Asian Journal of Research in Computer Science

8(4): 35-44, 2021; Article no.AJRCOS.68473
ISSN: 2581-8260

Data Hiding in Digital Image for Efficient Information Safety Based on Residue Number System

Joseph B. Eseyin[1*] and Kazeem A. Gbolagade[2]

[1]ICT Directorate, University of Jos, Nigeria.
[2]Department of Computer Science, College of Library and Information Technology Kwara State University, Malete Ilorin, Nigeria.

Authors' contributions

This work was carried out in collaboration between both authors. Author JBE designed the study, performed the statistical analysis, wrote the protocol, and wrote the first draft of the manuscript with the supervision of Author KAG. Author JBE and KAG carried out and managed the analyses of the study. Author JBE managed the literature searches and approved by author KAG. Both the authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2021/v8i430208
Editor(s):
(1) Dr. Hasibun Naher, BRAC University, Bangladesh.
Reviewers:
(1) Kritika Gautam, The Indian Cambridge School, India.
(2) Rajesh Kumar Bunkar, Awadhesh Pratap Singh University Rewa, India.
(3) Awanit Kumar, Rajasthan Technical University, India.
Complete Peer review History: http://www.sdiarticle4.com/review-history/68473

Original Research Article

Received 10 March 2021
Accepted 14 May 2021
Published 18 May 2021

ABSTRACT

The mass dispersal of digital communication requires the special measures of safety. The need for safe communication is greater than ever before, with computer networks now managing almost all of our business and personal affairs. Information security has become a major concern in our digital lives. The creation of new transmission technologies forces a specific protection mechanisms strategy particularly in data communication state.

We proposed a steganography method in this paper, which reads the message, converting it into its Residue Number System equivalent using the Chinese Remainder Theorem (CRT), encrypting it using the Rivest Shamir Adleman (RSA) algorithm before embedding it in a digital image using the Least Significant Bit algorithm of steganography and then transmitting it through to the appropriate destination and from which the information required to reconstruct the original message is extracted. These techniques will enhance the ability to hide data and the hiding of ciphers in steganographic image and the implementation of CRT will make the device more efficient and stronger. It reduces complexity problems and improved execution speed and reduced the time taken for processing the encryption and embedding competencies.

**Corresponding author: Email: eseyinjb@unijos.edu.ng, fxeseyin@gmail.com;*

Keywords: Text hiding; digital image; security; encryption; least significant bits.

1. INTRODUCTION

With computer networks now handling almost all of our business and personal affairs, the need for secure communication is greater than ever [1]. Security of information has become an important issue in our digital life. The creation of new transmission technologies forces a specific protection mechanisms strategy particularly in data communication state [2].

Day by day, the importance of network protection is increased as the amount of data being transmitted over the Internet [3]. Cryptography and steganography provide the most significant information security techniques [4]. The attacker's most powerful reason to profit from intrusion is the importance of the confidential data that he or she will gain by breaching the system [2]. Hackers may expose the data, alter it, distort it or use it for harder attacks [5]. A solution to this problem is to use the combination of cryptography with steganography advantage in one system [6,7].

Most current steganographic tools may offer hiding of perceptually indiscernible data, the stochastic prominence or unauthorized detection capability of hidden data remains a stimulating challenge. Stochastic visibility may be regarded as the possibility of unauthorized detection, based on hypothesis testing, to distinguish between the cover and host results [8]. A collection of criteria should be met by the steganographic system; the key requirement is to provide statistical indistinguishability between the cover data and host data.

Information hiding can be gotten in four stages namely: -

(i) preliminary step of implementation of an encryption technique.
(ii) Embedded process, in which an information hiding algorithm is used.
(iii) The transmission process and lastly
(iv) the step of extraction [1].

There has to be a proper security in each step so that sensitive correspondence, copyright protection, authentication is enhanced in different applications. Hiding information using the combination of cryptography and steganography is better than only ciphering because in the former, no one can tell that a message is hidden behind an image [8].

In steganography, carrier medium is the object carrying the secret information which is identified as the object. Stego-object is the resulting steganography output, which is transmitted to the target destination. Stego-key is the key to remove the secret data from the stego-object. Digital carriers are flexible where data can be covered.

Data can be inserted into: audio file where data may be obscured in the form of echoes or slightly altered in the amplitude of the signal, or inserted in an audio file or noise. Information can be concealed by changing the location of the lines or words in the text. In design or image, data can be obscured by manipulating image properties such as luminescence, or contrast and color.

The most common type carrier used with steganography is a digital image. An image can be represented as an array of numbers which are called pixels at various color points, representing high intensity [7]. The size of an image in pixels will be given. Pixels are indexed by coordinates x and y with values x and y for integer. In general, each pixel is stored as 24 bits or 8 bits. A 24-bit picture is distributed over three bytes, with each byte representing red, green, and blue. Colors are obtained by combining varying amounts of red, green and blue light [7].

In this paper we propose a protection mechanism that hides a message text into a digital image by first converting into its residue number system equivalent [9], encrypting the text using the RSA encryption algorithm, embedding the cipher text in the LSB of an image before transmitting it to the destination over the network[10].

Residue Number System (RNS) is a non-weighted number system which Garner proposed back in 1959 to accomplish rapid implementation of addition, subtraction and multiplication operations in special purpose computations. Unfortunately, in those days RNS has not turned out to be a popular alternative to the two-complement number system [9].

The rigidity of instruction set by the market-dominant computers and microprocessors architectures was then the main barrier for sustaining RNS-based applications development.

Technological advancement in semi-conductor technology has revived the interest in reconsidering RNS for application-specific computing in recent years. There are at least two specific motivations in modern digital signal processing applications which make RNS computations more appealing and applicable [1].

Firstly, residue number system modular and distributive properties are used to achieve performance improvements, especially in emerging distributed and universal computing platforms such as cloud, wireless ad hoc networks, and soft error tolerance applications.

Secondly, energy efficiency becomes a key driver in the continuous classification of digital integrated circuits consisting of complementary metal oxide semiconductor (CMOS).

The residue number system's high degree of computational parallelism provides a new degree of freedom for optimizing energy efficiency, particularly for arithmetic with very long word lengths, such as those used in cryptographic algorithm hardware implementation. RNS is based on a puzzle introduced by the Chinese mathematician Sun-Tzu, later referred to as the Chinese Remainder Theorem (CRT) [1]; It has some fascinating theoretical number properties and special characteristics that can be used to improve the speed of some electronic computations [11].

In steganography, masking & filtering techniques and the least significant bit (LSB) substitution are the most well-known techniques for hiding data in images. LSB is an easy way of embedding information in an image. By applying LSB technique to each byte of a 24-bit image, three bits can be encoded into each pixel, since each pixel is represented by three Bytes. Only one bit of each pixel can be encoded by applying the LSB technique to each byte of an 8-bit image, since each pixel is represented by one byte [12]. The Least Significant Bit (LSB) embedding is a simple technique for the implementation of steganography. Like all types of steganography, it incorporates the data into the mask, so it can not be identified by a casual observer.

The technique works by replacing some information stored in a given pixel with image data. Although data may be embedded on any bit-plane into an image. LSB embedding is done on the least significant bit(s) [12]. That minimizes the color variation generated by the embedding. The decryption and the extraction process at the destination is performed in a converse mode to the encryption and the embedding process.

The remaining part of this paper is organized into the following sections, section 2 talks about the related work while in section 3 we enumerated the algorithm for the scheme, our proposed algorithm, the slice creation algorithm and slice stacking algorithm. And the analysis of experimental result is given in section 4 and section 5 concludes the paper.

2. RELATED WORKS

Information hiding is a class of processes that are used to manipulate data in a way that the data should be imperceptible. The methods used to hide data differ depending on the nature of the data being hidden and the invariance needed for manipulation of such data. Data hiding should be capable of embedding data under the following conditions: that an outsider does not detect the presence of the data, the embedded data should be inserted directly into the media and the data should remain intact across various types of data files [8]. Finally, the embedded data should be resistant to changes that range from foreign and intelligent removal attempts to expected manipulations such as printing, scanning, analog to digital converters. In [2] Scott explains how every note corresponds to a letter to conceal messages in music scores.

Scott used the "Ave Maria" code in steganography, which involved the use of forty tables, each with 24 entries one for each letter, where each letter in the plaintext is swapped by the word that appears in the equivalent table entry, giving the plaintext the appearance of being a prayer.

It is possible to hide data in an image by changing the least Significant bit (LSB). In [3], two methods for hiding information based on LSB are suggested. One that replaces LSB with the sequence of Pseudo-noise (PN) and the other adds a sequence of PN to LSB. Another LSB data hiding method called "PatchWork" is proposed in [4], which selects n pairs (ai, bi) of points in an image and increases ai's brightness by one unit while at the same time reducing bi's brightness.

The concept on which PatchWork is based is that given two points A, B randomly selected from an

image, consider A's brightness and B's brightness so S= a-b must be equal to zero. Therefore, the standard deviation to S is 0 on its predicted value. The same results must be given when repeating the calculation of s different times. When it differs by more than a few standard deviations, this means this did not happen by chance but shows the existence of encoding to a high degree of certainty.

In PatchWork a series of artificial modifications are made to encode the information in a way that does not deviate from the expected value by the brightness of a number of nodes selected randomly (ai, bi). In a mode that is offhandedly dispersed by the drive in the image, the patch shape used for information encoding can be rectilinear, hexagonal, or random.

The downside of PatchWork algorithms is that the embedded data rate is small.

In [6], the authors offer a framework for data hiding that integrates information about three images into one image based on the Least Significant Bit (LSB). In which the two least significant bits of each pixel in the image are changed, each will contain the result after application of the edge detection filter, before and after connectivity of the gray scale stage. For three different images one pixel contains details. For data embedding, the value zero for the non-edge pixel or one for the edge pixel replaces the LSB of each pixel.

This is achieved by the use of the logical operators. So, the LSB provides the sign that an edge pixel exists. LSB extraction can be applied by checking the odd values and even the pixel values. After implementation of the pixel connectivity, the prior LSB is used to contain the binary edged picture. The initial picture in this case occupies the remaining 6 bits.

In [11], a steganographic method is suggested that divides the cover image into blocks of the same size and then the message is inserted in the edge of the block depending on the number of four pixel bits in the center. The embedding method is done as follows: a set of pixels, which will be used to cover the data, is selected first. The selected pixels 'gray level values are then changed to make them even, and this will represent 0. To represent 1, the gray level of the correct pixel is decremented by one. The embedding algorithm begins by splitting each pixel into two equal parts, it often counts the number of 1's, and it embeds a hidden message in the least part according to the number of bits correspondingly.

In [12], a technique for hiding defined data set in an image is proposed as follows, the image is considered as a matrix, and the data to be concealed is considered as a secret key which is a matrix of size m*n. The resulting hidden data is embedded in the preceding image, given a set of sums and bitwise AND operations between the two matrices image and secret key. The drawback in this strategy is to use binary AND, and the shift must take place at the positions where the hidden key has a value in 1.

3. THE ALGORITHM

This scheme is conceived with three levels of security. The conversion method to the Residue Number System uses the length three moduli set [10]. Then the three methods, the simple RSA public key cryptography, the application of CRT to minimize the time for modular exponentiation, the message to be sent is converted into RNS using the Chinese Reminder Theorem (CRT). Two keys are used which are public key and private key [9]. It is to ensure the key sharing process is swifter. Similarly, we are giving the RSA cryptosystem a boost by offering a scheme with a speed increase on the RSA decryption side by using the Chinese Remainder Theorem and also providing security by using two key pairs instead of a single public key [13].

And by inscribing the message in steganography by obscurity improved the message's security. The sender's give the receiver the public key. The receiver obtains the message sent in and encrypted form that is embedded in a cover image as a stego image [7]. The receiver extracts the stego picture using the private key that is sent to him, and the cipher text is deciphered to get the message.

3.1 The Proposed Algorithm

First, the proposed protection system uses LSB technique to conceal the secret message with the restriction that the cover of the image is larger than the secret text. Cryptography with steganography is a data securement technique. After the message is divided into 'n' bits, the individual stocks are sent to the destination through different communication channels, so that the attacker has less chance of obtaining full information.

We suggest a solution to the above security problem by encrypting the message using the RSA encryption. RSA is the most widely used asymmetric algorithm because it ensures the confidentiality, completeness, and accuracy of electronic communications and data storage. This method is used as well as creating a share before embedding. If intruder gets all the share now, he or she may not get any of the information because Secret message is encrypted itself. Our novel cryptostego technique is an n out of n method in which a secret message after RSA process encryption is split into 'n' shares, and we must have 'n' shares to decrypt the secret message. Special mixed-key generation (MKG) technique is used to generate the keys. This method produces size block of 8-byte keys and selects individual bits from each byte, as we have 8-byte word that we can perform parallel operation with 8 byte of source data. Fig. 1 shows the main technique for key generation. Each text is encrypted to form a Ciphertext by taking the keys generated by the MKG process. Since we can make 8 keys at a time this will enhance the efficiency of producing ciphertext.

This algorithm is a method of steganography that hides text in an image so that only the permitted destination can read the information embedded in the image. The algorithm works on two parts. The first is the authorized source which is the sender of the message and the authorized destination. The sender carries out the preliminary, encryption, and transmission phase and the destination carries out the decryption. For this, the encryption and decryption phases are linked to each other in a way that does not take much time to retrieve the hiding message from the approved destination.

3.1.1 Slice creation algorithm

Residue Number System based on the Chinese Remainder Theorem paradigm is used for stock formation. Since the proposed idea is n out of n approach, we've generalized it to 3 out of 3 approaches where 3 shares are generated and all 3 shares are required to get back the original message. Chinese Remainder Theorem can be used to stack the shares. The cycle of share-generation is as shown in Fig. 2.

The algorithm is as given below:

Step 1: Select 3 prime numbers m1,m2,m3 so that their product exceeds 255 and gcd of selected 3 numbers is 1 that is they are relatively prime.

Step 2: compute

Si1=X mod m1
Si2=X mod m2
Si3=X mod m3

Where, Si1, Si2, Si3 are residues of i^{th} pixel;
X is an individual pixel; m1, m2 and m3 which are selected prime numbers.

Step 3: Denote the residues Si1, Si2, Si3 as i^{th} pixel of share 1, 2 and 3 correspondingly.

Step 4: Redo step 2 and 3 until all pixels are processed.

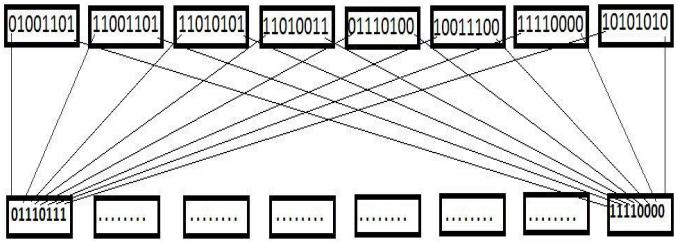

Fig. 1. Key Generation

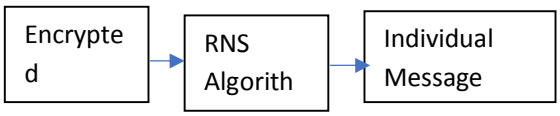

Fig. 2. Share creation process.

3.1.2 Slice stacking algorithm

Chinese Remainder Theorem principle is used for share stacking method. The entire process is shown in Fig. 3.

The block stacking algorithm is as follows:

Step 1 : Calculate the dynamic range M=m1.m2.m3
Step 2 : Calculate Ci = M/mi
Step 3 : Find the solution of congruence's

Ci.Pi mod mi

Where

Pi is multiplicative inverse of Ci

Step 4: Using the equation below we can get original pixels by CRT

x = $\left(\sum_{i=1}^{N} Ci\ Pi\ si\right)$ mod n

Step 5: Redo step 4 until all pixels of shares are processed.

4. EXPERIMENTAL RESULT AND DISCUSSION

The experimental setup was carried out and developed with the Matlab programming (MATLAB R2015a (8.5.0, 197613) 64bit (Win64)).

Different modules/functions were developed and linked to a graphical user interface for user interactivity and responsiveness. The developed systems made use of various element in MATLAB environment to develop and output result. The new system stressed the effect of Residue Number System on data hiding in image using the least significant bit and RSA Encryption for first stage data security.

The system was experimented with samples text messages which were inputted from a text file. The text message was converted into its ASCII code equivalent from which RSA encryption with CRT were carried out. The encrypted text is passed to the Residue Number System with the moduli set of $m_1 = 2^{n+1} - 1, m_2 = 2^n\ and\ m_3 = 2^n - 1$. The forward conversion process was used to further encrypt the text which was spitted into three residues before hiding the text in a cover image to form a stego image which was sent to the receiver.

The text is embedded in a cover images using the least significant bit technique of steganography to hide the message. The experiment reveals that the embedded text with RSA-CRT has faster computational time as compared to the one with RSA without CRT. The decryption stage was initiated by supplying the moduli set that was used for the process of reverse conversion, the reverse conversion is used to decrypt the message which combines the three residues to singular message producing the decrypted message, after which the RSA decryption is triggered with the same cipher key used for encryption which brings the inverse of the embedded message or which performs the extraction process to bring out the message that was embedded.

The system was tested over sample images by granularity which is an observable aspect of the code, that is typically defined in a subjective manner. For example, coarse granularity that is also called coarse-grain methods will typically calculate the per-process, per-procedure, or per-function time of execution. The obtained results are shown below for both the RSA without CRT and RSA with CRT. The tables show results of the time of encryption and decryption, embedding time and extraction time

Table 1 above shows the RSA encryption time and the RSA-CRT encryption time. And from this table we were able to compare the time taken for encryption with RSA alone and RSA with CRT. And from the result it was seen that the application of Chinese Remainder Theorem on RSA encryption drastically reduced the encryption time. This can be seen pictorially as shown in Fig. 1.

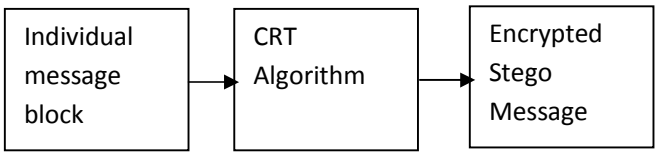

Fig. 3. Share stacking process

Table 1. Comparison of RSA encryption time with RSA-CRT encryption time

Cover image	RSA: Encryption time(s)	RSA - CRT: Encryption time(s)
Eagle	8.61868	0.695118
Monkey	8.63074	0.250232
Gold Fush	8.50735	0.267418
Pepper	8.54858	1.22769
Lena	8.49326	0.235429
Babooon	8.66876	0.267105
Redroses	8.5192	0.223229
Dog	8.4809	0.22172

Table 2. Comparison of RSA decryption time with RSA-CRT decryption time

Cover image	RSA: Decryption time(s)	RSA-CRT: Decryption time(s)
Eagle	8.28869	0.260270
Monkey	8.34312	0.218986
Gold Fush	8.28792	0.225894
Pepper	13.2226	0.235411
Lena	8.21274	0.238732
Babooon	8.24501	0.251824
Redroses	8.27723	0.239798
Dog	8.21651	0.223150

The decryption time with RSA without CRT was compared with the decryption time of RSA with CRT. And it was so glaring that there was a drastic reduction in time used for the decryption process with the application of Chinese Remainder Theorem. This was further presented graphically as shown in Fig. 5.

Table 3. Time taken for embedding for RSA and RSA-CRT

Cover image	RSA: Embedding time(s)	RSA-CRT: Embedding time(s)
Eagle	8.710210	1.04722
Monkey	8.758998	0.718713
Gold Fish	8.771061	0.70247
Pepper	8.722820	1.23352
Lena	8.756230	0.75759
Babooon	8.718631	0.721991
Redroses	8.729661	0.709288
Dog	8.805752	0.735144

The embedding time of text in image using RSA without the application of CRT takes a longer time as compared to the time taken for RSA with CRT. This was portrayed in Table 3 and graphically represented in Fig. 6.

Extraction time is the time taken for the original text to be taking out from the embedded image. The total time taken to do this for RSA without CRT is quite large compared to the time taken for and RSA-CRT. This was as represented in Table 4 and graphically represented in Fig. 7.

Since the proposed algorithm is intended to be ideal for online applications, it is important to reduce the time required to insert the message into the cover image as well as the time needed to recover the secret message that our solution has been able to achieve.

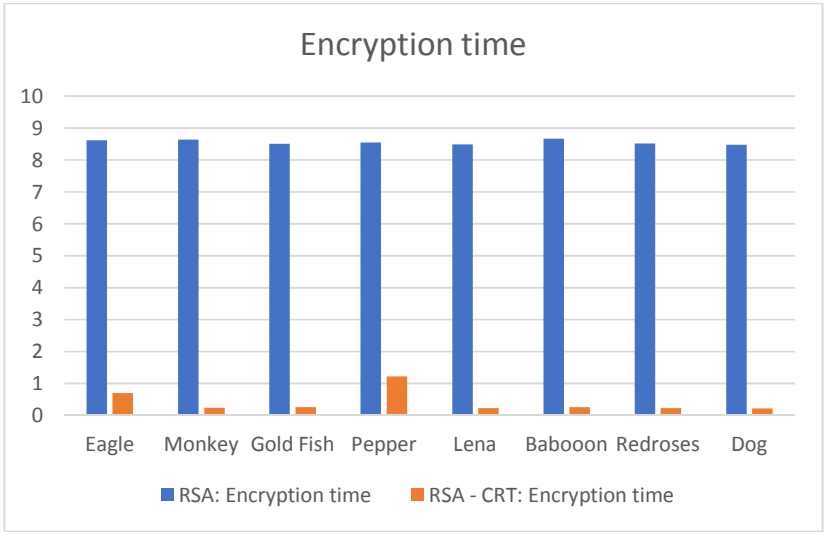

Fig. 4. Chart showing the time for encryption using RSA and RSA-CRT

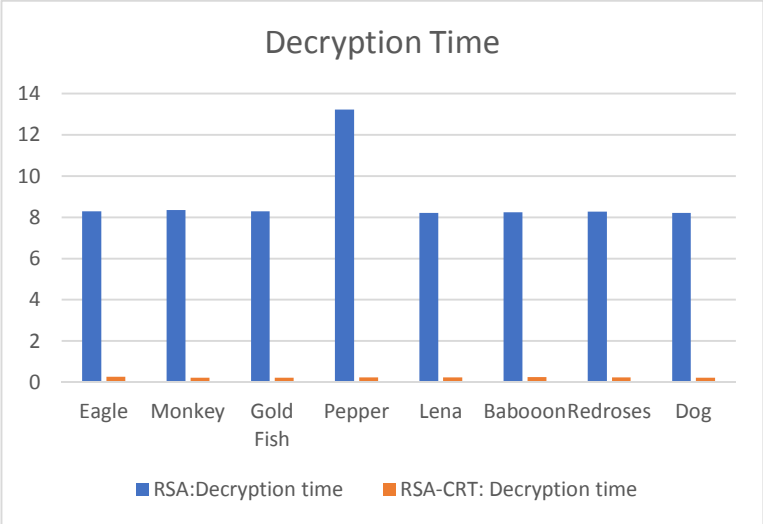

Fig. 5. Chart showing the time for decryption using RSA and RSA-CRT

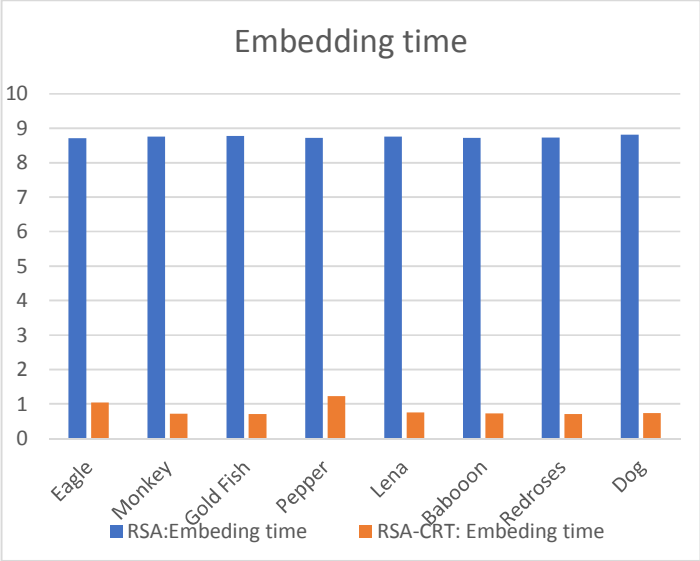

Fig. 6. Chart showing embedding time with RSA and RSA-CRT

Table 4. Time taken for extraction with RSA and RSA-CRT

Cover image	RSA: Extraction time(s)	RSA-CRT: Extraction time(s)
Eagle	5.16450	0.057799
Monkey	5.02793	0.0047456
Gold Fish	5.04051	0.005963
Pepper	5.01641	0.1231211
Lena	5.01482	0.0045289
Babooon	5.02233	0.0049606
Redroses	5.01106	0.0371686
Dog	5.01943	0.0050491

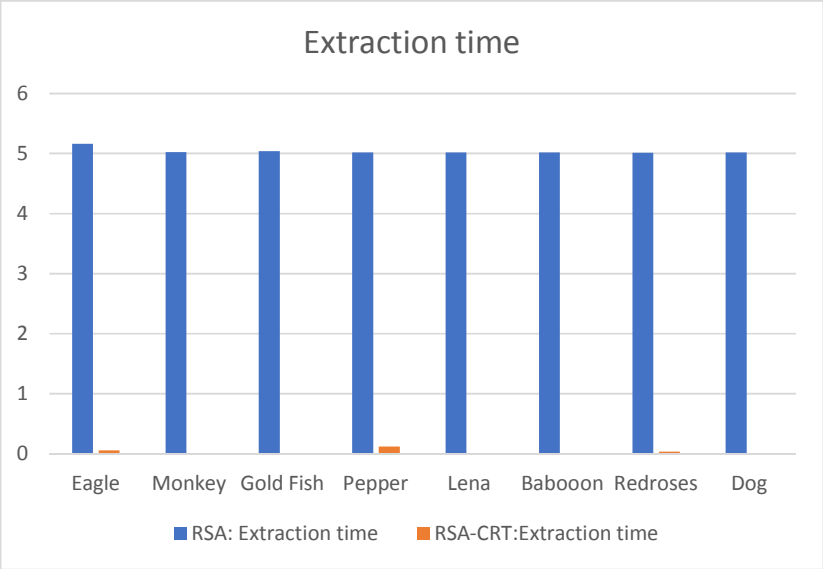

Fig. 7. Chart showing the extraction time with RSA and RSA-CRT

5. CONCLUSIONS

Data protection from unauthorized access is a major concern in today's fast-paced communications biosphere. One of the most effective techniques we discussed shows the combination of the most efficient RSA and LSB algorithm resulting in the most stable and secure systems. Hiding data in the cover image has a bigger advantage.

Using quick RSA-CRT techniques to solve decryption slow pace. We used the technique of RSA encryption for protection on two layers and we found that decryption of RSA-CRT is faster than the regular RSA. The application of RSA-CRT in the decryption process also made the message that is embedded in the cover image and the extraction time faster than the normal RSA as seen from the tabulated results as presented.

Analytical comparison of RSA and RSA-CRT decryption algorithms is designed and implemented to identify the RSA and RSA-CRT decryption algorithm deficiencies. Initial results showed decryption with RSA using CRT definitely and significantly improves decryption performance.

By identifying the disadvantages of RSA and RSA-CRT, this research will serve as a basis for future research to improve the speed of RSA decryption. More research and testing could lead to an improvement in the current Internet world in terms of adding more robustness without compromising protection.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Saheed YK, Gbolagade KA. Efficient RSA cryptosystem decryption based on chinese remainder theorem and strong prime. Anale. Seria Informatică. Tibiscus University Journal, Romania. 2017;XV fasc.(2):43-47.
2. Rahmani MKI, Kamiya Arora NP. A crypto-steganography: A survey. International Journal of Advanced Computer Science and Application. 2014;5:149–154.
3. Karthik JV, Reddy BV. Authentication of secret information in image stenography. International Journal of Computer Science and Network Security (IJCSNS). 2014;14(6):58.
4. Rajyaguru MH. Crystography-combination of cryptography and steganography with rapidly changing keys. International Journal of Emerging Technology and Advanced Engineering. 2012;2250–2459.
5. Seth D, Ramanathan L, Pandey A. Security enhancement: Combining cryptography and steganography.

International Journal of Computer Applications (0975–8887); 2010.

6. Abdulzahra H, Ahmad R, NOOR NM. Combining cryptography and steganography for data hiding in images. ACACOS, Applied Computational Science. 2014;978–960.
7. Mustafa Sabah Taha, Mohd Shafry Mohd Rahim, Sameer abdulsattar Lafta, Mohammed Mahdi Hashim, Hassanain Mahdi Alzuabidi. Combination of steganography and cryptography: A short survey. 2nd International Conference on Sustainable Engineering Techniques (ICSET 2019) IOP Conf. Series: Materials Science and Engineering. 2019; 518:052003.
8. Antonio H, Prasad PWC, Alsadoon A. Implementation of cryptography in steganography for enhanced security. Multimed Tools Appl. 2019;78:32721–32734. DOI:https://doi.org/10.1007/s11042-019-7559-7
9. Eseyin, Joseph B, Gbolagade, Kazeem A. A residue number system based data hiding using steganography and cryptography**.** KIU Journal of Social Sciences, [S.l.]. 2019;5(2):345-351. July. ISSN 2519-0474.
10. Eseyin, Joseph B, Gbolagade, Kazeem A. An overview of public key cryptosysems and application of residue number system. KIU Journal of Humanities**,** [S.l.]. 2019;4(2):37-44, ISSN 2522-2821**.**
11. Saheed YK, Gbolagade KA. RSA cryptosystem encryption based on three moduli set with common factors {2n+2, 2n+1, 2n}. Computing and Information Systems Journal, University of the West of Scotland, USA. In Press; 2018.
12. Samer A. A new algorithm for hiding gray images using blocks. Information, Security Journal, The Hashemite University, Jordan. 2006;15(6).
13. Jose M. Hiding image in image using LSB insertion method with improved security and quality. International Journal of Science and Research. 2014;3(9):2281-2284.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sdiarticle4.com/review-history/68473