Xtings -160: A Strong Diffusion Property Novel Hashing Algorithm

Christine Charmaine G. San Jose

Abstract: This paper proposed for a novel hash algorithm called XTINGS - 160 Hash. Existing hash algorithm that is commonly used such as MD5 and SHA-1 are no longer safe to use and is vulnerable to brute force attacks. The proposed novel hash algorithm produces a 160-bit message digest having six rounds of calculation on the right half 64 bit of plaintext. The paper also illustrates how the final design of algorithm was achieved. The first two designs produce an 80 bit message digest and had contributed into attaining the XTINGS Hash algorithm which is the final design. Findings revealed that the Novel Hash Algorithm was able to meet the avalanche effect minimum expected rate of 50% in just six rounds. The novel hash algorithm is showing significant result that displays the characteristic of cryptographic property of diffusion with a robust hash function.

Index Terms: Avalanche Effect, Cryptography, Diffusion, Hash Algorithm, Message Digest, Padding, Parsing .

1 INTRODUCTION

From the advent of internet and e-commerce usage, reports on breaching of security had continuously increased. Identity theft is a crime from which attackers may obtain important information of someone such as social security, driver's license and others that eventually lead to impersonate and gain access to someone's account. Statistics shows that there are about an increase of 12% cases for Identity theft in US in 2009 and an increase of 12.5% on annual fraud amounting to $54 Billion [1]. At present, username and password are commonly used by many as a log-in process on protected application such as logging into accounts, accessing applications web sites and many others. In information security, password is a critical component; it is for this reason that a password standard was created. This standard identifies the characteristics of strong and weak passwords [2]. Previous studies have shown [3] that most respondents supply password consist of only alphabetic characteristics and password derived from personal details, use of short password and is seldom changed. It was also stated [2] that the following characteristics falls on a weak or poor password: a password that contains eight characters or less, it contains personal information such as birthdays, address, phone numbers or names and others, a password that contains number patterns such as "aaabbb", "qwerty", "zyxwvuts", "123321", "welcome123", "password123" among others. These characteristics of password are weak and are prone from possible attacks. Moreover, the characteristic of a strong password should consist of at least fifteen alphanumeric value, contains both uppercase and lowercase characters, contains numbers and symbols, and is not based on personal information. The standard set in supplying strong password is very challenging because human memory is limited. It was observed that strong password such as "Xa&2#iCj1%s" can be very difficult and affects the password memorability. On the study of [1], it analyses ID-password usage and new log-in vulnerability measures. It was found out that those vulnerability credentials of internet users were explained using cybernetic theory and cognitive psychology theory. In the cognitive psychology theory implies that people may be able to remember a few unique identification and password without difficulty but has a great trouble in remembering them when the number of combinations increases. Moreover, attackers crack password using password-cracking software [4]. For seven characters, the software can try alphanumeric in 5.5 hours, every alphanumeric password with common symbols in 45 hours and every possible keyboard password in 480 hours. The storing of password in plain text or in readable form allows attacker to easily break-in into any account. To solve this issue, a security strategy can be adopted with the use of cryptographic hash function on stored password. Cryptography was defined as the study of mathematical techniques into the aspects of Information Security [5]. Cryptography used hash function as a technology which can be implemented to solve increasing security issues [6]. Hash Function has a one-way property making it difficult for a ciphertext to invert. It involves mathematical calculation that takes any plaintext or message as input, performs several processes such padding, bits calculation (xoring, adding and multiplying) several permutations, concatenation, bits shifting and many others, and produces fixed size alphanumeric value. Hash algorithm is being applied into many applications such as digital signature, message authentication, data integrity and key derivation [6, 7] Hashing algorithms such as MD5 (Message Digest 5) by Ronald Rivest of Massachusetts Institute of Technology (MIT) in 1991 and SHA1 (Secure Hash Algorithm 1) of National Institute of Standard and Technology (NIST) in 1993 are the commonly used cryptographic protocols and Internet communications. In the past years, studies have shown that these hashing algorithm were no longer safe and is vulnerable to brute force attack [5, 8]. The continued modification, improvement and even creation of new cryptographic hash function has been advised by information security professionals that is resilient to the number of attacks [8]. This paper is geared towards proposing a new hash algorithm called XTINGS Hash algorithm, simulate its mathematical calculation, compare and evaluate to famous hash algorithm in terms of cryptographic security property.

2 RELATED LITERATURE

There were several hash algorithms that was developed: Series of Message Digest (MD2, MD4, MD5, MD6), Series of Secure Hash Algorithms (SHA0, SHA1, SHA2, SHA3), Des-Like Message Digest Computation (DMDC), and Keyed-hashing Message Authentication Code (HMAC) [6]. The famous among these hash algorithms are MD5 and SHA1. The MD5 Operation produced a 128 bit which is computed as follows: 1) Padding of message into 152 bit, 2) Initialization of MD Buffer, 3) Calculation of Four Auxiliary Function (F,G,H and I) and 4) Calculation of FF,GG,HH and II. To obtain the value of a, the following calculation is performed: The value of initialized MD buffer for b,c and d is calculated using formula

for function. The value such as M[k], T[i] and F are xored together and will be shifted 5 times to left and will be xored to the value of b. The HMAC is a key-dependent algorithm that has the same properties to one-way hash function. The message digest output depends on the type of hash function to be used. The HMAC calculation is performed by the following: 1) Padding of key to form 512 bit and xored to the value of ipad to obtain Ω1= K'⊕ipad and latter will be xored to the concatenation of Ω1 and message M. The value H is obtained to the concatenation and calculation of IV. The value H is padded such that h=152 bits and finally the value of h will be concatenated to H[K ⊕ opad). The DMDC was created for the purpose of securing mobile communication and it can produce several message digest output. On the other hand, MD5 had adapted MD4 calculation. The mathematical scheme of SHA-1 was based on MD5 calculation [6]. The HMAC is a key-dependent algorithm and it uses either MD5 or SHA-1 to hash input message and will generate a key that will be calculated with permutation, E-bit selection and S-boxes. The four hashing standards: DMDC, MD5, SHA-1 and HMAC, in spite of its unique calculation from each other, it employs common cryptographic calculation such as: padding, XORing, multiplication, parity bit addition, shifting, switching and concatenation. There were several modification conducted to MD5 to address its weaknesses [9, 10, 11]. On the study of [9], it modified the MD5 by employing new technique using the six reserved bits of TCP header to password and applied hashed algorithm before it passed through the internet. This approach reduces the risk of identifying the original password. In the study of [11], they proposed to enhance MD5 by expanding the hash size of MD5 from 128 bits to 1280 bits. The new approach had resulted to a more secure algorithm that is resilient to hashing attacks with 10.9 ms additional execution time from MD5. In [12] proposed to a New Hash Algorithm based on MD5 and SHA-256. The result of security analysis of the study have shown a better performance when compared to SHA-256. The combination of the algorithm applying double-davis-meyer scheme have overcome the weaknesses of the existing functions. Enhancement of SHA1 was conducted by [13, 7]. A modification of SHA-1 on parsing method under pre-processing function of the algorithm was conducted [13]. The result shows an increase to its security performance but the additional operation added a minimal processing time to hash the message. In the study [7], it enhances SHA-1 by extending the message digest into 192 bit size. This enhancement had increased the security and more relevant result. Another property of cryptography [14] that must be considered to a more secure cipher is the confusion and diffusion property. This was identified by Claude Shannon in 1945 in a report "A mathematical Theory in Cryptography". The confusion property [15] means that there is untraceable connection between the plaintext and the ciphertext, while diffusion property refers to a single change of plaintext, will result to a fifty percent changes to its ciphertext. This property could practically be calculated by means of an avalanche effect.

3 STRUCTURE OF THE NEW HASH ALGORITHM

There were several designs in an attempt to propose a new hash algorithm. The XTINGS Hash Algorithm was attained based on prior design that is illustrated and discussed below. The XOX Function is the first design that produces an 80- bit message digest value. This hash algorithm mathematical computation consist of pre-processing phase (padding and parsing of plaintext), constant used and message digest formula. A plaintext will be padded to form 64 bits and later to split the plaintext to form four 16 bits word which will become the value of A,B,C and D. The constant used is an eighty bit consisting of five 16 bit registers (X1,X2,X3,X4 and X5). The proposed algorithm message digest formula based on the figure below is the concatenation of the value (P1,P2,P3,P4 and P5) and the final hash value is the xor value of P1 to P5 and many 1's consisting of 80 bits.

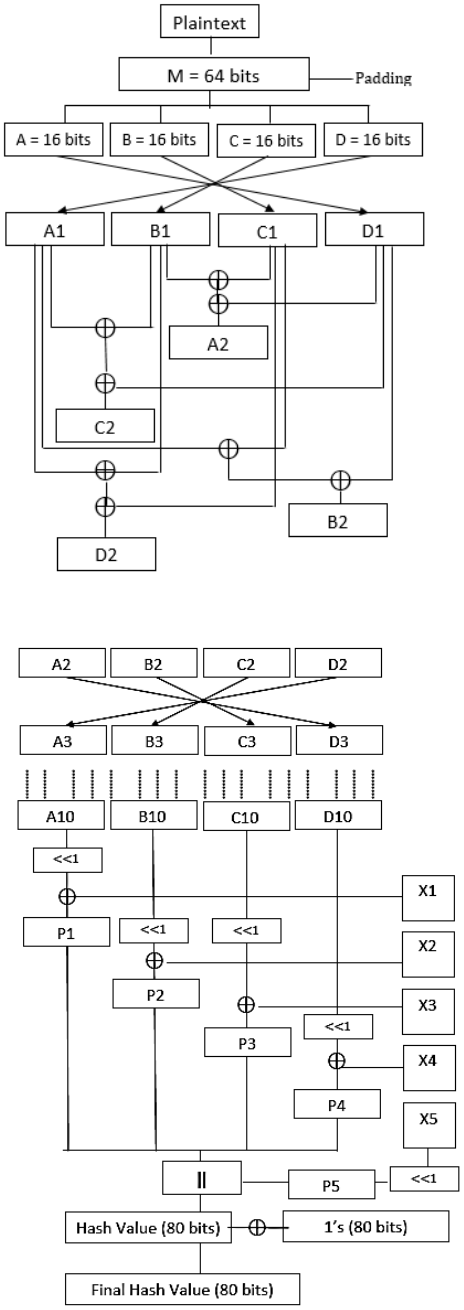

***Fig. 1**. Design 1: XOX function Structure*

3.1 Properties of XOX Hash Functions

3.1.1. Pre-processing

The pre-processing of XOX Function involve the Padding 0's

and 1's of plaintext below 64 bit to form 64 bit word. Parsing is dividing the message or plaintext into four 16 bit word. After which, is the shifting of words such that: A1= D; B1= C; C1=B and D1=A.

3.1.2. Constant Used

There are five 16 bit value of constant to be used which is in hexadecimal value.

X1= 1fe8
X2= cdad
X3= 1176
X4= b3f9
X5= a4c9

3.1.3. Computing the Message Digest

To compute the message digest, the following formulas are used:

Let A1= D; B1= C; C1=B and D1=A
For t = 0 to 10 do
$A_{(2..10)}=(B_{(2...10)} \oplus C_{(2...10)})\ D_{(2...10)}$;
$B_{(2...10)}=(A_{(2...10)} \oplus C_{(2...10)}) \oplus D_{(2...10)}$;
(1) $C_{(2...10)}=(A_{(2...10)} \oplus B_{(2...10)}) \oplus D_{(2...10)}$;
$D_{(2...10)}=(A_{(2...10)} \oplus B_{(2...10)})C_{(2...10)}$;

3.1.4. Computing Final Message Digest

To compute the final message digest; the following formulas are used:

P1 = S1(A10) ⊕ X1;
P2 = S1(B10) ⊕ X2;
P3 = S1(C10) ⊕X3;
P4 = S1(D10) ⊕X4;
P5 =S1(X5);
(2)

Final Hash Value=P1||P2||P3||P4||P4⊕1's(80Bits) (3)

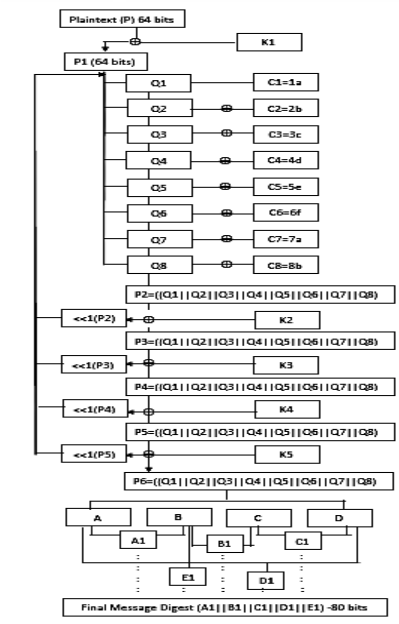

***Fig. 2**. Design 2: XTINGS-V1 Function Structure*

The properties of XTINGS-V1 includes padding, parsing, Qn(1..8) Calculation, K(1..5) Infusion, Message Digest and Final Message Digest calculation. After simulation, performance analysis on the prior design (XOX and XTINGS-V1), the author was not satisfied with the result and came up with a stronger version and led to the creation of a Novel hash algorithm, the XTINGS Hash algorithm. The new algorithm produces 160 bit fix hash value. The plaintext will form 128 bit block after padding. This block will be divided into two 64 bits block (L0, R0) and will xor to Key1 (K1), then parsing of the R0, a 64 bits into eight 8 bits block(Q1…Q8).

3.3 XTINGS Hash Function

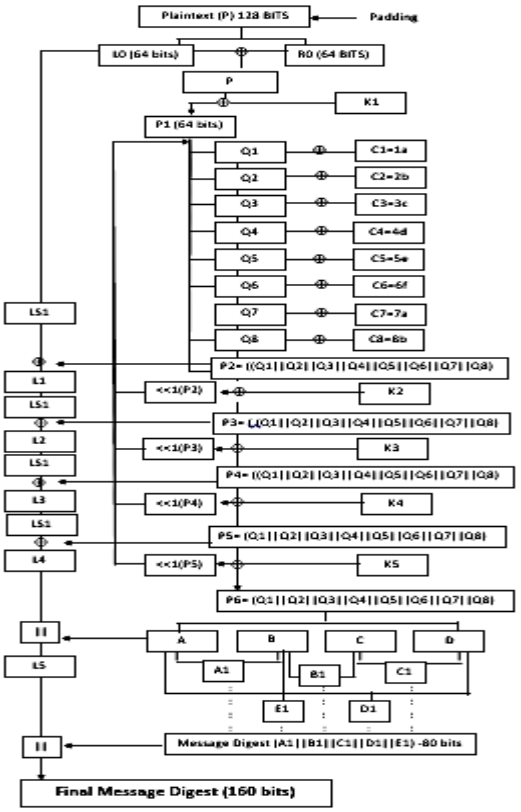

***Fig. 3**. Design 3: XTINGS Hash Function Structure*

3.3.1. Properties of XTINGS HASH Algorithm

The properties of the XTINGS Hash Algorithm are:

1. Padding of Plaintext
2. Parsing of plaintext into two 64 bit block
3. Qn=8 Calculation
4. Li Calculation
5. Key(1 to 5) Infusion
6. Message Digest Calculation
7. Final Message Digest Calculation

3.3.2. Padding of Plaintext

The XTINGS HASH Algorithm operates in a 128 bit block with a 64-bit key. The plaintext will undergo padding to form 128 bit blocks and will produce a 160 bit hash value for 6 rounds. The XTINGS HASH Algorithm is a combination of the following: padding, substitution, parsing, xoring, shifting, iteration and concatenation. The padding is a pre-processing operation by adding many 1 bits to the plaintext to form 128 bits block. The padding is performed in order to make the plaintext with an

equal 128 bits block before processing.

3.3.3. Parsing of Plaintext

The plaintext of 128-bit block and will be partitioned into two 64-bit block (L0 and R0). The 64-bit (L0 and R0) will have its own distinct calculation as shown on the succeeding operation.

3.3.4. Qn=8 Calculation

The message or plaintext Pi is partitioned into eight 8-bit block of Q1 to Q8 and to xor to an eight 8-bit constant value of C1 to C8. The following is the formula and the constant hexadecimal values:

Pi = Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8
Where:
Q1=Q1⊕(C1=1a);
Q2=Q2⊕(C2=2b);
Q3=Q3⊕(C3=3c);
Q4=Q4⊕(C4=4d);
Q5=Q5⊕(C5=5e);
Q6=Q6⊕(C6=6f);
Q7=Q7⊕(C7=7a);
Q8=Q8⊕(C8=8b);
Pi =Q1||Q2||Q3||Q4||Q5||Q6||Q7||Q8 (4)

3.3.5. Li Calculation

The computation of Li starts from partitioning the 128-bit plaintext into two 64-bit block of Lo and R0. The computation of Li is defined as: Li = <<1(Li)⊕ Pi.

Where:
<<1(Li) : The circular left shift by 1 bit of Li
⊕ : Bitwise logical XOR of Li and Pi

3.3.6. Key (K) (1 to 5) Infusion

There are five 64-bit constant value of Ki, where i=<5 to compute P2 to P6. The formula is defined as: Pi =Pi ⊕ Ki. The 64-bit constant value of Ki (1...5) is defined in hexadecimal value:

Where:
K1= 5f21bd46cb910e83
K2= 3fab08ec427d9b6c
K3= 2a8b1c9d5e4bb1f
K4= 100451939cdad082
K5= c0912f24e119d76cb

3.3.7. Message Digest

The message digest is calculated after achieving P6. The computation is to partition the P6 a 64-bits block into four 16-bit block of A,B,C and D. To compute the message digest, the following formulas are used:

Let A1=A⊕B; B1= B⊕C;
C1=C⊕D; D1=D⊕A and E1=B; (5)

3.3.8. Final Message Digest

The computation of the final message digest is achieved through concatenation of L5 and message digest. The following formulas are used:

Final Message Digest = Li || Message Digest (160 bit) (6)

4 SECURITY ASSESSMENT

To measure the security strength of any cryptographic algorithm, the calculation of avalanche effect has been used [16]. The avalanche effect is calculated on the following formula:

Avalanche Effect = (NC/TN) * 100 (7)

The NC is the number of the changed bits while TN means total number of bits in ciphertext [17].

4.1 Simulation of XTINGS Hash Avalanche Effect

Output

The table 1 shows the simulation and the result of the calculation on each processes of XTINGS Hash Algorithm on the two plaintext: “ISUCCSICT“ and “isuCCSICT“.

TABLE 1

SIMULATION OF PLAINTEXT USING XTINGS HASH ALGORITHM

PROCESS	*OUTPUT of Plaintext: ISUCCSICT*	*OUTPUT of Plaintext: isuCCSICT*
Plaintext Conversion	*4953554343534943 5411111111111111 (128 bits)*	*6973754343534943 5411111111111111 (128 bits)*
Value P1	*4263f91499d356d1*	*6243d91499d346d1*
Value P2	*5858c559c7bc2c5b*	*7868e559c7bc2c5a*
Value P3	*9531be12936ab951*	*95ace72655ec14e7*
Value P4	*655e7952c52d6816*	*6464cd3b5920337a*
Value L4	*9b1619d222a5c8275 09f*	*A3d7693d6f8beac43 27b*
Value P5	*50912f24e19d76cb*	*F2ea051cd59abd7b*
Value Message Digest	*3d50804fe6235b3c6 dcf*	*18431e3fffb7f9cb2a 34*
Value of Final Message Digest (160 Bits)	*9b1619d222a5c8275 09f3d50804fe6235b 3c6dcf*	*a3d7693d6f8beac43 27b18431e3fffb7f9c b2a34*

4.2 Comparison of XTINGS Hash Algorithm to XOX and XTINGS-V1 Function.

It is very evident on figures 1 and 2 where the prior hash design was illustrated and simulated and had gradually contributed in attaining XTINGS Hash algorithm on figure 3. Moreover, the table 2 signifies weaknesses of the two prior design that had led to the creation of a stronger hash algorithm.

TABLE 2

summary of the avalanche effect of the three design

Proposed Design	*No. of bits Flipped*	*Avalanche Effect*	*Hash Value*
XOX Function	*14*	*17.5 %*	*80 Bits*
XTINGS-V1 Function	*19*	*23.75%*	*80 Bits*
XTINGS Hash	*80*	*50%*	*160 Bits*

4.3 Measuring performance of XTINGS Hash

The table 3 shows the ciphertext of the three algorithm: XTINGS Hash, MD5 and SHA1 while the table 4 shows the result of XTINGS Hash as compared to the famous Hash Algorithms, the MD5 and SHA-1. The Avalanche Effect including the number of rounds /steps on the mathematical calculation is revealed on the table. All the algorithms hashed the text “ISUCCSICT” and changed the first three letters to lowercase “isuCCSICT”. Based on the results, the avalanche effect of MD5 obtained 52% on 64 rounds /steps, SHA1

obtained 47% on 80 rounds and the XTINGS Hash obtained 50% in just 6 rounds.

TABLE 3

THE CIPHERTEXT OF XTINGS HASH, MD5 AND SHA-1

Hash Algorithm	*Ciphertext of "ISUCCSICT"*	*Ciphertext of "isuCCSICT"*
MD5	*6fac22b533e8e3f43 0cd390793bf006b*	*C002f3ad88a506a0 42dbe63b7a7fc049*
SHA1	*4a804755bea8d4f84 817eada1752fdc55d 2d53a9*	*8884224913fa96dfb 0f4521cb2ffe7805a0 aec43*
XTINGS Hash	*9b1619d222a5c827 509f3d50804fe6235 b3c6dcf*	*a3d7693d6f8beac43 27b18431e3fffb7f9c b2a34*

TABLE 4

THE CIPHERTEXT OF XTINGS HASH, MD5 AND SHA-1

Hash Algorithm	*Hash Bit Size*	*Rounds/Step*	*Avalanche Effect*
MD5	*128*	*64*	*52%*
SHA1	*160*	*80*	*47%*
XTINGS Hash	*160*	*6*	*50%*

5 CONCLUSION

This paper presented the XTINGS Hash algorithm with a 160 bit message digest. It is a new hash algorithm that was gradually attained based on prior designs, the XOX and XTINGS-V1Function. The XTINGS Hash mathematical calculation includes pre-processing by performing padding of plaintext to form 128 bit block, constant used, constant key(1..5)value, computation of message digest and final message digest. Other methods were used such as: padding of bits, parsing, shifting of bits, switching of 16 bit word, xoring and concatenation of the final value of L5. The Comparison of XTINGS Hash on MD5 and SHA1 was also conducted. The Table 3 shows the ciphertext of the three algorithms. On Table 4, is the avalanche effect of MD5, SHA1 and XTINGS Hash. Based on result, MD5 avalanche rate is 52%, the SHA1 obtained less than the expected rate on 47% while the XTINGS Hash was able to meet the minimum expected rate of 50%. The XTINGS Hash algorithm has a low number of rounds but this does not affect the security performance and yet it produces a significant result displaying a characteristic of a cryptographic property of diffusion.

6 ACKNOWLEDGMENT

The author wish to thank Isabela State University, Echague – Main Campus for the financial support in this Research.

7 REFERENCES

[1] Y. Bang, D. Lee, Y. Bae, and J. Ahn, "Improving Information Security Management: An Analysis of ID-Password Usage and a New Log-in Vulnerability Measure", International Journal of Information Management, Volume 32, Issue 5, pp. 409 – 418, 2012.

[2] Sans.Org., Password Construction Guidelines, 2017 (online) available at https:// www.sans.org

[3] M. Zviran and W. Haga, Password Security: An Exploratory Study. Naval Postgraduate School, Monterey, California, 1990 (online) available at https://archive.org/stream/userauthenticati00cole/userauthenticati00cole_djvu.txt

[4] B. Schneier, Schneier Security Article: Cryptanalysis of SHA-1, 2005 (online) available at https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html

[5] A. Hossain, K. Islam, S. Kumar Das, and A. Nashiry, "Cryptanalyzing of Message Digest Algorithms MD4 and MD5", International Journal on Cryptography and Information Security (IJCIS), Vol. 2, No. 1, March 2012.

[6] Man Young Rhee, Internet Security, Cryptographic principles, algorithms and principles, pp.149, 2003 John Wiley & Sons, Ltd ISBN 0-470-85285-2.

[7] T. Lakshmanan, and M. Muthusamy, "A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes", The International Arab Journal of Information Technology Vol.9, No.3 pp. 262–267, 2012.

[8] X. Wang, and H.Yu, "How to break MD5 and other Hash Functions", Annual International Conference on the Theory and Application of Cryptographic Techniques, pp.19-35, 2005.

[9] M.D.A. Chawdhury and A. Habib, "Security Enhancement of MD5 Hashed Passwords by Using the Unused Bits of TCP Header", Proceedings of 11th International Conference on Computer and Information Technology (ICCIT), pp. 714 – 717, 2008.

[10] A. Bhandari, "Enhancement of MD5 Algorithm for Secured Web Development", Journal of Software, Vol. 12, No. 4, pp. 240 – 252, 2017.

[11] E. Maliberan, A. Sison and R. Medina, "A New Approach in Expanding the Hash Size of MD5", International Journal of Communication Networks and Information Security (IJCNIS), Vol. 10, No.2, pp. 374-378, 2018.

[12] R. Roshdy, M. Fouad, M. Aboul-Dahab, "Design and Implementation of a New Security Hash Algorithm Based on MD5 and SHA-256", International Journal of Engineering Sciences & Emerging Technologies (IJESET), Vol.6, Issue 1, pp.29 – 36, August 2013.

[13] C. San Jose, B. Gerardo, and B. Tanguilig III, "Enhanced SHA-1 on Parsing Method and Message Digest Formula", The Second International Conference on Electrical, Electronics Computer Engineering and their Applications (EECEA), pp. 1 - 9, 2015.

[14] Confusion and Diffusion (online) available at https://en.wikipedia.org/wiki/Confusion_and_diffusion

[15] Cryptography, "Why Does SHA1 have 80 rounds?" (online) available at https://crypto.stackexchange.com/questions/14971/why-does-sha-1-have-80-rounds

[16] A. Kumar, and N. Tiwari, "Effective Implementation of Avalanche Effect on AES", International Journal of Security, Privacy and Trust Management (IJSPTM), Vol. 1, 2012.

[17] Avalanche Effect, (online) available at https://en.wikipedia.org/wiki/Avalanche_effect, 2018.