Hash Function Balance and its Impact on Birthday Attacks

Mihir Bellare[1] and Tadayoshi Kohno[1]

Dept. of Computer Science & Engineering, University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093, USA.
Email: {mihir,tkohno}@cs.ucsd.edu
URL: http://www-cse.ucsd.edu/users/{mihir,tkohno}

Abstract. Textbooks tell us that a birthday attack on a hash function h with range size r requires $r^{1/2}$ trials (hash computations) to find a collision. But this is quite misleading, being true only if h is regular, meaning all points in the range have the same number of pre-images under h; if h is not regular, *fewer* trials may be required. But how much fewer? This paper addresses this question by introducing a measure of the "amount of regularity" of a hash function that we call its balance, and then providing estimates of the success-rate of the birthday attack, and the expected number of trials to find a collision, as a function of the balance of the hash function being attacked. In particular, we will see that the number of trials can be significantly less than $r^{1/2}$ for hash functions of low balance. This leads us to examine popular design principles, such as the MD (Merkle-Damgård) transform, from the point of view of balance preservation, and to mount experiments to determine the balance of popular hash functions.

1 Introduction

Birthday attacks. Let $h\colon D \to R$ be a hash function. In a birthday attack, we pick points $x_1, \ldots, x_q$ from D and compute $y_i = h(x_i)$ for $i = 1, \ldots, q$. The attack is successful if there exists a collision, i.e. a pair i, j such that $x_i \neq x_j$ but $y_i = y_j$. We call q the number of *trials.*

There are several variants of this attack which differ in the way the points $x_1, \ldots, x_q$ are chosen (cf. [4, 8–10]). The one we consider is that they are chosen independently at random from D.[1]

Textbooks (eg. Stinson [8, Section 7.3]) say that (due to the birthday phenomenon which gives the attack its name) a collision is expected within $r^{1/2}$ trials, where r denotes the size of the range of h. In particular, they say that

[1] One might ask how to mount the attack (meaning how to pick random domain points) when the domain is a very large set as in the case of a hash function like SHA-1 whose domain is the set of all strings of length at most 2^{64}. We would simply let h be the restriction of SHA-1 to inputs of some reasonable length, like 161 bits or 320 bits. A collision for h is a collision for SHA-1, so it suffices to attack the restricted function.

collisions in a hash function with output length m bits can be found in about $2^{m/2}$ trials. This estimate is the basis for the choice of hash function length m, which is typically made just large enough to make $2^{m/2}$ trials infeasible.

However Stinson's analysis [8, Section 7.3], as well as all others that we have seen, are misleading, for they assume the hash function is *regular*, meaning all points in the range have the same number of pre-images under h.[2] It turns out that if h is not regular, it takes *fewer* than $r^{1/2}$ trials to find a collision, meaning the birthday attack would succeed sooner than expected.

This could be dangerous, for we do not know that popular hash functions are regular. In fact they are usually designed to have "random" behavior and thus would not be regular. Yet, one might say, they are probably "almost" regular. But what exactly does this mean, and how does the "amount of regularity" affect the number of trials to success in the birthday attack? Having answers to such questions will enable us to better assess the *true* impact of birthday attacks.

THIS PAPER. To help answer questions such as those posed above, this paper begins by introducing a measure of the "amount of regularity" that we call the *balance* of a hash function. This is a real number between 0 and 1, with balance 1 indicating that the hash function is regular and balance 0 that it is a constant function, meaning as irregular as can be. We then provide quantitative estimates of the success-rate, and number of trials to success, of the birthday attack, as a function of the balance of the hash function being attacked.

This yields a tool that has a variety of uses, and lends insight into various aspects of hash function design and parameter choices. For example, by analytically or experimentally estimating the balance of a particular hash function, we can tell how quickly the birthday attack on this hash function will succeed. Let us now look at all this in more detail.

THE BALANCE MEASURE. View the range R of hash function $h\colon D \to R$ as consisting of $r \geq 2$ points $R_1, \ldots, R_r$. For $i = 1, \ldots, r$ we let $h^{-1}(R_i)$ be the pre-image of R_i under h, meaning the set of all $x \in D$ such that $h(x) = R_i$, and let $d_i = |h^{-1}(R_i)|$ be the size of the pre-image of R_i under h. We let $d = |D|$ be the size of the domain. We define the balance of h as

$$\mu(h) \;=\; \log_r \left[\frac{d^2}{d_1^2 + \cdots + d_r^2} \right] \;,$$

where $\log_r(\cdot)$ denotes the logarithm in base r. Proposition 1 says that for any hash function h, the balance of h is a real number in the range from 0 to 1. Furthermore, the maximum balance of 1 is achieved when h is regular (meaning $d_i = d/r$ for all i) and the minimum balance of 0 is achieved when h is a constant function (meaning $d_i = d$ for some i and $d_j = 0$ for all $j \neq i$). Thus regular functions are well-balanced and constant functions are poorly balanced, but there are lots of possibilities in between these extremes.

[2] They regard x_i as a ball thrown into bin $h(x_i)$ and then apply the standard birthday analysis. But the latter assumes each ball is equally likely to land in each bin. If $R_1, \ldots, R_r$ denote the range points then the probability that a ball lands in bin R_j is $|h^{-1}(R_j)|/d$ where $d = |D|$. These values are all the same only if h is regular.

RESULTS. We are interested in the probability C of finding a collision in q trials of the birthday attack, and also in the *threshold* Q, defined as the number of trials required for the expected number of collisions to be one. (Alternatively, the expected number of trials to find a collision.) Corollary 1 and Theorem 2, respectively, say that, up to constant factors,[3]

$$C \;=\; \binom{q}{2} \cdot \frac{1}{r^{\mu(h)}} \qquad \text{and} \qquad Q \;=\; r^{\mu(h)/2}\,. \tag{1}$$

These results indicate that the performance of the birthday attack can be characterized, quite simply and accurately, via the balance of the hash function h being attacked.

REMARKS. Note that when $\mu(h) = 1$ (meaning, h is regular) then Equation (1) says that, up to constant factors, $Q = r^{1/2}$, which agrees with the above-discussed standard estimate for this case. At the other extreme, when $\mu(h) = 0$, meaning h is a constant function, the attack finds collisions in $O(1)$ trials so $Q = 1$. The value of the general results of Equation (1) is that they show the full spectrum in between the extremes of regular and constant functions. As the balance of the hash function drops, the threshold Q of the attack decreases, meaning collisions are found faster. For example a birthday attack on a hash function of balance $\mu(h) = 1/2$ will find a collision in about $Q = r^{1/4}$ trials, which is significantly less than $r^{1/2}$. Thus, we now have a way to quantitatively assess how irregularity in h impacts the success-rate of the birthday attack.

We clarify that the attacker does not need to know the balance of the hash function in order to mount the attack. (The attack itself remains the birthday attack outlined above.)

BOUNDS RATHER THAN APPROXIMATE EQUALITIES. Corollary 1 provides both upper and lower bounds on C that are tight in the sense of being within a constant factor (specifically, a factor of four) of each other. (And Theorem 1 does even better, but the expressions are a little more complex.) Similarly, Theorem 2 provides upper and lower bounds on Q that are within a constant factor of each other.

We claim bounds are important. The estimates of how long the birthday attack takes to succeed, and the ensuing choices of output-lengths of hash functions, have been based so far on textbook approximate equality calculations of the threshold that are usually upper bounds but not lower bounds on the exact value. Yet, from a design perspective, the relevant parameter is actually a lower bound on the threshold since otherwise the attack might be doing better than we estimate.

The quality (ie. tightness) of the bounds is also important. Deriving a good lower bound on C required significantly more analytical work than merely producing a rough estimate of approximate equality. With regard to Q we remark that our upper bound, although within a constant factor of the lower bound, is not as tight as would like, and it is an interesting question to improve it.

---

[3] This assumes $d \geq 2r$ and, in the case of C, that $q \leq O(r^{\mu(h)/2})$.

Impact on output lengths. Suppose we wish to design a hash function h for which the birthday attack threshold is 2^{80} trials. A consequence of our results above is that we must have $r^{\mu(h)/2} = 2^{80}$, meaning must choose the output-length of the hash function to be $160/\mu(h)$ bits. Thus to minimize output-length we must maximize balance, meaning we would usually want to design hash functions that are almost regular (balance close to one).

The general principle that hash functions should be as close to regular as possible is, we believe, well-known as a heuristic. Our results, however, provide a way of quantifying the loss in security as a function of deviations from regularity.

Random hash functions. Designers of hash functions often have as target to make the hash function have "random" behavior. Proposition 2 together with Equation (1) enable us to estimate the impact of this design principle on birthday attacks. As an example, they imply that if h is a random hash function with $d = 2r$ then the expected probability of a collision in q trials is about $3/2$ times what it would be for a regular function, while the expected threshold is about $\sqrt{2/3}$ times what it would be for a regular function. In particular, random functions are *worse* than regular functions from the point of view of protection against birthday attacks, though the difference between random and regular functions decrease as the ratio d/r increases.

Thus, if one wants the best possible protection against both birthday and cryptanalytic attacks, one should design a function that is not entirely random but random subject to being regular. This is true both of the hash function itself, and of the hash function restricted to domains from which the adversary may draw points in its attack (eg. a restriction of SHA-1 to all 161-bit strings). This, however, may be more difficult than designing a hash function that has entirely random behavior, so that the latter remains the design goal, and in this case it is useful to have tools like ours that enable designers to estimate the impact of deviations from regularity on the birthday attack and fine tune output lengths if necessary.

Does the MD transform preserve balance? Given the above results we would like to be building hash functions that have high balance. We look at some elements of current design to see how well they reflect this requirement.

Hash functions like MD5 [7], SHA-1 [6] and RIPEMD-160 [3] are designed by applying the Merkle-Damgård (MD) [5, 2] transform to an underlying compression function. Designers could certainly try to ensure that the compression function is regular or has high balance, but this turns out not to be enough to ensure high balance of the hash function because Proposition 3 shows that the MD transform does not preserve regularity or maintain balance. (We give an example of a compression function that has balance one, yet the hash function resulting from the MD transform applied to this compression function has balance zero.)

Proposition 4 is more positive, showing that regularity not only of the compression function but also of certain associated functions does suffice to guarantee regularity of the hash function. But Proposition 5 notes that if the compression and associated functions have even minor deviations from regularity, meaning balance that is high but not equal to one, then the MD transform can amplify the imbalance and result in a hash function with very low balance.

Given that a random compression function has balance close to but not equal to one, and we expect practical compression functions to be similar, our final conclusion is that we cannot recommend, as a general design principle, attempting to ensure high balance of a hash function by only establishing some properties of the compression function and hoping the MD transform does the rest.

We stress that none of this implies any weaknesses in specific existing hash functions such as those mentioned above. But it does indicate a weakness in the MD transform based design principle from the point of view of ensuring high balance, and means that if we want to ensure or verify high balance of a hash function we might be forced to analyze it directly rather than being able to concentrate on the possibly simpler task of analyzing the compression function. We turn next to some preliminary experimental work in this vein with SHA-1.

Experimenting with SHA-1. The hash function SHA-1 was designed with the goal that the birthday attack threshold is about 2^{80} trials. As per the above, this goal would only be met if the balance of the hash function was close to one. More precisely, letting $\mathrm{SHA}_n\colon \{0,1\}^n \to \{0,1\}^{160}$ denote the restriction of SHA-1 to inputs of length $n < 2^{64}$, we would like to know whether SHA_n has balance close to one for practical values of n, since otherwise a birthday attack on SHA_n will find a collision for SHA-1 in less than 2^{80} trials.

The balance of SHA_n is however hard to compute, and even to estimate experimentally, when n is large. Section 6 however reports on some experiments that compute $\mu(\mathrm{SHA}_{32;t_1\ldots t_2})$ for small values of $t_2 - t_1$, where $\mathrm{SHA}_{n;t_1\ldots t_2}\colon \{0,1\}^n \to \{0,1\}^{t_2-t_1+1}$ is the function which returns the t_1-th through t_2-th output bits of SHA_n. The computed values for $\mu(\mathrm{SHA}_{32;t_1\ldots t_2})$ are extremely close to what one would expect from a random function with the same domain and range. Toward estimating the balance of SHA_n for larger values of n, Section 6 reports on some experiments on $\mathrm{SHA}_{n;t_1\ldots t_2}$ for larger n. Broadly speaking, the experiments indicate that these functions have high balance. This can be taken as some indication that SHA_n also has high balance, meaning SHA-1 is well-designed from the balance point of view.

Remarks. We clarify that while high balance is a necessary requirement for a collision-resistant hash function, it is certainly not sufficient. It is easy to give examples of high-balance hash functions for which it easy to find collisions. High balance is just one of many design criteria that designers should consider.

We also clarify that this paper does not uncover any weaknesses, or demonstrate improved performance of birthday attacks, on any specific, existing hash functions such as those mentioned above. However it provides analytical tools that contribute toward the goal of better understanding the security of existing hash functions or building new ones, and suggests a need to put more effort into estimating the balance of existing hash functions to see whether weaknesses exist.

2 Notation and Terminology

If n is a non-negative integer then we let $[n] = \{1, \ldots, n\}$. If S is a set then $|S|$ denotes its size. We denote by $h\colon D \to R$ a function mapping domain D to range R, and throughout the paper we assume that R has size at least two. We usually denote $|D|$ by d and $|R|$ by r. A *collision* for h is a pair x_1, x_2 of points in D such that $x_1 \neq x_2$ but $h(x_1) = h(x_2)$. For any $y \in R$ we let

$$h^{-1}(y) \;=\; \{\, x \in D \;:\; h(x) = y \,\} \; .$$

We say that h is *regular* if $|h^{-1}(y)| = d/r$ for every $y \in R$, where $d = |D|$ and $r = |R|$.

3 The Balance Measure and its Properties

We introduce a measure that we call the *balance*, and establish some of its basic properties.

Definition 1. *Let $h\colon D \to R$ be a function whose domain D and range $R = \{R_1, \ldots, R_r\}$ have sizes $d, r \geq 2$, respectively. For $i \in [r]$ let $d_i = |h^{-1}(R_i)|$ denote the size of the pre-image of R_i under h. The balance of h, denoted $\mu(h)$, is defined as*

$$\mu(h) \;=\; \log_r \left[\frac{d^2}{d_1^2 + \cdots + d_r^2} \right] \; , \tag{2}$$

where $\log_r(\cdot)$ denotes the logarithm in base r. ▮

It is easy to see that a regular function has balance 1 and a constant function has balance 0. The following says that these are the two extremes: In general, the balance is a real number that could fall somewhere in the range between 0 and 1. The proof is based on standard facts and provided in the full version of this paper [1] for completeness.

Proposition 1. *Let h be a function. Then*

$$0 \;\leq\; \mu(h) \;\leq\; 1 \; . \tag{3}$$

Furthermore, $\mu(h) = 0$ iff h is a constant function, and $\mu(h) = 1$ iff h is a regular function. ▮

The following lemma, which we prove in [1], will be useful later.

Lemma 1. *Let $h\colon D \to R$ be a function. Let $d = |D|$ and $r = |R|$ and assume $d \geq r \geq 2$. Then*

$$r^{-\mu(h)} - \frac{1}{d} \;\geq\; \left(1 - \frac{r}{d}\right) \cdot r^{-\mu(h)} \; , \tag{4}$$

where $\mu(h)$ is the balance of h as per Definition 1. ▮

```
For i = 1, ..., q do        // q is the number of trials
    Pick x_i at random from the domain of h
    y_i ← h(x_i)        // Hash x_i to get y_i
    If there exists j < i such that y_i = y_j but x_i ≠ x_j then
        return x_i, x_j        // collision found
    EndIf
EndFor
Return ⊥        // No collision found
```

Fig. 1. Birthday attack on a hash function $h\colon D \to R$. The attack is successful in finding a collision if it does not return $\perp$. We call q the number of *trials*.

4 Balance-based Analysis of the Birthday attack

The attack is presented in Figure 1. (Note that it picks the points $x_1, \ldots, x_q$ independently at random, rather than picking them at random subject to being distinct as in some variants of the attack [8]. The difference in performance is negligible as long as the domain is larger than the range.)

We are interested in two quantities: the probability C of finding a collision in a given number q of trials, and the *threshold* Q, defined as the expected number of trials to get a collision. Both will be estimated in terms of the balance of the hash function being attacked. Note that although Q is a simpler metric it is less informative than C since the latter shows how the success-rate of the attack grows with the number of trials. We begin with Theorem 1 below, which gives both upper and lower bounds on C that are within constant factors of each other. The proof of Theorem 1 is in Section 4.1 below.

Theorem 1. *Let $h\colon D \to R$ be a hash function. Let $d = |D|$ and $r = |R|$ and assume $d > r \geq 2$. Let C denote the probability of finding a collision for h in $q \geq 2$ trials of the birthday attack of Figure 1. Let $\mu(h)$ be the balance of h as per Definition 1. Then*

$$C \;\leq\; \binom{q}{2} \cdot \left[\frac{1}{r^{\mu(h)}} - \frac{1}{d}\right] . \tag{5}$$

Additionally, if α is any real number, we have

$$\left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot \binom{q}{2} \cdot \left[\frac{1}{r^{\mu(h)}} - \frac{1}{d}\right] \;\leq\; C \tag{6}$$

under the assumption that

$$q \;\leq\; \alpha \cdot \left(1 - \frac{r}{d}\right) \cdot r^{\mu(h)/2} . \qquad \blacksquare \tag{7}$$

The above may be a bit hard to interpret. The following, which simply picks a particular value for the parameter α and applies the above, may be easier to understand. It provides upper and lower bounds on C that are within a factor of four of each other assuming $q = O(r^{\mu(h)/2})$. The proof of Corollary 1 is in [1].

Corollary 1. *Let h: $D \to R$ be a hash function. Let $d = |D|$ and $r = |R|$ and assume $d \geq 2r \geq 4$. Let C denote the probability of finding a collision for h in $q \geq 2$ trials of the birthday attack of Figure 1. Let $\mu(h)$ be the balance of h as per Definition 1. Then*

$$C \;\leq\; \binom{q}{2} \cdot \frac{1}{r^{\mu(h)}} \,. \tag{8}$$

Additionally,

$$\frac{1}{4} \cdot \binom{q}{2} \cdot \frac{1}{r^{\mu(h)}} \;\leq\; C \tag{9}$$

under the assumption that $q \leq (1/5) \cdot r^{\mu(h)/2}$. ▮

As we mentioned before, we believe it is important to have close upper and lower bounds rather than approximate equalities when it comes to computing the success rate of attacks since we are making very specific choices of parameters, such as hash function output lengths, based on these estimates, and if our estimates of the success rates are not specific too we might choose parameters incorrectly.

Remark 1. The lower bound in Equation (9) is only valid when $2 \leq q \leq (1/5) \cdot r^{\mu(h)/2}$. The upper bound on q here is not particularly restrictive since we know that as q approaches $r^{\mu(h)/2}$, the probability C gets close to 1. However, note that we are implicitly assuming $2 \leq (1/5) \cdot r^{\mu(h)/2}$, meaning we are assuming a lower bound on $\mu(h)$. However the result only excludes functions of tiny balance. ▮

Next, we show that the threshold is $\Theta(r^{\mu(h)/2})$. Again, we provide explicit upper and lower bounds that are within a constant factor of each other. The proof of Theorem 2 is in Section 4.2.

Theorem 2. *Let h: $D \to R$ be a hash function. Let $d = |D|$ and $r = |R|$ and assume $d \geq 2r \geq 4$. Let Q denote the threshold, meaning the expected number of trials, in the birthday attack of Figure 1, to get a collision. Let $\mu(h)$ be the balance of h as per Definition 1 and assume $((\sqrt{7} - 2)/3) \cdot r^{\mu(h)/2} \geq 2$. Then*

$$(1/2) \cdot r^{\mu(h)/2} \;\leq\; Q \;\leq\; 72 \cdot r^{\mu(h)/2} \quad ▮ \tag{10}$$

Designers of hash functions often have as target to make the hash function have "random" behavior. We now state a result which will enable us to gage how well random functions fare against the birthday attack. (Consequences are discussed after the statement). Proposition 2 below says that if h is chosen at random then the expectation of $r^{-\mu(h)}$ is more than $1/r$ (what it would be for a regular function) by a factor equal to about $1 + r/d$. The proof of Proposition 2 is in the full version of this paper [1].

Proposition 2. *Let D, R be sets of sizes d, r respectively, where $d \geq r \geq 2$. If we choose a function h: $D \to R$ at random then*

$$\mathbf{E}\left[r^{-\mu(h)}\right] \;=\; \frac{1}{r} \cdot \left(1 + \frac{r-1}{d}\right) \,. \quad ▮$$

As an example, suppose $d = 2r$. Then the above implies that if h is chosen at random then

$$\mathbf{E}\left[r^{-\mu(h)}\right] \approx \frac{3}{2} \cdot \frac{1}{r} .$$

As per Theorem 1 and Theorem 2 this means that if h is chosen at random then the probability of finding a collision in q trials is expected to rise to about $3/2$ times what it would be for a regular function, while the threshold is expected to fall to about $\sqrt{2/3}$ times what it would be for a regular function. Although the difference in the efficacy of birthday attacks against regular and random functions becomes less as d/r increases, the above example with $d = 2r$ suggests that although hash functions are often designed to be "random", in terms of resistance to birthday attacks a more desirable goal is to have randomness subject to regularity. This also applies to all restrictions of the hash function to domains from which an adversary may draw during a birthday attack (eg. SHA-1 restricted to 161-bit inputs).

4.1 Proof of Theorem 1

We let $[q]_2$ denote the set of all two-element subsets of $[q]$. Recall that the attack picks $x_1, \ldots, x_q$ at random from the domain D of the hash function. We associated to any two-element set $I = \{i, j\} \in [q]_2$ the random variable X_I which takes value 1 if x_i, x_j form a collision (meaning $x_i \neq x_j$ and $h(x_i) = h(x_j)$), and 0 otherwise. We let

$$X \;=\; \textstyle\sum_{I \in [q]_2} X_I \; .$$

The random variable X is the number of collisions. (We clarify that in this manner of counting the number of collisions, if n distinct points have the same hash value, they contribute $n(n-1)/2$ toward the value of X.) For any $I \in [q]_2$ we have

$$\mathbf{E}\left[X_I\right] \;=\; \Pr\left[\, X_I = 1 \,\right] \;=\; \sum_{i=1}^{r} \frac{d_i(d_i - 1)}{d^2} \;=\; \sum_{i=1}^{r} \frac{d_i^2}{d^2} - \sum_{i=1}^{r} \frac{d_i}{d^2} \;=\; r^{-\mu(h)} - \frac{1}{d} \; . \tag{11}$$

By linearity of expectation we have

$$\mathbf{E}\left[X\right] \;=\; \sum_{I \in [q]_2} \mathbf{E}\left[X_I\right] \;=\; \binom{q}{2} \cdot \left[r^{-\mu(h)} - \frac{1}{d}\right] \; . \tag{12}$$

Let

$$p \;=\; r^{-\mu(h)} - \frac{1}{d} \; .$$

The upper bound of Theorem 1 is a simple application of Markov's inequality and Equation (12):

$$\Pr\left[\, C \,\right] \;=\; \Pr\left[\, X \geq 1 \,\right] \;\leq\; \frac{\mathbf{E}\left[X\right]}{1} \;=\; \binom{q}{2} \cdot p \; .$$

We proceed to the lower bound. Let $[q]_{2,2}$ denote the set of all two-elements subsets of $[q]_2$. Via the inclusion-exclusion principle we have

$$\begin{aligned}\Pr[\,C\,] &= \Pr\left[\bigvee_{I\in[q]_2} X_I = 1\right] \\ &\geq \sum_{I\in[q]_2} \Pr[\,X_I = 1\,] \;-\; \sum_{\{I,J\}\in[q]_{2,2}} \Pr[\,X_I = 1 \wedge X_J = 1\,]\,. \qquad (13)\end{aligned}$$

Equation (12) tells us that the first sum above is

$$\sum_{I\in[q]_2} \Pr[\,X_I = 1\,] \;=\; \sum_{I\in[q]_2} \mathbf{E}[X_I] \;=\; \mathbf{E}[X] \;=\; \binom{q}{2}\cdot p\,. \qquad (14)$$

We now claim that

$$\sum_{\{I,J\}\in[q]_{2,2}} \Pr[\,X_I = 1 \wedge X_J = 1\,] \;\leq\; \left(\frac{\alpha^2}{4}+\alpha\right)\cdot\binom{q}{2}\cdot p\,. \qquad (15)$$

This completes the proof because from Equations (13), (14) and (15) we obtain Equation (6) as follows:

$$\begin{aligned}\Pr[\,C\,] &\geq \binom{q}{2}\cdot p \;-\; \sum_{\{I,J\}\in[q]_{2,2}} \Pr[\,X_I = 1 \wedge X_J = 1\,] \\ &\geq \binom{q}{2}\cdot p \;-\; \left(\frac{\alpha^2}{4}+\alpha\right)\cdot\binom{q}{2}\cdot p \\ &= \left(1-\frac{\alpha^2}{4}-\alpha\right)\cdot\binom{q}{2}\cdot p\,.\end{aligned}$$

It remains to prove Equation (15).

Let E be the set of all $\{I,J\}\in[q]_{2,2}$ such that $I\cap J=\emptyset$, and let N be the set of all $\{I,J\}\in[q]_{2,2}$ such that $I\cap J\neq\emptyset$. Then

$$\begin{aligned}&\sum_{\{I,J\}\in[q]_{2,2}} \Pr[\,X_I = 1 \wedge X_J = 1\,] \\ &= \underbrace{\sum_{\{I,J\}\in E} \Pr[\,X_I = 1 \wedge X_J = 1\,]}_{S_E} + \underbrace{\sum_{\{I,J\}\in N} \Pr[\,X_I = 1 \wedge X_J = 1\,]}_{S_N}\,. \qquad (16)\end{aligned}$$

We now claim that

$$S_E \leq \binom{q}{2}\cdot\frac{1}{4}\cdot\alpha^2\cdot p \qquad (17)$$

$$S_N \leq \binom{q}{2}\cdot\alpha\cdot p\,, \qquad (18)$$

Equation (15) follows from Equations (16), (17) and (18). We now prove Equations (17) and (18).

To upper bound S_E, we note that if $\{I, J\} \in E$ then the random variables X_I and X_J are independent. Using Equation (11) we get

$$\begin{aligned} S_E &= \sum_{\{I,J\} \in E} \Pr[\, X_I = 1 \wedge X_J = 1\,] \\ &= \sum_{\{I,J\} \in E} \Pr[\, X_I = 1\,] \cdot \Pr[\, X_J = 1\,] \;=\; |E| \cdot p^2 \;. \end{aligned}$$

Computing the size of the set E and simplifying, we get

$$S_E \;=\; \frac{1}{2}\binom{q}{2}\binom{q-2}{2} \cdot p^2 \;=\; \binom{q}{2} \cdot p \cdot \frac{q^2 - 5q + 6}{4} \cdot p \;.$$

We now upper bound this as follows:

$$S_E \;<\; \binom{q}{2} \cdot p \cdot q^2 \cdot \frac{p}{4} \;\le\; \binom{q}{2} \cdot p \cdot \alpha^2 \cdot r^{\mu(h)} \cdot \frac{p}{4} \;\le\; \frac{1}{4} \cdot \alpha^2 \cdot \binom{q}{2} \cdot p \;.$$

Above the first inequality is true because Theorem 1 assumes $q \ge 2$. The second inequality is true because of the assumption made in Equation (7). The third inequality is true because $r^{\mu(h)} \cdot p < 1$. We have now obtained Equation (17).

The remaining task is to upper bound S_N. The difficulty here is that for $\{I, J\} \in N$ the random variables X_I and X_J are not independent. We let $d_i = |h^{-1}(R_i)|$ for $i \in [r]$ where $R = \{R_1, \ldots, R_r\}$ is the range of the hash function. If $\{I, J\} \in N$ then the two-elements sets I and J intersect in exactly one point. (They cannot be equal since I, J are assumed distinct.) Accordingly we have

$$\begin{aligned} S_N &= \sum_{\{I,J\} \in N} \Pr[\, X_I = 1 \wedge X_J = 1\,] \\ &= |N| \cdot \sum_{i=1}^{r} \frac{d_i(d_i - 1)^2}{d^3} \\ &< \frac{|N|}{d^3} \cdot \sum_{i=1}^{r} d_i^3 \;. \end{aligned} \tag{19}$$

We now compute the size of the set N:

$$\begin{aligned} |N| &= \frac{1}{2}\binom{q}{2}\binom{q}{2} - \frac{1}{2}\binom{q}{2} - \frac{1}{2}\binom{q}{2}\binom{q-2}{2} \\ &= \binom{q}{2} \cdot (q-2) \;. \end{aligned}$$

Putting this together with Equation (19) we have

$$S_N \;<\; \binom{q}{2} \cdot q \cdot \left[\frac{1}{d^3} \cdot \textstyle\sum_{i=1}^{r} d_i^3\right] \;. \tag{20}$$

To upper bound the sum of Equation (20), we view $d_1, \ldots, d_r$ as variables and consider the problem of maximizing $d_1^3 + \cdots + d_r^3$ subject to the constraint

$\sum_{i=1}^{r} d_i^2 \;=\; d^2 \cdot r^{-\mu(h)}$. The maximum occurs when $d_1 = d \cdot r^{-\mu(h)/2}$ and $d_i = 0$ for $i = 2, \ldots, r$, meaning that

$$\sum_{i=1}^{r} d_i^3 \;\leq\; d^3 r^{-3\mu(h)/2} \; .$$

Returning to Equation (20) with this information we get

$$S_N \;<\; \binom{q}{2} \cdot q \cdot \left[\frac{1}{d^3} \cdot \sum_{i=1}^{r} d_i^3\right] \;\leq\; \binom{q}{2} \cdot q \cdot \frac{1}{d^3} \cdot d^3 r^{-3\mu(h)/2} \;=\; \binom{q}{2} \cdot q \cdot r^{-3\mu(h)/2} \; .$$

We now use the assumption made in Equation (7), and finally use Lemma 1, to get

$$\begin{aligned} S_N &< \binom{q}{2} \cdot \alpha \cdot \left(1 - \frac{r}{d}\right) \cdot r^{\mu(h)/2} \cdot r^{-3\mu(h)/2} \\ &\leq \binom{q}{2} \cdot \alpha \cdot \left(1 - \frac{r}{d}\right) r^{-\mu(h)} \;\leq\; \binom{q}{2} \cdot \alpha \cdot p \; . \end{aligned}$$

This proves Equation (18) and thus concludes the proof of Theorem 1.

4.2 Proof of Theorem 2

We begin by proving the lower bound. Let the random variable Y denote the number of trials to collision. Let $C(q)$ denote the probability of finding a collision for h in $q \geq 2$ trials of the birthday attack in Figure 1, and let $D(q)$ denote the probability of finding the first collision on the q-th trial. Let $Q = r^{\mu(h)/2}$. From the definition of Y:

$$\mathbf{E}[Y] \;=\; \sum_{x=1}^{\infty} x \cdot D(x) \;\geq\; Q \cdot \sum_{x=Q}^{\infty} D(x) \;=\; Q \cdot (1 - C(Q-1)) \; .$$

We claim that

$$C(Q-1) < \frac{1}{2} \; . \tag{21}$$

It follows that

$$\mathbf{E}[Y] \;\geq\; Q \cdot (1/2) \;\geq\; (1/2) \cdot r^{\mu(h)/2} \; ,$$

as desired. We now justify Equation (21). From Equation (8) of Corollary 1 we know that

$$C(Q-1) \;\leq\; \binom{Q-1}{2} \cdot \frac{1}{r^{\mu(h)}} \;=\; \frac{1}{2} \cdot \Big((Q-1)^2 - (Q-1)\Big) \cdot \frac{1}{r^{\mu(h)}} \; .$$

Since $Q = r^{\mu(h)/2} \geq 2$ by assumption,

$$(Q-1)^2 - (Q-1) \;=\; Q^2 - 3 \cdot Q + 2 \;<\; Q^2 \;=\; r^{\mu(h)}$$

and

$$C(Q-1) \;<\; \frac{1}{2} \cdot r^{\mu(h)} \cdot \frac{1}{r^{\mu(h)}} \;=\; \frac{1}{2}$$

as desired

For the upper bound, we must be careful since there is an upper restrictions on q in Equation (9) and Equation (6). Fix $\alpha = (2\sqrt{7}-4)/3$ and $q = (\alpha/2) \cdot r^{\mu(h)/2}$. First note that

$$q \;=\; \frac{\alpha}{2} \cdot r^{\mu(h)/2} \;\leq\; \alpha \cdot \left(1 - \frac{r}{d}\right) \cdot r^{\mu(h)/2}$$

since we assume that $d \geq 2r$ and therefore that $1 - r/d \geq 1/2$. This means that we can use Theorem 1 with α and q defined as above. Combining Theorem 1 with Lemma 1 and the assumptions that $d \geq 2r$ and $q = (\alpha/2) \cdot r^{\mu(h)/2} \geq 2$, we have

$$\begin{aligned} C(q) &\geq \left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot \binom{q}{2} \cdot \frac{1}{2} \cdot \frac{1}{r^{\mu(h)}} \\ &\geq \left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot q^2 \cdot \frac{1}{8} \cdot \frac{1}{r^{\mu(h)}} \,. \end{aligned}$$

Replacing q with $(\alpha/2) \cdot r^{\mu(h)/2}$ we get

$$\begin{aligned} C(q) \;&\geq\; \left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot \left(\frac{\alpha}{2} \cdot r^{\mu(h)/2}\right)^2 \cdot \frac{1}{8} \cdot \frac{1}{r^{\mu(h)}} \\ &=\; \frac{1}{32} \cdot \left(\alpha^2 - \frac{\alpha^4}{4} - \alpha^3\right) \,. \end{aligned} \tag{22}$$

Now consider the following experiment that repeatedly runs the birthday attack, using $q = (\alpha/2) \cdot r^{\mu(h)/2}$ trials, until a collision is found:

For $j = 1, 2, \ldots$ do
  For $i = 1, \ldots, q$ do
    Pick $x_{q(j-1)+i}$ at random from the domain of h
    $y_{q(j-1)+i} \leftarrow h(x_{q(j-1)+i})$
    If there exists k such that $q(j-1) < k < q(j-1) + i$
    and $y_{q(j-1)+i} = y_k$ but $x_{q(j-1)+i} \neq x_k$ then
      return $x_{q(j-1)+i}, x_k$ // collision found in this block of q trials
    EndIf
  EndFor
EndFor

Let the random variable A denote the number of trials to success in the above experiment. We claim that

$$\mathbf{E}[Y] \;\leq\; \mathbf{E}[A] \tag{23}$$

and

$$\mathbf{E}[A] \;\leq\; \frac{q}{C(q)} \,, \tag{24}$$

and combining with Equation (22), it follows that

$$\mathbf{E}[Y] \;\leq\; \frac{q}{C(q)} \;\leq\; \frac{(\alpha/2) \cdot r^{\mu(h)/2}}{(1/32) \cdot (\alpha^2 - (\alpha^4/4) - \alpha^3)} \;<\; 72 \cdot r^{\mu(h)/2} \,,$$

giving the upper bound in the theorem statement.

To prove Equation (23) it is sufficient to note that, for any random tape T,

$$Y(T) \;\leq\; A(T)$$

since any collision in the above experiment is immediately a collision for the birthday attack in Figure 1.

To prove Equation (24), consider each inner loop of the above experiment an independent Bernoulli trial, and let Z denote the expected number of Bernoulli trials (inner loop executions) to collision. Since each inner loop has a success probability $C(q)$, standard results tell us that

$$\mathbf{E}[Z] \;\leq\; \frac{1}{C(q)}\;. \tag{25}$$

Let $F(i)$ denote the probability that the first collision in the above experiment occurs on the i-th trial. Let $G(j)$ denote the probability that the first collision is found in the j-th execution of the inner loop in the above experiment. Then

$$\begin{aligned}
\mathbf{E}[A] &= \sum_{i=1}^{\infty} i \cdot F(i) \\
&= \sum_{j=1}^{\infty}\sum_{i=1}^{q} (q\cdot(j-1)+i)\cdot F\,(q\cdot(j-1)+i) \\
&\leq q \cdot \sum_{j=1}^{\infty}\left(j \cdot \sum_{i=1}^{q} F\,(q\cdot(j-1)+i)\right)
\end{aligned}$$

Since, by the definition of $G(j)$, for any $j \geq 1$

$$\sum_{i=1}^{q} F\,(q\cdot(j-1)+i) \;=\; G(j)\;,$$

it follows that

$$\mathbf{E}[A] \;\leq\; q\cdot\sum_{j=1}^{\infty} j\cdot G(j) \;=\; q\cdot\mathbf{E}[Z]\;. \tag{26}$$

Combining Equation (25) with Equation (26) yields Equation (24), completing the proof.

5 Does the MD transform preserve balance?

We consider the following popular paradigm for the construction of hash functions. First build a *compression function* $H\colon\ \{0,1\}^{b+c} \to \{0,1\}^c$, where $b \geq 1$ is called the *block-length* and $c \geq 1$ is called the *chaining-length*. Then transform H into a hash function $\overline{H}\colon\ D_b \to \{0,1\}^c$, where

$$D_b \;=\; \{\, M \in \{0,1\}^* \;:\; |M| = nb \text{ for some } 1 \leq n < 2^b \,\}\;,$$

Function $\overline{H}(M)$
Break M into b-bit blocks $M_1 \| \cdots \| M_n$
$M_{n+1} \leftarrow \langle n \rangle_b$; $C_0 \leftarrow 0^c$
For $i = 1, \ldots, n+1$ do $C_i \leftarrow H(M_i \| C_{i-1})$ EndFor
Return C_{n+1}

Fig. 2. Hash function $\overline{H}$: $D_b \rightarrow \{0,1\}^c$ obtained via the MD transform applied to compression function H: $\{0,1\}^{b+c} \rightarrow \{0,1\}^c$.

---

via the Merkle-Damgård (MD) [5, 2] transform depicted in Figure 2. (In this description and below, we let $\langle i \rangle_b$ denote the representation of integer i as a string of length *exactly* b bits for $i = 0, \ldots, 2^b - 1$.) In particular, modulo details, this is the paradigm used in the design of popular hash functions including MD5 [7], SHA-1 [6] and RIPEMD-160 [3].

For the considerations in this section, we will focus on the restriction of $\overline{H}$ to strings of some particular length. For any integer $1 \leq n < 2^b$ (the number of blocks) we let $\overline{H}_n$: $D_{b,n} \rightarrow \{0,1\}^c$ denote the restriction of $\overline{H}$ to the domain $D_{b,n}$, defined as the set of all strings in D_b that have length exactly nb bits.

Our results lead us to desire that $\overline{H}_n$ has high balance for all practical values of n. Designers could certainly try to ensure that the compression function is regular or has high balance, but to be assured that $\overline{H}_n$ has high balance it would need to be the case that the MD transform is "balance preserving." Unfortunately, the following shows that this is not true. It presents an example of a compression function H which has high balance (in fact is regular, with balance one) but $\overline{H}_n$ has low balance (in fact, balance zero) even for $n = 2$.

Proposition 3. *Let b, c be positive integers. There exists a compression function H: $\{0,1\}^{b+c} \rightarrow \{0,1\}^c$ such that H is regular ($\mu(H) = 1$) but $\overline{H}_2$ is a constant function ($\mu(\overline{H}_2) = 0$).* ▮

Proof (Proposition 3). Let H: $\{0,1\}^{b+c} \rightarrow \{0,1\}^c$ map $B \| C$ to C for all b-bit strings B and c-bit strings C. Clearly $\mu(H) = 1$ since each point in $\{0,1\}^c$ has exactly 2^b pre-images under H. Because the initial vector (IV) in the MD transform is the constant $C_0 = 0^c$, and by the definition of H, the function $\overline{H}_2$ maps all inputs to 0^c. □

This example might be viewed as contrived particularly because the compression function H above is not collision-resistant (although it is very resistant to birthday attacks), but in fact it still serves to illustrate an important point. The popularity of the MD paradigm arises from the fact that it *provably* preserves collision-resistance [5, 2]. However, the above shows that it does not provably preserve balance. Even though Proposition 3 does not say that the transform will *always* be poor at preserving balance, it says that we cannot count on the transform to preserve balance in general. This means that simply ensuring high balance of the compression function is not a suitable general design principle.

(We also remark that there exist adversaries capable of finding collisions for any unkeyed compression function, including the compression functions in MD5, SHA-1, and RIPEMD-160, using exactly two trials. We just do not know what these adversaries are.)

Is there any other design principle whereby some properties of the compression function suffice to ensure high balance of the hash function? Toward finding one we note that the behavior exhibited by the function $\overline{H}_2$ in the proof of Proposition 3 arose because the initial vector (IV) of the MD transform was $C_0 = 0^c$, and although H was regular, the restriction of H to inputs having the last c bits 0 was not regular, and in fact was constant. Accordingly we consider requiring regularity conditions not just on the compression function but on certain related functions as well. If $H\colon \{0,1\}^{b+c} \to \{0,1\}^c$ then define $H_0\colon \{0,1\}^b \to \{0,1\}^c$ via $M \mapsto H(M\|0^c)$ for all $M \in \{0,1\}^b$, and for $n \geq 1$ define $H_n\colon \{0,1\}^c \to \{0,1\}^c$ via $M \mapsto H(\langle n\rangle_b\|M)$ for all $M \in \{0,1\}^c$. The following shows that if H, H_0, H_n are all regular, meaning have balance one, then $\overline{H}_n$ is also regular.

Proposition 4. *Let b, c, n be positive integers. Let $H\colon \{0,1\}^{b+c} \to \{0,1\}^c$ and let H_0, H_n be as above. Assume H, H_0, and H_n are all regular. Then $\overline{H}_n$ is regular.* ▮

Proof (Proposition 4). The computation of $\overline{H}_n$ can be written as

Function $\overline{H}_n(M)$
  Break M into b-bit blocks $M_1\|\cdots\|M_n$; $C_1 \leftarrow H_0(M_1)$
  For $i = 2, \ldots, n$ do $C_i \leftarrow H(M_i\|C_{i-1})$ EndFor
  $C_{n+1} \leftarrow H_n(C_n)$; Return C_{n+1}

It is not hard to check that the assumed regularity of H_0, H and H_n imply the regularity of $\overline{H}_n$. □

Unfortunately Proposition 4 is not "robust." Although $\overline{H}_n$ has balance one if H, H_0, H_n have balance one, it turns out that if H, H_0, H_n have balance that is high but not quite one, we are *not* assured that $\overline{H}_n$ has high balance. Proposition 5 shows that even a slight deviation from the maximum balance of one in H, H_0, H_n can be amplified, and result in $\overline{H}_n$ having very low balance. The proof of the following is in the full version of this paper [1].

Proposition 5. *Let b, c be integers, $b \geq c \geq 2$, and let $n \geq c$. Then there exists a compression function $H\colon \{0,1\}^{b+c} \to \{0,1\}^c$ such that $\mu(H) \geq 1 - 1/c$, $\mu(H_0) = 1$, and $\mu(H_n) \geq 1 - 2/c$, but $\mu(\overline{H}_n) \leq 1/c$, where the functions H_0, H_n are defined as above.* ▮

As indicated by Proposition 2, a random compression function will have expected balance that is high but not quite 1. We expect that practical compression functions are in the same boat. Furthermore it seems harder to build compression functions that have balance exactly one than close to one. So the lack of robustness of Proposition 4, as exhibited by Proposition 5, means that Proposition 4 is of limited use.

The consequence of the results in this section is that we are unable to recommend any design principle that, to ensure high balance, focuses solely on establishing properties of the compression function. It seems one is forced to look directly at the hash function. We endeavor next to do this for SHA-1.

6 Experiments on SHA-1

Let SHA_n: $\{0,1\}^n \to \{0,1\}^{160}$ denote the restriction of SHA-1 to inputs of length $n < 2^{64}$. Because SHA-1's range is $\{0,1\}^{160}$, it is commonly believed that the expected number of trials necessary to find a collision for SHA_n is approximately 2^{80}. As Theorem 2 shows, however, this is only true if the balance of SHA_n is one or close to one for all practical values of n. If the balance is not close to one, then we expect to be able to find collisions using less work. It therefore seems desirable to calculate (or approximate) the balance of SHA_n for reasonable values of n (eg. $n = 256$). A direct computation of $\mu(\mathrm{SHA}_n)$ based on Definition 1 is however infeasible given the size of the domain and range of SHA_n. Accordingly we focus on a more achievable goal. We look at properties of SHA_n that one can reasonably test and whose absence might indicate that SHA_n does not have high balance. Our experiments are not meant to be exhaustive, but rather representative of the types of feasible experiments one can perform with SHA-1.

Let $\mathrm{SHA}_{n;t_1\ldots t_2}$: $\{0,1\}^n \to \{0,1\}^{t_2-t_1+1}$ denote the function that returns the t_1-th through t_2-th output bits of SHA_n. We ask what exactly is the balance of $\mathrm{SHA}_{32;t_1\ldots t_2}$ when $t_2 - t_1 + 1 \in \{8, 16, 24\}$. And we ask whether the functions $\mathrm{SHA}_{m;t_1\ldots t_2}$, $m \in \{160, 256, 1024, 2048\}$, appear regular when $t_2 - t_1 + 1 \in \{8, 16, 24\}$. (Note that SHA_{256} is SHA-1 restricted to the domain $\{0,1\}^{256}$, not NIST's SHA-256 hash algorithm.)

Balance of $\mathrm{SHA}_{32;t_1\ldots t_2}$. We calculate the balance of $\mathrm{SHA}_{32;t_1\ldots t_2}$ for all pairs t_1, t_2 such that $t_2 - t_1 + 1 \in \{8, 16, 24\}$ and t_1 begins on a byte boundary (ie. we look at all 1-, 2-, and 3-byte portions of the SHA-1 output). The calculated values of $\mu(\mathrm{SHA}_{32;t_1\ldots t_2})$ appear in the full version of this paper [1]. Characteristic values are $\mu(\mathrm{SHA}_{32;1\ldots 8}) = 0.99999998893$, $\mu(\mathrm{SHA}_{32;1\ldots 16}) = 0.999998623$ and $\mu(\mathrm{SHA}_{32;1\ldots 24}) = 0.99976567$, indicating that, for the specified values of t_1, t_2, the balance of $\mathrm{SHA}_{32;t_1\ldots t_2}$ is high.

These results do not imply that the functions $\mathrm{SHA}_{n;t_1\ldots t_2}$ or SHA_n, $n > 32$ and t_1, t_2 as before, are regular. But it is encouraging that $\mu(\mathrm{SHA}_{32;t_1\ldots t_2})$ are high, and in fact very close to what one would expect from a random function (cf. Proposition 2), since a small value for $\mu(\mathrm{SHA}_{32;t_1\ldots t_2})$ for any of the specified t_1, t_2 pairs might indicate some unusual property of the SHA-1 hash function.

Experiments on SHA_{160}, SHA_{256}, SHA_{1024}, and SHA_{2048}. Let $n \in \{160, 256, 1024, 2048\}$. Although we cannot calculate the balance of SHA_n, we can compare the behavior of $\mathrm{SHA}_{n;t_1\ldots t_2}$, $t_2 - t_1 + 1 \in \{8, 16, 24\}$, on random inputs to what one would expect from a regular or random function. There are several possible approaches to take. Knowing that the balance of $\mathrm{SHA}_{n;t_1\ldots t_2}$ directly affects the expected number of trials to collision, the approach we take is to compute the average, over 10000 runs, of the number of trials to collision in a birthday attack against $\mathrm{SHA}_{n;t_1 \ldots t_2}$.

If the average number of trials to collision against $\mathrm{SHA}_{n;t_1 \ldots t_2}$ on random bits is approximately the same as what one would expect from a regular function, it would support the view that SHA_n has high balance. However, a significant difference between the results for $\mathrm{SHA}_{n;t_1 \ldots t_2}$ on random inputs and what one would expect from a regular function might indicate some unusual behavior with SHA-1, and this unusual behavior would deserve further investigation. Our experimental results are consistent with SHA_n having high balance. However, we again point out that these tests were only designed to uncover gross anomalies and are not exhaustive. Details are in [1].

Acknowledgments

Mihir Bellare is supported in part by NSF grants CCR-0098123, ANR-0129617 and CCR-0208842, and by an IBM Faculty Partnership Development Award. Tadayoshi Kohno is supported by a National Defense Science and Engineering Graduate Fellowship.

References

1. M. Bellare and T. Kohno. Hash function balance and its impact on birthday attacks. IACR ePrint archive, `http://eprint.iacr.org/2003/065/`. Full version of this paper
2. I. Damgård. A design principle for hash functions. *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science Vol. 435, G. Brassard ed., Springer-Verlag, 1989.
3. H. Dobbertin, A. Bosselaers and B. Preneel. RIPEMD-160, a strengthened version of RIPEMD. *Fast Software Encryption '96*, Lecture Notes in Computer Science Vol. 1039, D. Gollmann ed., Springer-Verlag, 1996.
4. A. Menezes, P. van Oorschot and S. Vanstone. Handbook of applied cryptography. CRC Press, 1997.
5. R. Merkle. One way hash functions and DES. *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science Vol. 435, G. Brassard ed., Springer-Verlag, 1989.
6. National Institute of Standards. FIPS 180-2, Secure hash standard. August 1, 2000.
7. R. Rivest. The MD5 message-digest algorithm. IETF RFC 1321, April 1992.
8. D. Stinson. Cryptography theory and practice, 1st Edition. CRC Press, 1995.
9. P. van Oorschot and M. Wiener. Parallel collision search with cryptanalytic applications, *Journal of Cryptology* 12(1), Jan 1999, 1–28.
10. G. Yuval. How to swindle Rabin. *Cryptologia* (3), 1979, 187–190.