ISBN 978-981-11-7861-0
Proceedings of 2018 the 8th International Workshop on Computer Science and Engineering (WCSE 2018)
Bangkok, 28-30 June, 2018, pp. 597-604
doi: 10.18178/wcse.2018.06.100

Securing a Client-Server Communication through Login Authentication Using Modified MD5

Esmael V. Maliberan[1+], Ariel M. Sison[1], Ruji P. Medina[1]

[1]Technological Institute of the Philippines, Philippines

Abstract. The study was accomplished to secure a client-server communication through login authentication using modified md5. The modified MD5 which produces 1280-bit hash code length will be used to hash the password entered by the user in a login form and eventually store it in the system's database. Findings affirmed that the modified algorithm was more secured compared to the original MD5 algorithm since it was not cracked using generic attacks such as dictionary, brute force and rainbow table and other cracking tools available online. The performance of the modified algorithm was measured with only 10.9 ms additional execution time from MD5.

Keywords*:* client-server communication, attacks, modified MD5 algorithm, hash algorithm, login authentication

1. Introduction

Protecting online commercial transactions and online client-server communication have always been a challenge for both IT professionals and online users in the past years. Sensitive Information that travels and sends across the internet must at all times be protected from all types of hacking. Information Security is an utmost concern in the domain of Information Technology [1]-[3]. In a client-server communication, login authentication plays a vital role to maintain data with extreme integrity. Passwords must be protected at all times by applying secured hash algorithms and encryption to ensure the safety of the entire system. Once the login authentication is compromised, an intruder can access the systems database, modify other user information and function modules which will put the entire system at risk [4].

Over the years, many security models and algorithms have been used in cloud environment, but most of them focus on a particular security threat rather than catering to the entire system [5]. With the development of internet, a lot of communication applications, such as electronic mail or the uses of World Wide Web browsers are receiving information [6]. When packets travel across a network, they are susceptible to being read, altered, or "hijacked." Hijacking occurs when a hostile party intercepts a network traffic session and poses as one of the session endpoints. A hacker examines streams of data to gather sensitive information. This attack usually involves sniffing network traffic but may include observing other types of data streams.

While there is a huge amount of transferring data in a cloud system, the risk of accessing data by attackers raises [7]. Because of these, there have been a lot of approaches and techniques in securing the client-server communication over the internet. During the past, the MD5 algorithm was one of the methods widely used to hash the password. However, it was broken using brute force attacks and rainbow tables. Several modifications and enhancement in MD5 algorithm had been developed to address its security issues [8]-[11]. [12] Proposed a new modified 256-bit MD5 algorithm with SHA compression function. The paper is a combination of some functions to reinforce these functions and also increasing hash code length up to 256 bit

+ Corresponding author.
E-mail address: malibs_28@yahoo.com

that makes it stronger algorithm against collision attacks. According to [13], to provide higher security protection, MD5-512 bit algorithm has been proposed. Although there were many enhancements being made to MD5 algorithm; however, these modifications remain vulnerable to attacks. Hence this study is directed towards the development of a modified 1280 bit MD5 algorithm which will be used in hashing a password for login authentication.

2. Review of Related Literature

Hash algorithms are significant components in many cryptographic applications and security protocol suites. The MD5 Message-Digest algorithm is a widely used cryptographic hash function that generates a 128-bit hash value. Developed in 1991, MD5 has been used in all sorts of systems and is an integral part of many standards for communications security. MD stands for "Message Digest," and MD5 was the fifth such algorithm that Rivest had designed. MD5 is the fastest hashing algorithm included in the .NET Framework, but the relatively small hash code size makes it more susceptible to brute force and birthday attacks.

Table 1: Summary of different hashing algorithms

Name	Input block size	Message limit (bits)	Hash code size (bits)
MD5	512	2^{64}	128
SHA-1	512	2^{64}	160
SHA-256	512	2^{64}	256
SHA-384	1024	2^{128}	384
SHA-512	1024	2^{128}	512

There have been a lot of researches conducted in using the MD5 algorithm to secure client-server communication particularly in cloud environment. Some of these papers have been proposed to discover the flaws of different hash algorithms particularly MD5 [14]-[17]. Thus, several types of research have been proposed to enhance and modify MD5 algorithm to increase its security level and performance.

[4] Proposed an improved MD5 that was used for login authentication in web applications. The approach was to create and combine digits or numbers and English characters from the original password created by the user to form a new password. The newly generated password then is hashed by the improved MD5 which is eventually stored in the database. Findings revealed that the security level was improved because the password stored in the database is no longer the original password created by the user. But, the disadvantage of the study is when the user forgets his password. In this case, he cannot employ the habitual way to change the password, only the administrator will be able to reset the password.

[18] Proposed the mixed encryption algorithm which is from MD5 and “XOR transformation” to further solve security concerns. The hash value produced form MD5 algorithm can be stored in a data file publicly. When a legitimate user login, the system will then evaluate the hash value of the key which is generated from the MD5 algorithm with the hash value stored in the database. If both the hash values are the same, the system will make changes on the key which is inputted by user constantly and carry as the key to encrypt perceptive information when it is long enough. Eventually, this key will carries on XOR changes with the perceptive information which is waiting for encrypt decipher to complete the entire process of encryption. However, MD5 is non-symmetrical system algorithm and has irreversibility and high security. Therefore, the author cannot expect to find out the counter algorithm of MD5 in good time to decode data.

[12] Proposed a new modified MD5 Algorithm merged with SHA Compression Function that generates an output of 256-bit hash size. This enhancement was used to prevent birthday attacks, rainbow table and brute force attacks. This approach was to extend the hash size up to 256 –bit from its original size of 128-bit. This is done by expanding the compression function block size which will be eventually used in any signing applications and data reliability. Complex data improvement techniques were used to achieve the necessary situations on the chaining variables and the message bits. Findings revealed that it is resistant to local collision and differential attacks. Nevertheless, the new approach was based on Double-Davies-Meyer that satisfied

Merkle-Damgard condition. This means that it cannot be secure as a random oracle no matter how good its compression function is.

According to [19], their study improves the hash value of MD5 algorithm from the original 128 bit to 160 bit of hash output size. The MD 160 bit is being used to authenticate data integrity. The weakness sighted in the study is the 160 bit is not sufficient enough to prevent a collision attack, brute force and rainbow table due to the existence of online cracker that can hash the message up to 256 bit.

[20] Proposed a unified architecture that can perform two hash functions using the MD5 and MD5-512 bit for the password to prevent different generic attacks. The authors improved the MD5 by expanding its size up to 512 bit and combined it with the original MD5 to make it a more secured hash algorithm. These two algorithms have similarities, but they differ in their speed and security level. The main purpose is to allow applications to make a selection out of the two algorithms based on different necessity. Although the proposed architecture had expanded its size, the way the algorithm was hashed used only simple operation such as AND and OR which is still vulnerable to attacks.

According to [21], user authentication is one of the important ways in ensuring the security of various e-governments, e-commerce, and other web applications. The modified MD5 512 bit algorithm can be an effective way to be used in such client – server communications. The proposed design used the combination of OR and AND operation to hash or digest the message to ensure the message integrity or signing application. The new approach can generate an output of 512-bit size. Although the approach had improved the security level, still, it followed the basic principle of how MD5 algorithm digest the data hence it is still vulnerable to attacks because the method used in hashing has already been learned by the hackers.

2.1. Securing a Client-Server Communication through Login Authentication Using the Modified MD5

To secure the login authentication of a client-server communication using the modified MD5, users are advised to create a password that is not easy to guess. So in the login form, the only password with a combination of characters and numbers with a capital letter will be accepted. These will prevent the attackers from surmising the right password. Then the script in the server will hash the password entered by the user into 1280 bit and compare the hash value to the stored hash value (1280 bit) of the password created by the user during registration. If both the hash values are not equal, then the server will not grant the request from the user. This can be represented by a diagram as shown below.

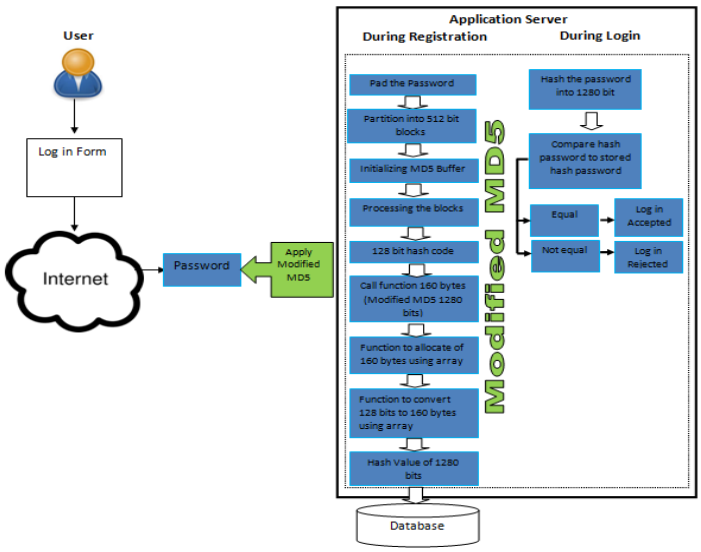

Fig. 1: Securing a client-server communication using modified MD5.

2.2. Expanding the Hash Size of MD5

The output of the MD5 algorithm which is 128-bit will be converted into hexadecimal value comprising of 32 characters. A function will be called to convert this value into decimal and allocate 16 blocks of an array comprising of 1 byte per block. There will be two (2) characters assigned and allocated for every block.

Another function is created to allocate 160 blocks of an array of size 1 Byte. This is intended for expanding the hash length to 160 bytes or 1280 bit as its target output. During this phase, the message will be hashed using XOR and AND operators and store the hashed message per 16 blocks of an array. Processing will be up to ten (10) rounds and 16 operations per round. Another function will be called to convert decimal back to hexadecimal and finally outputs the 320 hexadecimal hash value or 1280 bit hash value.

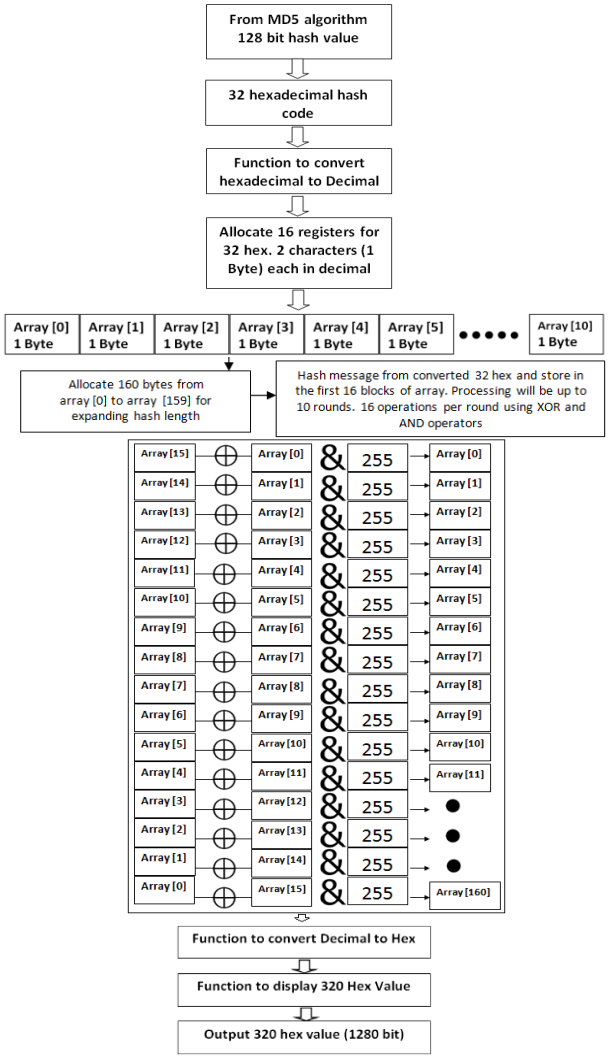

Fig. 2: Expanding the hash value of MD5 into 1280-bit.

3. Testing the Reliability of the Modified MD5

All generic attacks must be tested to measure the performance and reliability. To test the credibility of the modified hash algorithm, several attacks and tools have been used and tried out.

CrackStation is one of the powerful online cracking tools. It utilizes huge pre-computed lookup tables to break password hashes. These tables store a plot between the password hash and the exact password for that particular hash. The hash codes are listed and indexed for searching the database immediately for a given hash. If the hash is found, the password can be obtained in a split of a second. This is applied only for hashes without salt and supports current hash algorithms such as MD4, MD5, sha256, sha512, etc.

As shown in Figure 3, the modified MD5 algorithm with an output of 1280 bit hash value was tested using the tool. The 1280 hash value will be searched in the databases of existing hash algorithms. The tool is configured to find a possible match. Though, the CrackStation accepted the hash code as input, but, it could not generate the equivalent plaintext value of the hash code in its database. As a result, is the unrecognized hash format. It can be noted that the color code is red which means that the hash value is the unrecognized hash format and its type is unknown or not found in the database. Since the tool will work only on the existing known hashing algorithm such as MD5, sha256, sha512, MD160, etc.

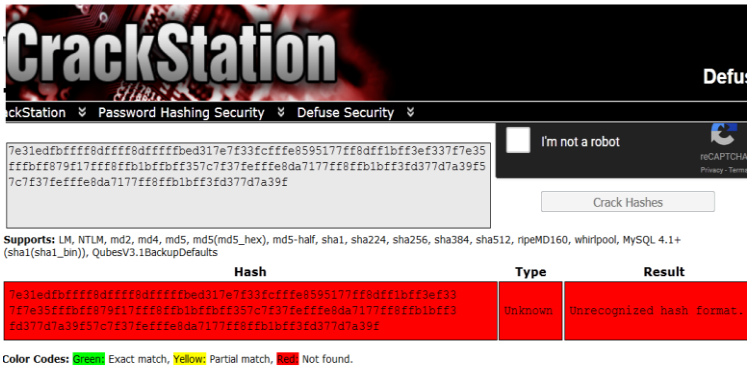

Fig. 3: Cracking the modified MD5 using CrackStation.

3.1. Testing a Dictionary Attack

A dictionary attack is another form of attack in hashing algorithm. It tries to conquer an authentication mechanism by analytically using every word in a *dictionary* as the password. Hash Kracker is a simple-to-use software program that can calculate and reveal hash passwords from hash text using several algorithms. It offers support for MD5, SHA1, SHA256, SHA384, and SHA512. The tool used a dictionary named passlist.txt as shown in figure 4 to look up for possible passwords that will match the hash code.

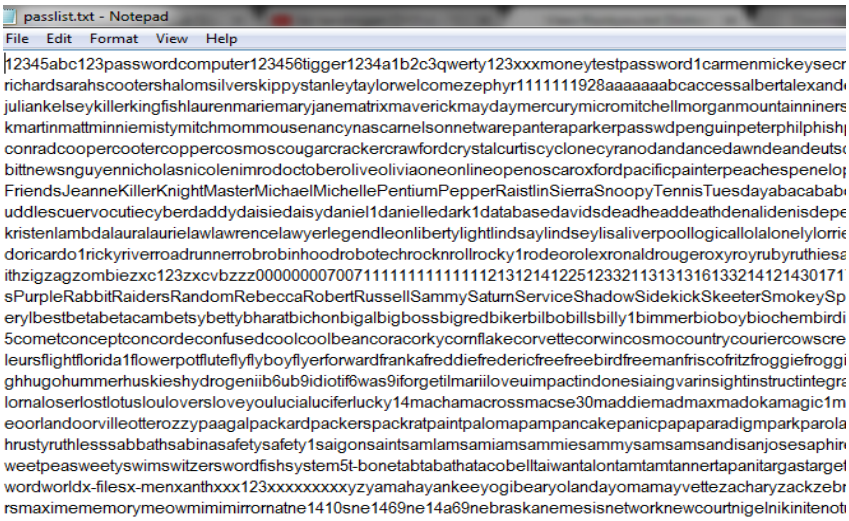

Fig. 4: A dictionary that contains possible passwords.

The tool accepted the 1280 bit hash value of the modified MD5, but a message appears which says “Input hash text and type do not match”. Make sure you have entered correct HASH Text and Type. It is shown in Figure 5.

Fig. 5: Using Hash Kracker tool to perform a dictionary attack on the Modified MD5.

Again the test of Dictionary attack on the modified MD5 failed because it only look ups to the tables of existing hash algorithms such as SHA1, SHA256, SHA512, MD5, MD160, etc.

3.2. Testing a Brute Force Attack

Another common attack in a hash algorithm is brute force attack. A brute force attack is guessing and a trial-and-error way to acquire information such as personal identification number (PIN) or a password. In this type of attack, a system is used to produce a huge number of successive guesses regarding the content of the preferred information. The researchers used the hashkracker console program to perform this attack to measure its reliability. Hash Kracker Console is the all-in-one command-line tool to find out the password from the Hash. Currently, it supports password recovery from following popular Hash types such as MD5, SHA1, SHA256, SHA384, and SHA512.

As shown in figure 6, this attack failed in the modified MD5 because it cannot recognize the type of the hash value of the modified MD5. It is impossible to brute force on a certain kind of hash value if the attacker does not know its source code or how the modified MD5 was being developed. Furthermore, the attacker must first be familiar with the source code of the modified MD5 before he can perform brute force attack.

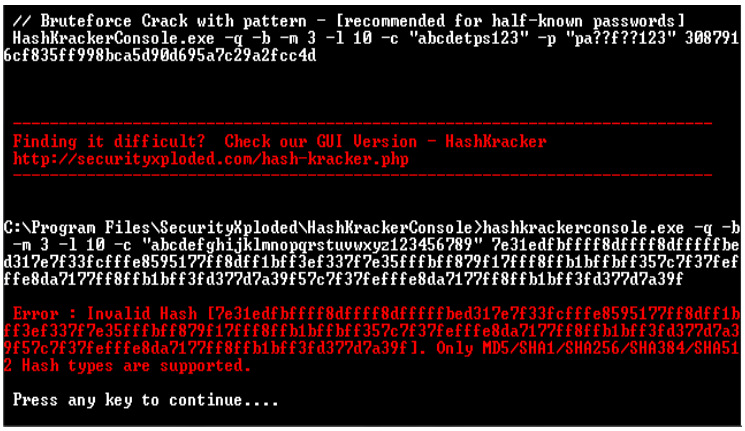

Fig. 6: Using HashKracker Console to perform brute force attack on the Modified MD5.

Another tool was tested to test the credibility of the Modified MD5 using Cain and Abel password cracking tool. This tool is a password recovery tool for Microsoft Windows. Indeed, it can recover many kinds of passwords using methods such as network packet sniffing, cracking various password hashes by using techniques such as dictionary attacks, brute force and cryptanalysis attacks as shown in figure 7. When the word “MYPASS” was converted into its equivalent hash value, findings revealed that it produces a hash value of “FAE563F7FBA59F68C0029ED873A1E54C” which is different from the hash value of the modified MD5.

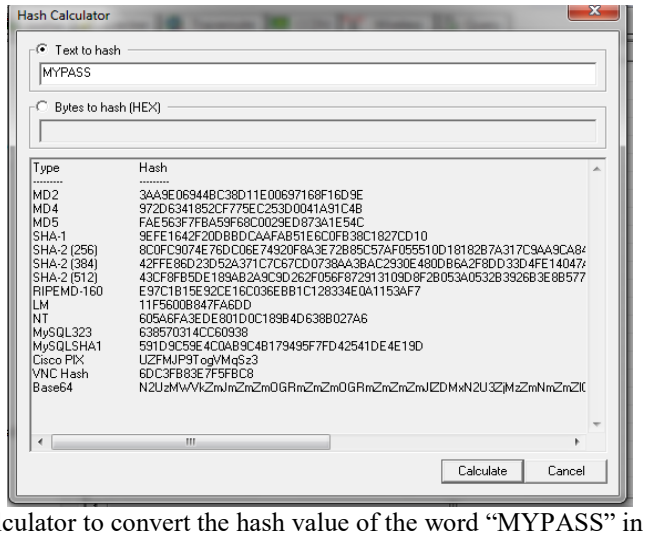

Fig. 7: Using Hash Calculator to convert the hash value of the word “MYPASS” in other hash algorithms

3.3. Testing a Rainbow Table Attack

Rainbow Table attack utilizes a pre-computed rainbow table. It consists of a database that has a large number of a hash function’s input and equivalent output. Its function is just to search and compare a password and its equivalent hash value inside the table. As shown in figure 8, a rainbow table attack is tested using

RainbowCrack software. Results showed that the attack failed to crack the hash value of "MYPASS" which is 1280-bit generated from the modified MD5. This is because the database only contains hashes of the existing hash algorithms such as MD5, SHA2, SHA256, SHA512, etc.

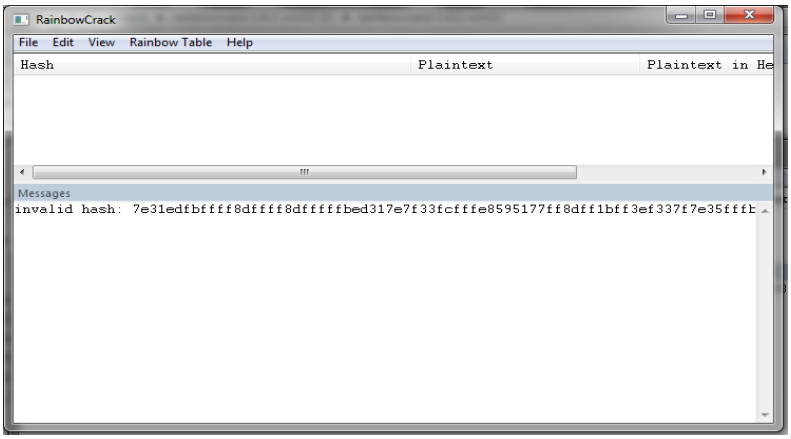

Fig. 8: Testing for rainbow table attack using Rainbow Crack

4. Conclusion

The authors proposed a new design that will expand the hash size of MD5 algorithm to secure a client-server communication through a login authentication in a web-based application. The new approach applied the combination of XOR and AND operators using PHP and C Language to the original 128-bit hash value of the MD5 algorithm to generate a 1280 bit hash size. Results have shown that the performance of the modified 1280-bit MD5 was measured 10ms additional execution time from MD5 and it furthers prevent from generic attacks such as brute force, dictionary and rainbow table during testing, thus improved its security level. For future research on this area, a variable output hash size can be used as an option in hashing the password in order to manage and maximize the execution time to further improve its performance. Likewise, it is also potential to extend the size of compression function block of the MD5 in order to improve its security level.

5. References

[1] E. Sediyono, K. Santoso and Suhartono, "Secure Login by Using One-time Password Authentication Based on MD5 Hash Encrypted SMS", *IEEE,* 2013.

[2] V. Mahalle and A. Shahade, "Enhancing the Data Security in Cloud by Implementing Hybrid (Rsa & Aes) Encryption Algorithm", *IEEE,* 2014.

[3] Q. Kester, L. Nana, A. C. Pascu and S. Gire, "A New Encyption Cipher for Securing Digital Images of Video Surveillance Devices using Diffie-Hellman-MD5 Algorithm and RGB shuffling", *European Modelling Symposium IEEE,* 2013.

[4] L. Zhong, W. Wan and D. Kong, "JAVAWEB LOGIN AUTHENTICATION BASED ON IMPROVED MD5 ALGORITHM", *IEEE,* 2016.

[5] A. Arora, A. Rastogi, A. Khanna and A. Agarwal, "Cloud Security Ecosystem for Data Security and Privacy", *7th International Conference on Cloud Computing, Data Science& Engineering – Confluence,* 2017.

[6] H. Kumar, S. Kumar, R. Joseph, D. Kumar, S. Singh, A. Kumar and P. Kumar, "Rainbow Table to crack Password using MD5 Hashing Algorithm", *Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT2013.),* 2013.

[7] N. Khanezaei and Z.M. Hanapi, "A Framework Based on RSA and AES Encryption Algorithms for Cloud Computing Services", *IEEE Conference on Systems, Process and Control (ICSPC 2014).* 2014

[8] Mehrabani, & Eshghi, "Design of an ASIP Processor for MD5 Hash Algorithm ", *20th Telecommunications Forum TELFOR*. 2012

[9] Vidhya and Sasilatha, "Performance analysis of Black Hole attack Detection scheme using MD5 algorithm in WSN", *International Conference on Smart Structures & Systems,* 2014.

[10] Y. Sasaki, "Improved Single-Key Distinguisher on HMAC-MD5 and Key Recovery Attacks on Sandwich-MAC-MD5 and MD5-MAC", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* pp. 26-38, 2015.

[11] A. Bhandari, "Enhancement of MD5 Algorithm for Secured Web Development", *Journal of Software,* vol.12, no. 4, pp. 240-252, 2017.

[12] A. Kasgar, J. Agrawal and S. Sahu, "New Modified 256-bit MD5 Algorithm with SHA Compression Function", *International Journal of Computer Applications,* vol. 42, no. 12, 2012.

[13] A. Pandey and P. Bonde, "A Modified Approach For Cryptograpic Hash Function Based On MD5 Algorithm", *International Journal of Engineering Research & Technology (IJERT),*vol. 2, no. 8, 2013.

[14] M. Stevens, A. Lenstra and B. Weger, "Chosen-prefix collisions for MD5 and applications", *Int. J. Applied Cryptography*, vol. 2, no. 4, 2012

[15] X. Wang and H. Yu, "How to Break MD5 and Other Hash Functions"

[16] J. Liang and X. Lai, "Improved Collision Attack on Hash Function MD5".

[17] O. Mikle, "Practical Attacks on Digital Signatures Using MD5 Message Digest". 2004

[18] X. Nan-bin and H. Xiang-dan, "The Mixed Encrption Algorithm Based on MD5 and XOR Transformation", *Second International Workshop on Education Technology and Computer Science,* 2010.

[19] P. Sallam, J. Agrawal and S. Sahu, "A New Approach 160-bit Message Digest Algorithm", *International Journal of Computer Applications*, vol. 38, no. 5, pp.22-26, 2012.

[20] V. Mishra and V. Pandey, "Architecture based on MD5 and MD5-512 Bit Applications. *International Journal of Computer Applications",* vol.74, no.9, 2013.

[21] P. Walia and V. Thapar, "Implementation of New Modified MD5-512 bit Algorithm for Cryptography", *International Journal of Innovative Research in Advanced Engineering (IJIRAE),* vol. 1, no. 6, 2014.