Journal of Basic and Applied Engineering Research
p-ISSN: 2350-0077; e-ISSN: 2350-0255; Volume 4, Issue 1; January-March, 2017, pp. 58-63

http://www.krishisanskriti.org/Publication.html

A Security Study to Compare Cryptographic Algorithms Based on Performance in Mobile Cloud Computing

Solomon Babatunde Olaleye

Department of Computer Science & Engineering,
Sharda University, Greater Noida, INDIA
E-mail: olaleye3@yahoo.com

Abstract—*Cryptography is an area that is consistently being studied because of its implementation in data security. The safety of data against attacks in mobile cloud computing is indispensable to ensure the confidentiality, integrity and availability of data. The internet through which data is transferred in MCC is unsecured. Hence, the need to study and being able to use an effective cryptographic algorithm based on its security, efficiency, speed of encryption and decryption among others to secure data. This paper therefore studied AES, DES, RSA and ECC cryptographic algorithms. AES and DES are symmetric key cryptographic algorithms while RSA and ECC are asymmetric key cryptographic algorithms. The algorithms were evaluated using encryption time, decryption time and throughput. The experimental results on different file sizes are presented and discussed. AES has better performance and more secured than DES. ECC with smaller key size is faster in encryption, decryption and consume less power than RSA.*

Keywords*: Cryptography, AES, DES, RSA, ECC, Encryption, Decryption, Throughput.*

1. INTRODUCTION

Cryptographic algorithms are security algorithms that ensure security of data against unauthorised access. There are many of such algorithms today among which are Data Encryption Standard (DES), 3DES, Advanced Encryption Standard (AES), Blowfish, Elliptic Curve Cryptography (ECC) and Rivest Shamir Adleman (RSA). Cryptography provides security for data by means of encryption and decryption [1-2]. It is used to provide security for data at rest as well as in transit in mobile cloud computing (MCC). The choice of a cryptographic algorithm depends on the environment, memory usage, speed, performance and strength against attacks. The choice of any cryptographic algorithm must fulfil data protection security goals such as confidentiality, data integrity, data availability, authentication and non-repudiation.

Mobile cloud computing is a combination of mobile computing and cloud computing. Mobile computing allows computing performance for a user on the move. This enables a user with a mobile device to carry out task anywhere, whether in his/her usual environment or elsewhere as long there is network availability [3]. Cloud computing has many definitions. The definition is based on how an individual or people see it. It can be likened to an elephant that can be described the way an individual or people see it. However, the most common definition is the one given by the National Institute of Standards and Technology (NIST) based in USA. The NIST definition of Cloud Computing as stated in [4] is

"a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (i.e. networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interactions."

Based on NIST definition, there are five (5) important characteristics of cloud computing. They are;

i. There is on-demand user's service such as server time and storage.
ii. There is network access which can be accessed by mobile phones, tablets, laptops etc.
iii. Availability of service providers' computing resources such as memory, processing, bandwidth etc.
iv. Resources can be rapidly provisioned to meet users demand.
v. Computing resources usage can be measured and billing is done based on resources usage by the customers (users).

2. OBJECTIVES

The objectives of this research paper are;

1. To compare cryptographic algorithms of DES, AES, RSA and ECC.
2. To evaluate their performances in terms of encryption, decryption and throughput.

3. SYMMETRIC KEY CRYPTOGRAPHY

Cryptographic algorithms are commonly grouped into two (2). They are symmetric key cryptography and asymmetric key cryptography [5].

Symmetric key cryptography is an encryption technique in which the sender of message and receiver of message use the same key. The sender encrypts message by using a key and the receiver as well uses the same key to decrypt the received message [6]. The symmetric key cryptography is also known as secret key cryptography. There are many symmetric key algorithms among are DES, AES, Blowfish, 3DES, Twofish, RC6 and IDEA. Symmetric key method is as depicted in fig. 1.

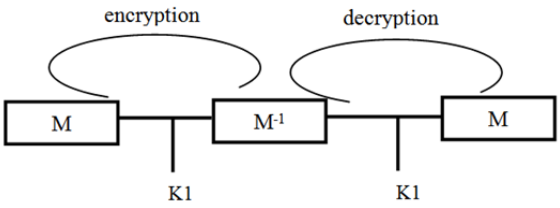

Fig. 1: Symmetric key method

Note that k1 = k1 = encryption/ decryption key, M = Message in plaintext and M^{-1} is the ciphertext (encrypted message).

3.1 DES – Data Encryption Standard

DES also called Data Encryption Algorithm (DEA) is a cryptographic algorithm developed by IBM in the early 1975. Recently DES was found vulnerable to powerful attacks. Therefore, its popularity and usage is decreasing [7]. The DES was well known as a symmetric-key algorithm for encrypting sensitive data. It uses 56 bits key to encrypt and decrypt. Hence, it will take at most 2^{56} attempts to obtain the correct key. On any 64 bits block of data, it completes 16 rounds of encryption [8]. The basic working of DES is pictorially represented in fig. 2.

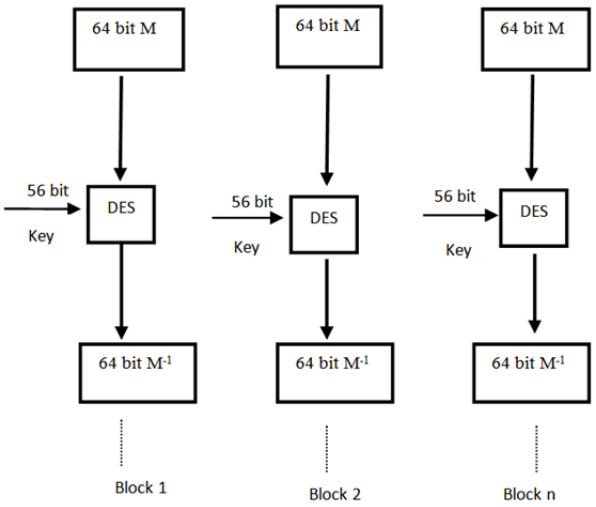

Fig. 2: DES working [7]

3.2 AES – Advanced Encryption Standard

Daemen Joan and Rijmen Vincent developed Advanced Encryption Standard (AES) in year 2000. The two (2) scientists were from Belgium. Based on competition, their work emanated as the best. AES allows block sizes and key sizes. AES is a block cipher [9-10] with a block of 128 bits. There are three (3) key lengths in AES which are 128, 192 and 256 bits. They are referred to as AES-128, AES-192 and AES-256. The different keys are generated using 10, 12 and 14 different numbers of rounds [11].Table 1 shows the summary of the key size and numbers of rounds.

Table 1: AES key size and number of rounds

Algorithm	Key size (words/bytes/ bits)	No of rounds
AES-128	4/16/128	10
AES-192	6/24/192	12
AES-256	8/32/256	14

For example in an encryption process if a plaintext has 128 bits and key of 256 bits size, the number of rounds in AES 256 is 14 [12].

AES is a symmetric key [13] algorithm that operates on 2-dimensional arrays called state. The state contains 4 rows for each byte. Based on the key size as stated in table 1, it protects against known attacks. Such as brute force and future attacks. AES is an encryption standard adopted by United State government. It had been analyzed extensively and used worldwide [14]. Fig. 3 shows the AES design for encryption while the inversed is for decryption.

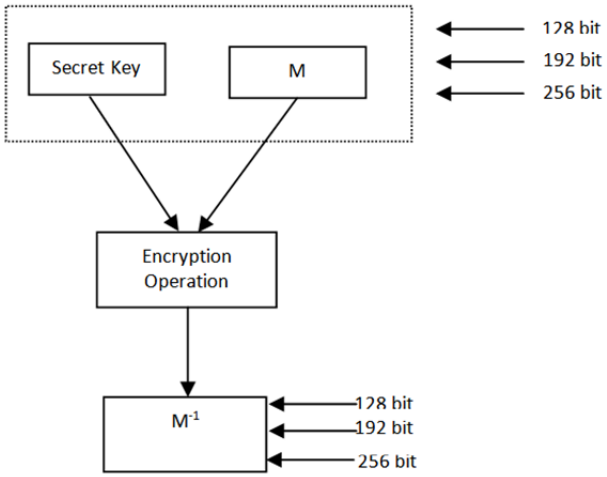

Fig. 3: AES design for encryption

The number of rounds in AES depends on the key size. The 128 bits key size uses 10 rounds of substitutions and permutations, 192 bits key size uses 12 rounds of substitutions and permutations, and 256 bits key size uses 14 rounds of substitutions and permutations. There are four (4) operations carried out on each of the rounds except the last round. The four (4) operations for encryption are;

1. Sub Bytes
2. Shift Rows
3. Mix Columns
4. Add Round Key

Note that for the last round of AES encryption process, Mix columns are excluded. AES was designed that way. Likewise for decryption process the four (4) operations are;

1. Inverse Shift Rows
2. Inverse Sub Bytes
3. Add Round Key
4. Inverse Mix Columns

For the last round in the decryption process the inverse mix column is excluded.

4. ASYMMETRIC KEY CRYPTOGRAPHY

Asymmetric key cryptography is also known as public key cryptography. This technique uses one key to encrypt message and uses another key to decrypt the same message. It makes use of two (2) keys which are public key and private (secret) key. The public key is used to encrypt a message by the sender and the receiver uses his/her private key to decrypt [15]. Asymmetric key method is illustrated in fig. 4. RSA, digital signature, ECC, and Diffie Hellman are the commonly used asymmetric key cryptography today.

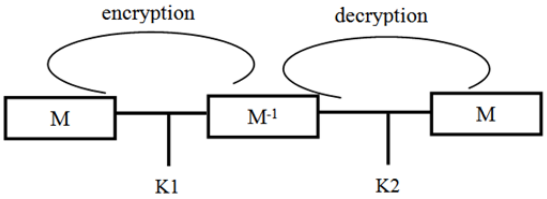

Fig. 4: Asymmetric key method

Here, K1 ≠ K2, K1 is used to encrypt message while K2 is used to decrypt the same message.

4.1 RSA- Rivest Shamir Adleman Algorithm

RSA algorithm was developed in 1977 by Rivest R., Sharmir A. and Adlemen L. It is a well-used asymmetric key cryptography. It is based on Mathematical computation of two large prime numbers. RSA algorithm is simple than a symmetric key cryptographic algorithm. But the real challenge is in the generation and selection of the public and private keys [16-17]. Today, it is used in many software products for digital signatures, encryption of data in small blocks and key exchange. It is used for secure communication over insecure channels. RSA encryption algorithm consumes substantial computing resources like mobile device battery, CPU time and memory space. It uses variable size encryption block and a variable size key [18]. The key sizes of RSA in bits are 512, 1024, 1536, 2048, 3072 and 7680.

Algorithm RSAencryption (M, A, B, N):

Input: select 2 large prime numbers say A and B

Output: M^{-1} which is the ciphertext derived from M (plaintext)

Let A and B be 2 large prime numbers

N ← A*B

Let K1 be encryption key but not a factor of (A-1) and (B-1).

Let K2 be decryption key such that K1*K2 mod (A-1) * (B-1) ← 1.

M^{-1} ← M*K1 mod N

return M^{-1}

A major demerit of RSA as an asymmetric key cryptography is that it is slower for encryption and decryption when compared with other symmetric key cryptographic algorithms.

4.2 ECC – Elliptic Curve Cryptography

Elliptic curve cryptography is an asymmetric cryptosystem that is based on difficulty of discrete logarithm of elliptic curve. The discrete points on its elliptic curve over a finite field are used as a cyclic group. It was first proposed by Neal Koblitz and Victor Miller in 1985. ECC gives the same level of security as other cryptographic algorithms. However, ECC is not popular like others. But it gives equal security level to RSA with smaller key size (as can be seen in table 2), hence, minimizing processing overhead.

Table 2: NIST key size recommendation

SN	ECC key size	RSA key size	Ratio
1	112	512	1:5
2	163	1024	1:6
3	192	1536	1:8
4	224	2048	1:9
5	256	3072	1:12
6	384	7680	1:20

The primary benefits of using ECC are its smaller key size, memory usage, low consumption of bandwidth and faster processing time among others [19-21]. In essence, ECC usage is advantageous in wireless devices such as in smartphones where memory, CPU and battery are limited. However, it is more complex and difficult in implementation when compared to RSA [22]. Fig. 5 and fig. 6 show typical elliptic curve graphs.

Generally, the equation of an elliptic curve is given as

$y^2 = x^3 + ax + b$

where the values of a, b are constant [23].

The elliptic curves are used to build the asymmetric key cryptography system. Such that the secret (private) key say d

is randomly selected from (1, n-1), where n is a prime number. The public key B is calculated by d*A, where A, B are points on the elliptic curve as depicted in fig. 5.

Further, based on the elliptic curve graph, we can carry out addition and point doubling operations.

4.2.1 Addition Operation

Assuming A and B are points on the elliptic curve, then we can define C = A + B

4.2.2 Point Doubling Operation

Doubling of a point is achieved on the elliptic curve when A = B, then C = A + A = 2A, see fig. 6.

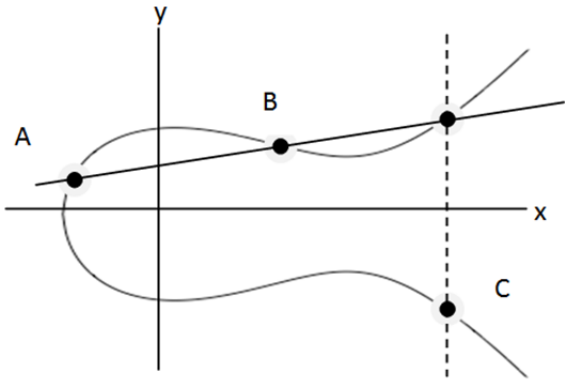

Fig. 5. Elliptic curve graph (2 points addition)

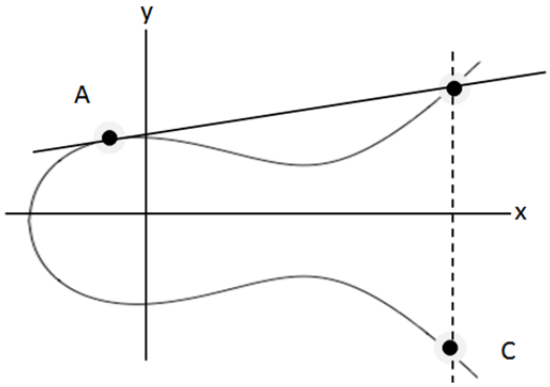

Fig. 6. Elliptic curve graph (point doubling)

The strength of ECC is based on elliptic curve discrete logarithm problem. How?

given d and A → it is easy to compute B, but

given B and A →it is very hard to find d, however, d has to be large enough.

Algorithm ECC key pair Generation

Input: Choose a secret key say d

Output: Public key B

Let d be the secret key between (1, n-1)

Let B ← d*A

return B

Elliptic Curve Cryptography Encryption

The ECC encryption using ElGamal encryption scheme [24] is illustrated thus; a plaintext is represented as a point M on the curve, it can be encrypted by the addition of M and rB where r is a random number and B is the receiver's public key. The sender then sends two points $P_1 = rA$ and $P_2 = M + rB$ to the receiver who uses his/her secret key d to derive the plaintext as follows;

$dP_1 = d(rA) = r(dA) = rB$

(substitute dA with B)

Hence, $M = P_2 – rB$, an attacker who tries to recover M must be able to calculate rB which is very hard.

Algorithm ECC ElGamalEncryption

Input: Public key B and plaintext M

Output: Ciphertext (P_1, P_2)

Let plaintext be denoted as a point M

Let r be a random number between (1, n-1)

$P_1 \leftarrow r*A$

$P_2 \leftarrow M + r*B$

return (P_1, P_2)

5. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The security study for the comparison was implemented using java. security and java. crypto packages available in Java Cryptography Extension and Java Cryptography Architecture which offer cryptographic cipher for encryption and decryption. The experiments were carried out on Intel® Core™ i7-4510U CPU @ 2.00 GHz 2.60 GHz processor with 8.00 GB of RAM. The experiments were repeated ten times to ensure that the results are dependable and acceptable for use in comparing the cryptographic algorithms on different file sizes. AES of 256 bit key size, DES of 56 bit key size, RSA of 1024 bit key size and ECC of 163 bit key size were used for the study.

The performance metrics for evaluation are the speed of encryption and decryption for each of the cryptographic algorithms on different sizes of data. Further, the throughput of the encryption was also computed. The throughput of encryption is computed by dividing the plaintext by total time for encryption. It is measured in KB/ms. High throughput

value means low power consumption for a cryptographic algorithm. The experimental results are presented below.

Table 3. Encryption time

File Size (KB)	AES (ms)	DES (ms)	RSA (ms)	ECC (ms)
200	104.2	71.6	126.3	92.4
500	147.1	133.1	291.8	120.1
750	169.7	160	386.2	178.7
1000	190.3	208.3	517.7	190
2000	231.5	267.4	733	221.6
5000	417.2	610.9	1388.2	403.5

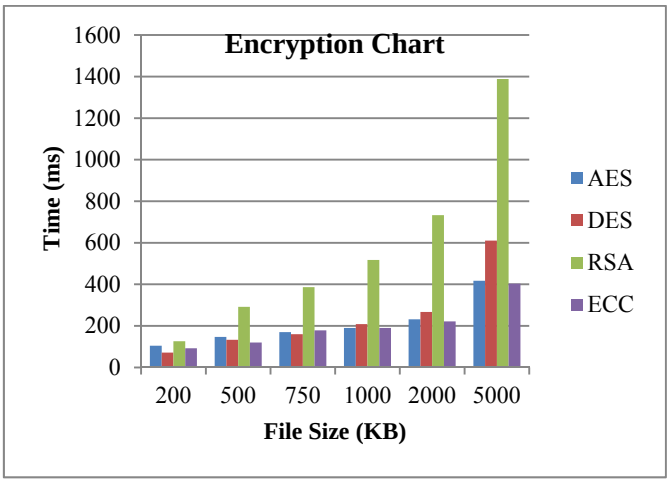

Fig. 7. Encryption chart

Table 4. Decryption time

File Size (KB)	AES (ms)	DES (ms)	RSA (ms)	ECC (ms)
200	97.6	101.3	104.3	90.1
500	121.3	156.1	288.4	109.7
750	160	189.7	341	131.2
1000	173.7	218.9	499.2	160.3
2000	218.3	270.1	754.2	210
5000	324.6	615.3	1011.8	297.8

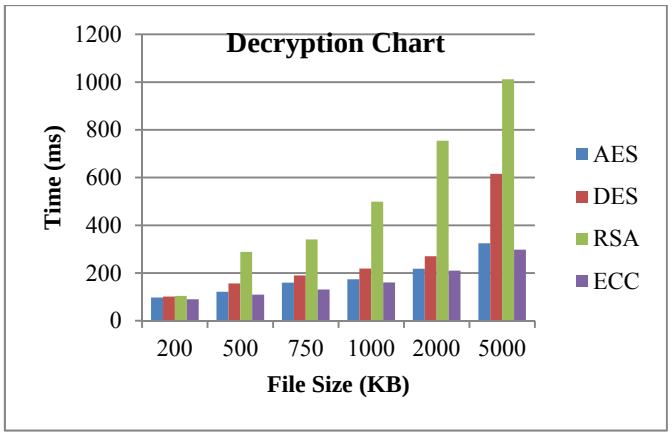

Fig. 8: Decryption chart

Table 5: Throughput of encryption

File Size (KB)	AES (ms)	DES (ms)	RSA (ms)	ECC (ms)
200	104.2	71.6	126.3	92.4
500	147.1	133.1	291.8	120.1
750	169.7	160	386.2	178.7
1000	190.3	208.3	517.7	190
2000	231.5	267.4	733	221.6
5000	417.2	610.9	1388.2	403.5
Average Time	1260	1451.3	3443.2	1206.3
Throughput (KB/ms)	7.5	6.5	2.7	7.8

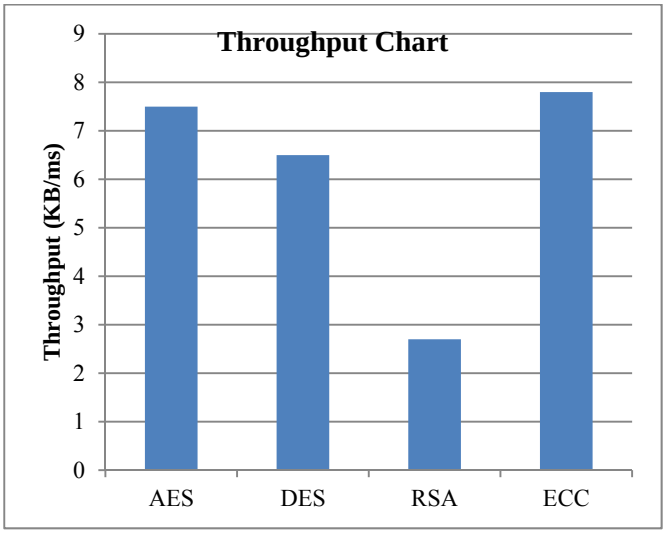

Fig. 9: Throughput chart

The experimental results, as showed in table 3, AES and DES encryption time increase as file size increases. Based on the decryption time showed in table 4, AES has less encryption time compared to DES. AES is more secure than DES in terms of security against brute force attack because of its large key size. Further, when comparing RSA and ECC. The encryption time and decryption time of ECC are far lesser than that of RSA even with ECC smaller key. This shows that ECC can be better used especially in devices where memory, storage and other computing resources are limited to provide better security. The high throughput values of AES and ECC showed that they consume less computing power comparing to DES and RSA.

6. CONCLUSION

This paper studied four different cryptographic algorithms. Encryption time, decryption time and throughput were used as performance metrics for evaluating each of the algorithms. It was observed that DES encryption time was faster than AES encryption time with smaller file sizes. But as the file sizes

increase the encryption time of AES were faster than DES encryption time. Considering the encryption time and decryption time of RSA and ECC, it showed that ECC with smaller key size of 163 bit was faster than RSA of 1024 bit. From the throughput results, it showed that AES, DES and ECC consumed less power compared to RSA. AES and ECC are found to be more secured for use in mobile devices in terms of security and resource constraints. In the future, since each of the algorithms has its merits and demerits, there is the need to merge the use of two or more cryptographic algorithms for better security without overlooking mobile devices resources limitations.

REFERENCES

[1] Thakur, J. and Kumar, N., "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis", *International Journal of Emerging Technology and Advanced Engineering*, 1, 2, December 2011, pp. 6 – 12.

[2] Kumar, A., Sinha, S. and Chaudhary, R., "A comparative analysis of encryption algorithms for better utilization", *International Journal of Computer Applications,* 71, 14, May 2013, pp. 19 – 23.

[3] Patil, B. D., Ramteke, P. L. and Chaudhari, D. N., "Development of android mobile application for cloud server", *International Journal on Recent and Innovation Trends in Computing and Communication,* 2, 12, December 2014, pp. 3906 – 3910.

[4] Jayaswal K. Kallakurchi J. Houde D. J. and Shah D. "Cloud Computing Black Book". India: Dreamtech Press, 2014.

[5] Ebrahim, M., Khan, S. and Khalid, U. B., "Symmetric algorithm survey: A comparative analysis", *International Journal of Computer Applications*, 61, 20, January 2013, pp. 12 – 19.

[6] Alshahrani, A. M. and Walker, S., "Different data block size using to evaluate the performance between different symmetric key algorithms", *International Journal of Computer Networks & Communications* (IJCNC), 6, 2, March 2014, pp. 89 – 97.

[7] Kahate A. "Cryptography and Network Security". India: McGraw Hill Education Private Limited, 2013.

[8] Bhanot R. and Hans R., "A review and comparative analysis of various encryption algorithms". *International Journal of Security and Its Applications,* 9, 4, 2015, pp. 289-306.

[9] Rihan, S. D., Khalid, A. and Osman S. E. F., "A performance comparison of encryption algorithms AES and DES", *International Journal of Engineering Research and Technology* (IJERT), 4, 12, 2015, pp. 151-154.

[10] Saini V. and Bangar P., "Design and implementation of advanced encryption standard algorithm -128 using verilog", *International Journal of Engineering and Advantage Technology* (IJEAT), 3, 5, 2014, pp. 265-268.

[11] Alsharani, A. M. and Walker S., "New approach in symmetric block cipher security using a new cubical technique", *International Journal of Computer Science and Information Technology* (IJCSIT), 7, 1, 2015, pp. 69-75.

[12] Dasari, S. and Swapanakumari, B., "Implementation of AES-256 encryption algorithm on FPGA", *International Journal of Emerging Engineering Research and Technology*, 3, 4, 2015, pp. 104-108.

[13] Bisht N. and Singh S., "A comparative study of some symmetric and asymmetric key cryptography algorithms", *International Journal of Innovative Research in Science, Engineering and Technology*, 4, 3, 2015, pp. 1028-1031.

[14] Westlund, H. B. "NIST reports measureable success of advanced encryption standard", *Journal of Research of the National Institute of Standards and Technology*, 2002.

[15] Iyer, K. B. P., Anusha, R. and Priya R. S., "Comparative study on various cryptographic techniques", *International Journal of Computer Applications*, ISSN: 0975-8887, 2014, pp. 37 – 42.

[16] Pavithra, S. and Ramadevi, E., "Study and performance analysis of cryptography algorithms", *International Journal of Advanced Research in Computer Engineering & Technology*, 1, 5, July 2012, pp.82 – 83.

[17] Singh, S., Maakar, S. K. and Kumar, S., "A performance analysis of DES and RSA cryptography", *International Journal of Emerging Trends &Technology in Computer Science* (IJETTCS), 2, 3, May – June 2013, pp. 418 – 423.

[18] Karule, K. P. and Nagrale, N. V. "Comparative analysis of encryption algorithms for various types data files for data security", *International Journal of Scientific Engineering and Applied Science* (IJSEAS), 2, 2, February 2016, pp. 495 – 498.

[19] Tripathi, A. and Yadav, P., "Enhancing security of cloud computing using elliptic curve cryptography", *International Journal of Computer Applications*, 57, 1, November 2012, pp. 26 – 30.

[20] Shin, S. and Eun, H., "Public key generation and encryption mechanism using the elliptic curve in smartphones", *International Journal of Security and Its Applications*, 9, 12, 2015, pp. 419 – 428.

[21] Patel, P, Patel, R. and Patel, N., "Integrated ECC and Blowfish for smartphone security", *Elsevier, Procedia Computer Science 78*, 2016, pp. 210 – 216.

[22] Rani, R. and Bagoria, R., "Optimum security technique for smartphones", *International Journal of Innovative Research in Science, Engineering and Technology,* 4, 5, May 2015, pp. 3484 – 3488.

[23] Brindha, K., Ramya, G. and Jayantila R. A., "Secured data transfer in wireless networks using hybrid cryptography", *International Journal of Advanced Research in Computer Science and Software Engineering,* 3, 10, October 2013, pp. 379 – 381.

[24] Hankerson, D., Menezes, A. and Vanstone, S., "Guide to elliptic curve cryptography", Springer-Verlag, New York Inc., 2004.