What hash functions are? and How could they be attacked?

Hash functions are methods to encrypt any type of file/message/transaction, called "pre-image", in a pseudorandom fixed-length string, which is known as "image". The pseudo"**random**" property means that the value doesn't seem to be logically related to the original message and even if the original message changes slightly, the hashed value changes completely, however, it is "**pseudo**"random because the hash function of a particular message will give always the same image.

The aims of cryptography in general, and in hash functions in particularly, is to show integrity of a document or file, to prove identity, for example on digital fingerprints, and to provide secrecy such as in password storage or in auction bidding.

There are several properties that hash functions should comply to be "cryptographic" functions as explained by Devadas (2015):

1. One-way functions: These functions are easy to compute in one way (from the original message to the hashed value) however, it should be very hard to compute the other way around (from the hashed value to the original message).
2. Collision Resistant: It should be hard or infeasible to see an x and x' ($x \neq x'$) that will be hashed to the same value ($H(x)=H(x')$).
3. Target Collision Resistant: For a given x, it should be hard or infeasible to find an x' ($x \neq x'$) which will be hashed to the same hashed value ($H(x)=H(x')$).

If the hash function loses, at least, one of these properties, it ceases to be cryptographic. This can happen when the hash function suffers an attack. There are 3 main types of attacks: Pre-Image Attack (attack the first property), Collision Attacks (attack the second property) and Second Pre-Image attack which attacks the last property.

Muller (2004) describes the different types of attacks and different approaches for the same type of attack for the MD2 hash function.

While discussing MD2 attacks, it is necessary to explain briefly the MD2 architecture: In the first step, padding is added to the original message to complete n blocks of 16 bytes. Then, a checksum is calculated by XOR operation and substitutions from an S list of 256 values and it is added to the message. In the last step, a digested message of 48 bytes is initialised with 0s, and the compression function calculates each of the values, again using the XOR operations and the substitution list, this function is done multiple times to increase the complexity of the algorithm. The first 16 bytes (128 bits) of the digested message represent the hashed value.

Muller (2004) sustains that MD2 is susceptible to collision attacks due to the initialisation of the digested message with 0s. However, he claims that this type of attack is not of concern to the public as they do not threaten the security scheme of certificates where it is used. On the other hand, the great concern comes from the potential Pre-Image attacks and Second Pre-Image attacks as the attacker could forge with an alternative certificate that would be hashed to the same image. Muller (2004) presents how these potential attacks will work and how much faster they are in comparison to the "brute force" algorithm, which means covering all the possibilities.

The Pre-Image attack consists of finding a pre-image (x') when the image (H(x)) is known, for any valid x, hence altering the one-way-ness of the hash function. Muller (2004) describes several algorithms of this type of attack. There are some algorithms that attack only the compression function and others can go further to recreate the original message. Out of the algorithms that attack the compression functions, one is faster and has more flexibility than the other. In the faster

algorithm, the attacker, in the first step, makes initial assumptions. Later, he makes a big number of tables filled mostly with random values, this is done for two types of tables, one represents the Digested Message table and the other, the initial message table. From them, he creates another table by doing some operations between them. In the last step, he finds the collisions and performs additional calculations. Muller (2004) understands that this attack can take some time but there are ways to speed up the algorithm by eliminating options using an early check. This algorithm should take ~ 2^{73} computations which is much faster than the "Brute Force" which is 2^{128} computations. In the case the attacker wants to get to an original message, then, he needs to do additional operations which will attack the checksum as well. This algorithm is slower than only attacking the compression function, but it is still faster than the "Brute Force". Hence, this is a feasible attack, and, as the title of Muller (2004) explains, "The MD2 hash function is not one-way".

The Second Pre-image attack comprises in finding and a pre-image (x') that will hash to the same image as the one obtained with a known pre-image (x). Muller (2004) explains that, if the MD2 did not have the checksum, this task could be straightforward similarly to the algorithms used for Pre-image attacks. However, the MD2 hash function has the checksum and, at the point where the paper was written, there was not a Second Pre-image attack published. He suggests that there is a possibility to perform a Second Pre-Image attack and recreate an x' but it would be very constrained in its length and its content. This means that an attacker that would like to forge a certificate, will not be able to produce the fake certificate he wants but the one that could hash to the same value. In other words, it would be unfeasible for an attacker to do this.

In the same way that MD2 could be easily attacked, and it has lost its key properties to be cryptographic, later developed hash functions such as MD4 and MD5, SHA-1 have also been attacked. There are several published papers (Aoki et al.,2009) (Sanadhya et al., 2008) (Guo et al., 2010) that show partial attacks on the SHA-2 family of hash functions, however, they are still considered to be cryptographic. The latest hash function family of the SHA series, SHA-3 (512 bits) is the most secured hash function. According to Arias D. (2019), Google recommends using SHA-2 or SHA-3. On the other hand, to improve the security of passwords or documents, a combination of hash functions, encryption methods and salting methods are strongly recommended.

References:

Arias, D., 2019. *How to Hash Passwords: One-Way Road to Enhanced Security*. [online] Auth0 - Blog. Available at: <https://auth0.com/blog/hashing-passwords-one-way-road-to-security/#Limitations-of-Hash-Functions> [Accessed 22 May 2021].

Aoki K., Guo J., Matusiewicz K., Sasaki Y., Wang L. (2009) *Preimages for Step-Reduced SHA-2*. In: Matsui M. (eds) Advances in Cryptology – ASIACRYPT 2009.

Devadas, S., 2015. *Lecture 21: Cryptography: Hash Functions | Lecture Videos | Design and Analysis of Algorithms | Electrical Engineering and Computer Science | MIT OpenCourseWare*. [online] Ocw.mit.edu. Available at: <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-design-and-analysis-of-algorithms-spring-2015/lecture-videos/lecture-21-cryptography-hash-functions/> [Accessed 22 May 2021].

Guo J., Ling S., Rechberger C., Wang H. (2010) *Advanced Meet-in-the-Middle Preimage Attacks: First Results on Full Tiger, and Improved Results on MD4 and SHA-2*. In: Abe M. (eds) Advances in Cryptology - ASIACRYPT 2010.

Muller F. (2004) *The MD2 Hash Function Is Not One-Way*. In: Lee P.J. (eds) Advances in Cryptology - ASIACRYPT 2004. ASIACRYPT 2004.

Sanadhya S.K., Sarkar P. (2008) *New Collision Attacks against Up to 24-Step SHA-2*. In: Chowdhury D.R., Rijmen V., Das A. (eds) Progress in Cryptology - INDOCRYPT 2008.