International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Crypt Analyzing of Message Digest Algorithms MD5 Using Quadratic Salt

Pradeep Singh Solanki[1], Vinit Agarwal[2]

[1]Gyan Vihar School of Engineering and Technology, Suresh Gyan Vihar University, Jagatpura, Jaipur, Rajasthan, India

[2]Suresh Gyan Vihar University, Jagatpura, Jaipur, Rajasthan, India

Abstract: ***Hashing based algorithms are the most commonly used method to strain passwords into hashes which are theoretically non decipherable. This paper proposes a new method and analyses of implementing one more tier to the message digest 5 algorithm using an enhancement of IDEA algorithm, a potential salt by the developer and an basic method to peruse a new root method to set the pattern for two roots as salt into the message digest 5 algorithm.***

Keywords: Data integrity, Authentication, Message Digest, Hashes.

1. Introduction

The title of the present project for the study has keywords the ‘private and public keys’ the ‘Cryptography’ and the ‘http’. The term ‘cryptography’ is made up by combining two words the ‘crypto’ and the ‘graph’; wherein both have been derived from Greek, the ’Crypto’ is standing for the ‘Krypton’ in Greek meaning the ‘hidden’ or the ‘secret’ and the ‘graph’ standing for ‘graph in’ means the ‘to write’. Combining the meaning of the two words together the cryptography conveys the meaning of secret writing; in other words the ‘cryptography’ can be defined as practice and study of creating a secret information. Cryptography, originally, devised to be an instrument for studying the principles and techniques to conceal the information into ciphers to be later revealed to a legitimate people implementing their secret key. It combines the complete space of key-controlled transformations of data in the form of different forms that are not possible or they are computationally irreverent for unauthorized users to copy or undo.

The attacks on hash algorithms are as follows:

1.1 Pre-image Attacks

With pre-image attacking techniques one can find the input that hashes to the pre-specified output.

1.2 Second pre image Attacks

Second pre image attacks are similar to pre image attacks, but here adversary is having additional information of one message that will be hashed to given digest.

1.3 Collision Attacks

With collision attacks an adversary tries to find two messages hashing to same digest. Once he finds two messages, he signs one message but he may pretend that he signed on the other message. Collisions are further divided into three types:

- **Pseudo collisions**: In pseudo collisions the initial values are different while input messages are same.
- **Collisions in compression function**: In these collisions initial values are same and input messages are different. But initial values cannot be selected.
- **Full collisions**: These are similar to collisions in compression function, but here initial values can be selected.

1.4 Birthday Attacks against One-Way Hash Functions

There are two brute-force attacks against a one-way hash function. The first is the most obvious: Given the hash of message, H(M), an adversary would like to be able to create another document, M´, such that H(M) = H(M´). The second attack is more subtle: An adversary would like to find two random messages, M, and M´, such that H (M) = H (M´). This is called a collision, and it is a far easier attack than the first one. The second attack is commonly known as a birthday attack.

1.5 Differential Attacks

Biham and Shamir [14] developed a method for attacking block ciphers, which they call differential cryptanalysis. Differential cryptanalysis for hash functions works as follows: If there is a change input, it produces some XOR difference in the chaining variables. If we can make the XOR difference zero at the end we will get the collision. There are different techniques used for this attack.

2. Related Works

A hash function is a one-way encryption function that takes a variable-size input plaintext m and generates a fixed size hash output. It is computationally hard to decipher the hash and any attempt to crack it is practically infeasible. A “secure” hash function should be able to resist pre-image attacks and collision attacks. Due to the pigeonhole principle and birthday paradox, there will be some inputs that will produce the same hash result. The output length is of fixed size 128 bits, making a total of 2128 possible output hash

values. This value, as big as it may seem, is not infinite, hence resulting in collisions. MD5 (Message Digest Algorithm 5) was designed by Ron Rivets in 1991. MD5 processes a variable-length message into a fixed-length output of 128 bits. MD5 is a popular hash function. It works on blocks of 512-bits, and processes each block through 4 rounds, where each round in turn 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.

3. Countermeasures Research

3.1 Information Entropy

Password strength is usually measured in terms of information entropy. In simple terms, the higher the information entropy, the stronger the password and hence the more difficult it is to crack it. A password of 6 characters would require only 26 attempts to exhaust all possibilities in a brute-force attack, while a password with 42 characters would need 242 attempts.

As can be seen, the longer the password and the larger the character set from which it is derived, the stronger the password. As best practice and preliminary requirement, the application should insist that the user uses a strong password during the registration process. Strong passwords run less risk of existing in dictionaries. Common simple passwords like “123456” have already been banned by Microsoft Hotmail.

3.2 Salting

One of the most common reasons to successful password cracking attacks like the one against LinkedIn was because they were using unsalted hashes. This makes it much easier for hackers to crack the system by using rainbow tables, especially given the fact that many users use very common, simple passwords and these similar passwords result in similar hashes. A salt is a secondary piece of information made of a string of characters which are appended to the plaintext and then hashed. Salting makes passwords more resistant to rainbow tables as the salted hashed password will have higher information entropy and hence much less likely to exist in pre-computed rainbow tables. Typically, a salt should be at least 48 bits.

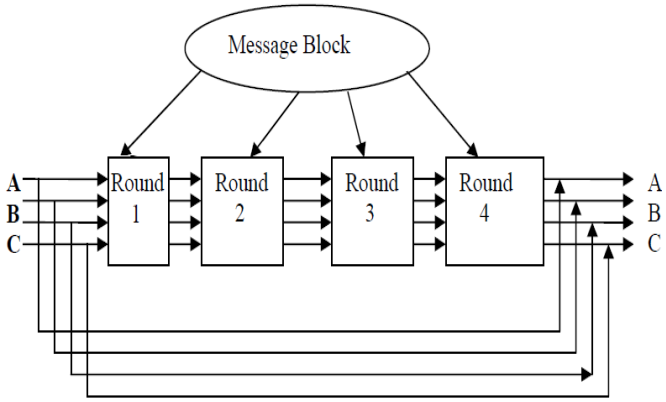

Figure 1: MD5 Main Loop

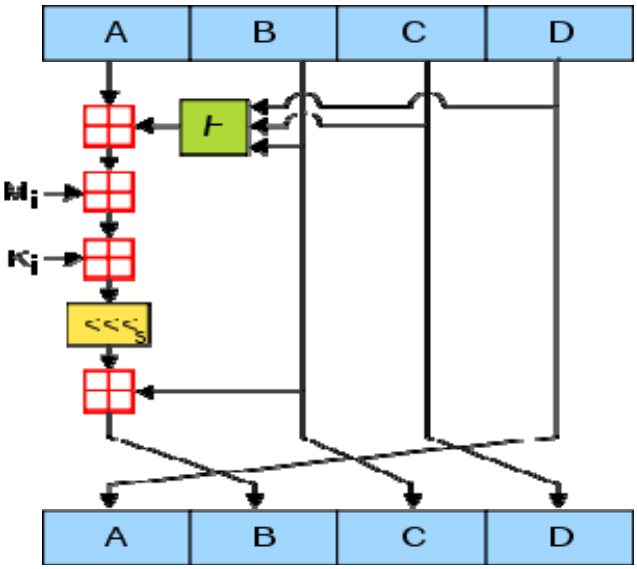

Figure 2: MD5 operation

4. Proposed Algorithm

In our proposed Algorithm to enhance the security and tensile strength of the message digest 5 algorithm we have devised a few changes in the hash input as in the upper section we described the improvement was been introduced by providing Any Hash (password + salt + key), that is the system of manipulation takes the user defined password, a developer defined salt and a key calculated from the system’s input as the initial inputs, here the algorithm has been suggested as inputting a new scenario as providing the salt into the quadratic equation as

Cyper = Root (Salt) (1) where Cyper will have two descendants as the two roots of the equation

Key = Machine Generated () (2)

Hash = Hash (Round {MD5 (password + Cyper[0]+ Key]}) (disintegrator) (1)

Hash = Hash (Round {MD5(password + Cyper[1]+ Key]}) (integrator) (2)

$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

5. Conclusion

The communications via electronic media is growing in importance, and also there is the growing need for data security. Data Encryption promises:

- Only legitimate user can access your data.
- Information cannot be changed or modified by others.
- The security provided by cryptography is beneficial for military.

The time when PGP (Pretty Good Privacy) was invented, the engineers were trying for achieving maximum privacy. IDEA was the optimum candidate for data encryption based

on its reliable built and its great respect. While IDEA is tend to be broken due to some weaknesses we would be implementing a new method of creating a salt for crypt & evaluate its performance with comparison to the https protocol in terms of efficiency.

In our formulation we have designed an algorithm where the security of message digest algorithm has been raised with the help of hashing this algorithm with other mechanism also that is using the salt given from the developer to be implemented with IDEA algorithm and quantifying a value through this procedure. Then this salt which is created as cipher text is now assumed as a qualifier variable that is, it has to be inputted into a quadratic equation. This provides the two roots for the equation. Finally the these roots are imposed into the message digest algorithm, as the first root is applied as salt in the expansion phase, while the second root evaluated from the quadratic equation computation is used as the salt in the decompression system phase in the same algorithm. Therefore at the final stage the view ability of the cipher text has changed and thus no existing frame can decrypt the cipher text created with this implementation entire setup from scratch.

References

[1] T. EL Gamal, (1985) “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”, IEEE Transactions on Information Theory, Vol. 31.
[2] R. Rivest, (1992) “The MD5 Message-Digest Algorithm”, RFC 1321.
[3] T.S. Ganesha, M.T. Fredericka, T.S.B. Sudarshanb, and A.K. Somania, (2007) “Hashchip: A shared-resource multi-hash function processor architecture on FPGA”, The VLSI journal, vol. 40. pp. 11-19.
[4] W. Diffie and M. E. Hellman, (1976) “New Directions in Cryptography “, IEEE Transactions on Information Theory, Vol. 22, No. 6.
[5] B. den Boer, and A. Bosselaers, (1994) “Collisions for the compression function of MD5”, Advances in Cryptology, Proc. Eurocrypt'93, LNCS 765, T. Hellseth, Ed., Springer Verlag,194, pp.293- 304.
[6] H. Dobbertin, (1996) “Cryptanalysis of MD5 compress”. Announcement on Internet.
[7] Hans Dobbertin, (1998) “Cryptanalysis of MD4” Journal of Cryptology Volume-11, Issue 04, pp 253-271.
[8] J. Deepakumara, H.M. Heys, and R. Venkatesan, (2001) “FPGA implementation of MD5 hash algorithm”, IEEE , vol.2, pp. 919 – 924**.**
[9] R. Rivest, (1992) “The MD4 Message-Digest Algorithm”, RFC 1320.
[10] R.L. Rivest, (2001) “The MD4 message Digest Algorithm”, Abstracts Crypto’91, pp.281- 291.
[11] R.L Rivest, (1991) “The MD5 message digest algorithm”, Presented at the rump session of Crypto'91.
[12] A. Menezes, P, van Oorschot, Vanstone S., “The goals of cryptography”, Important Part of Handbook of Applied Cryptography.
[13] R.L. Rivest, (1991) “The MD4 Message Digest Algorithm, Advances in Cryptology”Crypto ’90 Proceedings, Lecture Notes in Computer Science 537, Spring-Verlag, pp. 303-311.
[14] E. Biham, and A. Shamir, (1993) “Differential Cryptanalysis of Full 16-Round DES”, Advances in Cryptology- CRYPTO ’92 Proceedings, Springer-Verlag.
[15] B. den Boer and A. Bosselaers, (1992) “An attack on the last two rounds of MD4, Advances in cryptology”, Proc. Crypto'91, LNCS 576, J. Feigenbaum, Ed., Springer-Verlag,192, pp.194-203.

Author Profile

Pradeep Singh Solanki, born on March 24 1990, in Jaipur, Rajasthan; pursued B.Tech. in Electronics & Communication from Suresh Gyan Vihar University, Jaipur, and currently pursuing M.Tech in Information communication from Suresh Gyan Vihar University, Jaipur, Rajasthan, India.

Vinit Agarwal is a Jaipur based Asst. Professor in Suresh Gyan Vihar University, Jaipur, in the Department of Computer Science.