Computer Science Extended Essay:

Investigating security measures of SHA-5 and MD-5 algorithms for hashing.

Research question:

To what extent is SHA-5 hashing algorithm more efficient and secured as compared to MD-5

Essay word count: 3998

Table of Contents

1. Introduction

This essay focuses on algorithms that are used for hashing. Hashing, which is an essential process to validate data and keep it confidential, safe and unique every time.

This essay will specifically get into the details of the *Secure Hash algorithm 5 (SHA-5)* and *Message-Digest algorithm 5 (MD-5)*. These are two distinct types of hashing algorithms commonly used for validating data and keeping data confidential, secure and unique every time. Given that both algorithms use a single string inserted into them, we will investigate the more secure hash values for both these algorithms.

I chose these two algorithms for hashing since I was very keen to study how multinational companies transfer confidential data and check the integrity of the file being transferred. When the concept of bit coin was booming in 2017 I heard of a new concept known as crypto currency. Not only bit coin but there were many different crypto currency in play. I wanted to investigate further on this topic. In order to understand how hashing works, I took the help of my school teachers on the subject. As I got into the details of the topic, I understood the need to have so many different algorithms for hashing and how different and secure they were, leading me to chose MD-5 and SHA-5 algorithms and exploring the question to what extent is SHA-5 hashing algorithm more efficient and secured as compared to MD-5.

2. Theory

2.1 Message-Digest algorithm 5 (MD-5)

There are a few properties that make a hashing algorithm a good hashing algorithm. These properties are:

- The hash value cannot be reversed, it is one way
- The output of the hash value doesn't reveal anything about the real data

- There should be no collisions in the hash values which means the hash value should be unique. Hash value is the encrypted VARCHAR code generated by the algorithm.
- For every change in the string the hash value should be very different

The term Cryptography is commonly used to identify methods to hide (encrypt) and authenticate information[1]. MD-5 or Message-Digest algorithm 5 is a common term used in cryptography. MD-5 algorithm gives us a hash value of 128-bits.

MD-5 is used by many computer organizations for security and also to check the file integrity. The hash value given by this algorithm is typical 32 digit/ character with hexadecimal value.

All the hashing algorithms work on a similar logic. In MD-5 the hash function is a mathematical function that converts an input value into another compressed output value which will be referred to as hash value throughout the essay. The output value of the MD-5 algorithms is always 128-bits. The hash value will always be of fixed length depending on different algorithms.

An example of how hashing algorithm works can be seen in ***figure 2.1.1*** below.

[1] Spinellis, Diomidis. Cryptology. 7 June 2003. 22 October 2019 <https://www2.dmst.aueb.gr/dds/secimp/crypto/def.htm>.

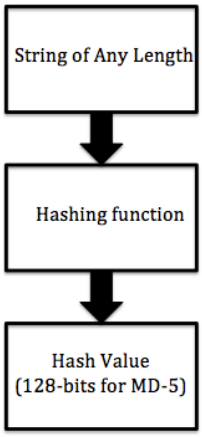


figure 2.1.1. Example of how hashing algorithm works.

The hash value of the message will be compressed hence it will have a size significantly lesser than the size of the string. This is an advantage as the hash value can be transmitted very easily without any challenge. Message-digest 5 was developed by Ron Rivest and is a very fast algorithm to produce the hash value of 128-bit.

How does Message digest work: [2]

1. A string of known length is taken (For test data we take a string of 1000 bits)
2. Padding the message. Padding means filling the unused space
   2.1. Padding is done such that the total length of the string is 64 bits less than the exact multiple of 512
   2.2. Example of padding

        1000 bits: size of the string

        512 × 2= 1024 (cannot be used because padding need to be 64 bits less than exact multiple of 512.)

        512 × 3= 1536 (can be used.)

        1536 − 64= 1472 (we need to make our string 1472 bits so that we can add a padding of 472 bits to 1000)

[2] Kanthety, Sundeep. YouTube. 18 April 2018. 17 October 2019 <https://www.youtube.com/watch?v=53O9J2J5i14&t=400s. >.

1000 + 472 = 1472 (now the string is padded and is 64 bits less than exact multiple of 512)

3. Appending the length of the string that we had before padding.
   3.1. We will use an operation of modulo 64.
   3.2. 1000 bits (original length). Length mod 2^{64}.
4. Divide it in 512 blocks.
5. Initialize 4-chaining variables.
   5.1. These are of 32bits. A,B,C and D (we already have the pre defined hexadecimal value of these variables).
6. Process blocks
   6.1. Copy the four chaining variables into their corresponding variables. {A=a ,B=b ,C=c ,D=d}
   6.2. Divide 512-bit block into 16(32 bit blocks)
   6.3. Run four rounds

   In the last step 16 sub blocks are added to a constant (T) and one round is completed.[3]

   The sum up of this process can be seen in ***figure 2.1.2*** below.

[3] TechDifferences. 19 April 2019. 22 October 2019 <https://techdifferences.com/difference-between-md5-and-sha1.html>.

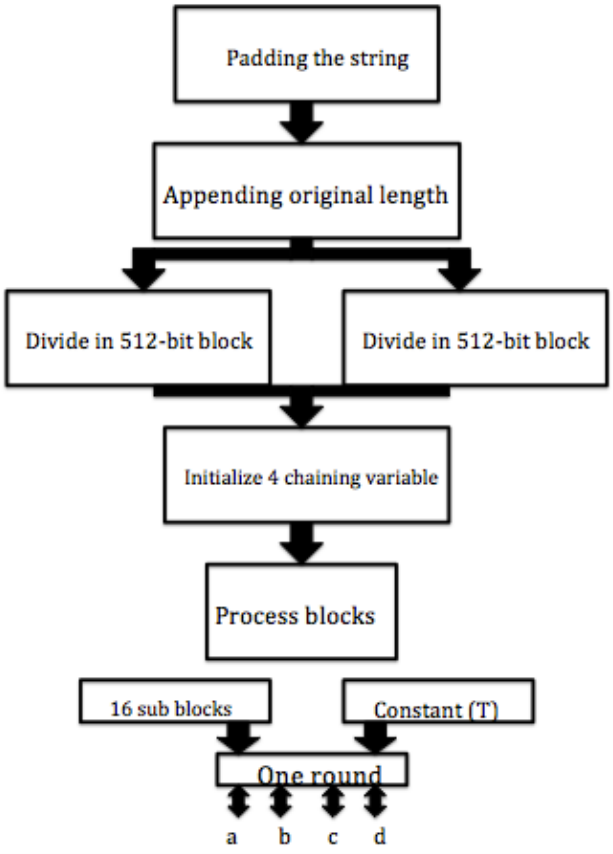
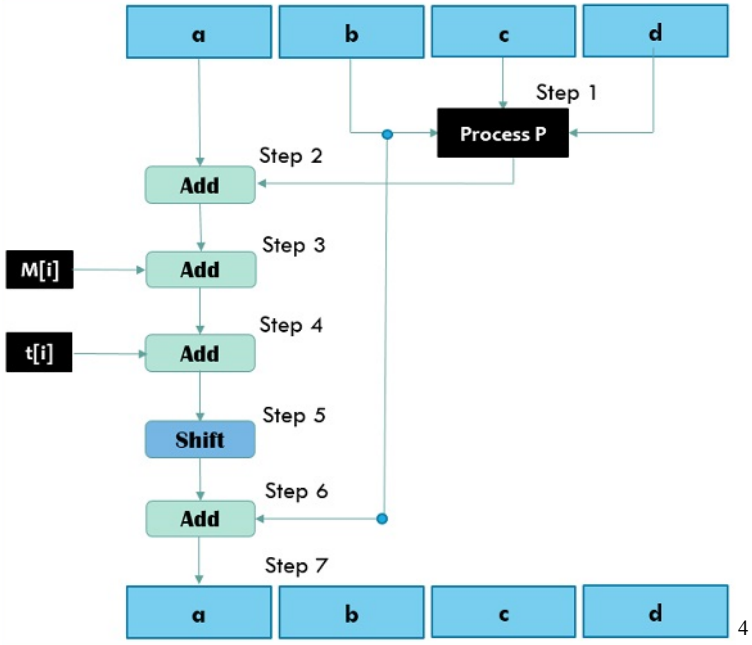


figure 2.1.2 char of how the MD-5 algorithm work.



[4]

[4] TechDifferences. 19 April 2019. 22 October 2019 <https://techdifferences.com/difference-between-md5-and-sha1.html>.

figure 2.1.3 process diagram of MD-5 algorithm

From the figure 2.1.3 we can see that that step 1 is the process block applied to variables b, c and d. In step 2 we add the process done to variables(b, c, d) to the line of commands. In step 3 the string(M[i]) is added. Step 4 the constant(t[i]) is added. In step 5 we have something known as shift, this is a circular left shift by "x" digits, the value of "x" is not constant. Step 6 variable b is added. Last step is where the hash value is generated. From this we can generate a formula:

a = b + ((a + Process P (b, c, d)+ M[i] + T[k])<<<circular left shift)[5]

a, b, c, d = Chaining variables

Process P = A non-linear operation

M[i] = M[q x 16 + i], which is the i^{th} 32-bit word in the q^{th} 512-bit block of the message

t[k] = A constant

<<<s = Circular-left shift by s bits[6]

Java code for MD-5 can be seen in the ***figure 2.1.4***

[5] Gupta, Piyush. "Comparitive Analysis of SHA and MD5 Algorithm." International Journal of Computer Science and Information Technologies 5.3 (2014): 4429-4495.

[6] TechDifferences. 19 April 2019. 22 October 2019 <https://techdifferences.com/difference-between-md5-and-sha1.html>.

```java
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

// Java program to calculate MD5 hash value
public class MD5 {
    public static String getMd5(String input)
    {
        try {

            // Static getInstance method is called with hashing MD5
            MessageDigest md = MessageDigest.getInstance("MD5");

            // digest() method is called to calculate message digest
            //  of an input digest() return array of byte
            byte[] messageDigest = md.digest(input.getBytes());

            // Convert byte array into signum representation
            BigInteger no = new BigInteger(1, messageDigest);

            // Convert message digest into hex value
            String hashtext = no.toString(16);
            while (hashtext.length() < 32) {
                hashtext = "0" + hashtext;
            }
            return hashtext;
        }

        // For specifying wrong message digest algorithms
        catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }

    // Driver code
    public static void main(String args[]) throws NoSuchAlgorithmException
    {
        String s = "Computer Science";
        System.out.println("Your HashCode Generated by MD5 is: " + getMd5(s));
    }
}
```
[7]

Figure 2.1.4 java code for MD-5 algorithm

2.2 Secure hash algorithm (SHA-512)

SHA-512 was created by National institute of Standards and Technology (NIST). SHA-5 or secure hash algorithm 5 is a modified version of MD-5. There are many types of SHA algorithms, SHA-1, SHA-256 and SHA-512. The output of SHA-512 algorithm is 160 bits in length. SHA-512 is a part of the second generation of secure hash algorithms, also known as SHA-2. SHA-2 includes SHA-224, 256, 384 and 512. The main reason for creating SHA-2 was to validate and sign digital security certificates and documents. Since SHA-512 is a modified version of MD-5, the initial steps of this algorithm are the same. The steps for SHA-512 algorithm are:

1. Padding (64 bit less than exact multiple of 512)[8]

[7] GeeksforGeeks. 20 Augest 2018. 10 October 2019 <https://www.geeksforgeeks.org/md5-hash-in-java/>.

2. Append the original length
3. Divide the input into 512-bit blocks
4. Five chaining variables (A, B, C, D, E)[9]
5. Processing blocks

    5.1. Copying the chaining variables

    5.2. Dividing the 512-bit block into 16 sub blocks (each block od 32-bit)

    5.3. apply four rounds.

The only difference in the process of MD-5 and SHA-5 is that SHA-5 uses 5 chaining variables (A, B, C, D, E) and MD-5 uses 4 chaining variable(A, B, C, D). Just this one difference between the two algorithms will have a big difference in the formula that can be generated from each of these algorithms.

A flow diagram of the SHA-512 algorithm can be seen in the ***figure 2.2.1 below.***

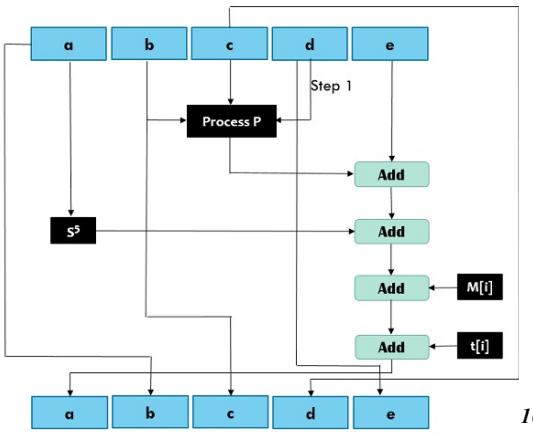


figure 2.2.1 process diagram of SHA-512 algorithm

The formula that we can generate from this diagram is:

abcde = (e + Process + s^5(a) +W [t] +K [t]), a, s^{30} (b),c,d

a, b, c, d, e: Five chaining variables

---

[8] Padding explanation on page 6

[9] Different from MD-5

[10] TechDifferences. 19 April 2019. 22 October 2019 <https://techdifferences.com/difference-between-md5-and-sha1.html>.

Process: the logical operation performed by the algorithm

S^t :random circular left shift by s digits

W[t] = A 32-bit derived from the current 32-bit sub-block

K[t] = On of the five additive constants. [11]

All the algorithms have one major feature in them and that is the hash function should be collision resistant. To make all the collision resistant, hash functions are made using a standard reference known as the Merkle-Damgard construction.

Java code for MD-5 can be seen in the ***figure 2.2.2***

```
public class SHA512 {
    public static String encryptThisString(String input)
    {
        try {
            // getInstance() method is called with algorithm SHA-512
            MessageDigest md = MessageDigest.getInstance("SHA-512");  // digest() method is called
            byte[] messageDigest = md.digest(input.getBytes());

            // Convert byte array into signum representation
            BigInteger no = new BigInteger(1, messageDigest);

            // Convert message digest into hex value
            String hashtext = no.toString(16);

            // Add preceding 0s to make it 32 bit
            while (hashtext.length() < 32) {
                hashtext = "0" + hashtext;
            }

            // return the HashText
            return hashtext;
        }

        // For specifying wrong message digest algorithms
        catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }

    // Driver code
    public static void main(String args[]) throws NoSuchAlgorithmException
    {

        System.out.println("HashCode Generated by SHA-512 for: ");

        String s1 = "Computer science";
        System.out.println("\n" + s1 + " : " + encryptThisString(s1));

        String s2 = "Computer science1";
        System.out.println("\n" + s2 + " : " + encryptThisString(s2));
    }
}
```
[12]

Figure 2.2.2 java code for SHA-512

[11] TechDifferences. 19 April 2019. 22 October 2019 <https://techdifferences.com/difference-between-md5-and-sha1.html>.

[12] GeeksforGeeks. 27 September 2018. 10 October 2019 <https://www.geeksforgeeks.org/sha-512-hash-in-java/>.

2.3 Merkle-Damgard Construction

All the hash functions need to be collision resistant. This property is very important for the hash function else it can result in a very big breach in the security of the file. Hence there exists a loop hole in the security system in a situation where the hash function creates a collision. If the hash function has a collision then this can be easily cracked. Merkle-Damgard is named after Ivan Damgard and Ralph Merkle, both of these individuals did an independent research which was fairly similar and revolving around creating a structure to construct a hash function which is collision resistance. Most of the well known hashing algorithms, including MD-5 and SHA-512 have been influenced by the Merkle-Damgard construction. A hash function is defined as H: $\{0, 1\}^* \rightarrow \{0, 1\}^t$. A message x can be represented as H(x) = y. Both Damgard and Merkle stated that if there is a fixed-length input collision resistant compress function f: $\{0, 1\}^b * \{0, 1\}^t \rightarrow \{0, 1\}^t$ if this is satisfied then a collision resistant H: $\{0, 1\}^* \rightarrow \{0, 1\}^t$ can be created. Now lets look at the Damgard's approach. Damgard divided his approach in two different cases.

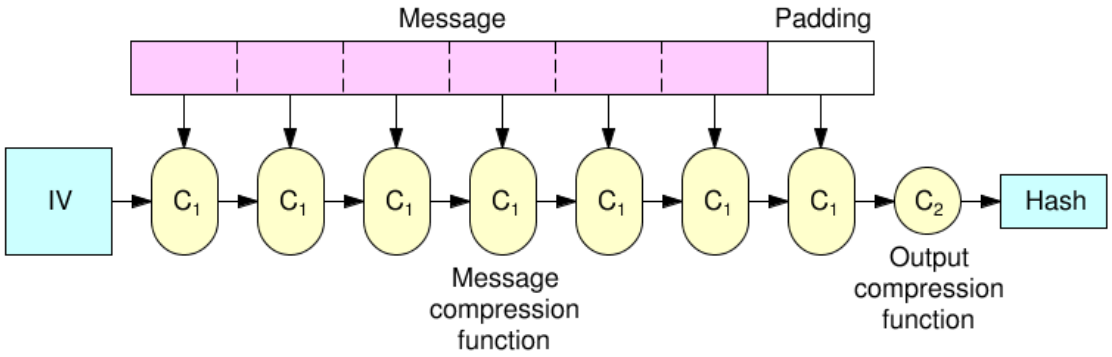
1. Case 1 [b-t<1] in this case we will be taking a message and dividing the message in different blocks. These blocks will be of the size b-t-1 and there will be a last block left with 0 bit of message. We can consider K as the number of 0 bits that we need. Then we can write the blocks in a sequence as $M_1, M_2, M_3 \ldots M_{L-1}$. The M_L block is an extra block that has the binary encoded representation of K (Number of 0 bits block). The intermediate chaining value are defined by: $H_1 = f(0^{t+1} || M_1)$

   $H_i = f(H_{i-1} || 1 || M_i)$ for i= 2,3…,t.
2. Case 2 [b – t = 1] in this case the message is hashed in the blocks itself and every block is of 1 bit. The intermediate chaining value is defined by:

   $H_i = f(H_{i-1} || M_i)$ for i= 2,3…,t.[13]

[13] Justin Holmgren, Justin. "PDF." Jan. 2018.

Figure 2.3.1 below represents the working of Merkle-Damgard structure



[14]

Figure 2.3.1

C: collision resistant hash function (compression function)

IV: fixed value

Big Messages are divided in blocks

What we can say by this diagram is that first we will apply the small compression function to the first message block along with the IV from. The out come of this will be a chaining variable. This process takes place with all the compression function present applied on the message blocks to give another chain variable. In the last step we will first append the padding block and then compress the last chaining variable and the output of this will be the hash value. So this was the Merkle-Damgard construction which has helped a lot of the hash function to be collision free.[15]

[14] Kaminsky, Alan. 2018. 10 October 2019 <https://www.cs.rit.edu/~ark/462/module11/notes.shtml>.

[15] Gauravara, Praveen. Cryptography Hash Function: Cryptanalysis, Design and Application. 2007. Queenslandland University of tecnolog,. Phd dissereation.

3. HYPOTHESIS AND APPLIED THEORY

We have explained both of the algorithms and described their working thoroughly. The working of each algorithm and the features of the algorithms have been explained. It is crucial to take into consideration which of these algorithms are more efficient and secure. We have talked about the security of both the function at the start of the essay, but we have not explored it while the two algorithms were being explored. We did talk about the Merkle-Damgard, which plays a massive role in the in terms of security of a hashing algorithm. A test will be carried out on both the algorithms to check the difference in each outcome and which is more secure. The variation of characters and length of the hash value of the hashing algorithm will be tested too. In the test, both the algorithms will have to go through a brute force attack which will finally determine which out of the two algorithms is safer.

As it was mentioned before that SHA-512 is a modified version of MD-5, SHA-512 might be a more secure algorithm. We also saw that SHA-512 has five chaining variables whereas MD-5 has four chaining variable, in terms of security SHA-512 might be ahead, but MD-5 could be faster as it has to perform less operation while hashing the function.
I will be using an online website[16] that will provide me with the algorithm for my test. I will be taking a fixed string and inserting the string in both the algorithms to analyze the outcome.

I hypothesize that SHA-512 will give a more secure hash value but will be slow; on the other hand, the MD-5 algorithm will give a less secure hash value but will be faster compared to SHA-512. There will be many criteria's that will be tested and analyze as the research is regarding security.

[16] FreeFormatter. 21 October 2019 <https://www.freeformatter.com/message-digest.html>.

4. Methodology

4.1 independent variable

The independent variable in this test will be the algorithm as I will be changing the algorithm. I will be using 2 algorithm MD-5 and SHA-512

I will also be making small changes to the string to see what will be the change in the hash value for small changes in the string. My plan is to increase a character in the string and get the output. I will be making 10 string, and every string will be having only one extra character. For example:

String number	String value
1	Computer science
2	Computer sciences
3	Computer sciences1

Table 4.1.1 example of strings inserted in the algorithms

4.2 dependent variable

The dependent variable tested for this test will be the hash value/ output from the input string put into both the algorithms. The output will be different for every string the I input therefore I will analyze all the outputs that I will receive after the test. I will also be checking the time stamp for both the algorithms to check how much time each takes to produce a hash value.

4.3 Controlled variable

Variable	Description	Specifications (if applicable)
Computer and operating system used	I will be running a java code on my MacBook pro 2012	I will be using BlueJ software to write my java code
Same algorithm	MD-5 and SHA-512 algorithms will be used only	
Online checking tool	I will be using an online tool to check if my java program is giving the right hash value	Website name: (FreeFormatter)
String character	I will be keeping the string	

	same for both the algorithm for each data collection	
Data type	I will only be using characters and integers for my string	

Table 4.3.1 Controlled variables

4.4 Process

1. verify the java code and find any flaws or bugs in it.
2. Chose a string to be inserted in the algorithms.
3. Input the String in both the algorithms one at a time.
4. Run the program 5 times to check if the output is the same every time with one of the string .
5. Record the time stamp and the hash value for each test.
6. Add a new character to the string.
7. Repeat step 3, 4 and 5 for 10 such string.

5. Data processing

5.1 Data collection and processing

Below is the MD-5 execution table showing the string and the hash value.

String value	Hash Value (MD-5)
Computer science	284fcfb183d1919532b3c7a6dba33873
Computer science1	d41ec200b38d7c03ad5353e084cbca72
Computer science1@	fbd022a4f69a28feaca36cf9f21d62c8
Computer science1@is	5a9de876accfc13e1143fe72c44daa85
Computer science1@is@	d223dd27c2de63e4359f24acb5293434
Computer science1@is@T	0aa401660eea675633863fe739a4d768

Computer science1@is@Th	678490d10d46c63af1477ad3992f0ac0
Computer science1@is@The	32125b2cc1c031c936b408e76cd0528f
Computer science1@is@The_	3c6532c7d5105377e842372bf544fe70
Computer science1@is@The_best	eea90f64eeb31246bec8022104a95261

Table 5.1.1 MD-5 execution

Below is the SHA-512 execution table showing the string and hash value.

String value	Hash Value (SHA-512)
Computer science	eb8a19c2ce6eb5a0c378831fd69867a421a16ce6ff298f2d865245c733ca5f5e ee98f7490c04e2bf0bbb6d96862ea7a0bcb3392b9af774ec86e2e9e1da71cd65
Computer science1	d927162879e8cceb061a1dbb2ac4219fb648727b8d08da1ad892f4232123e6d bbd4a46bf2e373792bfdf58b9bcaad9475fe1b3001a6176ba9193778e1af330c 6
Computer science1@	1a0e37b1c73a220bd51782b051ed8f36d4edd6aac7f4afa937b98b276a8f79b6 fa5f00e772ff69e9cb83a99ee3101c00662eb281f9b0ccb46a940035c252393f
Computer science1@is	49bf0766ac7aba33a5826440b367b93653e8e5cccce1cb48b6574785441c3a8 5d79d8a32e7b3e2b7d76c5153cababa7e327ae91c2cfb8936433f23fc6625c86 e
Computer science1@is @	9795ccf8c657c3cdd79fd99cc5d9c6bc255a6dcfc4d4e0a0dd6e6d0d7e761960 1f98bae79206baec55c2887e4d88d93fea3626efa5502edb181a60b9ab642ac5
Computer science1@is @T	9b4a6a22c3cf90ec4b4d7d2700d77d00c0395f27ea59d805ccb0fc51dccd0c38 aab5881e8a98fe1e5cb6c2a6f20c100ec643d6504ed5027a59017bb12aaef2d9
Computer science1@is @Th	921a617e31063b3a24be1b74fbeb614b40821a34fb583bdde81e75b7c1ac101 644106856fc6c04ca2d99ec2fb12f523c2eeff6c7293a264feaa661fa7e718abd
Computer science1@is @The	71c8f8dddcc0bda496243a4326848f827d456b4d59db6ee64e5cc60a19369ad 41a5e3f98ad3183ee888e125077151de95faa9a2cd1cc82cdefecb9b2469f1a4 5
Computer science1@is	e6fa5162219d738337d610b74ba9de70fc449bc902f806d6a270c6265afea4a0 62676748753b9653781fe705d55dda2841508ca2ec2b08122d8616cbeb492a

@The_	c0
Computer science1@is @The_best	88545a259c8aa0fbeb4b844bca537d64f025e73559d0293c17f44a8791239d0beeb8673daa1a0dec064990fb64beb6b5012ac2a5e3d484bd915a8d8d86294699

Table 5.1.2 SHA-512 execution

Table 5.1.1 and 5.1.2 show how both algorithms have created different hash values. My main aim was to check how secure each of the algorithms. From the data the I have collected I can clearly say that SHA-512 is a very secure algorithm. SHA-512 has 128 characters hash value, where as MD-5 has only 32 character hash value. As we had seen before that SHA-512 is a modified version of MD-5, it is proven that SHA-5 can produce a more secure hash value. When we talk about the speed of each algorithm to produce its hash value, MD-5 is the winner. Speed is a very big factor the is taken in consideration in many big companies, this is the reason some companies still use MD-5 instead of SHA-5.

6. Comparison(MD-5 and SHA-512)

Although MD-5 is an old algorithm it is still in implementation in many companies. We will be exploring the reason why people use MD-5.

Below ***table 6.1*** shows the differences in the two algorithm

Comparison	MD-5	SHA-512
Security	Not very secure, less secure than SHA-512	Very secure, modified version of MD-5
Speed	Very fast, 60 iteration	Slower than MD-5, 80 iteration
Successful attacks	Collisions and attacks	No attacks detected

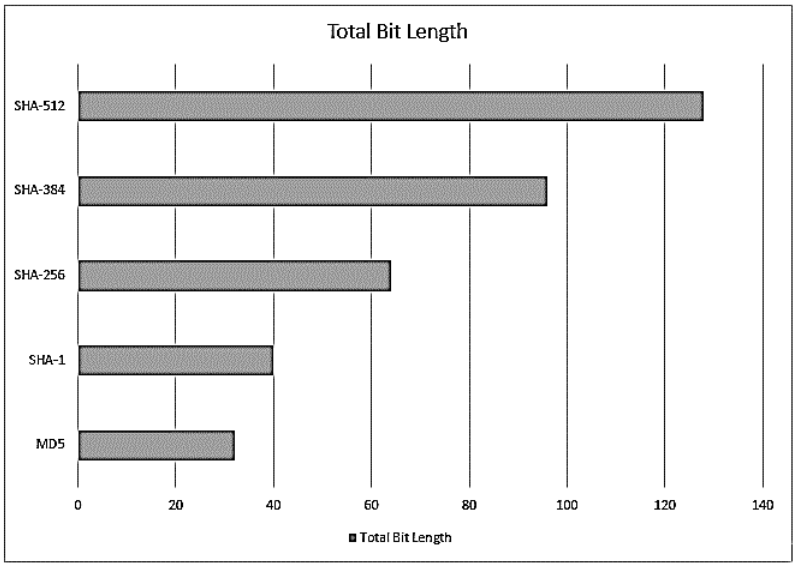
	detected	
How to find original message	2^{128} bit operation to crack and find original message from the hash value	2^{160} bit operation to crack and find original message from the hash value
Find a collision	2^{64} bit operation to crack and find a collision	2^{80} bit operation to crack and find a collision
Hash value length	128 bits	160 bits
Hash value character length	32 character	128 character
Chaining variables	4 chaining variables	5 chaining variables

Table 6.1 differences between both algorithm

There are many similarities in the two algorithms too

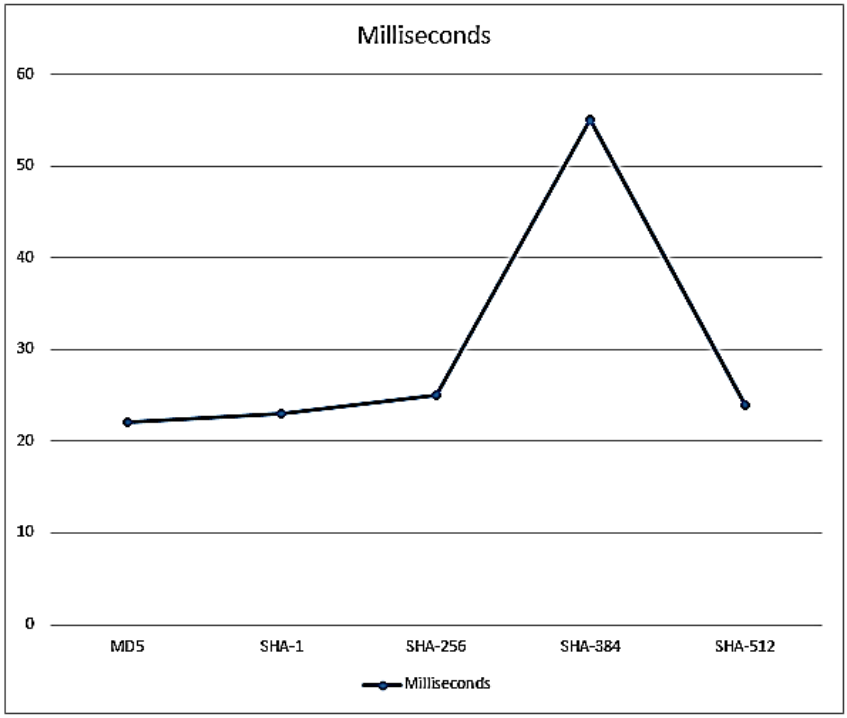
1. In both the algorithms padding is necessary without padding no hashing algorithm can work.
2. Both of the algorithms produce messages in bits
3. Both the algorithms utilize the same resources
4. Procedure of both the algorithms is the same
5. Both are finger prints, different each and every time.

The graphs below show the difference in speed of each algorithm and the message bit length



[17]

figure 6.1 Shows the difference in the message bit length of different hash functions



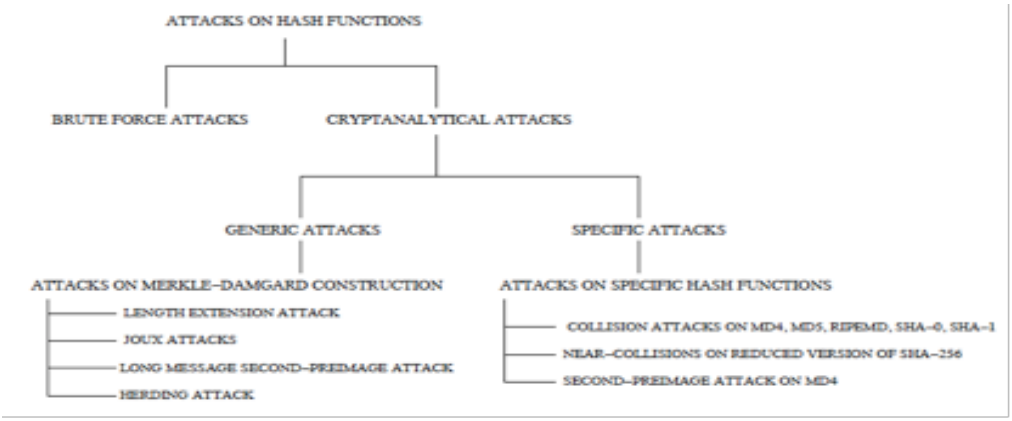
[18]

Figure 6.2 Speed of different algorithms to create a hash value

[17] Gupta, Piyush. "Comparitive Analysis of SHA and MD5 Algorithm." International Journal of Computer Science and Information Technologies 5.3 (2014): 4429-4495.
[18] Gupta, Piyush. "Comparitive Analysis of SHA and MD5 Algorithm." International Journal of Computer Science and Information Technologies 5.3 (2014): 4429-4495.

7. Attacks on Hash function

There are many attacks that can be performed on a hash function. A good hash function should not get affected by the attacks on. A classification of attacks on hash function can be seen in ***figure 7.1***



[19]

figure 7.1 different types of attacks that can occur on a hash function

7.1. Brute Force attack

Brute Force attack works on all the algorithms. There is no such algorithm that can surpass a brute force attack. Brute force attack is independent of the way the algorithm works, complexity and process. Brute force attack it a very lengthy process but the most effective and reliable process to crack an algorithm. Brute force is like an exhaustive search on the encryption algorithm to extract the secret key from the algorithm. In short, it is a method to try every possible outcome that can be generated from the algorithm. The factor that makes brute force attacks lengthy is a longer outcome with the mixture of character, upper case letter, symbols and numbers. This is the reason many companies insist the employees to have a password with symbols and numbers, as it can be harder to crack and can take much time to crack. There are different types of brute force attack, which are:

[19] Gauravara, Praveen. Cryptography Hash Function: Cryptanalysis, Design and Application. 2007. Queenslandland University of tecnolog,. Phd dissereation.

1. Brute-Force preimage attack: in this attack we will take a hashing algorithm which produces a x-bits message digest for y. The hacker will explore each and every possible input in the hash function until the hacker finds the value of y. For a good and ideal hash function it should take 2^x computation to crack the value of y. So a better hash function will create a hash function which gives a longer value of y which makes it a higher power of 2 to crack.
2. Brute-Force 2^{nd} preimage attack: in this attack, for a give message x and the hash function H that produces t-bits message digest, the attacker trials H with any possible input message so that x'!= x until the attacker obtains the value of H(x). Even for this attack the an ideal hash function will take around 2^t computations to crack the algorithm and find the value that was inputted by the user.

Brute force attack as discussed cannot be stopped, however we can make sure that the brute forces attack is less efficient. The longer and the more complex the key or the password it more exhaustive it becomes for the hacker to crack the data. One main reason why multi level authentication came into picture. When a hacker tries to crack the data he has to crack several layers of security

7.2. Cryptanalytical attacks

Cryptanalysis is the examination of ciphertext, figures and cryptosystems with the purpose of perceiving how they work and finding and improving methodologies for surviving or incapacitating them. For example, cryptanalysts hope to unscramble ciphertexts without data of the plaintext source, encryption key or the estimation used to scramble it; cryptanalysts in like manner target secure hashing, automated imprints and other cryptographic figurings. There are many different ways in which a person can attack any hashing algorithm or cryptography. Some of these ways are Frequency analysis, Algorithm

error, Dictionary attack, Social engineering and rubber hose attack. [20]A cryptanalytical attack (or merely an attack) on a hash function is defined as some analytical approach or method used to violate at least one of its security properties. A preimage attack (resp. second preimage attack, collision attack) violates the preimage resistance (resp. second preimage resistance, collision resistance) of a hash function. In a Frequency analysis, the attacker would look at blocks of encrypted data and try to find various patterns in it. The attacked will try to see how the letter a or e is repeated in the hash value. An algorithm attack is merely analyzing the algorithm to find any flaws in it . in a dictionary attack, the attacker will try to look into a dictionary to find words common in many passwords as most of the password are words. In a social engineering attack, the attacker will trick the person into disclosing the password or the encryption key. Rubber hose attack looks into attacks when the attacker will force or blackmail the person into disclosing the password.

Due to the fixed size of the hash values compared to the much larger size of the messages, collisions must exist in hash functions. However, for the security of the hash function, they must be computationally infeasible to find.[21]

8. Conclusion

This experiment aimed to focus on the two different algorithms used for hashing and cryptography. The flow chart of both the algorithms and the working has been explained in section 2[22] of the document. This essay intended to practically apply it to see the relationship and difference in the security and the complexity of the hash value that is being generated by the algorithms(MD-5 and SHA-512). As I hypothesized SHA-512 algorithm is more secure

[20] Subramanian, Prabhu. YouTube. 16 July 2019. 31 October 2019 <https://www.youtube.com/watch?v=8peLDbtyozw&t=66s>.

[21] Subramanian, Prabhu. YouTube. 16 July 2019. 31 October 2019 <https://www.youtube.com/watch?v=8peLDbtyozw&t=66s>.

[22] Page 4

in terms of security, SHA-512 algorithm creates a hash value of 128 character hexadecimal value, in comparison to SHA-512, MD-5 creates a hash of only 32 character hexadecimal value. SHA-512 creates a hash value of 160 bits, whereas MD-5 creates a 128-bit length hash value. Hence coming back to the security of the algorithms SHA-512 will take more time to crack as it will take 2160 combinations and permutations. MD-5 algorithm is an older algorithm which creates a hash value of 128 bit, which will take 2128 combinations to crack. As discussed before a good hashing algorithm should have no collision, there have been cases in which there have been some cases of collision that have been found in the algorithm. In this essay, I also explored the factor of time taken for the hashing algorithm to create its hash. In terms of time, no other algorithm is faster than MD-5, which makes it a factor that some of the company's are still using MD-5 algorithm.

To answer the research question of this essay, my answer would be that use of any of these algorithms will have different criteria and situations related to it. If a firm does not have very confidential data and has to use the algorithm for essential purpose, MD-5 will be enough to fulfill the needs of the firm. If a firm is looking for a fast encryption algorithm, even then, MD-5 is good enough as it is still one of the fastest algorithm.

Bibliography

Aggarwal, Surbhi. "A review of Comparative Study of MD5 and SHA Security Algorithm." International Journal of Computer Applications 104.14 (2014): 0975-8887.

Classes, Easy Engineering. YouTube. 18 November 2018. 17 October 2019 <https://www.youtube.com/watch?v=G_qtQgRmiWk>.

FreeFormatter. 21 October 2019 <https://www.freeformatter.com/message-digest.html>.

Gauravara, Praveen. "Cryptography Hash Function: Cryptanalysis, Design and Application." 2007.

Gauravara, Praveen. Cryptography Hash Function: Cryptanalysis, Design and Application. 2007. Queenslandland University of tecnolog,. Phd dissereation.

GeeksforGeeks. 20 Augest 2018. 10 October 2019 <https://www.geeksforgeeks.org/md5-hash-in-java/>.

GeeksforGeeks. 27 September 2018. 10 October 2019 <https://www.geeksforgeeks.org/sha-512-hash-in-java/>.

Gupta, Piyush. "Comparitive Analysis of SHA and MD5 Algorithm." International Journal of Computer Science and Information Technologies 5.3 (2014): 4429-4495.

Justin Holmgren, Justin. “PDF.” Jan. 2018.

Kaminsky, Alan. 2018. 10 October 2019 <https://www.cs.rit.edu/~ark/462/module11/notes.shtml>.

Kanthety, Sundeep. YouTube. 18 April 2018. 17 October 2019 <https://www.youtube.com/watch?v=53O9J2J5i14&t=400s. >.

Maetouq, Ali. "Comparission of Hash Function Algorithm Against Attacks: A Review." International journal of Advance Computer Science and Applications 9.8 (2018): 98-103.

Spinellis, Diomidis. Cryptology. 7 June 2003. 22 October 2019 <https://www2.dmst.aueb.gr/dds/secimp/crypto/def.htm>.

Subramanian, Prabhu. YouTube. 16 July 2019. 31 October 2019 <https://www.youtube.com/watch?v=8peLDbtyozw&t=66s>.

TechDifferences. 19 April 2019. 22 October 2019 <https://techdifferences.com/difference-between-md5-and-sha1.html>.