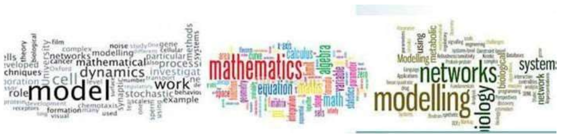

Journal, Advances in Mathematical & Computational Sciences
Vol. 12 No. 3, 2024 Series
www.isteams.net/mathematics-computationaljournal

Journal of Advances in Mathematical & Computational Sciences
An International Pan-African Multidisciplinary Journal of the SMART Research Group
International Centre for IT & Development (ICITD) USA
© Creative Research Publishers
Available online at https://www.isteams.net/ mathematics-computationaljournal.info
CrossREF Member Listing - https://www.crossref.org/06members/50go-live.html

Performance Comparison of Some Asymmetric Data Encryption Algorithms

[1]Oluwasanya, E.O., [2]Olabiyisi, S.O. & [3]Omotosho, O.I.
Department Of Computer Science
Ladoke Akintola University of Technology
Ogbomoso, Oyo State, Nigeria.
E-mail: [1]ekundayoowodogba@gmail.com; [2]soolabiyisi@lautech.edu.ng
[3]oluyinkaa14@gmail.com
Phone: [1]+2348066420481

ABSTRACT

This research compares the performance of different data encryption techniques using cryptographic evaluation metrics. Encryption in cryptography involves encoding data to prevent unauthorized access, transforming plaintext into ciphertext. Many encryption methods exist, but their effectiveness varies and has not been thoroughly assessed. Previous studies have focused on evaluating public-key cryptography, particularly RSA and ECC algorithms, across various parameters. Findings suggest risks associated with smaller key sizes. Therefore, this study evaluates the performance of RSA, DSA, PGP, and ECC algorithms. Ten JPEG medical images (ranging from 321kb to 675kb) and two video files (2.23 mb in AVI and MP4 formats) were obtained from www.pexel.com for testing purposes. The images were converted to grayscale and the video frames were converted to ASCII values to accelerate encryption and decryption processes. Python 3.11 was used to implement encryption algorithms, and their performance was evaluated based on encryption time, decryption time, peak signal-to-noise ratio (PSNR), and key length. The RSA, DSA, PGP, and ECC gave 0.0065s, 0.0754s, inf (dB), and 2048 bits; 0.0486s, 0.0391s, inf (dB), and 1024 bits; 0.1986s, 0.2779s, inf (dB), and 2048 bits; 0.0125s, 0.0031s, inf (dB), and 256 bits for encryption time, decryption time, peak signal-to-noise ratio (PSNR) and key length respectively for medical images. The corresponding values for video avi data were 0.0129s, 0.0609s, inf (dB), and 2048 bits; 0.1269s, 0.1339s, inf (dB), and 1024 bits; 0.2954s, 0.2428s, inf (dB), and 2048 bits; 0.0169, 0.0109s, inf (dB), and 256 bits respectively. Also, the corresponding values for video mp4 data were 0.0159s, 0.0919s, inf (dB), and 2048 bits; 0.1539s, 0.1399s, inf (dB), and 1024 bits; 0.3738s, 0.2518s, inf (dB), and 2048 bits; 0.0721, 0.0090s, inf (dB), and 256 bits respectively. The study found that ECC outperforms RSA, DSA, and PGP in operational efficiency and security. ECC demonstrates shorter decryption times and stronger key lengths, with all algorithms achieving infinite peak signal-to-noise ratios, maintaining data quality throughout encryption and decryption. While RSA has the shortest encryption time, ECC excels across other metrics. The study recommends adopting ECC for shorter encryption times, faster decryption times, higher peak-to-noise ratios, and longer keys, highlighting its overall superiority.

Keywords: Cryptography, Algorithm, Asymmetric, Rivest Shamir Adleman (RSA), Digital Signature Algorithm (DSA), Pretty Good Privacy (PGP), Elliptic Curve Cryptography (ECC)

1. INTRODUCTION

A sensible option for safeguarding the information of a company is data encryption, which is a widely used and reliable security technique. There are, however, numerous encryption techniques that can be used. Symmetric and asymmetric encryption techniques are two broad categories. In symmetric approaches, the encryption and decryption keys are identical or may be readily determined from the encryption key. The challenge with the symmetric approach is that parties must safely share a secret key. Asymmetric key cryptography (AKC) or public-key cryptography (PKC) uses two keys, typically a private key and a public key; the secret key is used for decryption or signature generation while the public key is used for encryption or signature verification. The PKC derives its notoriety by creating two ground-breaking ideas: first, a solution to the symmetric key cryptography's key distribution problem, and second, a method for digital signatures.

A pair of keys are employed in asymmetric methods to deal with the issue of key distribution. Given only the knowledge of the cryptographic algorithm and the encryption key, it is computationally impossible to derive the decryption key. Any kind of communication via a network, including documents, files, communications, and photos, can be encrypted. The difficult part is choosing the strategies an internet security professional should use based on the unique circumstances of their firm. The algorithms RSA, DSA, PGP, and ECC, were tested to see which one performed better when applied to medical images and video data. Before medical images and video data were encrypted and decrypted, the interactive Graphical User Interface (GUI) in PYTHON was displayed. The system encrypted various file sizes of downloaded medical imaging data during the experiment. Below are a few examples:

487kb,422kb,638kb,334kb,576kb,675kb,514kb,332kb,321kb,407kb,678kb,433kb,562kb, and 1.65mb video data.

Using the medical images, video data of avi and mp4 format PGP has the longest encryption time, decryption time and peak-signal-to-noise ratio being infinite, DSA has a longer encryption time, decryption time and peak-signal-to-noise ratio being infinite with stronger key length, RSA has shortest encryption time, long decryption time and peak-signal-noise ratio being infinite while ECC has the long encryption time, shortest decryption time, and peak-signal-noise ratio being infinite with the strongest key length. This also showed that ECC algorithm is more efficient and secure than PGP, DSA and RSA algorithms. During the experimental process, in the course of encrypting and decrypting avi video, it was discovered that all the algorithms encrypted and decrypted the medical video but the video was not shown on the software. This was carried out using four different browsers for the experiment. However, the result of the encryption was generated for each of the algorithm. This shows that there were still some deficiencies in encrypting and decrypting avi video. There were ten (10) downloaded medical images, and all were encrypted and decrypted successfully.

2. RELATED WORKS

Ron Rivest, Adi Shamir, and Len Adleman, (1978) explained cryptographic algorithm by saying that because it is difficult to factor huge prime integers, RSA is now the most popular asymmetric algorithm. This algorithm, which is frequently used in Internet transactions, was standardized by NIST.

The difficulty of solving the discrete logarithm issue on elliptic curves led to the creation of Elliptic Curve Cryptography (ECC) according to N. Koblitz and V. Miller in the 1980s. The elliptic algorithm, despite its intricacy, has been the subject of much academic research. The security levels of the ECEG and MVEC algorithms are 160-bit, 224-bit, 256-bit, 384-bit, and 521-bit key sizes, while those of the RSA algorithm are 1024-bit, 2048-bit, 3072-bit, 7680-bit, and 15360-bit key sizes. The National Institute of Standards and Technology (NIST) specifications for public key sizes with comparable security levels were the source of these key sizes (Alese, 2000).

Narasimham Challa *et al,* (2007) examined the performance of the RSA and NTRU asymmetric algorithms on variable text file sizes using key sizes of 51 bits and 20 bits for the encryption and decryption processes, respectively. They came to the conclusion that NTRU outperformed RSA in terms of authentication, encryption, and decryption. Abdul *et al.,* (2009) tested, compared and assessed the performance of six popular encryption methods, including AES (Rijndael), DES, 3DES, RC2, BLOWFISH, and RC6. The following factors were used for comparisons: encryption speed, battery power consumption, key size differences, data kinds, and block sizes. When the findings were shown in Base 64 or Hexadecimal Base encoding, it was determined that there was no discernible difference. Nevertheless, it was shown that BLOWFISH outperforms other widely used encryption algorithms in terms of packet size variation, with RC6 coming in second. Once more, it was discovered that 3DES is not as efficient as the DES algorithm. Ultimately, it is evident that altering the size of the key causes a noticeable shift in both energy usage.

Elminaam *et al.,* (2010) carried out a number of experiments in order to assess the effectiveness of symmetric algorithms, such as AES (Rijndael), DES, 3DES, RC2, Blowfish, and RC6. The experiments measured energy usage, power consumption, key sizes, data kinds, and packet sizes. Their findings demonstrated that, when it came to changing packet size, Blowfish outperformed other encryption techniques. El-Hadedy *et al.* (2008) asymmetric algorithms used in this study have a distinct encryption mode of operation. The performance of cryptographic algorithms in WSNs has been assessed in numerous researches, although the performance analysis is not standardized. Studies on the performance evaluation of cryptographic algorithms for WSNs, according to Margi *et al*. (2009), are frequently highly dissimilar in terms of technique, platform, metrics, and analysis focus, making it challenging to directly compare the results achieved. As a result, this research presents an analysis of the theoretical aspects of cryptographic algorithms like RSA, ECC, and MQQ in addition to their performance evaluation on embedded systems utilized in wireless sensor networks.

3. METHODOLOGY

This research assessed the performance of four distinct algorithms: RSA, PGP, DSA, and ECC using some asymmetric data encryption techniques. The entire framework of the system for the four algorithm is displayed in figure 1. The research methods employed are as follows:

a) Data acquisition: The following were used as test beds: ten (10) Joint Photographic Expert Group (JPEG) medical images in the range of 79.9kb to 829kb were downloaded online via www.pexels.com, and two video data (2.23mb) in Audio Video Interleave (avi), MPEG-4 (MP4) (2.23mb) were also downloaded online via www.youtube.com.

b) **Data preprocessing:** The downloaded medical image data were transformed into text, processed in grayscale, and subjected to mathematical operations. The image was individually converted to grayscale, each pixel having a value between 0 and 255 characters.

   i. **Conversion to grayscale (image):** In order to convert medical picture data to text, the corresponding characters were first written as ASCII characters, which were then read back and mathematically transformed into their corresponding pixel values.

   ii. **Conversion of video:** The video data was compressed, extracted and converted into binary. To speed up the encryption and decryption process, the video was divided into frames and transformed to the corresponding American Standard Code for Information Interchange (ASCII) value. Video compression involves the following steps: The input video was converted into frames, and after each frame was processed, the individual frames of the video were computed using DWT which is defined as $X_k = \sum_{n=0}^{N-I} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$ where k=0,..., N-1. N is the number of items.

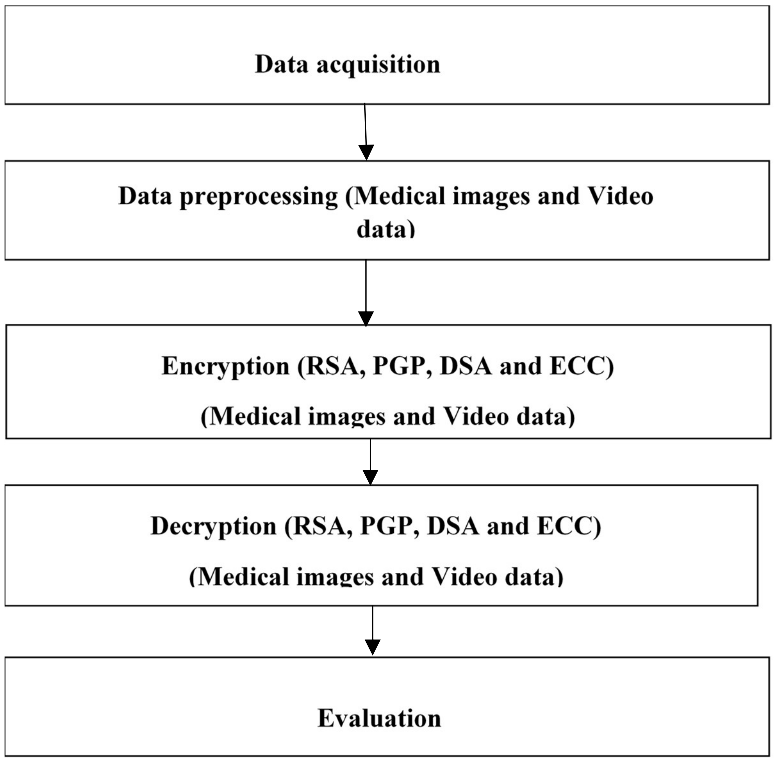
Data acquisition
Data preprocessing (Medical images and Video data)
Encryption (RSA, PGP, DSA and ECC) (Medical images and Video data)
Decryption (RSA, PGP, DSA and ECC) (Medical images and Video data)
Evaluation

Fig 1: Block Diagram of RSA, DSA, Pretty Good Privacy and ECC Algorithms.

Algorithm 1: Algorithm for the Generation of Public and Private Keys for RSA

Generation of Public and Private Keys for RSA

Step 1: Key Generation

i. Choose two very large prime numbers p and q.
ii. Compute $n = p*q$.
iii. Compute $\emptyset(n) = (p-1)(q-1)$. ($\emptyset$ is Euler's totient function).
iv. Choose an integer e such that, e is relatively prime to φ, i.e. $1 < e < \emptyset(pq)$, and $\gcd(e, \emptyset(n)) = 1$
v. Compute $d = e^{-1} \bmod [\emptyset(n)]$, where *d* is the private key and *e* is the public key
vi. Publish the public encryption key: (e, n), where *e* is the public key and *n* is the block size
vii. Keep secret private decryption key: (d, n), where *d* is the private key and *n* is the block size

Step 2: Encryption Process

i. Obtain the public key of recipient (e, n)
ii. Represent the information as an integer M in [0, n-1]
iii. Compute $c = M^e \bmod n$ where $M<n$, *c* is the Cipher text, *M* is the Plain text, *e* is the Encryption key and *n* is the block size.

Step 3: Decryption Process

i. Use private key (d, n)
ii. Compute $M = c^d \bmod n$ where *c* is the cipher text, *M* is the plain text, *d* is the decryption key, and *n* is the block size.
iii. Reconstruct the secret message

Algorithm 2: Algorithm for Generation of Public and Private Keys for DSA
Determination of Public and Private Key for DSA

Step 1: Choose parameters
Before the encryption and decryption process start, the following initialization is done:

i. Choose a prime number (p) of a given length (1024 or 2048 bits, for example).
ii. Choose a prime number q that is a divisor of $p-1$.
iii. Select an integer g such that $1<g<p-1$ and $g^{(p-1)q} \bmod p \neq 1$

Step 2: Generate private key

- Choose a private key x randomly where $0<x<q$

Step 3: Compute public key

- Compute the public key y using the formula $y = g^x \bmod p$

Step 4: Public key

- The public key is the pair (p,q,g,y).

Step 5: Private key

- The private key is the integer x

Algorithm 3: Algorithm for the Generation of Public and Private Keys for ECC.

```
ECC Algorithm Exhibits Key Generation, Encryption, and Decryption.

Global Public Elements
1: Choose an elliptic curve $E_q(a, b)$ with parameters a, b, and q, where q is a prime and > 3, or an integer of the form 2m.
2: Selects G(x, y) - a global point on an elliptic curve whose order is large value n.
Alice key generation
1: Selects a private key, $V_A$; where, $V_A < n$.
2: Calculates the public key, PA(x, y); $P_A(x, y) = VA \times G(x, y)$.
Bob key generation
1: Selects a private key, $V_B$; where $V_B < n$.
2: Calculates the public key, $P_B(x, y)$; $P_B(x, y) = V_B \times G(x, y)$.
Secret key calculation by Alice
1: $S_K(x, y) = V_A \times P_B(x, y)$.
Secret key calculation by Bob
1: $S_K(x, y) = V_B \times P_A(x, y)$.
Encryption by Alice using the public key of Bob
1: Alice chooses message Pm(x, y) and a random positive integer 'k' and $1 < k < q$.
2: Ciphertext, Cm((x, y),(x, y)); = ((k × G(x, y)), (Pm(x, y) + k × PB(x, y))).
Decryption by Bob using his own private key
1: Ciphertext, Cm((x, y),(x, y)).
2: Plaintext, Pm(x, y); = (Pm(x, y) + k × PB(x, y)) - (k × VB × G(x, y)) = Pm(x, y).
```

Here, the first coordinate of C_m gets multiplied with the private key of Bob i.e., V_B, which in turn becomes similar to Bob's public key. Finally, due to the subtraction of the resultant coordinate with the second coordinate of the cipher text C_m, all get canceled and only (x, y) Pm(x, y) gets left.

Algorithm 4: Algorithm for the Generation of Public and Private Key for PGP
Generation of Public and Private Key for PGP

Step 1: Generate a Random Key Pair
PGP typically uses the RSA algorithm for key pair generation.

- Choose two large prime numbers, p and q.
- Calculate n=p*q.
- Compute Euler's totient function $\varphi(n) = (p-1) \times (q-1)$.

Select a public exponent e that is relatively prime to $\varphi(n)$ (i.e., $1 < e < \varphi(n)$ and e and $\varphi(n)$ are coprime).
Compute the private exponent d such that $d \equiv e^{-1} \bmod \varphi(n)$.
Step 2: Public Key
The public key consists of the modulus n and the public exponent e. It is typically represented as (n, e).
Step 3: Private Key
The private key consists of the modulus n and the private exponent d. It is typically represented as (n,d).
Step 4: Key Generation Example

- Select two prime numbers: p=61 and q=53.
- Calculate $n = p \times q = 61 \times 53 = 3233$.
- Compute $\varphi(n) = (p-1) \times (q-1) = 60 \times 52 = 3120$.
- Choose e = 17 (commonly used) as the public exponent.
- Calculate d such that $d \equiv e^{-1} \bmod \varphi(n)$. In this case, d=2753.
- Public key: (n,e) = (3233,17).
- Private key: (n,d) = (3233,2753).

4. IMPLEMENTATION OF ASYMMETRIC ALGORITHMS IN PYTHON

A specific pattern was followed in the implementation of the algorithms employed in this research. This was implemented using a Windows 10 Professional 64-bit operating system, x64-based processor, Intel (R) core i5-3230M CPU running at 2.60GHz clock speed, and 4GB RAM using Python programming language (PYTHON 3.11). When the file was executed, the Graphical User Interface (GUI) window showed up as the file name. The quantity of pictures related to medicine that needed to be decrypted and encrypted were added. The interface was made explicit with the help of some created buttons such as “Encrypt and Decrypt”, “Select Encryption Type”, ”Key Length/Size”, “Encrypted Data”, “Original Data”, ”Download keys”, “Result”, “Run Algorithm” and Select Dataset buttons to select the video and image format, the algorithms to be used and tables to display results and summary result.

4.1 Performance Evaluation Metrics

The Performances of RSA, DSA, PGP and ECC using medical images and video data were evaluated from the parameters below:

a) **Encryption Time (Computation time/ Response time):** An encryption algorithm's encryption time is defined as the amount of time it takes to convert a plain text file into a cipher text. The entire plaintext in encrypted bytes divided by the encryption time is the encryption time, which is used to determine the throughput of an encryption system.
b) **Decryption time (Computation Time/ Response Time):** Decryption time is the time required by an encryption technique to convert a cipher text into a plain text.
c) **MSE/Peak signal-to-noise (PSNR):** The Peak Signal-to-Noise Ratio (PSNR) is a typical metric for assessing the quality of a reconstructed or compressed image or signal by contrasting it with the original, uncompressed version. It offers a numerical representation of the extent to which compression or other operations have lowered the signal's quality.

To compute the MSE for each of the cryptography algorithm implemented, the following equation was used: MSE = n1∑i = 1n(mi−ci)2
Where:
MSE represents the Mean Squared Error; n is the total number of elements in the message.
mi represents the ith element of the original message.
ci represents the ith element of the encrypted message.

The difference between the encrypted message and the original message is measured using this equation. Better encryption is shown by a smaller MSE, which shows how near the encrypted message is to the original message. The PSNR was derived using:
PSNR = $10 \cdot \log_{10} (255^2 / \text{MSE})$dB. Where MSE is the Mean Squared Error derived.

d) Key length / Key size: Key length is the number of bits in the key of an encryption algorithm. Insufficient security is indicated by a short key length. Longer keys do not, however, always equate to better security. The maximum number of combinations needed to crack an encryption technique depends on the key length. There are two to the nth power (2^n) potential keys for any key length of n bits. For instance, there are only two possible keys: zero and one, if the key is only one bit long and that one bit can be either a zero or a one. Nonetheless, there are 240 potential keys if the key length is 40 bits long.

5. RESULTS

Different asymmetric data encryption techniques, including RSA, DSA, PGP, and ECC, were tested to see which one performed better when applied to medical images and video data.

Table 1. The summary result of the encryption time, decryption time, Peak-Signal-To-Noise Ratio, and key length of 10 medical images in jpg format.

Algorithms	Encryption time (sec)	Decryption time (sec)	PSNR (dB)	Key length (bits)
RSA	0.0065	0.0754	inf	2048
DSA	0.0486	0.0391	inf	1024
PGP	0.1986	0.2779	inf	2048
ECC	0.0125	0.0031	inf	256

Before medical images and video data were encrypted and decrypted, the interactive Graphical User Interface (GUI) in PYTHON was displayed.

In Table1, it was shown that PGP has the longest encryption time, decryption time and peak-signal-noise ratio being infinite to show that the quality of the data was maintained during encryption and decryption process, DSA has a longer encryption time, decryption time and peak-signal-noise ratio being infinite with stronger key length, RSA has the shortest encryption time, long decryption time and peak-signal-noise ratio being infinite while Elliptic Curve Cryptography (ECC) has a long encryption time, shortest decryption time and peak-signal-noise ratio being infinite with the strongest key length. These suggest that a system will need a significant amount of time and space to run PGP, DSA, and RSA. Additionally, even though the algorithms are better because due to the longer key lengths, there would be a high-power consumption. However, because ECC uses less power and has shorter encryption and decryption times with the shortest key length, it will be more effective.

Table 2. The summary result of the encryption time, decryption time, Peak-Signal-To-Noise Ratio, and key length for one video in MP4 format.

Algorithms	Encryption time (sec)	Decryption time (sec)	PSNR (dB)	Key length (bits)
RSA	0.0159	0.0919	inf	2048
DSA	0.1539	0.1399	inf	1024
PGP	0.3738	0.2518	inf	2048
ECC	0.0721	0.0090	inf	256

In Table 2, PGP has the longest encryption time, decryption time and peak-signal-noise ratio being infinite, DSA has a longer encryption time, decryption time and peak-signal-noise ratio being infinite with stronger key length, RSA has the shortest encryption time, a long decryption time and peak-signal-noise ratio being infinite with strong key length while ECC has a long encryption time, shortest decryption time and peak-signal-noise ratio being infinite with the strongest key length. It was observed that ECC is more efficient and secure than PGP, DSA and RSA.

Table 3. The summary result of the encryption time, decryption time, Peak-Signal-To-Noise Ratio, and key length for one video in Avi format.

Algorithms	Encryption time (sec)	Decryption time (sec)	PSNR (dB)	Key length (bits)
RSA	0.0129	0.0609	inf	2048
DSA	0.1269	0.1339	inf	1024
PGP	0.2954	0.2428	inf	2048
ECC	0.0169	0.0109	inf	256

Also, in table 3, PGP has the longest encryption time, decryption time and peak-signal-noise ratio being infinite, DSA has a longer encryption time, decryption time and peak-signal-noise ratio being infinite with stronger key length, RSA has shortest encryption time, long decryption time and peak-signal-noise ratio being infinite while ECC has the long encryption time, shortest decryption time, and peak-signal-noise ratio being infinite with the strongest key length. This also showed that ECC algorithm is more efficient and secure than PGP, DSA and RSA algorithms.

6. CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Ensuring the security of a message is crucial while it is being transmitted between users. A comparison of RSA, PGP, DSA, and ECC was offered in this study. The Python (Python 3.11) programming language was utilized to implement the algorithms, and its performance was compared to the techniques employed based on a few parameters. The video data and medical pictures that served as the test bed were fully restored without alteration.
Experimental results indicate that ECC is more secure and operationally efficient than RSA, PGP, and DSA. Furthermore, this study has shed light on more effective methods and substitutes for the RSA, PGP, DSA, and ECC encryption algorithms. In particular, devices with limited resources are best suited for an ECC.

6.2 Recommendation

In order to increase the ECC algorithm's efficiency, it is advised that more methods be compared to it in future research. Other encryption measures that can be used as parameters include CPU process time, execution time, overall throughput, and battery power usage. Additionally, the findings can be used to other diverse data inputs, like voice input and mp4 image data files. Finding should also be carried out on why video data in avi format does not show the data but gives the result of the encryption and decryption.

REFERENCES

[1] Seth, S. M. and Mishra, R. (2011).Comparative Analysis of Encryption Algorithm for data Communication, *International Journal of Computer Science and Technology*, 2 (2): 292-294.

[2] Dindayal Mahto and Dilip Kumar Yadav (2018). *International Journal of Network Security*, Vol.20, No.4, PP.625-635, (DOI: 10.6633/IJNS.201807 20(4).04)

[3] Koblitz N.,and Miller V., (1987). "Elliptic curve cryptosystems, Mathematics of computation, vol. 48, no. 177, pp. 203–209,

[4] Alese B.K, Philemon E. D and Falaki, S. O. (2012). Comparative analysis of public-key encryption schemes, *International Journal of Engineering and Technology*, vol. 2, no. 9, pp. 1552–1568.

[5] Narasimham Challa and Jayaram Pradhan, (2007). Performance Analysis of Public key Cryptographic Systems RSA and NTRU, IJCSNS *International Journal of Computer Science and Network Security, VOL.7 No.8.*

[6] Abdul, D. S.; Kader A. H. M. and Hadhoud M. M. (2009). Performance Evaluation of Symmetric Encryption Algorithms, Communications of the IBIMA, 8 (1): 58-64.

[7] Elminaam, A.; Salama, D.; Mohamed, H.; Kader,A.; and Hadhoud M.M. (2010) "Evaluating The Performance of Symmetric Encryption Algorithms," vol. 10, no. 3, pp. 213–219.

[8] El-Hadedy M, Gligoroski D, and Knapskog S, (2008). High performance implementation of a public key block cipher-mqq, for fpga platforms, in Reconfigurable Computing and FPGAs, ReConFig'08. *International Conference on*. IEEE, 2008, pp. 427–432.