Improved rate upper bound of collision resistant compression functions

Richard Ostertág*

Department of Computer Science, Faculty of Mathematics, Physics and Informatics,
Comenius University, Mlynská dolina, 842 48 Bratislava, Slovak Republic
ostertag@dcs.fmph.uniba.sk
http://www.dcs.fmph.uniba.sk

Abstract. *Based on Stanek's results [1] we know that in model with integer rate PGV like compression functions no high speed collision resistant compression functions exist. Thus we try to study more general multiple block ciphers based model of compression functions with rational rate, like 6/5. We show a new upper bound of the rate of collision resistant compression functions in this model.*

1 Motivation and goals

The cryptographic hash functions are a basic building block of many other cryptographic constructions (such as digital signature schemes, message authentication code, ...). For more complete overview see e.g. [2, 3].

Majority of modern hash functions is based on Merkle-Damgård paradigm [4, 5]. Many compression functions are explicitly based on block cipher. Even some of "dedicated" hash functions (which were not constructed in this way) have this structure. For example, it is possible to extract 160 bits block cipher with 512 bits key (called SHACAL-1) from compression function implemented in SHA-1 hash function [6].

The idea of hash function construction by iterating block cipher is at least 30 years old [7]. Nevertheless no systematic analysis of this idea was done until 1994. In this year Preneel, Govaerts and Vandewalle done the first systematic study of 64 hash functions based on block cipher [8]. Thereafter Black, Rogaway and Shrimpton [9] analyzed these constructions in black-box model and showed that 20 of them are collision resistant up to birthday-attack bound.

At least from the usability point of view, speed is important property of hash function. So it is only natural to attempt to speedups it. One of possible speedups of iterated hash functions based on block ciphers is increasing the number of input message blocks processed by one use of block cipher. Another possibility of speedup is a restriction of keys used in all block ciphers to a small fixed set of keys. Then it is possible to pre-schedule subkeys for each round of used block ciphers, whereby saving a big amount of work.

Traditional constructions [8] of hash functions require one block cipher transformation per input message block (so called rate-1 hash functions) and they require rekeying for every input message block. Black, Cochran and Shrimpton [10] showed in year 2005 that it is not possible to construct a provably secure rate-1 iterated hash function based on block cipher, which uses only small fixed set of keys.

For these reasons our goal is to maximize rate of iterated hash function based on block cipher. In other words, we attempt to maximize the number of input message blocks processed by a single block cipher invocation.

2 Notation

We now briefly introduce basic definitions and notations, following closely [10, 9].

Let V_m be set of all m-ary binary vectors, i.e. $V_m = \{0,1\}^m$. Let $V_m^* = (V_m)^*$ be set of all binary strings that we get by concatenation of zero or more elements from V_m. Let k and n be positive integers. A block cipher is a function $E : V_k \times V_n \to V_n$, where for each key $K \in V_k$, the function $E_K(\cdot) = E(K, \cdot)$ is a permutation on V_n. Let $\mathrm{Bloc}(k, n)$ be the set of all block ciphers $E : V_k \times V_n \to V_n$. Let denote E^{-1} the inverse of block cipher E.

A block cipher based compression function is a function $f : \mathrm{Bloc}(k, n) \times V_a \times V_b \to V_c$, where a, b and c are positive integers such that $a+b \geq c$. We will write the first argument (the block cipher) as superscript of the compression function, i.e. $f^E(\cdot,\cdot) = f(E,\cdot,\cdot)$. An iterated hash function based on compression function $f : \mathrm{Bloc}(k, n) \times V_a \times V_b \to V_a$ is the hash function $H : \mathrm{Bloc}(k, n) \times V_b^* \to V_a$ defined by $H^E(m_1 \ldots m_l) = h_l$, where $h_i = f^E(h_{i-1}, m_i)$ and h_0 is fixed element from V_a (so called initialization vector). Let $H^E(\varepsilon) = h_0$ for empty string ε. We often omit superscript E of functions f and H when it is apparent from the context which block cipher is used.

If the computation of $f^E(h, m)$ uses t queries on E, then compression function f (and its iterated hash function H) is rate-r, where $r = (b/n)/t$. Often b is

* Supported by VEGA grant No. 1/0266/09.

divisible by n. The rate r represents average number of input message blocks processed by a single enciphering transformation E. For example, if $b/n = 3$ and $t = 2$ then we get rate-$\frac{3}{2}$ compression function.

2.1 Black-box model

Black-box model (see e.g. [9]) is also known as ideal-cipher model. In this model, an adversary A is given access to oracles E and E^{-1}, where E is a block cipher. We write the oracles as superscripts, i.e. $A^{E,E^{-1}}$. Where used oracles are clear from the context, the superscript of A will be omitted.

Adversary A tries to find collisions in the compression function. Other cryptographic properties of compression functions are also important, but we focus exclusively on collision resistance, as on the most "problematic" property of compression functions. We will see that our results are negative, so it is not necessary to analyze other properties.

In the black-box model the adversary's collision finding effort is measured by the number of queries made to oracles E and E^{-1}. Computational power of the adversary is not limited in any way — i.e. we assume information-theoretic adversary.

Attacks in this model treat the block cipher as a black-box. The only modeled structural property of the block cipher is its invertibility. This model cannot guarantee security of compression functions based on weak block ciphers with inappropriate properties (such as weak keys). On the other hand, black-box model is stronger than model in which block cipher is assumed to be random function, because adversary can compute E^{-1}.

We say that inputs (h, m) and (h', m') of compression function f collide, if they are distinct and $f^E(h, m) = f^E(h', m')$. We say that (h, m) collides with empty string, if $f^E(h, m) = h_0$, where h_0 is initialization vector.

We write random draw of element x from finite set S as $x \overset{\$}{\leftarrow} S$. We will use notation $(x, y) \leftarrow A^{E,E^{-1}}$ for computation of two colliding inputs x and y by adversary A (represented by probabilistic algorithm) with knowledge of oracles E and E^{-1}.

Definition 1 (Coll. res. of comp. function [9]). *Let f be block cipher based compression function, $f : \mathrm{Bloc}(k, n) \times V_a \times V_b \to V_c$. Fix a constant $h_0 \in V_c$ and an adversary A. Then the advantage of adversary A (denoted by $\mathbf{Adv}_f^{\mathrm{comp}}(A)$) in finding collisions in compression function f is the following probability:*

$$\Pr\Big[E \overset{\$}{\leftarrow} \mathrm{Bloc}(k, n);\ \big((h, m), (h', m')\big) \leftarrow A^{E,E^{-1}} : \big((h, m) \neq (h', m') \wedge f^E(h, m) = f^E(h', m')\big) \vee f^E(h, m) = h_0\Big].$$

For any $q \geq 0$ we write:

$$\mathbf{Adv}_f^{\mathrm{comp}}(q) = \max_A\{\mathbf{Adv}_f^{\mathrm{comp}}(A)\}$$

where the maximum is taken over all adversaries that ask oracles (E or E^{-1}) at most q queries.

Definition 2 (Collision resistance of hash function [9]). *Let H be hash function based on block cipher. Let A be an adversary. Then the advantage of the adversary A in finding collisions in hash function H is the following probability:*

$$\mathbf{Adv}_H^{\mathrm{coll}}(A) = \Pr\Big[E \overset{\$}{\leftarrow} \mathrm{Bloc}(k, n);\ (M, M') \leftarrow A^{E,E^{-1}} : M \neq M' \wedge H^E(M) = H^E(M')\Big] .$$

For any $q \geq 0$ we write:

$$\mathbf{Adv}_H^{\mathrm{coll}}(q) = \max_A\{\mathbf{Adv}_H^{\mathrm{coll}}(A)\}$$

where the maximum is taken over all adversaries that ask oracles (E or E^{-1}) at most q queries.

The Merkle-Damgård construction of iterated hash functions is based on the following theorem. It states that iterated hash function is collision resistant if underlying compression function is collision resistant.

Theorem 1 (Merkle-Damgård [4, 5]). *Let $f : \mathrm{Bloc}(k, n) \times V_n \times V_n \to V_n$ be a compression function and let H be an iterated hash function of f. Then $\mathbf{Adv}_H^{\mathrm{coll}}(q) \leq \mathbf{Adv}_f^{\mathrm{comp}}(q)$ for any $q \geq 1$.*

Birth-day attack is generic way of attacking collision resistance of any compression or hash function. The advantage of finding collision by applying birthday attack is $\Theta(q^2/2^n)$, where q is number of evaluation of the function and n is output length.

If q depends on n, then we assume that $q(n) = o(2^{n/2})$, because greater $q(n)$ does not make sense, as we can still use generic birth-day attack with lower $q(n) = 2^{n/2}$ with unacceptably high probability ($\approx 1/2$) of finding collision.

Compression function f (or hash function H) is usually called collision resistant up to birthday attack bound or simply collision resistant if $\mathbf{Adv}_f^{\mathrm{comp}}(q) = O(q^2/2^n)$ (or $\mathbf{Adv}_H^{\mathrm{coll}}(q) = O(q^2/2^n)$). Since birth-day attack is always possible, we can rewrite these equations into equivalent form $\mathbf{Adv}_f^{\mathrm{comp}}(q) = \Theta(q^2/2^n)$ (or $\mathbf{Adv}_H^{\mathrm{coll}}(q) = \Theta(q^2/2^n)$).

3 Known results

In [11] we have proposed a model of rate-r compression functions that cover all compression functions that process r input message blocks of length n per block cipher invocation with a key of length k. In that paper we have showed that $1 + k/n$ is the upper bound of rate of any collision resistant compression function in such a model.

For typical constructions, when $k = n$, we get that if any high-rate collision resistant function in our model exists, then it is rate-2 compression function.

Consequently we have analyzed in [11] all rate-2 generalizations of compression functions from [8] (all of them are covered by our model). We have proved that none of them is collision resistant in the black-box model. Staneková and Stanek showed in [12] that either hash functions constructed from them are not collision resistant.

But these functions does not cover whole set of rate-2 compression functions from our model. Hence the question, if there exist any rate-2 collision resistant compression function still remains open.

This question is answered by Stanek in [1], where he improves our upper bound by utilizing the possibility of asking q queries during the attack (before the adversary ask only one query).

Theorem 2 (Stanek [1]). *Let $E \in \mathrm{Bloc}(k,n)$. Let $f^X : V_a \times V_{rn} \to V_n$, $f^K : V_a \times V_{rn} \to V_k$ and $f^C : V_a \times V_{rn} \times V_n \to V_a$ be arbitrary functions. Let $f : V_a \times V_{rn} \to V_a$ be compression function defined by $f(h,m) = f^C\big(h, m, E_{f^K(h,m)}(f^X(h,m))\big)$. Let $q \geq 1$ denote maximum number of queries on E and E^{-1}. Let $r > 1 + \frac{k - \log_2 q}{n}$. Then $\mathbf{Adv}_f^{\mathrm{comp}}(q) = 1$.*

By substituting $q = n$, $a = n$ and $k = n$ into theorem 2 we get upper bound for rate r in the following form $r > 2 - \frac{\log_2 n}{n}$. If we take into account that in our model rate r is always an integer, then we get following corollary of previous theorem.

Corollary 1 (Stanek [1]). *Let $E \in \mathrm{Bloc}(n,n)$. Let $f^X : V_n \times V_{rn} \to V_n$, $f^K : V_n \times V_{rn} \to V_n$ and $f^C : V_n \times V_{rn} \times V_n \to V_n$ be arbitrary functions. Let function $f : V_n \times V_{rn} \to V_n$ be a compression function defined by $f(h,m) = f^C\big(h, m, E_{f^K(h,m)}(f^X(h,m))\big)$. Let $r > 1$. Then f is not collision resistant in black-box model.*

Now our result about nonexistence of rate-2 collision resistant PGV-like compression functions from [11] follows from corollary 1. But the attack based on theorem 2 has exponential time complexity (and asks n oracle queries, even if it is not necessary). Therefore our attacks from [11] constructed specifically for rate-2 PGV-like compression and hash functions are still justified as they use only polynomial time and ask at most two queries.

Until now we have not modeled any compression function which uses more block ciphers per one compression function computation. For example:

$$f(h,m) = f^C\Big(h, m, E_1\big(f_1^K(h,m), f_1^X(h,m)\big), \\ E_2\big(f_2^K(h,m), f_2^X(h,m)\big)\Big) \ .$$

If m is created from four input message blocks, then this will be rate-2 compression function but is not covered by model from [11]. Also using of multiple block ciphers allows compression functions with rational rate. For example, if m is created from three input message blocks, then we get rate-$\frac{3}{2}$ compression function. Therefore we have concentrated on creation of new more general model.

4 The generalized model of compression function

A compression function f based on t block ciphers[1] is function defined by $f : \mathrm{Bloc}(k,n)^t \times V_a \times V_b \to V_c$, where a, b and c are positive integers such that $a + b \geq c$. When we will need to emphasize number of used block ciphers t, then we will write t as superscript of compression function, i.e. f^t. Iterated hash function based on compression function $f : \mathrm{Bloc}(k,n)^t \times V_a \times V_b \to V_a$ is function $H : \mathrm{Bloc}(k,n)^t \times V_b^* \to V_a$ defined by $H((E_1, \dots, E_t), m_1 \dots m_l) = h_l$, where $h_i = f((E_1, \dots, E_t), h_{i-1}, m_i)$ and h_0 is fixed element from V_a. We define $H((E_1, \dots, E_t), \varepsilon)$ to be equal to h_0. If block ciphers used in functions f and H are clear from the context, then we will omit them as arguments of these functions.

Now we will start to define the general model of compression function $f^t : \mathrm{Bloc}(k,n)^t \times V_a \times V_b \to V_a$ based on t block ciphers. Model is based on following assumptions:

- Computation of compression function f^t asks exactly one query on each oracle E_i for the purpose of evaluation of $f^t(h,m)$.
  This assumption is without loss of the generality. We do not assume that in practice all $E_1, \dots, E_t$ are distinct, but the model allows it. If for computation of function f^t we need to evaluate E_i, e.g. two times, then we can set $E_{t+1} = E_i$ and use function f^{t+1} defined analogically as f^t but with the only exception, that in place of second evaluation of E_i evaluation of E_{t+1} will be used.

[1] As we will clarify in following paragraph, it is important that t queries on oracles are made during each evaluation of compression function f. It does not matter, if the same block cipher is invoked t times, or if t different block ciphers are invoked exactly once. Hence some of t block ciphers can be equal.

  Analogically, it does not make sense to specify block cipher E_i if it is not used during any calculation of compression function f.
  This assumption about f^t guarantees that every computation of f^t always asks exactly t queries on oracles.

- Computation of compression function f^t asks oracles E_i in order of their indexes[2]. Thus we can assume that evaluation of block cipher E_i had to occur before evaluation of E_{i+1}.
- The length of input message block m_i of compression function does not have to be divisible by block cipher E plain text block length n.

In following text we will often work with sequences, therefore we now clarify some necessary notation.

Definition 3. *Empty sequence will be denoted by* $()$. *We will write* $(a_1, a_2, \ldots, a_n)$ *for a sequence with* n *elements* a_1, a_2, $\ldots$, a_n. *Sequences will be denoted by upper case letters with a overscore, for example* $\overline{Y}$. *For addition of element* a_{n+1} *at the end of a sequence* $(a_1, a_2, \ldots, a_n)$ *we will use operation "$\cdot$" in the following way:* $(a_1, a_2, \ldots, a_n) \cdot a_{n+1} = (a_1, a_2, \ldots, a_n, a_{n+1})$.

Let for all $i \in \{1, 2, \ldots, t\}$ $f_i^X : V_a \times V_b \times V_n^{i-1} \to V_n$ and $f_i^K : V_a \times V_b \times V_n^{i-1} \to V_k$ be arbitrary functions. Let $f^C : V_a \times V_b \times V_n^t \to V_a$ be arbitrary function. Computation of compression function $f^t : \mathrm{Bloc}(k,n)^t \times V_a \times V_b \to V_a$ in generalized model is defined by the following algorithm 1.

Algorithm 1 The gen. model of compression function

```
function $f((E_1, \ldots, E_t); h; m)$
    $\overline{Y}_0 = ()$
    for $i = 1$ to $t$ do
        $X_i \leftarrow f_i^X(h, m, \overline{Y}_{i-1})$
        $K_i \leftarrow f_i^K(h, m, \overline{Y}_{i-1})$
        $Y_i \leftarrow E_i(K_i, X_i)$
        $\overline{Y}_i \leftarrow \overline{Y}_{i-1} \cdot Y_i$
    end for
    return $f^C(h, m, \overline{Y}_t)$
end function
```

Remark 1. Function f_i^X, respective function f_i^K prepares the plain text, respective the key for the block cipher E_i. Both inputs h and m are arguments of these functions together with all already computed cipher texts $Y_1, Y_2, \ldots, Y_{i-1}$. At the end of the algorithm, function f^C processes both inputs h and m with all intermediate results $Y_1, Y_2, \ldots, Y_t$ into final result. The algorithm uses t functions f_i^X and t functions f_i^K. But function f^C is just one. Introduction of analogous "postprocessing" for every block cipher (i.e. for each round) is needless. Calculation of local postprocessing at the end of i-th round can be incorporated into functions f_j^X and f_j^K of following rounds (i.e. for all $j > i$) and into function f^C.

The compression function f^t (and its iterated hash function H) have rate $r = (b/n)/t$.

This generalized model of compression functions covers all compression functions, which takes messages of length a and b and process them using exactly t block ciphers $E_1, E_2, \ldots, E_t$ from $\mathrm{Bloc}(k,n)$ in this specified order, into message of length a. All rate-1 schemes from [8] and their rate-2 generalizations fall into this model.

5 Upper bound of rate of collision resistant compression functions

Before proof of the upper bound we first define some auxiliary notions and prove some lemmas.

Definition 4. *Let* $i \in \{0, 1, \ldots, t\}$ *and* $(h,m) \in V_a \times V_b$. *If* $i = 0$ *then* $\overline{Y}_{i,(h,m)} = ()$. *If* $i > 0$ *then we define* $\overline{Y}_{i,(h,m)}$ *recursively as follows:*

$$\overline{Y}_{i-1,(h,m)} \cdot E_i\Big(f_i^K\big(h, m, \overline{Y}_{i-1,(h,m)}\big), f_i^X\big(h, m, \overline{Y}_{i-1,(h,m)}\big)\Big) \ .$$

Sequence $\overline{Y}_{i,(h,m)}$ represents individual Y_i calculated during individual rounds of $f^t(h,m)$ evaluation. It can easily be seen that $\overline{Y}_{i-1,(h,m)}$ is prefix of $\overline{Y}_{i,(h,m)}$ and that $\overline{Y}_{t,(h,m)}$ is equal to $\overline{Y}_t$, which is created during evaluation of compression function $f^t(h,m)$.

Definition 5. *Let* $i \in \{1, 2, \ldots, t\}$, $X \in V_n$, $K \in V_k$ *and let* $2^{n+k} > \alpha > 0$ *be an integer. Let* $S \subseteq V_a$, *where* $|S| = s > 0$. *Then* $D_\alpha^0 = S \times V_b$ *and* D_α^i *is union of* α *largest sets* $D_{X,K}^i$ *taken through all* X *and* K *(let denote them* $D^i_{X_1^i,K_1^i}, \ldots, D^i_{X_\alpha^i,K_\alpha^i}$*), where* $D_{X,K}^i$ *is defined as follows:*

$$D_{X,K}^i = \Big\{(h,m) \in D_\alpha^{i-1} \,\Big|\, f_i^X\big(h, m, \overline{Y}_{i-1,(h,m)}\big) = X \wedge \\ \wedge\, f_i^K\big(h, m, \overline{Y}_{i-1,(h,m)}\big) = K\Big\} \ .$$

Set $D_{X,K}^i$ is subset of D_α^{i-1}. It consists of those elements, which in next (i-th) round will lead to the same query $E_i(X,K)$ on oracle E_i. That means that to compute next round for all elements from $D_{X,K}^i$ one oracle

[2] Requirement of fixed evaluation order of block ciphers is not so restrictive as it can seem. We can simulate compression function with variable evaluation order of t block ciphers by compression function with fixed evaluation order of t^2 block ciphers. See e.g. discussion at the end of section 2 in [13].

query is sufficient. Construction of sets $D^i_{X,K}$ have of course exponential complexity, but does not require any oracle queries. Since we use black-box model, adversary have computationally unlimited power and is limited only by number of oracle queries.

Set D^1_α is the largest set of tuples $(h,m) \in S \times V_b$, for which we can made first round of compression function f^t with spending exactly α queries on oracle E_1. By definition D^1_α is union of α largest sets $D^1_{X^1_1,K^1_1}, \ldots, D^1_{X^1_\alpha,K^1_\alpha}$. For the calculation of the first round for elements from every set $D^1_{X^1_j,K^1_j}$ we need one query $E_1(X^1_j, K^1_j)$ on oracle E_1. Since all tuples (X^1_j, K^1_j) are distinct, we need exactly α queries for selected α sets.

We do not know how to estimate cardinality of set, which is the largest set of tuples $(h,m) \in S \times V_b$, for which we can do first two rounds of compression function f with at most 2α queries on oracles E_1 and E_2. However we know how to estimate cardinality of set D^2_α, which is such largest set of tuples $(h,m) \in D^1_\alpha$. Therefore we have constructed set D^i_α as subset of D^{i-1}_α. Then we are able to lower bound cardinality of set D^i_α in following way.

Lemma 1. *Let $1 \le \alpha \le 2^{n+k}$ and let $0 \le i \le t$ be integers. Then $|D^i_\alpha| \ge \alpha^i 2^{b-i(n+k)} s$.*

Proof. (Using mathematical induction over i.)
IND. BASIS: $|D^0_\alpha| = |S \times V_b| = s2^b \ge \alpha^0 2^{b-0(n+k)} s$.
IND. HYPOTHESIS: Let $|D^i_\alpha| \ge \alpha^i 2^{b-i(n+k)} s$.
IND. STEP: Then $|D^{i+1}_\alpha| \ge \alpha^{i+1} 2^{b-(i+1)(n+k)} s$.

Set $|D^{i+1}_\alpha|$ is by definition 5 union of $\alpha < 2^{n+k}$ largest sets $D^{i+1}_{X,K}$. Nonempty sets $D^{i+1}_{X,K}$ are all distinct and their union is equal to D^i_α. In other words, elements of the set D^i_α are divided into 2^{n+k} shelves. Then using pigeonhole principle we can estimate cardinality of α largest of them in the following way:

$$\begin{aligned}|D^{i+1}_\alpha| &\ge \alpha \frac{|D^i_\alpha|}{2^{n+k}} \ge \\ &\ge \alpha \frac{\alpha^i 2^{b-i(n+k)} s}{2^{n+k}} = \alpha^{i+1} 2^{b-(i+1)(n+k)} s \ .\end{aligned}$$

□

Lemma 2. *At most $t\alpha$ queries on oracles $E_1, \ldots, E_t$ are sufficient for computation of set D^t_α among with values of compression function $f^t(h,m)$ for all tuples (h,m) from the set D^t_α.*

Proof. We construct matrix M, which has on i-th row tuples $(X^i_1, K^i_1) \ldots (X^i_\alpha, K^i_\alpha)$ used during the construction of set D^i_α by taking the union of α largest sets $D^i_{X^i_1,K^i_1}, \ldots, D^i_{X^i_\alpha,K^i_\alpha}$. M has t rows and α columns, so matrix M have totally $t\alpha$ elements.

$$M = \begin{pmatrix} (X^1_1,K^1_1) \ldots (X^1_j,K^1_j) \ldots (X^1_\alpha,K^1_\alpha) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ (X^i_1,K^i_1) \ldots (X^i_j,K^i_j) \ldots (X^i_\alpha,K^i_\alpha) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ (X^t_1,K^t_1) \ldots (X^t_j,K^t_j) \ldots (X^t_\alpha,K^t_\alpha) \end{pmatrix}$$

The only place where queries are made during the computation of sets D^i_α is the computation of $\overline{Y}_{i,(h,m)}$. During the construction of set D^0_α no queries on oracles are necessary as it is $S \times V_b$ by definition. Similarly during the construction of set D^1_α no queries on oracles are necessary as $\overline{Y}_{0,(h,m)}$ is by definition empty.

During the construction of D^i_α for $i \in \{2, 3, \ldots, t\}$ all queries will be on oracles $E_1, \ldots, E_{i-1}$. Queries on oracle E_1 will be only from the first row of matrix M, queries on oracle E_2 will be only from the second row, and so on, ending with queries on oracle E_{i-1}, which are only from $(i-1)$-th row of matrix M. Last row of matrix M (together with all others) is used during the computation of values $f^t(h,m) = f^C(h,m,\overline{Y}_{t,(h,m)})$ for all $(h,m) \in D^t_\alpha$.

During the computation of D^i_α a new i-th row is created in the matrix M. Tuples (X^{i-1}_j, K^{i-1}_j) from $(i-1)$-th row are for the first time evaluated by oracle E_{i-1}. Queries on oracle E_l for $l < i-1$ will be only from already evaluated row l of matrix M. That follows from the fact that $D^i_\alpha \subseteq D^{i-1}_\alpha$. Therefore we will need at most $t\alpha$ queries on oracles $E_1, \ldots, E_t$ during the computation of D^t_α together with values of compression function $f^t(h,m)$ for all $(h,m) \in D^t_\alpha$ if we remember already asked queries together with corresponding answer. □

Theorem 3. *Let $f : \mathrm{Bloc}(k,n)^t \times V_a \times V_b \to V_a$ be arbitrary rate-r compression function defined by algorithm 1, while $r = \frac{b/n}{t}$. Let $q \ge 1$ be maximum allowed number of queries on oracles E_i and E_i^{-1}. Let q be an integer of the form $q = t\alpha$, where $\alpha \ge 1$ is also an integer*[3]*. Let $r > 1 + \frac{k}{n} - \frac{\log_2 \alpha}{n}$. Then $\mathbf{Adv}^{\mathrm{comp}}_f(q) = 1$.*

Proof. By asking at most q queries we are according to lemma 2 able to compute values of $f^t(h,m) \in V_a$ for all $(h,m) \in D^t_\alpha$. Let $S = V_a$, thus $s = |S| = 2^a$. According to lemma 1 we know that $|D^t_\alpha| \ge \alpha^t 2^{b-t(n+k)} s = \alpha^t 2^{a+b-t(n+k)}$. We can guarantee that between computed values there are at least two identical values if:

$$\begin{aligned}\alpha^t 2^{a+b-t(n+k)} &> 2^a \\ t\log_2\alpha + a + b - t(n+k) &> a \\ b &> t(n+k) - t\log_2\alpha \ .\end{aligned}$$

[3] This requirement is natural. For computation of f^t we need t oracle queries. Hence if we set $q = t\alpha$, then as if we allow α complete computations of f^t.

Now we rewrite this inequality into required form by using following equality $r = \frac{b/n}{t}$:

$$\begin{aligned} b &> t(n+k) - t\log_2 \alpha \\ \frac{b/n}{t} &> \frac{n+k}{n} - \frac{\log_2 \alpha}{n} \\ r &> 1 + \frac{k}{n} - \frac{\log_2 \alpha}{n} \quad . \end{aligned}$$

This means that with probability 1 we can find (and so the adversary) collision in the compression function f, while asking at most q queries on oracles. Hence $\mathbf{Adv}_f^{\mathrm{comp}}(q) = 1$ holds. □

Computation of the particular D_α^i has exponential complexity. Also finding the collision between values $f^t(h,m)$ for all $(h,m) \in D_\alpha^t$ has exponential complexity. But computationally unlimited adversary of black-box model can do all this unless he does not ask more than q queries on oracles.

Theorem 3 gives upper bound depending on number of oracle queries. The following corollary adapts the previous theorem in such a way, that instead of number of queries q, the number of output bits a of compression function is used in the inequality for r.

Corollary 2. *Let* $f^t : \mathrm{Bloc}(k,n)^t \times V_a \times V_b \to V_a$ *be arbitrary rate-r compression function defined by algorithm 1, where* $r = \frac{b/n}{t}$. *Let* $0 \le \varepsilon < \frac{1}{2}$ *be arbitrary constant. Let* $r > 1 + \frac{k}{n} - \varepsilon\frac{a}{n}$. *Then* f^t *is not collision resistant.*

Proof. Let $0 \le \lambda < 1$ be arbitrary constant. Then we set $q = t2^{\lambda\frac{a}{2}}$, i.e. $\alpha = 2^{\lambda\frac{a}{2}}$. Then by substituting into theorem 3 we get that $\mathbf{Adv}_f^{\mathrm{comp}}(q) = 1$ (that means according to size of q that f^t is not collision resistant) if:

$$\begin{aligned} r &> 1 + \frac{k}{n} - \frac{\log_2 \alpha}{n} \\ r &> 1 + \frac{k}{n} - \frac{\log_2 2^{\lambda\frac{a}{2}}}{n} \\ r &> 1 + \frac{k}{n} - (\lambda/2)\frac{a}{n} \quad . \end{aligned}$$

Now we make a substitution $\varepsilon = \lambda/2$ and required inequality follows:

$$r > 1 + \frac{k}{n} - \varepsilon\frac{a}{n}, \text{ where } 0 \le \varepsilon < \frac{1}{2} \quad .$$

□

As we have already mentioned, constructions of compression function based on block cipher, often have the same size of the key and the plain-text input of block cipher, i.e. $k = n$. Similarly, the output of compression function have usually the same size, i.e. $a = n$. For this typical situation we can simplify corollary 2.

Corollary 3. *Let* $f^t : \mathrm{Bloc}(n,n)^t \times V_n \times V_b \to V_n$ *be arbitrary rate-r compression function defined by algorithm 1, where* $r = \frac{b/n}{t}$. *Let* $r > 3/2$. *Then compression function* f^t *is not collision resistant.*

Proof. After substituting n for a and k into corollary 2 we get that compression function f^t is not collision resistant if $r > 1 + \frac{n}{n} - \varepsilon\frac{n}{n} = 2 - \varepsilon$ for an arbitrary constant $0 \le \varepsilon < \frac{1}{2}$.

That implies that compression function f^t is not collision resistant if $r > 2 - \frac{1}{2} = 3/2$. □

In generalized model rate r of compression function can be rational number and not only integer as in [11]. Therefore based on our results we cannot conclude that no high rate compression function exists in the generalized model. Still it is possible that e.g. rate-$\frac{6}{5}$ collision resistant compression function exists.

6 Conclusion

In our effort to find high speed collision resistant compression function we have introduced and studied new generalized model of compression function since in all previous models it was proved that no such functions exists. This model introduces rational rates, so we can study more precisely the rate upper bound of collision resistant compression functions. Based on previous results, it seems to be less than or equal to 2. We have improved this bound to be less than or equal to $\frac{3}{2}$.

References

1. M. Stanek: *Analysis of fast blockcipher-based hash functions.* In: Computational Science and Its Applications – ICCSA 2006, Springer, 2006, 426–435.
2. D.R. Stinson: *Cryptography: Theory and Practice*, Third Edition. Chapman & Hall/CRC, Boston, MA, USA, 2005.
3. A.J. Menezes, P.C. van Oorschot, S.A. Vanstone: *Handbook of Applied Cryptography.* CRC-Press, Boca Raton, FL, USA, 1996.
4. R.C. Merkle: *One way hash functions and DES.* Volume 435 of Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 1990, 428–446.
5. I.B. Damgård: *A design principle for hash functions.* Volume 435 of Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 1990, 416–427.
6. H. Handschuh, L.R. Knudsen, M.J. Robshaw: *Analysis of SHA-1 in encryption mode.* Volume 2020 of Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 2001, 70–83.
7. M.O. Rabin: *Digitalized signatures.* In Millo R.D., Dobkin D., Jones A., Lipton R., eds.: Foundations of Secure Computations, New York, Academic Press, 1978, 155–166.

8. B. Preneel, R. Govaerts, J. Vandewalle: *Hash functions based on block ciphers: A synthetic approach.* Volume 773 of Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 1994, 368–378.
9. J. Black, P. Rogaway, T. Shrimpton: *Black-box analysis of the block-cipher-based hash-function constructions from PGV.* Volume 2442 of Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 2002, 103–118.
10. J. Black, M. Cochran, T. Shrimpton: *On the impossibility of highly-efficient blockcipher-based hash functions.* Volume 3494 of Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 2005, 526–541.
11. R. Ostertág, M. Stanek: *On high-rate cryptographic compression functions.* Computing and Informatics **26**, 2007, 77–87.
12. L. Staneková, M. Stanek: *Generalized PGV hash functions are not collision resistant.* In: ITAT: Information Technologies – Applications and Theory, Seòa: PONT, 2006, 139–143.
13. P. Rogaway, J. Steinberger: *Security/efficiency tradeoffs for permutation-based hashing.* Volume 4965 of Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 2008, 220–236.