International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307

The Implementation and Comparative Analysis for Security Algorithms in Cloud Environment

D. Pharkkavi[1], Dr. D. Maruthanayagam[2]

[1]Research Scholar, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India
[2]Head/Professor, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India

ABSTRACT

Cloud computing is a rising technology that is still unclear to many security issues. The cloud data and services reside in massively scalable data centers and can be accessed everywhere. The growth of the cloud users has unfortunately been accompanied with a growth in malicious activity in the cloud. More and more vulnerabilities are discovered, and nearly every day, new security advisories are published. Millions of users are surfing the Cloud for various purposes, therefore they need highly safe and persistent services. So the most challenging issue today in cloud servers is to ensure data security and privacy of the users. The different algorithms have been proposed to provide security to critical data. This paper gives a comparative analysis of some existing security algorithms in Cloud Computing. To ensure the security of data in cloud environment, also analyzed the performance of existing algorithm based on three parameters namely Time Complexity, Space Complexity and Throughput.

Keywords: Cloud Computing, Security, Encryption, Decryption, Public Key and Private Key.

I. INTRODUCTION

Cloud Computing is the key driving force in many small, medium and large sized companies [1-2]. Cloud computing has three delivery models named as Saas, Iaas, Paas and four deployment models such as private cloud, public cloud, hybrid cloud and community cloud. As many cloud users seeks the services of cloud computing, the major concern is the security of their data in the cloud [5]. Data security is always of vital importance and plays an important role in trust worthiness of computing [4]. Due to the critical nature of cloud computing and the large amounts of complex data it carries, the need is even more important [3]. Hence forth, concerns regarding data privacy and security are proving to be a barrier to the broader uptake of cloud computing services [6]. Security is considered as one of the most critical aspects in everyday computing and it is not different for cloud computing due to sensitivity and importance of data stored on the cloud. Cloud Computing infrastructure uses new technologies and services, most of which haven't been fully evaluated with respect to the security [7]. Cloud Computing has several major issues and concerns, such as data security, trust, expectations, regulations, and performances issues [8-9].

To provide secure communication over the network, encryption algorithm plays a vital role. It is the fundamental tool for protecting the data. Encryption algorithm converts the data into scrambled form by using "the key" and only user have the key to decrypt the data. There are two main categories of encryptions used in cryptography to achieve data confidentiality, integrity, availability, authentication and non-repudiation. Non-repudiation means that when something has been sent from someone, there has to be a way to track back to the sender. There are symmetric and asymmetric encryption algorithms.

In symmetric encryption algorithm, encryption and decryption requires that the same algorithm and key are used to both encipher and decipher the message. There is a private key that is used to encrypt and decrypt the message at both ends. Symmetric encryption key method is extremely fast and efficient for processing encrypts and decrypt message. Symmetric encryption algorithm provides confidentiality, integrity and availability but it fails to provide authenticity and non-repudiation. Under the symmetric encryption algorithms are: Data encryption standard **(DES),** Advanced encryption standard **(AES)**, Ron's code, Triple DES and etc.

Asymmetric encryption algorithm uses two keys instead of one. One is a private key only known to the recipient of the message and the other is a public key known to everyone and can be freely distributed. Either key can be used to encrypt and decrypt the message. However if only key A is used to encrypt the message then only key B can be used to decrypt it. Conversely, if key B is used to encrypt the message then only key A can be used to decrypt it. Asymmetric algorithms are slower than symmetric algorithms. But it has better key distribution than symmetric algorithm. It has better scalability and also provides authenticity and non-repudiation. While examples of asymmetric encryption are: **RSA**, **Elliptic curve** and Diffie-Hellman **(DH)**.

In this paper, some existing security algorithms are discussed which can be implemented to the cloud to provide security. As discussed there are many security algorithms which are currently available in cloud computing. Apart from these there is a great need to develop many more efficient algorithms to increase the security level of cloud computing. A further enhancement can be done in the existing algorithms so that security of data in cloud can be increased. As the number of users is increasing rapidly in cloud computing with the passage of time so it becomes major issue to make their data completely secure. So, we are in a position to need some **better algorithm** to improve security in cloud computing environment.

II. COMPARISON SURVEY OF SECURITY ALGORITHMS

Encryption is the process of converting a plaintext message into cipher text which can be decoded back into the original message. An encryption algorithm along with a key is used in the encryption and decryption of data. There are several types of data encryptions which form the basis of network security. Encryption schemes are based on block or stream ciphers. The type and length of the keys utilized depend upon the encryption algorithm and the amount of security needed. In conventional symmetric encryption a single key is used. With this key, the sender can encrypt a message and a recipient can decrypt the message but the security of the key becomes problematic. In asymmetric encryption, the encryption key and the decryption key are different. One is a public key by which the sender can encrypt the message and the other is a private key by which a recipient can decrypt the message. In this paper presenting the performance evaluation ***based on some new key factors*** for popular security algorithms. After that, how those algorithms are utilized for data security in cloud computing.

1. RSA ALGORITHM

RSA is widely used Public-Key algorithm. **RSA** stands for ***Ron Rivest***, ***Adi Shamir*** and ***Len Adleman***, who first publicly described it in 1977.User data is encrypted first and then it is stored in the Cloud. When required, user places a request for the data for the Cloud provider, Cloud provider authenticates the user and delivers the data.

RSA is a block cipher, in which every message is mapped to an integer. RSA consists of Public-Key and Private-Key. In our Cloud environment, Pubic-Key is known to all, whereas Private-Key is known only to the user who originally owns the data. Thus, encryption is done by the Cloud service provider and decryption is done by the Cloud user or consumer. Once the data is encrypted with the Public-Key, it can be decrypted with the corresponding Private-Key only. How RSA is going to work in cloud environment is explained as: RSA algorithm is used to ensure the security of data in cloud computing. The ***advantage*** of ***RSA is Strong*** and ***can't easily break*** by Cryptanalyst. The ***disadvantage*** is ***slow*** compare than AES. ***Key Size is high*** with compare AES and other symmetric algorithms. This asymmetric algorithm has encrypted our data to provide security. The purpose of securing data is that only concerned and authorized users can access it. After encryption data is stored in the cloud. So that when it is required then a request can be placed to cloud provider. Cloud provider authenticates the user and delivers the data to user. As RSA is a Block Cipher in which every message is mapped to an integer. In the proposed cloud environment, Public key is known to all, whereas Private Key known only to user who originally owns the data. Thus encryption is done by the cloud service provider and decryption is done by the cloud user or consumer. Once the data is encrypted with the

Public key, it will be decrypted using the corresponding Private Key only.RSA algorithm involves three steps:

1. Key Generation
2. Encryption
3. Decryption

1. Key Generation:

Before the data is encrypted, Key generation should be done. This process is done between the Cloud service provider and the user.

Steps:

1. Choose two distinct prime numbers a and b. For security purposes, the integers a and b should be chosen at random and should be of similar bit length.
2. Compute n = a * b.
3. Compute Euler's totient function, Ø(n) = (a-1) * (b-1).
4. Chose an integer e, such that 1 < e <Ø(n) and greatest common divisor of e , Ø(n) is 1. Now e is released as Public-Key exponent.
5. Now determine d as follows: d = e-1(mod Ø(n)) i.e., d is multiplicate inverse of e mod Ø(n).
6. d is kept as Private-Key component, so that d * e = 1 mod Ø(n).
7. The Public-Key consists of modulus n and the public exponent e i.e, (e, n).
8. The Private-Key consists of modulus n and the private exponent d, which must be kept secret i.e, (d, n).

2. Encryption:

Encryption is the process of converting original plain text (data) into cipher text (data).

Steps:

1. Cloud service provider should give or transmit the Public-Key (n, e) to the user who wants to store the data with him or her.
2. User data is now mapped to an integer by using an agreed upon reversible protocol, known as padding scheme.
3. Data is encrypted and the resultant cipher text (data) C is C = me (mod n).
4. This cipher text or encrypted data is now stored with the Cloud service provider.

3. Decryption:

Decryption is the process of converting the cipher text (data) to the original plain text (data).

Steps:

1. The cloud user requests the Cloud service provider for the data.
2. Cloud service provider verify's the authenticity of the user and gives the encrypted data i.e, C.
3. The Cloud user then decrypts the data by computing, m = Cd (mod n).
4. Once m is obtained, the user can get back the original data by reversing the padding scheme.

2. AES ALGORITHM

Advanced Encryption Standard **(AES)**, also known as Rijindael is used for securing information. AES is a symmetric block cipher that has been analyzed extensively and is used widely now-a-days. How AES works in cloud environment? AES, symmetric key encryption algorithm is used with key length of 128-bits for this purpose. As AES is used widely now-a-days for security of cloud. The ***advantage*** of this algorithm ***speed*** *and* ***less key size*** compares RSA. The ***disadvantage*** is ***less strong*** compare RSA. The Implementation proposal states that First, User decides to use cloud services and will migrate his data on cloud. Then User submits his services requirements with Cloud Service Provider (**CSP)** and chooses best specified services offered by provider. When migration of data to the chosen CSP happens and in future whenever an application uploads any data on cloud, the data will first encrypted using AES algorithm and then sent to provider. Once encrypted, data is uploaded on the cloud, any request to read the data will occur after it is decrypted on the users end and then plain text data can be read by user. The plain text data is never written anywhere on cloud. This includes all types of data. This encryption solution is transparent to the application and can be integrated quickly and easily without any changes to application. The key is never stored next to the encrypted data, since it may compromise the key also. To store the keys, a physical key management server can be installed in the user‘s premises. This encryption protects data and keys and guarantees that they remain under user‘s control and will never be exposed in storage or in transit. AES has replaced the DES as approved standard for wide range of applications**.** The following figure describes the working method of AES Algorithm.

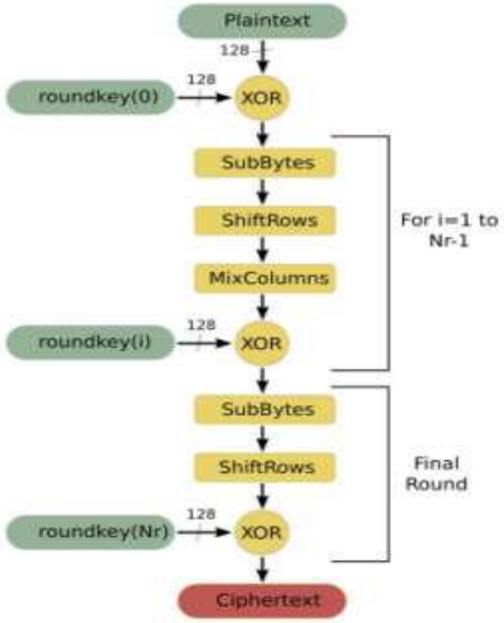

Figure 1. The Working Method for AES Algorithm.

AES work on blocks of three different sizes 128 bit, 192 bit and 256 bits. AES -128 uses 10 rounds, AES-192 has 12 rounds and AES-256 consists of 14 rounds. Each round goes through a series of steps like substitution byte, shift rows, mixed columns and adds round Key. AES Algorithm is comparatively ***more secure*** and has a ***strong*** avalanche effect. Attackers cannot easily decrypt the encrypted text by the brute force attack. Therefore AES has been used in many applications like it is used PDA communication.

3. ECC ALGORITHM

Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller from IBM and Neil Koblitz from University of Washington as an alternative mechanism for implementing public-key cryptography. This ECC (Elliptic Curve Cryptography) is Based on algebraic structures of elliptic curves over finite fields i.e. elliptic curve theory. ECC Create Faster, Smaller and more efficient keys as compared to other encryption algorithm. In this, encryption is done in elliptic curve equation (used in mathematics) form. ECC is that much efficient that it can yield a level of security with 164 bit key that other system require a 1,024-bit key to achieve that security level i.e.it offers the maximum security with smaller bit sizes that is why it consumes less power[25] and hence, Elliptic curve cryptography is good for battery backup also. The use of elliptic curves in cryptography was suggested independently by Neal Koblitz and Victor S. Miller in 1985. Elliptic curve cryptography algorithms entered wide use in 2004 to 2005. Basically, an elliptic curve is a plane curve over a finite field (rather than the real numbers).

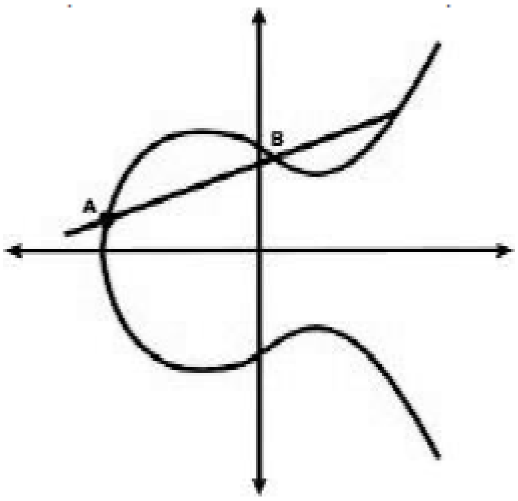

Figure 2. Elliptic Curve Representations

Figure 2. Elliptic Curve Representation Which consists of the point values satisfying the equation,

$$y^{2} = x^{3}+Ax+B\text{-------}(1)$$

Where a and b are the constant point values. In the encryption process of Elliptic curve cryptography, we have many options to use ECC cryptography but we will discuss simplest way.

According to this encryption technique,

1. The sender must first encode any message M as a point on the elliptic curve Pm.
2. The user must first encode any message M as a point on the elliptic curve Pm.
3. Select suitable curve & point G as in D-H.
4. Each user chooses private key nA<n and computes public key PA=nAG
5. For encryption encrypt: Pm : Cm={kG, Pm+kPb}, where k is a random number
6. For decryption decrypt Cm compute: m+*k*Pb–nB(*kG*) = Pm+*k*(nB*G*)–nB(*kG*) = Pm

The ***main advantage*** of ECC uses short key length which leads to ***fast encryption speed*** and ***less power consumption***. For example, a 160 bit ECC encryption key size provide the same level of security as 1024-bit RSA encryption key and it perform 15 times faster depending upon the platform on which it is implemented. The disadvantage of ECC is that it increases the size of encrypted text and second ***disadvantage*** is that ECC is dependent on very ***complex equations*** which lead to increase the complexity of encryption algorithm.

III. COMPARATIVE ANALYSIS AND EXPERIMENTAL RESULTS

Each of the encryption techniques has its ***own strong and weak points***. In order to apply a suitable cryptography algorithm to an application, we should have knowledge regarding performance, strength and weakness of the algorithms. Therefore, these algorithms must be analyzed based on ***several key factors***. In this paper, analysis is done with following ***metrics*** under which the cryptosystems can be compared: Encryption time, Decryption time, Time Complexity, Memory Space and Throughput.

The encryption time is considered to be the time that an encryption algorithm takes to generate a cipher text from a plain text. Encryption time is calculated by the throughput of an encryption scheme, is calculated as the total plaintext in bytes encrypted divided by the encryption time. Comparisons analyses of the results of the selected different encryption algorithm are performed. The different types and sizes of files with extension such as .jpeg, .txt, .doc, and .pdf are used to conduct experiments, where a comparison of three algorithms AES, RSA and ECC are performed. Cryptographic tool is used to conduct experiments.

The implementation of results in this section highlights the time of execution in upload and in download of files with different sizes. Our application is developed in java 7. The result obtained from Intel (R) Core (TM) 2 Duo CPU @ 3.48GHz (2 CPUs), with 4GB of RAM. The download time is greater than the upload time. This is explained by the addition of key recovery time on server.

In this section we have to implement AES, RSA and ECC algorithms and then analyze its performance based on different parameters such as Time complexity, Space complexity and through put. **Time Complexity:** Time complexity is commonly calculated by counting the total operations performed by the system where each operation takes a fixed amount of time. An algorithm performance time may vary with different input size therefore it is a common practice to express the time complexity in worst case denoted as T (n).

For instance the algorithm with T(n)=O(n) has linear time complexity whereas T(n)=O(n^2) is nonlinear and T(n)=O(2^n) is exponential. In our case we have computed the time complexity by varying the Private Key length of the RSA algorithm and finding the required execution time for each Private Key length. The time complexity of these three algorithms is analyzed by varying the key size in bits and noting the execution time for each key length. Here Chosen File is, **sun.docx** and File Size: **42265KB.**

Table 1. Time Complexity of ECC Algorithm

ECC Algorithm				
Key size	**Key Gen Time(ms)**	**Encrypt Time(ms)**	**Decrypt Time(ms)**	**Time Complexity(MS)**
160	25.21	12.34	13.81	147.67
224	27.34	29.98	10.33	237.60
256	31.45	30.39	66.45	457.00
384	36.23	37.72	79.12	779.23
512	37.23	46.98	88.71	901.12

Table 2. Time Complexity of RSA Algorithm

RSA Algorithm				
Keysize	**Key Gen Time(ms)**	**Encrypt Time(ms)**	**Decrypt Time(ms)**	**Time Complexity(MS)**
512	35.23	37.53	25.35	212.45
1024	39.01	40.22	31.12	542.41
2048	42.23	41.53	32.01	678.96
4096	45.01	43.65	40.17	817.32
8192	47.86	44.77	41.52	968.23

Table 3. Time Complexity of AES Algorithm

AES Algorithm				
Keysize	**Key Gen Time(ms)**	**Encrypt Time(ms)**	**Decrypt Time(ms)**	**Time Complexity(MS)**
128	45.23	42.53	15.35	245.37
192	39.01	40.22	37.12	601.04
256	52.23	38.53	36.57	701.12
320	53.01	39.65	38.45	905.22
384	54.86	40.77	40.52	997.27

Table 4. Comparison of Time Complexity between ECC,RSA and AES Algorithms

Time Complexity(MS)			
Key size (ECC:RSA:AES)	**ECC**	**RSA**	**AES**
160: 512: 128	147.67	212.45	245.37
224: 1024: 192	237.60	542.41	601.04
256: 2048: 256	457.00	678.96	701.12
384: 4096: 320	779.23	817.32	905.22
512: 8192: 384	901.12	968.23	997.27

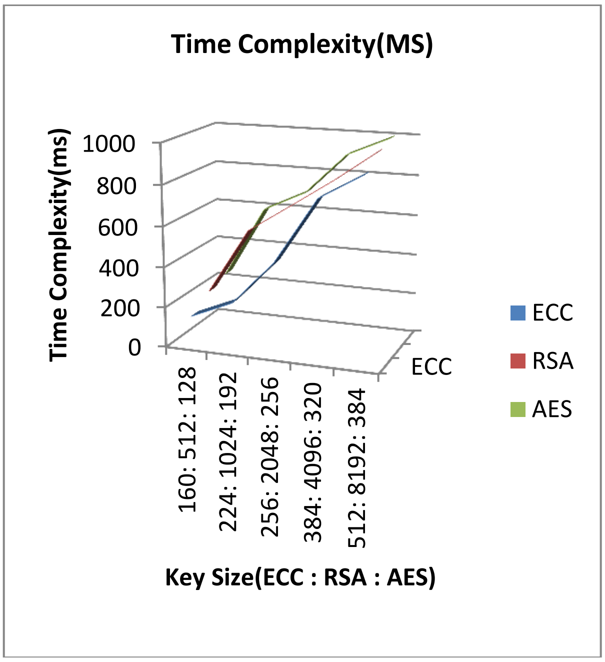

Figure 3. Comparison of Time Complexity between ECC,RSA and AES Algorithms

Space complexity: Apart from Time complexity, space complexity is also an important measure to judge the performance of an algorithm. It is the amount of memory which the algorithm needs for performing its computations. A good algorithm keeps the amount of memory as small as possible. The way in which the amount of storage space required by an algorithm varies with the size of the problem it is solving. Space complexity is normally expressed as an order of magnitude, e.g. O(N^2) means that if the size of the problem (n) doubles then four times as much working storage will be needed. We have analyzed the space complexity between private key length which is in bits and run time memory consumed by system. A summary of the different Private Key length in bits and run time memory taken by the system is given below in table 5 .

Table 5. Space complexity(Bits) and run time memory

Space Complexity(Bits)			
Key size (ECC:RSA:AES)	**ECC**	**RSA**	**AES**
160: 512: 128	236210	347320	456431
224: 1024: 192	237030	348040	459151
256: 2048: 256	237507	348608	459719
384: 4096: 320	238377	349488	450599
512: 8192: 384	240037	351048	462159

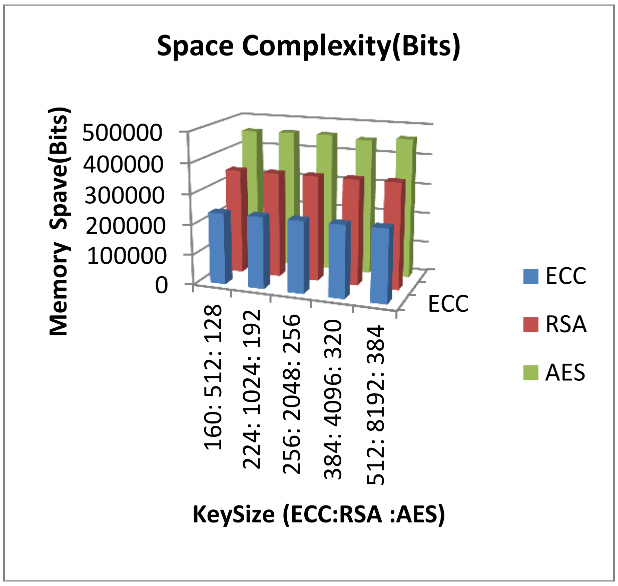

Figure 4. Comparison of Space complexity (Bits) and run time memory in ECC, RSA and AES

Throughput: In communication systems throughput is the rate of successful data delivery over a noisy communication channel. Throughput is usually measured in bits per second and sometimes we measure it in terms of packets per second. We have calculated the throughput of the algorithm by dividing the total data in bytes by encryption time. Higher the throughput higher is the efficiency of the system. Table given below gives us the comparison between the throughput and the message signal. We have calculated the throughput for 32, 64,128 and 256 bytes of messages. In any cryptographic algorithm, it is essential to understand the size of the input and the size of output as this is one of the important property of an avalanche effect. Larger the size of the Cipher text compared with the Plaintext, more secure is the Cipher text against any Brute-Force attack. The table 3 below gives us the throughput for different data length.

Table 6. Throughput (Bits/Sec) Comparison of AES, RSA and ECC for different data length

Throughput (Bits/Sec)				
	Keysize (ECC:RSA: AES)	**ECC**	**RSA**	**AES**
Sample.txt (3445KB)	160: 512: 128	1241.841	1352.952	2463.063
	224: 1024: 192	983.13	1094.024	2005.13
	256: 2048: 256	573.141	684.251	795.361
	384: 3096: 320	332.56	443.67	554.78
	512: 4192: 384	0.114	0.2254	1.1265
sun.docx (42265KB)	160: 512: 128	11845.00	15956.12	26067.23
	224: 1024: 192	9967.23	10213.01	21324.12
	256: 2048: 256	7852.02	8756.88	9867.90
	384: 3096: 320	325.33	567.12	678.25
	512: 4192: 384	33.52	77.26	95.37
Flower.PDF (567336KB)	160: 512: 128	102356.43	138452.54	249563.65
	224: 1024: 192	85623.56	102159.02	213260.13
	256: 2048: 256	5683.52	78124.96	89235.07
	384: 3096: 320	1325.57	6132.52	7243.63
	512: 4192: 384	253.01	717.26	829.37

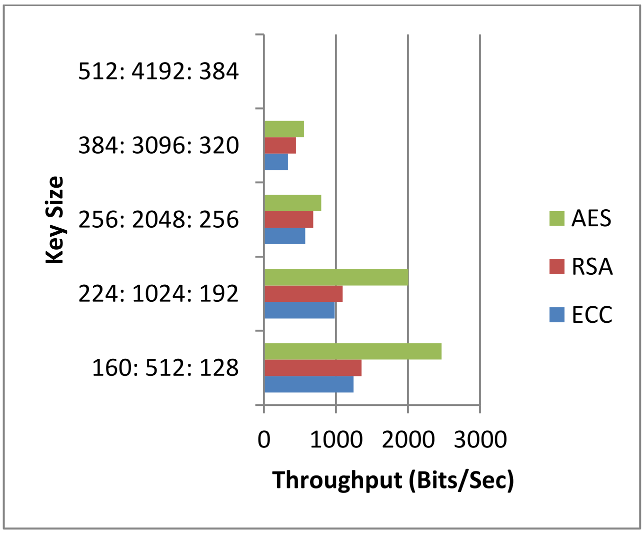

Figure 5: Throughput (Bits/Sec) Comparison of AES, RSA and ECC from Sample.txt(3445KB) file execution

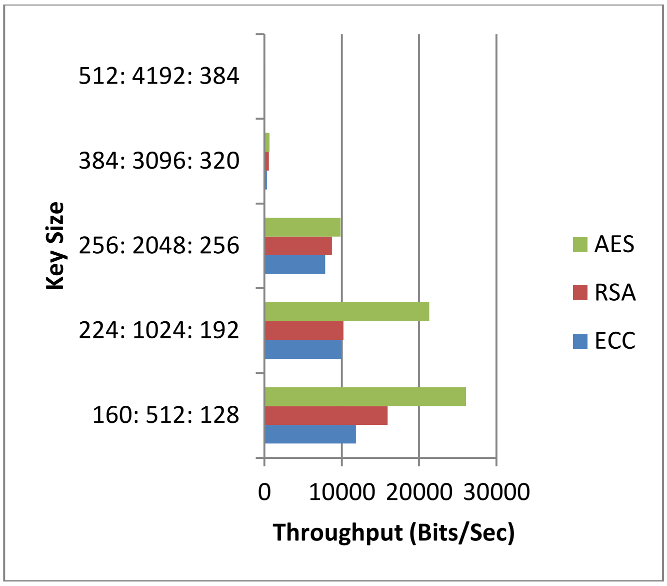

Figure 6. Throughput (Bits/Sec) Comparison of AES, RSA and ECC from sun.docx (42265KB) file execution

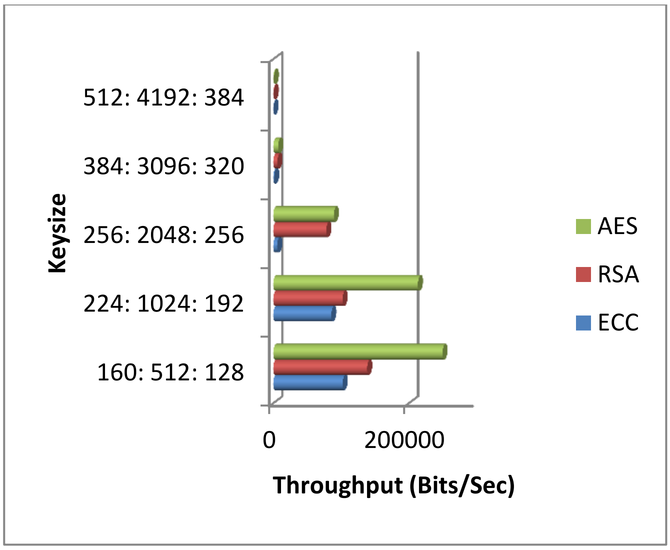

Figure 7. Throughput (Bits/Sec) Comparison of AES, RSA and ECC from Flower.pdf (567336KB) file execution

IV. CONCLUSION

In today's world, Cloud computing is rising as a new brand factor. There are several issues in cloud computing but the major issue concerns security issue. Many of the organizations are moving their data on the cloud but are concerned about security of their data. Thus cloud security is must which will be able to break the hindrance (barrier) to the acceptance of the cloud by the organizations. There are number of existing techniques used to implement security in cloud. This paper has been described so far the performance comparison of various security algorithms in cloud computing environment. In this work, those algorithms have evaluated based on three parameters namely (**Time Complexity, Space Complexity and Throughput)**.Our future work will be considering some problems related to existing security algorithms and implements a better version of **DES, 3DES, AES, RSA, IDES, Blowfish**.

V. REFERENCES

[1]. Kevin Hamlen, Murat Kantarcioglu, Latifur Khan, BhavaniThuraisingham, "Security Issues for Cloud Computing". The University of Texas at Dallas, USA, International Journal of Information Security and Privacy, 4(2), 39-51, April-June 2010.

[2]. Anup R. Nimje, "Cryptography In Cloud-Security Using DNA (Genetic) Techniques", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue5, pp.1358-1359, September- October 2012.

[3]. Danish Jamil, Hassan Zaki, "Cloud Computing Security", International Journal of Engineering Science and Technology (IJEST).

[4]. RachnaArora, AnshuParashar, "Secure User Data in Cloud Computing Using Encryption Algorithms", International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, www.ijera.com , Vol. 3, Issue 4, pp.1922-1926, Jul-Aug 2013.

[5]. Engr: Farhan Bashir Shaikh, SajjadHaider, "Security Threats in Cloud Computing", 6th International Conference on Internet Technology and Secured Transactions, 11-14 December 2011, Abu Dhabi, United Arab Emirates.

[6]. Dr. P. Dinadayalan, S. Jegadeeswari, Dr. D. Gnanambigai, "Data Security Issues in Cloud Environment and Solutions", World Congress on Computing and Communication Technologies 2014.

[7]. NatanAbolafya, Secure Documents Sharing System for Cloud Environments, Master of Science Thesis Stockholm, Sweden 2012.

[8]. Abdullah Al Hasib, AbulAhsan Md. MahmudulHaque, "A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography", Third International Conference on Convergence and Hybrid Information Technology, 2008.

[9]. Cloud Security Alliance, (2009) Security Guidance for Critical Area of Focus in Cloud Computing V2.1. [Online]. Available : https://cloudsecurityalliance.org/csaguide.pdf , accessed on Feb 2012.

VI. ABOUT THE AUTHORS

D.Pharkkavi received her M.Phil Degree from Tiruvalluvar University, Vellore in the year 2013. She has received her M.C.A Degree from Anna University, Chennai in the year 2012. She is pursuing her Ph.D (Full-Time) Degree at Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India. Her areas of interest include Cloud Computing and Mobile Computing.

Dr.D.Maruthanayagam received his Ph.D Degree from Manonmaniam Sundaranar University, Tirunelveli in the year 2014. He received his M.Phil Degree from Bharathidasan University, Trichy in the year 2005. He received his M.C.A Degree from Madras University, Chennai in the year 2000. He is working as HOD Cum Professor, PG and Research Department of Computer Science, Sri Vijay Vidyalaya College of Arts & Science, Dharmapuri, Tamilnadu, India. He has above 15 years of experience in academic field. He has published 4 books, 25 papers in International Journals and 28 papers in National & International Conferences so far. His areas of interest include Computer Networks, Grid Computing, Cloud Computing and Mobile Computing.