International Journal of Technology and Engineering System (IJTES)
Vol 7. No.4 2015 Pp. 378-383

available at : www.ijcns.com
ISSN: 0976-1345

IMPROVING ONLINE SECURITY BYUSING CRYPTOGRAPHIC ALGORITHMS FOR EFFICIENT STORAGE OF PASSWORD

S.B.SARAVANAN [1], K.BABU [2],
[1] *U.G.Scholar, B.E.ComputerScience and engineering*
[2] *Assistant Professor, Dept. of Computer Science*
MRK Institute of Technology, Kattumannarkoil
Email:saravanancse.45@gmail.com,babukumarit@gmail.com

ABSTRACT

Cryptographic algorithms are used to store the password in back end with securely and efficiently. Cryptographic hash functions is used to convert variable size of data into fixed size and the word cryptography means scramble the plain text to cipher during store in the database. The passwords are storing in the database in the back end by using the some cryptographic algorithm. But programmers are using only one or two algorithm for converting plain text into scrambled text. But in our concept Passwords are hashed by more than two algorithms. Due to this method hackers can't able to get the password and if unfortunately they get the password and try to crack they will again get the encrypted format. So given input (original password) will be hashed many times like, inputs are given by SHA-512, RIPEMD-60, MD5 (), encode, decode-64 functions for generate strong password.

Index Terms: Cryptography, SHA-512, RIPEMD-60, MD5 ()

1. INTRODUCTION

Nowadays most of all transactions are done by via online. So we are using the more than one algorithm for storing password in the database .Here we are using some good and strong encryption modified MD5()algorithm[vol1]ijirae,SHA512()[vol5(6)]ijcsit,RIPEMD,cryptographic algorithm() Most of hackers get the password from the database but it will in the encrypted form . Hijacker can able to get the password from crypt analytic method .Because of most of database(BACK END) are storing the password by using single algorithm or two algorithm, so due to this reason password will be easily cracked and then do the cybercrime activities . So we are using more than two algorithms for encrypt the password, so hijacker can't get the password easily. We can also use for encrypt the password by using the AES for increase the rounds, but it is very difficult to develop to improve the new algorithm.

1.1 E-EPIDEMIOLOGICAL SCENARIO

Let us consider any bank client A will create the new online banking account on any bank .All banks are using various algorithms are used to store the password in the database. But hackers can easily get the password and decrypt easily. Now the back end process is store the password very efficiently and strongly and doesn't able to crack the password when unfortunately hacker can get then password. Now hacker can get the password and decrypt them then it will gives output again encrypted text due to using more than one algorithm But user can get the password when click the forget password then it could be send the link when we are using the MD5 () algorithm, or send the original password to the user but here we send only the link for creating new password. Due to we are using more than one algorithm. But here we are using three algorithms are used to apply hash function for given password

1.2 CHALLENGES

Two factors to be consider here, first one is database can be secure efficiently by using some complex logics.

Second one is Programmer can think about both client and server, because user can give small password for easily remember, and programmer can give the same security for the weak password, Because nowadays all website can askpassword in caps,small,specialcharacter, numeric then only it will be accepted. But user can't able to easily remember the password that's why we are using more than one algorithm for the single password whether it is small or large password.

1.3 RELATED WORKS

The algorithms are taken from the Piyush Gupta et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 4492-4495,for encrypt the password using MD5() hash function,SHA-512(), Here are my proofs for the SHA1 and MD5 collision attackvulnerabilities:http://www.schneier.com/blog/archives/2012/10/when_will_we_se.html,http://eprint.iacr.org/2010/413.pdf http://people.csail.mit.edu/yiqun/SHA1AttackProceedingVersion.pdf, http://conf.isi.qut.edu.au/auscert/proceedings/2006/gauravaram06collision.pdf and Understanding sha-1 collision weaknessSingle hash function is easy to crack source

2. EXISTING SYSTEM

In old techniques we are using only one are two algorithm by using store the password in the database. The hash functions are applied the passwords at one time or maximum two times.

2.1DISADVANTAGES

- Easy to crack the password.
- User can must give the large password for secure password.
- Able to attack via brute force attack or any other type of attacks (e.g. Trojans, etc.), man in middle attacks etc.
- The well-known hash functions MD5 and SHA1 should be avoided in new applications. Collision attacks against MD5 are well documented in the cryptographic literature and have already been demonstrated in practice. Therefore, MD5 is no longer secure for certain applications.
- Collision attacks against SHA1 have also been published, though they still require computing power, which is somewhat out of scope. As computing power increases with time and the attacks are likely to get better, too, attacks against systems relying on SHA1 for security are likely to become feasible within the next few years.
- When storing password hashes, it is a good idea to prefix a salt to the password before hashing, to avoid the same passwords to hash to the same values and to avoid the use of rainbow tables for password recovery. Unlike suggested in other articles, there is no security advantage in putting the salt in the middle, or even at both the beginning and the end, of the combined salt-password-string.

3. PROPOSED SYSTEM

In our technique we are using more than two algorithms are used to applied hash function. The given password will applied more than two hash function to apply the hashing on the password and the password will store randomly orderlike sequence as

- First the input will give as the md5 () hash function.
- The md5 hash output will again give input of sha-512.
- The sha-512 algorithm is again given input of RIPEMD-60()

3.1 ADVANTAGES

- Difficult to get the password
- Ifunfortunately password will be get it doesn't able to crack them
- If they will decrypt then it will again give the scrambled format
- Attackerdoesn't able to decrypt the password

4. SYSTEM ARCHITECTURE

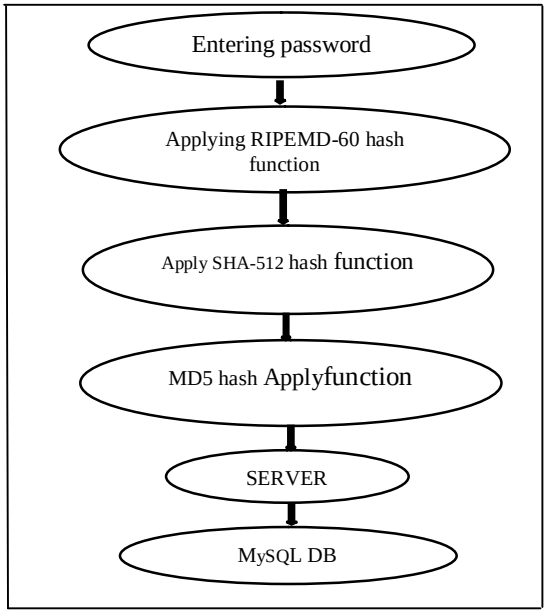

Fig1.1 Diagram for applying hash function sequence

5. MODULES USED

5.1 MODULE1: Given Input Password.
After user login the website then the password will encrypted by using more than one encryption algorithm for hashing the password.

5.2 MODULE2: Apply hash Function on Password of Particular Sequence.
Give the input for first hashing algorithm like as MD5 () hash function. Second the encrypted password will give again input of hashing function SHA-512() algorithm
Third the input again given as RIPEMD160 algorithm
MODULE3: Encrypt the Password by Using More than two Hashing algorithm and Store in Database.
At last the password was stored in the database with strong encryption sequence with high security.
MODULE4: Try to crack the password as a hacker for check the efficiency. Try to decrypt the password that will stored in the database for sample the single encrypted password will give the input md5 () decrypt or sha-512 decrypt for testing the security of the password

HASHING
Hashing is defined as the converting the variable size of data into fixed size of data that will be used into converted scrambled text is only stored in fixed size of bits in server Data BaseMD5 ():-MD5 is a message digest algorithm developed by Ron Rivest, which has its roots in a series of message digest algorithms. MD5 is quite fast and produces 128 bit message digests.
SHA 512:-The SHA 512 algorithm takes a message of length 2^{128} bits and produces a message digest of size 512 bits. The input is divided into blocks of size 1024 bits each.
RIPEMD 160:-At the logic in each of the 10 rounds of the processing of one bit block. Each round consists of a sequence of 16 steps
SALT: -A salt is usually a random string that you add at the end of all you passwords when you hash them. Using a salt means if someone gets your database they cannot check the hashes for common passwords. Checking the database is called using a rainbow table. You should always use a salt when hashing!!

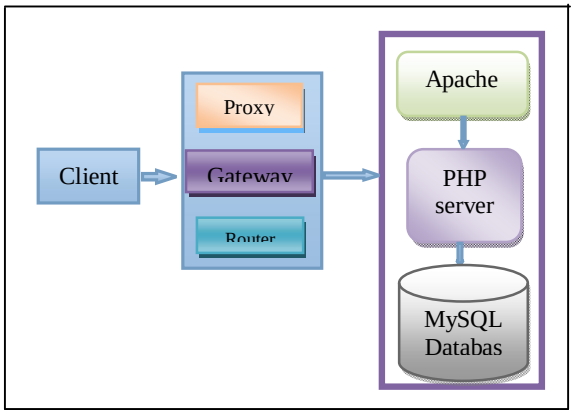

Fig 1.2 Diagram basic client server architecture

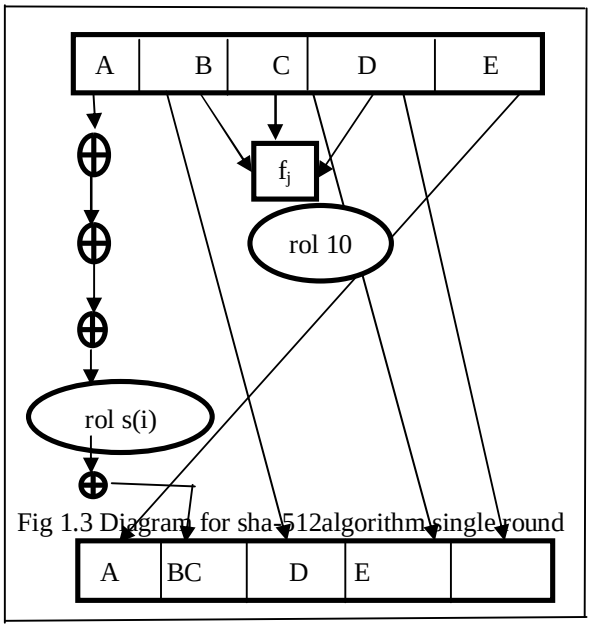

Fig 1.3 Diagram for sha-512algorithm single round

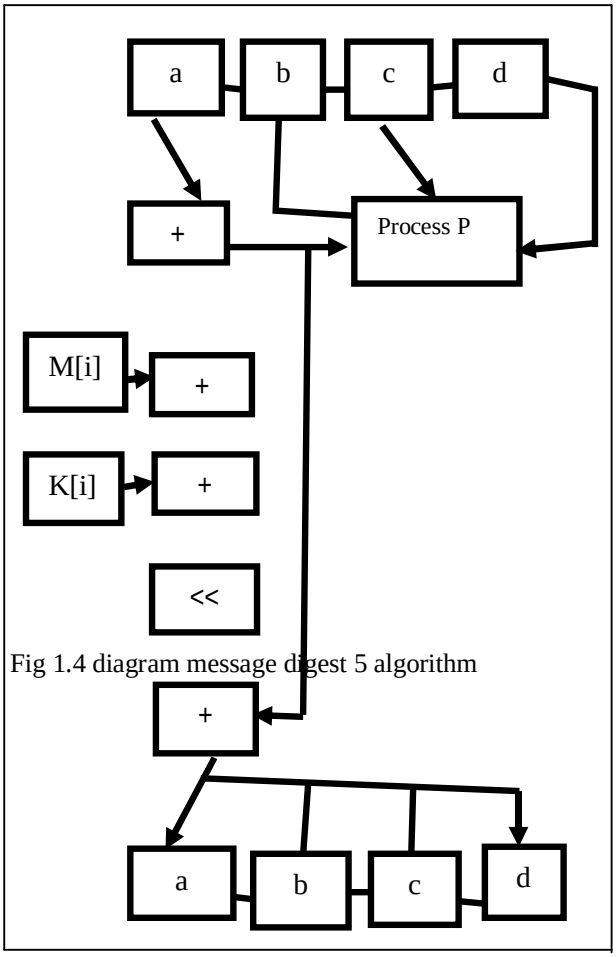

Fig 1.4 diagram message digest 5 algorithm

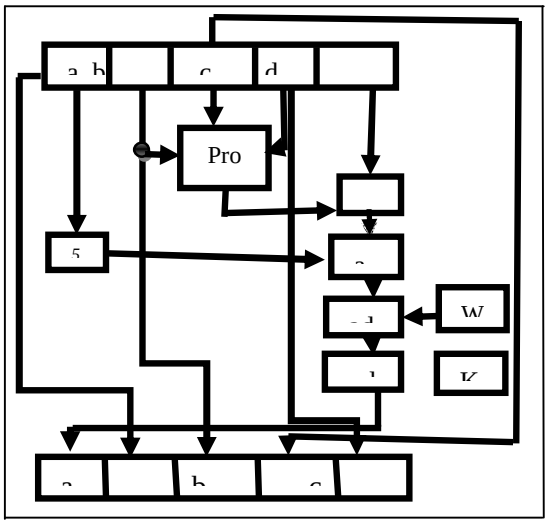

Fig 1.5 ripemd 160 compression function

5. ALGORITHMS

5.1 ALGORITHM FOR SHA-512SECURE HASHING ALGORITHM

Step 1:-Padding

Add Padding to the end of the genuine message length is 64 Bits and multiple of 512.

Step2:- Appending length

In this step the excluding length is calculated

Step3:- Divide the Input into 512-bit blocks

In this step we divide the input in the 512 bit blocks

Step4:-Initialize chaining variables

In this step we initializing chaining variables here we initialize 5 chaining variables of 32 bit each=160 bit of total.

Step5:-Process Blocks

1) Copy the chaining variables

2) Divide the 512 into 16 sub blocks

3) Process 4 rounds of 20 steps each [2].

5.2 ALGORITHM FOR MD5 () HASH FUNCTION

Step 1:- Padding bits and Append Length

Padding of the bits is compulsory with '0' and '1' first and last respectively until the resulting ≠bit length which = 448 mod 512,and the last of bit length of the original message as 64-bit integer. The last bit length of the message which is already padded is 512N for a true integer N.

Step 2:-Divide the input into 512-bit blocks

The message which is already padded is now partitioned into N successive 512-bit blocks m1, m2.....mn.

Step 3:- Initialize Channing variables

Initialization of 32-bit number in the form of chaining Variables (A, B, C, D) these values are represented in hash only

A = 01 17 2d 43

B = 89 AB CD EF

C = FE DC BA 98

D = 76 54 32 10

Step 4:- Process blocks

The four buffers (A, B, C and D) messages (content) are joined now with the input words, using the four auxiliary functions (W, X, Y and Z).4 rounds are performed and each involves 16 basic operations. The Processing block P is applied to the four buffers (A, B, C and D), by using message word M[i] and constant K[i]. The item "<<<s" denotes a binary left shift by s bits. The four type of IRF (info related functions) thateach take as input three 32-bit words and produce same bits of output i.e. 32-bit word.

They apply the logical operators ^, v! Andxor to the input bits.

Q (A, S, D) = AS v not (A) F

W (A, S, D) = AS v S not (F)

E (A, S, D) = A xor S xor F

R (A, S, D) = S xor (A v not (F))

The bits of A, S, and D are totalitarian and balance the each bit of Q (A, S, D) will be totalitarian and balance.

The functions (A, S and D) = P, in that they do job in "Bitwise parallel" to produce the reliable output from the bits of A, S and D.In such a way that if the be similar bits of D, E and F are autarchic and balanced, then each bit of W (A, S, D), E (A, S, D) and R (A, S, D) will be totalitarian and balance.

Step 5:- Hashed Output

There are 4 rounds performed in message digest 5 (MD5) which is of 128 bits. Fig 1 shows One MD5Operation [1] [2]

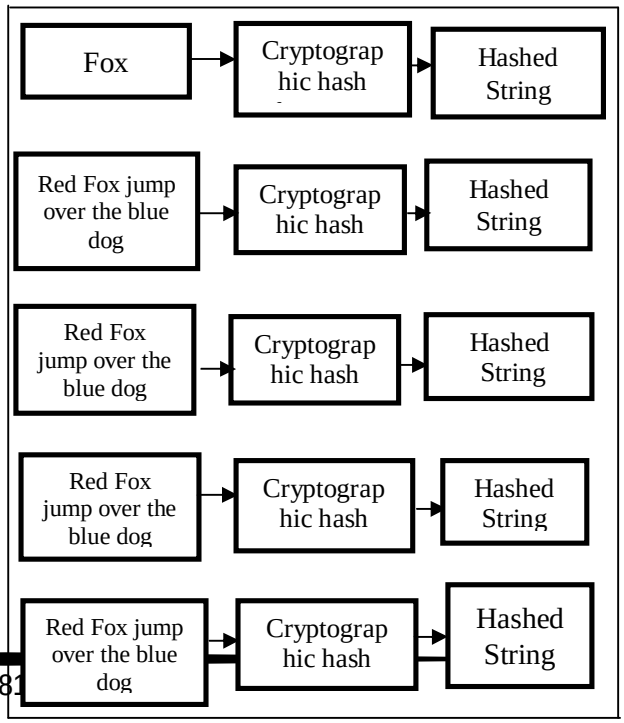

FIG 1.5 DIAGRAM FOR HASH FUNCTION

Sample Hashed String:-

saravanan123 (Password that stored in the database)➔c7290bcd2a064c3e24c3f1b4f6f0bb40

6. PARAMETERS USING MD5 () AND SHA 512 HASHING FUNCTION

6.1 PARAMETERS USING MD5 ()

Below equation shows a single MD5 operation.

1) Default Parameters

a = b + ((a + Process P (b, c, d) + M[i] + t[k]) <<< s)

Here: - a, b, c, d = are Chaining variables

Process P=A nonlinear operation

M[i] =For M [q x 16 + i], which is the

i^{th} 32-bit word in the 512-bit block of the message t[k] =a constant

<<<s =circular-left shift by s bits [2].

2) Actual Parameters.

Key Length: 64 bits, 128 bits, 256 bits, 512 bits

Block Size: 128 bits

Cryptanalysis: Resistance Strong against Digital Certificate and very fast on 32bit machines Security Secure

Rounds: 4

Steps: 16

6.2 PARAMETERS USING SHA ()

Below equation shows a single SHA operation.

1) Default Parameters.

abcde (e+process p_s5 (a)+W[t]+k[t]),a, s30 (b), c, d

Here:-

a, b, c, d, e =chaining variables

Process p =status of logical operations st =<<<

W[t] =derived other 32 bits bytes

K[t] =five additives constants are defined [2] [3].

2) Actual Parameters.

Key Length: 128 bits

Block Size: 160 bits

Cryptanalysis: Resistance Strong against Digital Certificate.

Rounds: 4

Total Steps: 20

7. CONCLUSION

Unfortunately the data will be cracked by cybercrimes that the time the data will be secure and safe. At the end of my project I will give the full assurance of secure storing of the data that will in the both client and server side and the project is open source i.e. PHP is technology I will used the server side scripting language

8. FEATURE ENHANCEMENT

Security is our focus for safety and secure use of internet. Today internet plays important role for large type of online transactions. So our project can be able to enhance the more security by using in future by new algorithms so database security will also be secured with strong encryption.

9. REFERENCES

[1] Rivest R., 1992, “The MD5 Message-Digest Algorithm,”RFC 1321,
MIT LCS and RSA Data Security y, Inc.

[2] Kahate, Atul, 2003, "Cryptography and Network Security y", Tata McGraw-Hill, India.

[3] Kasgar A. K., Agrawal Jitendra, Sahu Santosh, 2012, “New Modified
256-bit MD5 Algorithm with SHA Compression Function”, IJCA
(0975–8887) Volume 42 (12), pp47-51.

[4] William Stallings, Cryptography and NetworkSecurity:Principles and Practice, 5th
Edition Prentice Hall; 5 edit ion (January
24, 2010).

[5] Vandana P., V.K Mishra, Architecture based on MD5 and MD5-512
Bit Applications,IJCA (0975 – 8887) Vol. 74– No.9, July 2013.

[1] R. Rivest, “The MD5 Message-Digest Algorithm,” RFC 1321, Apr. 1992.

[2] H. Dobbertin, A. Bosselaers and B. Preneel, “RIPEMD-160: A Strengthened Version of RIPEMD, Fast Software
Encryption,” LNCS 1039, pp. 71-92, Springer-Verlag, 1996.

[3] W. Stallings, Cryptography and Network Security, 2nd ed., Now York: Prentice-Hall, 1997.

[4] S. Dominikus, “A hardware implementation of MD4-family hash algorithms,” Proc. 9th Int. Conf. on Electronics,
Circuits and Systems, vol. 3, pp. 1143-1146, 2002.

[5] Chiu-Wah Ng, Tung-Sang Ng and Kun-Wah Yip “A UNIFIED ARCHITECTURE OF MD5 AND RIPEMD-160
HASH ALGORITHMS”, Department of Electrical & Electronic Engineering, The University of Hong Kong Pokfulam Road, Hong Kong

[6] Anh Tuan Hoang, Katsuhiro Yamazaki and Shigeru Oyanagi “Multi-stage Pipelining MD5 Implementations on
FPGA with Data Forwarding“, Ritsumeikan University, 1-1-1 Noji Higashi, Kusatsu, Shiga 525-8577, Japan

[7] Changxin Li1, Hongwei Wu2, Shifeng Chen1, "Efficient Implementation for MD5-RC4 Encryption", Xiaochao
Li2* and Donghui Guo1,2
[8] Guang Hu, Jianhua Ma and Benxiong Huang, "High Throughput Implementation of MD5 Algorithm on GPU"
Guang Hu, Department of Electron and Information, Huazhong University of Science and Technology, Wuhan,
China huguang@mail.hust.edu.cn, Jianhua Ma, Faculty of Computer and Information Sciences, Hosei University,
Tokyo 184-8584, Japan, jianhua@hosei.ac.jp
[9] Benxiong Huang, Department of Electron and Information, Huazhong University of Science and Technology,
Wuhan, China, huangbx@mail.hust.edu.cn
[10] Alok Kumar Kasgar, Jitendra Agrawal, Santosh Sahu , "New Modified 256-bit MD5 Algorithm with SHA
Compression Function ", School of IT School of IT School of IT Rajiv Gandhi Technical University Rajiv Gandhi
Technical University Rajiv Gandhi Technical University Bhopal (M.P.) Bhopal (M.P.) Bhopal (M.P.)
[11] Chiu-Wah Ng, Tung-Sang Ng and Kun-Wah Yip "A UNIFIED ARCHITECTURE OF MD5 AND RIPEMD-160
HASH ALGORITHMS", Department of Electrical & Electronic Engineering, The University of Hong Kong Pokfulam Road, Hong Kong.
[12] F. Chabaud, A. Joux. "Differential Collisions in SHA-0". In Advances in Cryptology CRYPTO'98, Santa Barbara,
CA, Lecture Notes in Computer Science 1462. Springer-Verlag, NY, pp. 56–71, 1998.
[13] E. Biham, R. Chen, A. Joux, P. Carribault, W. Jalby and C. Lemuet. "Collisions in SHA-0 and Reduced SHA-1- In
Advances in Cryptology" – Eurocrypt'05, Springer-Verlag, 2005.
[14] NIST FIPS PUB 180-1. Oct. 2001.
[15] NIST, "Secure Hash Standard (SHS)", FIPS PUB 180-2, 2002.
[16] S. Chang, M. Dworkin, Workshop Report, The First Cryptographic Hash Workshop, Report prepared, NIST 2005.
[17] E. Biham, R. Chen, "New results on SHA-0 and SHA-1" Crypto 2004 Rump Session, Aug. 2004.
[18] K. Matusiewicz and J. Pieprzyk "Finding good differential patterns for attacks on SHA-1" eprint 2004 Available :
http://eprint. iacr. org/2004/364. pdf.