EXPLORING THE BIRTHDAY ATTACK / PARADOX[1]
: A Powerful Vehicle Underlying Information Security

Khosrow Moshirvaziri, *Information Systems Dept., California State University, Long Beach, Long Beach, CA 90840, 562-985-7965, moshir@csulb.edu*

Mahyar A. Amouzegar, *Department of Economics and Finance, University of New Orleans, New Orleans, LA 70148, 504-280-6595, mahyar@uno.edu*

Fahimeh Rezayat, *Information Systems & Operations Management, California State University, Dominguez Hills, Carson, CA 90747, 310-243-3484, frezayat@csudh.edu*

ABSTRACT

The primary objective of this research is to investigate the derivation of some important forms of *Birthday Paradox*, and its various collisions scenarios. Additionally, this paper aims to survey significant application areas of Birthday Paradox and the fundamental role it plays in information security. We present both computational, as well as simulated results on various derivations for verification purposes.

BACKGROUND

The Birthday paradox exists in various settings and situations in science, business, and technology related applications. In this research we will analyze this paradox in its various forms and explore important occurrence of new application areas. The birthday paradox is concerns with unexpectedly high probability of at least two people having the same birthday in a randomly selected group. In fact, by the so-called pigeonhole principle, the probability approaches 1 when the number of people reaches 367. Surprisingly this probability exceeds .99 with only 70 people, and over .5 with as low as 23 people. One of the main applications of this paradox is in the area of cryptographic attack called the birthday attack, which uses this probabilistic model to reduce the complexity of finding a collision for a hash function.

Therefore, we will start with short remarks on cryptography. The term cryptography is known as the art of securing digital information by transforming it into an unreadable format. This is fundamentally performed through the development of more sophisticated hash functions. Computing-wise, hashing in cryptography is a one-way operation that transforms a stream of data into a more compressed form called a message digest. All of the message digests or hash values generated by a given hash function have the same size regardless of the size of the input value.

The goal of the scheme is to decrease the probability of detecting the hash and thus making it hard, if not impossible, to hack protected data. The Birthday Problem and its underling theory would provide a vehicle to achieve this goal and beyond via the cryptographic hash functions.

[1] This is an abbreviated version. The full paper, including simulation results may be available upon request from the author.

INTRODUCTION

The Birthday problem is a fundamental problem, which continues to generate a vast amount of research in a variety of business analytics, economics, engineering, and computer sciences disciplines. This research is heavily statistical, mathematical, and operational in nature. To make this first reading easy to digest by a non-technical readers we tend to avoid mathematical derivations. However, we also intend to apply mathematical theory of Computer Science coupled with Management Sciences' techniques by the time of Camera Ready submission to achieve the goals and objectives set forth in the next category.

Goal. The primary objective of this research is to investigate the derivation of some quite important forms of Birthday Paradox, and its various collisions. Additionally, to survey significant application areas in of Birthday Paradox and the fundamental role it plays in information security. This seems to be a key in the development of modern secured and reliable online business Data Mining and Warehousing. A more elaborate description would require use of technical terms and mathematical derivations which will be omitted at this point in order to keep this paper simply understood by non-technically oriented reviewers, while they will be included in the final work.

Methodology and Anticipated Outcomes. By taking the opportunity of exploring the significance of Birthday Attack and Birthday Paradox, we aim to stress the use of computational techniques and theories as an efficient and effective means of analyzing the birthday collisions and signal the use of classical statistical problems such as Birthday Paradox to fine-tune design schemes for information security.

The latter is achieved by developing functional equations that enables one to approximate the probability of birthday collisions under various scenarios and validate results through simulation models which will be developed. This research should provide a new vehicle for other researchers to pay closer attention to Birthday Attack and its wide applications and eventually lead to development of more sophisticated techniques in information security. Finally, we hope readers will strive to further explore the use of computational scheme and theory such as those we present to tackle data security problems and in particular discover yet unknown applications of the birthday collisions and extend its use in all branches of Information security.

APPLICATIONS

The mathematics of the birthday problem is used in the birthday attack, a brute-force cryptographic attack against hash function problems. These hash functions are either well-defined procedures or mathematical functions which convert large amounts of data into a small, single-integer datum, called a hash, hash value, hash code, hash sum, or checksum. This is done in order to speed up the looking up of items in a database. However, since there are a limited amount of hashes, there can be many collisions of hashes. This is the base on which the birthday attack operates. In informational security purposes, cryptographic hash functions exist where the data is taken and converted into a fixed-size bit string. An ideal cryptographic hash functions, must be easy to compute a hash value for a message, infeasible to create a message with a given hash, infeasible to modify a message without changing the hash, and infeasible to find different messages with the same hash. With this functional characteristics, any tiny changes in the input will lead to a drastically different output. The goal for the attack is that for two inputs of $x_1 \neq x_2, f(x_1) = f(x_2)$.

Now, the variable H is the number of values, and in the usual birthday problem it is set to 365 days in a year. In fact, in the context of birthday problem, let $p(n; H)$ denote probability of no collision (or no birthday match) where H is the number of days in a year and n denotes the selected sample size.

$$p(n; H) = \frac{P_n^H}{H^n} \quad (1)$$

Where P_n^H is the number of permutations of the occurrence of the event and H^n is the cardinality of the sample space of the event. We can derive interesting results, based in part on using Stirling's approximation of $n!$,

$$n! \approx \left(\frac{e}{n}\right)^n \sqrt{2\pi n}$$

using Taylor's series expansion of the logarithms, and convergence of $\lim_{\mathrm{H}\to\infty} \left(1 + \frac{\mathrm{n}}{\mathrm{H}}\right)^{\mathrm{H}} \longrightarrow e^n$, for large values of H. This leads to:

$$\ln(1 - p(n; H)) \approx \ 1 - e^{-\frac{n(n-1)}{2H}} \cong 1 - \ e^{-n^2/(2H)} \quad (2)$$

We further simplify the expression to obtain a closed form solution to the expected number of values before finding a collision. To that end, let define $n(p; H)$ to be the least number of values to choose such that the probability for finding a collision is at least p. Using the notation above, we can arrive at:

$$n(p; H) \approx \ \sqrt{-2H\ ln(1 - p)}$$

Note that since $p < 1,\ -2H \ln(1 - p) > 0$. So for the value of $p = .50$, the expected minimum value that has to be chosen before finding the first collision is greater than

$$n(.5; H) \approx \ \sqrt{\ln(4)\ H} \quad \text{or} \quad n \ \approx \ 1.17741\ \sqrt{H} \quad (3)$$

Let $H = \ 365$, we have, $n \ \approx \ 1.17741\ \sqrt{365} = 22.494$ or 23 for p to exceed .50 as shown in figure below. In terms of actual use of this birthday attack, suppose a hacker plans to trick a user into signing a fraudulent contract. So, he creates both a fraudulent and fair contract. Then, on the fraudulent contract, he makes minor change, say changes the contract by adding or deleting commas, having synonyms of the words, etc. Eventually, in theory, with enough fraudulent contracts, there will exist one which has the same hash value as the fair one. He then gets the user's signature on the fair one and attaches it to the fake one, which both have the same hash value. However, this birthday attack can be made unfeasible by increasing the hash value output size until it is unfeasible to find a collision.

EXPLORING VARIOUS COLLISIONS

In the sequel, we will explore variations of this basic result and expand the derivations for more interesting cases involving multiple pairs, trios, four-tuples and near collision probabilities and will attempt to present, where possible, closed form solutions for the distribution functions. It must be noted that this is a challenging area in computer science and many questions remain still as open problems. Where such a closed form solution is not available we attempt to apply simulation techniques to approximate the collision probabilities.

The classical case: Let's return to probability distribution of finding at least one collision or $f(n;H) = 1 - p(n;H)$ for $p(n;H)$ as given in (1) and that of exactly one collision

$$P(Exactly\ one\ collision) = C_1^H C_2^n \left(\frac{1}{H}\right)^2 \frac{P_{n-2}^{H-1}}{H^{n-2}} = C_1^H C_2^n \frac{P_{n-2}^{H-1}}{H^n} \quad (4)$$

Where the notation of the term P_{n-2}^{H-1} in (4), denotes permutation of $(H-1)$ choose $(n-2)$.

Multiple Pairs Collision: In our discussion a pair refers to collision or matching two. We can reason its extension of single match to multiple, say $k, k = 1,2,\cdots,c$ matches and denote the probability of such k pairs collision by $P(X=k)$ as follows:

$$P(X=k) = C_k^H\ C_2^n\ C_2^{n-2} C_2^{n-4}\ \cdots\ C_2^{n-2(k-1)} \left(\frac{1}{H}\right)^{2k} \frac{P_{n-2k}^{H-k}}{H^{n-2k}}$$

This in turn leads to$f(x;k)$, the density function for probability of k pairs of collisions:

$$f(x;k) = C_k^H\ P_{2k}^n\ \frac{P_{n-2k}^{H-k}}{2^k\ H^n} \quad (5)$$

Letting again H=365 in (4), we have the expression below for the density function, as depicted on the right panel of the above figure.

$$P(Exactly\ one\ collision) = \frac{n(n-1)P_{n-2}^{365-1}}{2\ (365)^{n-1}}$$

Triple collision: Computing the probability of at least one triple collision which is also referred to as a trio is not a straightforward task. However, we could extend our previous reasoning to first derive a density function for exactly one trio collision,

$$P(Exactly\ one\ of\ trio\ collision) = C_1^H C_3^n \left(\frac{1}{H}\right)^3 \frac{P_{n-3}^{H-1}}{H^{n-3}} = C_1^H C_3^n \frac{P_{n-3}^{H-1}}{H^n} \quad (6)$$

Multiple pairs of trio collision: In our discussion, a trio refers to collision or matching three. We can reason its extension of a single trio to multiple, say $k, k = 1, 2, \cdots, c = \lfloor n/3 \rfloor$ pairs and denote the probability of such k pairs of trio collisions by $P(X=k)$ as follows:

$$P(X=k) = C_k^H\ C_3^n\ C_3^{n-3} C_3^{n-6}\ \cdots\ C_3^{n-3(k-1)} \left(\frac{1}{H}\right)^{3k} \frac{P_{n-3k}^{H-k}}{H^{n-3k}}$$

This in turn leads to$f(x;k)$, the density function for probability of k pairs of trio collisions:

$$f(x;k) = C_k^H\ P_{3k}^n\ \frac{P_{n-3k}^{H-k}}{3^k\ H^n} \quad (7)$$

General k-tuple collisions: The probability of *k*-tuple collision may be derived in the same manner as we derived for the case of one pair, multiple pairs, and trios. Let X_k denote collision of size k, or a k-tuple collision, for $k, k = 1, 2, \cdots, c$,

$$P(X_k = 1) = C_1^H\ C_k^n \left(\frac{1}{H}\right)^k \frac{P_{n-k}^{H-1}}{H^{n-k}} = C_1^H\ C_k^n\ \frac{P_{n-k}^{H-1}}{H^n} \quad (8)$$

This is very similar to (4) or that of (6) above for a pair or a trio collision, if we let $k = 2, 3$, respectively in (8). It turns out, as we show in the sequel, that we can efficiently compute these probabilities for large values of *k*. (See plots of collisions density functions of length k=1:6 in Appendix A).

Distribution function for collisions - The derivation for the distribution functions of various collisions requires more computational steps. For the case of a single pair or one collision, we will have to

compute probability of finding at least one collision or $1 - F(X_k)$, using notation introduced in (8) and for $k = 2$. Additionally using (1) above, we can obtain this probability easily,

$$P(X_2 \geq 1) = 1 - F(X_2) = 1 - p(n; H) = 1 - \frac{P_n^H}{H^n} \quad (9)$$

For the case of a trio, this is obtained as follows:

$$P(X_3 \geq 1) = 1 - \sum_{k=0}^{\lfloor n/2 \rfloor} C_k^H \, P_{2k}^n \, \frac{P_{n-2k}^{H-k}}{2^k \, H^n} \quad (10)$$

Where, this is computed by the complement of the case of all pairs of $k, k = 0,1, \cdots, \lfloor n/2 \rfloor$. For the birthday problem, $H = 365$, the density is shown for (10) in Figure 3 below.

By the time of Camera Ready submission, we intend to follow up with derivations of more complicated collision scenarios and present their simulated approximations with various charts and plots to provide a full exploration of this interesting subject.

CONCLUDING REMARKS

Clearly, this research fundamentally relates to underlying data security, a vital factor of modern design scheme in online communication and has the potential for further improvement in protecting vital personal and organization data. The birthday problem is an interesting model, which attempts to calculate the probability of various birthday collisions in 365 days in a year. Although 365 different days seems like a large amount, as shown earlier, only 23 people are required so that the probability of at least one collision exceeds 50%, hence the reason for the word paradox. There can be many different variations on the original problem, as we have shown a sample above, and many other problems can be solved using the same method as presented above. The birthday attack also demonstrates this, as it computes the expected number of attempted values before finding a collision.

Overall, the birthday problem is quite an interesting research topic and deserves more in-depth analysis to discover its many interesting properties and yet unknown applications areas.

REFERENCES

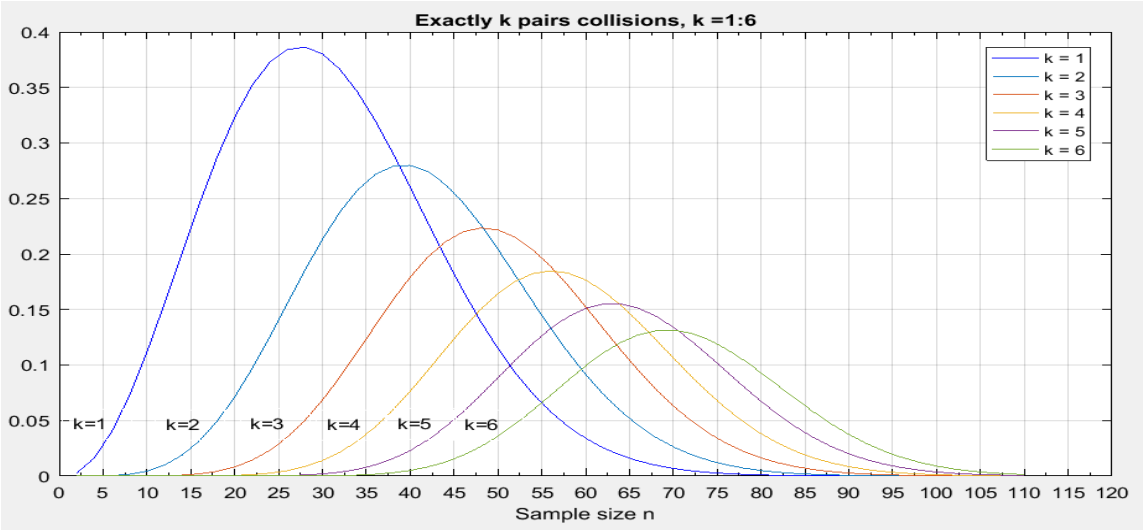
[1] Gleich, David, “Birthday distribution”, David Gleich’s Notebook. Stanford University, 2010.
Source: http://www.stanford.edu/~dgleich/notebook/2009/04/birthday_distribution.html

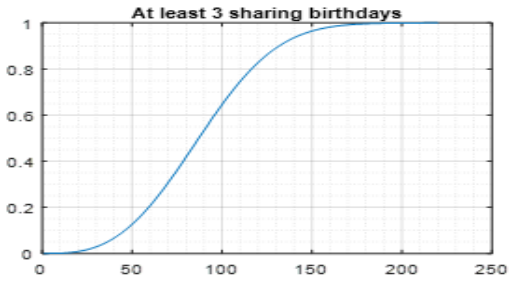
[2] “Cryptographic Hashes”, CS461/ECE422
Source: https://wiki.engr.illinois.edu/download/attachments/183272958/cryptohash.pptx?

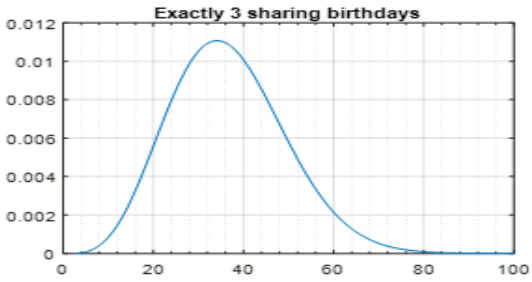
[3] Stamm, Stephanie “Hash Functions and the Birthday Attack”. United States Naval Academy. Midshipman 1/C. April 23, 2010. Source:
http://www.dean.usma.edu/departments/math/courses/ma498/SASMC/slides/Stamm%20USNA.pdf

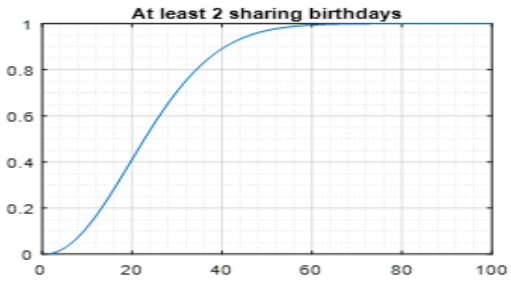
[4] Zimmerman, Neetzan. “Infographic Illustrates Most Common Birthdays, Baby-Making Days”. Gawker, 2012. Source:

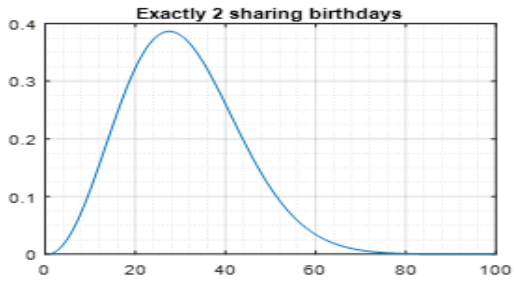
[5] Sun, Jared, “The Birthday Paradox: A Simple and Surprising Probability Problem”
August, 2011, Cluster 6

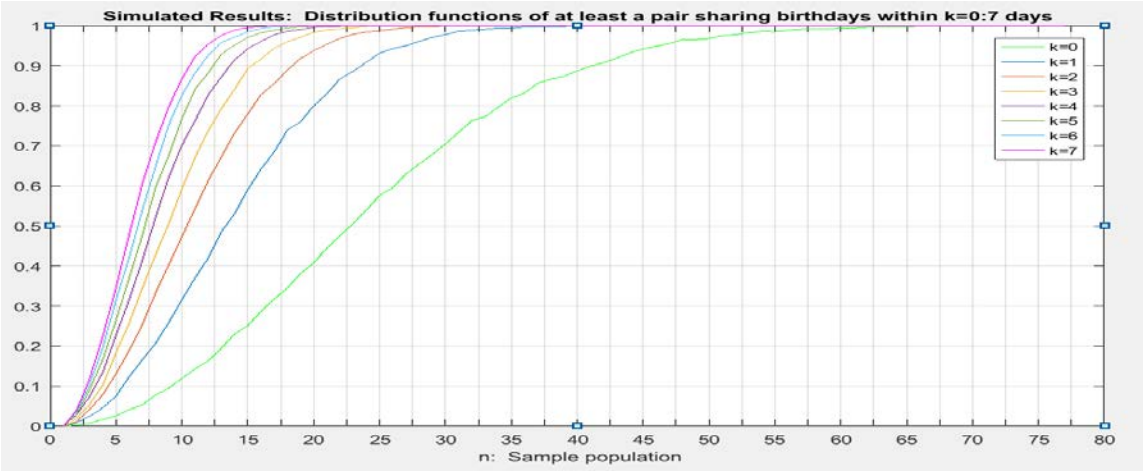

Exactly k pairs collisions, k =1:6
k = 1
k = 2
k = 3
k = 4
k = 5
k = 6
k=1
k=2
k=3
k=4
k=5
k=6
Sample size n


At least 3 sharing birthdays


Exactly 3 sharing birthdays


At least 2 sharing birthdays


Exactly 2 sharing birthdays


Simulated Results: Distribution functions of at least a pair sharing birthdays within k=0:7 days
k=0
k=1
k=2
k=3
k=4
k=5
k=6
k=7
n: Sample population