Data security using random dynamic salting and AES based on master-slave keys for Iraqi dam management system

Hussam J. Ali[1], Talib M. Jawad[2], Hiba Zuhair[3]
[1]Iraqi Commission for Computers and Informatics, Informatics Institute for Postgraduate Studies, Iraq
[2]Ashur University College, Baghdad, Iraq
[3]College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

Article Info

Article history:

Received Feb 20, 2021
Revised Jul 4, 2021
Accepted Jul 7, 2021

Keywords:

AES
Data security
DMS
Dynamic salt
Multi-tier secure model

ABSTRACT

In the present time, dam management is considered one of the important challenges for e-government in Iraq, becuase it needs information technology infrastructure, data integrity, and protection of user privacy against Internet threats that render such vital infrastructure ineffective. This struggle between the proposed dam management system (DMS) and a multi-tier secure model specifically for the Fallujah dam (and generally for all dams) which is addressed in this paper as a case study. To do this, a relational database design will discuss the development of a multi-tier secure model for integration of the dam management framework with its functions. This paper will discusse encryption and decryption of the dam data using the advanced encryption standard (AES) algorithm with derived keys via PBKDF2 and RNG sequences generator and Slave key for salting protection. The experimental results and analysis on the speed of encryption/decryption process, entropy value, plain text sensitivity, key sensitivity, keyspace analysis, and histogram analysis will prove the the proposed scheme can impede the known attacks like brute force attacks, statistical, and differential. Thus, the encryption scheme can be implemented on the proposed DMS and any other information system, as the implementation which will be presented in the results.

Corresponding Author:

Hussam J. Ali
Department of Computer Science
Informatics Institute for Postgraduate Studies
Baghdad, Iraq
Email: hussam.technology2018@gmail.com

1. INTRODUCTION

Internet threats infect approximately all e-government systems in many countries [1]. These threats affect the infrastructure of institutions such as dam stations in Iraq, which lack data protection. Moreover, most of dam employees are unaware of internet threats by useing traditional means such as social media applications and e-mails which cause the problem of exchanging wrong information between dam stations and the Iraqi National Water Management Center [2]. In addition to the misuse and unauthorized useage of data acquisition tools by terrorists may lead to unreliable sources of dam readings, loss of data online, data breaches, and wrong decision-making by managers of the Iraqi National Center of Water Management [2].

So, the aforementioned cases may cause the flow of water in large quantities and destroy the control of the dam gates that lead to a flood [3] which needs great efforts to be taken for this potential disaster. However, the Iraqi government is still failing to implement information technology and infrastructure for these important institutions and organizations, in addition to weak mutual data protection and user privacy [4], which in turn increase the probabilities of unauthorized disclosure, unreliable data sources, short-term history of user records, and water resource archive [2], [5]

To address the aforementioned issues and protect data, this paper contributes as: i) presenting a proposed design for a dam data management system according to a multi-tier secure model for locally application at Fallujah Dam Station/Anbar Governorate/Iraq; ii) implementes the database scheme and data encryption mechanisms.

Accordingly, via observing some recent studies in analyzing the efficiency of the most famous encryption algorithms, AES algorithm is meeting excellent results in comparision with some encryption mechanisms. Researchers in [6], [7] shows that it takes less time to encryption data with less resource consumption than the Twofish algorithm on a 240KB text file, and with another experience, the researcher declares that AES surpassed Serpent in speed and performance [6], which gives speed in encryption and decryption and a high-security rate of data with the use of CBC cipher mode [8].

Through that, the data encryption method will be adopted in this work by applying the AES algorithm and 256-bit size derived key with the suggestion of adding a dynamic random salt method to the ciphertext and the usefulness of the proposed method is to give different encryption result every time with same plaintext, also to easy retriving data encrypted form database and perform this work on proposed Dams data management system, with the implementation of practical experiments linked to CBC cipher mode, in comparison with several algorithms under specific conditions.

2. BACKGROUND

This section reviews in the first subsection several of the related works to dams' data management, and the traditional mechanisms and their effectiveness, beside the methods that have been reviewed in encrypting the data. The second subsection investigates the case study of Iraqi dams (Fallujah dam), identifying the current deficiencies in dam management in Iraq, and carrying out the investigation together to redefine the adequate technical and security mechanisms necessary to achieve integration and information security.

2.1. Related works

Researchers in [9] have developed a CLPIMS China dam project information management system, which is based on WebGIS and relies on a multi-tier structure in browser/server mode to enter data for dam managers with SSL encryption. The system is based on a combination of SQL Server database and storage model ArcSDE data for rapid storage of large amount of complex data types obtained through the application of Active Server Pages technology to achieve comprehensive central management. However, with this important work, the developers did not focus on issues of securing user data protection, database encryption.

Also, a web-based application has developed by authors in [10] a with a relational database for real-time dam management under the conditions of weather, river flow, and water temperature. The designed system has predicted the forecasts of water temperature and rainfall by using a data mining approach in real-time application. However, no security parameters have been considered to protect the data center.

In this paper [11], a web-based client-server system is proposed, in which the data is encrypted using AES with 256 keys, the research solved the problem of organizations losing their sensitive data to unauthorized people. So, the authors will propose a hybrid encryption pattern to support a private key cipher system that is a combination of AEC and ECC with 192-bit key size and 12 rounds, thus increasing the overall security of the system by implementing software-based countermeasures to prevent possible vulnerabilities that it poses Timing side-channel attack [12]. This paper [13] presentes data encryption using the AES algorithm and a 256-bit key with mixing salt values resulting from the dynamic generation and merging them with the plain text before the encryption process, as shown in Figure 1, where an linear congruential generator (LCG) is used to generate random values. In the practical experiment, the proposed method is used with a maximum of 30 characters, and a larger size of the values is not applied. On the other hand, the researcher does not explain the method of storing salt values when encrypting large data and storing them in databases, and the method for retrieving salt values and decrypt data for any row in the database later.

2.2. Case study: Fallujah dam

Through the pilot study, will shed light on the current situation of the Fallujah dam on the Euphrates River, which is located in Anbar Governorate, to clarify the technical, organizational, and security challenges. It is observed that 45 (managers, departments managers, engineers, and operators) of Fallujah dam station are interviewed and acquired to fill out the questionnaires. They face many technical, security, and organizational challenges as summarized in Table 1. Where the number 1 represents strongly disagree, 2: disagree, 3: medium, 4: agree, 5: strongly agree. Questionnaires and responses which are addressed refer that the dam management lacks information systems infrastructure, insecure data transfer through traditional means (social media), and weak data integration and retrieval. Moreover, Fallujah dam workers and managers have supported the use of IT systems, especially in relation to data management requirements, and the solving of these issues.

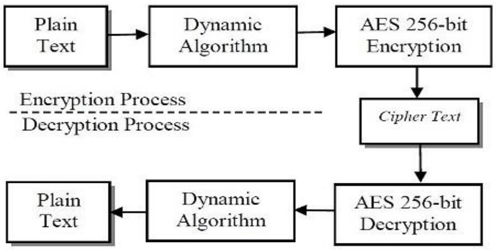


Figure 1. General system design dynamic encryption

Table 1. Summary of highlights addressed via the pilot study in Fallujah dam

Issues	Question	1	2	3	4	5
Technical issues	Is a special information system used to transfer data between the dam and the center?	☑				
	Is social media being used to send dam data from the operator to the manager?				☑	
	Is social media being used to send dam data from the manager to the center?				☑	
	Is the method of sending dam data from the operator accurate?			☑		
	Is the process of reading data and sending it in an accurate time?			☑		
	Is there a mistake in reading the received telegraphs from NC?			☑		
Security issues	Are telegraphs received from the NC through a special information system?	☑				
	Are telegraphs received securely by regular means?		☑			
	The telegraphs reaches the project manager in secret without disclosing it from the rest of the project managers?		☑			
	Is the safety factor available and the privacy of the currently used work mechanism?		☑			
Organizational issues	Do you support the use of information technology in government departments?					☑
	Do information systems facilitate tasks for the organization and the employee?				☑	
	Do information systems help in analyzing big data on dams?				☑	
	Can technology help organizations make error-free, sound decisions?				☑	

3. OVERVIEW OF CRYPTOGRAPHY

Cryptography is defined by William Stallings as “the type of operations used for transforming plaintext to ciphertext, the number of keys used, and the way in which the plaintext is processed” [14]. The types of encryption algorithms are divided into several groups [15], as shown in the Figure 2.

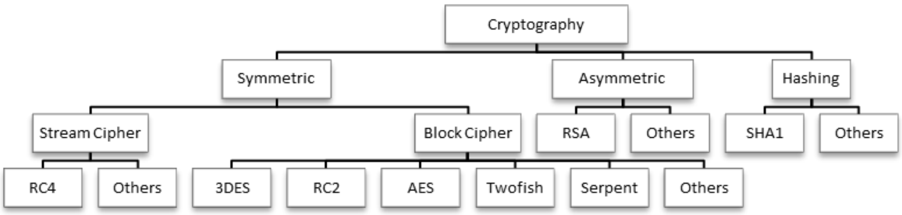


Figure 2. Cryptosystem’s classification

3.1. Symmetric block ciphers

Block cipher converts plaintext to a fixed length size, known as the block size. If the length of the plain text is greater than the specified block size, it is divided into several equal blocks. A block cipher is based primarily on a symmetric key. The same secret key is used for encryption and decryption [7]. There are several symmetric block cipher algorithms, TDES, RC2, AES, Twofish, Serpent. These algorithms work in different operating modes in encrypting each block, the most famous of which are: Cipher block chaining (CBC), electronic code book (ECB), and cipher feedback (CFB) In this research will adopt the CBC operating mode with AES because it has a high-performance speed compared to with CFB [8], each block is also uniquely encrypted [16], on the other hand, it is safer than ECB [17].

3.1.1. Triple DES (TDES)

TDES is developed to solve apparent weaknesses in DES without building a fully new cryptosystem. The concept of TDES is based on extending the size of the DES key three times by executing the algorithm in series with three different keys [18]. The length of one key is 56 bits and the sum of the length of the three keys is 168 bits, The goal of developing TDES is to ward off attacks targeting DES by giving a relatively simple way to increase the key size [19].

3.1.2. Rivest cipher (RC2)

The RC2 is developed by Ron Revist 1987 to replace DES [14]. The data block size of RC2 is 64 bits, while the key size ranges from 40 to 1024 bits (5-128 byte). The key size gradually increases by 8 bits from the previous size. On the other hand, messages encrypted in RC2 can be broken in short time by using a brute force attack, so it is considered unsecured. In RC2 The key is compound from an initialization vector (IV) and KEY [14].

3.1.3. Advanced encryption standerd (AES)

The standard (AES) is based on a symmetric block cipher defined by the National Institute of Standards and Technology (NIST) in 2001 [8]. It supports block size 128 bits of data, and it handles three sizes of keys 128, 192, 256 bits [20]. The maximum number of rounds processed by AES is 14, and the number of rounds depends on the size of the key used, where the key size 128 corresponds to 10 rounds that are processed, while 192 bits correspond to 12 rounds, finally 256 corresponds to 14 rounds [7].

3.1.4. Twofish

The algorithm of the two fish is purchased in 1998 by Bruce Schneier. It depends on the type of block cipher and uses keys with sizes 128, 192, and 256 as a maximum, and the data block is 128 bits, and it is considered one of the five final contenders with Rijndael, but does not reach the first level and is considered almost complex, except It works with different encryption modes [21].

3.1.5. Serpent

The serpent algorithm is a block cipher. It has high security, as it competes with Rijndael's algorithm. This is because Serpent's algorithm was competing with AES, but its slow speed caused it not to be adopted, so Rijndael was chosen as AES. Although there are many applications of the serpent algorithm; As an example, images are encrypted with it [22].

3.2. Hash functions

It is also known as one-way encryption or message digests, hashing is used to produce a unique, fixed-size value to represent a variable amount of data, and no need for a key, as the reference represents the key itself. They are widely used in storing passwords, digital signatures, as they work to match the hash values of two sets of references and if they match, the result is true [23].

One of the common hashing methods in cryptography is known as SHA-1, which produces hash values of 160 bits [24] equivalent to 40 digits long as hexadecimal (20 bytes) [25], that published in FIPS 180-1 [26]. Another type of hash algorithm which will be used in this work to derive keys is HMAC-SHA-512 and used as an HMAC or hash-based message authentication code [27]. The HMAC process mixes message data with the secret key, hashes the output with the hash function, and repeat mixes the output hash value with the secret key, and then implements the hash function again [28]. The output length of the hash value is 512 bits. HMAC-SHA-512 is a better choice to achieve security and to ensure data confidentiality and perfection of the security algorithm [29].

3.3. Random numbers generator (RNG)

Random number generator (RNG) along with encryption algorithm is the basis of cryptography systems [30]. So it is considered that no matter how complex encryption algorithms are implemented, they become weak without the random number generator present in the root of this system and also to say that the RNG is the basis of cybersecurity as a whole. Therfore, it is extremely important to ensure the quality of a mechanism for generating completely random numbers. This research will use the RNG implemented by the cryptographic service provider (CSP) library which generates random numbers with high quality and no way to predict or reproduce key sequence [31].

3.4. PBKDF2

Password-based key derivation functionality (PBKDF2) version 2 that Implemented through Rfc2898 Derive Bytes [31], by using a random number generator based on HMAC-SHA-512. To provide effective protection against brute-force attacks, PBKDF2 proffers CPU-intensive operations. The basis of these operations are based on the iterated of a pseudo-random numbers generator function that assigns input values to a derived key [32]. Figure 3, shows the three main inputs of PBKDF2 algorithm.

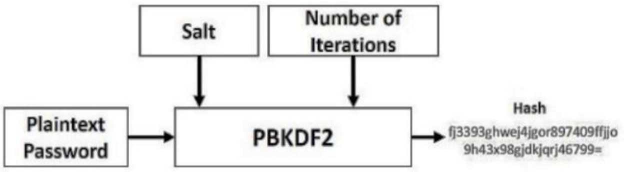


Figure 3. Password-based key derivation functionality

4. METHOD AND MATERIAL

From the aforementioned technical and security highlights, two research questions are posed to solve: “How does the management of Fallujah dam improve technically toward an effective dam management system?”, And “How can the data security of this dam management system be sufficiently enriched? This section will answer these two questions by presenting the design and implementation of a dam management system and recommending a multi-tier secure model to maintain data integrity by encrypting data via AES algorithm with random generating of the dynamic salt value, and this algorithm is chosen for their good performance after an analysis of practical comparison of several algorithms using Cryptool 1.4.4 and Chilkat libraries, before and after merging dynamic salt process.

4.1. Design of dam management system

The schema which is suggested for DMS in this paper, includes the following components: user information, dam data, real-time dam hydrological data, telegraphs, water history data, and employee movement history, see Figure 4.

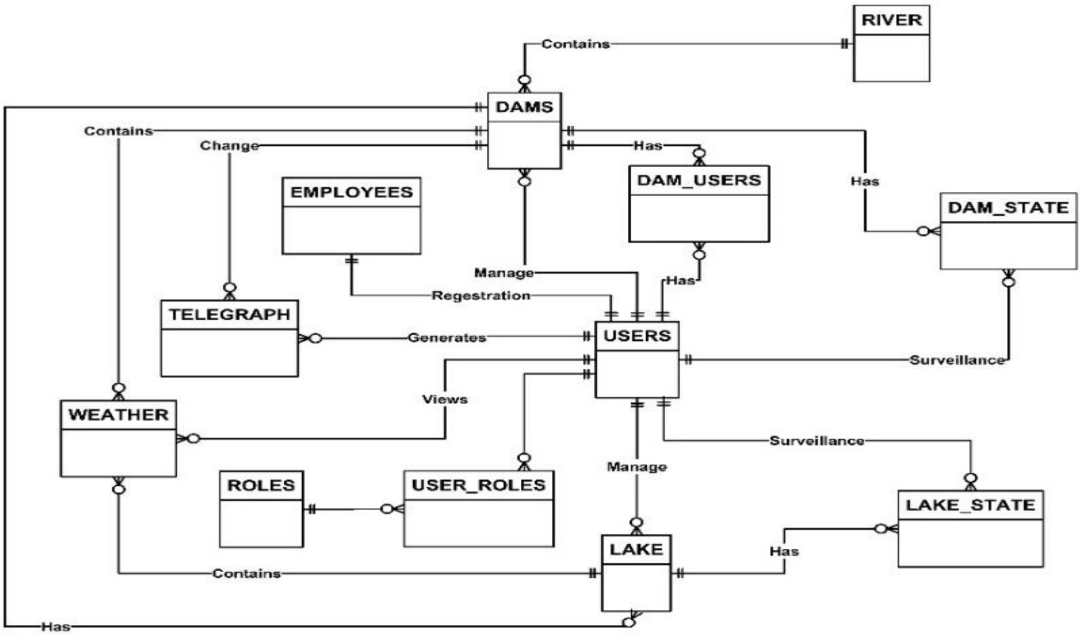


Figure 4. Conceptual design of dam relational schema

4.2. Multi-Tier secure model

To meet all security issues of data integrity and users privacy protection, it is highly recommended to consider the following multi-tier secure model proposed as shown in Figure 5, with the components of model.

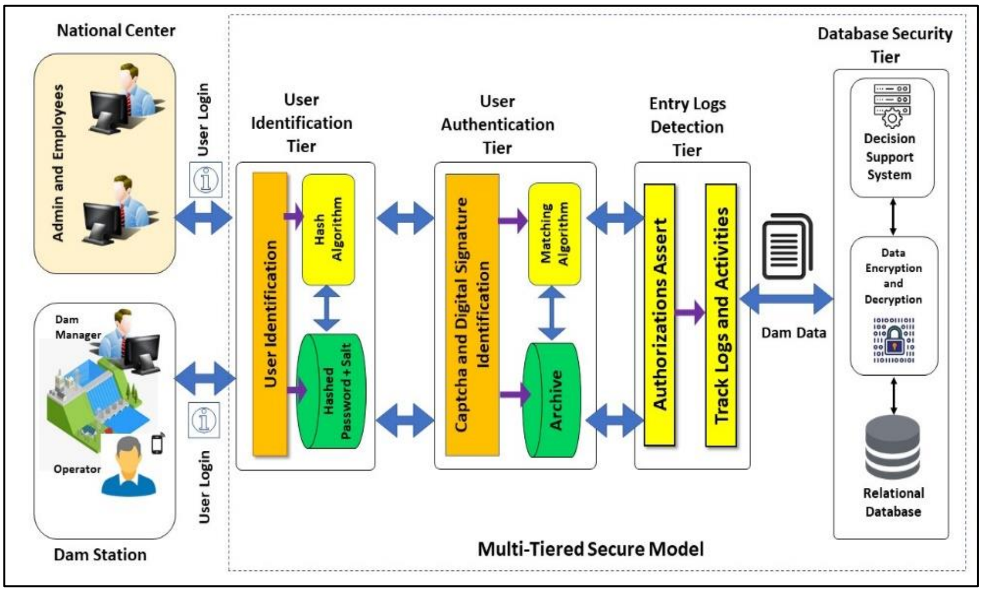


Figure 5. The multi-tier secure model for dam management system

4.3. Encryption/decryption proposed scheme

This section explains the main stages of the proposed model for the encryption and decryption process of the text, in the first stage the parameters and basic values of the salt size are initialized, the initial primary key string is initialized from the password generated through RNG, the next stage is the process of generating the salt value also through RNG, after that the stage of deriving the encryption key through PBKDF2, then comes the stage of encryption the text through AES, and finally the stage of generating the secondary key and protecting the salt value through XOR Proccess. Figure 6 illustrates the stages of the encryption and decryption process in detail. The next subsections describe these stages.

4.3.1. Initialization basic parameters and values stage

In the proposed model, *p* denotes for plain text, *C* is the ciphertext, *S* is the salt value, while SK is the secondary key. This stage is implemented through the following steps:
- Step 1: Assign the password string.
- Step 2: Initialize S length to a fixed size (32 bytes).
- Step 3: Convert plaintext to Ascii and then to decimal and set length p byte from the plaintext length.
- Step 4: Initialize values of *p* byte [0- (i-1)] from the decimal valus produced in *step 3*.

4.3.2. Generating the salt values through RNG crypto service provider stage

The random salt chain achieved through the following steps:
- Step 1: Set length of salt [I]=32 from S length that initialized from the first stage.
- Step 2: RNG Crypto algorithm implements under *salt[I]* length defined.

4.3.3. Deriving the encryption key through PBKDF2

The PBKDF2 algorithm's robustness and efficiency in deriving an unpredictable cipher key depend on the fragmentation of the elementary elements and the execution of the HMAC-SHA-512 hash algorithm with a number of rounds ranging from 1 to more than 10,000 rounds. In this paper, only 8 rounds are applied due to less time consumption and at the same time, to obtain a completely secure hash key. The following steps explain the implementation of this stage for the first round:
- Step 1: Taking a random salt, and flip bits, giving K1.
- Step 2: Calculating the SHA-512 hash of K1 plus password string, giving H1.
- Step 3: Iterations count = count++.

All steps above repeated until the condition loop equals true, then computing the final hash.

4.3.4. Encryption stage through AES

AES algorithm require configuration initialize vector and cipher mode, after that the encryption process begins as the following steps:

- Step 1: Initialize keyByte () ranged (0-31) from the hash key that produces in stage 3.
- Step 2: Set IV length = 128 /8 => 16 byte
- Step 3: Initialize the value of IVByte (0-15) from the hash key.
- Step 4: Create Encryptor(byte(array) keyByte, byte(array) IVByte).
- Step 5: Set cipher mode = CBC.
- Step 6: Segmentation *p byte(array)* to BlockSize(0-15) array and make transformation process to ciphertext C(array) ranged [k0-(k-1)].

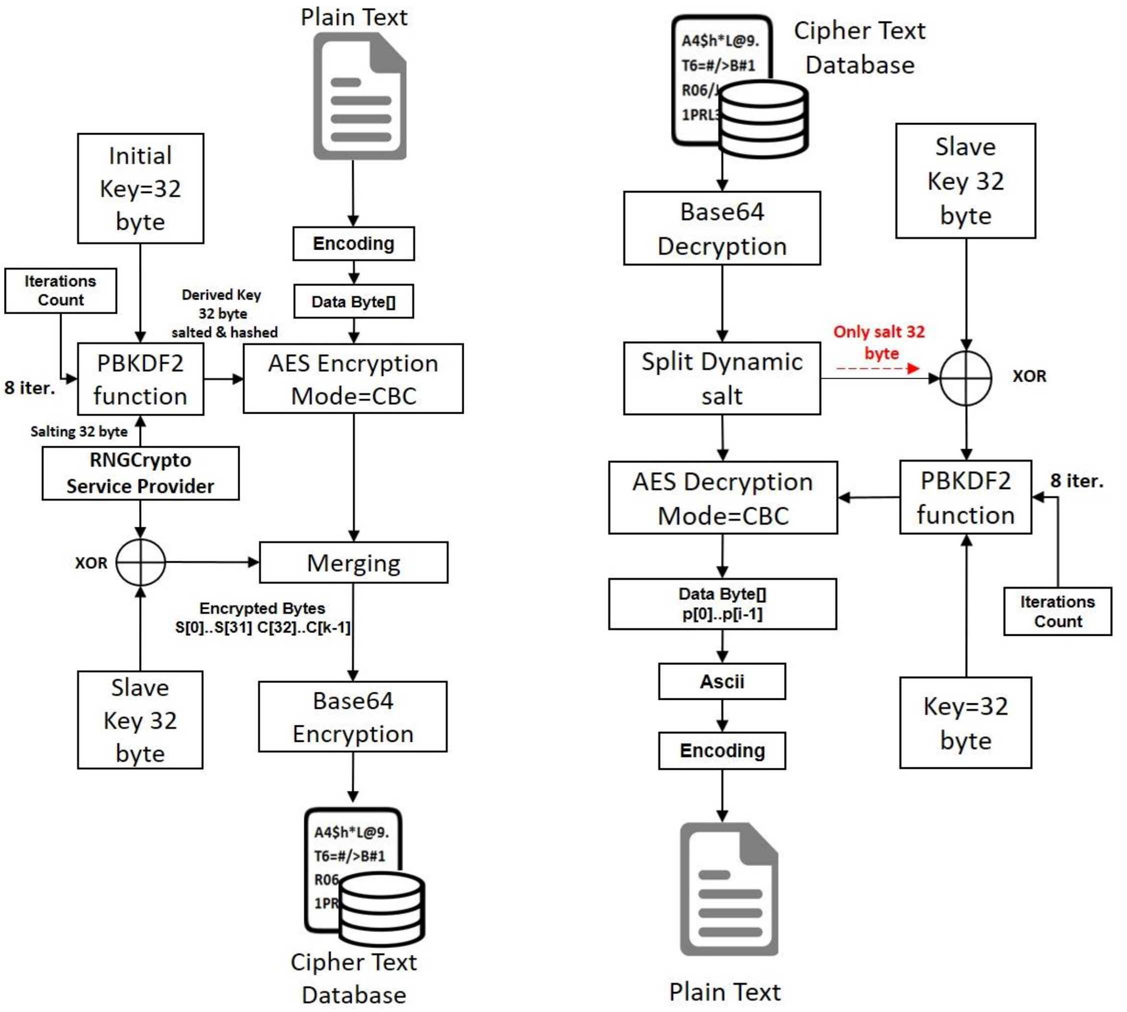


Figure 6. Proposed method of; (a) encryption process and (b) decryption process

4.3.5. Initializing the independent slave key

The benefit of the independent key is to protect the value of the salt, and when retrieving the encrypted data from the database, the value of the salt is retrieved with this key, which was previously generated by RNG independently with a size of 32 bytes, and the following steps explain this stage:

- Step 1: Initializing random salt(array) value that generated via RNG algorithm in stage 2.
- Step 2: Initializing SK (array) values ranged (0-31).
- Step 3: Compute final salt (array) by XORed with SK(array), and appended with ciphertext array, as shown in Figure 7.
- Step 4: The final ciphertext encrypted through the Base64 algorithm.

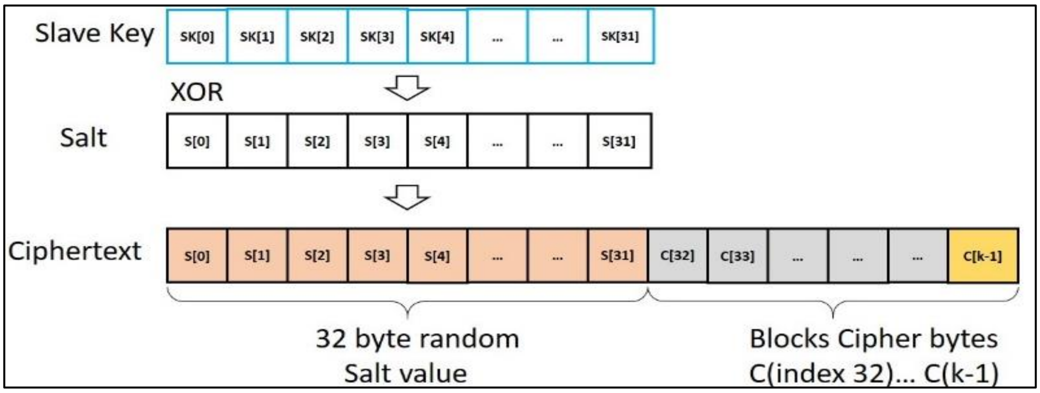


Figure 7. Merging random salt value and ciphertext

5. ENCRYPTION ANALYSIS

This section provides a test of the validity and accuracy of the proposed model in terms of derived keys robustness and encryption quality. On this basis, a computer with an Intel® Core (TM) i5 M460 2.53 GHz x 2 Cores processor was used, and 6G Ram (DDR3) speed 1016 MHz, Windows 10 Pro 64-bit, and programming language used is C# .net core version 7.3 on .Net Core 2.1 framework.

5.1. Key space

Key space in cryptography is the total number of keys that can be generated and used in the cryptography process [33]. The security and strength of the algorithm depend on the length of the key, as the key increases in length; the algorithm becomes immune to brute force attacks [34]. In order to actually achieve this, the key space must be greater than 2^{100} [35] length of the encryption key used in this paper is 256-bits, the key space will be 2^{256} ($1.16\text{x}10^{77}$), and the same way for the secondary key space is 2^{256}, so the final key space is equal:

$$Size\ of\ Key\ space\ =\ 2^{[Master\ key\ size]+[Slave\ key\ size]} \tag{1}$$

$$= 2^{256+256} = 2^{512} \tag{2}$$

From the above equation, the keyspace is sufficiently large to resist brute-force attacks.

5.2. Key sensitivity

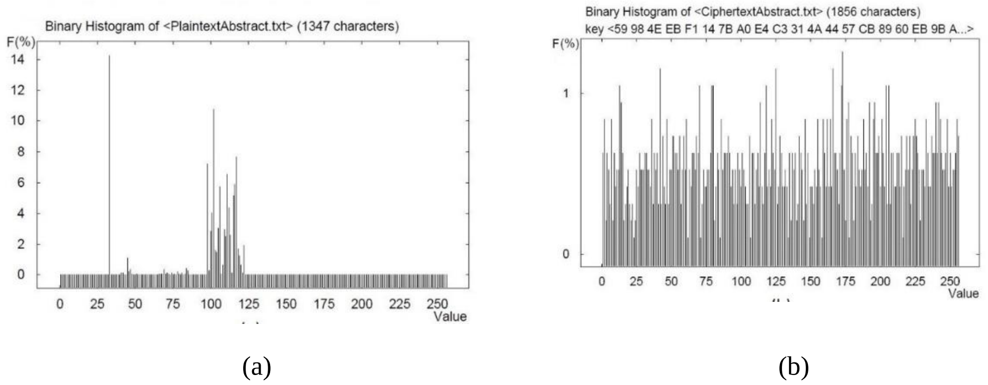
Good encryption is very sensitive to tiny secret key changes [33]. Table 2, displays plaintext with a size of 61 characters, after encryption it with two very little different keys, the ciphertext will appear completely different and there are no data associated with the plaintext, so the proposed encryption model has a high sensitivity to tiny changes in the secret keys.

Table 2. Key sensitivity example

Plaintext	key	Ciphertext
Good encryption is very sensitive to tiny secret key changes	D9D708… …6E89F**3**	W2HAGyYXRVKEjmahOazFRMoxXkb77ImKGTxY3TIzO9VMYUuqEbYfD37Zz km6Hd/BvcOyb7xmQV65hkRAKSLnbA==
	D9D708… …6E89F**4**	at9QY0c6PHrKwTKhODXpOjpL3MEbjBpak6QQfwGtiCPOC8n5dW80KAywCIN 6S6REE1RGJskbbukFbrfJtUGNFw==

5.3. Frequency analysis

Frequency analysis comprehensively describes the encoded text through the histogram, as it shows the number of times a particular symbol appears throughout the text, so this type of attack may provide information about the key or the original text. Figure 8 (a), displays the plaintext consisting of 1,347 characters (the abstract of this paper), while the Figure 8 (b) displays the histogram of the ciphertext and shows all the symbols uniformly, so the proposed model is immune to attacks of frequency analysis.



(a) (b)

Figure 8. Histograms; (a) plain text and (b) cipher text

5.4. Information entropy analysis

Information entropy is the mesurment of uncertainty, in statistical analysis, it is used to measure the secure value of a cipher. It is defined as follows [36]. The *H(m)* is a random of message m with N length, then its entropy is

$$H(m) = \sum_{i=0}^{2^N} P(mi) log_2 \frac{1}{P(mi)} \quad (3)$$

P(m$_i$) denotes the probability of m$_i$. If every symbol has an equal probability, i.e., $m=\{m_0,m_1,\ldots.m_{255}\}$, the maximum entropy is 8 . In this paper, four other recognized algorithms are tested and their results compared, where the entropy value is calculated before and after applying the proposed model. As in the results presented in Table 3, the proposed one gives better entropy value especially with AES, sometimes followed by Twofish and RC2, since AES has a higher key space than RC2, so it is considered effective and safe with the proposed model.

Table 3. Entropy results

Plain Text	Before merging random salting					After merging random salting				
	3DES	RC2	AES	Twofish	Serpent	3DES	RC2	AES	Twofish	Serpent
Plain Text in Table-2	5.67	5.73	5.87	5.76	5.65	6.58	6.56	6.57	6.55	6.48
Abstract as plaintext	7.74	7.83	7.84	7.84	7.82	7.82	7.84	7.86	7.85	7.83

5.5. Speed analysis

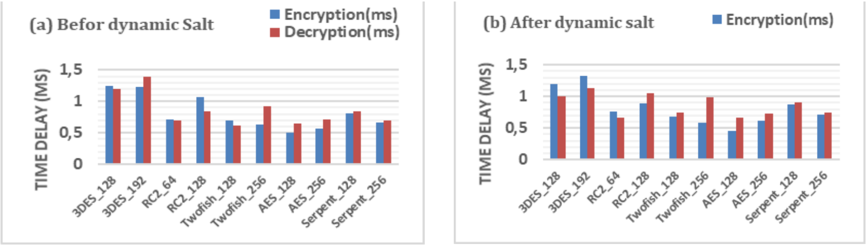
The encryption algorithm that has durability and high performance must also have speed in process with real-time applications. In this paper, as shown in Figure 9, the delay time for each case and several tested algorithms are compared with different key sizes, and a text file containing 6000 characters. Whereas, AES with a 256-bit key size is derived by applaying (8) rounds through the PBKDF2 algorithm to derive the keys, which will achieve good speed performance with the proposed model compared to the rest of the algorithms. Figure 9 (a), shows the results before implementing the dynamic salt merging and the encryption time is 0,56 ms, while the decryption time is 0.72 ms, and after applying the merging, Figure 9 (b), presents the results for encryption time is 0.61, while the decryption time is 0.73 ms. From the foregoing, the proposed model does not affect the speed except to a very small extent:

Encryption= (after merging salt) - (before merging salt) - -> 0.61-0.56 = 0.05 ms (4)

Decryption= (after merging salt) - (before merging salt) - - > 0.73 – 0.72=0.01 ms (5)

5.6. Randomness analysis

The strength of the derived keys depends on the randomness of the generated sequences, so the randomness of the keys produced through the proposed model is tested through the standard Cryptool V1.4.41 program, which includes number of tests, including the FIPS PUB-140-1 battery test. It measures the entropy per 2500 bytes, this tool measures the entropy (0-8) [37], so 2500 bytes are tested for derived keys set, and the results shown in the Table 4, where the significance level alpha equal (0.01), in Table 5 results of FIPS PUB-140-1 test presented and the entropy of testes is 7.92 From 8.00. The results indicate the robustness and quality of the randomisation produced through the model.



(a) (b)

Figure 9. These figures are; (a) delay time before merging salt; (b) delay time after merging salt

Table 4. Cryptool randomness results

Test Name	Frequency	Poker	Runs	Serial
Maximal test value	6.635	6.635	13.280	9.210
Test result	1.376	1.228	1.531	1.038
Test State	passed	passed	passed	passed

Table 5. FIPS PUB-140-1 battery test.

Test Name	Long Run	Monobit	Poker	Runs	Final FIPS Result
Test State Result	passed	Passed	passed	passed	passed

6. RESULTS AND DISCUSSION

In this section, the results of the system implementation and evaluation are presented.

6.1. System implemntation

Thus, acknowledging the performance of the proposed model from the foregoing encryption analysis, the DMS system is designed and implemented and the AES algorithm applied with deriving the keys through PBKDF2 and RNG to generate random salt and protect it with the secondary key and then merge it with the encrypted text in the database in order to retrieve dams' information entries encrypted and make sound decisions for the integrity of the dam structure and the preservation of people's lives. Where the system containes a secure login interface with an interface for sending telegrams from the authorized employees to the National Center for Decision-Making, and Figure 10 illustrates the most important interfaces and database representation in SQL Server.

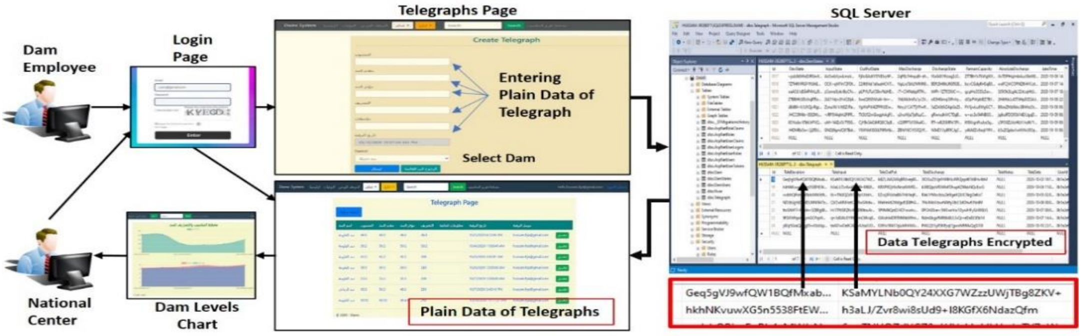


Figure 10. System implementation

6.2. Evaluation

Accordingly, to reviewing the outstanding issues of dam management in the second section (case study) of Fallujah dam, this section explains the results of implementing the proposed DMS system and evaluating the performance after examining and testing the system in Fallujah dam. The opinions of 45 employees are presented as shown in Figure 11, and the same employees who were questioned in assessing issues and problems prior to the system design.

The evaluation questionnaire is prepared on three axes (technical, organizational, and security) containing 15 items. The results are calculated on a Likert scale with the calculation of the arithmetic mean and the degree of standard deviation. As the result of the evaluation indicates the effectiveness of the proposed system and the extent to which it achieves its objectives to solve the issues for which the system is designed, as well as the satisfaction of the dam management employees, and ease of use.

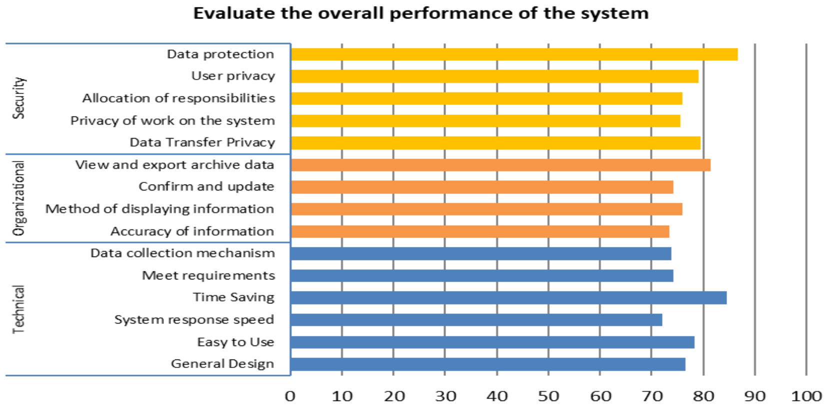


Figure 11. Proposed system evaluation

7. CONCLUSION

The management of Fallujah dam is studied as a model of vital infrastructure in Iraq to address the lack of security that must be solved and to enrich the insufficient performance in the future. This is done by presenting a design for the proposed system with a multi-tier secure model for dam management in Iraq. A proposed encryption model based on the AES algorithm plus derived keys size 256-bit and merging the cipher result with the dynamic random salt values and protect it via slave key, and base64 encryption is implemented for the final result. AES algorithm is selected based on high performance and speed according to encryption analysis executed between several algorithms. It can successfully decrease risks of certain issues related to data and system security such as theft of sensitive data from database. This paper testes Fallujah dam administration and evaluates it in real-time and the results of the assessment will reveal the contingency plans and maintenance strategies to be achieved in the future. Continuously, it is hoped that the overall work, including system design, multi-tier secure model, testing, and maintenance will serve both academia and industry in Iraq on how to securely improve the IT infrastructure of dams and similar electronic physical systems.

REFERENCES

[1] L. Sundberg, “Electronic government: Towards e-democracy or democracy at risk?,” *Saf. Sci.*, vol. 118, no. May, pp. 22–32, 2019, doi: 10.1016/j.ssci.2019.04.030.

[2] T. P. DiNAPOLI, “Dam infrastructure: Understanding and managing the risks,” *An Annu. Rep. by Off. New York State Comptrol. Div. Local Gov. Sch. Accountability.*, 2019.

[3] J. Jeon, J. Lee, D. Shin, and H. Park, “Development of dam safety management system,” *Advances in Engineering Software*, vol. 40, no. 8, pp. 554–563, 2009.

[4] M. A. Alameri, “Application of E-government concept In Iraqi correction service,” *Iraqi Journal of Information Technology*, vol. 8, no.1, p. 1, 2017.

[5] D. Torres, “Cyber security and cyber defense for Venezuela: An approach from the soft systems methodology,” *Complex & Intelligent Systems*, vol. 4, no. 3, pp. 213–226, 2018, doi: 10.1007/s40747-018-0068-x.

[6] D. Roy, S. Paul, and S. Das, “A comparative study of AES, Blowfish, Two fish and serpent cryptography algorithms,” *Elixir Inform. Tech.*, vol. 72, pp. 25218–25219, 2014.

[7] M. S. S. E. Jeevalatha, “Evolution of AES, Blowfish and Two fish Encryption Algorithm 1,” *International Journal of Scientific & Engineering Researc*, vol. 9, no. 4, pp. 115–118, 2018.

[8] S. Almuhammadi and I. Al-Hejri, “A comparative analysis of AES common modes of operation,” *2017 IEEE 30th Canadian conf. on electr. and comp. eng. (CCECE)*, 2017, pp. 1–4, doi: 10.1109/CCECE.2017.7946655.

[9] B. Zhang, Y. Ye, X. Shen, G. Mei, and H. Wang, “Design and implementation of levee project information management system based on WebGIS,” *Royal Soc. open sci.* vol. 5, no. 7, 2018, doi: 10.1098/rsos.180625.

[10] J. C. Ellison, *et al.*, “Real-time river monitoring supports community management of low-flow periods,” *Journal of Hydrology*, vol. 572, no. February, pp. 839–850, 2019.

[11] O. N.- Boateng, M. Asante, and I. K. Nti, “Implementation of advanced encryption standard algorithm with key length of 256 bits for preventing data loss in an organization,” *International Journal of Science and Engineering Applications*, vol. 6, no. 3, pp. 88–94, 2017, doi: 10.7753/ijsea0603.1004.

[12] N. Mathur and R. Bansode, "AES based text encryption using 12 rounds with dynamic key selection," *Procedia Computer Science*, vol. 79, pp. 1036–1043, 2016, doi: 10.1016/j.procs.2016.03.131.

[13] M. M. Bachtiar, T. H. Ditanaya, S. Wasista, and R. R. Kurnia Perdana, "Security Enhancement of AES Based Encryption Using Dynamic Salt Algorithm," *2018 International Conference on Applied Engineering (ICAE)*, 2018, pp. 1–6, doi: 10.1109/INCAE.2018.8579381.

[14] N. A. Sharma and M. Farik, "A performance test on symmetric encryption algorithms-RC2 Vs Rijndael," *International Journal Of Scientific & Technology Resear*, vol. 6, no. 7, pp. 292–294, 2017.

[15] M. Alrammahi and U. Kaur, "Development of advanced encryption standard (AES) cryptography algorithm for wifi security protocol," *International Journal of Advanced Research in Computer Science*, vol. 5, no. 3, pp. 62-67, 2014,. doi: 10.13140/RG.2.2.20993.97124.

[16] E. M. de Los Reyes, A. M. Sison, and R. P. Medina, "File encryption based on reduced-round AES with revised round keys and key schedule," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 897–905, 2019, doi: 10.11591/ijeecs.v16.i2.pp897-905.

[17] A. Altigani, S. Hasan, S. M. Shamsuddin, and B. Barry, "A multi-shape hybrid symmetric encryption algorithm to thwart attacks based on the knowledge of the used cryptographic suite," *Journal of Information Security and Applications*, vol. 46, pp. 210–221, 2019, doi: 10.1016/j.jisa.2019.03.013.

[18] H. O. Alanazi, B. B. Zaidan, A. A. Zaidan, H. A. Jalab, M. Shabbir, and Y. Al-Nabhani, "New comparative study between DES, 3DES and AES within nine factors," *Journal of Computing*, vol. 2, no. 3, pp. 152–157, 2010.

[19] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish," *Procedia Computer Science*, vol. 78, pp. 617–624, 2016, doi: 10.1016/j.procs.2016.02.108.

[20] A. M. Abdullah, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," *Cryptography and Network Security*, vol. 16, pp. 1-11, 2017.

[21] A. Devi and B. S. Ramya, "Two fish algorithm implementation for lab to provide data security with predictive analysis," *International Research Journal of Engineering and Technology*, vol. 4, no. 5, pp. 3033–3036, 2017.

[22] T. Shah, T. U. Haq, and G. Farooq, "Improved SERPENT algorithm: Design to RGB image encryption implementation," *IEEE Access*, vol. 8, pp. 52609–52621, 2020, doi: 10.1109/ACCESS.2020.2978083.

[23] A. V. Tutueva, A. I. Karimov, L. Moysis, C. Volos, and D. N. Butusov, "Construction of one-way hash functions with increased key space using adaptive chaotic maps," *Chaos, Solitons and Fractals*, vol. 141, p. 110344, 2020, doi: 10.1016/j.chaos.2020.110344.

[24] J. Tchórzewski and A. Jakóbik, "Theoretical and experimental analysis of cryptographic hash functions," *Journal of Telecommunications and Information Technology*, no. 1, pp. 125–133, 2019, doi: 10.26636/jtit.2019.128018.

[25] S. Hameed and G. G. Jumaa, "Digital signature based on hash functions," *International Journal Of Advancement In Engineering Technology, Management and Applied Science (IJAETMAS)*, vol. 4, no. 1, pp. 88–89, 2017.

[26] S. Debnath, A. Chattopadhyay, and S. Dutta, "Brief review on journey of secured hash algorithms," *2017 4th Int.l Conf. Opto-Electronics and Applied Optics (Optronix)*, 2018, pp. 1–5, doi: 10.1109/OPTRONIX.2017.8349971.

[27] M. A. Berlin, S. Muthusundari, C. S. Anita, D. Rajalakshmi, M. Rajkumar, and R. Dheekshitha, "A HMAC algorithm based secure online transaction system using block chain technology," *Materials Today: Proceedings*, 2020, doi: 10.1016/j.matpr.2020.10.065.

[28] H. Yang, M.-Z. Liu, Y.-H. Xu, Y.-J. Wu, and Y.-N. Xu, "Research of automotive ethernet security based on encryption and authentication method," *International Journal of Computer Theory and Engineering*, vol. 11, no. 1, pp. 1–5, 2019, doi: 10.7763/ijcte.2019.v11.1230.

[29] Y. Li, D. Zhao, H. Li, and Z. Kou, "HMAC-SHA1 applied in the diagnosis service for security access," *Lecture Notes in Electrical Engineering*, vol. 328, pp. 507-512, 2015, doi: 10.1007/978-3-662-45043-7_52.

[30] Ü. Çavuşoğlu, A. Akgül, A. Zengin, and I. Pehlivan, "The design and implementation of hybrid RSA algorithm using a novel chaos based RNG," *Chaos, Solitons and Fractals*, vol. 104, pp. 655–667, 2017, doi: 10.1016/j.chaos.2017.09.025.

[31] P. Bajpai and R. Enbody, "Dissecting. net ransomware: Key generation, encryption and operation," *Network Security*, vol. 2020, no. 2, pp. 8–14, 2020, doi: 10.1016/S1353-4858(20)30020-9.

[32] A. Visconti, O. Mosnáček, M. Brož, and V. Matyáš, "Examining PBKDF2 security margin—Case study of LUKS," *Journal of Information Security and Applications*, vol. 46, pp. 296-306, 2019, doi: 10.1016/j.jisa.2019.03.016.

[33] M. G. Avasare and V. V. Kelkar, "Image encryption using chaos theory," *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, 2015, doi: 10.1109/ICCICT.2015.7045687.

[34] S. Zhu, C. Zhu, and W. Wang, "A new image encryption algorithm based on chaos and secure hash SHA-256," *Entropy*, vol. 20, no. 9, 2018, doi: 10.3390/e20090716.

[35] A. Mousa and O. S. Faragallah, "Security analysis of reverse encryption algorithm for databases," *International Journal of Computer Applications* vol. 66, no. 14, pp. 19–27, 2013.

[36] M. Roohi, C. Zhang, and Y. Chen, "Adaptive model-free synchronization of different fractional-order neural networks with an application in cryptography," *Nonlinear Dynamics*, vol. 100, no. 4, pp. 3979–4001, 2020, doi: 10.1007/s11071-020-05719-y.

[37] C. Easttom, A. Ibrahim, A. Chefranov, I. Alsmadi, and R. Hansen, "Towards a deeper NTRU analysis: A multi modal analysis," *International Journal on Cryptography and Information Security (IJCIS)*, vol. 10, no. 2, pp. 11–22, 2020, doi: 10.5121/ijcis.2020.10202.