Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 28 (2025), No. 3, pp. 753–764
DOI : 10.47974/JDMSC-1946

A novel security algorithm for text and image data using multi node cryptographic algorithm

A. K. Mohapatra *
Department of Information Technology
Indira Gandhi Delhi Technical University for Women
Delhi 110006
India

Pankaj Lathar †
Department of Computer Science and Engineering
Delhi Skill and Entrepreneurship University
Dwarka 110077
Delhi
India

Shailendra Singh Gaur §
Department of Information Technology
Bhagwan Parshuram Institute of Technology
Rohini 110089
Delhi
India

Abstract

In today's world of Ubiquitous Computing, Network security plays an important role in preventing and reducing the chances of various network attacks and working on the various cryptographic algorithms to reduce the risk of getting affected. It is also a challenging situation to reduce the energy consumption of nodes and increasing the life time of sensor nodes. In this paper, Threshold Cryptography is studied and a novel cryptographic algorithm using multi node cryptography for text and image data is proposed. The proposed algorithm is an asymmetric cryptographic algorithm, which uses a key of variable length and encrypts the text and image files (all formats) and provides 2 factor encryption tothe data.

* *E-mail:* akmohapatra@igdtuw.ac.in (Corresponding Author)
† *E-mail:* pankaj.lathar@dseu.ac.in
§ *E-mail:* shailendrasinghgaur@bpitindia.com

The proposed algorithm is fast, lightweight, simple and offers a 2 factor protection to the data which makes it suitable for low resource devices.

Subject Classification: *94A60 Cryptography.*

Keywords: *Threshold cryptography, Proposed algorithm, Implementation, Block diagram, Grey bit encryption, Ascii bit map.*

1. Introduction

In today's world, with the rise of Internet Of Things, everything is doing some sort of communicationwith one and other. Thus the devices which contain low resources to work often compromise the integrity ofdata and tend to communicate in clear text[2]. This compromises the security of data and the user that isusing it. Thus a cryptographic algorithm is required that can use the use the cluster nature of these nodes andprovide a higher standard of security.

Network Security is used to protect the important and information of the client for accessing the reliable connection and at the same time protect it from the threats. Various organizations follow the solution and deployed the solutions for the network security for securing the information data. Various uses of Network Security in organization include business operations and customer services etc.[12]

The proposed algorithm meets these demands and provides a simple solution. The proposed algorithm workssimilar to the Threshold Cryptographic Algorithm but is lighter on resources, faster in implementation andprovides a *2 factor security*. It distributes the key in all the nodes in the cluster thus for data to be decryptedall keys from all the nodes are required. It also uses a random generated ascii map which is produceddifferent for each codebase. This provides an added layer of security to the data as an attacker will have tocompromiseallthe nodes in order to get the clear text data and will have to leak the code for acii_map value. This makes it more suitable for implementation in multi cluster environments.

1.1 *Cryptography*

Cryptography is a field of computer science that uses the application of mathematical theorems and logic to produce unbreakable encryption methods[1]. Cryptography allows communication between two nodes orcomputers secure in electronic world. It allows people can perform

communication over the internet withoutworrying about their data might get compromised [8], [11].

Cryptography has been used forthe construction and analysis of protocols that prevent any third-party toaccess the private data that is being communicated between two nodes. Nowadays cryptography is beingused to deal with the encryption and decryption of private data being communicated through the networkingprotocols between the computer systems, which is a branch of cyber security more specifically networksecurity [5], [10].

The cyber crypto graphic algorithms are used to communicate data between nodes with the help of networking protocols so no third-party could compromise theintegrity of the data[2].

1.2 *Thresholdcryptography*

1.2.a Introduction

The Threshold Cryptography algorithm shields the data by encrypting it and distributing it among all the nodes in the network. The data is encrypted with the assistance of a public key, and the pairing private key is distributed to all the nodes participating in that cluster[3].The threshold cryptography requires the threshold number of nodes that were inthe cluster in order to decryptthe encrypted data[3]. In **Shamir's Secret Sharing algorithm, the** Lagrange basis Polynomial **is given as:**

$$f(x) = \sum_{i=0}^{K-1} yili(x) \quad (1)$$

Where, K-1 is Degree, li is Lagrange identities.

1.2.b Implementation

The following shows a simple implementation of the Threshold Cryptographic algorithm in python 3.7.5 as shown in Figure 1. The programs used third-party library threshold_crypto for its implementation which can be found on the git hub site. The script first asks for a text that is to be encrypted and then the number of total nodes and threshold nodes [3], [4]. The program will encrypt the text according to entered information by the user with a2048 bit key and will display the output.

```
#!/usr/bin/env python3

from threshold_crypto import ThresholdCrypto, ThresholdParameters
from base64 import b64encode

f = open('4.jpg','rb')
kay = b64encode(f.read())

m = kay
n = 5
t = 3

key_param = ThresholdCrypto.static_2048_key_parameters()
thresh_params = ThresholdParameters(t,n)
pub_key, key_shares = ThresholdCrypto.create_public_key_and_shares_centralized(key_param,thresh_params)


enc_mssg = ThresholdCrypto.encrypt_message(m.decode('utf-8'),pub_key)
```

Figure 1

Implementation of Threshold Cryptography

II. Proposedalgorithm

A. *Introduction*

The proposed algorithm is an asymmetric image and text based encryption algorithm which uses grey bit xoring and offsetting bytes to achieve encryption. The offsetting of bytes is done through a ascii_map which generated different for each codebase. The algorithm first grey bit xors the image bytes that are stored in the xor (a list object), then it displaces the xor bit by the ascii_map numeric value of the characters in the variable key length password (generated randomly or user entered), this creates an encrypted text that can only be decrypted with the variable length password[7].

Once the text or image files are encrypted, the program converts the variable length password to base64form, which is then divided into the numbers of nodes user wishes to be used in order to encrypt and decrypt the file. Now since the nodes have pieces of base64 encrypted password, decoding the nodes won't give an understandable output.

B. *Working*

1. Proposed Algorithm

The algorithm employs grey bit xoring (that is a practice used in digital circuits to encode the binary bits)and transposition of bytes of image with the variable length key phrase generated by the algorithm similar to vigenere cipher but different approach is taken here[6]. The whole strength of the algorithm depends on the secret key generated by

the algorithm and the order in which each node gets the distributed key, these factorsmust be kept a secret.

First the algorithm converts the image to byte array which can be further manipulated according to theprogrammer's wish. The byte array is then grey bit xor and stored in a list called xor.

The grey bit xoring can be explained by considering 3 bits lets call them a,b, and c also let's consider the the grey bit xor encoded bits as G1, G2, and G3. The xor function is represented as ^ symbol thus grey bit xor can performed as following.

$$a = G1, a \wedge b = G2, b \wedge c = G3 \tag{2}$$

Now we have grey bit xored bytes in a list xor. Once done this all we need to do is generate a long secure strings we could transpose the grey bit xored bytes thus we have a second layer of security with a password. During the transposition we will do simple addition operation on the bytes thus making thealgorithm light and quick even on low resource cluster nodes.

In the code given in the research paper, I have used the python library's random function to generate the variable length key but other random string generators can also be used. We will be converting the character to its integer value, this integer value is once generated with the help of ascii_map algorithm and once generated and used in the code base it needs to be kept a secret. With this randomly generated ascii_map even if the password is compromised the data will have its integrity intact thus it provides an extra layer ofsecurity.

$$\text{Encoding} = \text{xor}[a] + \text{ascii\_map}[key] \tag{3}$$

$$\text{Decoding} = \text{xor}[a] - \text{ascii\_map}[key] \tag{3}$$

This operation is done with each byte saved in xor list and resulting value is also stored in xor. Thus now the xor stores the transposed value of image data which are then written to a file named encrypted and the random pass key generated for this encryption is base64 encoded and broken into 'n' number of keys which are then stored in various nodes.

2. ASCII_MAP Generator

The ascii_map generator used a list called symbols which contains all the symbols you will use in yourvariable length key. One may expand this list according to his/her needs or requirements. The program willassign a random integer to each symbol present in that list and will create a

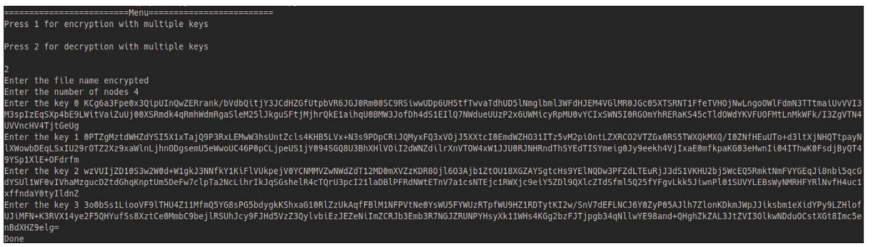


Figure 2

Encryption of Image Data

Figure 3

Decryption of Image Data

dictionary called ascii_map which used in the proposed algorithm to get integer value of the ascii symbol.

This provides an added layer of security as even if the all pass key nodes are compromised the attacker willstill require the bitmap to reproduce the original data[9]. Thus providing a light high security cryptographicalgorithm.

C. *Block Diagram*

1. Encryption

The algorithm takes image bytes then goes through all the bytes and grey bit encrypt them, this adds an extralayer of protection and increases the entropy of encrypted data clearly shown in Figure 2, which further transposed with the variable length key thus making it even more difficult to hack.

2. Decryption

The Decryption works exactly as shown in Figure 3, that reverse as the encryption with one major change, in this process first the password i.e variable length key is generated from the nodes that are entered by the

user then the key is used to de-transpose the data to obtain the grey bit xor form of the data, which is then decrypted to obtain the original image.

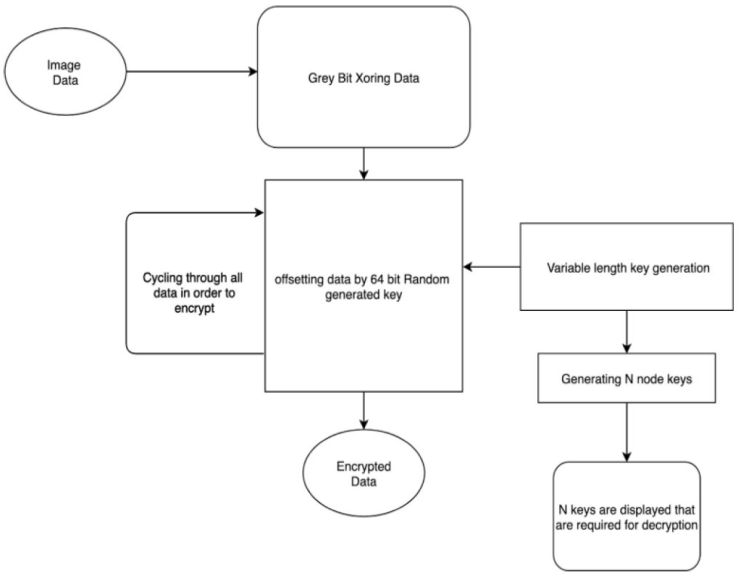


Figure 4

Block Diagram of Encryption Process

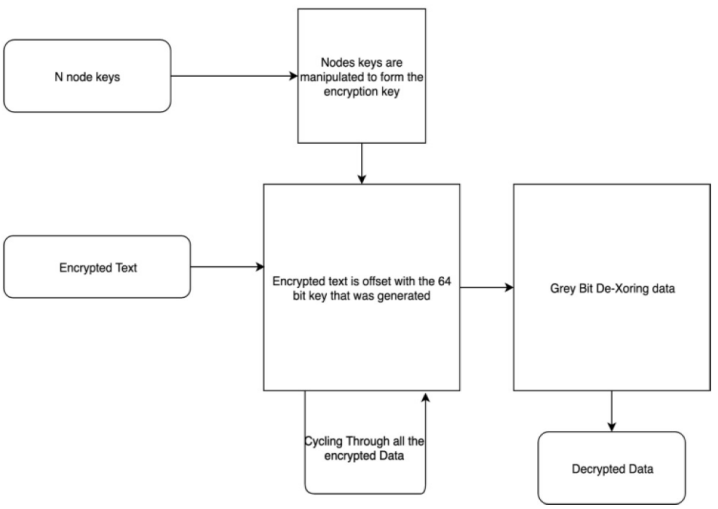


Figure 5

Block Diagram of Decryption Process

D. *Implementation*

Below is the code in which we have implemented the proposed algorithm for encryption as shown in Figure 4 and 6 and similarly for decryption in Figure 5 and 7. The following code is implementedand tested with python v3.7.5 and has been run on computer with the following specs: Operating System: Arch Linux, 4GB DDR3 with i3 Duo Core 64 bit Architecture.

```
def xor_image(a,nodes):

    f1 = open(a,'rb')
    data1 = bytearray(f1.read())

    size = len(data1)
    xor = [None]*size

    letters = string.ascii_letters + string.digits  + '!@#$%^&*()|'
    password = ''.join(random.choice(letters) for i in range(200))
    print (password)
    password = list(password)
    tmp = 0
    xor[0] = data1[0]
    for i in range(1,size):

        xor[i] = data1[i-1] ^ data1[i]
        xor[i] = xor[i] + ascii_map(password[tmp])

        if(tmp<len(password)-1):
            tmp = tmp +1
        else:
            tmp = 0

    f1.close()
    with open('encrypted','wt') as f:
        f.write(str(xor))

    pass2 = ''.join(password)
    b64 = b64encode(str(pass2).encode('utf-8')).decode('utf-8')
    create_nodes(nodes,b64)
```

Figure 6

Code Implementation of Encryption

```
def unxor_image(image,password):
    password = str(password)
    password = list(password)
    f = open(image,'rt')
    c = str(f.read())
    xor = []
    for i in c.split(','):
        try:
            xor.append(int(i[1:]))
        except:
            xor.append(i[1:])
    xor[-1] = int(c.split(',')[-1][:-1])

    size = len(xor)
    data1 = bytearray(size)
    tmp = 0

    data1[0] = xor[0]
    for i in range(1,size):

        xor[i] = xor[i] - ascii_map(password[tmp])

        data1[i] =  data1[i-1] ^ xor[i]
        if(tmp<len(password)-1):
            tmp = tmp +1
        else:
            tmp = 0

    with open('wow.jpg','wb') as f:
        f.write(data1)
```

Figure 7

Code Implementation of Decryption

E. *Performance Analysis*

The following testsare performedusing 4 nodesand 1025bytes of keylength withthe hardware asfollows:

- OS: Arch Linux
- Specs: 8 GBRAM DDR3, i3duo Core CPU

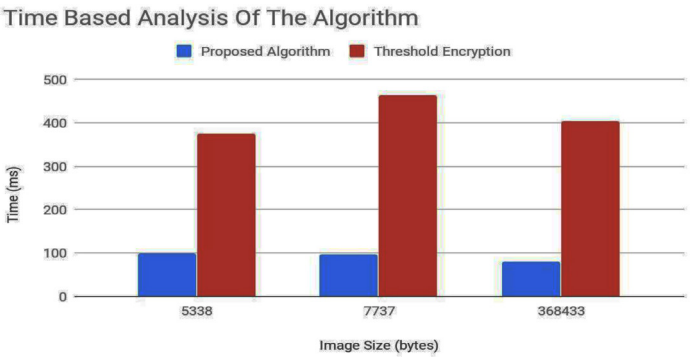
The following imagewas usedin the processof encryption:

- Image type: JPEG image data, JFIF standard1.01

Table 1

Encryption Bytes/time(ms)

Imagesize (Inbytes)	Proposed algorithmtime (Ms)	Thresholden cryptiontime (Ms)
5338	100	375
7737	98	464
368433	80	404



Graph 1

Encryption Time (Bytes/time(ms))

The graph shows the speed analysis of the two cryptographic algorithms i.e the Threshold Cryptographic algorithm and the Proposed algorithm. The x axis show the images size i.e used for the encryption and the yaxis shows the Time taken by the algorithm to do encryption inmilliseconds are shown in the Table 1 in terms of bytes per ms with the proposed and threshold cryptography and Graph 1 depicts the Encryption time..

Note: The Threshold Encryption is combined with *base64encryption* so it is able to encrypt.

The proposed algorithm uses less time because the algorithm employs grey bit xoring and this practice used in digital circuits to encode the binary bits and transposition of bytes of image with the variable length key phrase generated by the algorithm. Secondly, during the transposition we have added simple addition operation on the bytes thus making the algorithm light and quick even on low resource cluster nodes.

III. Futurescope

The proposed algorithm is low on resources and fast in execution and uses the cluster nature of the nodes inorder to increase the security of the data being transferred and stored. This makes it an ideal algorithm forimplementation clustered networks. Thus a working model of network with integrating this algorithm asencryption base will be covered in the future scope.

IV. Conclusion

In conclusion the new proposed asymmetric algorithm shows its perks over traditional symmetric multi node cryptographic algorithms such as threshold cryptography. The proposed algorithm enjoys more speed and simplicity as major advantages and thus can be utilised by low computing power nodes. It also offers 2factor authentication thus has more security than the Threshold Cryptography.

The strength of proposed cryptography algorithm completely depends on the strength of password beingused thus with strong enough password it can be used inpeer to peer network etc.

The algorithm does do anything for encryption of the individual keys created and stored on the nodes so a secondary encryption on the created keys can advance this algorithm further.

References

[1] S. S. Gaur, Dr. A. K. Mohapatra, and R. Roges, "A Secure Approach to Network Security Based on ID Based Encryption Techniques and Their Comparative Analysis," *International Journal of Computer Science*

& Information Technology Research Excellence, vol. 7, no. 4, pp. 1-9, Jul.-Aug. (2017), ISSN 2250-2734 (Print), 2250-2742 (Online).

[2] S. S. Gaur, A. K. Mohapatra, and S. Masood, "Design of an Optimized Novel Cryptographic Algorithm and Comparative Analysis with the Existing Cryptographic Algorithms," *IJCTA*, vol. 9, no. 34 (2016).

[3] K. T. Selvi and S. Kuppuswami, "Enhancing security in Optimized Link State Routing protocol for MANET using threshold cryptography technique," in Proc. 2014 International Conference on Recent Trends in Information Technology, IEEE (2014).

[4] H. Dey and R. Datta, "Monitoring threshold cryptography based wireless sensor networks with projective plane," in Proc. 2012 5th International Conference on Computers and Devices for Communication (CODEC), IEEE (2012).

[5] W. Wang, L. Yu-bing, and C. Tie-ming, "The study and application of elliptic curve cryptography library on wireless sensor network," in Proc. 2008 11th IEEE International Conference on Communication Technology, IEEE (2008).

[6] S. K. Mandal and A. R. Deepti, "A Cryptosystem based on Vigenere cipher by using multi level encryption scheme," International Journal of Computer Science and Information Technologies, vol. 7, no. 4, pp. 2096-2099 (2016).

[7] A. Y. Athavale, K. Singh, and S. Sood, "Design of a private credentials scheme based on elliptic curve cryptography," in Proc. 2009 First International Conference on Computational Intelligence, Communication Systems and Networks, IEEE (2009).

[8] P. Gong and P. Li, "Further improvement of a certificate less signature scheme without pairing," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2083-2091 (2014).

[9] D. V. Chudnovsky and G. V. Chudnovsky, "Sequences of numbers generated by addition in formal groups and new primality and factorization tests," *Advances in Applied Mathematics*, vol. 7 (1986).

[10] A. I. Hamed and S. E. El-Khamy, "New low complexity key exchange and encryption protocols for wireless sensor network clusters based

on elliptic curve cryptography," in Proc. 2009 National Radio Science Conference, IEEE (2009).

[11] L. Levent and W. Lu, "ECC based threshold cryptography for secure data forwarding and secure key exchange in MANET," Springer-Verlag (2005).

[12] M. K. Hasan, M. Shafiq, S. Islam, B. Pandey, Y. A. Baker El-Ebiary, N. S. Nafi, and D. E. Vargas, "Lightweight cryptographic algorithms for guessing attack protection in complex internet of things applications," Complexity (2021).

Received September, 2023