An Algorithm for Securing User Credentials by Combining Encryption and Hashing Method

Awlad Hossain[1], Hasibur Rahaman[1], Arafat Jamil[1], Dr. M.A. Khan[1, 2*]
[1]Dept. of Electronics and Telecommunication Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj, Bangladesh
[2]Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong
[*]corresponding authors: asad.khan@bsmrstu.edu.bd, arzu1013@gmail.com

Abstract –*Internet browsers, run on electronic devices usually preserve sensitive information, for example, client identifications (passwords). In the present-day innovation, most of the internet browsers uses password logins as an initial validation to demonstrate a client's individuality. Passwords perform as the primary safeguard against provoker manipulation. However, some internet browsers usually preserve client passwords in the records as plaintext. The client's saved password help the attackers to promptly get client accreditations and break it. This paper proposes a new algorithm for the security of user credentials using the encryption and the hashing method. Specifically, the motivation behind this technique is to confirm client identifications against data fraud. This approach ensures client accreditations utilizing a new algorithm that at the first stage keeps a password and then encrypt the password. After that, the respected encrypted data is hashed and sent to the internet browser server to store. This devised method is easy to develop and executed using the internet browser.*

Keywords: *Encryption, Hashing, Security SHA 256, User credential*

Article History
Received 22 July 2020
Received in revised form 29 September 2020
Accepted 30 September 2020

I. Introduction

As cyber-attacks rapidly increased and which open the door for hackers. Most basic susceptibilities in internets are extensive and which lead to serious security problems. From that point, intruders misuse these imperfections in Web applications with the guide of toolboxes. The presence of these security problems makes a need to utilize passwords as validation in Web applications. In this way, passwords were initialized to check a client's individuality during verification.

Consequently, password hashing method is an important issue. In hashing method, passwords are changed to hash values which are not shown as plaintext in databases. This paper introduces, examine, and investigate the constraints of existing approaches. And uses a SHA256, md32 and encryption protocols to secure the client identifications in the Google Chrome browser.

A. Rationales of the study

This paper starts off with giving an overview of sensitive data, encryption algorithms, hashing methods and another required knowledge. The aim of this paper is to find the best way to encrypt data / hashed data with a focus on protecting sensitive data in web applications. Data sent through the URL or body of a web application could be sensitive data, and exposure of sensitive data can be devastating for governments, companies and individuals. As we are introducing a new algorithm through which we are providing optimal security to the user credentials, it eliminates the devastating exposure of user credential. This algorithm will be very important for those organizations that require higher use credential security where the execution time is not a major factor.

The objective of our study is to develop an algorithm by combining encryption and hashing method that provide ultimate security to the user credential. We expect to develop a simple and platform independent algorithm to prevent any unauthorized access of secured user credential (e. g. password).

B. Research Methodology

To achieve our goal, it essentials to create a different way of encapsulating user credential in such a way

where we will use 2 layer of encryption and finally hashing the encrypted data for optimal security of the credentials with performance issue in mind.

The results of the encryption algorithms and hashing are compared and displayed together with a prototype of the web application. The results are then analyzed in two different sections: security of the encryptions and performance tests. With the codes given we conclude which one of the techniques is the most suitable for our web application, and otherwise when encrypting data through the URL of a web application.

For the development of the algorithm, we will follow the procedure mentioned in the flowchart shown in Fig. 1. According to the flowchart first we will take the user password as a plain text, encrypt them and hash the encrypted data before the data is stored into the database.

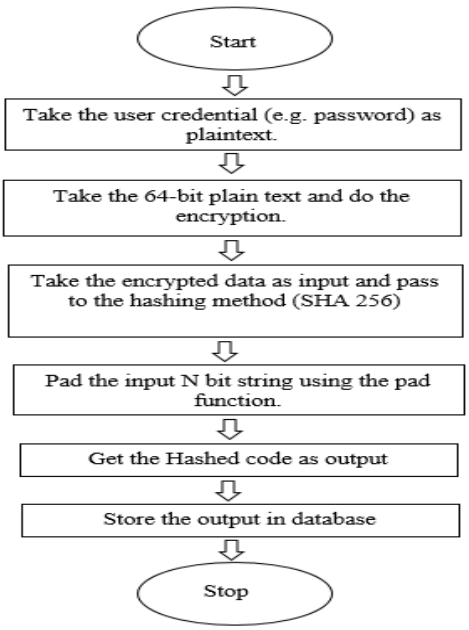

Fig. 1. Flowchart of proposed algorithm.

II. Literature Review

A. Existing works

One-way functions can eliminate the problem of stealing but still there are two major vulnerabilities to kill firstly, overhearing through the medium and secondly, a very simple password. Along with Lamport's proposal of hash chain method in 1981, above problems are eliminated. However, the Lamport's method were limited of high hash overhead and password resetting case. In 1990 and in 1998, A. Shimizu's proposal of validation systems named CINON and PERM for email sending has diminished the high hash overhead issue as well as the password resetting issue. The PERM method has additionally solved the arbitrary number remembrance issue brought by the CINON method. In 2000, Sandirigama and Shimizu have claimed that MITM type of attack is not eliminated by CINON and PERM. Hence, they have proposed a new proposal called SAS (Simple and Secure Password Authentication Protocol) which can be succeed against MITM. Lin, Sun and Hwang in 2001, have demonstrated that SAS is as yet incapable against SV attack, replay attack and DoS attack. In substitution to SAS, they have additionally proposed another method called OSPA (Optimal Strong Password Authentication).

An upgraded OSPA protocol named E-OSPA is proposed in 2003, by Lin, Shen and Hwang, yet has later seen as unprotected against DoS attack and replay attack. A short time later, Chan and Jan proposed ROSI (Robust and Simple Authentication Protocol) protocol yet was later seen as defenseless against SV attack and DoS attack [1]. In 2004, W.C. Ku, HC Tsai and MJ Tsaur has proposed a better secure hash-based password validation method of smart-card less solution [2]. The author has found that the above existing methods are not validated due to the limitations of two unsolved problems. Firstly, if anyone can hack the key then they can cat as a real or authentic user and secondly, the interaction or conversation between two parties are not verified again of lack of security. Hence, W.C. Ku proposed a new method to overcome the above two problems using the timestamp instead of the smart card.

In 2005, Kim and Koc have exhibited SV attack, DoS attack, replay attack and impression attack on W.C. Ku's plan [3]. They have invented their attack by expecting that by certain methods they have the verifier and can hinder the correspondence. The attacker can now effectively produce messages and can confirm to the server utilizing replay attack and client pantomime attack. Later in 2005, SPAPA (Strong Password Authentication Protocol with User Anonymity) convention has been proposed by Mangipudi and Katti [4]. This etiquette is exceptionally straightforward and contains just hash capacities and XOR activities. The creators have asserted that this convention is secure against guessing attacks, SV attack and DoS attack. In any case, later in 2007, Mitchell and Lynn have shown MITM attack, SV attack and replay attack, and demonstrated synchronization issue in SPAPA. Weragama and Sandirigama have demonstrated security shortcomings of SPAPA method and proposed another

form of SAS convention named SAS-3 [5]. A short time later, Tsai, Lee and Hwang and IE Liao et al. have characterized objectives to be accomplished and attacks to be opposed by a perfect password validation system [6]. They have made a comparison between former and recent various schemes goals and security desires. Furthermore, in 2009, Sood, Sarje and Singh also have devised such kind of above comparisons [7]. In 2010, M. Kumar have explained that W.C. Ku's method is unsuccessful to provide session key generation part to validate communication and password altering stage to improve user openness.

After that in 2014, Zhuang, Chang and Wang have verified a MITM attack on W.C. Ku's method and proposed a smart-card less geometric hashing-based method [8]. In 2017, combine encryption method is introduced to protect password [9]. Recently in 2019 several researchers in [10]-[12] have proposed different methods with hash code claimed that it can resists replay, SV and DoS attack but they have used the complex smart-card policy.

B. SHA(Hashing)

A one-way cryptographic method having a place with cryptographic hash capacities called the Secure Hash Algorithms (SHA) was introduced by NIST. It was presented as a U.S. Government Information Processing Standard (FIPS).

All SHA-family algorithms have approval for security functions. A Cryptographic Module Validation Program is jointly run by NIST, the USA and the CSE, Canada.

C. Research Methodology

After 1990, the speed of exhaustive key searches increased. This was used against DES. That was a reason of discomfort amongst DES users.

A combination of 3 different keys namely K_1, K_2 and K_3 are first generated under key K, so the 3TDES key is 168 bits in length with each 56 bits, as shown in Fig. 2.

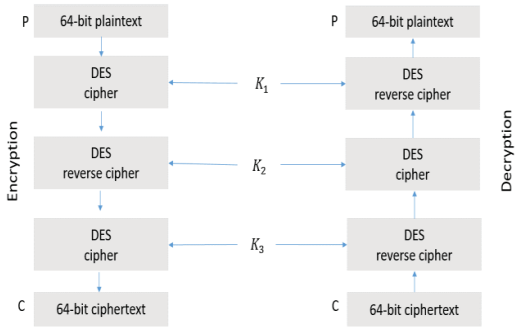

Fig. 2. Triple DES Mechanism.

Sequentially, the encryption-decryption procedure is as follows:

- Firstly, the K_1 key encrypt the plain text using single DES,
- Secondly, the K_2 key decrypt the output of first step,
- Thirdly, the K_3 key encrypt the following output using single DES,
- Then the output of the previous is the cipher text.

D. AES

AES is at least 6 times faster than triple DES. It is a symmetric block cipher. Depending on the key's length, AES has different number of rounds. It uses 10, 12, and 14 rounds for 128-bit keys, 192-bit keys, and 256-bit keys respectively. The features of AES are:

- Both key and block types are symmetric in this method
- Data size is 128-bit, size of keys - 128/192/256-bit
- More secure and faster than Triple-DES
- Full specifications and design details are provided
- Can be implemented in C and Java

This method is based on the substitution-permutation network.

E. Brute Force Attack

In this method, the attacker uses several passwords or passphrases, being hopeful with the fact that, ultimately predicting correctly. The attacker tries all possible passwords and passphrases until they find the correct one. The attacker might use a key derivation function to produce passwords. This is called exhaustive key search.

Right now, assailant submits numerous passwords or passphrases, being cheerful with the way that, in the end

speculating accurately. The assailant attempts every single imaginable secret phrase and passphrases until they locate the right one. The assailant may utilize a key deduction capacity to produce passwords. This is called comprehensive key hunt.

A calculation is made for every possible combination which could make up the password and checking every time a new combination is created. By increasing the length of the password, the time required to find the correct password increase exponentially.

F. Popular Hashing Technique RSA

The cryptosystem, in which the encryption key is public but the decryption key is private, is called a public-key cryptosystem. RSA is one of these kinds of cryptosystems. This system is widely used for protecting data while transmitting. In RSA, this asymmetry depends on the "factoring problem". That is the complexity when two large prime numbers are multiplied and then factorized. The acronym RSA comes from the surnames of three scientists to publicly describe the algorithm in 1977. Another mathematician named Clifford Cocks developed a similar system in 1973, but that remained classified until 1997.

G.Extended Sparse Linearization

The XSL algorithm is a problem-solving method. This method is used for solving polynomial equations with multiple variables, based on linearization method. This method was proposed in 2002. Its purpose was to exploit block ciphers like AES and Serpent. After it was proposed, algebraic attacks against AES came into broader consideration. At that time, it attracted the cryptographic community.

III. Analysis of the Data

A. Analysis Procedure

(1) **Development of New Method**

A new technique is introduced in Fig. 3 which can protect the user's credentials in an internet browser. The proposed technique aims a limited change to client experiences and internet browser.

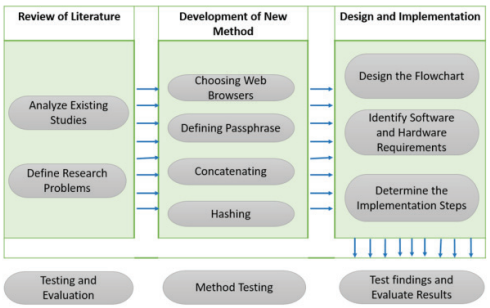

Fig. 3. The methodology model.

a. **Choosing Web Browser**: It is difficult to choose a suitable internet browser to test system execution. In this research work we chose the Google Chrome to execute the proposed method. Greenberg (2013) stated that the Chrome browser kept their client's accreditations in plaintext. Therefore, the program module was intended to be completed in Google Chrome to watch the viability of the program expansion.

b. **Defining Passphrase**: Passphrase provide more security than usual passwords because of its length. Passwords are intended to be short and straightforward with the goal that clients can easily recall them. However, passphrases don't have the constrained extent of passwords and are commonly longer as an additional safety effort. There are a couple of favorable circumstances of utilizing passphrases during client confirmation. First, passphrase is moderately simpler for clients to recall than passwords. Next, passphrases are harder to break than passwords. Password is known to be powerless against breaking. The hackers always use latest or upgraded equipment to break the difficult password. There are also a lot of clients who use jargon words from a dictionary. This prompts hackers to introduce dictionary attacks which can break the password shortly. Subsequently, passphrase that is relatively longer have a possibility of surviving brute-force attack.

c. **Concatenation**: String link is a procedure of combining at least two strings. The significant strings are connected with no spaces in the middle of them. In this work clients are required to enter their own secret key in the wake of approving a particular passphrase. At that point, the entered secret phrase was converged with a client characterized passphrase and website domain name. Moreover, string connections are acted to make longer strings for the hashing procedure. This is on the grounds that more extended strings offer better safety efforts.

d. **Hashing**: Hashing is an irreversible method for scrambling information utilizing an algorithm. During the way toward hashing, most information is eliminated, and a moderately few hash value is generated compared with the first information. Hashing is unidirectional process, so it is called an irreversible function. Therefore, it is problematic to determine hash values back to their underlying sources of information. In this study, we chose SHA256 as the hashing function. When the message input is any character length under 264 bits, the SHA256 method figures a yield known as a message digest that is 256 bits in length.

(2) **Design and Implementation**

The development stage is trailed by the design stage, where each plan is outlined. From here, the dissected information is changed over into a stream graph to draw how the program augmentation acts in confirming client accreditations. By expecting clients to enroll a record for a specific website page, the proposed program expansion begins when the clients enter their username into the module. The username is utilized to recognize the presence of a passphrase in the nearby database.

On the off-chance that the augmentation database does not have an important passphrase, the clients need to characterize a passphrase inside a three-word limit. However, if a client passphrase as of now exists in the nearby database, the augmentation shows the passphrase with a segment of it blanked out. From that point, clients need to embed the concealed piece of the passphrase to verify their personality. Before long, the augmentation needs to get the area name of the website page.

The space name is the present site being seen by clients, for example, Tumblr or Twitter. After acquiring the space name, it is put away with the passphrase in the nearby database. On the client's side, they continue to embed their secret key in the internet browser. Upon passage, the augmentation connects the secret phrase, passphrase, and space name into a string. Afterward, the applicable string is hashed with the SHA256 algorithm and changed over into an adequate secret phrase group for sites.

Fig. 2 shows how the program module functions when the site is visited by clients. Assuming that it is the first-run through login for the clients, the program augmentation will convey the yield of the hash an incentive to be put away as secret key in database. If the clients have signed in different occasions previously, the processed string is contrasted and is put away in the program database.

If both hashed strings coordinate, at that point the client will have the option to login effectively. Expecting that the assailants figured out how to infiltrate the program database, it would be difficult to break the acquired passwords.

(3) **Testing and Evaluation:**

In this stage, testing was performed to affirm whether the proposed program augmentation satisfies the requirements. Thusly, every capacity in the created expansion was tried. Testing was completed to survey the program augmentation as far as fathoming the constraints that were earlier expressed. The before composed restrictions were that passphrases put away in the nearby database were not hashed and that there was no word limit for the client's passphrase. Wherefore, the program expansion should be tried to analyze how it reacts to these two impediments. To test the proposed strategy, a site was worked with login and enlistment page, illustrative of present-day sites. This was to test the exhibition of program augmentation with no inclusion from a web server. The showed site applies challenge-reaction verification, which interfaces with the MySQL Server.

B. Proposed Method for Securing User Credential

For securing the user data, we first took the user data as a plain text, then with the help of the DES algorithm we encrypted the plain text using the public key. This was followed two times and after that we passed the encrypted data to the hashing method. For hashing purpose, we used SHA 256 and finally stored it into the database as shown in Fig. 4.

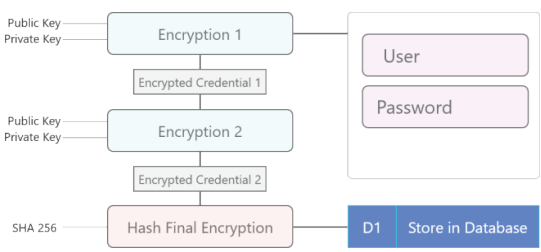

Fig. 4. Process of algorithm

C. Proposed Algorithm

Step 1: Take the user credential as plaintext.
Step 2: Go to step 5.
Step 3: Take the encrypted data as input for passing it to the hashing method (SHA 256). Go to step 9.
Step 4: Take the result as output and Exit.

Step 5: Take the 64-bit plain text and handed over to initial permutation function.
Step 6: Initial permutation produces two halves of the permuted block; says Left plain Text and Right Plain Text.
Step 7: Each LPT and RPT are rejoined and a Final Permutation is performed on the combined block.
Step 8: Return the result to step 3.
Step 9: Take the input as a bit string N.
Step 10: Pad the input N using the pad function.
Step 11: break P into n consecutive r bit pieces.
Step 12: Absorb the input into state by extending with a string of Zero bits, XOR and apply permutation.
Step 13: Initialize the variable Z with S.
Step 14: Truncate Z to d bits.
Step 15: Return to Step 4.

IV. Results and Discussion

In this paper we have developed an algorithm for securing the user credentials (e.g. password). Our proposed algorithm took the user password as a plain text and it was encrypted into a plain text depending on a specific public key using the DES encryption method and the result of this was again encrypted using the DES encryption method.

An example is illustrated in Fig. 5 with the password – 'accesskey'.

Fig. 5. Implementation of the algorithm

At the first encryption, the plain text converts into:
"fviExxyZ4UXUOh4nF1OvmQ==".

Then at the second encryption it converts to:
"H5i21RVPzWP70H0LbFb3aQGZAqsRDFcYwk18GyA/KzYJbTZMXxPJ/A==".

At the final stage when we pass the output of the second encrypt for hashing it converts to:
"04E87763FCE1B493F060673DDA89371AF5609056025A15BE551092BE417D6694".

Here, after the final stage a simple text (password) 'accesskey' is converted into a form which is difficult to decrypt and it will be successfully sustained against the dictionary attack. The execution time is 4.860002547502518ms, which may vary with the length of the password is used.

Based on given information in Table I the encryption data length is 24 digits which is less protected than 64 digits to hacker to break the password. Whereas SHA256 and our proposed method have same 64 digits message digest, but our proposed method is more protected than SHA256. This is because the hashing technology works in a way that, this is a 'ONE WAY PROCESS'. That means hashes are encrypted and made non retrievable.

If two different files produce the same unique hash value this is called a collision and it makes the algorithm essentially useless. So, we have used both the encryption and the hash combined to save our password in more secure way. Even if some technology arrives that can decode the hashes back, the passwords will still be protected with a custom key given by the server owners.

The storage required in this case will depend on the type of the hashing technology such as md5, SHA1 and SHA2 are used. For example, if a hashing technology requires 40 letters to be represented, it will always require 40 letters. So, the storage required is always the same as it was before.

TABLE I
COMPARISON OF PASSWORD CRYPTOGRAPHIC ALGORITHMS

Data (e.g. password)	accesskey
Encryption Value	FVIEXXYZ4UXUOH4NF1OVMQ== TOTAL LENGTH 24 DIGITS
Hashing value (SHA256)	7ad702bdafa630cc901cc6165536c25d083b a8ef56236db136340a9940231964 TOTAL LENGTH 64 DIGITS
Combining Encryption and Hashing value	04E87763FCE1B493F060673DDA89371AF5 609056025A15BE551092BE417D6694 TOTAL LENGTH 64 DIGITS

V. Conclusion

This paper gives an overview of a new Cryptographic Hash algorithm for the security of user credentials (precisely the user passwords) using both the encryption and the hashing method. The benefit of having both the encryption and the hashing method is that it is quite

impossible to decrypt the hashed output. The encrypting the password and then hashing before taking this to the database is, even if the database hashes are leaked, hacked or accessed by an attacker, they will not be able to decrypt the password then and there.

Because we have encrypted the hashes beforehand, with our own keys. So this process makes it difficult for the attacker to gain access to the system. Therefore, by using the proposed method we can avoid the existing only encryption and only hashing method as they are compromised and not secure for user credentials.

References

[1] Chi-Kwong Chan and L.M. Cheng, "Cryptanalysis of a Timestamp-Based Password Authentication Scheme", *Computers & Security, Volume 21*, Issue 1, *pp. 74-76, 1st Quarter 2001.*

[2] W.C. Ku, HC Tsai and MJ Tsaur, "Stolen Verifier attack on an efficient smartcard-base one-time-password authentication scheme", *IEICE TRANSACTIONS on Communications, Vol.E87-B No.8 pp. 2374-237, 2004.*

[3] C.K. Koc, M. K. "A simple attack on recently introduced hash based strong password authentication scheme", *International Journal of Network Security, Vol.1, No.2, pp.77–80,* 2005.

[4] Katti, and K. M. "A hash based strong password authentication protocol with user anonymity", *International journal of Network Security, Vol.2, No.3, pp. 205–209, 2005.*

[5] W.C.Ku, H. C., "Two simple attacks on LSH's strong password authentication protocol", *ACM SIGPOS Operating Systems Review, Volume 37*, Issue 4*, pp. 26-31, October 2003.*

[6] IE Liao, CC Lee and MS Hwang, "A password authentication scheme over insecure networks", *Journal of computer and system science, pp. 727-740, 2006.*

[7] S. K. Sood, A. K. Sarje and K. Singh, "Cryptanalysis of password authentication schemes: Current status and key issues", *Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS), Delhi, pp. 1-7, 2009.*

[8] X Zhuang, C. C., "A simple password authentication scheme based on geometric hashing function", *International Journal of Network Security, pp. 237-243, May, 2014.*

[9] S. Eman and I. Harba, "Secure Data Encryption Through a Combination of AES, RSA and HMAC", *Engineering, Technology & Applied Science Research, Vol.7, No.4, pp. 1781-1785,* 2017.

[10] M. H. Ali, E. S. Ismail and F. M. Hamzah, "A Practical and Secure Hash Function-Based Password Authentication Scheme", *Journal of Computer Science*, *pp. 954-960, July, 2019.*

[11] M. A. Hossain, A. Ullah, N. I. Khan and M. F. Alam, "Design and Development of a Novel Symmetric Algorithm for Enhancing Data Security in Cloud Computing", *Journal of Information Security, vol. 10, pp 199-236. October 14, 2019.*

[12] F. E. De Guzman, B. D. Gerardo and R. P. Medina, "Implementation of Enhanced Secure Hash Algorithm Towards a Secured Web Portal," *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, *pp. 189-192, 2019.*