International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Comparative Analysis of RSA and ECC Algorithm

Priyanka Durge[1], Prof. Hirendra R. Hajare[2]

[1]M. Tech Student, Dept. of Computer Science and Engineering, BIT College, Ballarpur, Maharashtra, India
[2]Associate Professor, Dept. of Computer Science and Engineering, BIT College, Ballarpur, Maharashtra, India

---

Abstract – *Cryptography is the art and science of making a cryptosystem that is capable of providing information security. In this paper, we compare the two Asymmetric Algorithm i.e. RSA (Rivest Shamir Adelman) and ECC (Elliptic Curve Cryptography), these are known as the most efficient Public Key Cryptography among all asymmetric Encryption algorithm which are most commonly use now days for security purpose. We are trying to find out better strategy to increase security, by which we can protect our data more effectively.*

Key Words: *Cryptography, Cryptosystem, RSA, ECC, Asymmetric Algorithm*

1. INTRODUCTION

The art and science of concealing the messages to introduce secrecy in information security is recognized as cryptography. The word 'cryptography' was coined by combining two Greek words, 'Krypto' meaning hidden and 'graphene' meaning writing.

1.1 What is Cryptography?

Cryptography deals with the actual securing of digital data. It refers to the design of mechanisms based on mathematical algorithms that provide fundamental information security services. You can think of cryptography as the establishment of a large toolkit containing different techniques in security applications.

1.1.1 Security Services of Cryptography

The primary objective of using cryptography is to provide the following four fundamental information security services. Let us now see the possible goals intended to be fulfilled by cryptography.

Confidentiality

Confidentiality is the fundamental security service provided by cryptography. It is a security service that keeps the information from an unauthorized person. It is sometimes referred to as privacy or secrecy.

Confidentiality can be achieved through numerous means starting from physical securing to the use of mathematical algorithms for data encryption.

Data Integrity

It is security service that deals with identifying any alteration to the data. The data may get modified by an unauthorized entity intentionally or accidently. Integrity service confirms that whether data is intact or not since it was last created, transmitted, or stored by an authorized user.

Data integrity cannot prevent the alteration of data, but provides a means for detecting whether data has been manipulated in an unauthorized manner.

Authentication

Authentication provides the identification of the originator. It confirms to the receiver that the data received has been sent only by an identified and verified sender.

Authentication service has two variants –

- **Message authentication** identifies the originator of the message without any regard router or system that has sent the message.
- **Entity authentication** is assurance that data has been received from a specific entity, say a particular website.

Apart from the originator, authentication may also provide assurance about other parameters related to data such as the date and time of creation/transmission.

Non-repudiation

It is a security service that ensures that an entity cannot refuse the ownership of a previous commitment or an action. It is an assurance that the original creator of the data cannot deny the creation or transmission of the said data to a recipient or third party.

Non-repudiation is a property that is most desirable in situations where there are chances of a dispute over the

exchange of data. For example, once an order is placed electronically, a purchaser cannot deny the purchase order, if non-repudiation service was enabled in this transaction.

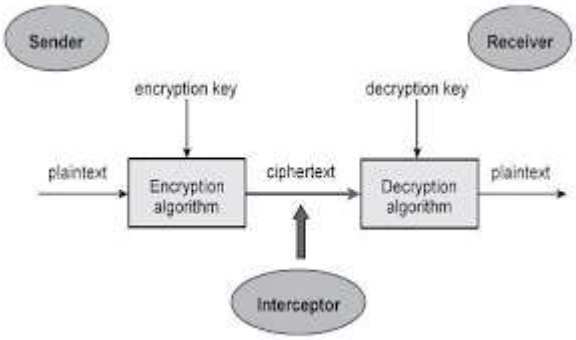

Fig-1: working of Cryptosystem

2. LITERATURE SURVEY

2.1 Types of Cryptosystems

Fundamentally, there are two types of cryptosystems based on the manner in which encryption-decryption is carried out in the system –

- Symmetric Key Encryption
- Asymmetric Key Encryption

The main difference between these cryptosystems is the relationship between the encryption and the decryption key. Logically, in any cryptosystem, both the keys are closely associated. It is practically impossible to decrypt the cipher text with the key that is unrelated to the encryption key.

2.1.1 Symmetric Key Encryption

The encryption process where same keys are used for encrypting and decrypting the information is known as Symmetric Key Encryption. The study of symmetric cryptosystems is referred to as symmetric cryptography. Symmetric cryptosystems are also sometimes referred to as secret key cryptosystems.

A few well-known examples of symmetric key encryption methods are – Digital Encryption Standard (DES), Triple-DES (3DES), IDEA, and BLOWFISH.

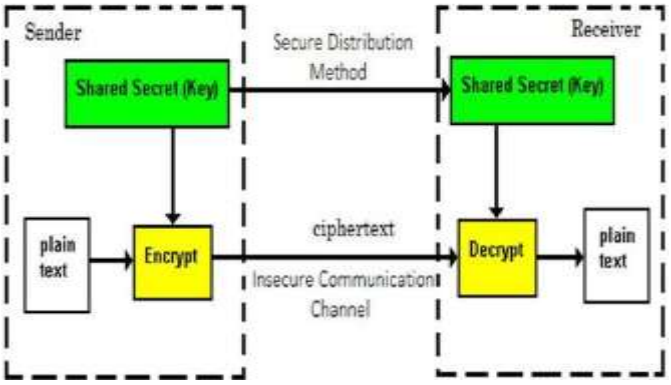

Fig-2: Symmetric key Encryption

2.1.2 Asymmetric Key Encryption

The encryption process where different keys are used for encrypting and decrypting the information is known as Asymmetric Key Encryption. Though the keys are different, they are mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible. The process is depicted in the following illustration –

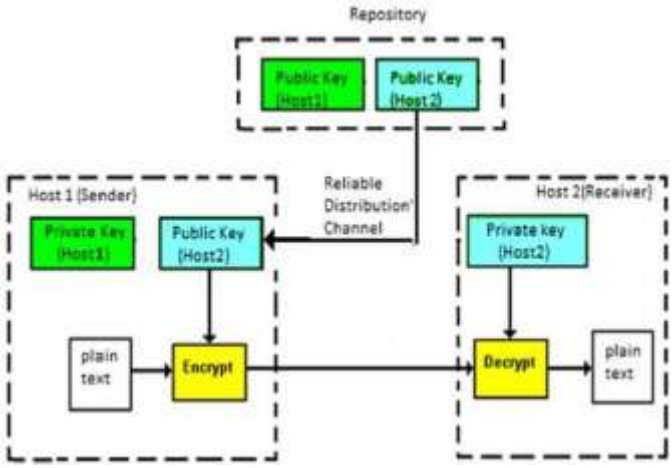

Fig-3: Asymmetric Key Encryption

2.1.2.1 RSA (*Rivest Shamir Adelman*)

RSA is considered as the first real life and practical asymmetric-key cryptosystem. The security of RSA lies with integer factorization problem. Here, the key generation is done by each party, once key generation gets over, they can communicate each other securely. The algorithm for RSA is given below.

RSA Algorithm

Key Generation

Step I. Select p, q

p and q both are primes, $p \neq q$

Step II. Calculate $n = pq$

Step III. Calculate $\Phi(n) = (p-1)(q-1)$

Step IV. Select integer e,

$\gcd(\Phi(n), e) = 1; 1 < e < \Phi(n)$

Step V. Calculate d,

$d \equiv e^{-1} \pmod{\Phi(n)}$

Step VI. Public key PU = $\{e,n\}$

Step VII. Private key PR = $\{d, n\}$

Encryption

Step I. Plaintext: $M<n$

Step II. Cipher text: $C=M^e \bmod n$

Decryption

Step I. Cipher text: C

Step II. Plaintext: $M=C^d \bmod n$

Here, key generation is to be done by each party, so that they can communicate each other securely. In the RSA algorithm, 'e' is for encryption, should be chosen such that gcd(Φ(n), e) is equal to 1. Once 'e' is selected, corresponding, 'd' that is for decryption should be generated with the help of finding the inverse of 'e' mod Φ(n).

In encryption process, a sender has to encrypt the message (i.e., in decimal digit) with the help of receiver's public key, i.e., 'e' and 'n'.

In decryption process, the receiver has to decrypt the cipher text with the help of his private key, i.e., 'd' and 'n'.

2.1.2.2 ECC (Elliptic Curve Cryptography)

ECC is promising asymmetric key cryptosystems, this type of systems is most suitable for memory constraint devices such as Smartphone etc. An ECC requires comparatively less or smaller parameters for encryption and decryption than RSA, but with equivalent levels of security.

ECC Algorithm

ECC algorithm exhibits key generation, encryption, and decryption.

Global Public Elements

Step I. Chooses an elliptic curve Eq(a, b) with parameters a, b, and q, where q is a prime and > 3, or an integer of the form 2^m.

Step II. Selects G(x, y) - a global point on elliptic curve whose order is large value n.

User Alice Key Generation

Step I. Selects a private key, VA; where, VA < n

Step II. Calculates the public key, PA(x, y)

PA(x, y) = VA × G(x, y).

User Bob Key Generation

Step I. Selects a private key, VB; where, VB < n.

Step II. Calculates the public key, PB(x, y);

PB(x, y) = VB × G(x, y).

Calculation of Secret Key by User Alice

Step I. SK(x, y) = VA × PB(x, y)

Calculation of Secret Key by User Bob

Step I. SK(x, y) = VB × PA(x, y).

Encryption by Alice using Bob's Public Key

Step I. Alice chooses message Pm(x, y) and a random positive integer 'k' and 1 < k < q

Step II. 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

Step I. Ciphertext, Cm((x, y),(x, y))

Step II. Plaintext, Pm(x, y);

= (Pm(x, y) + k × PB(x, y)) - (k × VB × G(x, y))

= Pm(x, y).

Here, first coordinate of Cm gets multiplied with the private key of the Bob i.e., VB, which in turns becomes similar to Bob's public key. Finally, due to subtraction of resultant

coordinate with the second coordinate of the ciphertext Cm, all get canceled and only Pm(x, y) gets left.

Table-1: RSA and ECC –Cryptography Key Length (In BITS)

	Key Size	
Security Bits Level	**RSA**	**ECC**
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	1536	512

3. COMPARISION OF RSA AND ECC

This paper implements RSA and ECC for secrecy of the information. The efficiency of ECC over RSA is shown in Fig Based on experimentation, it is observed that RSA is very efficient in encryption but slow in decryption while ECC is slow in encryption but very efficient in decryption. Overall ECC is more efficient and secure than RSA as shown in the figures and table

Table-2: 8 BITS ENCRYPTION, DECRYPTION AND TOTAL TIME (IN SECONDS)

Security	Encryption		Decryption		Total	
Bits	ECC	RSA	ECC	RSA	ECC	RSA
80	0.4885	0.0307	1.3267	0.7543	1.8152	0.7850
112	2.2030	0.0299	1.5863	2.7075	3.7893	2.7375
128	3.8763	0.0305	1.7690	6.9409	5.6453	6.9714
144	4.7266	0.0489	2.0022	13.6472	6.7288	13.6962

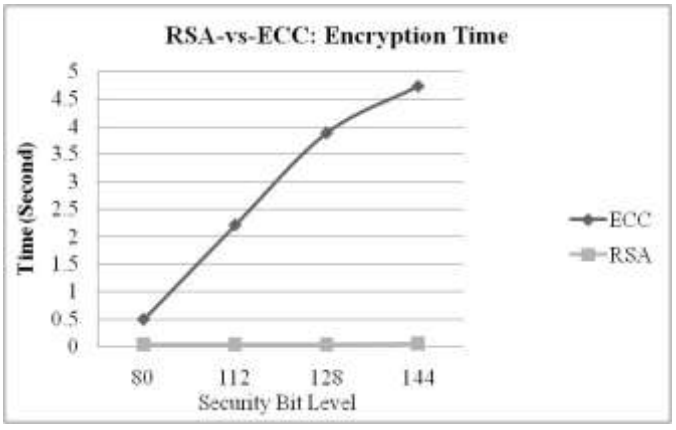

Fig-4: 8 bits - Encryption Time (in seconds)

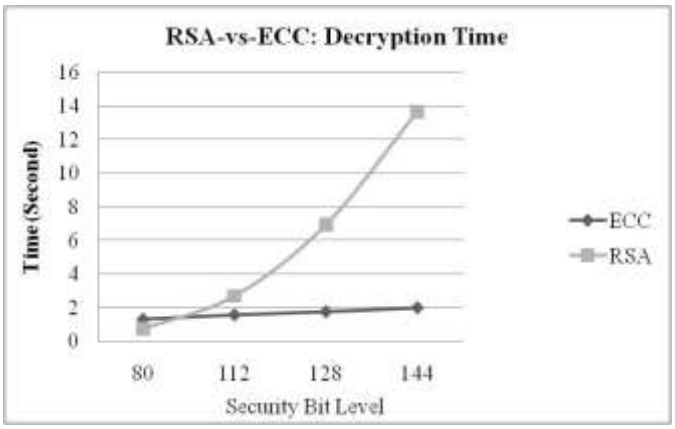

Fig-5: 8 bits - Decryption Time (in seconds)

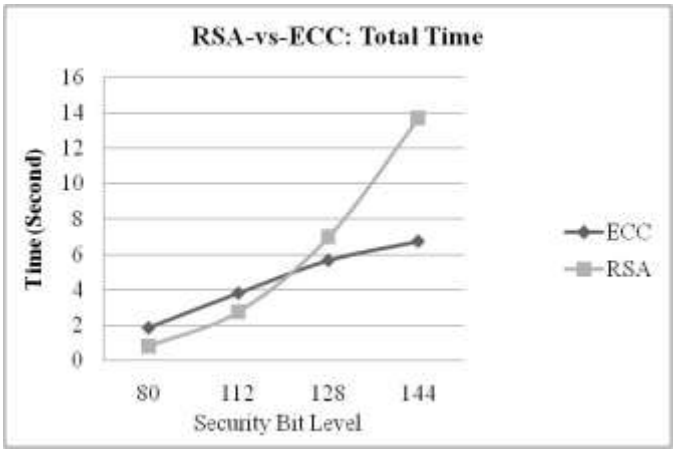

Fig-6: 8 bits - Total (Enc. & Dec.) Time (in seconds)

Table 3: 64 BITS ENCRYPTION, DECRYPTION AND TOTAL TIME (IN SECONDS)

Security	Encryption		Decryption		Total	
Bits	ECC	RSA	ECC	RSA	ECC	RSA
80	2.1685	0.1366	5.9099	5.5372	8.0784	5.6738
112	9.9855	0.1635	6.9333	20.4108	16.9188	20.5743
128	15.0882	0.1672	7.3584	46.4782	22.4466	46.6454
144	20.2308	0.1385	8.4785	77.7642	28.7093	77.9027

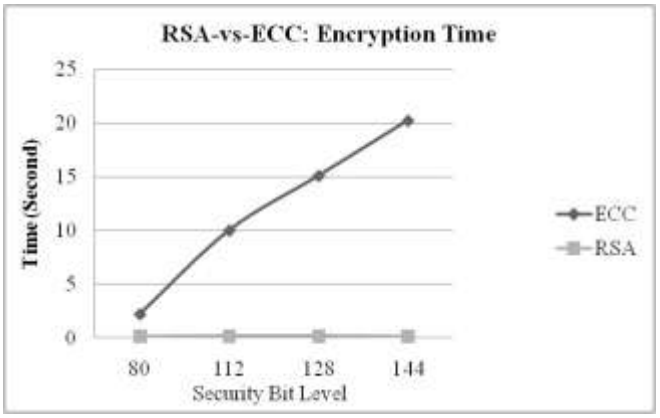

Fig-7. 64 bits - Encryption Time (in seconds)

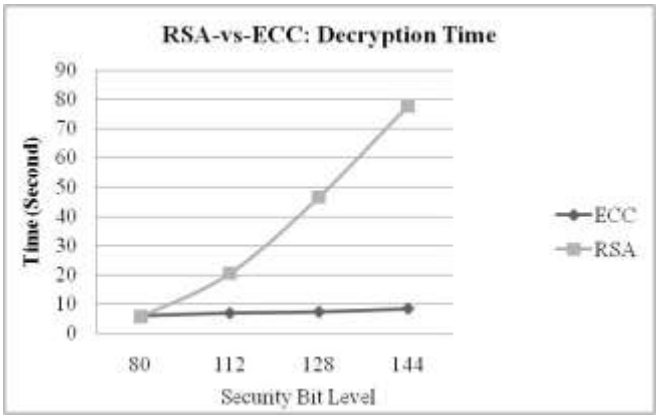

Fig-8. 64 bits - Decryption Time (in seconds)

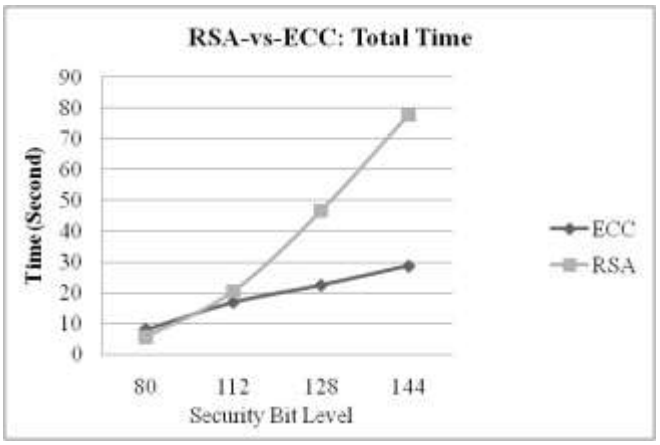

Fig-9. 64 bits - Total (Enc. & Dec.) Time (in seconds)

4. CONCLUSION

Security of data communication is very important while data are being transmitted from one user to another user or system. Cryptography is one of the techniques to provide data communication security. This paper presented a performance study and an analysis of RSA and ECC. Based on experimentation, it was found that the elliptic curve discrete logarithm problem makes ECC most efficient.

For better and stronger security of data, bigger key sizes require, which means more overhead on the computing systems. Nowadays small devices are playing an important role in the digital world, which has less memory but needs security to cope with market demand. In this scenario, RSA becomes second thoughts and ECC become first.

A comparative analysis of both the algorithms has been done and observed that RSA is one of the effective public key cryptographic algorithms, which needs time and memory whereas ECC provides a strong alternative with smaller key lengths and more secure.

5. FUTURE SCOPE

There is a direct relationship between variable key lengths and level of security in asymmetric encryption algorithms. The bigger the key size, the more secure the algorithm it is. But in the other hand, bigger key size requires more computational power and resources. And rationally these prerequisites will lower the algorithm's performance.

The need to improve the performance ECC can be satisfied by improvement in scalar multiplication algorithm as the fundamental algorithm in ECC. To achieve this, we need to design an efficient algorithm that can enhance both scalar arithmetic and point arithmetic. Furthermore, some security measurement checking need to be performed to ensure the security of algorithm against side channel attacks. This algorithm hopes to be well-balanced in term of its cost and effectiveness and will surpass the other existing algorithm in term of efficiency and effectiveness.

ACKNOWLEDGEMENT

We would like to thank our colleagues, Head of Department of Computer Applications, Dean (R & C) and Director of our Institute for guiding directly or indirectly in this research work.

REFERENCES

1. N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz, “Comparing Elliptic Curve Cryptography and RSA on 8-bit cpus,” in *Cryptographic Hardware and Embedded Systems - CHES 2004*, ser. Lecture Notes in Computer Science, M. Joye and J. Quisquater, Eds. Springer Berlin Heidelberg, 2004, vol. 3156, pp. 119–132.
2. V. B. Kute, P. Paradhi, and G. Bamnote, “A software comparison of rsa and ecc,” *Int. J. Comput. Sci. Appl*, vol. 2, no. 1, pp. 43–59, 2009.

3. B. Alese, E. Philemon, and S. Falaki, "Comparative analysis of publickey encryption schemes," *International Journal of Engineering and Technology*, vol. 2, no. 9, pp. 1552–1568, 2012.
4. J. Bos, M. Kaihara, T. Kleinjung, A. K. Lenstra andP. L. Montgomery,On the Security of 1024-bit RSAand 160-bit Elliptic Curve Cryptography, TechnicalReport, 2009.
5. D. Hankerson, A. J. Menezes and S. Vanstone,Guideto Elliptic Curve Cryptography, Springer Science &Business Media, 2006.
6. D. Mahto, D. A. Khan and D. K. Yadav, "Securityanalysis of elliptic curve cryptography and RSA," inProceedings of the World Congress on Engineering,vol. 1, 2016.
7. D. Mahto and D. K. Yadav, "Rsa and ECC: A com-parative analysis,"International Journal of AppliedEngineering Research, vol. 12, no. 19, pp. 9053–9061,2017.
8. M. J. B. Robshaw and Y. L. Yin, "Elliptic curvecryptosystems,"An RSA Laboratories TechnicalNote, vol. 1, p. 997, 1997.
9. S. R. Singh, A. K. Khan and S. R. Singh, "Perfor-mance evaluation of RSA and elliptic curve cryp-tography," in2nd International Conference on Con-temporary Computing and Informatics (IC3I'16),pp. 302–306, 2016.
10. Dr.M.Gobi, R.Sridevi, R.Rahini priyadharshini, A Comparative Study on the Performance and the Security of RSA and ECC Algorithm, National Conference on Advanced Networking and Applications, March 2015.
11. N.,Koblitz. Elliptic curve cryptography, Mathematics of Computation 48: 203-209. 1987.
12. W. Stallings, Cryptography and Network Security: Principles and Practice,Prentice Hall, Sixth Edition, 1998.
13. Maqsood F, Ahmed M, Mumtaz M, Ali M. Cryptography: A Comparative Analysis for Modern Techniques. Int J Adv Comput Sci Appl. 2017;8(6):442–8.
14. Kak A. Lecture 12 : Public-Key Cryptography and the RSA Algorithm Lecture Notes on " Computer and Network Security " by Avi Kak (kak@purdue.edu) Goals : Comput Netw Secur. 2018;1–94.
15. Rivest, R.L., Shamir, A. and Adleman, L., 1978. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), pp.120-126.
16. Jansma, N. and Arrendondo, B., 2004. Performance comparison of elliptic curve and rsa digital signatures. nicj. net/files.
17. Gura, N., Patel, A., Wander, A., Eberle, H. and Shantz, S.C., 2004, August. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In CHES (Vol. 4, pp. 119-132)