Faculty of Communication and Information Sciences, University of Ilorin, Ilorin, Nigeria
Proceedings of the 3rd International Conference on ICT for National Development and Its Sustainability, May 21st – 24th, 2024. (ICT4NDS2024) pp. 15-34

Development of a Multi-Level Data Encryption Standard with Residue Number System for Data Security

Aminat ADEBAYO[1], Abidemi Emmanuel ADENIYI[2], Jumoke AJAO[3], Rafiu ISIAKA[4], Kazeem GBOLAGADE[5], Sulaiman ABDULSALAM[6]

[1]Department of Computer Science, Al-Hikmah University, Ilorin, Nigeria;
ayakub@alhikmah.edu.ng
[2]Department of Computer Science, College of Computing and Communication Studies, Bowen University, Iwo, Nigeria.
abidemi.adeniyi@bowen.edu.ng
[3,4,5,6]Department of Computer Science, Kwara State University, Malete, Nigeria.
jumoke.ajao@kwasu.edu.ng, abdulrafiu.isiaka@kwasu.edu.ng, kazeem.gbolagade@kwasu.edu.ng, sulaiman.abdulsalam@kwasu.edu.ng

Abstract

Data security is critical in ensuring the privacy of data including sensitive material that ought to only be known by a few people. Every society needs secured data to maintain the integrity and authentication of the data. Data Encryption Standard (DES) is a block cipher algorithm that has been used to secure data or information over the years. Despite the tremendous efforts made by researchers on DES algorithm and efforts to reduce its computational complexity, DES is still susceptible to brute force attack. The need to increase the degree of security of DES algorithm led to the introduction of Residue Number System (RNS) to the DES algorithm as proposed in this study. The method for DES-RNS multilevel encryption uses a 64bits plaintext message which was encrypted using the DES technique. The 64bits plaintext message was divided into two equal halves; 32bit left plaintext (LPT) and 32bit right plaintext (RPT). The RPT was encrypted using 48bit sub-keys and the result was XORed with LPT. The transformation of RPT and LPT was performed for sixteen (16) rounds to produce encrypted text of 64bits. The encrypted text of DES was converted to American Standard Code for Information Interchange (ASCII) and passed through RNS forward conversion. The RNS made use of the moduli set 〖m_1=2〗^n+1,m_2= 2^n and m_3=2^n-1. The decryption was performed using Chinese Remainder Theorem (CRT). The result was evaluated when it comes to cryptographic time, encryption/decryption memory, encryption/decryption throughput and security on three varying text sizes (256, 800 and 1472 bit) for DES only and DES-RNS multilevel cyrptosystem. Using DES, time and throughput shows a better performance for 256bit, 800bit and 1472bit message size but lesser performance in memory and security for 256bit, 800bit and 1472bit message size. On the other hand, using DES-RNS, time and throughput gives a lesser performance for 256bit, 800bit and 1472bit message size. Therefore, DES-RNS multilevel encryption model outperformed the conventional DES model in regard to storage utilization and safety, thereby achieving the aim of this research. Consequently, this DES-RNS model can be employed where security and memory conservation is of utmost concern.

Keywords: *Cryptography, Block Cipher, Data Encryption Standard (DES), Residue Number System (RNS).*

Online ISSN: 2645-2960
Print ISSN: 2141-3959

1. Introduction

Data security is a collection of techniques designed to protect data from unwanted access or change. Data protection is critical in ensuring the privacy of sensitive content that ought to only be accessed by specified parties (Logunleko et al, 2020). Every society needs secured data to maintain the integrity and authentication of the data. Cryptography is a sub-field under data security which deals with safeguarding sensitive data. It's a way of maintaining and disseminating information in a manner that only those who are authorized can access it supposed to read and process it can do so (Maurya, 2021). Cryptography was introduced to keep our information secure. Two categories of cryptographic methods which are used to secure data are Asymmetrical cryptography and symmetric cryptographic encryption are two types of encryption. There are many cryptographic algorithms that are developed and widely used for information security such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), Tripple Data Encryption Standard (TDES), Blowfish etc. Some of these cryptographic algorithms were developed using different techniques and structures. Among these techniques are caesar cipher, monoalphabetic cipher, and feistel cipher model. DES is one of the algorithm that uses the structure of feistel cipher, which is one of the major structure used for the design of block cipher.

DES is a block cipher, which means that rather than one bit at a time, an encryption key and procedure are deployed to a block of data at the same time (Jawahar and Nagesh, 2018). DES divides an original message into 64-bit blocks to encrypt it. Using permutations and replacement, each block is enciphered into a 64-bit ciphertext using the private key. The method consists of 16 rounds, with the final encryption block being reliant on all preceding blocks. Decryption is just the inverse of encryption, with the identical stages but with the keys applied in the opposite order. The most fundamental form of attack for any cipher is brute force, which includes attempting each key until you locate the correct one. The length of the key influences the number of potential keys and thus the attack's viability. DES employs a 64-bit key, however eight of those bits are dedicated to parity tests, thereby reducing the key to 56 bits. As a result, it would take a maximum of 256 attempts, or 72,057,594,037,927,936, to locate the proper key. DES is weak against brute force attack, susceptible to linear and differential cryptanalysis attack (Mahajan and Sachdeva, 2013). Due to this brute force attack, it is imperative to enhance the security level of DES. In ameliorating this problem, this research work employed Residue Number System (RNS) to further enhance the security of DES algorithm using a multi-level approach.

RNS is a numeric scheme that quickens numerical calculations by dividing numbers into smaller units that are independent of one another (Babatunde et al, 2016). RNS designs are usually made up of three major components: a binary to residual converter, residue mathematical units, and a residue to binary converter. The Chinese Remainder Theorem (CRT) and Mixed Radix Conversion are two of the most extensively utilized reverse conversion procedures (MRC). This research work employed forward and reverse conversion technique.

The motivation of this study was as a result security challenges of Data encryption standard. In today's environment, encryption is critical for protecting confidential material from eavesdropping, intruders, and unauthorized users. There have been several assaults in which unknown individuals attempted to disrupt the interaction between two approved devices. DES can be conclusively said to have security and space complexity issues. Therefore, this study focus on the improving the security flaws of DES algorithm by enhancing its

structure in terms of security. In addition enhance the performance of the algorithm through the use of RNS, thus, deliver an efficient algorithm in terms of security and memory consumption.

The remaining part of this study is divided into five sections. Section two discussed the review of related works with the summary of literature. The material and methods used in this study was discussed in section three. The detailed of the method including the DES and DES+RNS algorithm was given in details. Section four gives the results generated in the study with detailed discussion. The performance and metric used was display with tables and figures in this section. The conclusion of this study with future research work was given in section five.

2. Related Works

To examine and improve the performance of different cryptography algorithms, a lot of investigation and the symmetric encryption scheme has been improved. This literature provides an overview of various improvements and analysis performed on the DES algorithm and other cryptographic algorithms.

Reyad et al, (2021) worked on Key-Based Enhancement of DES for Text Security. New key distribution function (KE-DES) was created for the key and data for more security. The security is limited to the key and the input data is textual.

Kumar et al, (2021) present a hybrid approach for cloud security using DES and RSA cryptographic algorithms. It was found that when compared to the present system, the model raises data security to the highest level possible while also taking less time to upload and retrieve text files of DES alone.

Maurya, (2021) carried out a comparative study of security algorithm for data transfer. The cyrptographic algorithms compared are DES, AES and RSA. The experimental results are implemented using the Visual studio Net packages. Seven factors were used as the performances metrics, which include key length, cipher type, block size, designed, crypto algorithms, safety, possibility key, ACSII readable letter keys, and time necessary to verify all possible keys are also compared for different file size. Based on the security factor, AES was reported to be excellent secured, DES was said to be less secured and RSA was reported to be very less secured.

Logunleko et al, (2020) compared the symmetric cryptographic mechanisms DES, AES, and EB64 for data security. This study focused on cryptography as a means of achieving zero tolerance for data in transit security. Performance of those algorithm is evaluated by considering the parameters such as simulation time taken, encryption time and decryption. The result showed that AES is excellent secured, DES is not secured enough and EB64 is adequately secured.

Kumar and Saxena, (2020) conducted a research on System Development File Transmission Safety Using AES and DES Encryption Smartphone. Three cryptographic algorithms were combined i.e AES and DES. It was discovered that the hybrid method to secure the transmission of information from the sender to the recipient via cloud service is more dependable than traditional asymmetrical or symmetrical encryption and decryption of information.

Alhassan et al, (2020) carried out a research on Enhancment of RNS and k-shuffle are used to secure images during communication. The k-shuffling approach was used to jumble picture pixels and then dissect them using RNS. The result shows that images are secured, no loss of information and reduction in images size as well as transmission speed was achieved.

Odunayo (2020) worked on improving AES with RNS. Instead of transmitting just one private key straight to the receiver, the proposed approach sends two private keys to the recipient decoding of the two cipher text is converted returned to original message. At the same time, there is a speed issue. The results demonstrated the improved AES's effectiveness on text data. As a result, while the traditional AES is slightly faster than the implemented algorithm, the proposed algorithm's performance can still be improved.

Mahmoud and Alqumboz, (2019) performed an EBMSR traditional database cryptography relies on multilayer security. The cryptographic algorithms combined are AES-256, DES and RC4. It was concluded that cryptography centered on a layered technique improved database security and created challenges for intruders to reveal the data.

Anbazhagan et al, (2019) carried out a research on enhanced privacy preserving using multilevel encryption technique. The technique combines blowfish with AES and it was reported that a multilevel data encryption technique reduces the theft of data.

Alhag and Mohamed, (2018) presented an enhancement of DES algorithm (KE-DES). The research focuses on increasing the key length to (1024 bits) thereby making it better than the conventional DES which is limited to 64 bits. This research is limited to key length expansion.

Kaur et al, (2017) carried out a study of multilevel cryptography algorithm: Multi-Prime RSA and DES. In this work, the Multi-Prime RSA was used to encrypt the shared DES key thereby making it more secured. The result showed that the data is more protected with the dual protections of DES and Multi-Prime RSA.

Table 1. Summary of Reviewed studies.

Author	Method	Result	Gap
Kaur et al. (2017)	Multi-Prime RSA and DES	Both algorithms are secured	Computational cost
Alhag and Mohamed (2018)	Enhanced DES (KE-DES)	Perform better than classical DES	Key length expansion limitation
Anbazhagan et al, (2019)	Combined blowfish and AES	Reduces the theft of data	Still vulnerable to attacks
Mahmoud and Alqumboz (2019)	AES-256, DES and RC4	Enhanced security	Time complexity
Odunnayo (2020)	AES and RNS	Traditional AES is faster than the proposed algorithm	Issue with speed and memory complexity

Alhassan et al. (2020)	RSA and K-shuffle	No loss of information during encryption of image file	Vulnerability attacks
Kumar and Szxena (2020)	AES and DES	It is more reliable for cloud service security	Computational cost
Logunleko et al, (2020)	DES, AES and EB64	AES is excellent secured while DES is not	Challenges with DES security
Maurya (2021)	DES, AES and RSA	AES outperform other two algorithms	DES is vulnerable to attacks.
Kumar et al. (2021)	Hybrid DES and RSA	The hybrid approach increase the data security	Complexity issues
Reyad et al. (2021)	Key-Based Enhancement of DES	Improve in the data security of textual data	The security is limited to the key input on text data.

2.1 Inference from the Literature

From all the reviewed research works, DES can be conclusively said to have security and space complexity issues. In addition, none of the multilevel approaches reviewed used RNS as a means of enhancement. The security problem of DES as well as memory space consumption can be improved upon through the proposed multilevel DES-RNS scheme employed in this research.

3. Material and Methods

The intended work's goal is to create a model to enhance the security vulnerability of DES using the RNS. The stage involved in this research are:

i. Design of the DES-RNS model
ii. Implementation of the DES-RNS model
iii. Evaluation of the implemented model in step 2.

Design of the DES-RNS Model

The multilevel DES-RNS cryptosystem model is shown in Figure 1. The model takes input plaintext of 64bits and generate 64bits key for encryption. Out of this 64bits key, 48bits was used to encrypt with the RPT of 32bits and the result was XORed with LPT. This was done for 16 rounds to generate the DES ciphertext. The 64bits ciphertext was converted to ASCII which was later passed to RNS forward conversion to produce DES-RNS ciphertext.

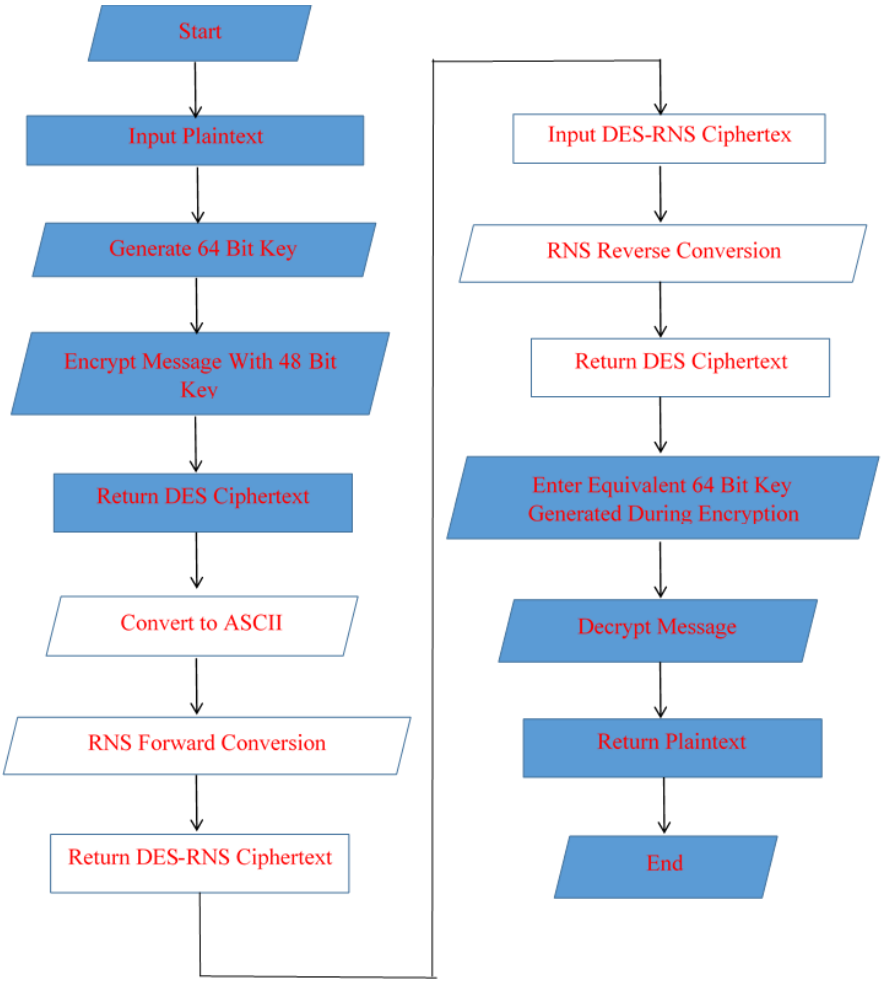


Figure 1: The flow Diagram for the Designed model

DES Encryption Process

Message Input

Random messages of varying bits was utilized to analyze and assess the model's performance. Three sizes of message types were chosen for experimental purposes. The 256 message bit size, 800 message bit size and 1472 message bit size.

Encryption Phase

The encryption phase considered using the DES Algorithm as stated in Algorithm 1.

Algorithm 1 : DES Encryption	
Step 1:	*Input Message*
Step 2:	*Generate 64 bit long key*
Step 3:	*Encrypt Message*
Step 4:	*Return DES cipher-text*

Decryption Phase

The decryption phase considered using the DES Algorithm as stated in Algorithm 2

Algorithm 2 : DES Decryption

Step 1: *Input ciphertext*
Step 2: *Enter Equivalent 64 bit long key generated during encryption*
Step 3: *Decrypt Message*
Step 4: *Return plaintext*

DES-RNS Encryption Process

Message Input

Random messages of varying bits was utilized to analyze and assess the model's performance. Three sizes of message types were chosen for experimental purposes. The 256 message bit size, 800 message bit size and 1472 message bit size.

Encryption Phase

The encryption phase considered using the DES+RNS as stated in Algorithm 3.

Algorithm 3 : DES-RNS Encryption

Step 1: *Input Message*
Step 2: *Generate 64 bit long key*
Step 3: *Encrypt Message*
Step 4: *Return DES ciphertext*
Step 5: *Convert Cipher Text to ASCII values*
Step 6: *Convert ASCII values to Residues (Encrypted Output) using forward conversion.*

Decryption Phase

The decryption phase considered using the DES with RNS enhancement. The sequential integration is given in Algorithm 4.

Algorithm 4 : DES-RNS Decryption

Step 1: *Input DES-RNS ciphertext*
Step 2: *Perform reverse conversion using CRT*
Step 3: *Return DES ciphertext*
Step 4: *Enter Equivalent 64 bit long key generated during encryption*
Step 5: *Decrypt message*
Step 6: *Return plaintext*

64 bit plaintext was encrypted using the DES encryption algorithm. Initial Permutation was performed on the plaintext after which it was divided into two equal halves; the 32 bit left plaintext (LPT) and the 32 bit right plaintext (RPT). DES uses a 64 bit key and this by eliminating every eighth bit of the original key, a 64-bit key is turned into a 56-bit key which was then used as parity check bit. As a result, a 56-bit key is accessible for every cycle. For each round, a distinct 48-bit Sub Key was produced from this 56-bit key using a procedure in key transformation called compression permutation. The 56-bit key is compressed to 48 bits during the key translation procedure. The extension permutations procedure then increases the 32-bit RPT to 48 bits. The 48-bit key has been utilized to encode the 48-bit RPT as well as the result was XORed with the 48 bit of the LPT, the output of which served as the input to the next round of the DES encryption process. This process was repeated for 16 rounds.

ASCII Conversion

The ASCII conversion is being used to convert the cipher text of the DES which are resultant combination of string output of characters and numbers. The ASCII was triggered in the MATLAB using the double conversion technique.

RNS Forward Conversion (Encryption)

RNS is defined by a collection of k integers m 1, m 2, m 3, ldots, m k, known as the moduli, which are commonly assumed to be pairwise coprime (that is, any two of them have a greatest common divisor equal to one). RNS have been defined for non-coprime moduli, however they are rarely utilized due to their poor characteristics. The moduli set $\{m_1 = 2^n + 1, m_2 = 2^n \; and \; m_3 = 2^n - 1\}$, was used for the implementation of the proposed scheme. The RNS forward conversion helped to generate a secured cipher text that further secured the DES algorithm. The forward conversion method is used for the encryption process of the RNS thereby forming the residues. Forward conversion is the process of converting from decimal to RNS notation. The most direct technique to change from a standard notation to a residue illustration is to divide by each of the specified moduli and then capture the remnants. The residues can be defined as

$x_i = X \bmod m_i.$ (1)

Where

x_i = Residues of X with respect to modulo m_i,

X = Decimal Number

Mod = Modulus operation

m_i = Moduli sets.

RNS Backward Conversion (Decryption)

The resultant decrypted text by the DES-RNS technique which produces residues are being passed to the RNS backward conversion with the supply of equivalent moduli set $\{m_1 = 2^n + 1, m_2 = 2^n \; and \; m_3 = 2^n - 1\}$ that was used during the encryption stage. The Backward conversion process utilizes the CRT to reverse the residues back to the DES cipher text.

$$X = \sum_{i=1}^{n} \left| M_i \left| M_i^{-1} \right|_{m_i} x_i \right|_M \quad (2)$$

where

$M = \prod_{i=1}^{n} m_i$

$M_i = M/_{M\,i}$

and

M^{-1} is the multiplicative inverse of M with respect to m_i

Implementation of the DES-RNS Model

The DES-RNS model was performed done and created using Matlab programming (MATLAB 2016A). For user engagement and reactivity, many functionalities were built and linked to a visual user interface. To build and output the results of the produced work, the developed systems used several component environments in Matlab.

4. Result and Discussion

The developed system DES with RNS was experimented upon with samples of messages with varying bit sizes so as to determine the performance and efficiency of the implemented system. The experiment considered two case studies which are to determine the performance of the RNS on the DES Algorithm, the first case considered encryption and decryption of text messages with only the DES algorithm while the second phase considered the combination of the DES and RNS.

Performance Evaluation of DES and DES+RNS

The implemented system was evaluated on three varying text sizes. The results are shown in the section below.

256 Bit Message Size

A 256-bit size of sample was provided into the built software module, which contained the two studied models (DES alone and DES+RNS). Table 1 displays the results collected in these four separate modules during this approach. Encryption Process (ET), Decoding Time (DT), Encryption Memory (EM), Decryption Memory (DM), Encryption Throughput (ETP), and Decryption Throughput (DT) results are detailed (DTP).

Table 2. 256 Bit Message Results

Evaluation Measures	DES (Seconds)	DES+RNS (Seconds)
Encryption Time	0.334354	1.85846
Decryption Time	0.120378	0.169121
Evaluation Measures	**DES (Byte)**	**DES+RNS (Byte)**
Encryption Memory	1.38183E09	1.38106E09
Decryption Memory	1.38106E09	1.38106E09
Evaluation Measures	**DES (Kilobyte/Seconds)**	**DES+RNS (Kilobyte/Seconds)**
Encryption Throughput	765.66472	137.74846
Decryption Throughput	2126.59910	1513.71807

800 Bit Message Size

The two examined systems were fed 800 bit data into the built programming module (DES only and DES+RNS), which were taken into consideration, the experimentation results was measured with varying evaluation measures the Table 2 gives the results evaluation. The results detailed the Encryption Time (ET), Decoding Time (DT), Encrypted Memory (EM), Decoding Memory (DM), Encrypted Throughput (ETP), and Decoding Throughput (DT) are the several types of encryption times (DTP).

Table 3. 800 Bit Message Results

Evaluation Measures	DES (Seconds)	DES+RNS (Seconds)
Encryption Time	0.26136	1.18305
Decryption Time	0.10733	0.45262
Evaluation Measures	**DES (Byte)**	**DES+RNS (Byte)**
Encryption Memory	1.37526E09	1.36131E09
Decryption Memory	1.37472E09	1.36271E09
Evaluation Measures	**DES (Kilobyte/Seconds)**	**DES+RNS (Kilobyte/Seconds)**
Encryption Throughput	3060.91215	676.21825
Decryption Throughput	7453.64763	1767.48708

1472 Bit Message Size

The two examined models were fed 1472 bit data size into the produced software application (DES only and DES+RNS), which were taken into consideration, the experimentation results was measured with varying evaluation measures the Table 3 gives the results evaluation. The results detailed the Encryption Time (ET), Decoding Time (DT), Encrypted Memory (EM), Decoding Memory (DM), Encrypted Throughput (ETP), and Decoding Throughput (DT) are the several types of encryption times (DTP).

Table 4. 1472 Bit Message Results

Evaluation Measures	DES (Seconds)	DES+RNS (Seconds)
Encryption Time	0.269351	3.86896
Decryption Time	0.154326	0.92687
Evaluation Measures	**DES (Byte)**	**DES+RNS (Byte)**
Encryption Memory	1.38794E09	1.38056E09
Decryption Memory	1.38056E09	1.32954E09
Evaluation Measures	**DES (Kilobyte/Seconds)**	**DES+RNS (Kilobyte/Seconds)**
Encryption Throughput	5464.98806	380.46400
Decryption Throughput	9538.25019	1588.14073

Graphical Illustration

The section shows the graphical illustration of the obtained results based on the varying sizes of the messages used.

Performance Chart of the 256 Bit Message Size

The encryption time shows the execution time for each algorithm to complete the cycle of their operation or processing for encrypting an input message while the decryption time shows the execution time for each algorithm to complete the cycle of their processing for conversion of encrypted message into its original message

Encryption and Decryption Time (256 Bit Message Size)

The encryption and decryption time shows that the DES model outperformed the DES+RNS model because the later uses a multi-level approach and will definitely take more time to run compared to DES only. In this work, time complexity is being traded-off for security.

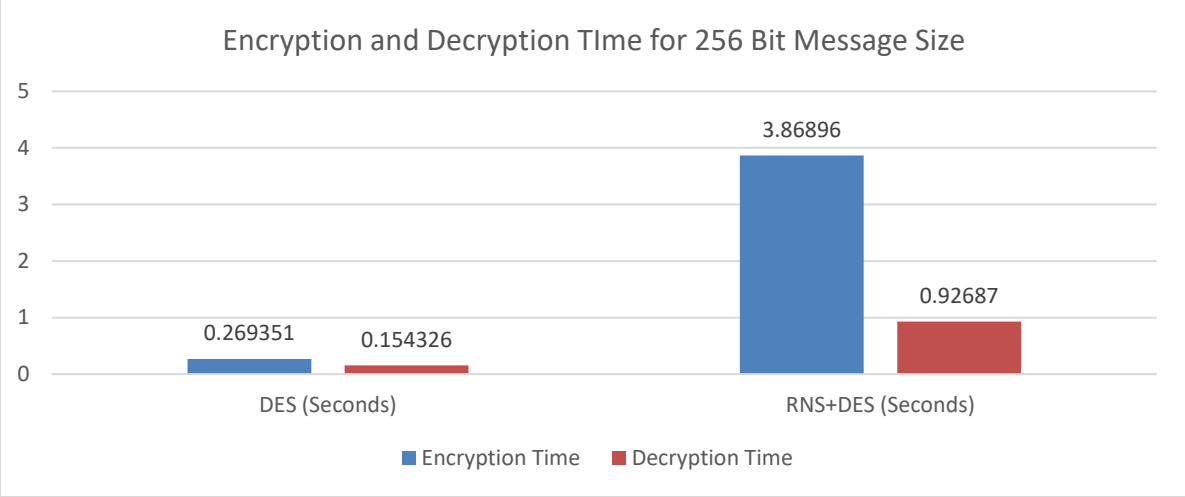


Figure 2. Encryption and Decryption Time for 256 Bit Message

Encryption and Decryption Memory (256 Bit Message Size)

The encryption and decryption memory shows the space complexity used for each of the algorithm to process their total algorithms operations. The encryption and decryption memory shows a better efficiency with the DES+RNS model as compared to the DES algorithm.

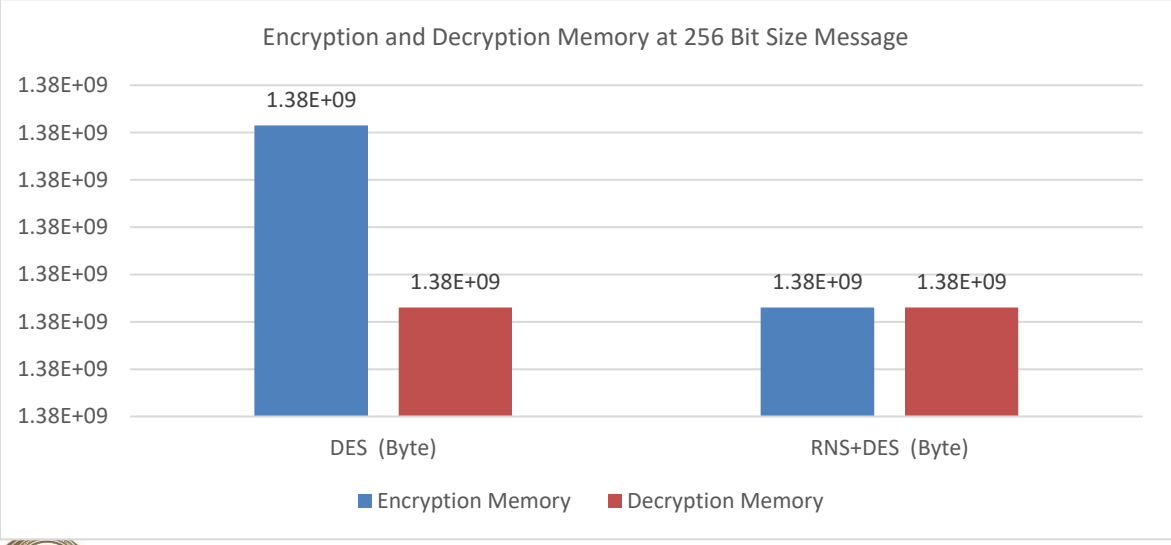


Figure 3. Encryption and Decryption Memory at 256 Bit Message

Encryption and Decryption Throughput (256 Bit Message Size)

The throughput analysis based on the encryption time is measured by the size of the input text per seconds. (Kilobyte/Secs) while the throughput analysis based on the decryption time is measured by the size of the input text per seconds. (Kilobyte/Secs). The higher the encryption or decryption throughput validates the better model, as for the implemented models, the DES model outperformed the DES+RNS model at 256 Bit Message. Though, time complexity is being traded-off for security in the DES+RNS model.

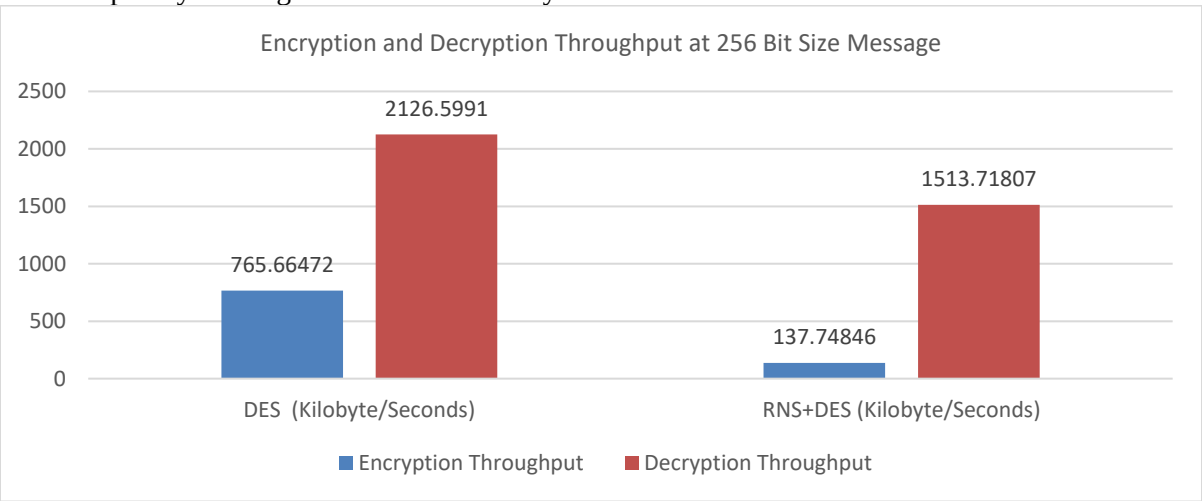


Figure 4. Encryption and Decryption Throughput at 256 Bit Message.

Performance Chart of the 800 Bit Message Size

Encryption and Decryption Time (800 Bit Message Size)

The encryption and decryption time still shows that the DES model outperformed the RNS+DES model because the later uses a multi-level approach and will definitely take more time to run compared to DES only. In this work, time complexity is being traded-off for security as security is of more concern.

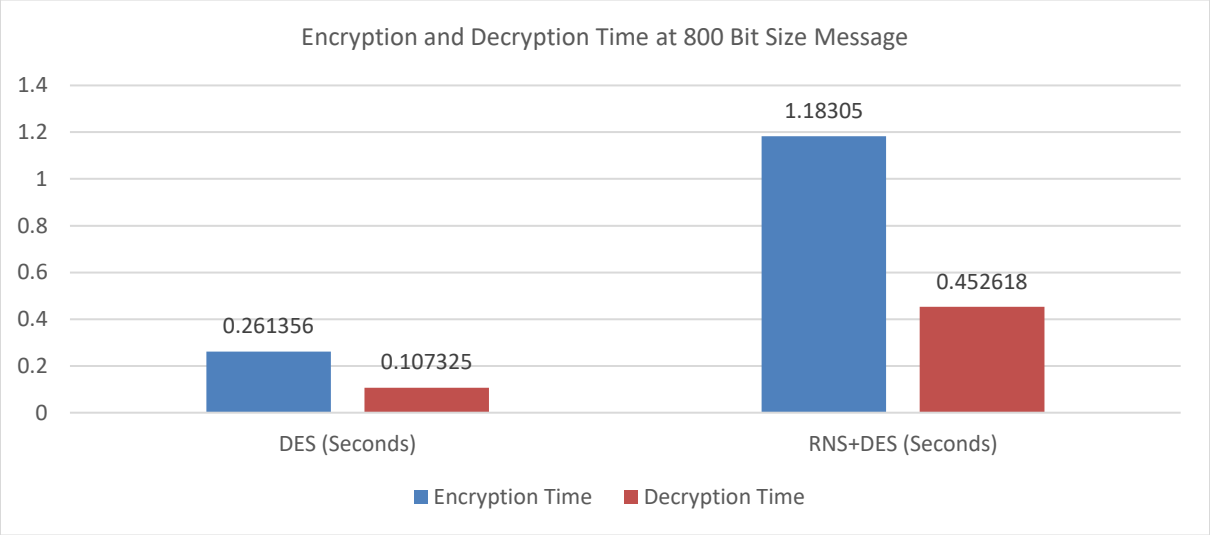


Figure 5. Encryption and Decryption Time for 800 Bit Message

Encryption and Decryption Memory (800 Bit Message Size)

The encryption and decryption memory shows the space complexity used for each of the algorithm to process their total algorithms operations. The encryption and decryption memory shows a better efficiency with the RNS+DES model as compared to the DES algorithm at 800 bit message size.

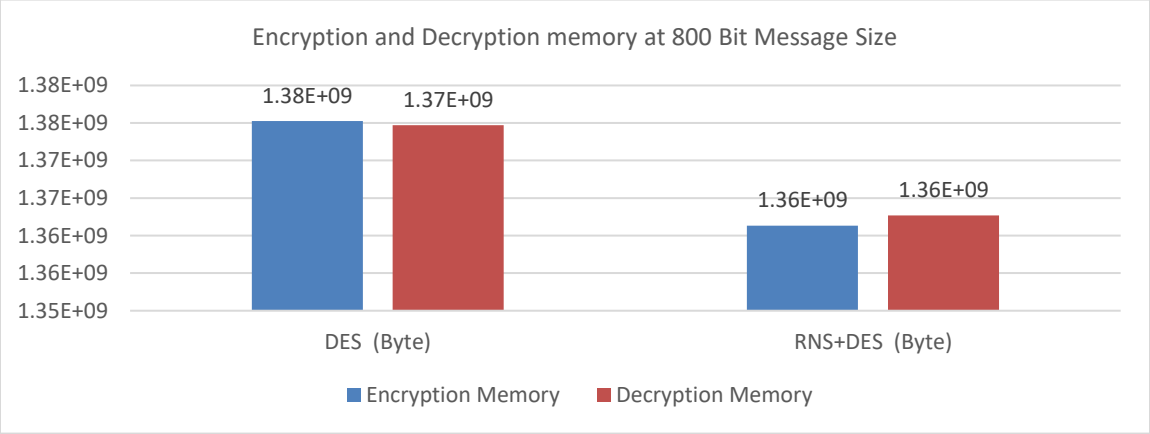


Figure 6. Encryption and Decryption Memory at 800 Bit Message

Encryption and Decryption Throughput (800 Bit Message Size)

The throughput analysis based on the encryption time is measured by the size of the input text per seconds. (Kilobyte/Secs) while the throughput analysis based on the decryption time is measured by the size of the input text per seconds. (Kilobyte/Secs). The higher the encryption or decryption throughput validates the better model, as for the implemented models, the DES model outperformed the DES+RNS model at 800 Bit Message. In this work, time complexity is being traded-off for security.

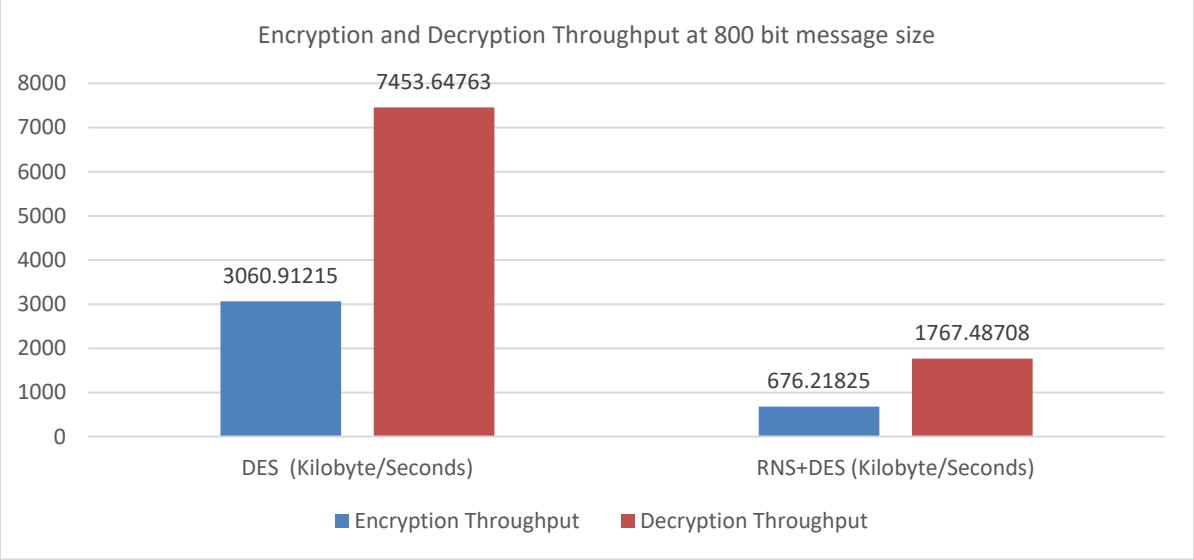


Figure 7. Encryption and Decryption Throughput at 800 Bit Message.

Performance Chart of the 1472 Bit Message Size
Encryption and Decryption Time (1472 Bit Message Size)

The encryption and decryption time still shows that the DES model outperformed the RNS+DES model because the later uses a multi-level approach and will definitely take more time to run compared to DES only. In this work, time complexity is being traded-off for security.

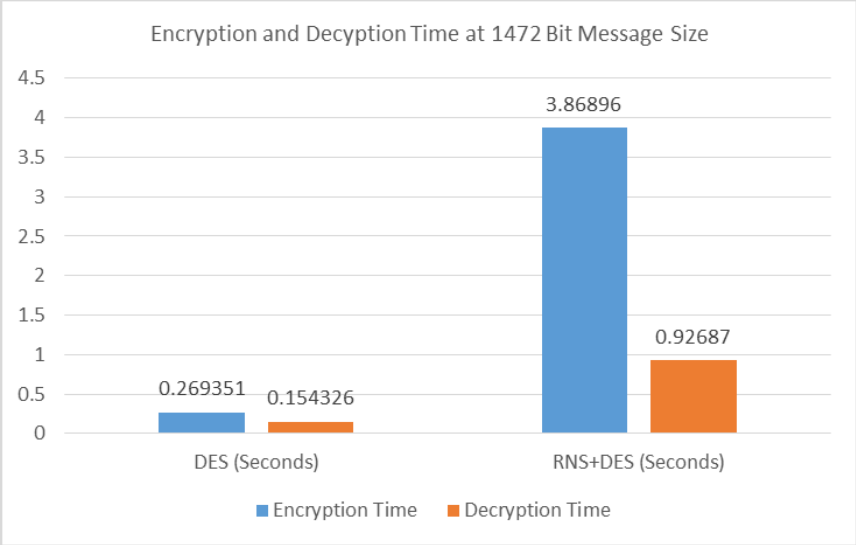


Figure 8. Encryption and Decryption Time for 1472 Bit Message

Encryption and Decryption Memory (1472 Bit Message Size)

The encryption and decryption memory shows the space complexity used for each of the algorithm to process their total algorithms operations. The encryption and decryption memory shows a better efficiency with the RNS+DES model as compared to the DES algorithm even at 1472 bit message size.

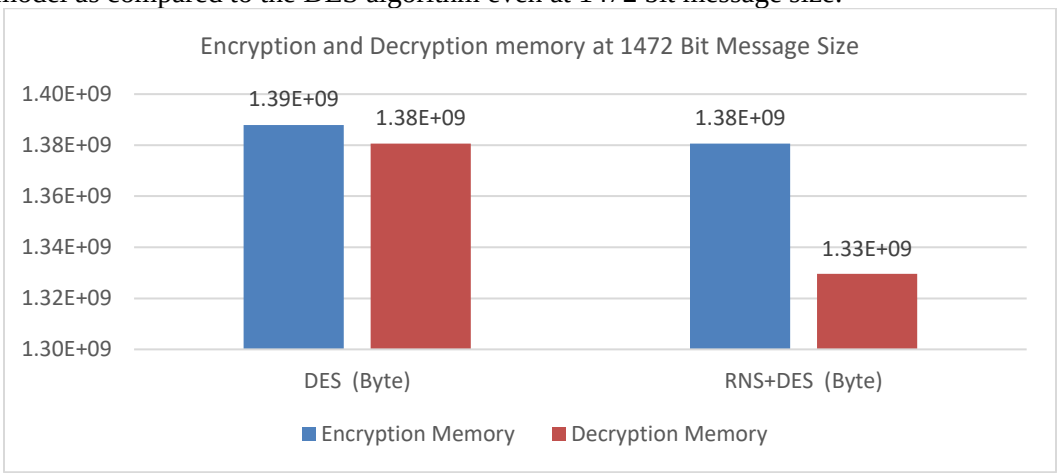


Figure 9. Encryption and Decryption Memory at 1472 Bit Message

Encryption and Decryption Throughput (1472 Bit Message Size)

The throughput analysis based on the encryption time is measured by the size of the input image per seconds. (Kilobyte/Secs) while the throughput analysis based on the decryption time is measured by the size of the input image per seconds. (Kilobyte/Secs). The higher the encryption or decryption throughput validates the better

model, as for the implemented models, the DES model outperformed the DES+RNS model even at 1472 Bit Message. Though, Security is the concern of this work; thereby trading-off time complexity for security.

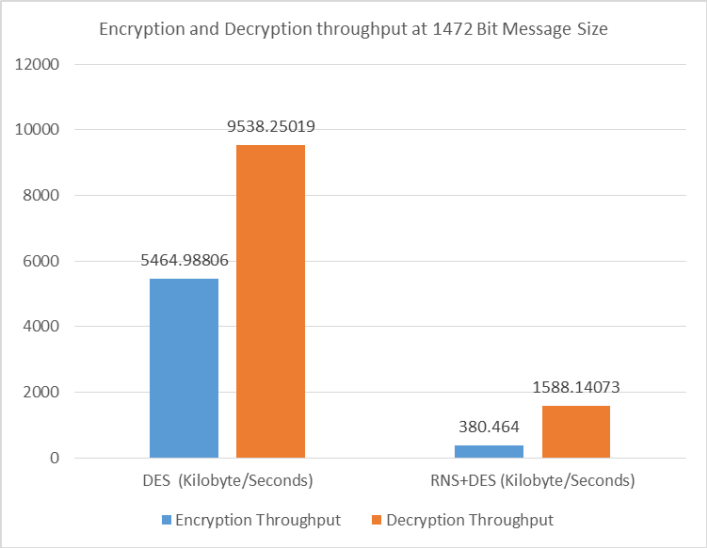


Figure 10. Encryption and Decryption Throughput at 1472 Bit Message.

Results Comparison of the proposed system with other related works

As shown in the Table 4 Maurya (2020) reported that DES is less secured, Logunleko et al (2020) concluded that DES is not secured enough, Chourasia and Singh(2016) in their result discussed that DES is not secured but runs faster than RSA and Hybrid. Prerna and Abhishek also made it known that DES is not secure and is vulnerable to Linear and differential cryptanalysis. Adebayo et al in this research has been able to show that though DES takes a lesser time to run as compared to DES+RNS cryptosystem, but the later is excellently secured.

Table 4. Result comparison of the proposed system with other related works

Author	Security/Vulnerability Of DES	Time(DES)	DES+RNS
Maurya (2021)	Less Secured	NIL	NIL
Logunleko et al (2020)	Not Secured Enough	NIL	NIL
Chourasia and Singh (2016)	Not Secured	DES Outperforms RSA and Hybrid	NIL
Prerna and Abhishek (2013)	Not Secured Enough/Brute Forced, Linear and differential cryptanalysis attack	NIL	NIL

Proposed Method	Not Secured and Vulnerable to attacks	Lesser time compared to DES + RNS	Excellent Secured

5. Conclusion

The DES+RNS model was developed in this thesis to cater for the security vulnerability of DES algorithm. The DES algorithm is susceptible to brute force, linear and differential attacks, Hence the need to work on its security. The designed model (i.e DES+RNS) is more secured as compared to conventional DES encryption. It is a multi-level encryption using DES with RNS. Apart from the security aspect worked on by the proposed technique, it also maximize the memory usage though time complexity serves as a trade-off. From the results, the DES+RNS model outperformed the DES model based on the aim of this thesis. Future study could focus on improving this model by increasing the key size. Another area of future research could be to build a fault-tolerant architecture that allows for redundant data to be added. The DES-RNS design can also be improved by incorporating the RNS inside the DES algorithm itself to address the time complexity issue.

Reference

Aldahdooh, R. M. (2018). Parallel Implementation and Analysis of Encryption Algorithms (Doctoral dissertation).

Alhag, N. M. M., & Mohamed, Y. A. (2018, August). An Enhancement of Data Encryption Standards Algorithm (DES). In 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE) (pp. 1-6). IEEE.

Alhassan, I. Z., Ansong, E. D., Abdul-Salaam, G., & Alhassan, S. (2020). Enhancing Image Security during Transmission using Residue Number System and k-shuffle. Earthline Journal of Mathematical Sciences, 4(2), 399–424. https://doi.org/10.34198/ejms.4220.399424

Alhassan, S., & Gbolagade, K. (2013). Enhancement of the Security of a Digital Image using the Moduli Set. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2(7), 2223-2229.

Ali, A. H., Duhis, A. H., Alzurfi, N. A. L., & Mnati, M. J. (2019). Smart monitoring system for pressure regulator based on IOT. International Journal of Electrical and Computer Engineering, 9(5), 3450.

Anbazhagan, P., Suruthisudar, M., Madhumitha, V., & Ezhilarasan, M. (2019). ENHANCED PRIVACY PRESERVING USING MULTILEVEL ENCRYPTION TECHNIQUE.

Artiles, J. A., Chaves, D. P., & Pimentel, C. (2019). Image encryption using block cipher and chaotic sequences. Signal processing: image communication, 79, 24-31.

Babatunde, A. N., Jimoh, R. G., & Gbolagade, K. A. (2016). An algorithm for a residue number system based video encryption system. Computer Science Series Journal, 14(2), 136-147.

Behrouz, A. and Forouzan, D. (2012). "Cryptography and Network Security", 2nd Edition, Tata McGraw Hill.

Chourasia, S., & Singh, K. N. (2016). An efficient hybrid encryption technique based on DES and RSA for textual data. In Information Systems Design and Intelligent Applications (pp. 73-80). Springer, New Delhi.

Ebrahim, M., Khan, S., & Khalid, U. B. (2014). Symmetric algorithm survey: a comparative analysis. arXiv preprint arXiv:1405.0398.

Gbolagade, K. A., & Cotofana, S. D. (2009, May). An O (n) residue number system to mixed radix conversion technique. In 2009 IEEE International Symposium on Circuits and Systems (pp. 521-524). IEEE.

Gupta, R., & Kapoor, R. (2016). A Review Paper of Data Security in Cloud Computing. Imperial Journal of Interdisciplinary Research (IJIR), 2(8), 501-504.

Hasan, L. F., Ibrahim, A. A. E. A., & Syed, S. Z. B. (2021). Variable Rounds Block Cipher Algorithm Design. International Journal of Engineering and Artificial Intelligence, 2(2), 1-17.

Heeks, R., & Wall, P. J. (2018). Critical realism and ICT4D research. The Electronic Journal of Information Systems in Developing Countries, 84(6), e12051.

Hema, V., & Durga, M. G. (2014). Data integrity checking based on residue number system and chinese remainder theorem in cloud. International Journal of Innovative Research in Science Engineering and Technology, 3(3), 2584–2588.

Hossain, M. J., Hossain, M. B., & Morshed, K. M. (2012, December). Reconfigurable encryption system: Encrypt digital data. In 2012 15th International Conference on Computer and Inf

IBRAHIM, M. B., & GBOLAGADE, K. A. (2019). PERFORMANCE COMPARISON OF HUFFMAN CODING AND LEMPEL-ZIV-WELCH TEXT COMPRESSION ALGORITHMS WITH CHINESE REMAINDER THEOREM. University of Pitesti Scientific Bulletin Series: Electronics and Computer Science, 19(2), 7-12.

Jawahar, T. and Nagesh K. (2018) "DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis," in International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, Volume 1, Issue 2, December 2011), pp.6-12.

Jincharadze, E. (2017). Critical analysis of some cryptography algorithms. Scientific and Practical Cyber Security Journal, 1(2), 60-68.

Kaur, S., Bharadwaj, P., & Mankotia, S. (2017). Study of multi-level cryptography algorithm: multi-prime RSA and DES. International Journal of Computer Network and Information Security, 11(9), 22.

Kumar, J., & Saxena, V. (2020). Hybridization of Cryptography for Security of Cloud Data. International Journal of Future Generation Communication and Networking, 13(4), 4007-4014.

Kumar, S., Karnani, G., Gaur, M. S., & Mishra, A. (2021, April). Cloud security using hybrid cryptography algorithms. In 2021 2nd International conference on intelligent engineering and management (ICIEM) (pp. 599-604). IEEE.

Kuppuswamy, N., Alspach, A., Uttamchandani, A., Creasey, S., Ikeda, T., & Tedrake, R. (2020, April). Soft-bubble grippers for robust and perceptive manipulation. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 9917-9924). IEEE.

Logunleko, K. B., Adeniji, O. D., & Logunleko, A. M. (2020). A comparative study of symmetric cryptography mechanism on DES AES and EB64 for information security. Int. J. Sci. Res. in Computer Science and Engineering, 8(1).

Luen, Y. W. (1974). Knowledge of Mathematics and Science in Ching-hua-yüan. Oriens Extremus, 21(2), 217-236.

Mahajan, P., & Sachdeva, A. (2013). A study of encryption algorithms AES, DES and RSA for security. Global Journal of Computer Science and Technology.

Mahmoud, A. Y., & Alqumboz, M. N. A. (2019, October). Encryption based on multilevel security for relational database EBMSR. In 2019 International Conference on Promising Electronic Technologies (ICPET) (pp. 130-135). IEEE.

Mahyar, H. (2012). Reliable and high-speed KASUMI block cipher by residue number system code. World Applied Sciences Journal, 17(9), 1149–1158.

Maurya, S. (2021). Comparative Study of Security algorithm for Data Transfer.

Muhamin, A., Bariu, K., Alagbe, K., Kumar, R. H. S., Aithal, G., Kumar, P. H. R., Sudeepa, K. B., Mahyar, H., Kar, A., Sur, K., Godara, S., Basak, S., Mukherjee, D., Sukla, A. S., Das, R., Choudhury, R., Alhassan, A., Saeed, I., Agbedemnab, P. A., … Alhassan, S. (2020). A New Cryptographic Model based on Residue Number System and Ribonucleic Acid Properties. Communications on Applied Electronics, 4(9), 1211–1219. https://doi.org/10.34198/ejms.4220.399424

Muhammed, K. J., & Gbolagade, K. A. (2019, December). Enhanced MORE algorithm for fully homomorphic encryption based on secret information moduli set. In 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 469-473). IEEE.

Murali Krishna, B., Khan, H., Madhumati, G. L., Praveen Kumar, K., Tejaswini, G., Srikanth, M., & Ravali, P. (2017). FPGA implementation of des algorithm using DNA cryptography. Journal of Theoretical and Applied Information Technology, 95(10), 2147–2158.

Musliyana, Z., Arif, T. Y., & Munadi, R. (2015). Security enhancement of Advanced Encryption Standard (AES) using time-based dynamic key generation. ARPN Journal of Engineering and Applied Sciences, 10(18), 8347–8350.

Nie, T., & Zhang, T. (2009, January). A study of DES and Blowfish encryption algorithm. In Tencon 2009-2009 IEEE Region 10 Conference (pp. 1-4). IEEE.

Odunayo, T. K. (2020). Improving Advanced Encryption Standard Performance with Residue Number System (Doctoral dissertation, Kwara State University (Nigeria)).

Pedamkar, P. (n.d.). DES Algorithm. EDUCBA. Retrieved June 18, 2022, from https://www.educba.com/des-algorithm

Rahman, M. T., Pinandito, A., & Pramukantoro, E. S. (2017). Perbandingan Performansi Algoritme Kriptografi Advanced Encryption Standard (AES) dan Blowfish pada Text di Platform Android. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2548, 964X.

Reyad, O., Mansour, H. M., Heshmat, M., & Zanaty, E. A. (2021). Key-Based Enhancement of Data Encryption Standard for Text Security. Proceedings - 2021 IEEE 4th National Computing Colleges Conference, NCCC 2021. https://doi.org/10.1109/NCCC49330.2021.9428818

Sa'adah, N., Astawa, I. G. P., & Sudarsono, A. (2018). Trusted Data Transmission Using Data Scrambling Security Method with Asymmetric Key Algorithm for Synchronization. EMITTER International Journal of Engineering Technology, 6(2), 217-235.

Salamudeen, A. (2013). Enhancement of the Security of Digital Images Using Some Moduli Sets. 2(7), 2223–2229.

Saraiva, D. A., Leithardt, V. R. Q., de Paula, D., Sales Mendes, A., González, G. V., & Crocker, P. (2019). Prisec: Comparison of symmetric key algorithms for iot devices. Sensors, 19(19), 4312.

Satheesh Kavuri, A., Kancherla, G. R., & Bobba, B. (2017). An Improved Integrated Hash and Attributed based Encryption Model on High Dimensional Data in Cloud Environment. International Journal of Electrical and Computer Engineering, 7(2), 950.

Singh, H., Patra, P. K., Singh, R., Das, S., Dey, N., & Victoria, A. D. C. (2016). Replication and resubmission based adaptive decision for fault tolerance in real time cloud computing: A new approach. International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), 7(2), 46-60.

Singh, S., & Agarwal, G. (2010). Use of Chinese Remainder Theorem to generate random numbers for cryptography. International Journal of Applied Engineering Research, 1(1), 165-174.

Singh, S. K., Gopi, V. P., & Palanisamy, P. (2014). Image security using des and RNS with reversible watermarking. 2014 International Conference on Electronics and Communication Systems, ICECS 2014. https://doi.org/10.1109/ECS.2014.6892688

Stoyanov, B., & Nedzhibov, G. (2020). Symmetric key encryption based on rotation-translation equation. Symmetry, 12(1), 73.

Tabassum, T., & Mahmood, M. A. (2020). A Multi-Layer Data Encryption and Decryption Mechanism Employing Cryptography and Steganography. ETCCE 2020 - International Conference on Emerging Technology in Computing, Communication and Electronics, 1–6. https://doi.org/10.1109/ETCCE51779.2020.9350908

Verma, J., & Prasad, S. (2009). Security enhancement in data encryption standard. Communications in Computer and Information Science, 31, 325–334. https://doi.org/10.1007/978-3-642-00405-6_34

Wang, X., & Huson, B. (2020). Robust distributed symmetric-key encryption. Cryptology ePrint Archive.

Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., & Khamayseh, Y. (2017, August). Comprehensive study of symmetric key and asymmetric key encryption algorithms. In 2017 international conference on engineering and technology (ICET) (pp. 1-7). IEEE.