2. Sweet32

After gaining an overview about the general concept, this section will focus more on the security aspects with the example of Sweet 32, a collision attack on 64-bit block ciphers in TLS and OpenVPN.

2.1 Probability of collisions

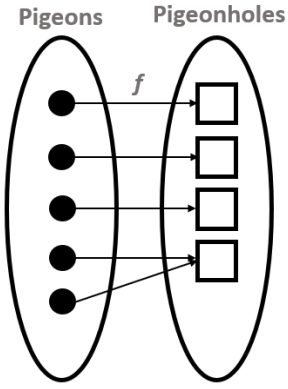
A collision describes the infrequent event of two inputs resulting in the same ciphertext which can be used to decryption every other ciphertext in the communication. To better understand what makes collision attacks possible, the likelihood of an event to occur must be determined, therefor basic knowledge of probability theory is necessary.

2.1.1 Probability

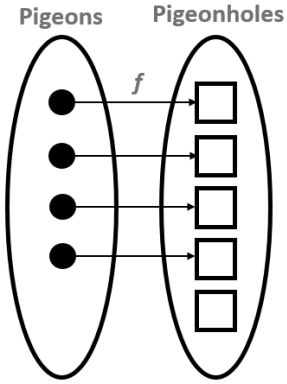
The possibility of a specific outcome can easily be calculated by taking an event and dividing it by the number of every possible result. To give an example, the possibility of someone having the 01 January as their birth date would be $\frac{1}{365} \approx 0{,}0027397260$. The event of one specific date is divided by a year, the 365 days make up every possible outcome, the consequence is a at first glance seemingly low probability. It is also important to note that any other day in the year does have the same probability and 366 days, as explained in 2.1.2, are needed to ensure that two people have the same birthday.

2.1.2 The Pigeonhole Principle

To make sure a collision will happen we need a probability of 100%, this can be guaranteed by the Pigeonhole Principle [1]. If $n + 1$ objects are put into n boxes, then at least one box contains two objects. This can be visualized by supposing Pigeons and Pigeonholes are finite sets and $f: Pigeons \rightarrow Pigeonholes$ is any function.



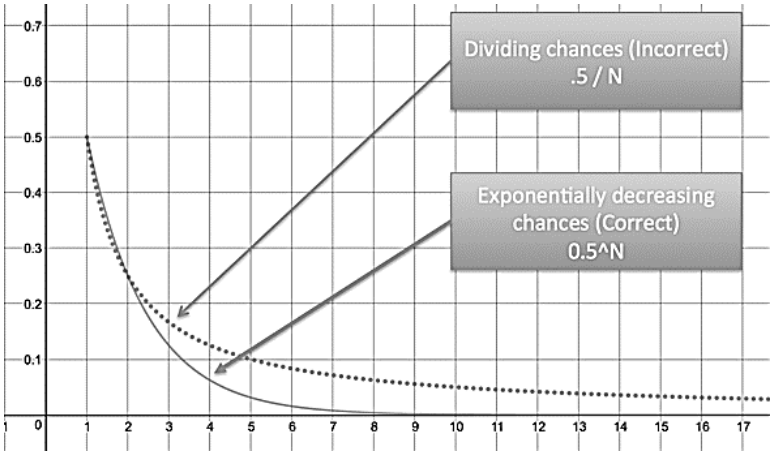
(1.1) |A| > |B| not injective



(1.2) |A| < |B| not subjective

2.1.3 The Problem with exponential thinking

To show that even low probabilities can result in collisions the calculation of coin flips can be used to better understand the difference between linear and exponential functions. A flip can produce two possible outcomes, each with an equal probability of 50%. The correct, exponential approach decreases the output for every additional flip by 50%. Mathematically this can be written as 0.5^n, where n is the number of coin flips. The linear approach would divide the initial possibility by the number of coin flips, $0.5/n$.



(2) Chance of N heads [2]

The graph demonstrates that the numbers differ largely the higher the total of flips, untill the probabilities of both approach zero.

2.1.4 The Birthday Problem

In 1939 Richard von Mises introduced the Birthday Problem as an easy way to visualize human tenderizes to think in a more linear fashion, a more current version can be found in [3]. It states: How many people do we need that at least two of them have the same birth date (without year) with a probability of more than 50%. Counter-intuitive to the average guess of 183, the correct answer is 23. The proof conducted in [4] calculates the probability of the function $p(23)$ which equals the probability of two people having the same day of birth in a group of 23. Since the probability of two people not having the same birthday is easier to calculate, $p(a) = 1 - \overline{p}(a)$ will be solved for. This can be done because p and $\overline{p}$ are the only two possibilities and are also mutually exclusive. To get the overall probability every one of the 23 individuals must be calculated und multiplied independently. The first person has a 100% chance of not having the same birthday as someone else, since he is the only participant. Continuing in this fashion the second person not having the same birth date as the first, is multiplied with the third person, not having the same birth date as the first and the second, this will continue till the 23 event is reached. To get the probability of two birthday occurring the outcome has to be subtracted from a 100%.

$$\overline{p}(23) = \frac{365}{365} * \frac{364}{365} * \frac{363}{365} * \dots * \frac{343}{365} \approx 0{,}49$$

$$p(23) \approx 1 - 0{,}49 \approx 0{,}51$$

The assumption was correct and can be generalized $p(r)$ to calculate every other probability of multiple events in relation to the outcomes. Taking the Pigeonhole Principle in account the number of people needs to be smaller or equal to the days in a year. As before ever individual r must be multiplied together making the limits 365 and $365 - (r - 1)$.

$$\overline{p}(r) = 1 - \left(\frac{365}{365} * \frac{364}{365} * \frac{363}{365} * \dots * \frac{365 - (r-1)}{365}\right)$$

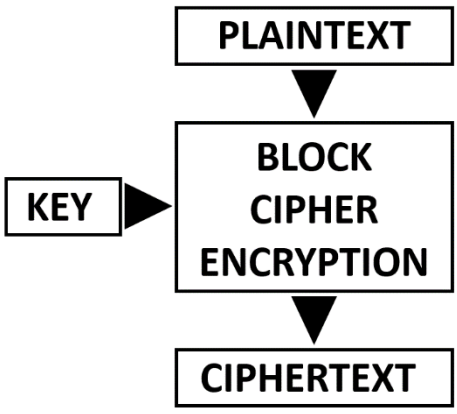
$$= 1 - \left(\frac{365 * 364 * \dots * (365 - r + 1)}{365^r}\right)$$

$$= 1 - \frac{365!}{!\,365^r * (365 - r)}$$

With the help of this function, we can calculate the number of people needed to achieve a probability of 99% of two people having the same birthday, which is 70, approximately $\frac{1}{5}$ of all possible outcomes.

2.2 The Birthday Attack

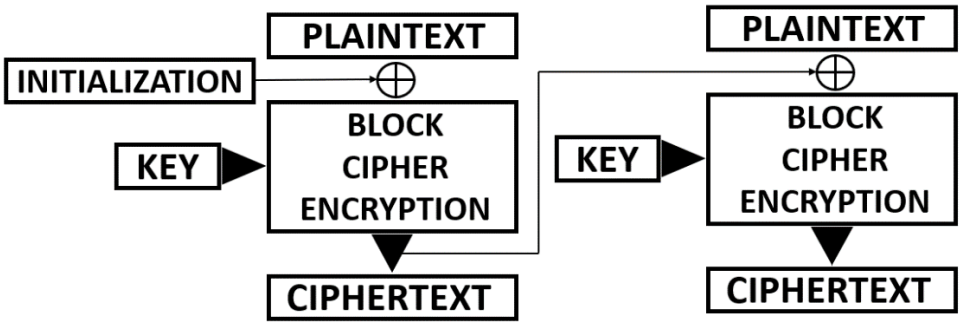
With these mathematical basics a common collision exploit called Birthday Attack can be elucidated. Despite common perception not only the key length but also the block size, which determines the amount of data that can be encrypted under the same key, defines the security of a block cipher. A cryptographic hash function should resist attacks on its preimage which is the set of possible outputs. In the example of a 64-bit block cipher a preimage attack, that tries to find a message that has a specific hash value, should not generate a match for 2^{64} queries.



(3) Simple block cipher

2.2.1 Cipher Block Chaining

The concept of encrypting large amounts of text has already been established in 3.4, a different method besides ECB is Cipher Block Chaining which is one of the more common modes and extensively described in [5]. Plaintext is split into 64-bit blocks and separately encrypted. The blocks are composed of the plaintext and the ciphertext of the last block. These two components are merged with a simple XOR, for the first block a random initialization ciphertext is generated and shared with the receiver. At the beginning of every communication between sender and receiver a shared key is exchanged that is used throughout the whole session to generate the Block Cipher Encryption.



(4) Cipher Block Chaining

The longer the initial cleartext the more events are needed that can lead to more collisions, each 64-bit block can be viewed as a person that would increase the possibility of two of them having the same birthday.

2.2.2 Birthday bound

The previous observations about exponential functions in connection with collisions can now be modified to incorporate the increase of events and to result in the approximate number of guesses needed. Since it is a common practice in Cryptology to estimate this bound, the exact function will only be explained briefly a more in-depth description can be found in [6]. The goal is to find two inputs of $x_{1 \neq x_2,} f(x_1) = f(x_2)$, in addition the number of values H and the probability of a collision p. are given.

$$r(p, H) \approx \sqrt{-2H * \ln(1-p)}$$

This can be approximated by assuming that if a value can take on N different forms it is also expected to collide after $\sqrt{N}$ random elements. Demonstrated by [7] by choosing k elements there are $r * \frac{(r-1)}{2}$ pairs of elements each with a $\frac{1}{N}$ probability of generating a collision. The chance of a pair with equal values is nearly at $r * \frac{(r-1)}{2*N}$, when r $\approx \sqrt{N}$ the probability is about 50%. The resulting number is commonly reverted to as the so-

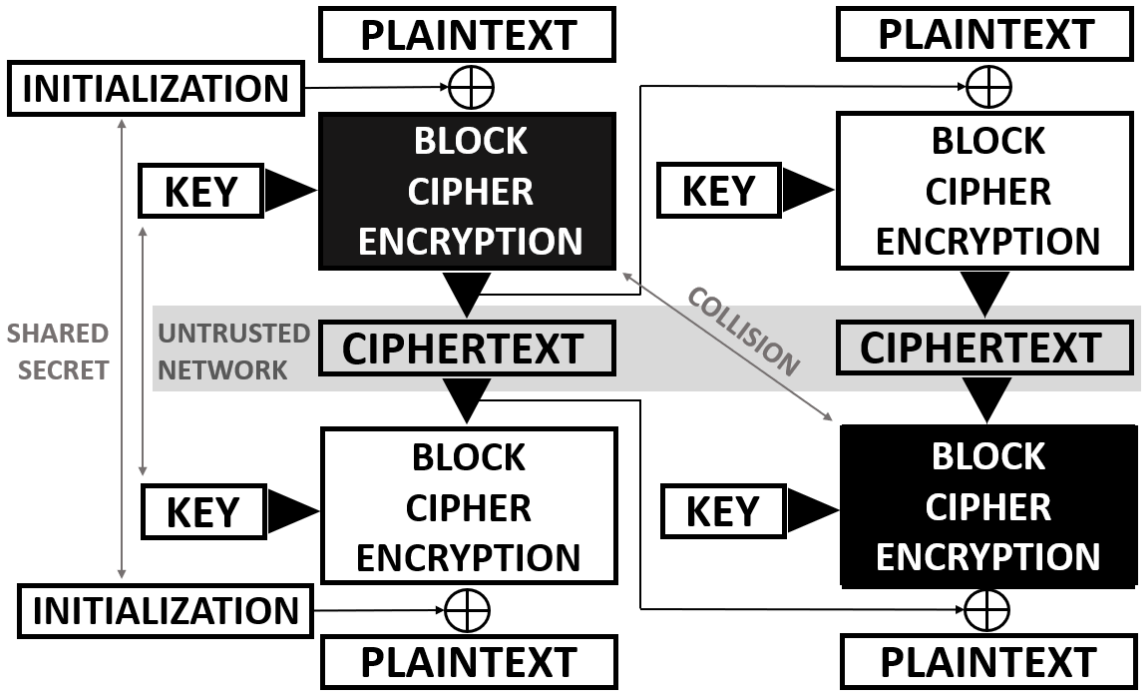
called birthday bound, for a 64-bit long cipher this would produce a bound at $\sqrt{2^{64}} =$ 4294967296.

2.3 Application of Sweet32

A Sweet32 Attack consists of two parts, first data must be collected, this usually happens trough a man-in-the-browser or man-in-the-middle attack. An attacker as described in [8] intercepts transactions between two parties who seemingly communicate directly. This allows the collection of large amounts of data. For a key length of 64 bit and a mode of operation that is birthday bound 32GB would be sufficient.

$$2^{Blocksize/2}\ Queries * Blocksize\ = \frac{2^{32} * 64\ bit}{3072} = 32\ GB$$

For this attack to work the communication needs a fixed key that will not changes during the capture of encrypted packages with tcpdump and the extraction of cipher blocks with custom code. Both HTTPS and OpenVPN queries are sent in separate records which contain the plaintext and additional information like package numbers, padding, Mac and so on, which makes it easy to know which plaintext block corresponds each cipher block. Secondly collisions are produced by calculation the XOR value of randomly guessed plaintexts with every captured ciphertext and by comparing it with the next ciphertext. This process takes about 19 hours for 64bit block ciphers with an average of $\sqrt{2^{64}}$ queries in comparison to years when 2^{64} queries would be necessary. With the collision of two block cipher encryption blocks the transmission key can easily be calculated and used to decrypt every other plaintext of the communication.



(5) Sweet32 attack on an untrusted network

2.4 Conclusion

Sweet32 was the first concrete attack on mainstream Internet protocols which was followed by a suite of exploits using the same underlaying principles like BEAST and NOMORE [9]. Since the discovery from Karthikeyan Bhargavan and Gaëtan Leurent in 2016 the use of most 64-bit block cipher is not to be recommended anymore, algorithm like Triple-DES have since been degraded to medium level of security. There is no divined way to mitigate this kind of attacks. Recommendations to minimize the risk are:

1. Configurations should be set to prefer 128-bit ciphers like AES.
2. TLS libraries should limit the length of TLS sessions with TLS renegotiation.
3. OpenVPN users should force frequent rekeying with reneg-bytes 64000000.

About 1% [9] of modern systems and browsers still use 64-bit block ciphers like 3DES or Blowfish, especially legacy and company environments are vulnerable, with hacking tools like Metasploit featuring an easy way to execute such attacks it is still an ongoing thread.

References

[1] Chen Beifang. The Pigeonhole Principle. 2005. URL: https://www.math.ust.hk/~mabfchen/Math391I/Pigeonhole.pdf.

[2] Kalid Azad. Understanding the Birthday Paradox. 2006. URL: https://betterexplained.com/wp-content/uploads/2017/01/birthday-paradox-coin-flip-odds.png.

[3] Jan Crusius, Anna Gast, and Joris Lammers. Correcting misperceptions of exponential coronavirus growth increases support for social distancing. In PNAS, 117 (28) 16264-16266, pages 1-6, June 24, 2020. URL: https://www.pnas.org/content/117/28/16264.

[4] Trevor Fisher, Derek Funk, and Rachel Sams. The birthday problem and generalizations. Carlton College, Mathematics Comps Gala, May 21st, 2013.

[5] Kumar Gulshan. Singh Balwinder and Singh Mandeep. Application of AES-128 Cipher Block Chaining in WSNs. In National Conference on Computers, Communication & Controls (N4C11), pages 3-4, April 2011.

[6] A. Mahyar Amouzegar, Khosrow Moshircaziri and Fahimeh Rezayat. 1 Exploring the birthday attack / paradox. 2017. URL:http://wdsinet.org/Annual_Meetings/2017_Proceedings/CR%20PDF/cr88.pdf

[7] Niels Ferguson, Bruce Schneier and Tadayoshi Kohno. Cryptography Engineering: Design Principles and Practical Applications. John Wiley & Sons P&T pages 63-76, October 2015.

[8] A. R. Chordiya, S. Majumder and A. Y. Javaid, "Man-in-the-Middle (MITM) Attack Based Hijacking of HTTP Traffic Using Open Source Tools," 2018 IEEE

International Conference on Electro/Information Technology (EIT), Rochester, MI, 2018, pp. 0438-0443, doi: 10.1109/EIT.2018.8500144.

[9] Karthikeyan Bhargavan and Gaëtan Leurent. 2016. On the Practical (In-) Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16). Association for Computing Machinery, New York, NY, USA, 456–467. DOI:https://doi.org/10.1145/2976749.2978423