Theoretical Framework of MD5 Algorithm for Password Encryption

Ashley Shaji
PG Scholar
Amal Jyothi College Of Engineering
Kanjirapally,Kottayam
ashleyshaji@mca..ajce.in

Jinson Devis
Department of Computer Applications,
Amal Jyothi College Of Engineering
Kanjirapally,Kottayam
jinsondevis@amaljyothi.ac.in

***Abstract*—This document is an elaboration on how MD5 algorithm is used to encrypt data and files. The MD5 message-digest algorithm takes an arbitrary-length message as input and generates a 128-bit "fingerprint" or "message digest" as output. The MD5 algorithm is designed for digital signature applications in which a large file must be securely "compressed" before being encrypted with a private (secret) key using a public-key cryptosystem like RSA. Two conditions should be met by MD5It's difficult to make two inputs that yield the same hash function and It is difficult to produce a message with the same hash value as the previous one.MD5 was originally designed to store a one-way hash of a password, and some file servers often have a pre-computed MD5 checksum of a file, which the user can compare to the checksum of the downloaded file.**

***Keywords*— hash function, MD5 algorithm ,digital signatures, collision**

I. INTRODUCTION

A secure password hash is an encrypted sequence of characters generated by applying particular algorithms and manipulations to a user-supplied password, which is typically weak and easy to guess. There are a variety of hashing techniques that can be quite useful for password security. You must regenerate the password hash each time a user logs into the programme and compare it to the hash saved in the database. If the user forgets his or her password, you must issue him a temporary password and ask him to replace it with his or her new password. Isn't it usual these days? The MD5 Message-Digest Algorithm generates a 128-bit (16-byte) hash result and is a frequently used cryptographic hash function. It's very simple; the primary idea is to translate variable-length data sets to fixed-length data sets.

II. LITERATURE REVIEW

A. Digital Signatures

The MD5 hash function was created with the intention of being used as a secure cryptographic hash algorithm for digital signature authentication.The algorithm takes an arbitrary-length message as input and generates a 128-bit 'fingerprint' or'message digest' of the input as output. It is hypothesised that producing two messages with the same message digest, or any message with a pre-specified target message digest, is computationally impossible. The MD5 algorithm is designed for digital signature applications in which a large file must be securely 'compressed' before being encrypted with a private (secret) key using a public-key cryptosystem like RSA.

B. Hash Functon

A hash function is a one-way encryption function that generates a fixed-size hash output from a variable-size input plaintext m. Deciphering the hash is computationally difficult, and any attempt to crack it is essentially impossible. Pre-image and collision attacks should be resistant to a "secure" hash function. There will be certain inputs that return the identical hash result due to the pigeonhole principle and the birthday paradox. There are a total of 2128 potential output hash values because the output length is fixed at 128 bits. As large as this figure appears to be, it is not limitless, resulting in collisions.[1]

C. MD5 ALGORITHM

Ron Rivest created MD5 (Message Digest Algorithm 5) in 1991. MD5 converts a variable-length message into a 128-bit fixed-length output. MD5 is a widely used hashing function. It works with 512-bit blocks and processes each block via four rounds, each of which processes 16 sub-blocks (each 32-bits). The 512-bit message is divided into 16 sub-blocks before processing. If a message block is not up to 512-bits, it is padded as shown in Fig. 1.[1]

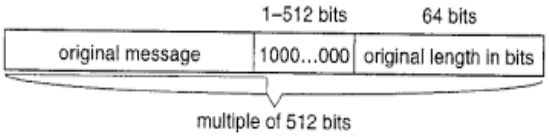

Fig 1: Length of message after padding (in bits)

D. Collision

According to some studies, the MD5 algorithm can be deciphered using a collision attack, and the security of its implementation has been questioned. This essay examines the use of the MD5 algorithm in password encryption and its security, as well as the physical steps used to ensure the application's security. The hash algorithm compresses a piece of information with a random length into a fixed length value, which is referred to as information abstraction. The so-called "collision" is a knowledge abstract created from two pieces of different plain code. The following are the conditions for a secure Hash function: first, two pieces of separate plain code produce the same information abstract, which can not be calculated and is referred to as collision; second, a specific information abstract cannot be calculated by other plain

10.5281/zenodo.5105543
ISBN:978-93-5426-386-6@2021 *MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam*

information generating the same abstract, implying that the initial state cannot be deduced from the results.[2]

III. ENCRYPTION PRINCIPLE

The MD5 algorithm divides the supplied data into 512-bit groups, each of which is then broken into sixteen 32-bit sub-groups. After a series of processing, the algorithm's output is made up of four 32-bit groups, which are then cascaded to create a 128-bit hashed value. The information is first filled in the MD5 algorithm, hence the result of the bit length deriving remaining from 512 is 448The filling method is to fill a 1 and many 0 after the information, and the filling shall not be halted until the aforementioned conditions are met; following this, the length of the information before filling (given in 64 bit binary) shall be enclosed. In MD5, four 32-bit integer parameters are termed linked variable integer parameters, and they are A=0x01234567, B=0x89abcdef, C=0xfedcba98, and D=0x76543210. Following the proper setup of these four linked variables, four rounds of circulated calculations are initiated, with the frequency of circulation equal to the quantity of 512 bit information grouping in the data. A to a, B to b, C to c, and D to d are the four connected variables that are replicated to another four variables. The main circulation includes four rounds, each with 16 operations. At each operation, three of a, b, c, and d are conducted by nonlinear function operation for one time, then all acquired results are added by the fourth variable, and all acquired results are loop-shifted by a variable number toward the right, and added by one of a, b, c, or d, and lastly the result shall substitute one of a, b, c, or d. [3]

After that, a, b, c, and d are added to A, B, C, and D, respectively.Then, to continue with the operation method, the following grouping data are employed, and eventually, the cascading of A, B, C, D is outputted, with A being low bit and D being high bit, and DCBA forming the 128 bit output result.[3]

IV. WORKING

Step 1: Append Padding Bits

- Padding refers to the addition of extra information to the initial message. As a result, the original message in MD5 is padded to a length in bits that is congruent to 448 modulo 512. Padding is used to reduce the total number of bits to 64, which is a multiple of 512 bits.
- And if the length of the original message is already congruent to 448 modulo 512, padding is completed. The only bit in padding bits is 1, and the rest of the bits are 0.

Step 2: Append Length

Following the padding, 64 bits are inserted at the end to record the original input length. Modulo 2^64. The resulting message has a length multiple of 512 bits at this stage.

Step 3: Initialize MD buffer.

The values for the message digest are computed using a four-word buffer (A, B, C, D). A, B, C, and D are 32-bit registers that are initialised as follows:

Word A	01	23	45	67
Word B	89	Ab	Cd	Ef
Word C	Fe	Dc	Ba	98
Word D	76	54	32	10

Step 4: Processing message in 16-word block

MD5 makes use of auxiliary functions, which take three 32-bit numbers as input and output 32-bit numbers. Logic operators such as OR, XOR, and NOR are used in these functions.

F(X, Y, Z)	XY v not (X)Z
G(X, Y, Z)	XZ v Y not (Z)
H(X, Y, Z)	X xor Y xor Z
I(X, Y, Z)	Y xor (X v not (Z))

Using this auxiliary buffer, the contents of four buffers are combined with the input, and 16 rounds are completed using 16 simple operations.[2]

Output:

After all rounds have been completed, the MD5 output is stored in the buffers A, B, C, and D, beginning with lower bit A and ending with higher bit D.[2]

V. IMPLEMENTATION

MD5 algorithm has been used to hash all of the user passwords and the hash value has
been stored in the storage Database. Even if any outsider gets his hands on the Database,
passwords won't be stolen as they have been hashed and stored.
When any user tries to log in to the system, the hash function checks if:
MD5(user entered password) = Hash value stored in the Database.
If true, then User successfully logs in

```
"paswd"=>md5($_POST["paswd"])
```

Output:

	loginid	email	paswd	usertype	lo
Edit Copy Delete	45	rose1998@gmail.com	52b1d1eed4467d2a2acfe261770e18dc	R	A
Edit Copy Delete	46	ash@gmail.com	6fcbbd280419eadbba775b36de1bef3a	R	A
Edit Copy Delete	49	donamathew141@gmail.com	5f32b85162a02e6295fb4ba26578f891	R	A
Edit Copy Delete	51	ashleyshaji1998@gmail.com	53ead9a0499a04770dc1679fa537fb89	O	A

MD5 algorithm based improvement to encryption method

The MD5 algorithm processing method can be altered, or intervening information added to the user password, so that the cypher text information in the MD5 algorithm cypher text database is no longer the simple MD5 message digest value.

10.5281/zenodo.5105543
ISBN:978-93-5426-386-6@2021 *MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam*

A. Transformation to MD5 processing process

1) The frequency of MD5 processing is increased; the basic idea is that a user password that requires MD5 encryption is encrypted two or three times, i.e., the 128 bit message digest value is computed once more by MD5 computing.[3]

2) The meaning of the message digest is intercepted. The basic idea is that the user password is first run through MD5 computing once to get the MD5 message digest value; then some values are chosen to run through MD5 computing again to get the final result.[3]

3) The meaning of the message digest is split. The basic idea is that the user password is first processed through MD5 computing once to derive MD5 message digest value H; then H is split into two 64-bit classes, left and right, and processed through MD5 computing again to derive similar message digest values, which are given by HL and HR, respectively; HR and HL are then combined (front and back) into a character string, which is then subjected to MD5 computing to obtain the final result.[3]

4) Encryption algorithm specified by the user. The basic idea is that the user password is encrypted using a self-defined encryption algorithm to generate the cypher text, which is then processed using MD5 computing to generate the MD5 message digest value. character shall correspond to a single bit cypher key (if the cypher key is more than 10, it shall be intercepted, if it is less than 10, it shall be supplemented by 0); the data in need of encryption is grouped by taking ten characters as one group (if it is less than 10, it shall be supplemented by blank); the data in need of encryption is grouped by taking ten characters as one group (if it is less than 10, it shall be supplemented by blank); Each bit shall be of character interchange subject to the corresponding cypher key value and the value's location (e.g., the first bit cypher key value is 2, the first bit character is interchanged with the second bit character), and so on until all 10 bits are interchanged according to the relevant cypher key value.[3]

B. Transformation to MD5 encryption content

1) A particular character string is enclosed. The basic idea is that the user password is added with a particular prefix or postfix, such as a section specific character string “!m%T#?@)>" before the password, or a string of random code before the password (such as verification code) The user password is lengthened to more than 20 bits before being encrypted using the MD5 algorithm.[3]

2) Additional material determines the character string. The basic idea is that the user name and system time are considered additional details, and that the MD5 algorithm is used to process "user name+ user password+ system time."[3]

3) The additional system created the random character string on its own. The basic idea is that when a user enters a password, the machine generates a random character string R, and when the user enters the password, the password and random character string R are both encrypted using MD5, and two cypher texts are computed per bit.[3]

VI . CONCLUSION

With all of our data being stored in the cloud and on the internet these days, it's more important than ever to keep data protection a top priority. To encrypt private data, the most reliable algorithm should be used. According to recent research, the SHA algorithm should be prioritised over MD5 because MD5 is more susceptible to collision attacks. Researchers, on the other hand, are proposing new algorithms that are more stable and resistant to attacks than SHA256.

REFERENCES

[1] Ah Kioon, Mary Cindy; Wang, Zhao Shun; Deb Das, Shubra (2013). Security Analysis of MD5 Algorithm in Password Storage. Applied Mechanics and Materials, 347-350(), 2706–2711. doi:10.4028/www.scientific.net/amm.347-350.2706

[2] Zheng, Xiaoling; Jin, Jidong (2012). [IEEE 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) - Chongqing, Sichuan, China (2012.05.29-2012.05.31)] 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery - Research for the application and safety of MD5 algorithm in password authentication. , (), 2216–2219. doi:10.1109/FSKD.2012.6234010

[3] ZHENG, Xiao-ling (). [WORLD SCIENTIFIC The International Conference on Computer Science and Technology (CST2016) - Shenzhen, China (8 – 10 January 2016)] Computer Science and Technology - Research on Security of MD5 Algorithm Application. , (), 264–271. doi:10.1142/9789813146426_0030