DOI: 10.21917/ijct.2015.0176

WSES: HIGH SECURED DATA ENCRYPTION AND AUTHENTICATION USING WEAVING, ROTATION AND FLIPPING

A. Yesu Raja[1] and S. Arumuga Perumal[2]
Department of Computer Science, S.T. Hindu College, India
E-mail: [1]a_yesuraja@yahoo.co.in and [2]visvenk@yahoo.co.in

Abstract

Data security is the very important part in the network data communication. Avoidance of the information hacking and steeling are very challenging part for network data communication. Now-a-days people are using many encryption and decryption techniques for data security. But all encryption and decryption techniques are having more time occupation or less security for the process. This paper proposed high level security approach to encryption and decryption for data security. Two levels of securities are used in this proposed method. First one is data encryption and the second one is hash value generation. The proposed Weaving based Superior Encryption Standard (WSES) uses a novel weaving based approach. The weaving array generation is done by Elementary Number Theory Notation (ENTN) method. The weaving array has multiple private keys for XOR encryption. After encryption the error value is extracted from the encrypted array and weaving array. This error value is sent to the other side. The novel approach for hash value generation uses the encrypted array. After encryption, the encrypted array is rotated into four degrees and each degree data are converted to vector format and arranged on by one under the vector. Finally a 2D Rotational Encryption Matrix (REM) is obtained. After this process a REM copy is converted to mirror flip and it is need as Flipped Matrix (FM). The FM is concatenated under the REM and converted to vector using the zigzag operation. Finally this process gives two bytes hash value from the vector. This proposed method executes very fast and provide high security. This method is much reliable to small size applications and also used for any type of data security.

Keywords:

Data Security, XOR, WSES, Encryption, Decryption, Hash, ENTN, Weaving, REM, Flipped Matrix

1. INTRODUCTION

Secret sharing schemes have been independently introduced by Blakley [1] and Shamir [2]. In the Shamir's secret sharing scheme said by, a secret m is divided into *n* number of pieces $m_1; \ldots ; m_n$, which satisfies, 1) knowledge of whichever *t* or more m_i pieces makes m easily computable, and 2) knowledge of whichever *t*-1 or fewer m_i pieces leaves m completely undetermined. Shamir's secret sharing scheme, which is also called as (*t*, *n*) secret sharing, can be proven to be information-theoretically secure.

Sahai and Waters [3] are addressed this issue by introduce the notion of Attribute Based Encryption (ABE). The ABE is a new public key based one-to-many encryptions and that encryption standard enables access control over encrypted data using access policies which are associated with private keys [4].

Visual cryptographic scheme (VCS) is a category of secret image sharing schemes. In a threshold (k, n), where $k \leq n$, a secret image is encrypted into *n* shadow images (shadows) by growing a secret pixel into *m* sub-pixels. The visual quality of reconstructed image is very poor by the VCS. Another one polynomial-based secured image sharing scheme can recover the distortion levels very less in the secret image, but its decoding process needs Lagrange interpolation [5], [6]. The authors combined the VCS and polynomial-based secured image sharing to design and development by a two-in-one VCS, with different types of decoding options [7], [8]. This two-in-one Visual cryptographic scheme is having two parts, first stage is stacking to see a formless reconstructed image and the second stage is used to reconstruct the secret image perfectly by Lagrange interpolation.

In the symmetric key cryptography, both transmitter and receiver use a same key for encryption and decryption and it needs a secure channel or a trusted third party to exchange the key. In other words, the security of the symmetric key cryptography depends on the way of key exchange to some extent. To avoid this issue, public-key cryptography was used, in which a transmitter and a receiver own different private keys and share a public key. The keys are mathematically related. Encryption and decryption are performed by using the public key and the private key. The security of the public-key cryptography mainly lies on the complexity of the mathematical relationship between the public key and private key [9], [10].

In previous years the author of paper [11] have analyzed exam results from the Data Security course and noticed that students had trouble understanding cryptographic algorithms, which resulted in lower grades on the part of the exam covering this area and consequently the final grades for the exam. The reason for this was the difficulty for students to closely follow the execution of algorithms, because they are too complex to be calculated manually on paper.

In order to help students to better understand the material many teachers in different areas use educational software systems. These systems have a significant presence in engineering sciences, where it is important for students to have practical work in addition to classes. Systems differ significantly depending on the area in which they are used. It is common in hardware related courses to use software simulators in order to avoid the high price of the necessary equipment for the laboratory exercises. In software related areas that teach algorithms, software systems for visual representation of the algorithms [12] are often used for both teaching and laboratory exercises.

In [13], an algorithm is introduced to the encryption and decryption method is used to speed up the execution process. That algorithm is called up Advanced Encryption Standard (AES) cryptosystems. Even though the traditional cryptographic techniques could be applied separately to communication channels, exchange the secret keys are required to between transmitter and intended receiver. The Advanced Encryption Standard (AES) design implementation utilized 16 map online and offline key expansion. A simple design utilized only eight cores for online key expansion and six cores for offline key expansion. This implementation explores to the data level and task level parallelism. This implementation's main aim is to reduce the core and increased the workload [14].

Patient health record system is encrypted with Attribute Based Encryption (ABE). It is mainly focus on multiple data owner's situation. The multiple security domains reduced the key management complexity for owner and user [15].

In cloud security with the involvement of the third party cloud services, a crucial problem is that the identity attribute in the Access Control Policies (ACPS) often reveal privacy-sensitive information about users and leak secret information about the content. The secrecy of the content and the privacy of the users are, thus, not fully confined if the identity attributes are not protected. Further confidentiality, both individual as well as organizational, all process is considered a key requirement in all solutions, it is including cloud services, for Digital Identity Management (DIF) [16].

Radio Frequency Identification security protocols often rely on hash function methods. Some of the applications need protected from collision and some of them do not needing pre-image security. An interesting case is established by keyed Message Authentication Codes (MAC) frequently used in this context. Here, a lightweight hashing method can require less area and then a lightweight block cipher in a Message Authentication Codes mode at a permanent level of offline and online protection. Message Authentication Codes can be also designed using sponge primitives [17].

The most widely used hash function is SHA family, which contains SHA-0, SHA-1, SHA-2 published by National Institute of Standards & Technology (NIST). In the recent years, the cryptanalysts processes have establish collisions on the MD4, MD5 and SHA-0 hashing algorithms. Moreover, a method for discovery SHA-1 collisions with minimum expected amount of work has been published. In this case, NIST decided to hold a public competition for a new Standard Hash Algorithm (SHA-3) in 2007 [18]. After two round competition, 14 candidate algorithms were selected [19].

It was not until recently though that some important work on trivial hash functions has been a performed [20] describes ways of using the PRESENT block cypher in hashing methods of operation and [21] take the approach of designing a committed trivial hash function based on a sponge building [22], [23] resulting in couple of hash functions QUARK and PHOTON.

A hash function is a function that maps a bit sequence of arbitrary length to a fixed-length output. As a rule, three security requirements are mentioned for a hash function. First, it should be pre-image resistant: Given an output (hash value), it should be hard to find an input (message) that would map to this output. Second, it should be second pre image resistance: Given a hash value and a corresponding message, it should be difficult to find another message with the same hash value. Finally, it should be collision resistant: It should be infeasible to find two messages with the same hash value.

2. PROPOSED ENCRYPTION METHOD

Data security is very important in now a day because most of the important information can be communicate through computer.

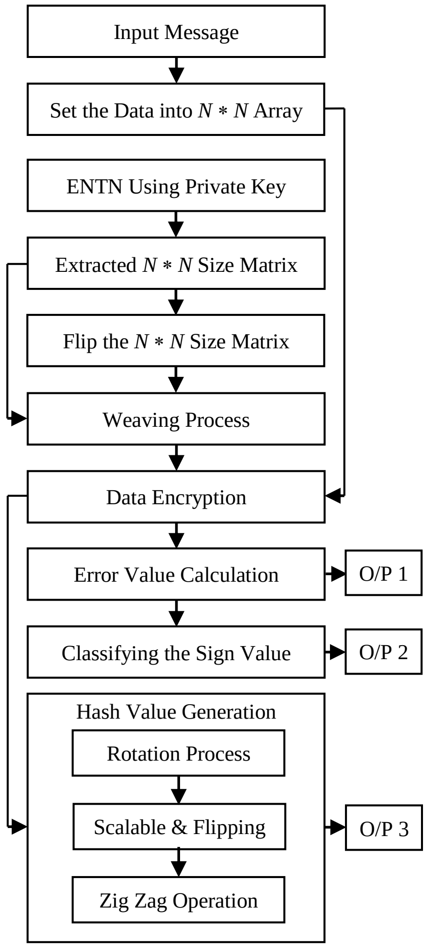


Fig.1. Block Diagram for Encryption and Hashing

The communication can be done with locally or World Wide Web. Data steeling is very challenging trouble in the modern society. The encryption can be done the data with fast and more security. Low secured data encryption can be easily decrypting the data with hackers. This paper mainly proposed a novel method of encryption and decryption for data with hashing method. If the information can be changed while the transmission, the hashing signature value also change. If the hashing signature value can be changed, the process will be rejected.

The Fig.1 explains how can be the data encryption and generation of the hash value. Input data can be set into the $N * N$ matrix. Generate the matrix using the Elementary Number Theory Notation (ENTN) supported with private key. The private key can be converted near by the prime number. This matrix size can be greater than the input $N * N$ message matrix. O/P1, O/P2, O/P3 are representing the three outputs. O/P1 is sending the error value while O/P2 is sending the sign values. The sign values are having – and + signatures. Finally the O/P3 contains hash value. These three outputs are sent to the other side.

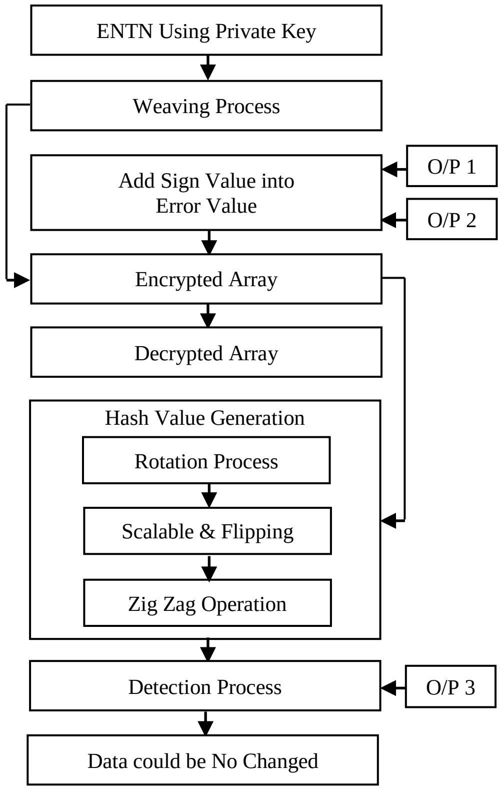


Fig.2. Block Diagram for Decryption

The Fig.2 can be indicating the decryption process of the data. The ENTN can be used to generate the weaving array using the private key. The error values are reconstructed with error and sign value. These are received from client. The reconstructed error value is subtracted with the Weaving Array. After the process, the subtracted value is called the encrypted array. The encrypted array can be converted to decrypted array using with the weaving array.

In addition, the encrypted array can be used to the hash value creation process. After hash value computation, the hash value is matched with the clients hash value. If the hash value is same, the authentication process is successful otherwise the data not successful because may be changed while the data sending.

2.1 KEY MATRIX

Elementary number theory concerning $Z = \{\ldots,-2, -1, 0, 1, 2,\ldots\}$ of integers and the set $N = \{0, 1, 2, \ldots\}$ of natural numbers. For any integer and positive integer n, there are unique integers q and r such that $0 \le r < n$ and $a = qn + r$.

The value $q = \lfloor a/n \rfloor$ is the quotient of the division. The value $r = a \bmod n$ is the reminder (or residue) of the division. We have that n/a if and only if a mod $n = 0$.

The integers can be divided into n equivalence classes according to their reminders modulo n. The equivalence class modules n contain an integer, a_i as following Eq.(1).

$$[a_i]_n = a + \left(\left(\sum_{p=0}^{i+1} p\right) - 1\right)n + kn \tag{1}$$

$$z = \mathrm{mod}(PWD, n)$$

where,

$k \in N$ - Assume $k = [0, a]$

n - Any prime number using single byte

a - find the nearby prime number for z (Greater than z)

i - Row of elementary number series (assume $I = [0, N]$)

$k \in N$ - Natural Number

2.2 HORIZONTAL FLIPPING

The Secret Message (SM) has been converted into $N * N$ size of matrix it is Secret Message Array (SMA). The dimension of private key matrix is 2 * Height of secret message. Divide the private key matrix into two equal parts A and B. The second part matrix B has horizontally flipped using the Eq.(2).

$$B'_{i,j} = B(i, W_B - 1 - j) \tag{2}$$

where,

$i \in [0, W_H - 1]$; $j \in [0, W_B - 1]$

W_H - B Matrix Height; W_B - B Matrix Width.

2.3 WEAVING MATRIX

Weaving process has very important in this proposed method. The private key is used to encrypt the input message. The Weaving Matrix (WM) contains multi range of keys. The weaving matrix is constructed by using the following Eq.(3).

$$WM_{i,j} = \begin{cases} A_{i,j} & \mathrm{mod}(i,j)=0 \ \& \ \mathrm{mod}(j,2)=0 \\ B'_{i,j} & \mathrm{mod}(i,j)=0 \ \& \ \mathrm{mod}(j,2)=1 \\ B'_{i,j} & \mathrm{mod}(i,j)=1 \ \& \ \mathrm{mod}(j,2)=0 \\ A_{i,j} & \mathrm{mod}(i,j)=1 \ \& \ \mathrm{mod}(j,2)=1 \end{cases} \tag{3}$$

$$A = \begin{pmatrix} A_1 & A_2 & A_3 & A_4 & A_5 \\ A_6 & A_7 & A_8 & A_9 & A_{10} \\ A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{16} & A_{17} & A_{18} & A_{19} & A_{20} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \end{pmatrix}$$

$$B' = \begin{pmatrix} B'_1 & B'_2 & B'_3 & B'_4 & B'_5 \\ B'_6 & B'_7 & B'_8 & B'_9 & B'_{10} \\ B'_{11} & B'_{12} & B'_{13} & B'_{14} & B'_{15} \\ B'_{16} & B'_{17} & B'_{18} & B'_{19} & B'_{20} \\ B'_{21} & B'_{22} & B'_{23} & B'_{24} & B'_{25} \end{pmatrix}$$

$$WM = \begin{pmatrix} A_1 & B'_2 & A_3 & B'_4 & A_5 \\ B'_6 & A_7 & B'_8 & A_9 & B'_{10} \\ A_{11} & B'_{12} & A_{13} & B'_{14} & A_{15} \\ B'_{16} & A_{17} & B'_{18} & A_{19} & B'_{20} \\ A_{21} & B'_{22} & A_{23} & B'_{24} & A_{25} \end{pmatrix}$$

Above the matrix represents a sample structure of constructing *WM* from *A* and *B*. The matrix *A* is represented the SMA size of matrix which extracted from the Eq.(1). Matrix B' represents flipped matrix from the matrix *A*. The *A* and $B'$are the same size of SMA. After flipping process the matrix B' can be weaving on the matrix *A* based on Eq.(3). The matrix *WM* represents sample structure for Weaving Matrix. For example the matrix *A* size is 5 ∗ 5 and matrix B' size is 5 ∗ 5. After applying the weaving process *WM* size is also 5 ∗ 5.

The *WM* values are having large numeric value. So the normalization process is applied to restrict the values *PK* within 0 to 255. The values are constructed by the Eq.(4).

$$MV = \max(WM)$$

$$PK_{i,j} = fix\left(\frac{WM_{i,j}}{MV}\right) * R \tag{4}$$

where,

R - Maximum storage capacity of bytes data types

MV - Maximum value in WAM matrix

$PK_{i,j}$ - Private Key Weaving

2.4 DATA ENCRYPTION

This encryption standard encrypted the original image and data using two steps of private keys. First one is gives from user and another one, multi private keys is automatically created. The size of private key matrix is same to the secrete message matrix size. The XOR operation is used for the data encryption. This type of data encryption is highly secured and reliable. The TL_X and TL_Y are defining the direction of the encryption process.

$$TL_X = \{0,-1;1,0\}$$

$$TL_Y = \{-1,0;0,1\}$$

$$ED_{i,j} = XOR\left(SM_{i,j}, PK\left(TL_Y(0)+i, TL_X(0)+j\right)\right) \tag{5}$$

$$EDM_{i,j} = XOR\left(ED_{i,j}, PK\left(TL_Y(k)+i, TL_X(k)+j\right)\right)$$

where,

$k \in [1, 3]$, *TL* - Template

$i \in [0, SM$ Height]; $j \in [0, SM$ Width]

EDM - Encrypted Data Matrix

2.5 ERROR VALUE CALCULATION

Subtraction of the encrypted data from weaving array is called error value. The extracted values are sent to the server by client. The real encrypted data is not sent to the server. The error values are the one and only component used to decrypt the secrete message. The error values separated using the Eq.(6).

$$ET_{i,j} = WM_{i,j} - EDM_{i,j} \tag{6}$$

The *ET* values are distributed from negative numbers to positive numbers. This type of format takes higher memories. To control two byte data memory into one byte data memory the negative and positive sign value are extracted from *ET* data.

$$SBM_{i,j} = \begin{cases} 0, & ET_{i,j} < 0 \\ 1 & Otherwise \end{cases} \tag{7}$$

The Sign Bit Matrix (*SBM*) has 0 and 1 values. The matrix value 0 represents negative sign and 1 represents positive sign.

$$ED_{i,j} = abs\left(ET_{i,j}\right) \tag{8}$$

After applying the absolute function on the *ET*, The *ED* is generated without the sign. The *ED* and *SBM* are sending to the server by client.

3. PROPOSED HASH VALUE CREATION

The hash value is created after the encrypted data is created. A hash value is sending form client to server. This hash value is created from the real encrypted data. If the real data is changed during sending time, the server computes a different hash value and compared with client side hash value. This hash value is used against the replay attack.

3.1 ROTATIONAL ENCRYPTED MATRIX

Encrypted matrix can be rotated into four directions respectively 00, 900, 1800 and 2700 for the hash value creation. The directional matrix for rotation can be defined as follows.

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

The column rotation based matrix multiplication is achieved using the Eq.(9).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{9}$$

The following directional templates are used for the multi directions such as 900, 1800 and 2700 respectively.

- 90° Template → [0, -1; 1, 0]
- 180° Template → [-1, 0; 0, 1]
- 270° Template → [0, 1; -1, 0]

After rotation process four 2D matrixes related with 00, 900, 1800 and 2700 are obtained. The 2D matrixes can be converted into vector format. After vector conversion four vectors are generated. These four vectors can be arranged one by one. The REM can be used to explain, how to rearrange the vectors from multi directional rotational matrix into 2D matrix named as Rotational Encrypted Matrix (REM). The following four directional matrixes can be represented the sample structure of multi directional rotational matrix.

$$0^\circ = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \quad 90^\circ = \begin{bmatrix} G & D & A \\ H & E & B \\ I & F & C \end{bmatrix}$$

$$180^\circ = \begin{bmatrix} I & H & G \\ F & E & D \\ C & B & A \end{bmatrix} \quad 270^\circ = \begin{bmatrix} C & F & I \\ B & E & H \\ A & D & G \end{bmatrix}$$

The following matrix can be constructed from the above matrix. This *REM* matrix can be sample structure of the Rotational Encrypted Matrix.

$$REM = \begin{bmatrix} A & B & C & D & E & F & G & H & I \\ G & D & A & H & E & B & I & F & C \\ I & H & G & F & E & D & C & B & A \\ C & F & I & B & E & H & A & D & G \end{bmatrix}$$

In the *REM* matrix first row represents the vector of 00 2D matrix, second row represents the linear array of 900 2D matrix, third row represents the linear array of 1800 2D matrix and fourth row represents the linear array of 2700 2D matrix. The 4 × *N* dimensional *REM* is achieved from the multi directional rotational encrypted matrixes.

3.2 HORIZONTAL MIRROR FLIPPING

The *REM* is undergone the horizontal mirror flipping process. The output matrix of flipping process *FM* is jointed at the bottom of *REM*.

The flipping process is performed using the Eq.(10).

$$FM_{i,j} = REM\left(i, W_B - 1, j\right) \tag{10}$$

where,

$i \in [0,\ W_H - 1]$

$j \in [0,\ W_B - 1]$

W_H - REM Height; W_B - REM Width

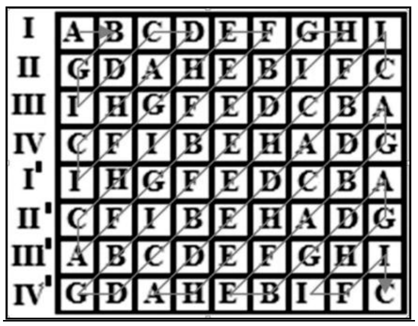
The *FM* is concatenated with the *REM* and it is named as Flipped Rotational Encrypted Matrix (FREM). This process can be represented by the following matrix. The dimension of the concatenated matrix is $2WH \times WB$.

$$FREM = \begin{bmatrix} A & B & C & D & E & F & G & H & I \\ G & D & A & H & E & B & I & F & C \\ I & H & G & F & E & D & C & B & A \\ C & F & I & B & E & H & A & D & G \\ I & H & G & F & E & D & C & B & A \\ C & F & I & B & E & H & A & D & G \\ A & B & C & D & E & F & G & H & I \\ G & D & A & H & E & B & I & F & C \end{bmatrix}$$

The above matrix represents the first four rows are *REM* matrix and remaining four rows are *FM* matrix.

3.3 ZIGZAG VECTOR CONVERSION

The 2D FREM matrix is converted into vector form for the hash value generation using zigzag scanning method. This can be illustrated in the Fig.5(a).



(a)

0	1	2	3	4	5	…	80
A	B	G	I	D	C	…	C

(b)

Fig.3. (a) Flow of zigzag in the FREM (b) Vector Conversion Using Zigzag

In the Fig.3(a) 2D matrix has been converted into a Vector Matrix Fig.3(b) using zigzag operation and it is called as Zigzag Vector Matrix (ZVM). The Fig.3(a) indicates the flow of the data which shows in 2D matrix. The Fig.3(b) indicates the rearranged data from Fig.3(a) matrix. The rearrangement Fig.3(b) matrix is converted into vector data.

3.4 TWO BYTE HASH VALUE CREATION

Zigzag Vector Matrix contains 0 to *n*-1 indexes. In vector each value is having one byte memory size. If the hash value is in one byte memory size, the hash value easily hacks with the attacker. So this method is created with two byte memory hash value.

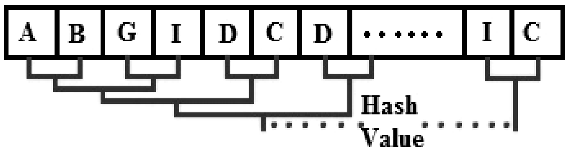


Fig.4. Sample Structure for Hash Value Generation

The Fig.4 indicates how to generate the two bytes single hash value. Two bytes memory hash value is created using the Eq.(11).

$$A = XOR\left(IntF\left(ZVM_0, ZVM_1\right), IntF\left(ZVM_2, ZVM_3\right)\right) \tag{11}$$

$$IntF(x, y) = (x \ll 8) + y$$

$$A = XOR\left(A, IntF\left(ZVM_i, ZVM_{i+1}\right)\right) \tag{12}$$

$$i \in [4, n-1] \text{ and } \mathrm{mod}(i, 2) = 0$$

where,

A - *XOR* Result

IntF - Integration Function to merge the sequence two bytes into a two bytes data.

ZVM - Input Vector

XOR() - Function to perform exclusive OR operation

<< - Left Shift operator

n - Total number of Input Vectors

The Eq.(11) is used to extract the initial XOR value (*A*). The first *A* value is constructed by the elements 0 to 3. So the next equation is started with the element 4 and uses up to *n*-1 elements. The left shifting is used to shift the single byte data into left side by 8 times. After shifting process the memory is filled by two byte data range. The left most individual bytes (among the two bytes data rang) can be represented by *x*. the right most individual bytes (among the two bytes data range) can be represented by *y*. After processing all i^{th} elements Eq.(12) a new two byte HASH value is generated. Finally get the two bytes value is called hash value.

4. ANALYSIS

This experimental analysis is performed based on the time factor for various sizes of data. This time factor performance is computed with the system CPU 2.13GHz with 2GB RAM. This time factor analysis is compared with the different size of data bytes. The bytes data are compared by the different types of methodologies namely Advance Encryption Standard (AES), Data Encryption Standard (DES), Blowfish (BF) [24] and the proposed Weaving based Superior Encryption Standard (WSES).

Table.1. Time Taken for Encryption

Data Size (bytes)	Encryption time taken (seconds)			
	Methods			
	AES	DES	BF	WSES
7200	6	12	8	0.4
154000	2	20	10	0.8
203000	9	25	12	1
351000	12	40	17	2
476000	15	52	22	3
589000	17	64	25	4
718000	20	77	30	5
1222000	32	130	48	9
1715000	42	180	66	14
3156000	76	329	119	30

The Table.1 indicates the time taken of various methods for the data encryption process. Numbers of various bytes data are used in the time calculation process. The Fig.5 indicates the graphical representation of time taken for the various data encryption processes.

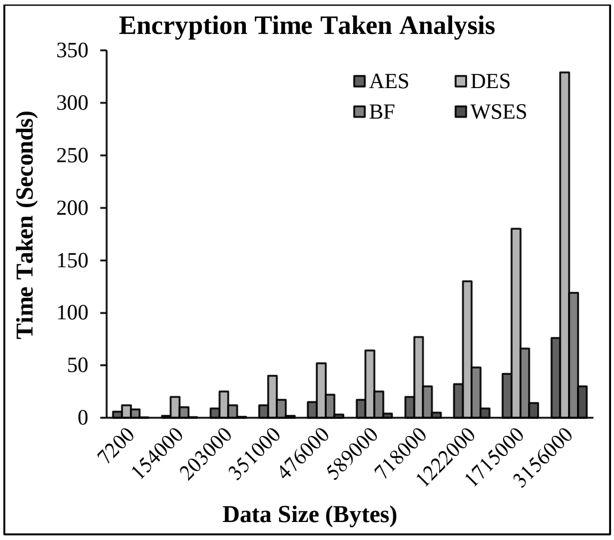


Fig.5. Time taken graphical representation for data encryption

Decryption is the reverse process of the encryption method. In the encryption real data could be changed to different format. So the encrypted data can be converted to real data and it is called decryption. The decryption process time taken for WSES method is very less when compared with the existing methods.

Table.2. Time Taken for Decryption

Data Size (bytes)	Data Decryption			
	Methods			
	AES	DES	BF	WSES
7200	2	8	4	0.3
154000	4	16	6	0.7
203000	4	21	7	1
351000	8	36	13	2
476000	11	48	18	3
589000	13	60	21	4
718000	16	73	26	5
1222000	28	126	44	9
1715000	38	176	62	13
3156000	72	325	115	29

The Table.2 indicates the time taken of various methods for the data decryption process. Number of various bytes data are using in the time calculation process. The Fig.6 indicates the graphical representation of time taken analysis for the various data decryption process.

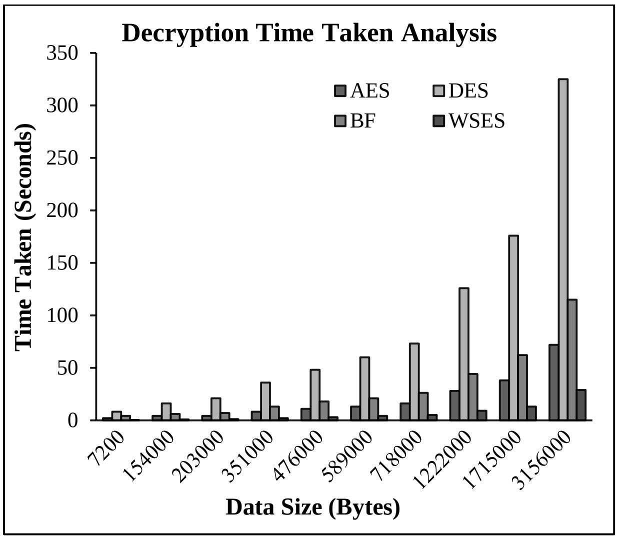


Fig.6. Time taken graphical representation for data decryption

Total time taken is calculated for the data encryption and decryption process individually. In this encryption and decryption methods, time calculation is performed using 10 different sizes of data. The Encryption total time taken Fig.7 is computed by summing up the time consumption of each different size of data to do encryption process.

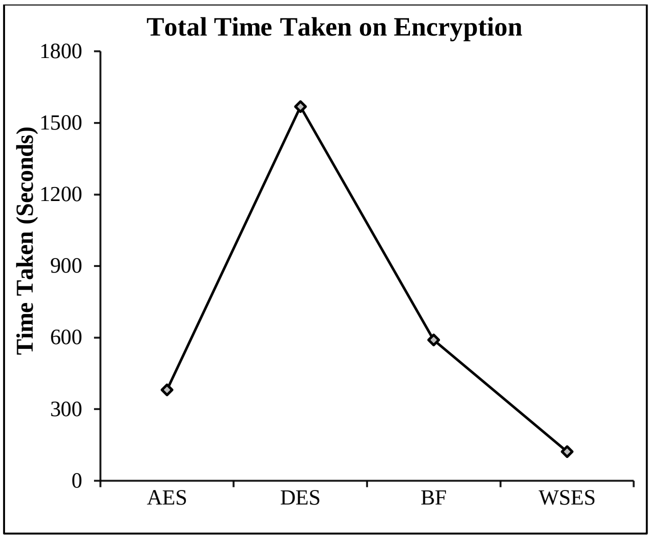


Fig.7. Encryption Total Time Taken Analysis Chart

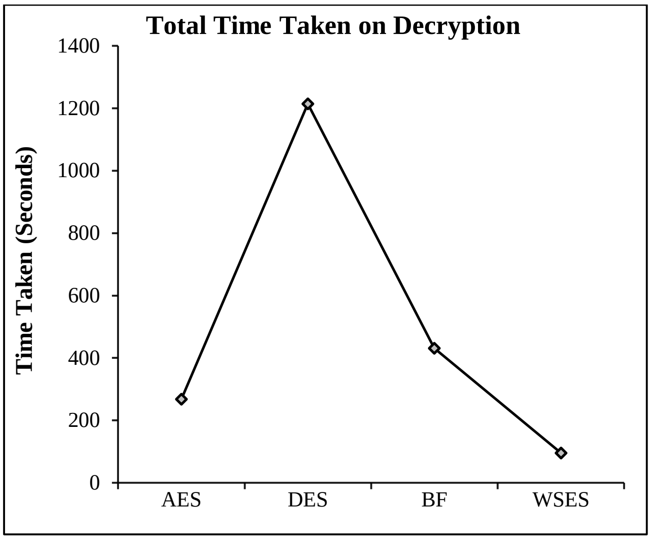


Fig.8. Decryption Total Time Taken Analysis Chart

The Fig.8 shows the Decryption total time taken for 10 difference type of data. The Decryption total time taken is computed by summing up the time consumption of each different size of data to do decryption process. From the Fig.7 and Fig.8 it can be easily understand that the proposed WSES method is paramount in encryption methods.

5. CONCLUSION

This is used for network data communication to any type of data security. This security system has the novel approach of encryption based Weaving Algorithm and also it proposes a new Hash value generation method for high secured authentication. The proposed security system avoids information hacking and steeling. If the information is hacked and has committed any change, the hash value also changed. This Weaving Algorithm is more secured and speed than the earlier versions of encryption techniques. The proposed Encryption is 179.16% faster in execution and Decryption is 211.78% faster than the existing AES technique. The proposed security system is more suitable for the applications wherever high security as well as higher execution speed is required.

REFERENCES

[1] G.R. Blakley, "Safeguarding Cryptographic Keys", *Proceedings of the AFIPS National Computer Conference*, pp. 313-317, 1979.

[2] Adi Shamir, "How to Share a Secret", *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, 1979.

[3] Amit Sahai and Brent Waters, "Fuzzy Identity-based Encryption", *Proceedings of 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, pp. 457-473, 2005.

[4] Vipul Goyal, Omkant Pandey, Amit Sahai and Brent Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data", *Proceedings of the 13th ACM conference on Computer and Communications Security*, pp. 89-98, 2006.

[5] Chih-Ching Thien and Ja-Chen Lin, "Secret Image Sharing", *Computers & Graphics*, Vol. 26, No. 5, pp. 765-770, 2002.

[6] Chih-Ching Thien and Ja-Chen Lin, "An Image-Sharing Method with User-Friendly Shadow Images", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 12, pp. 1161-1169, 2003.

[7] Sian-Jheng Lin and Ja-Chen Lin, "VCPSS: A Two-in-One Two-Decoding Options Image Sharing Method Combining Visual Cryptography (VC) and Polynomial-Style Sharing (PSS) Approaches", *Pattern Recognition*, Vol. 40, No. 12, pp. 3652-3666, 2007.

[8] Peng Li, Pei-Jun Ma, Xiao-Hong Su and Ching-Nung Yang, "Improvements of a Two-in-One Image Secret Sharing Scheme based on Gray Mixing Model", *Journal of Visual Communication and Image Representation*, Vol. 23, No. 3, pp. 441-453, 2012.

[9] O. Goldreich, "*Foundations of Cryptography: Volume 2, Basic Applications*", First Edition, Cambridge University Press, 2004.

[10] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, Vol. 21, No. 2, pp. 120-126, 1978.

[11] Z. Stanisavljevic, V. Pavlovic, B. Nikolic and J. Djordjevic, "SDLDS-System for Digital Logic Design and Simulation", *IEEE Transactions on Education*, Vol. 56, No. 2, pp. 235-245, 2013.

[12] Guido Rößling, Markus Schüer and Markus Schüer, "The ANIMAL Algorithm Animation Tool", *Proceedings of the 5th Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education*, Vol. 32, No. 3, pp. 37-40, 2000.

[13] Abbasi-Moghadam, V.T. Vakili, and A. Falahati, "Combination of Turbo Coding and Cryptography in Non-Geo Satellite Communication Systems", *Proceedings of International Symposium on Telecommunications*, pp. 666-670, 2008

[14] Bin Liu and Bevan M. Baas, "Parallel AES Encryption Engines for Many Core Processor Arrays", *IEEE*

Transactions on Computers, Vol. 62, No. 3, pp. 536-547, 2013.

[15] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren and Wenjing Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption", *IEEE Transactions on Parallel And Distributed Systems*, Vol. 24, No. 1, pp. 131-143, 2013.

[16] Liberty Alliance, Available at http:// projectliberty.org/.

[17] Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche, "Sponge Based Pseudo Random Number Generators", *Proceedings of the 12th International Conference on Cryptographic Hardware and Embedded Systems*, pp. 33-47, 2010.

[18] National Institute of Standards and Technology, "Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family", *Federal Register*, Vol. 72, pp. 62212-62220, 2007.

[19] Andrew Regenscheid, Ray Perlner, Shu-jen Chang, John Kelsey, Mridul Nandi and Souradyuti Paul, "Status Report on the 1st Round of the SHA-3 Cryptographic Hash Algorithm Competition", NISTR 7620, Information Technology Laboratory, National Institute of Standards and Technology 2009.

[20] Andrey Bogdanov, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw and Y. Seurin, "Hash Functions and RFID Tags: Mind the Gap", *Proceeding of 10th International Workshop Cryptographic Hardware and Embedded Systems*, pp. 283-299, 2008.

[21] Jean-Philippe Aumasson, Luca Henzen, Willi Meier and María Naya-Plasencia, "Quark: A Lightweight Hash", *Journal of Cryptology*, Vol. 26, No. 2, pp. 313-339, 2010.

[22] Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche, "On the Indifferentiability of the Sponge Construction", *Proceedings of the 27th Annual International Conference on Theory and Applications of Cryptographic Techniques*, pp. 181-197, 2008.

[23] G. Bertoni, J. Daemen, M. Peeters and G. Van Assche, "Sponge Functions", *Proceedings of Ecrypt Hash Workshop*, 2007.

[24] C.J. Ezeofor and A.G. Ulasi, "Analysis of Network Data Encryption & Decryption Techniques in Communication Systems" *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 3, No. 12, pp. 17797-17807, 2014.