Combined Enhanced LWSE Algorithm with OTP Algorithm for Secure Database

Kaladhar Ganta, Bing Zhou
Department of Computer Science
Sam Houston State University
Huntsville, Texas, USA
{kxg034@shsu.edu, bxz003@shsu.edu}

***Abstract*—** Data encryption is the process in which data is encoded using various encryption algorithms so that only authorized entities will be able to view and understand the data. Various encryption algorithms such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple DES, and Light Weight Symmetric Encryption Algorithm (LWSE) are being used for the data encryption, with each algorithm having its own merits and demerits. The proposed methodology concerns with the two major demerits of LWSE Algorithm which are the data size and allowed characters in the dataset. Un-like LWSE algorithm, the proposed methodology works fine for variable data size and can encrypt all the types of special characters. Instead of using the same key for encryption in LWSE, the proposed methodology uses different keys that are generated using a random key generator. This enhanced LWSE algorithm is combined with OTP algorithm to add another layer of security.

***Index Terms*— Decryption, Encryption, Folding, OTP, Protection, Transposition, Security, Shifting, Substitution,**

I. INTRODUCTION

Data whether it could be small or big, is maintained in a database for easy retrieval and access when in need at any time. These are designed for sharing data efficiently among different users. Security is the primary concern to maintain a friendly environment and prevent losing of sensitive data. Once data is compromised then it leads to many data leaks causing financial and personal loss. There-fore, database systems must be maintained with high levels of security to avoid any leaks and loss in the future. In some cases, the data stolen from a database by the employees of the firm and the database administrators goes undetected. Owing to these threats every firm needs to employ its own data security mechanisms, to protect the data from either intruder or any malicious insider.

In recent years, intruder attacks are in high number due to the access levels/privileges given to the users in an organization. To avoid security attacks, the one stop solution is the encryption technique; promising a minimal loss to the sensitive data and is considered as the most cost-effective technique holding a huge amount of data. Database Encryption [3] is the one mechanism which can prevent data from being misused in case a malicious attacker gets a hand on it as the data will be in encrypted format and not human understandable, which will be of no use. Various Encryption algorithms are being used by different organizations for encrypting the data to be stored in the database. The safety of the data is ensured by the complexity of the encryption algorithms.

Many efficient researches exist in this area. The most used encryption algorithms are Data Encryption Standard (DES) [4], and advanced Encryption Standard (AES) [2] etc. But the hunt for effective approach never ends; as technology advances the ways to break in is made possible through the exposure of the data. Like-wise an encryption technique known as Light weight symmetric (LWS) algorithm is brought in to light with its vast range of features for its complexity and dynamic nature to any type of data and size [2]. But it lacks certain features due to the advancement of security and availability levels of data.

This paper focuses on the demerits of the LWS algorithm and implements an enhanced LWS algorithm by merging it with the One Time Pad (OTP) algorithm [4]. The improvements followed are firstly, managing the size of data dynamically depending on the input; secondly, usage of all special characters as data varies for different firms; thirdly, adding the feature of confusion by using randomly generated keys to encrypt and decrypt the data and finally merging the resultant output with the OTP algorithm there by creating an additional security layer.

The rest of the paper is organized as follows. Section 2 defines the literature review of the existing approaches. Methodology is explained in Section 3. Section 4 shows the experimental setup and results, and section 5 concludes the paper.

II. DATA ENCRYPTION METHODS

The key to safeguard data from any type of intruders is by Encryption, the process of changing the readable text to unreadable. The approach to validate the data is through decryption which is done by using a key, given to the trusted client, the owner of the data.

Security is one of the challenging concerns of the organization as it leads to financial loss and also its earned reputation. Initially to safeguard the data in databases cryptography came into existence with convincing features, one of them is mixed cryptography database (MCDB) [3]. Here in [3] approach keys are used in different forms to encrypt the data. The earliest use was in Diffie-Hellman key exchange approach which uses a symmetric key for its

communication. An implementation to public key cryptography was used by Rivest Shamir and Adleman algorithm (RSA), in mobile nodes, which are more susceptible to attacks [7]. To confirm the validation and integrity of the received message by the client/receiver, one of the famous technique used is Digital Signature [8]. There are mainly three forms of encryption- Symmetric, Asymmetric and Hash Functions which are developed to provide security to data. It was concluded that there is not a single technique or algorithm which is efficient enough to be carried forward in case of security threats; so researches have been proposing different approaches/solutions to safeguard the integrity of confidential data from being misused. Chinese Remainder Theorem (CRT) [6] is used for performing experiments on the records in the database. Column wise encryption is performed on database through indexing [2].

The optimization ways differ in many algorithms (TSFS, DES, AES) according to the techniques or security mechanism enforcement; due to this there are many pros and cons. Based on the technique used, the database performance can be effected for every query that is to be run [9]. With the use of fixed field type, Kaur et.al, [5] proposed Numeric data encryption. Later Agarwal et.al, [13] enhanced the encryption of [5] by imposing the required operations to be performed on particular sets of encrypted data. Many applications so far in industry use the DES algorithm which is considered as insecure [4], and was replaced with AES. Manjvannan and Sujarani [2] proposed an algorithm which reduces the processing time carried on the encryption and decryption operations, known as TSFS.

Later an enhanced TSFS is proposed which is an extended work on TSFS which can encrypt the data that contains alphanumeric and few special characters ensuring high level of security to encrypted data but imposes few constraints on the data size and the special characters used [2].

III. METHODOLOGY

The proposed methodology is the improvement of the existing LWS algorithm also known as TSFS algorithm which is an accumulation of four phases with few enhancements in its implementation to improve the security. The four phases include- Transposition, Substitution, Folding and Shifting. Additionally, the proposed methodology includes merging the output of the existing TSFS algorithm to OTP algorithm to improvise the security level. The illustration of algorithm and its augmentation is as given below:

A. LWS:

The existing ETSFS algorithm includes the following alphanumeric characters and a few of special symbols (*, -, /, :, @ and _) only used to encrypt the data that is taken as input. But in the proposed methodology it included almost all the special characters. It is a symmetric encryption algorithm which can be inversed that cancels the encryption.

The proposed ELWS algorithm accepts different data sizes and transform the data in to a nearest square matrix and replaces the remaining left out index spaces of matrix with asterisk (*). For example, if an input is 14 characters in size then the algorithm opts for a 4X4 matrix which has 16 characters' size. As, the input is 14 in size the remaining two indexes in matrix are occupied by *'s to make it complete and now the algorithm is carried out on this input matrix. The phases of the existing LWS algorithm is listed below:

1. *Transposition:*

   The given input after transforming it into a complete square matrix is carried out for the first phase of encryption algorithm. Here in this phase a zigzag pattern is performed on the input matrix; initially starting from the top most left corner index to the end of the matrix laterally followed by a diagonal transition continuously representing a pattern. It is illustrated in the figure below for input TOK47LSH2XD. The output data is obtained by retaining the first element in the matrix.

INPUT **OUTPUT**

T	O	K	4
7	L	S	H
2	X	D	*
*	*	*	*

T	O	7	2
L	K	4	S
X	*	*	D
H	*	*	*

The transposition in the given input is performed such that zigzag pattern (TO72K4..***) is followed by starting at the initial index of the matrix which is “T” in the data matrix and continued by following a pattern till the end of matrix. The output data matrix will be the insertion of elements followed as it is in zigzag pattern. This is given as input to the substitution phase.

2. *Substitution:*

The output from the transposition phase is piped to the second phase called substitution, where the keys K1, K2 and the modulus M is used where M referred to the size of the arrays. The three arrays included in this algorithm are the alphabet, number and special characters. For example, if the substitution is to be performed on a character present in specified index of the matrix then it looks out for the character array and takes M=26 and performs the substitution, typically applying the following encryption equation based on the index value and its array value (M). The encryption of the index value is given by:

$$\mathbf{E(x) = (((K1+p) \bmod M + K2) \bmod M} \quad (1)$$

The encryption is performed on each element of the data matrix by applying the function in (1). In the existing LWS algorithm the keys K1, K2 are used from [1]. But the proposed methodology uses the random generator to generate the keys K1 and K2, which are always different for every use of algorithm.

INPUT **OUTPUT**

T	O	7	2
L	K	4	S
X	*	*	D
H	*	*	*

E	Z	3	3
V	L	8	I
F	+	-	I
R	(	.	)

This creates a confusion and provides a high level of security. For example: the first index value of the data matrix is a character and is applied to the function in (1) where p=0 as the element considered is in index 0. For the proposed algorithm K1 and K2 are generated randomly but for existing algorithm it is taken from [1]. The arrays used here are the alphabets ranging from (0-25), numbers (0-9) and special characters (0-7) (in case of existing algorithm) but 0-29 in case of proposed algorithm. Thus the input data matrix is encrypted using (1).

A reverse operation can also be performed to decrypt the encoded data using the same keys and the decryption of encrypted data is:

$$\mathbf{D(E(x)) = (((E(x) - K2) \bmod M) - K1) \bmod M} \quad (2)$$

The above figure shows the example of encryption of data matrix with the given keys (keys of proposed algorithm). And their decryption is shown as by applying D(E(x)) shown in (2). The encrypted text can be verified/validated when decryption is applied shown in equation (2).

3. *Folding:*

The output of the substitution phase is given as input to folding phase. Here in this phase the first row is replaced with the last row similarly the columns of the data matrix. The remaining left out elements are diagonally replaced within them. The approach differs for a odd square matrix say 5X5, the folding is performed for the second column and rows flipping with the fourth column and row by leaving only one element without exchange. So the center element is left as it is. The below figures illustrates the folding output for a given output. Later the output is passed as input to shifting phase. The resulting output is passed to shifting.

INPUT **OUTPUT**

E	Z	3	3
V	L	8	I
F	+	-	I
R	(	.	)

(	(	.	R
I	-	+	V
I	8	L	F
3	Z	3	E

4. *Shifting:*

This is the last phase of the algorithm where the input data matrix elements are shifted based on the respective arrays. The character's array ranging from (0-25), number elements array (0-9) and special characters' array (0-29) are considered for shifting the elements based on their position.

For example: if the element in the input at index 8 is a character say 'K', then from the character array the 'K' alphabet should be counted from 1 till it reaches 8th positioned element which is 'S' in character array. Similarly, for number array and special characters' array.

INPUT **OUTPUT**

(	(	.	R
I	-	+	V
I	8	L	F
3	Z	3	E

(	)	!	U
M	#	"	C
Q	7	V	Q
5	M	7	T

B. Improvement of LWS:

The existing algorithm has few constraints and restricted to limited data sizes and special characters' size (limited to only 7 characters). Enhancements are made in order to improve the efficiency and take advantage of the limitations. Here are the few enhancements made to the existing algorithm:

a. ASCII codes are used for not only special characters but also for all characters and numbers present in a data matrix, in order to improve the level of security to avoid leaks.
b. Data size is not limited; it can support any size of input data dynamically. Based on the analysis of available data size for encryption, the data matrix is determined such that only a square matrix is considered (the maximum size of the matrix is assumed to be 10x10). For example, if the data size is 14, a 3x3 matrix cannot accommodate all the data elements; a very 4x4 matrix is considered.
c. Random generator is used to generate the keys to create confusion. As it produces different keys for each time the algorithm executes.
d. Merging the output with OTP algorithm. It applies an XOR operation on the output obtained from the shifting phase which is an encrypted data which is considered as a final output.

C. Architectural Design of Proposed Methodology:

The design consists of an input which is uploaded and the system performs the respective operations to encrypt the plaintext to cipher text and later stores the encrypted data into database to avoid any compromised access view of data which might be misused. The system performs the encryption and decryption in four phases as listed above and merges the output of the system with the OTP algorithm for effective results. The OTP algorithm is nothing but applying the XOR operation with the key K1 which is generated by random generator.

The shifting output of the proposed methodology is given as input to the OTP algorithm. The following figure illustrates the finalized output of the encryption algorithm.

INPUT			
(	)	!	U
M	#	"	C
Q	7	V	Q
5	M	7	T

OUTPUT			
,	/	!	]
E	#	&	K
S	7	V	R
=	I	4	]

The decryption is similar to the encryption but the inverse of every phase followed in encryption with the inverse OTP algorithm. The architectural design of the encryption is shown below:

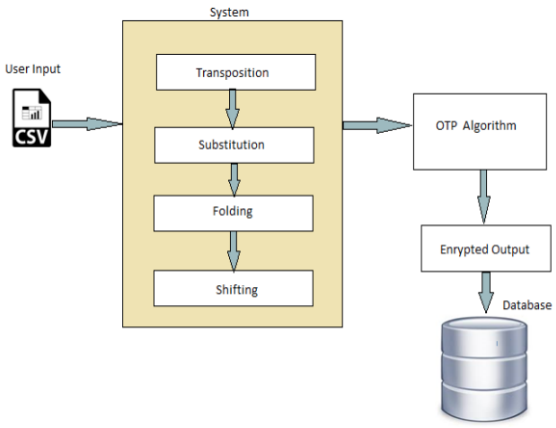

Fig.1: Architectural Design of Encryption Algorithm

Fig.1 illustrates the flow of the encryption algorithm. The web application is designed for the whole process where the user is prompted to encrypt a single data set or encrypt a file. If the user is prompts to select for encrypting a file; the file is selected from the user stored location. Then the file is given to the system where the four phases of the algorithm are been processed and the output form the system is given as input to the OTP algorithm where an XOR operation is performed with the generated key K1. Thus the finalized encrypted data is obtained and later stored into respective database for future use and retrieval. Thus the file or the single dataset is being encrypted.

To validate the encrypted output, decryption process should fall in place.

The flow of the decryption process is shown in fig.2. The requested data is selected from the database which will be in encrypted format due to the encryption algorithm. The encrypted text is passed through the inverse OTP algorithm and later it is sent as input to the system where the inverse shifting, unfolding, inverse substitution and inverse transposition operations are performed. The output which is obtained from these phases is the finalized decrypted data also known as plain text.

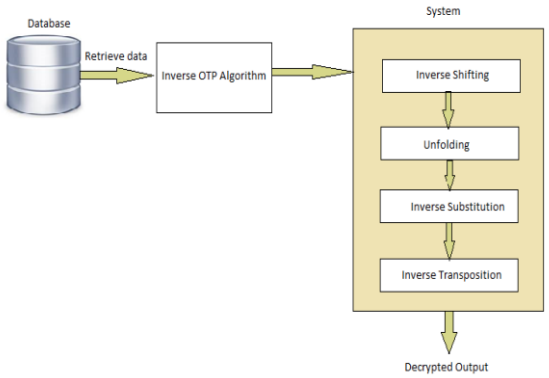

Fig.2: Architectural Design of Decryption Algorithm

IV. Experimental Results

This section provides an overview of the experimental setup; software and hardware requirements, the dataset; size and type of data used as input and the finalized results with the time estimate and performance.

The experiment is carried out for different datasets holding 100, 500, 1000 rows of data. the dataset used contained information of individual profile working in different companies. The confidential data in this dataset included the email id, Social Security Number(SSN), passwords are encrypted by the proposed algorithm to avoid unintentional misuse of data. Later the encrypted data is stored in database for easy and fast retrieval. The dataset is converted to a CSV file and provided as input to the algorithm to perform the encryption process. The pro-posed algorithm supports for a single field of data and for multiple fields of different types of data.

Fig.3 and Fig. 4 show the Execution time for encrypting same data with the AES, DES and ELWSA with OTP algorithms. Encryption and decryption time for different number of tuples is calculated and plotted as a graph. When the number of tuples being encrypted is less the time taken for each algorithm to encrypt the data is closer. As the number of tuples to encrypt increases the difference in execution time for each algorithm is clearly observed. The enhancement achieved robust and effective results by overcoming the demerits of TSFS algorithm and moving it to next level of security by merging it with OTP algorithm.

The performance of the system can be further enhanced using threads. While using the proposed ELWSEA combined with OTP algorithm, the encryption of each block of data is completely independent of the result from the previously encrypted block. Whereas in AES and DES the encrypted output from the previous block effects the encryption of the current block. This dependence prevents AES and DES from using threads. As the proposed ELWSEA combined with OTP

algorithm is independent, we can split the data to be encrypted into multiple parts and assign the encrypting of each part to a different thread. This increases the efficiency of the algorithm wherein we can use multiprocessing, a feature which we can't use in AES and DES. This feature can be implemented as a future extension to this Algorithm which would improvise the efficiency of the encryption and decryption of the proposed ELWSEA combined with OTP algorithm.

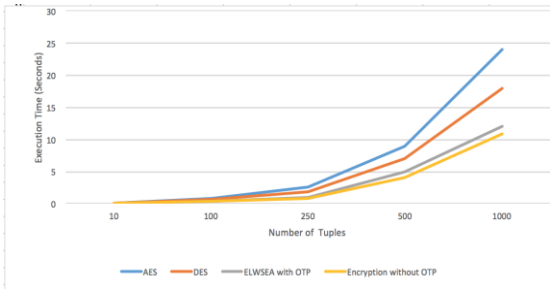

Fig.3: The relation between execution time and number of tuples encrypted for three algorithms along with ELWSEA without OTP

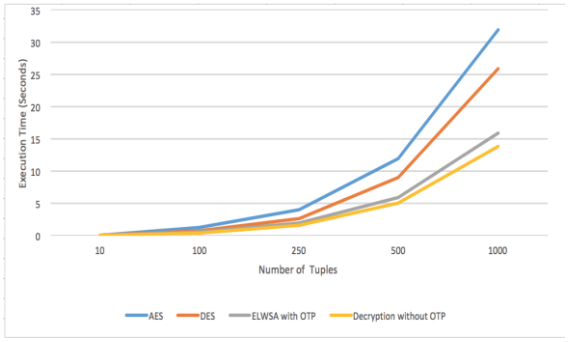

Fig.3: The relation between execution time and number of tuples decrypted for three algorithms along with ELWSEA without OTP

V. CONCLUSIONS

Data security is the primary concern and a challenging issue in today's technology as there is growing increase in new equipment's and tools. Even the attackers are gaining a wide range of information on new trends and implying them in practice. After a consolidated study and research to maintain the security of the data; researches proved that the effective measure is to encrypt the data before storing it into database minimizing the risks due to compromised access and security leaks. In this paper, the idea of enhancing the existing LWSE algorithm is used to propose a new methodology by enhancing its features to a new level. The pro-posed implementation mainly focused on providing a dynamic nature for the existing algorithm by providing different data varied on type and its size; which is a requested demand in today's advancements. The other important feature is the use of the random generator for the keys to encrypt and decrypt the data as it is very difficult for a human to break the keys generated as it changes every single time the algorithm is in use. Finally, merging the output with the OTP algorithm bringing the security standards to next level to maintain integrity.

REFERENCES

[1] Hanan A, Abeer, Heba, "Lightweight Symmetric Encryption Algorithm for Secure Database," IJACSA International Journal of Advanced Computer Science and Applications Special Issue on Extended Papers from Science and Information Conference 2013, page(s): 53-62, Saudi Arabia.

[2] Manivannan, R. Sujarani, Light weight and secure database encryption using TSFS algorithm, Proceedings of the International Conference on Computing Communication and Networking Technologies, 2010, pp. 1-7.

[3] Kadhem, H.; Amagasa, T.; Kitagawa, H.; A Novel Framework for Database Security based on Mixed Cryptography; Internet and Web Applications and services, 2009. ICIW '09. Fourth International Conference on; Publication Year: 2009, Page(s): 163 –170D.

[4] I. Widiasari, "Combining advanced encryption standard (AES) and onetime pad (OPT) encryption for data security", The International Journal of Computer Applications (2012)..

[5] Kamaljit Kaur, K.S Dhindsa and Ghanaya Singh, "Numeric to numeric encryption: using 3KDEC algorithm", IEEE International Conference on Advance Computing, (2009).

[6] G. I. Davida, D. L. Wells, and J. B. Kam. A database encryption system with sub keys. ACM Trans. Database Syst., 6(2):312–328, 1981

[7] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," ACM, Vol. 47, No.653.

[8] Vocal,"http://www.vocal.com/cryptography/dsadigital-signature- algorithm/," Dated: 13- dec-2012 at13:18.

[9] Cohen W, Ravi Kumar P and Fienberg S. "a comparison of string distance metrics for name-matching tasks." In: Proceedings of IJCAI-03 Workshop of Information Integration, 2003.

[10] Ali Makhmali, Hajar Mat Jani; Comparative Study on Encryption Algorithms and Proposing a Data Management Structure. International Journal of Scientific & Technology Research Volume 2, Issue 6, June 2013 ISSN 2277-8616

[11] Noor Habibah Arshad, Saharbudin Naim Tahir Shah, Azlinah Mohamed, Abdul Manaf Mamat," The Design and Implementation of Database Encryption", International Journal Of Applied Mathematics And Informatics, Issue 3, Volume 1, 2007

[12] Z. Yang, S. Sesay, J. Chen, D. Xu, A Secure Database Encryption Scheme, Proceedings of

Consumer Communications and Networking Conference, 2005.

[13] A. Rakesh, K. Jerry, S. Ramakrishna, Order preserving encryption for numeric data, Proceedings of ACM SIGMOD International Conference on Management of Data, 2004, pp.563-574.