Technoscience Journal for Community Development in Africa 3 (2024) 155-166
ISSN 2659-0573

Research Article

An enhanced Honey Encryption to protect transmitted data using Residue Number System

Tawakalitu Afoluwaso Giwa[1,*], Kazeem Alagbe Gbolagade[2], AbdulRafiu Mope Isiaka[2]

[1]Department of Computer Science, University of Abuja, Nigeria
[2]Department of Computer Science, Kwara State University, Nigeria

Abstract: This study proposed an Enhanced Honey Encryption to protect Transmitted Data using Residue Number System**.** This is to allow Honey Encryption (HE) carter for the security of numeric and text data transmitted through an unsecure network. Data is transmitted either offline or online, and as technology advances, online transmission of data gets more attention, and as a result, unauthorized people try to intrude on the privacy of the data being transmitted. Different encryption algorithms have been used to protect the privacy of this data. However, most of the existing methods are vulnerable to brute force attack because the cipher text remains unintelligible and unmeaningful. This confirms invalid key, until the original data is found. Honey Encryption (HE) helps to eliminate vulnerability and withstand brute force attack. However, its problem is that, it works only on numeric data. Hybridization of 'HE' with other algorithms like the Advanced Encryption Standard (AES), Blowfish, etc. were carried out by some researchers, to protect text data but these lead to computational problem and slow processing time. The proposed system hybridized HE with Residue Number System (RNS) improves the capability of HE in protecting text data with less processing time. Distribution Transformation Encoder (DTE) was used to encode each action word by mapping it to its perfect dictionary word and antonyms, and standard conversion table was use to convert each character ASCII code. Traditional Moduli Set (2^n-1, 2^n, 2^n+1) was used to generate the key which produces the ciphertext while Chinese Remainder Theorem (CRT) together with DTE were used for decrypting the ciphertext. When an incorrect key is used to access the data, HE generates an indistinguishable fake but meaningful data, this helps deter attacker from trying further. The proposed system allows HE to protect text data with less processing time compared to other systems. N-values, message size, processing time and security were used to test and evaluate its efficiency. Comparing the obtained results with the existing ones, the proposed system processing time is lesser and more secured because the ciphertext produced is deceiving, and to decrypt it, it requires a lot of efforts like finding the sequence of algorithms used, getting the moduli set used, determining the value of n for the moduli set, knowing the character table and also determining the Distribution Transformation Encoder design.

Keywords: Honey Encryption, Distribution Transformation Encoder, Residue Number System, Moduli Set, Chinese Remainder Theorem

1. INTRODUCTION

More than half of the globe use the internet to share data and information, which makes the security of a transmitted data a battlefield, between the attackers and the defenders. The common method used by the defenders is cryptography. Different encryption algorithms are used to ensure that the data stays secured. Although encryption is a very clever design, it has a major flaw that can be exploited; it can be cracked by brute force if enough time and resources is given by the attacker. Since computational power is ever increasing, the attacker has easier time to break encrypted data. So far, many advancements have been made to prevent breaking of encrypted data. One of the advancements is increasing the length of the encryption key (by increasing n size), so that it will take longer to break it. This will work for the time being at a decent security level, but for how long? (Lindholm & Costin, 2019).

Researchers here over the years proposed various data security methods like compression, cryptography, digital scrambling, stagnography and watermarking, for protecting Information from

*Corresponding author's:
Email: tawakalitu.giwa@uniabuja.edu.ng,

KWASU PRESS AND PUBLISHING

unauthorized access. Encryption is a cryptographic technique and a method used in information security to transform a plaintext into a form unreadable, invisible and unintelligible, during transmission or storage. Cipher text is the product of encryption. Encryption is a widely established and suitable technique to address security issues. The security of an encrypted data depends on the secrecy of the key and the strength of the cryptographic algorithm (Meghashree & Sujatha, 2018), while cryptographic strength is determined by the resource and time it would take to authorizely recover the plaintext. Data encryption is essential in protecting valuable data and information. The need for highly secured communication through a transmission medium is of paramount interest, recognizing the fact that most of the businesses and personal matters are involved (Chang al et., 2011). Many people are using traditional cryptography algorithms to ensure confidentiality of a transmitted data. Encrypted data is sent to ensure that the data is secure and private, and public keys encryption schemes such as RSA and AES are used for this purpose. While the security is achieved at the time of transmitting the data, the data cannot be process in an encrypted format and it can also be attacked before it gets to the recipient and decrypted (Dharshini et al., 2017). To resolve this issue, Homomorphic Encryption technique was developed by IBM in 2009. Homomorphic Encryption provides the facility of processing the data without being decrypted (Michaell & Said, 2021) but the downside is that it is extremely slow and computationally expensive, to the point that it's not currently practical. Other encryption algorithms have also been used in order to ensure the security of a transmitted data. Among these is Triple Data Encryption Standard (3DES), Blowfish, Twofish and the likes. Most of these algorithms have one essential problem; they are vulnerable to brute-force attacks (Ronja, 2019; Lijimol & Dileesh, 2020; Adams, Jourdan, Levac, & Prevost, 2010).

A new highly secured data encryption technique called Honey encryption (HE) was introduced in 2014 by Ari and Thomas. HE is a simple method of encrypting messages using Non min-entropy keys such as passwords and encryption key (Murty, & Mulchandani, 2017*)*. Honey Encryption produces a cipher text which, when decrypted with incorrect keys, yields plausible appearing but bogus plaintexts, called honey messages (Ansari & Sahu, 2017). HE alone, solved the problem of a brute force attack but only on small numeric data. With the increase rate of data exchange in electronic way, information security is becoming more important in data transmission and because of a wide usage of data in industrial process, it is important to protect the confidential information from unauthorized access. For Honey Encryption (HE) to prevent brute force attack on other type of data like text data, researchers introduced hybridization with other encryption algorithm like Advanced Encryption Standard (AES) and Blowfish, the system was able to withstand the problem of brute force attack but, it suffered computation complexity which lead to extremely slow processing time. The computational issue and low processing time made the researchers proposed "An enhanced Honey Encryption to protect Transmitted Data using RNS. Residue Number System (RNS) is known for its computational power (it encode large numbers in to a set of smaller numbers to speed up computation). We implemented a multi-layer encryption by hybridizing Honey technique with Residue Number System to withstand brute force attack with less processing time. Hybridization of HE with RNS ensures a more secured transmitted data with less computational complexity.

The system is used in protecting and preventing transmitted and stored numeric and text data from brute force attack and unauthorized access. The system can be used by any organization as a means of electronic communication to protect their data privacy, especially high profile data. To use the system, every user would be registered on the web platform and granted a unique user id and password. The system was implemented with MATLAB, C# programming and MySQL for the database.

2. LITERATURE REVIEW

Honey Encryption turns each wrong password/key guess made by a hacker to a perplexing dead-end (Edwin, Samsudin & Tan, 2017). When an application or user sends and enters a key to get an encrypted data or database, so long as the key is correct, the data is decrypted and accessible in its original and readable format (Ansari & Sahu, 2017; Yin, Indulska & Zhou, 2017). If the key is incorrect,

the original data will not be decrypted; instead, fake messages will be displayed. If attackers created 100 key attempts, they would get 100 plain text success, even if one of the keys were correct (Yin, Indulska & Zhou, 2017), the real data would be equal from the bogus data (Ansari & Sahu, 2017), but the major drawback of this algorithm is that, it can only work on numeric data. Honey Encryption was introduced by Juels and Ristenpart after observing the high rate of password been compromised. For instance, using conventional encryption scheme, if plaintext, P is a 16-digit Master card number encoded via ASCII and the conventional password-based encryption scheme is used as cipher, the probability that any Pi≠P is a valid ASCII encoding of a 16-digit string which is negligible, at (10/256)16 < 2 −74. Hence, an adversary can reject incorrect messages and recover P with a high probability. Honey encryption provides security beyond the brute-force barrier by producing ciphertext which, upon decryption with wrong keys, yields plausible-looking plaintexts. The idea is to confound the life of the adversary by making message recovery even after trying every candidate keys to be impossible. The central component of HE is called a distribution transforming encoder (DTE) (Ari & Thomas, 2014; Win & Moe, 2018).

Residue Number System

Residue number system (RNS) is known for its parallel arithmetic and it has been used in recent decades in various important applications, from digital signal processing and deep neural networks to cryptography and high-precision computation (Konstantin & isupov, 2021). The RNS is of interest to scientists dealing with computationally intensive applications as it provides efficient highly parallelizable arithmetic operations (Stamenkovie, Jovanovie & Stojanovie, 2010). This number coding system is defined in terms of pairwise coprime integers called moduli, and a large weighted number is converted into several smaller numbers called residues, which are obtained as the remainders when the given number is divided by the moduli (Bello, & Gbolagade, 2017, Qaisar et al, 2016). A useful feature is that the residues are mutually independent, and for addition, subtraction and multiplication, instead of big word length (multiple-precision) operations, we can perform several small word length operations on these residues without carry propagation between them (Popoola, 2019).

RNS is a numeral system representing integers by their values modulo several pairwise coprime integers called the moduli. This representation is allowed by the Chinese remainder theorem, which asserts that, if *N* is the product of the moduli, there is, an interval of length *N*, exactly one integer having any given set of modular values (Mahtab & Shiv, 2021). The arithmetic of a residue numeral system is also called multi-modular arithmetic (Nikolai et al., 2019). Multi-modular arithmetic is widely used for computation with large integers, typically in linear algebra, because it provides faster computation than with the usual numeral systems, even when the time for converting between numeral systems is taken

$$X_i = x \bmod m_i, \quad (1)$$

$$0 \leq x_i < m_i$$

Let *M* be the product of all the m_i. Two integers whose difference is a multiple of *M* have the same representation in the residue numeral system defined by the m_is. More precisely, the Chinese remainder theorem asserts that each of the *M* different sets of possible residues represents exactly one residue class modulo *M*. That is, each set of residues represents exactly one integer in the interval 0... *M*. RNS has two phase, the forward conversion and reverse conversion (Aremu & Gbolagade, 2017).

2.2 Related Works

Kunjal (2020) proposed Differential Privacy and Natural Language Processing to generate contextually similar decoy messages in honey encryption scheme. The author used machine learning algorithm to train the system and generate the decoy message.

Abiodun and Aman (2019) proposed a modified honey encryption scheme for encoding natural language message. They modified honey encryption to support natural language technique by encoding the message in a binary using Natural Language Processing (NLP) tool. The approach is semantically sound and coherent, but it does not address the issue of adversary having prescience about the context of the message. It only works for the situations where there is zero foreknowledge of the messages (Kunjal, 2020).

Ansari and Sahu (2017) proposed securing messages from Brute Force Attack by combined Approach of Honey Encryption and Blowfish. In this work, they combined the honey encoding with both AES and Blowfish separately and then compared the performance of both algorithms. Their experiments show that Blowfish algorithm, when combined with the honey encryption scheme produces results in less time.

Mok, Samsudin and Tan (2017) proposed an extended Honey Encryption (XHE) scheme that enables additional protection method on the encrypted data. Murty and Mulchandani (2017) applied honey encryption to protect the MasterCard's and fundamental content. They develop the essential Honey encryption plan to bolster open key encryption. The paper illustrated the advantages and disadvantages of both the Blowfish algorithm and Honey encryption, indicating the latter being the better for strengthening encryption also based on keys. Their focused was on numeric data only.

Soo and Azman (2017) proposed enhanced Security of Internet Banking Authentication with EXtended Honey Encryption (XHE) Scheme. They enable additional protection mechanism to the existing user authentication that redirect attackers to fake user account and by these attackers will believe they hold of the band data.

Babatunde, Jimoh, and Gbolagade (2016) developed a residue number system based encryption algorithm for encrypting all formats of video data. The algorithm was developed to look into the possibility of ameliorating the computational complexity problems in total video encryption. The paper presents a scheme that reduces the computational complexity in the total video encryption. It utilizes residue number system and was implemented using Java programming language to efficiently secure video data from unauthorized access during transmission and storage. Hema and Durga (2014) proposed a data correctness scheme in which a Third Party can audit the data stored in the cloud and assure the customer that the data is safe. Number theory based systems using Residue Number System and Chinese Remainder Theorem was used to guarantees correct data possession and assures irretrievability upon some data corruptions. This assure the owner of the data that the data is intact and the client can retrieve it from the server.

Juels and Ristenpart (2014) provided a security where too little entropy is available to withstand brute-force attacks that try every key. Their work provides security beyond conventional brute-force bounds. Also provide a hedge against partial disclosure of high min-entropy keys and significantly improves security in a number of practical settings. They built concrete HE schemes for password-based encryption of RSA secret keys and credit card numbers. The key challenges of their work are development of appropriate instances of a new type of randomized message encoding scheme called a distribution-transforming encoder (DTE) and the computational problem due to RSA.

Alhassan and Gbolagade (2013) proposed a new security enhancement scheme for digital images. The scheme employed two methods: Residue Number System (RNS) to Decimal (R/D) encoding and decoding using the moduli set (2^n-1, 2^n, 2^n+1) and a modified Arnold transform algorithm. The encryption process used RNS to Decimal (D/R) converter (encoder) to decompose a plain image into three residual images. The residual images are fused together and encrypted using the modified Arnold transform. In the decryption process, the modified Arnold transform was used to decrypt the cipher image which is then decomposed into three residual images. An R/D converter (decoder) is then used to recover the plain image.

Weyori, Akobre & Armah (2009) proposed a new secured data encryption technique, residue number system was applied to data encryption with (2^n -1, 2^n, 2^n + 1) moduli set which is an adaption of the Traditional Huffman's algorithm of encrypting data where by the frequency of occurrences is used to generate binary codes. The algorithm leads to unreadable encrypted sets of bits which only the decoder with the moduli set use can decrypt.

2.3 Contribution to Knowledge

The study is the first to combine Honey Encryption and Residue Number System (RNS) for protecting text transmitted and storing text data. Compared with the existing hybridized honey encryption, the proposed system processing time is faster and more secured because the result of cipher-text produced is complex and to decrypt it, it requires lot of effort such as: finding the sequence of algorithms used, determine the moduli set used, determine the value of n for the moduli set and also the character table. With this study, people can successfully use honey encryption for text data in every organization and authorizely recover the data in lesser time.

3. MATERIALS AND METHOD

Honey Encryption was hybridized with Residue Number System to improve its security, capability and processing time. This enables it secured to both numeric and text data.

The encryption and decryption steps are highlighted below. The steps explain what is required at each

level of the system development for both the encryption and decryption.

i. A character table was created to convert each of the English keyboard character to a decimal numeral, using standard conversion table (ASCII code).

ii. Honey technique was used to encode the data using the Transformation Distribution Encoder (DTE) before forward conversion was performed to further encrypt the data,

iii. The decimal numbers were transposed using forward conversion based on the selected moduli set $\{2^n-1, 2^n, 2^n+1\}$, to generate residues for each of the numbers, when the value of "n" is entered. The residue generated at this stage is the ciphertext that was sent to the recipient in which the decryption process was performed at the recipient end.

iv. Chinese Remainder Theorem was used as a reverse conversion algorithm for the decrypting ciphertext which is residues based on the moduli set $\{2^n-1, 2^n, 2^n+1\}$, to get back the decimal number.

v. Each of the decimal numbers generated from the reverse conversion was substituted back to English character equivalent.

vi. Then, honey encryption class was called upon to check if the key was correct or not. For the incorrect key, the inverse DTE (I-DTE) check dictionary library for inverse and synonyms of the action word that fit the seed space.

vii. Keyboard character was used as the dataset and different data was picked to evaluate and compare efficiency of the system compare to the existing one. Also the start and end time of execution was record to know the total processing time.

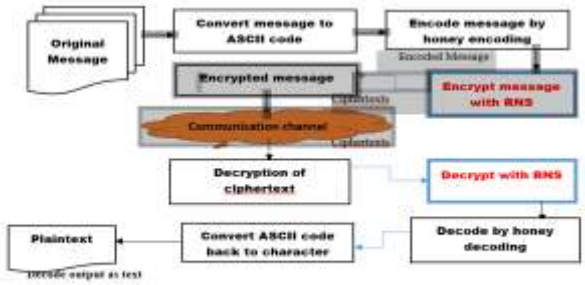

Figure 1: System Framework

The framework shows the steps involved in the system development, and how honey encryption is being hybridized with residue number system to improve its efficiency. Keyboard characters were used as the dataset, and different data were generated for the evaluation, among these is 'hello dammy' which was used as the message sent from the sender to the receiver. The diagram indicates how the plaintext was first substituted to decimal number at the beginning of the encryption process. This was done using standard character to decimal conversion table, after which the substituted data was encoded based on the choice of DTE design chosen. Then, the encoded message was further encrypt with RNS to generate the ciphertext. The ciphertext generated was sent through a communication channel to the recipient.

HEnc (K, M)
S ⟵ $ encode (M)
R ⟵ $\{0, 1\}^n$
C_2 ⟵ F (R||K) ⊕ S
Return (R, C_2)

The algorithm above shows how message is being encoded and decoded with honey encryption C is the ciphertext, K represent the key, F is the cryptographic function, M is the message, R is the string while S represent the seed.

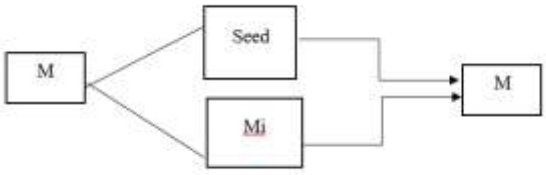

Figure 2: DTE Design for Word Selection

The diagram illustrates how the message was encode with Honey Encryption. Original message M was encoded to seed S which was picked from dictionary library, then the message is encrypted by a generated key K, by the traditional moduli set.

Henc (K, M). Given the symmetric key K, and a message M, let the H be the algorithm and n is the number of random bits, select a uniformly random;

s←$ encode (M), and

R ←$ {0,1}n, outputs the ciphertext,

C = H (R,K) ⊕ s.

Residue Number System (RNS) forward converter was used with 2 and 3 digit numbers to generate the key. This is given in procedure 1;

i. Given the moduli set$\{2^n - 1, 2^n, 2^n + 1\}$, take n=2. Consider X representing character of a message to be encoded, and ASCII character representation X =32. Then the conversion process is as follows;
If X = 32, and n =2 to encrypt X by RNS, for moduli set $\{2^n - 1, 2^n, 2^n + 1\}$, we have,
the moduli $(2^2 - 1, 2^2, 2^2 + 1)$
$= (4 - 1, 4, 4 + 1)$
$= (3, 4, 5)$.

The dynamic range M of moduli (3, 4, 5) = 3 * 4 * 5 = 60.
To get the residue, we have
32 = 100000 = (6 bits, since X is a 2n- bit number).
X is partitioned into 3 blocks; we partition X=32 into 3-bits block.
Thus, $B_1 = 10$, $B_2 = 00$ and $B_3 = 00$
Therefore; $r_1 = |B_1 - B_2 + B_3|_{2^n+1}$

$$|32|_{2^2+1} = |32|_5 = |2 - 0 + 0|_5 = 2$$
$$|32|_{2^2+1} = |32|_5 = 2$$

$r_3 = |B_1 + B_2 + B_3|_{2^n-1}$

$$|32|_{2^2-1} = |32|_3 = |2 + 0 + 0|_3 = 2$$
$$|32|_{2^2-1} = |32|_3 = 2$$

r_2 is the n LSB of X in binary = 00 = 0, that is, 2^n where n =2.
This implies that $|32|_{3,4,5} = \{2, 0, 2\}$. This is the ciphertext that the recipient got, r_1, r_2 and r_3 represent the residues of decimal, X. M is the dynamic range which is the multiplication of all the moduli in the moduli m_i.
which is

$$M = \prod_{j=1}^{N} m_j = 3 \times 4 \times 5 = 60$$

The figure 3 illustrates how the original data "hello dammy" was converted to ASCII code. Each word has been encoded with honey encoding then the decimal number is encrypted with Residue Number System using forward conversion method based on moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ and value of 'n' to generate the residue (ciphertext). With this, the system was able to encrypt the text data. The residue is the ciphertext gotten by the recipient.

Figure 3: Encryption Method

Figure 3: Encryption Method

At the recipient side, ciphertext was decrypted using Chinese remainder theorem which called honey class to decode the data and finally converted it to the original message. Distribution Transformation Encoder (DTE) design chosen in the study maps the action words in the message to another action words in the dictionary table and also antonyms of the words.

$|\hat{m}_1|_{m1}$ is the multiplicative inverse of m_i with respect to the moduli

$$|\hat{m}_1|_{m1} = |4 \times 5|_3 = |20|_3 = 2 \text{ then } \left|\frac{1}{\hat{m}_1}\right|_3 = 2 \quad (2)$$

$$|\hat{m}_2|_{m2} = |3 \times 5|_4 = |15|_4 = 3, \left|\frac{1}{\hat{m}_2}\right|_4 = 3 \quad (3)$$

$$|\hat{m}_3|_{m3} = |3 \times 4|_5 = |12|_5 = 2, \left|\frac{1}{\hat{m}_3}\right|_5 = 3 \quad (4)$$

$$|x|_{60} = |2|20 * 2|_3 + 0|15 * 3|_4 + 2|12 * 3|_5|_{60} \quad (5)$$

$$= |80+0+72|_{60} \quad (6)$$

$$= |152|_{60} \quad (7)$$

$$= 32 \quad (8)$$

So, the decimal number x for residue (2, 0, 2) for moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ and moduli {3, 4, 5} when n = 2, is 32. The decimal was then converted

back to character to get the original message. On the ASCII conversion table, the 32 represent space. This apply to other characters too.

HDec (K, C)
(R, C_2) C
S ⟵ F (R||K) ⊕ C_2
M ⟵ Decode (S)

Return M

A Distribution Transforming Encode (DTE) consists of a pair of algorithms, such that DTE = (encode, decode) (Omolara, Janlan, & Abiodun, 2019). The encode algorithm takes as input a message M∈$\mathcal{M}$ and outputs a set of seed value, S from seed space. The deterministic decode algorithm takes as input a ciphertext S∈C and outputs a message M∈$\mathcal{M}$. The correctness of DTE algorithm follows as for any M∈$\mathcal{M}$, Pr[decode(encode (M)) =M], Inverse DTE (I-DTE). An I-DTE consists of a pair of algorithms, such that I-DTE = (decode, I-encode).

The encode algorithm runs the Cumulative Distribution Function (F (R||K)), Fn such that with a pre-defined message distribution ψm and $\mathcal{M}$ = {M1, M2, ..., M|$\mathcal{M}$|}. The decode algorithm is the inverse of the encode, such that I-decode =Fn-1(S).

A HE [DTE, SE] algorithm is a pair of algorithms (HEnc, HDec) that encrypts a message by using the DTE algorithm, subsequently re-encrypts the output of DTE algorithm by using the key generated by the moduli set as follows: HEnc (K, M). Given the key, K and a message M, let the H be the algorithm and n is the number of random bits, select a uniformly random, s←$ encode(M) and R ←$ {0,1}n, outputs the ciphertext, C = H(R,K) ⊕ s. The process of HEnc (K, M) is illustrated in figure 2 and 3.

The above shows how the ciphertext is XORed with the key to obtain the seed The DTE inversely maps the seed to the original plaintext message. Dictionary table was used to map the action words in the message to another action word and antonyms of the word. When decrypting the message, the DTE design used calculated cumulative probability to search for the dictionary words that perfectly inverse or synonym that matches the action words. For the correct key, the original word is display.

4. RESULTS AND DISCUSSION

Figure 4 is the interface where the message is being sent before the encryption process takes place. Every user would registered on the web platform with unique user id and password. The sender sent the message with a particular value of N which form the encryption key based on moduli set.

Figure 4: Interface of the System Where the Message Is Being Sent.

The figure 5 below shows several messages that has been delivered to recipient's inbox. The message delivered is an encrypted message (ciphertext) which needs to be decrypt by the recipient to get original information. A particular message is being selected by the receiver in order to decrypt it. To do this, the receiver needs to know the steps involved in decrypting the message, the algorithms used, the moduli set used and also, the value of n. The Chinese remainder theorem was used as the reverse conversion algorithm to generate the decimal before honey class was called upon to check for the correctness of the key entered. Then finally, the decimal number generated from RNS is converted to text using the standard conversion table (ASCII table).

Figure 5: Encrypted Message Selected from Recipient Inbox in order to Decrypt.

Figure 6 shows the result of the decryption when the correct key was used, it displays message sent from the sender.

Figure 6: Decryption Result when Original Key is Used

In the figure below, the wrong key was used but it has the same length with the original key. So, a meaningful message was displayed but not the original message. With this, attacker would think that is the original message since the information is meaningful.

Figure 7: Screenshot of the Decryption Result when wrong Key that has same Length with the Original Key is Used

Analysis of the Results

Table 1 is the analysis of the encryption and decryption time in millisecond per value of n. According to this result, the encryption and decryption time increase at Big O (1) i.e at a constant rate as the value of n increases. Figure 6 showed the encryption and decryption time against value of n.

Table 1: Encryption and Decryption Time against N-value

n-values	Encryption time in millisecond	Decryption time in millisecond
1	2.8078	2.8701
2	2.8196	2.9412
3	2.8753	3.0152
4	2.9104	3.0753
5	2.9390	3.5121

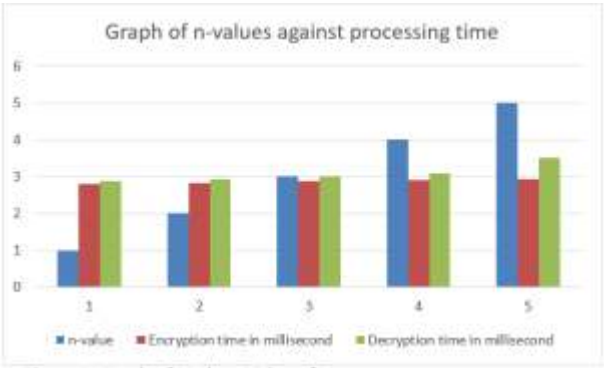

Figure 8: Graph of Various Value of N against Processing Time

Table 2 is the analysis of the encryption and decryption time in millisecond per kilobytes length of message. Encryption time increases at the rate of Big O (1) as the message length increases by 1kb, while the decryption time increases at the rate of Big O (n) per increase in message length.

Figure 2: Encryption and Decryption Time against Massage Size

Message string length	Encryption time in millisecond	Decryption time in millisecond
1kb	3.097	15.243
2kb	3.1175	17.803
3kb	3.235	20.624
4kb	3.2723	23.123
5kb	3.3413	27.003

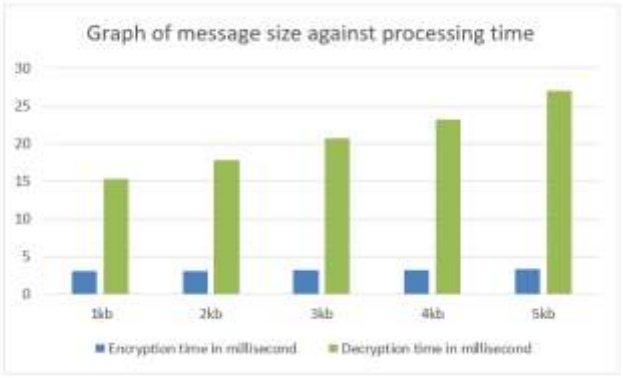

Figure 9: Graph of Varied Message Size in Kilobyte against Processing Time

Table 3 is the analysis of the proposed system (Honey Encryption with RNS) compared to the existing of Honey Encryption with AES, and with Blowfish. The result shows that Honey with AES takes a longer time to excuse the message while the proposed system take lesser time. The propose system is faster compared to when honey encryption was hybridized with AES and Blowfish.

Table 3: Comparison of Proposed System with the Existing One

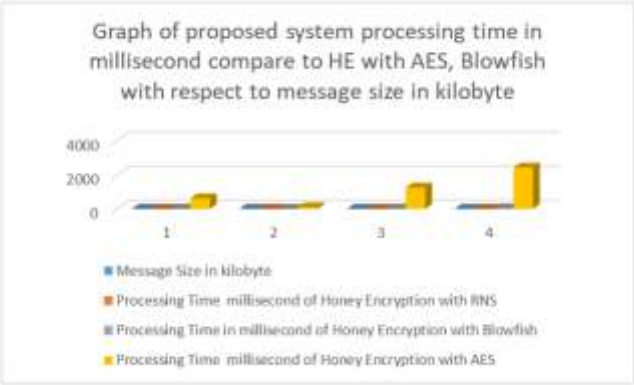

Figure 10: Graph of Proposed System Processing Time in Millisecond Compared to HE with AES, Blowfish with Respect to Message Size in Kilobyte

Table 4 shows the comparison of the proposed system and HE with AES, Blowfish interm of message length, complexity, security, processing and ciphertext produced.

Table 4: Comparison of Proposed System with the Existing One

System	The proposed scheme	Honey encryption with AES	Honey encryption with Blowfish
Message length	Same	Same	Same
Complexity	Low	High	Low
Security	More secure	Secure	Secure
Execution time in millisecond	0.18	7.5	625
Ciphertext	Residue	Gibberish	Gibberish

The proposed encryption method ensures security of text data and displays fake but meaningful message to attacker for every incorrect key by switching the message action words with the inverse that matches the message space. With Residue Number System, it takes lesser time to execute with high computational power.

5. CONCLUSION

Due to vulnerability of most of the data security techniques to brute-force attack, the study proposed a hybridized Honey Encryption with Residue Number System to secure data transmitted through unsecure medium (internet). The scheme enhanced capability of honey encryption in protecting to compared other type of data as numeric. ASCII table was used to convert the text, while Distribution Transformation Encode (DTE) encodes the message by mapping the action with dictionary table XORed, to get the perfect or opposite of the word that matched the seed in the message space after which traditional moduli set was used to generate the key and produce the residue. Chinese Remainer Theorem and Inverse DTE (I-DTE) were used in decrypting the message to get the plaintext

When the attacker attempts to access the encrypted data with his/her guess key, instead of

rejecting their data access as conventional data encryption scheme, the HE algorithm generates an indistinguishable bogus message that are closely related to the original message, so long as the wrong key matches with the length of the original key. Compared to the existing hybridized honey encryption, the proposed scheme processing time is faster and more secured because the result of cipher-text produced is complex and to decrypt it, it requires lot of effort such as: finding the sequence of algorithms used, determine the moduli set used, determine the value of n for the moduli set and also the character table. In the future, several aspects of this work can be further explored, such as extending it into file, folders, image and video protection and also possibly, on cloud data storage**.**

REFERENCES

Omolara, A. E., & Jantan, A. (2019). Modified honey encryption scheme for encoding natural language message. *International Journal of Electrical and Computer Engineering (IJECE)*, *9*(3), 1871-1878.

Adams, C., Jourdan, G. V., Levac, J. P., & Prevost, F. (2010, August). Lightweight protection against brute force login attacks on web applications. In *2010 Eighth International Conference on Privacy, Security and Trust* (pp. 181-188). IEEE.

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.

Aremu, I. A., & Gbolagade, K. A. (2017). An overview of Residue Number System.International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),6(10), 2278-1323.

Juels, A., & Ristenpart, T. (2014). Honey encryption: Encryption beyond the brute-force barrier.IEEE security & privacy,12(4), 59-62.

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.

Bello, H., & Gbolagade, K. (2017). An efficient CRT based reverse converter for {22n+ 1-1, 2n-1, 22n-1} Moduli Set.Journal of Advances in Mathematics and Computer Science,25(6), 1-9.

\Merkow, M. S., & Breithaupt, J. (2014).Information security: Principles and practices. Pearson Education.

Kim, C. K., Kim, J. T., Yu, C. Y., & Kim, J. H. (2011). A Mechanism of Medical Data Encryption Method Using Bucket Index and Bloom filter with the range property.Journal of the Korea Institute of Information and Communication Engineering,15(2), 371-381.

Dharshini, P., Arokia Renjith, J., & Mohan Kumar, P. (2016). Screening the covert key using honey encryption to rule out the brute force attack of AESa survey.Security and Communication Networks,9(18), 6379-6385.

Mok, E., Samsudin, A., & Tan, S. F. (2017, February). Implementing the honey encryption for securing public cloud data storage. InFirst EAI International Conference on Computer Science and Engineering(pp. 272-280).

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.

Juels, A., & Ristenpart, T. (2014). Honey encryption: Security beyond the brute-force bound. InAdvances in CryptologyEUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings 33(pp. 293-310).

Springer Berlin Heidelberg.

Isupov, K. (2021). High-performance computation in residue number system using floating-point arithmetic.Computation,9(2), 9.

Panchal, K. (2020). Differential privacy and natural language processing to generate contextually similar decoy messages in honey encryption scheme.arXiv preprint arXiv:2010.15985.

Lawal, T. D. (2022).Development of Lossless Data Compression Algorithms Using Residue Number System(Doctoral dissertation, Kwara State University (Nigeria).

James, L., & Dileesh, E. D. (2020). Technique to Thwart Brute-Force Attack: A Survey.Lindholm, R. (2019).Honey Encryption: implementation challenges and solutions(Master's thesis).

Taghizadeghankalantari, M., & TaghipourEivazi, S. (2021). Design of efficient reverse converters for Residue Number System.Journal of Circuits, Systems and Computers,30(08), 2150141.

Mok, E., Samsudin, A., & Tan, S. F. (2017, February). Implementing the honey encryption for securing public cloud data storage. InFirst EAI International Conference on Computer Science and Engineering(pp. 272-280).

Murty, S., & Mulchandani, M. (2017). Improving security of honey encryption in database: Implementation (ICSESD).

Chervyakov, N., Lyakhov, P., Babenko, M., Nazarov, A., Deryabin, M., Lavrinenko, I., & Lavrinenko, A. (2019). A high-speed division algorithm for modular numbers based on the Chinese remainder theorem with fractions and its hardware implementation.Electronics,8(3), 261.

Gaid, M. L., & Salloum, S. A. (2021, May). Homomorphic encryption. InThe International Conference on Artificial Intelligence and Computer Vision(pp. 634-642). Cham: Springer International Publishing.

Meghashree, B. S., & Sujatha, B. R. (2018). AES based image encryption and decryption using Matlab.International Journal of Engineering Research & Technology NCESC (IJERT),6(13).

Omolara, A. E., Jantan, A., & Abiodun, O. I. (2019). A comprehensive review of honey encryption scheme.Indonesian Journal of Electrical Engineering and Computer Science,13(2), 649-656.

Popoola, D. D. (2019).Data Integrity Using Caesar Cipher and Residue Number System(Doctoral dissertation, Kwara State University (Nigeria).

Al Badawi, A., Polyakov, Y., Aung, K. M. M., Veeravalli, B., & Rohloff, K. (2019). Implementation and performance evaluation of RNS variants of the BFV homomorphic encryption scheme.IEEE Transactions on Emerging Topics in Computing,9(2), 941-956.

Sahu, R., & Ansari, M. S. (2017). Securing messages from brute force attack by combined approach of honey encryption and blowfish.International Journal, 4.

Tan, S. F., & Samsudin, A. (2018). Enhanced security of internet banking authentication with extended honey encryption (XHE) scheme.Innovative Computing, Optimization and Its Applications: Modelling and Simulations, 201-216.

Stamenkovic, N., Jovanovic, B., & Stojanovic, V. An Improved Residue to Binary Converter Based on Mixed-Radix Conversion for the Moduli Set {22n− 1, 22n, 2n− 1}.

Tan, S. F., & Samsudin, A. (2018). Enhanced security of internet banking authentication with extended honey encryption (XHE) scheme.Innovative Computing, Optimization and Its Applications: Modelling and Simulations, 201-216.

Weyori, B. A., Akobre, S., & Armah, G. K. (2009). Application of RNS to

Huffman‟ s Method of Secured Data Encryption Algorithm.International Journal of Soft Computing,4(5), 197-200.

Win, T., & Moe, K. S. M. (2018).Protecting private data using improved honey encryption and honeywords generation algorithm(Doctoral dissertation, MERAL Portal).

Yin, W., Indulska, J., & Zhou, H. (2017). Protecting private data by honey encryption.Security and communication networks,2017.