv}^{\mathrm{epre}\neq 0}_{\mathrm{Tan}}(q) \leq \frac{16\alpha}{N} + \frac{8q}{N^2(N-2)} + 2 \cdot \left(\frac{2eq}{\alpha N}\right)^{\alpha} + \frac{4q}{\alpha N}.$$

*Proof.* Let $U \| V \neq 0^n \| 0^n$ be the point to invert, chosen by the adversary before making any queries to $E$.

We manage free queries exactly as for Abreast-DM; more precisely, when $N/2$ queries are made to $E$ under a given key, we give the remaining $N/2$ queries under that key for free to the adversary, and this constitutes a "super query". No other free queries are given.

In the case of Tandem-DM, the adversary obtains a preimage of $U \| V$ precisely if it obtains queries of the form $(A, B \| L, R)$, $(B, L \| R, S)$ such that $A \oplus R = U$, $B \oplus S = V$. It is easy to see these two queries must be distinct, otherwise we would have $A = B = L = R = S$ and therefore $U \| V = 0^n \| 0^n$. We call two queries as above a "winning pair" of queries, where the two elements of a winning pair need not be adjacent in the query history (and could be in any order). We speak again of the "first" and "second" query in a winning pair referring to the order in which they appear in the query history.

We define the events $\mathsf{Lucky}(\mathcal{Q})$, $\mathsf{WinNormal}(\mathcal{Q})$, $\mathsf{WinSuper}_1(\mathcal{Q})$ and $\mathsf{WinSuper}_2(\mathcal{Q})$ as in the proof of Theorem 3 (but with respect, of course, to the new definition of "winning pair"). If the adversary wins, one of the events

$$\mathsf{Lucky}(\mathcal{Q}),\ \mathsf{WinNormal}(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q}),\ \mathsf{WinSuper}_1(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q}),$$

$$\mathsf{WinSuper}_2(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})$$

must occur. We upper bound the probability of each of these events separately.

As in the case of Theorem 3, we have

$$\Pr[\mathsf{Lucky}(\mathcal{Q})] \leq 2 \cdot \left(\frac{2eq}{\alpha N}\right)^{\alpha} + \frac{4q}{\alpha N}. \tag{9}$$

To upper bound $\Pr[\mathsf{WinNormal}(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})]$, we again use wish lists. There are two wish lists, $\mathcal{W}_{\mathrm{T}}$ and $\mathcal{W}_{\mathrm{B}}$, which are initially empty and which are updated after each new query $(X, K\|L, Y)$ placed into the query history, according to the following rules:

1. If $X \oplus Y = U$ then $(K, L\|Y, K \oplus V)$ is added to $\mathcal{W}_{\mathrm{B}}$.
2. If $X \oplus Y = V$ then $(L \oplus U, X\|K, L)$ is added to $\mathcal{W}_{\mathrm{T}}$.

The same four properties from Theorem 3 are easy to check: (i) a query never "adds itself" to a wish list (this uses $U\|V \neq 0^n\|0^n$); (ii) the elements within each wish list are all distinct from one another; (iii) the adversary obtains a winning pair precisely if it obtains a query that is already in one of its wish lists (at the moment of insertion of that query into the query history). And by definition of $\mathsf{Lucky}(\mathcal{Q})$, the wish lists never exceed length $2\alpha$ as long $\neg\mathsf{Lucky}(\mathcal{Q})$ holds.

Let $\mathsf{NormalWishGranted}_{\mathrm{T},i}$, $\mathsf{NormalWishGranted}_{\mathrm{B},i}$ be defined as in (the proof of) Theorem 3. Then, using exactly the same analysis as in the proof of Theorem 3, we have that

$$\Pr[\mathsf{NormalWishGranted}_{\mathrm{T},i}] \leq 2/N, \qquad \Pr[\mathsf{NormalWishGranted}_{\mathrm{B},i}] \leq 2/N$$

and that

$$\Pr[\mathsf{WinNormal}(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})] \leq 8\alpha/N. \tag{10}$$

Then also arguing word for word as in the proof of Theorem 3, we find that

$$\Pr[\mathsf{WinSuper}_1(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})] \leq 8\alpha/N. \tag{11}$$

We finally bound $\Pr[\mathsf{WinSuper}_2(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})]$. Note the event $\mathsf{WinSuper}_2(\mathcal{Q})$ can only occur when a super query occurs for a key of the form $L\|L$, and when that super query results in the triples $(U \oplus L, L\|L, L)$, $(L, L\|L, L \oplus V)$ being added to the query history. The probability that $E_{L\|L}(U \oplus L) = L$ is at most $2/N$, and, conditioned on the event that $E_{L\|L}(U \oplus L) = L$, the probability that $E_{L\|L}(L) = L \oplus V$ is at most $1/(N/2 - 1)$. Since at most $2q/N$ super queries occur, we thus find that

$$\Pr[\mathsf{WinSuper}_2(\mathcal{Q}) \wedge \neg\mathsf{Lucky}(\mathcal{Q})] \leq \Pr[\mathsf{WinSuper}_2(\mathcal{Q})] \leq 4q/N^2(N/2 - 1). \tag{12}$$

The theorem follows by summing (9), (10), (11) and (12). □

As for Abreast-DM, we have the following corollary (with the same proof):

**Corollary 2.** *We have*

$$\mathbf{Adv}_{\mathrm{Tan}}^{\mathrm{epre}}(2^{2n-10}) \leq 1/2 + o(1)$$

*where the $o(1)$ term tends to 0 as $n \to \infty$.*

## 7 Conclusion

In this work, we developed and applied new techniques for determining lower bounds with respect to preimage resistance. As opposed to existing techniques, statements on the security beyond the birthday bound are possible. We applied successfully these techniques to the three popular double-block-length, double-call, block cipher-based compression functions, these being Abreast-DM, Tandem-DM and Hirose's scheme.

Although these techniques allow for proving asymptotically optimal bounds, these bounds differ by constant factors from the best possible bound. This raises the question whether more accurate bounds can be derived, possibly revealing differences in the preimage resistance between the three constructions. A related question is the estimation of non-trivial upper bounds on the preimage resistance.

## References

1. E. Fleischmann, M. Gorski and S. Lucks: On the security of Tandem-DM. FSE 2009, LNCS 5665, pp. 84–103. Springer, Heidelberg (2009)
2. E. Fleischmann, M. Gorski and S. Lucks: Security of cyclic double block length hash functions. Cryptography and Coding, 12th IMA International Conference, Cirencester, UK, LNCS 5921 pp. 153–175. Springer, Heidelberg (2009)
3. S. Hirose: Provably secure double-block-length hash functions in a black-box model. ICISC 2004, LNCS 3506, pp. 330–342. Springer, Heidelberg (2005)
4. S. Hirose: Some plausible constructions of double-block-length hash functions. FSE 2006, LNCS 4047, pp. 210–225. Springer, Heidelberg (2006)
5. X. Lai and J. Massey: Hash function based on block ciphers. Eurocrypt 1992, LNCS 658, pp. 55–70. Springer, Heidelberg (1993)
6. J. Lee and D. Kwon: The security of Abreast-DM in the ideal cipher model. `http://eprint.iacr.org/2009/225.pdf`
7. J. Lee, M. Stam and J. Steinberger: The security of Tandem-DM in the ideal cipher model. Crypto 2011, LNCS 6841, pp. 561–577. Springer, Heidelberg (2011).
8. O. Özen and M. Stam: Another Glance at Double-Length Hashing. Cryptography and Coding, 12th IMA International Conference, Cirencester, UK, LNCS 5921, pp. 94–115. Springer, Heidelberg (2009)
9. M. Rabin: Digitalized signatures. Foundations of Secure Computations, pages 155–166. Academic Press (1978)
10. P. Rogaway and T. Shrimpton: Cryptographic hash-function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision-resistance. FSE 2004, LNCS 3017, pp. 371–388. Springer, Heidelberg (2004)
11. M. Stam: Block cipher-based hashing revisited, FSE 2009, LNCS 5665, pp. 67–83. Springer, Heidelberg (2009)