International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-10 Issue-8, June 2021

Security Design Pattern for Login System through Authentication using Modified Sha-384 Algorithm

Yogini C.Kulkarni, S.D. Joshi

Abstract: ***The research has been carried out to develop secure login system by authenticating the login using modified SHA-384 algorithm. It derives 896-bit hash value for the password entered by the user in the user registration form and saves the credentials entered by the user in system's database. Results obtained are evaluated by resolving the general attacks confirmed that the modified SHA-384 algorithm was more secured compared to the original SHA-384 algorithm as it was not broken using generic attacks such as brute force, rainbow table and other cracking tools available online such as Cain and Abel. The performance of the modified algorithm was measured with only 2 ms additional execution time from SHA-384.***

Keywords: Modified SHA-384 hash algorithm, attacks, login authentication

I. INTRODUCTION

Security of the software is considered as an additional non-functional requirement. To fulfil the security aspect patches are applied for maintenance of the software. This strategy is expensive and time consuming for both the sides of the software developers as well as the user side if the errors are found after the implementation of the software system. Hence it is essential to take into consideration the security approach using existing techniques to be applied to protect the software since beginning of the software development at every stage of the software development life cycle. Various methods and techniques are available for applying the security aspect throughout the life cycle development of the software.

In the domain of cryptography, cryptographic hash functions play a key role. The hash functions are used in many applications like digital signature, key generation, message authentication etc. Hash functions takes a message of variable length and produces output of fixed length which is referred as hash code of the input message. The generated hash code could not be deduced s done in encrypted message which can be decrypted to get the original message. This is why the hash code is also called as digital finger print of the message. Lot of approaches are available for generating the hash values using various versions of SHA algorithms. This paper presents the method of developing the secure login system using hash functions to secure the client server communication. The SHA-2 algorithm generates an encrypted message which is called as message digest. But due to short hash value SHA-2 is not perfect as the generated hash code can be cracked easily using rainbow table and brute force attack. Also the researchers have found the first collision attack against SHA-384 hash function. Hence there are many modifications done in SHA-384 to prevent the attacks. [1]

Manuscript received on June 17, 2021.
Revised Manuscript received on June 21, 2021.
Manuscript published on June 30, 2021.

* Correspondence Author

Yogini C. Kulkarni*, Department of Computer Engineering, Research Scholar, JJT University, Jhunjhunu (Rajasthan), India. E-mail: yoginikulkarni77@gmail.com

Prof. Dr. S.D. Joshi, Department of Computer Engineering, Bharati Vidyapeeth (Deemed To be university), Pune (Maharashtra), India. E-mail sdj@live.in

II. REVIEW

Various cryptographic methods and their logical characteristics are analyzed in the security pattern repository to find out the specific cryptographic technique that would finally help in the later phases of the design process of secure software development. With this review the SHA-384 is finalized as cryptographic algorithm for generating the hash code and enhancing the security of the software.

The following table summarizes the properties of the SHA-2 family hash algorithms.

Table 1 Properties of the hash algorithms

Algorithm	Message Size	Block Size(n bits)	Word Size (In bits)	Message digest size	Number of rounds	Security Size(bits)
SHA-256	< 2 ^ 64	512	32	256	64	128
SHA-384	< 2 ^ 128	1024	64	384	80	192
SHA-512	< 2 ^ 128	1024	64	512	80	256

The different columns in the above table stands for:

1.Message Size: It shows the capacity of the algorithm to handle the upper bound limit of the message.

2.Block Size: It is the size of each bit block that the message is divided into the number of blocks.

3.Word size: It refers to the word size that is being processed during the processing of generating of the message digest.

4.Security: This column shows how many messages to be generated before two can found the same hash code.

While software development to design the security pattern lot of research has been conducted using SHA algorithm. Some of the papers are studied to discover the limitations to enhance the security in the client-server communication.

[1] Proposes the modifications in the MD5 algorithm. The modifications are suggested generating the hash value using compression function by combinations of the OR and operations. Hash code is used to assure the authentication of the signing user. This new approach proposed by the author generates an output of message digest of 512-bits.

[2] Presents a new approach for generating the hash code. The authors suggested a method for generating the message digest that is of length 192-bits in size and used the expansion function for extending the hash length to increase the security of the system. This algorithm is similar to SHA-1. The expansion function is added by increasing the number of chaining variables to expand the original length of SHA-1 192 bits.

[3] According to **[3]** the SHA-192 can be cast-off in many applications such public key cryptosystem, digital signcryption, message authentication etc. The authors have proposed a new technique for digital signature using SHA-192 hash algorithm.

[4] Proposed a novel message digest algorithm by modifying the MD5 hash algorithm. Using some combinations, the message digest length is increase up to 256-bits to enhance the security. The author has verified the algorithm by resolving the general attacks link Rainbow table and Brute Force attack.

Secure Hashing Algorithm is a hash function in cryptography which converts an input into a 20- byte (160-bit) hash equivalent known as a message digest consistently concluded as a hexadecimal number, 40 digits long. In this research we proposed modified SHA-384 algorithm. SHA 384 is cryptographic hash function designed by NSA and belongs to the SHA 2 group of cryptographic hashes. It produces 384-bit (48-byte) hash value which is represented in hexadecimal number of message digest of 96-bit long. The leftmost 4 words are truncated from the output of SHA-512 to get the hash code of 384 bits (512-64=384). To produce a SHA-384 hash, a SHA-512 hash is used, then 128 bits of the result are discarded, leaving 384 bits. SHA-384 is a truncated version of SHA-512.The SHA-384 and SHA-512 algorithms have similarities, but they differ in their speed and security level. That's why it's better to always use SHA-384 instead of the other variants.

III. SECURE LOGIN SYSTEM THROUGH AUTHENTICATION BY MODIFYING THE SHA-384 ALGORITHM

The objective of this case study is to modify the existing SHA-384 to generate the hashing algorithm of 892 bits' size to enhance the strength of security and to prevent the system from the attacks. Using SHA-384 the modification is done by expanding the original message digest of 384-bits into 896 - bits by recursively doing XOR and AND operation using 16 buffer registers each of 7 bytes for storing the hash code and generating new in each iteration. Total 80 rounds are used and each round comprises 20 steps for regenerating the hash code.

The security factor depends on the length of the message digest generated by the hash functions which is restricted by the size of input to the algorithm. By modifying the SHA-384 to extend the length of hash code, the security gets more tightened and attacker neither break the password nor guess the original message.

As we are providing two level security, the obtained results show this web based secure login system provides better security than the existing one. The simulated results of proposed extended SHA-2 are analysed and tested against the general attacks like Brute force attack, Rainbow table attack, Cain and Abel attacks. From the experimental results, it is identified that the modified SHA-384 is the best algorithm to secure the system.

3.1 Methodology

The recommended structure of extended and modified SHA-384 algorithm is similar to SHA-384 algorithm excepting that it has modified by adding 512 bits to generate 896-bit hash value presented in hexadecimal number 96-digits message digest. The elementary functional block processing of message is shown in Figure 1.

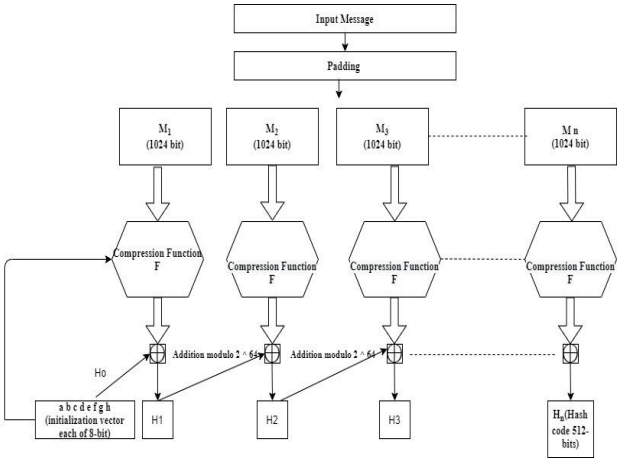

Fig.1 Elementary functions for modified SHA-384 algorithm [5]

The modified SHA-384 uses the padding algorithm, breaking the message into 1024-bit blocks and adding the length as a 128-bit number at end. The output transformation is used in a final step to map the n bit to variable length m bits' results called the message digest. The hash value obtained from SHA-384 is extended by adding 512-bits to expand to 896-bits which is rendered further to produce a 96-bit bit message digest. It is same as SHA-384 message digest function, but execution time is more but more secure [3] .

The proposed modified SHA-384 algorithm has following three processing steps:

- **Pre-processing:** This involves padding, parsing the padded message into m- bit block and setting initial values to be used in iterated processing.
- **Iterated processing:** The iterating process has 80 steps in all and in each step there is a fundamental function which calculates a message digest every time and sends it to the next step.

Output transformation: To fulfil the security requirement of the algorithm the length of the message digest should be increased to make it more difficult to crack and recognize the original message.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

We have proposed a new hash algorithm that undertakes an important change in the elementary function of the secure hash algorithm and gives us a message digest of length 896-bits.

Proposed integrated architecture that can perform two hash functions using the SHA-384 and SHA-(384+512) bit for generating the password to avoid different common attacks. The new approach can generate an output of 896-bit size. The SHA-384 is enhanced by processing the original message into 64-bit word (1024-bit chunks). Expand these 16 words into 80 words by mixing and shifting using operations AND and OR operations using 80 rounds and combined it with the original SHA-384 to ensure message integrity and making it a more secured hash algorithm. The figure 2 shows the general model of modified SHA-384.

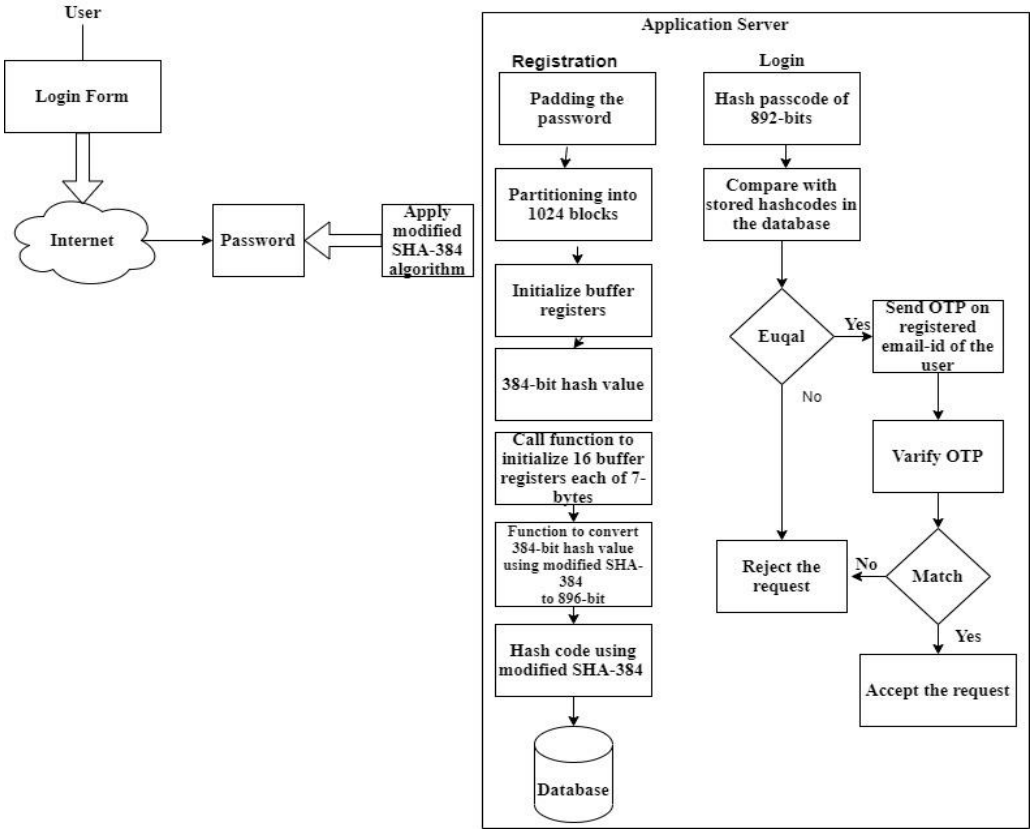

Fig.2 General model of modified SHA-384 algorithm

3.2 Algorithm

The proposed algorithm is being separated into two sections. First section involved the pre-processing of the message and second section includes the login authentication pattern for secure login system [6].

3.2.1 Phase 1: Pre-processing of the message using modified SHA-384

It involves following steps:

Step 1. Padding

Step 2. Parsing the padded message into 1024 blocks.

Step 3. Initializing the initial vectors for using in the iterative processing. There are 80 rounds for processing of the hash code. For this the initialization vectors a,b,c,d,e,f,g,h each of 16-bits are initialized and assigned the constants.

Step 4. Updating the hash values of previously generated message block by adding it into the temporary registers.

Pre-processing of the message digest generated by padding the message length and then parsing it into 1024 blocks shown in the following section.

- **Padding the Message**

If the length of the message generated is exactly equal to block length then it is kept as it is, but if it is less than the block size then it should be padded by adding 1 followed by all zeroes On the other hand if it is greater than the block size then it is shortened by using hash value of the message. To ensure that the length of the message is multiple of 512, padding is required. If the length of the message is M, then 1 bit is appended to the end of the message followed by k zero bits, where k > 0. For e.g Suppose the length of the message M, in bits, is n bits. Append the bit "1" to the end of the message, followed by k zero bits, where k is the smallest non-negative solution to the equation n +1+ k ≡ 896 mod 1024. Then append the 128-bit block that is equal to the number n expressed using a binary representation. For example, the (8-bit ASCII) message "abc" has length 8 × 3 = 24 . Each character is expressed as an 8-bit binary block, using the ASCII Values. The numbers 10 and 2 in subscript represent the Decimal and Binary ASCII Values respectively.

- **Expanding the Padded Message Of Hash Size SHA-384**

After a message has been padded, it must be parsed into N m-bit (64-bits) blocks before the hash computation can begin.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

The "abc" message has a single (N=1) 1024-bit block (ASCII code of "abc" is converted into hexadecimal), Parse the 1024-bits into 16, sequence of 64-bit words M (1),M (2),.....M(15), (in hexadecimal):

61626380	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000018

These 16, 32 bit words are expanded to 64-bit words W, W0, W1, W2 ....Wt followed by hash computation process as shown in figure 3

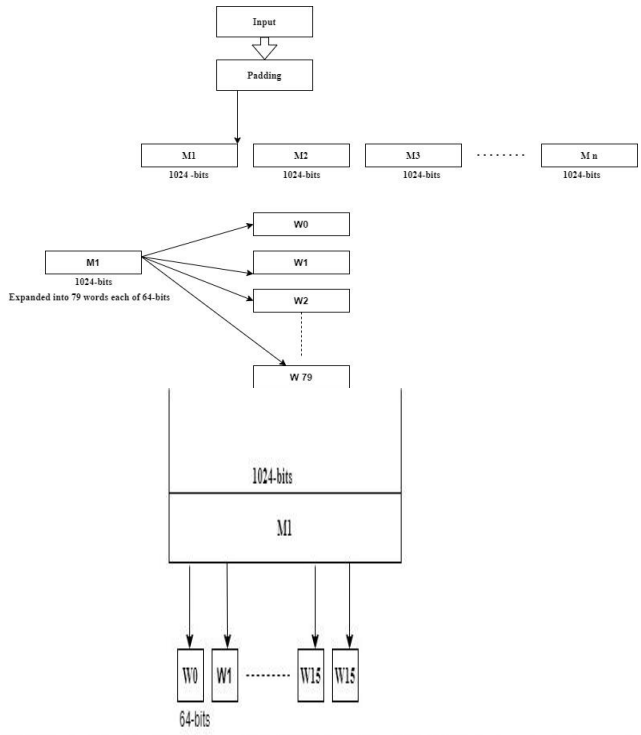

Fig.3 Expansion of message into 64-bits word

The message blocks are processed one at a time, beginning with the initials hash values called message digest buffer.W_t is used in each of the 80 rounds of each block where t = 0 to 79. Each W_t is of 64-bit in length. W_t is calculated as follows.

First 16 Wt s (0 to 15) are taken as it is from the message (16*64) . Rest of the W_t are calculated from the following formula.

$$W_t = W_{t-16} +_{64} \sigma 0 (W_{t-15}) +_{64} (W_{t-7}) +_{64} \sigma 1 (W_{t-2})$$

where

$$\sigma 0(x) = ROTR^1 (x) \oplus ROTR^8 (x) \oplus SHR^7 (x)$$

$$\sigma 1(x) = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6 (x)$$

$ROTR^n(x)$ = circular right shift of the 64 bit arg by n bits

$SHR^n(x)$ = right shift of the 64 bit arg by n bits with padding by zeros on the left

+64 = addition module 2 ^ 64

For example W_{60} can be calculated from previous generated four words as

$$W_{60} = W(60\text{-}16) +_{64} \sigma 0\ W(60\text{-}15) +_{64} W(60\text{-}7) +_{64} \sigma 1\ W(60\text{-}2)$$

$$W_{60} = W(44) +_{64} \sigma 0\ W(45) +_{64} (53) +_{64} W(58)$$

Reset of Hash buffers with initialization vectors

The initialization vectors are declaredusing the register variable a,b,c,d,e,f,g,h each of 8-bits in length. The hash buffers H^1 through H^n shown are used for storing the intermediate results of previous processing and used for generating the next hash code. As there are 8 register variables declared, its required to initialize 8 initial vectors H^0 (0) to H^7 (0) and K_0 to K_{79} constant values to avoid any asymmetry. The last hash buffer stores the final hash code resulting from entire rounds used for processing of the hash code. They are calculated by the fractional parts of the square roots of the first 8 prime numbers i.e 2 to 19 (SHA-512) or the next 8 prime numbers i.e 23 to 53 (SHA-384) and of the cube roots of the first 80 prime numbers. The result of square root of the prime number is converted into hexadecimal. Fractional part is considered as the initial vector by putting the result in big endian format. For e.g the initial vector value for a will be calculated as follows.

Square root of (23) = 4.79583152331

Hex value of (4.79583152331) = 4 .

CBBB9D5DBE0823390EF

Only the fractional part is considered as the initial vector value for a, i.e ,in big endian format the value for a = cbbb9d5d

- **Updating the Hash Values**

The hash values are updated by following 80 rounds to process the hash code. So 80 constants are to be declared for generating 80 rounds. These 80 constants are generated by taking the cube root of first 80 prime numbers from the range 0 to 512(2,3….409) . The first 64-bit fractional part is considered as the constant. For e.g suppose the last prime number is 409 , then taking cube root of 409 and converting into hexadecimal ,the fractional part is considered as the 80th constant as shown below.

Cube root of 409 = 7.42291412044

$(7.422914120\ 44)_2 = 111.0110110001000100001$

$(7.422914120\ 44)_{16} = 7.6C44198C4E70910DB51A$

Only fractional part is considered as the 80th constant.

Hence 80th constant = $(6C43D46B26BF8769EC2D)_{16}$

All operations involved in generating the hash code are same as in SHA-512, only the difference is that last 128 bits that is the values of last two registers are truncated so as to get the hash code of length 384-bits.

3.2.2 Phase 2: Login authentication using modified SHA-384

To get authenticated for login into the system the users are initiated to generate the password with combination of characters and numbers which is difficult to identify by the third person. This will circumvent the attackers from predicting the precise password.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

The script written to generate the hash value in the server will hash the password entered by the and verify the hash value with the stored hash value (896 bit) of the password. If both the hash values are unequal, then the server will reject the request from the user. The second level authentication is done in continuation with this by directing the user to enter OTP received at user's registered email id. The server will grant the request of login if the OTP is matching otherwise user's request will not be accepted.

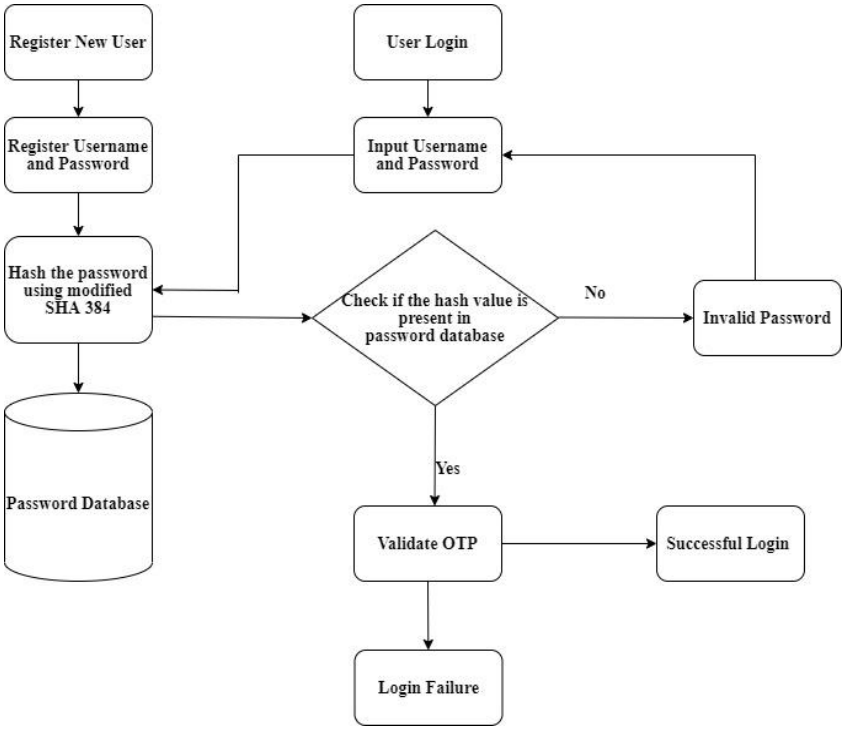

Fig.4 Login authentication using modified SHA-384

IV. EVALUATING THE RELIABILITY OF MODIFIED SHA-384 BY RESOLVING THE ATTACKS

To check the reliability and performance of the algorithm the possible general attacks must be tested using several online tools. Crackstation is one among the most powerful online password cracking and guessing tool. This tool uses different technique like brute force attack, Rainbow table attack, dictionary attack etc. for guessing the password .To break the password Crackstation utilizes pre-computed lookup table which contents stored password hash and exact password for hash. While searching the password the hash codes are indexed for matching the required hash. If the hash is found, then the passwords are mapped and can be obtained easily.

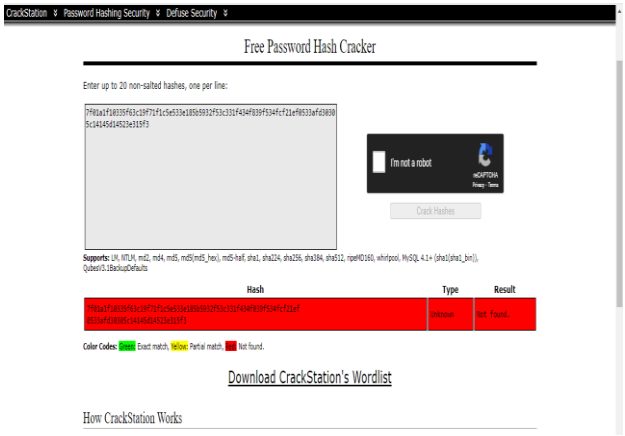

Fig. 5 Testing the password using Crackstation

One more tool is used to check the reliability of the modified SHA-384 using Cain and Abel password cracking tool [1] .This is the password cracking and password recovery tool for Microsoft Windows . As shown in figure 6 and 7, when the word "a" ,"abc" was converted into its equivalent hash value, findings discovered that it produces a hash value of which is different from the hash value of the modified SHA-384.

Table 2 Using the Hash calculator to convert the text "a" and "abc" into hash value

Text	Hash Value(SHA-384)
"a"	54A59B9F22B0B80880D8427E548B7C23ABD873486 E1F035DCE9CD697E85175033CAA88E6D57BC35EF AE0B5AFD3145F31
"abc"	CB00753F45A35E8BB5A03D699AC65007272C32AB0 EDED1631A8B605A43FF5BED8086072BA1E7CC235 8BAECA134C825A7

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

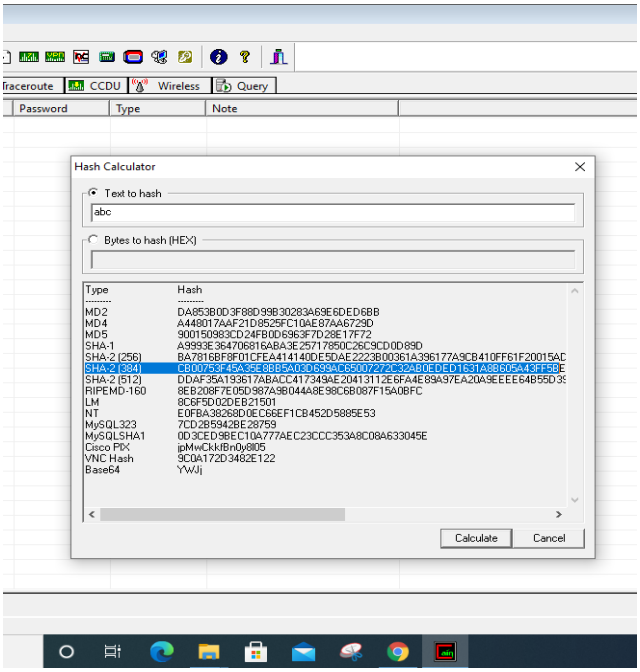

Fig. 6 Using the Hash calculator to convert the text "a" into hash value

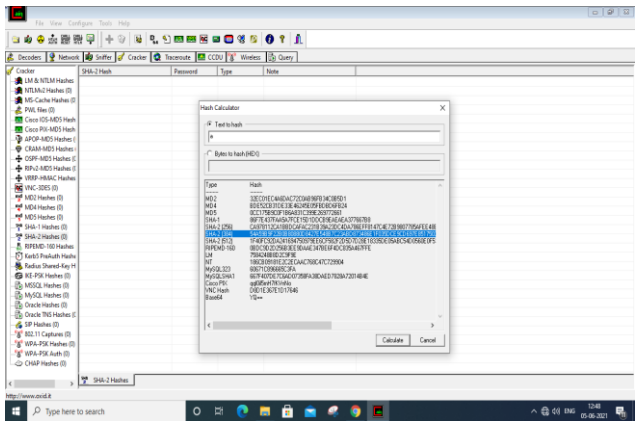

Fig. 7 Using the Hash calculator to convert the text "abc" into hash value

V. RESULTS AND DISCUSSION

This research work carries out the development of the security model for incorporating the security accomplishments during the software development. The major attention must be given on security aspect to lessen the security faults at every phase of the software development lifecycle. The recognition of the security requirement before the system design helps the software developer to uphold the security thread. To identify the appropriate cryptographic technique, the security services are planned to improve security design pattern for web based secure login system. We have proposed two level security implementation by the modifications in the SHA_384 algorithm to extend the length of the message digest to tighten and maintain the security of the software. The results obtained and evaluated against the Brute force attack and Cain and Abel password cracking tool.

REFERENCES

1. Esmael V. Malibera , Ariel M. Sison1 , Ruji P. Medina," Securing a Client-Server Communication through Login Authentication Using Modified MD5" Proceedings of 2018 the 8th International Workshop on Computer Science and Engineering (WCSE 2018)
2. Harshvardhan Tiwari, Dr. Krishna Asawa, "A Secure Hash Function MD-192 With Modified Message Expansion", (IJCSIS) International Journal of Computer Science and Information Security, Vol. VII , No. II, FEB2010 .
3. 3.Thulasimani Lakshmanan1 and Madheswaran Muthusamy ,"A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes". The International Arab Journal of Information Technology, Vol. 9, No. 3, May 2012
4. Alok kumar kasgar, Jitendra Agrawal, Santosh Sahu ."New Modified 256-bit MD5 Algorithm with SHA Compression Function" International Journal of Computer Applications (0975 – 8887) Volume 42– No.12, March 2012
5. H. B. Pethe1 , Dr. S. R. Pande." An overview of Cryptographic Hash Functions MD-5 and SHA", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727
6. Esmael V. Maliberan," Modified SHA1: A Hashing Solution to Secure Web Applications through Login Authentication", International Journal of Communication Networks and Information Security (IJCNIS),2019

AUTHORS PROFILE

Yogini Kulkarni, received her B.E. degree in Computer Engineering from Shri Sant Gajanan Maharaj College of Engineering, Shegaon from Amaravati University, Amaravati, Maharashtra in 1990 , M.E. (Computer Engineering) Degree from Bharati Vidyapeeth Deemed University Pune. She is currently working as Assistant Professor in Information Technonolgy Department , Bharati Vidyapeeth (Deemed To be University) College of Engineering, Pune since 2010. Currently she is pursuing Ph.D.from Shri Jagdishprasad Jhabarmal Tibrewala University, Rajasthan . Her research interests include software engineering. She is a sincere teacher devoted to Education and Learning for the past 27 years. Her research projects include Software Engineering and Image Processing.She will be available at yckulkarni@bvucoep.edu.in for any further communication.

Shashank Joshi, received his B.E. degree in Electronics and Telecommunication from Govt. College of Engineering, Pune in 1988, MBA (Systems) from Pune University, M.E. (Computer Engineering) and Ph. D. (Computer Engineering) Degree from Bharati Vidyapeeth Deemed University Pune. He is currently working as the Dean,Faculty of Engineering and Technology, Bharati Vidyapeeth (Deemed To be University) Pune and Professor in Computer Engineering Department Bharati Vidyapeeth (Deemed To be University) College of Engineering, Pune since 1990. His research interests include software engineering. Presently he is engaged in SDLC and secure software development methodologies. He is innovative teacher devoted to Education and Learning for the last 30 yrs. His research projects include Software Engineering, Object Engineering, Software Project Management, RTC Systems, Software security, Visual and Generic Modeling. He will available at sdj@live.in for any further communication.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication