An extended abstract of this paper appears in *Advances in Cryptology – EUROCRYPT '04*, Lecture Notes in Computer Science Vol. 3027, C. Cachin and J. Camenisch eds., Springer-Verlag, 2004. This is the full version.

Hash Function Balance and its Impact on Birthday Attacks

Mihir Bellare* Tadayoshi Kohno†

May 2004

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 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 as a function of the balance of the hash function being attacked. In particular, we will see that the number of trials to find a collision 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.

Keywords: Hash functions, birthday attacks, collision-resistance.

*Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-Mail: `mihir@cs.ucsd.edu`. URL: `http://www-cse.ucsd.edu/users/mihir`. Supported in part by NSF grants CCR-0098123, ANR-0129617 and CCR-0208842, and by an IBM Faculty Partnership Development Award.

†Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-mail: `tkohno@cs.ucsd.edu`. URL: `http://www-cse.ucsd.edu/users/tkohno`. Supported by a National Defense Science and Engineering Graduate Fellowship.

Contents

1 Introduction

Let h: $D \to R$ be a function, where the domain D and range R are finite sets. We say that h is a *hash function* if $|D| > |R|$. A *collision* for h is a pair x, y of distinct points in D for which $h(x) = h(y)$. Cryptographers are interested in hash functions that are collision-resistant, meaning it is computationally infeasible to find a collision. (Such functions have numerous uses, for example for digital signatures.) The best known general collision-finding attack is the so-called birthday attack. In this paper, we assess the performance and success rate of this attack.

The birthday attack. 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 is a pair i, j such that x_i, x_j form a collision for h. 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. [9, 15, 18, 20]). The one we consider is that they are chosen independently at random from D.

Picking random points from the domain may be prohibitive in the case of a hash function like SHA-1 [11] whose domain is the set of all strings of length at most 2^{64}. In such cases we would simply attack the function $h = \mathrm{SHA}_n$: $\{0,1\}^n \to \{0,1\}^{160}$, the restriction of SHA-1 to inputs of length $n < 2^{64}$, for some suitable value of n, say $n = 161$. This works because a collision for SHA_n is also a collision for SHA-1. This is to be understood henecforth in discussing attacks on hash functions like SHA-1.

We let $C_h(q)$ be the probability that the birthday attack on hash function h: $D \to R$ succeeds in finding a collision in q trials. We are interested in how this function grows with q.

Current beliefs. It is generally stated that

$$C_h(q) \approx \binom{q}{2} \cdot \frac{1}{r} , \tag{1}$$

where $r = |R|$ is the size of the range of h and $q \le O(\sqrt{r})$. This implies that a collision is expected in about $r^{1/2}$ trials. In particular, it predicts that collisions in a hash function with output length m bits would take about $2^{m/2}$ trials to find. This estimate is the basis for the choice of output length m, which is typically made just large enough to make $2^{m/2}$ trials infeasible.

How is Equation (1) obtained? The apparent reasoning is to view the range points $y_1, \ldots, y_q$ computed in the attack as being uniformly and independently distributed in R. Then the standard birthday phenomenon says that the probability that there exist distinct i, j such that $y_i = y_j$ is, up to constant factors, $\binom{q}{2}/r$.

However, this argument is actually not correct, because the point $h(x)$, for x drawn at random from D, is not necessarily uniformly distributed in R. Rather, the probability that $h(x)$ equals a particular range point y is $|h^{-1}(y)|/|D|$, where $h^{-1}(y)$ is the set of all pre-images of y under h. So the range points computed in the attack are uniformly distributed over R if and only if h is *regular*, meaning every range point has the same number of pre-images under h.[1]

If h is not regular, then $C_h(q)$ is actually *larger* than $\binom{q}{2}/r$, meaning that the attack would find a collision in *fewer* than the expected $r^{1/2}$ trials. But how much fewer? Is there cause for concern?

Random functions. Before proceeding we note that there is another way that the perfomance of the birthday attack is often assessed, which is to consider the probability $C^{\$}_{D,R}(q)$ that the birthday attack succeeds in q trials in the thought experiment where h is chosen at random from the set of all maps of D to R. One can show (cf. [7, 16] and Theorem 7.3) that $C^{\$}_{D,R}(q) \approx \binom{q}{2}/r$.

Now one might argue that a given, "good" hash function h has "random behavior" and hence

[1] We refer the reader to Section 4 for more details.

$C_h(q)$ is also about $\binom{q}{2}/r$ for such a h. This argument is however not mathematically sound. There is no sound definition of what it means to "have random behavior" and it is unclear a suitable one can be found. We end up not analyzing the given h, but rather analyzing an abstract and ideal object that ultimately has no connection to h, regardless of the design principles underlying h. Indeed, the analysis of the attack ignores the actual hash function entirely: whether it be MD5, SHA-1, RIPEMD-160 or some other function, there is no change in the analysis, for the latter looks at a random function.[2]

Our goal is to obtain results enabling is to assess $C_h(q)$ for any given, specific h, rather than one chosen at random. Still, the random hash function model is something to keep in mind and we will return to it, in particular contrasting results there with our own, at several points in this paper.

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 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 5.2 says that the maximum balance of a hash function is 1 and is achieved when h is regular (meaning $d_i = d/r$ for all i), while the minimum balance is 0 and 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.

RESULTS. Theorem 6.1 and Corollary 6.2 say that, up to constant factors,[3]

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

Thus, a collision is expected in about $r^{\mu(h)/2}$ trials. This indicates that the performance of the birthday attack can be characterized, quite simply and accurately, via the balance of the hash function h being attacked.

Note that when $\mu(h) = 1$ (meaning, h is regular) then Equation (2) agrees with Equation (1). At the other extreme, when $\mu(h) = 0$, meaning h is a constant function, the attack finds collisions in $O(1)$ trials. The value of the general result of Equation (2) is that it shows the full spectrum in between the extremes of regular and constant functions. For example the birthday attack on a

[2] We clarify that we are not critiquing the (heuristic) design principle of attempting to make hash functions have random behavior, for this is crucial to obtain protection against cryptanalytic attacks. We are, rather, critiquing the value of the random hash function model as a way to assess security against attacks.

[3] This assumes $d \geq 2r$ and $q \leq O(r^{\mu(h)/2})$.

hash function of balance $\mu(h) = 1/2$ will find a collision in about $r^{1/4}$ trials, which is significantly less than $r^{1/2}$.

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

Above we said the output-length of the hash function should be "about" $160/\mu(h)$ bits. The inexactitude here arises from the hidden constants underlying Equation (2), and leads us to another point. Our results provide both upper and lower bounds on $C_h(q)$ that are tight in the sense of being within a constant factor of each other. We claim that it is important to have both upper and lower bounds, and the closer to each other the better, for this enables us to make calcluations like the one we just did with precision, choosing output lengths for hash functions large enough to prevent attacks without incurring costs from being larger than strictly necessary.

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.)

Random hash functions versus regular ones. It is interesting to note that from the point of view of security against birthday attacks, regular functions actually fare (slightly) better than random ones. This is illustrated by Proposition 7.4 which says that if h: $D \to R$ is regular and $d \geq 2r$ then $C^{\$}_{D,R}(q) > (8/5) \cdot C_h(q)$.[4]

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 [12], SHA-1 [11] and RIPEMD-160 [8] are designed by applying the Merkle-Damgård (MD) [10, 6] 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 8.1 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 8.2 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 8.3 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

[4] We clarify that we are not saying that random functions fare poorly against the birthday attack: in fact, as we noted above, $C^{\$}_{D,R}(q) \approx \binom{q}{2}/r$ (cf. [16, 7] and Theorem 7.3). Rather we are noting, as a mathematical curiosity and because it might seem counter-intuitive, that regular functions actually fare even better. This is explained and discussed more in Section 7.

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 11 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 11 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.

What textbooks say. It is informative to briefly survey what textbooks say about the birthday attack on hash functions.

Stinson, in the first edition of his book [15], shows that Equation (1) is true under the assumption that h is regular. There is no information regarding the case where h is not regular.

In the second edition of his book [16], Stinson drops this result in favor of an analysis in the random oracle model [4], showing that $C^{\$}_{D,R}(q) \approx \binom{q}{2}/r$. (Our Theorem 7.3 is a more precise version of this statement, with bounds rather than approximate equalities.) Delfs and Knebl [7] also assume that h is random. The weakness of these results, as noted above, is that an analysis for random functions ultimately provides no real information or guarantees about what happens with a specific real function.

Buchmann's discussion of the attack says: "We assume that strings from the domain can be chosen such that the distribution on the corresponding hash values is the uniform distribution" [5]. Under this assumption he correctly argues Equation (1), but it is unclear how to realize this assumption unless h is regular.

Stallings [14, Section 11.5] says "the strength of a hash function against brute-force attacks depends solely on the length of the hash code produced by the algorithm." This is wrong. (The strength does depend on the length of the hash code, but not solely on this). Schneier [13, Section 7.4] says that to prevent birthday attacks one should choose the output length m large enough that $2^{m/2}$ trials is infeasible, without giving any indication that this is not true in general but only if

For $i = 1, \ldots, q$ do
$x_i \stackrel{\$}{\leftarrow} D$; $y_i \leftarrow h(x_i)$
If $(\exists j : j < i \ \& \ y_i = y_j \ \& \ x_i \neq x_j)$ then return x_i, x_j EndIf
EndFor
Return $\perp$

Figure 1: Birthday attack on a hash function h: $D \to R$. The integer $q \geq 2$ is the number of trials. The attack either returns a collision for h, or the symbol $\perp$ to indicate it did not find one. When the highlighted text is omitted, the attack becomes one to find possibly trivial collisions rather than collisions.

---

the hash function is random or regular. Later [13, Section 18.1] he says: "Most practical one-way hash functions produce 128-bit hashes. This forces anyone attempting the birthday attack to hash 2^{64} random documents to find two that hash to the same value." This is wrong in that the mere fact that the output-length is 128 bits does not force the attacker to use 2^{64} trials. (That is only true if h is regular or random.)

2 Notation and Terminology

We let $\mathbb{N} = \{1, 2, 3, \ldots\}$ denote the set of positive integers. If n is a non-negative integer then we let $[n] = \{1, \ldots, n\}$. If S is a set then $|S|$ denotes its size, and $s \stackrel{\$}{\leftarrow} S$ denotes the operation of picking a random element of S and denoting it by s.

We denote by h: $D \to R$ a function mapping domain D to range R, and throughout the paper we assume that R has size at least two. We usually let $d = |D|$ and $r = |R|$. We say that h is a *hash function* if $d > r$. A *collision* for h is a pair x_1, x_2 of points in D such that $x_1 \neq x_2$ but $h(x_1) = h(x_2)$. A *possibly trivial collision* for h is a pair x_1, x_2 of points in D such that $h(x_1) = h(x_2)$. (That is, unlike for a collision, we do not require $x_1 \neq x_2$). 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 $y \in R$.

3 The attack and associated metrics

The attack we consider is presented in Figure 1. It picks points $x_1, x_2, \ldots, x_q$ independently at random from the domain D of the given hash function h: $D \to R$. If two of these points form a collision for h they are returned, and if no collision is found the attack returns $\perp$. (Variants of the attack, that differ in the way the points $x_1, x_2, \ldots, x_q$ are chosen, are discussed in Section 9.2). We say that the attack was *successful* if it returns a collision. We refer to the integer $q \geq 2$ as the *number of trials*. We are interested in the following quantities associated to h via this attack.

Definition 3.1 Let h: $D \to R$ be a hash function. For any integer $q \geq 2$ we let $C_h(q)$ denote the probability that the birthday attack of Figure 1 succeeds. For any real number c with $0 \leq c < 1$ we let

$$Q_h(c) \ = \ \min \{ \, q \, : \, C_h(q) \geq c \, \}$$

For $i = 1, \ldots, q$ do

$y_i \stackrel{\$}{\leftarrow} R$

If $(\exists j : j < i \ \& \ y_i = y_j)$ then return i, j EndIf

EndFor

Return $\perp$

Figure 2: Experiment to find collisions in a set R by random sampling. The integer $q \geq 2$ is the number of trials. The experiment either returns two distinct indices i, j for which $y_i = y_j$, or $\perp$ to indicate it did not find such indices.

---

denote the minimum number of trials required for the probability of success to exceed c. We refer to C_h as the *collision probability function* of h and to Q_h as the *collision threshold function* of h. ▮

To facilitate some later discussions it is useful to also introduce the following. Consider the birthday attack of Figure 1 *with the highlighted text omitted*, and let $C_h^*(q)$ denote the probability that this attack is successful. (This is the probability that the attack finds a possibly trivial collision.) Then let $Q_h^*(p)$ denote the minimum value of q for which $C_h^*(q) \geq p$.

4 Approaches to the analysis of the birthday attack

Let us consider how the analysis of the birthday attack may be approached. We begin by reviewing some information about the classical birthday problem.

The classical birthday problem. Let R be a set of size r that in our context is the range of the hash function. Figure 2 depicts a simple sampling experiment in which we draw points $y_1, y_2, \ldots, y_q$ uniformly and independently at random from R and declare success if some two of them are equal. Clearly the probability of success in this experiment is the probability of a collision in throwing q balls randomly and independently into r bins. (A collision here means two different balls land in the same bin.) We denote this value by $B_r(q)$, and let $P_r(c)$ denote the minimum value of q for which $B_r(q) \geq c$. It is well-known that

$$B_r(q) \ = \ \Theta(1) \cdot \binom{q}{2} \cdot \frac{1}{r} \qquad \text{and} \qquad P_r(c) \ = \ \Theta\left(\sqrt{rc}\right) \ , \tag{3}$$

assuming $q \leq O(\sqrt{r})$ and some appropriate upper bound on c. For precise bounds, see for example [1].

Relation to hash functions. Conventional wisdom appears to be that $C_h(q) = B_r(q)$ and $Q_h(c) = P_r(c)$ for any $h\colon D \to R$, where $r = |R|$. This conclusion appears to arise by viewing the points $y_1, \ldots, y_q$ computed in the birthday attack of Figure 1 as corresponding to the points $y_1, \ldots, y_q$ in the sampling experiment of Figure 2. This view is however (in general) false, for two reasons, as we now explain.

First, and most importantly, in Figure 2, the points $y_1, \ldots, y_q$ are uniformly distributed over R, while in Figure 1, whether or not the points $y_1, \ldots, y_q$ are uniformly distributed over R depends on h, in particular being true if and only if h is regular. To see why this is true, let $R_1, \ldots, R_r$ denote the points in R, and for each $j \in [r]$ let

$$p_j \ = \ \frac{|h^{-1}(R_j)|}{|D|} \ . \tag{4}$$

This is the probability that $h(x) = R_j$ if we choose x at random from D. Thus for each $j \in [r]$ and each $i \in [q]$, when x_i is drawn at random from D and y_i is set to $h(x_i)$, we have $\Pr[y_i = R_j] = p_j$. So $y_1, \ldots, y_q$ are uniformly distributed over R if and only if $|h^{-1}(R_1)| = \cdots = |h^{-1}(R_r)|$, or, in other words, if and only if h is a regular function.

Second, the sampling experiment of Figure 2 is successful if $y_i = y_j$ for some $i \neq j$, while success in Figure 2 requires additionally that $x_i \neq x_j$. In other words, an analogy between the two corresponds to seeking only possibly trivial collisions rather than collisions in the birthday attack. This is less important than the first point, though, because if D is sufficiently larger than R, the probability that some two of $x_1, \ldots, x_q$ are equal can be neglected.

The conclusion from the above is that if h is a regular function then $C^*_h(q) = B_r(q)$ and $Q^*_h(c) = P_r(c)$, and if D is sufficiently larger than R these serve as approximate estimates of $C_h(q)$ and $Q_h(c)$ respectively. However if h is not regular then the sampling experiment and classical birthday analysis do not appear to have any particular bearing on the birthday attack on h and on the values of $C_h(q)$ and $Q_h(c)$.

Extending the approach. One way to proceed is to consider a more general version of the birthday problem in which the probability that a ball lands in bin j is not simply $1/r$ where r is the number of bins, but rather is a number p_j, where $p_1 + \cdots + p_r = 1$. Let $\boldsymbol{p} = (p_1, \ldots, p_r)$, let $B_{\boldsymbol{p}}(q)$ denote the probability of a collision in this game, and let $P_{\boldsymbol{p}}(c)$ be the smallest value of q for which $B_{\boldsymbol{p}}(q) \geq c$. Then it is easy to see that if $p_1, \ldots, p_r$ are defined by Equation (4) then $C^*_h(q) = B_{\boldsymbol{p}}(q)$ and $Q^*_h(c) = P_{\boldsymbol{p}}(c)$.

However, this generalized birthday problem is (surprisingly) not analyzed in the literature, so the analogy between it and our birthday attack problem does not yield any immediate results or analysis for the latter. Furthermore, this approach continues to consider possibly trivial collisions, while the object of interest is collisions, and even though the difference is low order, we prefer not to ignore it a priori. For these reasons, we analyze the birthday attack, and estimate $C_h(q)$ and $Q_h(c)$, directly.

We mention that using the same techniques, one can obtain results for the (easier) generalized birthday problem, and in Section 10 we state these. (The latter are not used in this paper, but might be of interest in other contexts.)

5 The Balance Measure and its Properties

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

Definition 5.1 Let h: $D \to R$ 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

$$\mu(h) \;=\; \log_r \left[\frac{d^2}{d_1^2 + \cdots + d_r^2} \right] \,, \qquad (5)$$

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

For some intuition about what this is measuring, note that

$$\frac{1}{r^{\mu(h)}} \;=\; \frac{d_1^2 + \cdots + d_r^2}{d^2}$$

is the probability that $h(a) = h(b)$ if a, b are drawn independently at random from the domain D.

It is easy to see that a regular function has balance one and a constant function has balance zero. The following says that these are the two extremes:

Proposition 5.2 *Let h be a function. Then $0 \le \mu(h) \le 1$. Furthermore $\mu(h) = 0$ iff h is a constant function, and $\mu(h) = 1$ iff h is a regular function.* ▮

Proof of Proposition 5.2: The proof is based on standard facts. Let

$$S = \{ (x_1, \ldots, x_r) \in \mathbb{R}^r \ : \ x_1 + \cdots + x_r = d \} .$$

Define the function f: $S \to \mathbb{R}$ by $f(x_1, \ldots, x_r) = x_1^2 + \cdots + x_r^2$ for any $x_1, \ldots, x_r \in S$ and let

$$\begin{aligned} \mathsf{Min}_S(f) &= \min\{ f(x_1, \ldots, x_r) \ : \ (x_1, \ldots, x_r) \in S \} \\ \mathsf{Max}_S(f) &= \max\{ f(x_1, \ldots, x_r) \ : \ (x_1, \ldots, x_r) \in S \} . \end{aligned}$$

The definition of $\mu(h)$ implies thaty

$$\mathsf{Min}_S(f) \ \le \ \frac{d^2}{r^{\mu(h)}} \ \le \ \mathsf{Max}_S(f) .$$

The extremums of f over S are well studied, and it is known that f achieves its minimum on S when $d_i = d/r$ for all $i \in [r]$, which implies $\mathsf{Min}_S(f) = r(d/r)^2 = d^2/r$ and corresponds to h being regular, with all points in the range having pre-image size d/r. On the other hand f achieves its maximum when $x_i = d$ for some $i \in [r]$ and $x_j = 0$ for all $j \in [r] - \{i\}$, which implies $\mathsf{Max}_S(f) = d^2$ and corresponds to h being a constant function that maps all d points in the domain to some single point in the range. We thus get

$$\frac{d^2}{r} \ \le \ \frac{d^2}{r^{\mu(h)}} \ \le \ d^2 .$$

Dividing by d^2 and re-arranging terms we get

$$1 \ \le \ r^{\mu(h)} \ \le \ r .$$

Taking logarithms to base r yields the Proposition. ▮

The following will be useful later.

Lemma 5.3 Let h: $D \to R$ be a function. Let $d = |D|$ and $r = |R|$ and assume $d \ge r \ge 2$. Then

$$r^{-\mu(h)} - \frac{1}{d} \ \ge \ \left(1 - \frac{r}{d}\right) \cdot r^{-\mu(h)} , \tag{6}$$

where $\mu(h)$ is the balance of h as per Definition 5.1. ▮

Proof of Lemma 5.3: Note that

$$r^{-\mu(h)} - \frac{1}{d} \ = \ \left(1 - \frac{r^{\mu(h)}}{d}\right) \cdot r^{-\mu(h)} .$$

Proposition 5.2 says that $\mu(h) \le 1$, and this implies that $r^{\mu(h)} \le r$. This in turn implies

$$1 - \frac{r^{\mu(h)}}{d} \ \ge \ 1 - \frac{r}{d} .$$

This concludes the proof. ▮

6 Balance-based Analysis of the Birthday attack

We state the main results and discuss them, and then go on to the proofs.

6.1 Bounds on $C_h(q)$

Theorem 6.1 below gives both upper and lower bounds on $C_h(q)$ that are within constant factors of each other. The proof of Theorem 6.1 is in Section 6.3.

Theorem 6.1 *Let h: $D \to R$ be a hash function. Let $d = |D|$ and $r = |R|$ and assume $d > r \geq 2$. Let $\alpha \geq 0$ be any real number. Then for any integer $q \geq 2$*

$$(1 - \alpha^2/4 - \alpha) \cdot \binom{q}{2} \cdot \left[\frac{1}{r^{\mu(h)}} - \frac{1}{d}\right] \;\leq\; C_h(q) \;\leq\; \binom{q}{2} \cdot \left[\frac{1}{r^{\mu(h)}} - \frac{1}{d}\right] , \tag{7}$$

the lower bound being true under the additional assumption that

$$q \;\leq\; \alpha \cdot \left(1 - \frac{r}{d}\right) \cdot r^{\mu(h)/2} . \quad \blacksquare \tag{8}$$

As we mentioned before, it is important to have upper and lower bounds on $C_h(q)$ that are close to each other, because we are making very specific choices of hash function parameters, in particular output lengths, based on these estimates, and if our estimates are not close to the reality then we might choose parameters incorrectly. Accordingly Theorem 6.1 strives for good bounds, and achieves this, since as $\alpha \to 0$, the lower bound of Equation (7) approaches the upper bound, so the bounds can be made as close as we want. However one must keep in mind that there is a tradeoff: as $\alpha \to 0$ the lower bound is valid across smaller and smaller ranges of q due to the restriction of Equation (8).

The following Corollary may be simpler and easier to understand or work with than the theorem, at least at first. By restricting attention to hash functions with domain is at least twice as large as the range it removes the $1/d$ term from Equation (7). Then by plugging in a particular value of α, precise constants emerge, showing that

$$C_h(q) \;=\; \Theta(1) \cdot \frac{q^2}{r^{\mu(h)}}$$

as long as q is not too large. Here we choose $\alpha = 2/5$, but the choice is arbitrary and made purely for the sake of illustration. The proof of Corollary 6.2 is in Section 6.4.

Corollary 6.2 Let h: $D \to R$ be a hash function. Let $d = |D|$ and $r = |R|$ and assume $d \geq 2r \geq 4$. Then for any integer $q \geq 2$

$$0.28 \cdot \binom{q}{2} \cdot \frac{1}{r^{\mu(h)}} \;\leq\; C_h(q) \;\leq\; \binom{q}{2} \cdot \frac{1}{r^{\mu(h)}} , \tag{9}$$

the lower bound under the assumption that $q \leq 0.2 \cdot r^{\mu(h)/2}$. $\blacksquare$

Typically, the bounds of Equation (9) are good enough, and Corollary 6.2 can be used directly, but in case one needs very close estimates of $C_h(q)$ one can return to Theorem 6.1.

In practice we do not expect the restriction $q \leq 0.2 \cdot r^{\mu(h)/2}$ to significantly reduce the applicability of Corollary 6.1, because $C_h(q)$ gets close to one as q gets close to $r^{\mu(h)/2}$, meaning the probability of a collision is already significant enough that we can view the attack as successful.

One should note, though, that due to this restriction (and the corresponding Equation (8) of Theorem 6.1), these results do not apply to some hash functions, namely those of extremely tiny balance. For example if $\mu(h) = 0$ then the assumed upper bounds on q together with the assumed lower bound $q \geq 2$ mean that in fact the result is vacuous, saying nothing about what happens in this case. However, in practice, hash functions with extremely tiny balance are unlikely to arise or be of interest, so the practical applicability of the result is not particularly impacted.

6.2 Bounds on $Q_h(c)$

Next, we present lower and upper bounds on $Q_h(c)$. The following theorem says that

$$Q_h(c) \;=\; \Theta(\sqrt{c}) \cdot r^{\mu(h)/2}$$

as long as c is not too close to one. The proof of Theorem 6.3 is in Section 6.5.

Theorem 6.3 *Let h: $D \to R$ be a hash function. Let $d = |D|$ and $r = |R|$ and assume $d \geq 2r \geq 4$. Let $\alpha \geq 0$ be any real number such that $\beta = 1 - \alpha^2/4 - \alpha > 0$. Let c be a real number in the interval $0 \leq c < 1$. Then*

$$\sqrt{2c} \cdot r^{\mu(h)/2} \;\leq\; Q_h(c) \;\leq\; 1 + \sqrt{\frac{4c}{\beta}} \cdot r^{\mu(h)/2}\,, \tag{10}$$

the upper bound being true under the additional assumption that

$$c \;\leq\; \left(\alpha \cdot (1 - r/d) - r^{-\mu(h)/2}\right)^2 \cdot \frac{\beta}{4}\,. \quad \blacksquare \tag{11}$$

Again, the statement of the following Corollary is simpler and easier to understand or work with at first. The proof of Corollary 6.4 is in Section 6.6.

Corollary 6.4 Let h: $D \to R$ be a hash function. Let $d = |D|$ and $r = |R|$ and assume $d \geq 2r \geq 4$. Then

$$\sqrt{2c} \cdot r^{\mu(h)/2} \;\leq\; Q_h(c) \;\leq\; 1 + 2.36 \cdot \sqrt{2c} \cdot r^{\mu(h)/2}\,, \tag{12}$$

the upper bound being true under the additional assumptions $c \leq 0.006$ and $r^{\mu(h)} \geq 2,200$. $\blacksquare$

As Equations (10) and (12) indicate, the lower and upper bounds on $Q_h(c)$ are quite close to each other. Let us now discuss the restrictions.

The upper bound on $Q_h(c)$ requires an upper bound on c, meaning is not valid for values of c close to 1. Specifically, the upper bound of Equation (12) of Corollary 6.4 requires that the probability c not exceed 0.6%. In practice, once the collision probability is as high as this, we would conclude that the attack succeeds, so this upper bound may not be too much of a restriction. Nonetheless, we would, ideally, prefer a result holding for larger values of c, but do not know how to obtain it.

Also required for the upper bound of Equation (12) of Corollary 6.4 is the condition $r^{\mu(h)} \geq 2,200$. Say $r = 2^n$. Then, for this condition to hold, it suffices that $\mu(h) \geq 11/n$. In practice r is very large, with n in the range 128–160, so the condition implies only that the result does not apply to functions of very small balance. (For example if $r = 2^{160}$, the range size of SHA-1, then it suffice that $\mu(h) \geq 11/160 \geq 0.068$). But as discussed above, such functions are not likely to arise in practice.

6.3 Proof of Theorem 6.1

Let

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

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

$$\mathbf{E}\left[X_I\right] \;=\; \Pr\left[X_I = 1\right] \;=\; \sum_{i=1}^{r} \frac{d_i(d_i - 1)}{d^2} \;=\; \sum_{i=1}^{r} \frac{d_i^2}{d^2} - \sum_{i=1}^{r} \frac{d_i}{d^2} \;=\; r^{-\mu(h)} - \frac{1}{d} \;=\; p \;. \tag{13}$$

By linearity of expectation we have

$$\mathbf{E}\left[X\right] \;=\; \sum_{I \in [q]_2} \mathbf{E}\left[X_I\right] \;=\; \binom{q}{2} \cdot p \;. \tag{14}$$

The upper bound of Theorem 6.1 is a simple application of Markov's inequality and Equation (14):

$$C_h(q) \;=\; \Pr\left[X \geq 1\right] \;\leq\; \frac{\mathbf{E}\left[X\right]}{1} \;=\; \binom{q}{2} \cdot p \;. \tag{15}$$

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

$$\begin{aligned} C_h(q) \;&=\; \Pr\left[\textstyle\bigvee_{I \in [q]_2} X_I = 1\right] \\ &\geq\; \sum_{I \in [q]_2} \Pr\left[X_I = 1\right] \;-\; \sum_{\{I,J\} \in [q]_{2,2}} \Pr\left[X_I = 1 \wedge X_J = 1\right] . \end{aligned} \tag{16}$$

Equation (14) tells us that the first sum above is

$$\sum_{I \in [q]_2} \Pr\left[X_I = 1\right] \;=\; \sum_{I \in [q]_2} \mathbf{E}\left[X_I\right] \;=\; \mathbf{E}\left[X\right] \;=\; \binom{q}{2} \cdot p \;. \tag{17}$$

We now claim that

$$\sum_{\{I,J\} \in [q]_{2,2}} \Pr\left[X_I = 1 \wedge X_J = 1\right] \;\leq\; \left(\frac{\alpha^2}{4} + \alpha\right) \cdot \binom{q}{2} \cdot p \;. \tag{18}$$

This completes the proof because from Equations (16), (17) and (18) we obtain the lower bound of Equation (7) as follows:

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

It remains to prove Equation (18).

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

$$\begin{aligned}
&\sum_{\{I,J\}\in[q]_{2,2}} \Pr[\, X_I = 1 \wedge X_J = 1 \,] \\
&= \underbrace{\sum_{\{I,J\}\in E} \Pr[\, X_I = 1 \wedge X_J = 1 \,]}_{S_E} + \underbrace{\sum_{\{I,J\}\in N} \Pr[\, X_I = 1 \wedge X_J = 1 \,]}_{S_N} \, . \qquad (19)
\end{aligned}$$

We now claim that

$$S_E \;\leq\; \binom{q}{2} \cdot \frac{1}{4} \cdot \alpha^2 \cdot p \qquad (20)$$

$$S_N \;\leq\; \binom{q}{2} \cdot \alpha \cdot p \, . \qquad (21)$$

Equation (18) follows from Equations (19), (20) and (21). We now prove Equations (20) and (21).

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 (13) 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

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

We now upper bound this as follows:

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

Above the first inequality is true because Theorem 6.1 assumes $q \geq 2$. The second inequality is true because of the assumption made in Equation (8). The third inequality is true because $r^{\mu(h)} \cdot p < 1$. We have now obtained Equation (20).

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

$$S_N \;=\; \sum_{\{I,J\}\in N} \Pr[\, X_I = 1 \wedge X_J = 1 \,] \;=\; |N| \cdot \sum_{i=1}^{r} \frac{d_i(d_i - 1)^2}{d^3} \;<\; \frac{|N|}{d^3} \cdot \sum_{i=1}^{r} d_i^3 \, . \qquad (22)$$

We now compute the size of the set N:

$$
\begin{aligned}
|N| &= \frac{1}{2}\binom{q}{2}\binom{q}{2} - \frac{1}{2}\binom{q}{2} - \frac{1}{2}\binom{q}{2}\binom{q-2}{2} \\
&= \binom{q}{2} \cdot \left[\frac{1}{2}\binom{q}{2} - \frac{1}{2}\binom{q-2}{2} - \frac{1}{2}\right] \\
&= \binom{q}{2} \cdot \left[\frac{q(q-1)}{4} - \frac{(q-2)(q-3)}{4} - \frac{1}{2}\right] \\
&= \binom{q}{2} \cdot (q-2)\ .
\end{aligned}
$$

Putting this together with Equation (22) we have

$$
S_N \ < \ \binom{q}{2} \cdot q \cdot \left[\frac{1}{d^3} \cdot \textstyle\sum_{i=1}^{r} d_i^3\right]\ . \tag{23}
$$

To upper bound the sum of Equation (23), 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 $\sum_{i=1}^{r} d_i^2 \ = \ d^2 \cdot r^{-\mu(h)}$.[5] The maximum occurs when $d_1 = d \cdot r^{-\mu(h)/2}$ and $d_i = 0$ for $i = 2, \ldots, r$, meaning that

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

Returning to Equation (23) with this information we get

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

We now use the assumption made in Equation (8), and finally use Lemma 5.3, to get

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

This proves Equation (21) and thus concludes the proof of Theorem 6.1.

6.4 Proof of Corollary 6.2

The upper bound on $C_h(q)$ is directly from Theorem 6.1. For the lower bound, begin by observing that

$$
\begin{aligned}
\left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot \binom{q}{2} \cdot \frac{1}{2} \cdot \frac{1}{r^{\mu(h)}} \ &\leq \ \left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot \binom{q}{2} \cdot \left(1 - \frac{r}{d}\right) \cdot \frac{1}{r^{\mu(h)}} \\
&\leq \ \left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot \binom{q}{2} \cdot \left[\frac{1}{r^{\mu(h)}} - \frac{1}{d}\right]\ .
\end{aligned}
$$

Above the first inequality is true because the assumption $d \geq 2r$ made in the Corollary implies $1/2 \leq 1 - r/d$. The second inequality uses Lemma 5.3. Setting $\alpha = 2/5$, the upper bound on q in

[5] There is another constraint as well, namely $d_1 + \cdots + d_r = d$. The maximum when this constraint is also considered could be lower than the one we discover, which would improve the bounds in the theorem, but we do not know how to do the maximization with this additional constraint.

Equation (8) is implied by the upper bound on q in the statement of the Corollary, again because $1/2 \leq 1 - r/d$. Now the proof of the lower bound of the Corollary follows from the lower bound of Theorem 6.1 once we check that

$$0.56 \;\leq\; 1 - \frac{\alpha^2}{4} - \alpha$$

when $\alpha = 2/5$.

6.5 Proof of Theorem 6.3

From Equation (7) of Theorem 6.1 we have

$$C_h(q) \;\leq\; \underbrace{\binom{q}{2} \cdot \frac{1}{r^{\mu(h)}}}_{U(q)} \, .$$

The (quadratic) equation $U(q) = c$ in unknown q has as its (only) non-negative root the value

$$q \;=\; \frac{1}{2} + \sqrt{\frac{1}{4} + 2cr^{\mu(h)}} \;\geq\; \sqrt{2cr^{\mu(h)}} \, .$$

This proves the lower bound of Equation (10). We now move to the upper bound. The assumption $d \geq 2r$ implies $1 - r/d \geq 1/2$. Now using Lemma 5.3 and Equation (7) of Theorem 6.1 we get

$$\begin{aligned} C_h(q) \;&\geq\; \beta \cdot \binom{q}{2} \cdot \left[\frac{1}{r^{\mu(h)}} - \frac{1}{d}\right] \\ &\geq\; \beta \cdot \binom{q}{2} \cdot \left(1 - \frac{r}{d}\right) \cdot r^{-\mu(h)} \\ &\geq\; \underbrace{\beta \cdot \binom{q}{2} \cdot \frac{1}{2} \cdot r^{-\mu(h)}}_{L(q)} \, . \end{aligned}$$

The (quadratic) equation $L(q) = c$ in unknown q has as its (only) non-negative root the value

$$q \;=\; \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{4cr^{\mu(h)}}{\beta}} \;\leq\; \frac{1}{2} + \sqrt{\frac{1}{4}} + \sqrt{\frac{4cr^{\mu(h)}}{\beta}} \;=\; \underbrace{1 + \sqrt{\frac{4cr^{\mu(h)}}{\beta}}}_{q_u} \, .$$

This proves the upper bound of Equation (10) as long as we can ensure that $q = q_u$ meets the restriction of Equation (8). Given the assumption made in Equation (11) we have

$$\begin{aligned} q_u \;&\leq\; 1 + \left[4r^{\mu(h)} \cdot \left(\alpha \cdot (1 - r/d) - r^{-\mu(h)/2}\right)^2 \cdot \frac{\beta}{4} \cdot \frac{1}{\beta}\right]^{1/2} \\ &=\; 1 + \left(\alpha \cdot (1 - r/d) - r^{-\mu(h)/2}\right) \cdot r^{\mu(h)/2} \\ &=\; 1 + \alpha \cdot (1 - r/d) \cdot r^{\mu(h)/2} - 1 \\ &=\; \alpha \cdot (1 - r/d) \cdot r^{\mu(h)/2} \, . \end{aligned}$$

Thus Equation (8) is true.

6.6 Proof of Corollary 6.4

Let $\alpha = (\sqrt{17}-3)/2 \approx 0.56155$ and let $\beta = 1 - \alpha^2/4 - \alpha$. (This choice of α maximizes $\alpha^2\beta$, in an attempt to get the largest possible range for c under the restriction of Equation (11).) Apply Theorem 6.3. The lower bound of Equation (12) is that of Equation (10). From the upper bound of the latter we have

$$Q_h(c) \;\leq\; 1 + \sqrt{\frac{4c}{\beta}} \cdot r^{\mu(h)/2} \;=\; 1 + \sqrt{\frac{2}{1-\alpha^2/4-\alpha}} \cdot \sqrt{2c} \cdot r^{\mu(h)/2} \;\leq\; 1 + 2.36 \cdot \sqrt{2c} \cdot r^{\mu(h)/2} \, .$$

It remains to check that Equation (11) is true. The assumption $r^{\mu(h)} \geq 2,200$ implies $r^{-\mu(h)/2} \leq \alpha/26$. So

$$\begin{aligned}
\left[\alpha(1-r/d) - r^{-\mu(h)/2}\right]^2 \cdot \frac{\beta}{4} \;&\geq\; (\alpha/2 - r^{-\mu(h)/2})^2 \cdot \frac{\beta}{4} \\
&\geq\; (\alpha/2 - \alpha/26)^2 \cdot \frac{\beta}{4} \\
&=\; (6\alpha/13)^2 \cdot \frac{\beta}{4} \\
&=\; (6\alpha/13)^2 \cdot \frac{1-\alpha^2/4-\alpha}{4} \\
&\geq\; 0.00603 \, .
\end{aligned}$$

Thus the condition $c \leq 0.006$ implies that Equation (11) holds.

7 Special classes of hash functions

We consider (and contrast) two classes of hash functions, namely regular ones and random ones. In this section we fix a domain D and range R with $d = |D| > r = |R|$.

7.1 Regular functions

A symmetry argument says that if $h_1, h_2\colon D \to R$ are regular functions, then $C_{h_1}(q) = C_{h_2}(q)$. Accordingly we denote this value by $C^{\mathrm{reg}}_{D,R}(q)$. Similarly $Q_{h_1}(c) = Q_{h_2}(c)$ and this value is denoted by $Q^{\mathrm{reg}}_{D,R}(c)$. Now, one can show:

Proposition 7.1 *If $h\colon D \to R$ is not regular then $C^{\mathrm{reg}}_{D,R}(q) < C_h(q)$ and $Q^{\mathrm{reg}}_{D,R}(c) > Q_h(c)$.* ▮

In other words, regular functions are the best with regard to security against the birthday attack. The collision probability is the lowest possible, and the collision threshold the highest possible.

7.2 Random functions

Designers of hash functions often have as target to make the hash function have "random" behavior. To assess how this impacts their security against the birthday attack, we consider the performance of the birthday attack when the function h is random.

Let $\mathsf{Maps}(D,R)$ denote the set of all functions with domain D and range R. Let us choose a function h at random from $\mathsf{Maps}(D,R)$, and then run the birthday attack of Figure 1. (This means we are in the random oracle model [4]). We let $C^{\$}_{D,R}(q)$ denote the probability that the attack succeeds, where q is the number of trials and the probability is over the initial choice of h and the choices of $x_1, \ldots, x_q$ made in the attack. We let $Q^{\$}_{D,R}(c)$ denote the smallest value of q for which $C^{\$}_{D,R}(q) \geq c$.

Now, when we draw h at random, it has some non-zero probability of being non-regular. (In fact it has some non-zero probability of being a constant function, for which the birthday attack will succeed after selecting only two distinct points in the domain). Given Proposition 7.1 we may conclude that:

Proposition 7.2 $C^{\$}_{D,R}(q) > C^{\mathrm{reg}}_{D,R}(q)$ *and* $Q^{\$}_{D,R}(c) < Q^{\mathrm{reg}}_{D,R}(c)$. ▮

In other words, random functions offer less security than regular functions against the birthday attack.

However, Proposition 7.2 is a qualitative statement, not a quantitative one. How much less is "less," and is it enough to matter in practice? Towards answering this question we begin by obtaining bounds on $C^{\$}_{D,R}(q)$. The proof of the following is in Section 7.4.

Theorem 7.3 *Let D, R be sets with $d = |D| > r = |R|$. Let $\alpha \geq 0$ be any real number. Then for any integer $q \geq 2$*

$$(1 - \alpha^2/4 - \alpha) \cdot \binom{q}{2} \cdot \left(1 - \frac{1}{d}\right) \cdot \frac{1}{r} \;\leq\; C^{\$}_{D,R}(q) \;\leq\; \binom{q}{2} \cdot \left(1 - \frac{1}{d}\right) \cdot \frac{1}{r}\,, \tag{24}$$

the lower bound being true under the additional assumption that

$$q \;\leq\; \alpha \cdot r^{1/2}\,. \quad ▮ \tag{25}$$

Theorem 7.3 improves on [16, Section 4.2.2] by presenting good bounds under precisely stated conditions, as opposed to approximate equality calculations.

It is interesting to compare Theorem 7.3 to the case $\mu(h) = 1$ (namely the case where h is regular) of Theorem 6.1. We see that the bounds are very similar but not identical. The difference can become detectable when d is close to r. To illustrate, the following, whose proof is in Section 7.5, shows that if $d = 2r$ then $C^{\$}_{D,R}(q)$ is more than $C^{\mathrm{reg}}_{D,R}(q)$ by a constant factor.

Proposition 7.4 *Suppose $|D| = 2|R| \geq 10$. Then*

$$C^{\$}_{D,R}(q) \;>\; \frac{8}{5} \cdot C^{\mathrm{reg}}_{D,R}(q) \tag{26}$$

for all q satisfying $2 \leq q \leq 0.1 \cdot r^{1/2}$. ▮

To be concrete, consider hash functions mapping $\{0,1\}^{n+1}$ to $\{0,1\}^n$ for some $n \geq 4$. If h is chosen at random then the probability of a collision in q trials is higher, by a factor of $8/5 = 1.6$, than it would be if h were regular. In particular, if we imagine that SHA-1 has random behavior, then the probability of a collision in q trials, when attacking the restriction of SHA-1 to inputs of length 161 bits, is higher, by 60%, than it would be for a regular function of 161 bits to 160 bits.

One might note that if $|D|$ increases with $|R|$ fixed, then $C^{\$}_{D,R}(q)$ approaches $C^{\mathrm{reg}}_{D,R}(q)$, meaning the difference between random and regular functions decreases as the size of the domain, relative to the size of the range, increases. Still, an adversary attacking a hash function with a very large domain D might restrict its choices of domain elements to some smaller subset of D. Thus the case $d = 2r$ is quite relevant, and in this case the difference between random and regular functions becomes greater.

7.3 Discussion

The conclusion that random functions do not fare as well as regular ones against the birthday attack may go against some prevailing intuition. One might argue that the performance of the attack on random functions is captured by the standard birthday phenomenon, since the image of each point drawn in the attack is equally likely to equal any range point. But, if so, isn't this the case where the attack fares least well? The above shows that the answer is no, but some intuition as to why this is the case, and why $C^{\$}_{D,R}(q) > C^{\mathrm{reg}}_{D,R}(q)$, might help.

Having picked x_1 from D, let $y_1 = h(x_1)$ and consider choosing x_2 at random from D. Assume $x_2 \neq x_1$, since otherwise x_2, x_1 do not form a collision. If h is random then the probability that $h(x_2) = h(x_1) = y_1$ is exactly $1/r$. But if h is regular then this probability is a little less, namely it is $1/r - 1/d$. This is the phenomenon that ultimately accounts for the difference.

Now, one might also suggest that the difference between random and regular functions pointed out above arises because our version of the birthday attack draws random domain points rather than random and distinct ones. Namely, one might think the difference is due to collisions in the domain points arising in the attack. This is not true. Suppose we consider the attack in which we draw random but distinct domain points $x_1, \ldots, x_q$ rather than random but independent domain points as in Figure 1. In that case it is true that the performance of the attack on random functions is captured exactly by the standard birthday phenomenon. However, even in this case, regular functions fare better than random functions. The underlying cause is the same as indicated above. Namely, once a point $x_i \in D$ has been selected, the probability of selecting a distinct point $x_j \in D$ such that $h(x_i) = h(x_j)$ is $1/r$ if h is random but is less, namely at most $1/r - 1/d$, if h is regular. In other words, the difference arises from more basic causes than collisions in domain points.

We would like to stress that saying regular functions fare better against the birthday attack than random ones does not mean that random ones fare poorly; as the above indicates the difference is small. In particular, our results and discussion are not a critique of the design principle of attempting to make hash functions have random behavior. We believe that this principle is sound and central to security. In practice, hash functions need to be designed not only to resist the birthday attack but also to resist cryptanalytic attacks, and for this, random behavior appears to be important. If it were possible to design hash functions that have random behavior subject to being regular it might improve security slightly, but this might be a harder task than designing hash functions that simply attempt to have random behavior. Our contrasting of random functions with regular ones was done more to highlight what we consider theoretically interesting phenomenon and shed some light on some aspects of the birthday attack.

7.4 Proof of Theorem 7.3

We follow the outline of the proof of Theorem 6.1, only indicating the changes. Let

$$p \;=\; \left(1 - \frac{1}{d}\right) \cdot \frac{1}{r} \;.$$

Now proceeding as in the proof of Theorem 6.1 we have

$$\mathbf{E}\,[X_I] \;=\; \Pr\,[\,X_I = 1\,] \;=\; p\;.$$

So Equation (15) is true with $C_h(q)$ replaced by $C^{\$}_{D,R}(q)$, proving the upper bound of Equation (24). We proceed to the lower bound. Continue substituting $C^{\$}_{D,R}(q)$ for $C_h(q)$ in the proof of Theorem 6.1. We claim that Equation (18) continues to hold, completing the proof of the lower bound of Equation (24) in the same way as in the proof of Theorem 6.1. To establish Equation (18), we claim that Equations (20) and (21) continue to hold.

We now justify Equation (20). Using the value of $|E|$ from the proof of Theorem 6.1, and then using Equation (25), we have

$$S_E \;=\; |E| \cdot p^2 \;=\; \frac{1}{2} \cdot \binom{q}{2} \cdot \binom{q-2}{2} \cdot p^2 \;\le\; \binom{q}{2} \cdot p \cdot \frac{q^2}{4} \cdot p \;\le\; \binom{q}{2} \cdot p \cdot \frac{\alpha^2 r}{4} \cdot p \,.$$

Equation (20) follows since $rp = 1 - 1/d < 1$.

We now justify Equation (21). Using the value of $|N|$ from the proof of Theorem 6.1, and then using Equation (25), we have

$$S_N \;=\; |N| \cdot \left(1 - \frac{1}{d}\right)^2 \cdot \frac{1}{r^2} \;=\; \binom{q}{2} \cdot (q-2) \cdot p^2 \;\le\; \binom{q}{2} \cdot p \cdot \alpha \cdot r^{1/2} \cdot p \,.$$

Equation (21) follows since $r^{1/2}p < rp = 1 - 1/d < 1$. This concludes the proof.

7.5 Proof of Proposition 7.4

For any real number $\alpha \ge 0$ we have:

$$\begin{aligned}
C^{\$}_{D,R}(q) \;&\ge\; (1 - \alpha^2/4 - \alpha) \cdot \binom{q}{2} \cdot \left(1 - \frac{1}{d}\right) \cdot \frac{1}{r} && (27)\\
&=\; 2 \cdot (1 - \alpha^2/4 - \alpha) \cdot \left(1 - \frac{1}{d}\right) \cdot \binom{q}{2} \cdot \left[\frac{1}{r} - \frac{1}{2r}\right] \\
&\ge\; 2 \cdot (1 - \alpha^2/4 - \alpha) \cdot \left(1 - \frac{1}{d}\right) \cdot C^{\mathrm{reg}}_{D,R}(q) && (28)
\end{aligned}$$

Equation (27) used the lower bound of Equation (24). Equation (28) used the upper bound of Equation (7) for the case $\mu(h) = 1$, and the assumption $d = 2r$ made in the theorem statement. Now, set $\alpha = 0.1$. Since $d \ge 10$ we get

$$2 \cdot (1 - \alpha^2/4 - \alpha) \cdot \left(1 - \frac{1}{d}\right) \;\ge\; \frac{359}{200} \cdot \left(1 - \frac{1}{10}\right) \;=\; \frac{3231}{200} \;>\; \frac{8}{5} \,.$$

8 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 $b \ge 1$ is called the *block-length* and $c \ge 1$ is called the *chaining-length*. Then transform H into a hash function $\overline{H}\colon D_b \to \{0,1\}^c$, where

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

via the Merkle-Damgård (MD) [10, 6] transform depicted in Figure 3. (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 [12], SHA-1 [11] and RIPEMD-160 [8].

For the considerations in this section, we will focus on the restriction of $\overline{H}$ to strings of some particular length. For any integer $1 \le n < 2^b$ (the number of blocks) we let $\overline{H}_n\colon D_{b,n} \to \{0,1\}^c$ 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.

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}

Figure 3: Hash function $\overline{H}\colon D_b \to \{0,1\}^c$ obtained via the MD transform applied to compression function $H\colon \{0,1\}^{b+c} \to \{0,1\}^c$.

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

Proof of Proposition 8.1: Let $H\colon \{0,1\}^{b+c} \to \{0,1\}^c$ 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 [10, 6]. However, the above shows that it does not provably preserve balance. Even though Proposition 8.1 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.

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 8.1 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. Recall that $\overline{H}_n$ is the restriction of $\overline{H}$ to the domain $D_{b,n}$.

Proposition 8.2 *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 of Proposition 8.2: The computation of $\overline{H}_n$ can be written as

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

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 8.2 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 8.3 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.

Proposition 8.3 *Let b, c be integers, $b \geq c \geq 2$, and let $n \geq c$. Then there exists a compression function H: $\{0,1\}^{b+c} \rightarrow \{0,1\}^c$ 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.* ▮

Proof of Proposition 8.3: Let H: $\{0,1\}^{b+c} \rightarrow \{0,1\}^c$ be defined as

$$
H(B\|C) = \begin{cases} \langle B \rangle_c & C = 0^c \\ \langle C \ll 1 \rangle_c \oplus (0^{c-1}1) & C \neq 0^c \end{cases}
$$

where B is b-bits long, C is c-bits long, $\langle B \rangle_c$ is the right-most c bits of B, and $\langle C \ll 1 \rangle_c$ is the left shift of C by one bit (ie. $\langle C \ll 1 \rangle_c$ is a c-bit string, the left-most $c-1$ bits of which are the right-most $c-1$ bits of C, and the right-most bit of which is 0).

Clearly $\mu(H_0) = 1$. To see that $\mu(H) \geq 1 - 1/c$ we note that there are 2^{c-1} points $X \in \{0,1\}^c$ with a right-most bit of 0 and each of these points has 2^{b-c} pre-images (corresponding to the set $\{0,1\}^{b-c}X0^c$). There is one point of the form $0^{c-1}1$ and it has $2^b + 2^{b-c}$ pre-images (corresponding to the sets $\{0,1\}^b10^{c-1}$ and $\{0,1\}^{b-c}0^{c-1}10^c$). There are $2^{c-1} - 1$ additional points $Y \in \{0,1\}^c$ with a right-most bit of 1 and each of these points has $2^{b+1} + 2^{b-c}$ pre-images (corresponding to the sets $\{0,1\}^{b+1}Y'$ and $\{0,1\}^{b-c}Y0^c$, where Y' is the left-most $c-1$ bits of Y). Let

$$
\begin{aligned}
S &= 2^{c-1}(2^{b-c})^2 + (2^b + 2^{b-c})^2 + (2^{c-1} - 1)(2^{b+1} + 2^{b-c})^2 \\
&\leq 2^{2b+c+1} .
\end{aligned}
$$

It follows that

$$
\mu(H) \ = \ \log_{2^c} \left[\frac{2^{2b+2c}}{S} \right] \geq \log_{2^c} \left[\frac{2^{2b+2c}}{2^{2b+c+1}} \right] = \frac{c-1}{c} .
$$

We lower bound $\mu(H^n)$ as follows. Let $R = \sum_{C \in \{0,1\}^c} d_C^2$, where d_C^2 is the number of pre-images of $C \in \{0,1\}^c$ under H^n. We divide the analysis into three cases. In the first case we assume that the right-most bit of $\langle n \rangle_b$ is 0. This implies that there will be one point in $\{0,1\}^{c-1}0$ with one pre-image and all the remaining points in $\{0,1\}^{c-1}0$ will have no pre-image. Of the 2^{c-1} points in $\{0,1\}^{c-1}1$, all but point $0^{c-1}1$ will have two pre-images, and $0^{c-1}1$ will have one pre-image. Thus $R < 2^{c+1}$.

Let $\langle n \rangle_c$ be the right-most c bits of $\langle n \rangle_b$. In the second case we assume that the right-most bit of $\langle n \rangle_b$ is 1 and and that $\langle n \rangle_c \neq 0^{c-1}1$. All the points in $\{0,1\}^{c-1}0$ have no pre-images, $2^{c-1} - 2$ points in $\{0,1\}^{c-1}1$ have two pre-images, the point $\langle n \rangle_c$ has three pre-images, and the point $0^{c-1}1$

has one pre-image. In this case $R < 2^{c+2}$. In the final case we assume that the right-most bit of $\langle n \rangle_b$ is 1 and that $\langle n \rangle_c = 0^{c-1}1$. All the points in $\{0,1\}^{c-1}0$ have no pre-images and all the points in $\{0,1\}^{c-1}1$ have two pre-images. In this case $R = 2^{c+1}$. These results imply that

$$\mu(H^n) \;=\; \log_{2^c}\left[\frac{2^{2c}}{R}\right] \ge \log_{2^c}\left[\frac{2^{2c}}{2^{c+2}}\right] = \frac{c-2}{c}\;.$$

Let us now consider the balance of $\overline{H}_n$. Let $M = M_1\|\cdots\|M_n \in \{0,1\}^{bn}$ be a string and let $|M_i| = b$. Then if $M_1 \notin \{0,1\}^{b-c}0^c$, we have that $\overline{H}_n(M) = 1^c$; i.e. $|\overline{H}_n^{-1}(1^c)| \ge 2^{bn} - 2^{bn-c}$. This allows us to upper bound $\mu(\overline{H}_n)$ as follows:

$$\mu(\overline{H}_n) \le \log_{2^c}\left[\frac{(2^{bn})^2}{(2^{bn}-2^{bn-c})^2}\right] \le \log_{2^c}\left[\frac{2^{2bn}}{2^{2bn}-2^{2bn-c+1}}\right]$$

Using the assumption that $c \ge 2$,

$$\mu(\overline{H}_n) \le \log_{2^c}\left[\frac{2^{2bn}}{2^{2bn-1}}\right] = \frac{1}{c}$$

as desired. ▌

As indicated by Proposition 7.2, 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 8.2, as exhibited by Proposition 8.3, means that Proposition 8.2 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.

9 Extensions and variations

We consider two issues. One is extending the definitions and results here to families of hash functions rather than individual functions, and the other is variants of the attack that differ in the way the points are chosen from the domain.

9.1 Treating families of hash functions

A *family of functions* is a map $H\colon K \times D \to R$, where K is the set of keys, D is the domain and R is the range of the family. It is a family of hash functions if $|D| > |R|$. For each key $k \in K$ we let the function $H_k\colon D \to R$ be defined by $H_k(x) = H(k,x)$ for all $x \in D$. We say that H_k is a *member* of the family H. In usage, a key k is drawn at random and made public, specifying a particular hash function H_k.

This approach is particularly important in theoretical treatments involving proofs of security of collision-resistance [6, 3], for there appears to be no meaningful formalization of a notion of collision-resistance for single functions as opposed to families. We, however, are not discussing the notion of collision-resistance, but rather the performance of a particular attack, so in our case consideration of a single hash function as opposed to a family is meaningful. It also more directly reflects practice, where we have hash functions like MD5, SHA-1 and RIPEMD-160 not overlain by any explicit families.

Still, one might be interested in how the birthday attack fares against a family of functions H. Here we discuss how our metrics and results lift easily from single functions to families.

First, let us extend the collision-probability metric. When function H_k of family H has been chosen, the probability of finding a collision in q trials is $C_{H_k}(q)$. Since the choice of k is made at random from K, the metric of interest, which we denote $C_H(q)$, is

$$C_H(q) \;=\; \frac{1}{|K|} \cdot \sum_{k \in K} C_{H_k}(q) \;.$$

Each member H_k of H also has an associated balance $\mu(H_k)$. We extend the balance measure to families by defining for H a balance $\mu(H)$, computed as a function of the balance of the members of the family, as follows:

Definition 9.1 Let $H\colon\ K \times D \to R$ be a family of hash functions whose domain D and range R have sizes $d, r \geq 2$, respectively. The *balance* of H, denoted $\mu(H)$, is defined as

$$\mu(H) \;=\; \log_r \left[\frac{1}{|K|} \cdot \sum_{k \in K} \frac{1}{r^{\mu(H_k)}} \right]^{-1}$$

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

In other words,

$$\frac{1}{r^{\mu(H)}} \;=\; \frac{1}{|K|} \cdot \sum_{k \in K} \frac{1}{r^{\mu(H_k)}} \;.$$

Now we claim that

$$C_H(q) \;=\; \Theta(1) \cdot \binom{q}{2} \cdot \frac{1}{r^{\mu(H)}}$$

assuming some appropriate upper bound on q. The precise bounds are as follows:

Theorem 9.2 *Let $H\colon\ K \times D \to R$ be a family of hash functions. Let $d = |D|$ and $r = |R|$ and assume $d > r \geq 2$. Let $\alpha \geq 0$ be any real number. Then for any integer $q \geq 2$*

$$(1 - \alpha^2/4 - \alpha) \cdot \binom{q}{2} \cdot \left[\frac{1}{r^{\mu(H)}} - \frac{1}{d} \right] \;\leq\; C_H(q) \;\leq\; \binom{q}{2} \cdot \left[\frac{1}{r^{\mu(H)}} - \frac{1}{d} \right] \;,$$

the lower bound being true under the additional assumption that

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

This shows that Theorem 6.1 lifts in a simple and natural way from individual functions to families of functions. This theorem follows easily from Theorem 6.1 and the above definitions, so we omit the proof.

9.2 Variants of the attack

The version of the birthday attack on a hash function $h\colon\ D \to R$ that we consider draws points $x_1, \ldots, x_q$ uniformly and independently at random from the domain D. Other possibilities for how these points may be chosen have been mentioned and considered in the literature.

One is to use a fixed sequence of points. For example if the domain is $\{0,1\}^n$ for some n, then we identify it with $\{1, \ldots, 2^n\}$ and use the points $1, \ldots, q$. This strategy is effective if the function h is random (cf. Section 7), and in this case the performance of the attack is captured by the standard birthday problem. However, this is not a viable attack strategy in general, because there are certainly hash functions where this attack fails even though collisions may be easy to find and may be found by other variants of the birthday attack. Accordingly we do not consider this attack.

Another possibility is to use a sequence of random but distinct points $x_1, \ldots, x_q$, as opposed to random and independently chosen ones. We have not considered this attack in detail because the lack of independence makes a precise analysis harder. However, as long as $d = |D|$ is somewhat larger than $r = |R|$, say $d \geq 2r$, the performance of this attack will approach that of the one we consider, since the probability that some two of $x_1, \ldots, x_q$ are equal in Figure 1 is small compared to the probability of a collision. Thus in practice we do not expect serious performance differences between this attack and the one we consider.

10 The generalized birthday problem

In the birthday problem, we have q balls $1, \ldots, q$ and r bins $1, \ldots, r$. In the standard version of the problem, we throw the balls at random into the bins, each ball having probability $1/r$ of landing in each bin, and the probabilities for different balls being independent. A collision occurs if there are two different balls that land in the same bin.

One can however consider a more general problem in which the probability of a ball landing in bin i depends on i rather than being equal to $1/r$ for each i. To discuss this, let us say that $\boldsymbol{p} = (p_1, \ldots, p_r)$ is a *probability vector* if $p_1 + \cdots + p_r = 1$ and $0 \leq p_i \leq 1$ for all $i \in [r]$. Now in the *balls in bins game associated to* $\boldsymbol{p}$, we throw the q balls at random into the r bins in such a way that for every $i \in [q]$ and $j \in [r]$, ball i has probability p_j of landing in bin j, and the probabilities for different balls are independent. Again, a collision is said to occur if there are two different balls that land in the same bin. The following defines some metrics of interest for this game:

Definition 10.1 Let $\boldsymbol{p} = (p_1, \ldots, p_r)$ be a probability vector, where $r \geq 2$. For any integer $q \geq 2$, we let $B_{\boldsymbol{p}}(q)$ denote the probability of a collision in the balls in bin game associated to $\boldsymbol{p}$. For any real number c with $0 \leq c < 1$ we let

$$P_{\boldsymbol{p}}(c) \quad = \quad \min \{ \, q \; : \; B_{\boldsymbol{p}}(q) \geq c \, \}$$

denote the minimum number of balls required for the probability of a collision to exceed c. We refer to $B_{\boldsymbol{p}}$ as the *collision probability function* of $\boldsymbol{p}$ and to $P_{\boldsymbol{p}}$ as the *collision threshold function* of $\boldsymbol{p}$. ▮

Let $\boldsymbol{r} = (1/r, \ldots, 1/r)$. The literature contains analyses of the standard birthday problem, meaning bounds on $B_{\boldsymbol{r}}(q)$ (we had denoted this by $B_r(q)$ in Section 4), e.g. [1]. The more general version of the problem seems natural and potentially useful, in particular in the context of problems like the ones considered in this paper, but we have not found any analysis or bounds for this general problem in the literature. Accordingly we provide some here. To do so we first introduce a measure, which we again call the balance, associated to a probability vector.

Definition 10.2 Let $\boldsymbol{p} = (p_1, \ldots, p_r)$ be a probability vector, where $r \geq 2$. The *balance* of $\boldsymbol{p}$, denoted $\mu(\boldsymbol{p})$, is defined as

$$\mu(\boldsymbol{p}) \;=\; \log_r \left[p_1^2 + \cdots + p_r^2 \right]^{-1} \;,$$

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

For some intuition about what this is measuring, note that

$$\frac{1}{r^{\mu(\boldsymbol{p})}} \;=\; p_1^2+\cdots+p_r^2$$

is the probability that any two particular balls collide. Now the following shows that

$$B_{\boldsymbol{p}}(q) \;=\; \Theta(1)\cdot\binom{q}{2}\cdot\frac{1}{r^{\mu(\boldsymbol{p})}} \;=\; \Theta(1)\cdot\binom{q}{2}\cdot(p_1^2+\cdots+p_r^2)\;,$$

assuming some appropriate upper bound on q. The proof is in Section 10.1.

Theorem 10.3 *Let $\boldsymbol{p}=(p_1,\ldots,p_r)$ be a probability vector, where $r\geq 2$. Let $\alpha\geq 0$ be any real number. Then for any integer $q\geq 2$*

$$(1-\alpha^2/4-\alpha)\cdot\binom{q}{2}\cdot\frac{1}{r^{\mu(\boldsymbol{p})}} \;\leq\; B_{\boldsymbol{p}}(q) \;\leq\; \binom{q}{2}\cdot\frac{1}{r^{\mu(\boldsymbol{p})}}\;, \tag{29}$$

the lower bound being true under the additional assumption that

$$q \;\leq\; \alpha\cdot r^{\mu(\boldsymbol{p})/2}\;. \quad \blacksquare \tag{30}$$

The bounds in the theorem are good. In particular as $\alpha\to 0$ the lower bound approaches the upper bound, although being valid across a smaller range of values for the number q of balls thrown. In particular we remark that in the case of the standard birthday problem, namely when $\boldsymbol{p}=\boldsymbol{r}$, the above enables us to obtain better lower bounds than shown in [1], but valid across a smaller range of q.

We discussed in Section 4 how the generalized birthday problem is related to the analysis of the birthday attack for a function h: $D\to R$. Namely if $R=\{R_1,\ldots,R_r\}$, let $d_i=|h^{-1}(R_i)$ and let $p_i=d_i/d$ $(1\leq i\leq r)$. Then $B_{\boldsymbol{p}}(q)=C_h^*(q)$ is the probability of finding possibly trivial collisions in the birthday attack. We attacked the analysis of the birthday attack on h directly, rather than via the generalized birthday problem, only to ensure that our analysis is about collisions, not trivial collisions. However the heart of the problem is really the generalized birthday problem.

10.1 Proof of Theorem 10.3

We follow the outline of the proof of Theorem 6.1, only indicating the changes. Let

$$p \;=\; \frac{1}{r^{\mu(\boldsymbol{p})}}\;.$$

Let $[q]_2$ denote the set of all two-element subsets of $[q]$. We associate to any two-element set $I=\{i,j\}\in[q]_2$ the random variable X_I which takes value 1 if balls i,j form a collision (meaning land in the same bin), and 0 otherwise. Then for any $I\in[q]_2$ we have

$$\mathbf{E}\left[X_I\right] \;=\; \Pr\left[\,X_I=1\,\right] \;=\; \sum_{i=1}^{r}p_i^2 \;=\; p\;.$$

We continue to follow the proof of Theorem 6.1, the random variable X being defined as there, and $B_{\boldsymbol{p}}(q)$ being substituted for $C_h(q)$. The proof of the upper bound of Equation (29) is unchanged, and we proceed to the lower bound. We need to make amendments only when we get to the upper bounding of S_N. In place of Equation (22) we have

$$S_N \;=\; \sum_{\{I,J\}\in N}\Pr\left[\,X_I=1\wedge X_J=1\,\right] \;=\; |N|\cdot\sum_{i=1}^{r}p_i^3\;. \tag{31}$$

Now, identify d_i/d (in the proof of Theorem 6.1) with p_i (in the current proof) and continue. This will conclude the proof.

11 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 6.3 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 5.1 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 are shown below. These values indicate that, for the specified values of t_1, t_2, the balance of $\mathrm{SHA}_{32;t_1\ldots t_2}$ is high.

$t_2 - t_1 + 1 = 8$	$t_2 - t_1 + 1 = 16$	$t_2 - t_1 + 1 = 24$
$\mu(\mathrm{SHA}_{32;1\ldots8}) = 0.99999998893$	$\mu(\mathrm{SHA}_{32;1\ldots16}) = 0.999998623$	$\mu(\mathrm{SHA}_{32;1\ldots24}) = 0.99976567$
$\mu(\mathrm{SHA}_{32;9\ldots16}) = 0.99999998941$	$\mu(\mathrm{SHA}_{32;9\ldots24}) = 0.999998604$	$\mu(\mathrm{SHA}_{32;9\ldots32}) = 0.99976548$
$\mu(\mathrm{SHA}_{32;17\ldots24}) = 0.99999998972$	$\mu(\mathrm{SHA}_{32;17\ldots32}) = 0.999998620$	$\mu(\mathrm{SHA}_{32;17\ldots40}) = 0.99976553$
$\mu(\mathrm{SHA}_{32;25\ldots32}) = 0.99999998884$	$\mu(\mathrm{SHA}_{32;25\ldots40}) = 0.999998627$	$\mu(\mathrm{SHA}_{32;25\ldots48}) = 0.99976561$
$\mu(\mathrm{SHA}_{32;33\ldots40}) = 0.99999999079$	$\mu(\mathrm{SHA}_{32;33\ldots48}) = 0.999998641$	$\mu(\mathrm{SHA}_{32;33\ldots56}) = 0.99976582$
$\mu(\mathrm{SHA}_{32;41\ldots48}) = 0.99999998909$	$\mu(\mathrm{SHA}_{32;41\ldots56}) = 0.999998620$	$\mu(\mathrm{SHA}_{32;41\ldots64}) = 0.99976559$
$\mu(\mathrm{SHA}_{32;49\ldots56}) = 0.99999998912$	$\mu(\mathrm{SHA}_{32;49\ldots64}) = 0.999998626$	$\mu(\mathrm{SHA}_{32;49\ldots72}) = 0.99976558$
$\mu(\mathrm{SHA}_{32;57\ldots64}) = 0.99999999083$	$\mu(\mathrm{SHA}_{32;57\ldots72}) = 0.999998625$	$\mu(\mathrm{SHA}_{32;57\ldots80}) = 0.99976581$
$\mu(\mathrm{SHA}_{32;65\ldots72}) = 0.99999998923$	$\mu(\mathrm{SHA}_{32;65\ldots80}) = 0.999998627$	$\mu(\mathrm{SHA}_{32;65\ldots88}) = 0.99976575$
$\mu(\mathrm{SHA}_{32;73\ldots80}) = 0.99999999083$	$\mu(\mathrm{SHA}_{32;73\ldots88}) = 0.999998637$	$\mu(\mathrm{SHA}_{32;73\ldots96}) = 0.99976577$
$\mu(\mathrm{SHA}_{32;81\ldots88}) = 0.99999998925$	$\mu(\mathrm{SHA}_{32;81\ldots96}) = 0.999998622$	$\mu(\mathrm{SHA}_{32;81\ldots104}) = 0.99976558$
$\mu(\mathrm{SHA}_{32;89\ldots96}) = 0.99999998987$	$\mu(\mathrm{SHA}_{32;89\ldots104}) = 0.999998617$	$\mu(\mathrm{SHA}_{32;89\ldots112}) = 0.99976554$
$\mu(\mathrm{SHA}_{32;97\ldots104}) = 0.99999998862$	$\mu(\mathrm{SHA}_{32;97\ldots112}) = 0.999998624$	$\mu(\mathrm{SHA}_{32;97\ldots120}) = 0.99976567$
$\mu(\mathrm{SHA}_{32;105\ldots112}) = 0.99999998826$	$\mu(\mathrm{SHA}_{32;105\ldots120}) = 0.999998626$	$\mu(\mathrm{SHA}_{32;105\ldots128}) = 0.99976562$
$\mu(\mathrm{SHA}_{32;113\ldots120}) = 0.99999998959$	$\mu(\mathrm{SHA}_{32;113\ldots128}) = 0.999998616$	$\mu(\mathrm{SHA}_{32;113\ldots136}) = 0.99976566$
$\mu(\mathrm{SHA}_{32;121\ldots128}) = 0.99999998999$	$\mu(\mathrm{SHA}_{32;121\ldots136}) = 0.999998634$	$\mu(\mathrm{SHA}_{32;121\ldots144}) = 0.99976556$
$\mu(\mathrm{SHA}_{32;129\ldots136}) = 0.99999999052$	$\mu(\mathrm{SHA}_{32;129\ldots144}) = 0.999998636$	$\mu(\mathrm{SHA}_{32;129\ldots152}) = 0.99976563$
$\mu(\mathrm{SHA}_{32;137\ldots144}) = 0.99999998916$	$\mu(\mathrm{SHA}_{32;137\ldots152}) = 0.999998615$	$\mu(\mathrm{SHA}_{32;137\ldots160}) = 0.99976554$
$\mu(\mathrm{SHA}_{32;145\ldots152}) = 0.99999998769$	$\mu(\mathrm{SHA}_{32;145\ldots160}) = 0.999998626$	
$\mu(\mathrm{SHA}_{32;153\ldots160}) = 0.99999998993$		

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, 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.

Acknowledgments

We thank Don Coppersmith and Michael Wiener for comments on the preliminary version of this paper [2].

References

[1] M. Bellare, J. Kilian and P. Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences* 61(3), Dec 2000, 362–399.

[2] M. Bellare and T. Kohno. Hash function balance and its impact on birthday attacks. *Advances in Cryptology – EUROCRYPT '04*, Lecture Notes in Computer Science Vol. 3027, C. Cachin and J. Camenisch ed., Springer-Verlag, 2004. Preliminary version of this paper.

[3] M. Bellare and P. Rogaway. Collision-resistant hashing: Towards making UOWHFs practical. *Advances in Cryptology – CRYPTO '97*, Lecture Notes in Computer Science Vol. 1294, B. Kaliski ed., Springer-Verlag, 1997.

[4] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. *First ACM Conference on Computer and Communications Security*, ACM, 1993.

[5] J. Buchmann. Introduction to cryptography. Springer, 2000.

[6] 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.

[7] H. Delfs and H. Knebl. Introduction to cryptography: Principles and applications. Springer, 2002..

[8] 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.

[9] A. Menezes, P. van Oorschot and S. Vanstone. Handbook of applied cryptography. CRC Press, 1997.

[10] 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.

[11] National Institute of Standards. FIPS 180-2, Secure hash standard. August 1, 2000.

[12] R. Rivest. The MD5 message-digest algorithm. IETF RFC 1321, April 1992.

[13] B. Schneier. Applied cryptography. Second edition. Wiley, 1996.

[14] W. Stallings. Cryptography and network security: Principles and practices. Third edition. Prentice Hall, 2003.

$i \leftarrow 0$; found $\leftarrow$ FALSE
While (found = FALSE) do
$i \leftarrow i+1$; $x_i \stackrel{\$}{\leftarrow} D$; $y_i \leftarrow h(x_i)$
If $(\exists j : j < i \ \& \ y_i = y_j \ \& \ x_i \neq x_j)$
then found $\leftarrow$ TRUE ; $q \leftarrow i$
EndIf
EndWhile
Return x_i, x_j

Figure 4: Extended birthday attack on a hash function h: $D \to R$. The attack continues until a collision is found. The number of trials q is now a random variable.

[15] D. Stinson. Cryptography theory and practice, First edition. CRC, 1995.

[16] D. Stinson. Cryptography theory and practice, Second edition. Chapman and Hall/CRC, 2002.

[17] D. Stinson. Some observations on the theory of cryptographic hash functions. Manuscript, 2004. http://www.cacr.math.uwaterloo.ca/~dstinson/.

[18] P. van Oorschot and M. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology* 12(1), Jan 1999, 1–28.

[19] P. Rogaway and T. Shrimpton. Cryptographic hash-function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision-resistance. *Fast Software Encryption '04*, Lecture Notes in Computer Science Vol. 3017 , B. Roy and W. Meier ed., Springer-Verlag, 2004.

[20] G. Yuval. How to swindle Rabin. *Cryptologia* (3), 1979, 187–190.

A The expected number of trials to find a collision

There is another metric related to the birthday attack that is of interest. Suppose we do not fix the number of trials a priori, but rather run the attack until it succeeds. We call this the extended birthday attack, and it is depicted in Figure 4. We record the number of trials q taken to find a collision. This is now a random variable, and we are interested in its expectation. The latter is the expected number of trials to find a collision. We denote it by E_h.

Given Theorem 6.3, we would expect that $E_h = \Theta(r^{\mu(h)/2})$. The following confirms this, providing both upper and lower bounds. The proof is in Section A.1.

Theorem A.1 *Let h: $D \to R$ be a hash function. Let $d = |D|$ and $r = |R|$ and assume $d \geq 2r \geq 4$. Assume $((\sqrt{7}-2)/3) \cdot r^{\mu(h)/2} \geq 2$. Then*

$$(1/2) \cdot r^{\mu(h)/2} \ \leq \ E_h \ \leq \ 72 \cdot r^{\mu(h)/2} \ . \quad \blacksquare \tag{32}$$

We note however that the bounds are not very good. This metric appears to be harder to analyze, or obtain good bounds for, as compared to the metrics we have considered in Section 6.

A.1 Proof of Theorem A.1

We begin by proving the lower bound. Let the random variable Y denote the number of trials to collision. Let $D_h(q)$ denote the probability of finding the first collision on the q-th trial. Let $Q = r^{\mu(h)/2}$. From the definition of Y:

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

We claim that

$$C_h(Q-1) < \frac{1}{2} \,. \tag{33}$$

It follows that

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

as desired. We now justify Equation (33). From the upper bound of Equation (9) of Corollary 6.2 we know that

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

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

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

as desired.

For the upper bound, we must be careful since there is an upper restriction on q in Equation (9) and Equation (7). 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 6.1 with α and q defined as above. Combining Theorem 6.1 with Lemma 5.3 and the assumptions that $d \geq 2r$ and $q = (\alpha/2) \cdot r^{\mu(h)/2} \geq 2$, we have

$$C_h(q) \;\geq\; \left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot \binom{q}{2} \cdot \frac{1}{2} \cdot \frac{1}{r^{\mu(h)}} \;\geq\; \left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot q^2 \cdot \frac{1}{8} \cdot \frac{1}{r^{\mu(h)}} \;.$$

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

$$C_h(q) \;\geq\; \left(1 - \frac{\alpha^2}{4} - \alpha\right) \cdot \left(\frac{\alpha}{2} \cdot r^{\mu(h)/2}\right)^2 \cdot \frac{1}{8} \cdot \frac{1}{r^{\mu(h)}} \;=\; \frac{1}{32} \cdot \left(\alpha^2 - \frac{\alpha^4}{4} - \alpha^3\right) \;. \tag{34}$$

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, ... do
    For i = 1, ..., q do
        Pick x_{q(j-1)+i} at random from the domain of h
        y_{q(j-1)+i} ← h(x_{q(j-1)+i})
        If ∃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} ≠ 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

$$\mathbf{E}[Y] \;\leq\; \mathbf{E}[A] \tag{35}$$

and

$$\mathbf{E}[A] \;\leq\; \frac{q}{C_h(q)} \;, \tag{36}$$

and combining with Equation (34), it follows that

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

giving the upper bound in the theorem statement.

To prove Equation (35) 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 (36), 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 $C_h(q)$, standard results tell us that

$$\mathbf{E}[Z] \;\leq\; \frac{1}{C_h(q)} \;. \tag{37}$$

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

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

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

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

it follows that

$$\mathbf{E}\left[A\right] \;\leq\; q \cdot \sum_{j=1}^{\infty} j \cdot G(j) \;=\; q \cdot \mathbf{E}\left[Z\right]\ . \tag{38}$$

Combining Equation (37) with Equation (38) yields Equation (36), completing the proof.