University of London

MSc Data Science and Artificial Intelligence

DSM070 Blockchain Programming Coursework

Cryptographic Hash Functions

Program: 50% - Essay: 50%

Table of Contents

Table of Figures

Introduction:

Cryptography is considered as the art and science of hiding messages or information. There are two types of cryptography classic and modern. Classical cryptography was first used to send messages by Ancient Egyptians, government communication, and protect information during the war (Kessler, 1998).

In the last few decades, with the introduction of computers and internet usage worldwide, privacy and security have become a major concern. Therefore, modern cryptography has arisen to protect electronic data from unauthorized entities using the terms encryption and decryption. Accordingly, cryptography has evolved into a science of converting messages to secret forms called ciphertext and transforming them back to their original form (Katz and Lindell, 2019).

There are four main purposes when utilizing cryptography: confidentiality or privacy, authentication, non-repudiation and authentication. Those are the desired goals that are not meant to be all achieved by any cryptographic system (Kessler, 1998).

This essay will focus on describing Cryptography and its various types. Moreover, it will cover the hash functions in cryptography with a deep dive into MD2 as one of the early examples of hash functions and the steps to generate a digested message. Lastly, the essay will concentrate on how hash functions are attacked and the different kinds of attacks.

Types of Cryptography:

Cryptography can be classified into several types (public key, private key and hash function). Public key cryptography, also called asymmetric, uses two different keys. One key is for encryption, while the other is for decryption. Whereas in the private key type, which is called symmetric, a single key is used for encryption and decryption (Kessler, 1998).

While encryption is a two-way function, hashing is a one-way process with no key because it is computationally infeasible for the original message to be recoverable (to decrypt it) from the ciphertext. It is used to transfer the text into an encrypted version with a fingerprint to ensure the integrity of the message (Broemeling, 2011).

Secret Key Cryptography:

Public Key Cryptography:

Hash Functions:

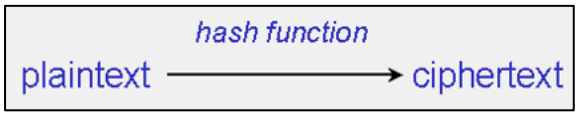

Figure 1: The Cryptography Types

(Kessler, 1998)

Cryptographic Hash Functions:

Hash functions take any input size and transfer it to a fixed output length. It is also known as one-way encryption or message digest (MD) (Mironov, 2005). It can have two types, keyed and un-keyed hash functions. The first type is when a secret key is used, also called Message Authentication Code (MAC). On the other hand, the un-keyed hash functions or just hash functions when a secret key is not used and referred to as Manipulation Detection Code (MDC) (Preneel, 1999) (Sobti and Ganesan, 2012a).

The cryptography hash functions can be secure if it takes any length of input and produces an output of length n bits:

$H: \{0, 1\}^* \rightarrow \{0, 1\}^n$

There are many hash function algorithms such as MD family, SHA family and other algorithms like HAVAL, RIPEMD, Whirlpool and Tiger. MD family was developed by RSA Labs. It takes an arbitrary length of input and produces a fixed 128-bit hash value. MD2 is considered the oldest hash function from the MD family and was proposed first by Ron Rivest in 1989. It was mainly developed for the digital signature. Then was swiftly followed by MD4 and MD5 (Kessler, 1998).

MD4 was broken with just 2^6 inputs, whereas MD5 was broken with 2^{37}. Broken means that an attacker can find two inputs for the same output, which is one of the methods hash functions are attacked. This essay will discuss attacks in more detail (*Cryptography: Hash Functions*, 2016).

As a result, in 1993, the SHA family, which refers to the Secure Hash Algorithm, was introduced by the National Institute of Standards and Technology (NIST) as a new hash function. Examples of the SHA family are SHA-0, SHA-1, SHA-2 and SHA-256 (Stevens, 2012).

Hash Functions Properties:

The desirable properties of hash functions can be summarized into:

- **One-way function or preimage resistance**, meaning it is computationally infeasible to invert the original input from the digested result
- **Computable**, whatever the data, the computations are easy to generate the hash value
- **Weak collision resistance** or **second preimage resistance**, given an input, it is hard to find a different input with the same hash value
- **Strong collision resistance**, it is computationally hard to find two unique inputs that can produce the same digest
- **A random oracle** assumes that if the input has been seen before, it returns a deterministic output while flipping the coin for the new input. However, this model is not easy to be achieved in practice as it requires ample storage. Therefore, a pseudo-random function is recommended in practice
- **Avalanche effect or puzzle-friendliness**, any minor changes to the input will produce a completely different hash value
- **Non-malleability**, given a hash value for a string, it is infeasible to generate a different hash value for a related input

(*Cryptography: Hash Functions*, 2016) (UoL)

Hash Function Applications:

Hash functions can be applied to different applications such as:

- **Message Authentication** verifies that the message is created and signed by the owner of the secret key
- **Privacy/ confidentiality,** such as a digital signature which protects the data from being seen by unauthorized entities
- **Integrity** ensures that no alteration has been made to the original message. The usage of this application can be in detecting any changes in the system files. Moreover, it detects any corruption when downloading files.
- **Password identification** is a method to transfer the original passwords from plaintext to hash values to reduce the risk of storing the original passwords.
- **Non-repudiation** is a way of verifying and authenticating the message sender

(Preneel, 1999)

MD2 Algorithm:

MD2 (Message Digest 2) algorithm can be described as stages. The first stage is to append the padding message. "x" bytes of size "x" is added to the message. Subsequently, the padded

message length modulo 16 results in zero (or until the padded message reach a length multiple of 16-bytes)

Following that, an algorithm calculates the checksum and appends it to the padded message, which will result in adding more blocks (n+1) where n is a 16-byte block number.

Checksum works by calculating the current message position XOR the previous checksum value. Next, that output is taken to find the corresponded value in the "S" table. Afterwards, the calculated checksum position XOR the current checksum (Adel, 2017).

After the checksum, the message digest is initialized to zero using a three 16-byte buffer. The next step will generate hashing to process the message in 16-byte blocks. Finally, the digested message is printed, the first 16-byte string, and represented as hexadecimal values (Kaliski, 1992).

Cryptographic Hash Functions Attacks:

Cryptographic hash functions are mathematical algorithms that are heavily utilized across various industries in different applications such as forensic analysis, digital certificates, authentication and the development of many protocols. A hash function typically maps an input message into a bit array of hash or fixed size, usually between 128-512 bits (Muller, 2004).

Hash functions cease to be usable when they are attacked, and this attack results in a break in one of their security properties which are (Preimage, 2nd Preimage and Collision Resistance). In the successful attack scenario, the function is considered to be vulnerable and broken. Hence, it is not recommended to be used. Furthermore, even if a function is not broken, but a variant of that function is successfully attacked, that hash function's overall confidence becomes questionable (Muller, 2004).

There are three main types of valid attacks which focus on the classical properties of a particular hash function, which are the following:

- The first type is the **Preimage attack**, in which for a particular hash function H() and for every message M the attacker will try to generate an input message until a value H is achieved.
- The second type is **2nd Preimage attack**. In this type, the attacker will try different input messages M', which is not equal to the original message M until the value of H(M) is achieved.
- The last type of attack is the **Collision attack**. The attacker, in this case, will try to find two messages which satisfy the condition that the messages are different, but the values of the hash function for each of the messages are equal (Muller, 2004).

Muller (2004) focused in his paper mainly on two kinds of attacks. The first is concerned with the preimage attacks, and the second target the second preimage attack. Moreover, the pseudo-preimage attack was described as the fastest attack on MD2 (in the compression function) with 2^{73} complexity only, with more freedom since many solutions can exist (Muller, 2004).

In some cases, when the checksum is excluded, the preimage MD2 attack (*when the intermediate hash values are given and the goal is to find the relevant message*) is considered straightforward and can be faster than the brute force attack (Muller, 2004).

In general, the success of a cryptanalytic attack on a certain hash function is measured by how much faster it is compared to brute force. For example, the pseudo-preimage attack against the full hash function using the meet-in-the-middle technique has been completed with the complexity of $2^{101,}$ which is faster than the brute force attack of complexity $2^{128,}$ and this has proved that an attack faster than brute force is feasible (Muller, 2004).

On the other hand, Stevens and co-authors (2017) argue that collision attacks take more effort on attack around 2^{63} GPU computation power. While the authors agree with Muller on the speed of collision when compared to brute force attack (Stevens *et al.*, 2017).

The critical point regarding the utilization of MD2 nowadays is the fact that millions of certificates on the internet have been generated with MD2 historically and are still used in particular in public-key infrastructure such as Verisign certificates (Muller, 2004).

In 2004, the MD5 hash function was successfully attacked using Floyd's cycle-finding algorithm technique. By using this technique, the researchers were able to find a collision, meaning generating two input points which produce the same output (Wang *et al.*, 2004).

Other attacks also include birthday attack which the exploits the probability of finnding at least two people with a similar birthday. With this attack, it is possible to find a collision of a particular hash function. Another example is Meet in the Middle Attack which is considered as another variation of the birthday attack. It allows the opponent to construct a message which relates to a specific message digest (Bakhtiari, Safavi-Naini and Pieprzyk, 1995).

The other notable functions are MD5 & SHA-256. While MD2 and MD5 functions have been successfully attacked, SHA-256 is considered a secure function; hence no collision has been found yet. In terms of the block size and digest length. Both MD2 and MD5 are 128 bits. Whereas 256 bits in SHA-256 (Muller, 2004) (Buchanan, 2017).

The attacks on the compression function proved that MD2 should not be considered a secure hash function. Furthermore, the attacks proved that MD2 is not a one-way function since the attack succeeded against the full hash (Muller, 2004).

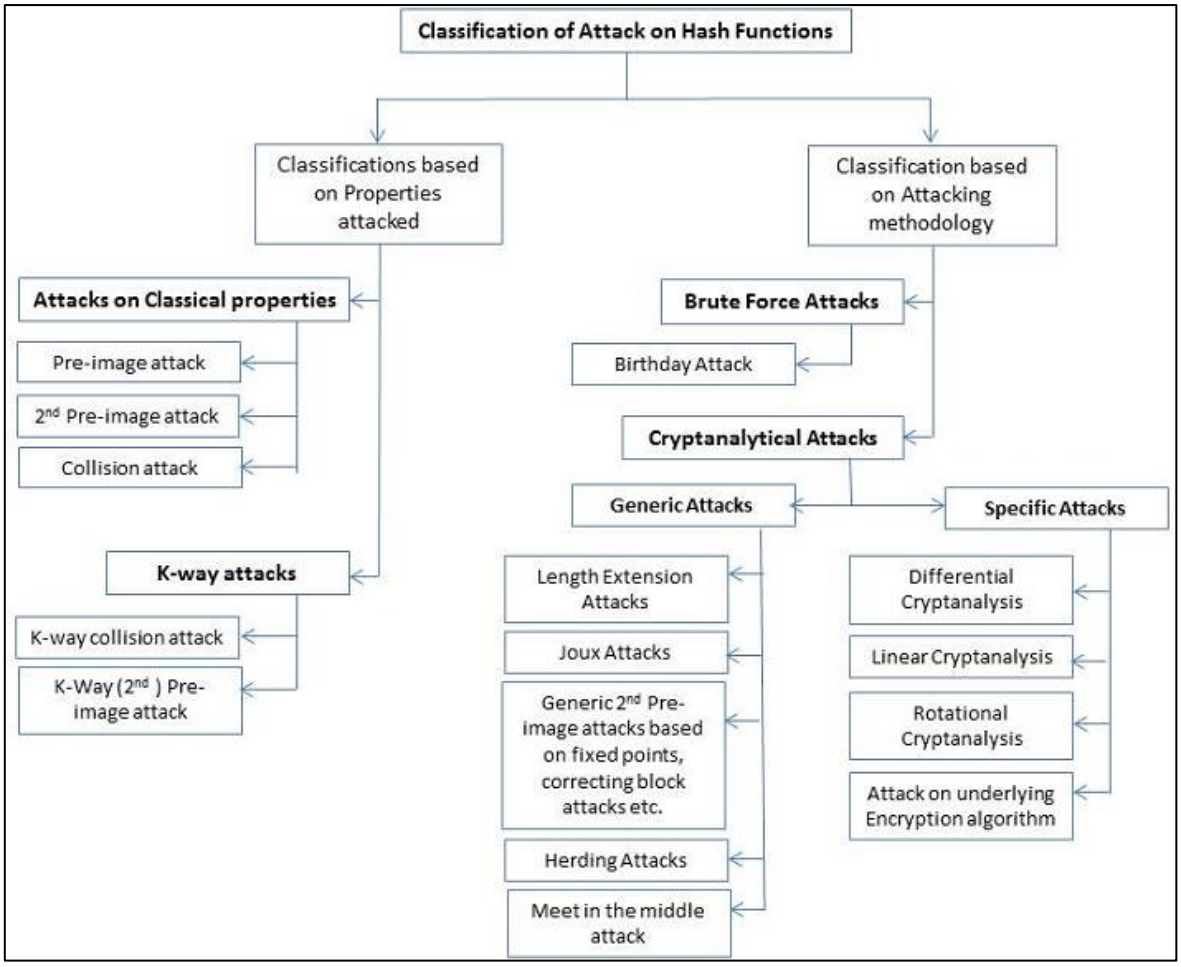

Figure 2: Hash Functions Attacks

(Sobti and Ganesan, 2012)

Conclusion:

In summary, from the previous research, it can be concluded that the hash function, although it is widely used, is cryptographically broken and suffers from considerable vulnerabilities specially as computing processing power improves.

With respect to the MD2 function, it is concluded that by omitting the checksum, it is straightforward to apply the previous attacks directly to MD2, and it could leak information about the keys with collision attacks. Moreover, although MD2 remains used in public infrastructures, it is rarely used as it takes a long time to compute and is no longer considered secure.

References:

1. Adel, N. (2017) *MD2 (Hashes and message digests), Youtube*. Available at: https://www.youtube.com/watch?v=BTf2zsvjvS0&t=907s (Accessed: May 20, 2022).
2. Bakhtiari, S., Safavi-Naini, R. and Pieprzyk, J. (1995) "Cryptographic Hash Functions: A Survey."
3. Broemeling, L.D. (2011) "An Account of Early Statistical Inference in Arab Cryptology," *The American Statistician*, 65(4), pp. 255–257. doi:10.1198/tas.2011.10191.
4. Buchanan, W.J. (2017) "Cryptography," in *Cryptography*.
5. *Cryptography: Hash Functions* (2016). Available at: https://www.youtube.com/watch?v=KqqOXndnvic&t=1185s (Accessed: May 20, 2022).
6. Kaliski, B. (1992) "The MD2 Message-Digest Algorithm." RFC Editor (Request for Comments). doi:10.17487/RFC1319.
7. Katz, J. and Lindell, Y. (2019) *Introduction to Modern Cryptography*. Milton, UNITED KINGDOM: CRC Press LLC. Available at: http://ebookcentral.proquest.com/lib/londonww/detail.action?docID=6425020.
8. Kessler, G.C. (1998) "An Overview of Cryptography." Available at: http://www.garykessler.net/library/crypto.htmlhttp://www.garykessler.net/library/crypto.html (Accessed: May 20, 2022).
9. Mironov, I. (2005) "Hash functions: Theory, attacks, and applications."
10. Muller, F. (2004) "The MD2 Hash Function Is Not One-Way." Available at: https://learn.london.ac.uk/pluginfile.php/254832/mod_resource/content/1/CW1Paper%203%20A%20research%20paper%20.pdf.
11. Preneel, B. (1999) "The State of Cryptographic Hash Functions," in Damgård, I.B. (ed.) *Lectures on Data Security: Modern Cryptology in Theory and Practice*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 158–182. doi:10.1007/3-540-48969-X_8.
12. Sobti, R. and Ganesan, G. (2012a) "Cryptographic Hash Functions: A Review Design of New hash Function using MCC View project," *Article in International Journal of Computer Science Issues* [Preprint]. Available at: https://www.researchgate.net/publication/267422045 (Accessed: May 21, 2022).
13. Sobti, R. and Ganesan, G. (2012b) "Cryptographic Hash Functions: A Review Design of New hash Function using MCC View project," *Article in International Journal of Computer Science Issues* [Preprint]. Available at: https://www.researchgate.net/publication/267422045 (Accessed: May 22, 2022).
14. Stevens, M. (2012) "Attacks on Hash Functions and Applications," *Leiden University Scholarly Publications* [Preprint].
15. Stevens, M. *et al.* (2017) "The first collision for full SHA-1," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10401 LNCS, pp. 570–596. doi:10.1007/978-3-319-63688-7_19/FIGURES/6.
16. UoL (no date) "A More Formal Look at Cryptographic Hash Functions."
17. Wang, X. *et al.* (2004) *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*. Available at: https://eprint.iacr.org/2004/199 (Accessed: May 21, 2022).

Word Count:

Total number of words: **1823** words