Enhancing Security in Cloud Storage using ECC Algorithm

Ravi Gharshi[1], **Suresha**[2]

[1]M. Tech. Scholar, Department of Computer Science and Engineering, Reva Institute of Technology and Management, Bangalore, India
[2]Professor, Department of Computer Science and Engineering, Reva Institute of Technology and Management, Bangalore, India

Abstract: ***Security in cloud computing is an evolving area in today's world. It is subject of concern for Cloud Technology Services. One of the measures which customers can take care of is to encrypt their data before it is stored on the cloud. This work is intended towards providing security service such as confidentiality in the cloud services can use Elliptic Curve Cryptography (ECC) algorithm instead of familiar and generalized RSA for data encryption because of its advantages in terms of smaller key sizes, lower CPU time and less memory usage.***

Keywords: Security, Elliptic Curve Cryptography, RSA, Key Size and Cloud Technology.

1. Introduction

With the invention of cloud, the days of keeping all your documents, photos, music files etc. on your computer's hardware are gradually coming to a close. Today, the cloud storage is fulfilling the need for more storage space to hold all of your digital data. Cloud storage providers operate large data centers, and people who require their data to be hosted buy or lease storage capacity from them. The data center operators, in the background, virtualizes the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers. As shown in Figure 1, cloud storage can be used from smaller computing devices to desktop computers and servers.

Figure 1: Cloud storage

Cloud storage services may be accessed through a web service API or through a Web-based user interface. The cloud storage architectures build a single virtual cloud storage system or cloud of clouds system. The data when stored on cloud has the following threats:

1. When data is distributed, it is stored at multiple locations increasing the risk of unauthorized physical access to the data.
2. The number of people with access to the data who could be compromised (i.e. bribed or coerced) increases dramatically.
3. It increases the number of networks over which the data travels. Instead of just a local area network (LAN) or storage area network (SAN), data stored on a cloud requires a WAN (wide area network) to connect them both.
4. Sharing of storage and networks with many other users/customers it is possible for other customers to access your data.

To secure data, most systems use a combination of techniques, including:

1. Encryption, which means they use a complex algorithm to encode information. To decode the encrypted files, a user needs an encryption key. While it's possible to crack encrypted information, most hackers don't have access to the amount of computer processing power they would need to decrypt information.
2. Authentication processes, which require creating a user name and password.
3. Authorization practices -- the client lists the people who are authorized to access information stored on the cloud system.

Many corporations have multiple levels of authorization. For example, a front-line employee might have very limited access to data stored on a cloud system, while the head of human resources might have extensive access to files. Cloud storage approach poses a potential security threat to your data and moreover, only the password access to storage is not sufficient as the password can be hacked by an intruder. Also the data can be captured en-route to the storage services. The need to access cloud storage on thin clients and mobile devices is becoming an emerging application. But due to smaller processor speed and run time memory; these devices need an algorithm which can be used in such small computing devices. Security of stored data and data in transit may be a concern when storing sensitive data at a cloud storage provider.

1.1 Existing Technology

RSA algorithm is the most widely used public key cryptography algorithm for encryption and decryption by many vendors today. This is the first generation algorithm that was used for providing data security. It can be used to encrypt a message without the need to exchange a secret key separately. The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers. Party A can send an encrypted message to party B without any prior exchange of secret keys. A just uses B's public key to encrypt the message and B decrypts it using the private key, which only he knows. RSA can also be used to sign a message, so A can sign a message using their private key and B can verify it using A's public key.

Encryption of a message, m, involves exponentiation, c = me mod n, which requires a lot of mathematical computations. In RSA cryptosystem, user A (say Alice) picks up two large primes p and q and computes their product,
n = p*q. Now Alice’s public key is a pair of integers {n, e} and the private key is d.

```
Key Generation:
INPUT: Security parameter l.
OUTPUT: RSA public key (n, e) and private key d.
1. Select two primes p and q of the same bit length l/2.
2. Compute $n = p*q$ and $\varphi = (p-1)*(q-1)$.
3. Select arbitrary integer e with $1 < e < \varphi$ and $\gcd(e, \varphi) = 1$.
4. Compute integer d satisfying $1 < d < \varphi$ and $e*d \equiv 1 \pmod{\varphi}$.
5. Return (n, e, d).
```

```
Encryption:
INPUT: RSA public key (n, e), plaintext $m \in [0, n-1]$.
OUTPUT: Ciphertext c.
1. Compute c = me mod n.
2. Return(c).
```

```
Decryption:
INPUT: Public key (n, e), private key d, cipher text c.
OUTPUT: Plaintext m.
1. Compute m = cd mod n.
2. Return (m).
```

1.2 Advantages of ECC over RSA

1. Shorter keys are as strong as long key for RSA.
2. Low on CPU consumption.
3. Low on memory usage.
4. Size of encrypted data is smaller.

In today’s world ECC algorithm is used in case of key exchanges by certificate authority (CA) to share the public key certificates with end users. Elliptic Curve Cryptography is a secure and more efficient encryption algorithm than RSA as it uses smaller key sizes for same level of security as compared to RSA. For e.g. a 256-bit ECC public key provides comparable security to a 3072-bit RSA public key. The aim of this work is providing an insight into the use of ECC algorithm for data encryption before uploading the documents on to the cloud.

Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington) as an alternative mechanism for implementing public-key cryptography. Public-key algorithms create a mechanism for sharing keys among large numbers of participants or entities in a complex information system. Unlike other popular algorithms such as RSA, ECC is based on discrete logarithms that are much more difficult to challenge at equivalent key lengths.

Every participant in the public key cryptography will have a pair of keys, a public key and private key, used for encryption and decryption operations. Public key is distributed to all the participants where as private key is known to a particular participant only.

Section 2 describes the literature survey, section 3 proposes ECC algorithm for encryption and cloud storage, Section 4 specifies the simulation environments, section 5 shows the test results in simulated environment. Section 6 analyzes the test results. The paper concludes with the conclusion of this work and the future enhancements.

2. Literature Survey

Cloud storage is a model of networked enterprise storage where data is stored not only in the user's computer, but in virtualized pools of storage which are generally hosted by third parties. Ravi Shankar Dhakar et al [17] talk about the "Modified RSA Encryption Algorithm (MREA)" where the talk about factorization in RSA cryptosystem, and their implementation compares the existing system and their system with key sizes up to 1024 bit. The authors claim their system to be better than existing system for the brute-force attack. Suli Wang et al [18] talk about the "File encryption and decryption system based on RSA algorithm" where they used RSA for encryption and decryption of files with smaller sizes. Maryam Savari et al [19] in "Comparison of ECC and RSA Algorithm in Multipurpose Smart Card Application" compare the security of RSA 1024-bit key versus ECC 160-bit key sizes. P.R. Vijayalakshmi et al [20] in "Performance Analysis of RSA and ECC in Identity-Based Authenticated New Multiparty Key Agreement Protocol" compare ECC algorithm with 128 bits with that of RSA algorithm with 1024 bits key size. Kamlesh Gupta et al [21] in "ECC over RSA for Asymmetric Encryption: A Review" demonstrated the use ECC for portable devices and applications. Arjun Kumar et al (22) propose a method that allows user to store and access the data securely from the cloud storage in "Secure Storage and Access of Data in Cloud Computing". Xiao Zhang et al (23) talk about the physical security of data in data centers "Ensure Data Security in Cloud Storage". Somani, U et al (24) proposed implementation digital signature with RSA algorithm to enhance data security in cloud storage. Chakraborty, T.K et al (25) proposed a model for data security in cloud. Over last 10 years, a great deal of work has taken place to ensure that ECC meets these goals and is specified in an ever-increasing number of standards. It started with the IEEE P1363 in 1994 (becoming a standard in 2000), and now includes many accredited standards organizations:

i) ISO (in ISO 14888-3: ECDSA and other ECC-based signature schemes)
ii) IEEE (in IEEE 1363-2000 for public-key cryptography)
iii) The American National Standards Institute (in ANSI X9: cryptography for financial-services industry).

NIST also specifies ECC in FIPS 186-2: Federal Information Processing Standards ECDSA and SP 800-56: Special Publication on Key management. While in Europe, BSI in Germany also specifies ECC. Though there are several papers published on the comparison of ECC and RSA in terms of key sizes and security, this paper talks about the reduced key generation time, comparison of encrypted file (cipher text) size against the original plaintext file and its application to the cloud storage. The simulation system provides the key sizes of up to 15360 bits. In this work the used file sizes are up to 40 MB for the simulation i.e. encryption and decryption. This work compares the security of ECC in the key range of 160 - 512 bits and RSA key sizes ranging from 512 - 3072 bits. The simulation experiments compare the ECC and RSA at different levels of key sizes and block sizes.

3. Proposed System: ECC Algorithm

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers. Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications. ECC was developed by Certicom, a mobile e-business security provider, and was recently licensed by Hifn, a manufacturer of integrated circuitry (IC) and network security products. RSA has been developing its own version of ECC. The properties and functions of elliptic curves have been studied in mathematics for 150 years. Their use within cryptography was first proposed in 1985, (separately) by Neal Koblitz from the University of Washington, and Victor Miller at IBM. An elliptic curve is not an ellipse (oval shape), but is represented as a looping line intersecting two axes (lines on a graph used to indicate the position of a point). ECC is based on properties of a particular type of equation created from the mathematical group (a set of values for which operations can be performed on any two members of the group to produce a third member) derived from points where the line intersects the axes. Multiplying a point on the curve by a number will produce another point on the curve, but it is very difficult to find what number was used, even if you know the original point and the result. Equations based on elliptic curves have a characteristic that is very valuable for cryptography purposes: they are relatively easy to perform, and extremely difficult to reverse.

3.1 Computation of Point on the Curve

The security of ECC algorithm depends on its ability to compute a new point on the curve given the product points and encrypt this point as information to be exchanged between the end users.

3.2 Choice of Field

Although RSA public key cryptosystem is a secure asymmetric-key cryptosystem, its security comes with a price of larger key sizes and computational power. Many researchers have looked for an alternative to this system with a smaller key size while maintaining the same level of security. The ECC system is based on the concepts of Elliptic Curves. To analyze the time taken by an algorithm researches have introduced polynomial time algorithms and exponential time algorithms. Algorithms with smaller computation can be evaluated with polynomial time algorithms and complex computations can be evaluated with exponential time algorithms. The equation of an elliptic curve is given as,

y2 = x3 + ax + b

3.3 Key Generation

Key generation is an important part where an algorithm should generate both public key and private key. The sender will be encrypting the message with receiver's public key and the receiver will decrypt its private key. Now, select a number, d within the range of n. Generate the public key using the following equation,

Q = d * P

Where d = the random number selected within the range of (1 to n-1). P is the point on the curve, Q is the public key and d is the private key.

3.4 Encryption

Let 'm' be the message that has to be sent. Consider 'm' has the point 'M' on the curve 'E'. Randomly select 'k' from [1 - (n-1)]. Two cipher texts will be generated let it be C1 and C2.

C1 = k * P
C2 = M + (k * P)

3.5 Decryption

Use the following equation to get back the original message 'm' that was sent.

M = C2 - d * C1

M is the original message that was sent.

4. Simulation

4.1 System Configuration

The System configuration used in the experiments is a Windows 7 operating system with 4 GB RAM and 2.6 GHz processor.

4.2 Assumptions

i) Block Size: The block sizes are assumed to be as follows which are the actual block sizes to be used for RSA. The same block sizes are used for both ECC and RSA which is based on the key size.
Encryption Block Size: ((keySize / 8) - 11)
Decryption Block Size: (keySize / 8)

ii) Key Size: As per The National Institute of Standards and Technology (NIST) Guidelines for Public-Key Cryptography, the ECC and RSA comparable key sizes with equivalent Security Levels are shown in Table 1.

Table 1: Key sizes with equivalent security levels

ECC	*RSA*
160	1024

224	2048
256	3072
384	7680
512	15360

iii) Parameters: To compare the performance characteristics of the RSA and ECC encryption algorithms, the parameters used for simulation are:

- Key Generation Time
- Encryption Time
- Decryption Time

Because of the timing mismatch in each of these 3 parameters simulation performed repetitive tests for each parameter for about 20 times to get the average timings of the parameters.

5. Results

Table 2 provides the key generation, as well as encryption and decryption times for ECC and RSA.

Table 2: Test results

File Size (KB)	*Key Size (bits)*	*ECC Algorithm*				*RSA Algorithm*			
		Key Gen Time (ms)	*Encrypt Time (ms)*	*Decrypt Time (ms)*	*Size of Encrypted File (KB)*	*Key Gen Time (ms)*	*Encrypt Time (ms)*	*Decrypt Time (ms)*	*Size of Encrypted File (KB)*
6534	160	252	2374	1381	6534				
6534	224	262	943	1033	6534				
6534	256	270	1039	964	6534				
6534	384	282	772	755	6534				
6534	512	312	698	687	6534	654	14031	111019	7890
6534	1024					872	18190	300529	7148
6534	2048					1996	29970	998038	6827
6534	3072					16692	41872	210353	6727

6. Analysis of Test Results

6.1 Key Generation Time

In both systems the key generation times were not the same every time, even though the key length is the same and it can sometime a take very long time to generate the keys. Figure 2 shows that for smaller key sizes the key generation time is almost equal in both cases, but as the key size grows RSA takes more amount of time to generate the keys and this time increases exponentially by the key size. Fig2 shows the comparison of the key generation times for RSA and ECC.

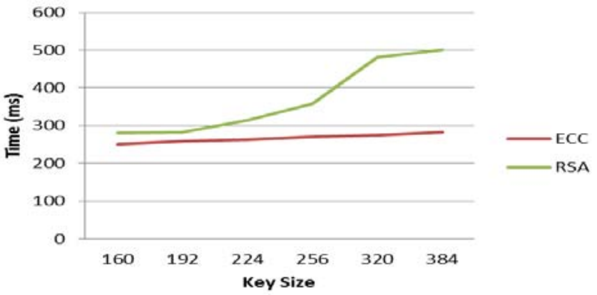

Figure 2: Comparison of Key Generation Time

6.2 Encryption/Decryption Time

Figure 3 shows the Encryption times for ECC and RSA algorithms. Since JAVA implementation of RSA doesn't support key sizes lesser than 512 bits length, simulation had to compare the encryption/decryption times between these two algorithms with different key sizes. Looking at the results, for smaller key sizes ECC provides much faster encryption/decryption as compared to RSA. Since RSA uses higher key sizes the encryption/decryption times grow exponentially with the given key size.

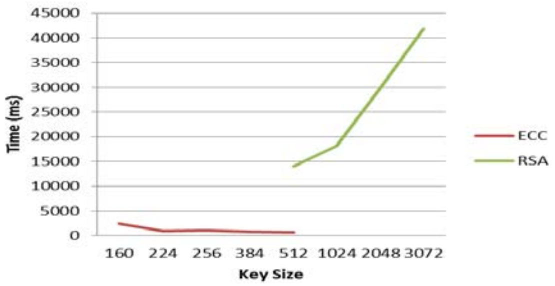

Figure 3: Comparison of Encryption times

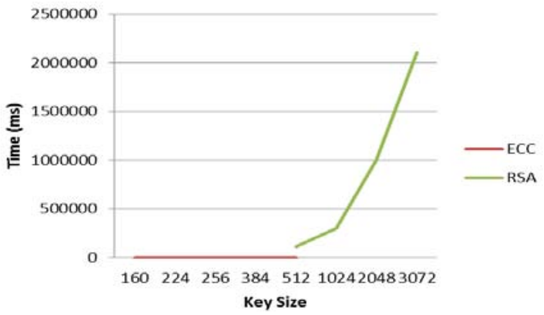

Figure 4: Comparison of Decryption times

Figure 4 shows the decryption times. Time taken by two algorithms for encryption shows that; ECC is much faster than RSA. Based on the input key size ECC encryption time varies linearly whereas in case of RSA it increases exponentially due to the large amount of computation involved and it remains the exponential increase in decryption time too, as shown in the Figure 4. Even though the decryption time is lesser than the encryption time in both algorithms, the decryption time varies exponentially with key size for RSA and it remains linear for ECC as the case with encryption.

7. Conclusion & Future Scope

Elliptic Curve Cryptography provides greater security and more efficient performance than the first generation public key techniques like RSA now in use. As vendors look to upgrade their systems they should seriously consider the elliptic curve alternative for the computational and bandwidth advantages they offer at comparable security. Although ECC's security has not been completely evaluated, it is expected to come into widespread use in various fields in the future. After comparing the RSA and ECC ciphers, the ECC has proved to involve much less overheads compared to RSA. The ECC has many advantages due to its ability to provide the same level of security as RSA yet using shorter keys. However, its disadvantage which may even hide its attractiveness is its lack of maturity, as mathematicians believed that enough research has not yet been done in ECC.

The future of ECC looks brighter than RSA as today's applications (smart cards, pagers, and cellular telephones etc.) cannot afford the overheads introduced by RSA. At least, in today's small computing devices ECC can be used for encryption and decryption as it requires smaller key sizes and has lesser computing complexity as compared to RSA.

Thus, ECC makes it an ideal choice for portable, mobile and low power applications and their integration with cloud services. This work compares the time taken by the two algorithms for key generation and encryption. The importance of this work is to use ECC algorithm in cloud storage which has better security services. This work can be extended to compare ECC with other algorithms used for digital signatures, key exchanges as well as to provide the data integrity.

References

[1] Elliptic curve cryptography, https://en.wikipedia.org/wiki/Elliptic_curve_cryptography

[2] RSA (algorithm), http://en.wikipedia.org/wiki/RSA_(algorithm)

[3] JavaTM Cryptography Extension (JCE), Reference Guide. http://docs.oracle.com/javase/1.5.0/docs/guide/security/jce/JCERefGuide.html

[4] Berta, I.Z., and Z. A. Mann. "Implementing Elliptic Curve Cryptography on PC and Smart Card", Periodica Polytechnica Ser. El. Eng. Vol 46. NO 1-2, PP 47. 2002.

[5] Brown, M., D. L. Hankerson, J. L_opez and A. Menezes. "Software implementation of the NIST Elliptic curves over prime fields". In Progress in Cryptology - CT-RSA, D. Naccache, Ed, vol. 2020 of Lecture Notes in Computer Science, pp. 250-265. 2001.

[6] Neal Koblitz, Alfred J. Menezes, "A Survey of Public-Key Cryptosystems". Advanced Computing & Communication Technologies (ACCT), Second International Conference, 2012.

[7] Certicom Corp. "An elliptic curve cryptography (ECC) primer". White paper, Certicom. 2004.

[8] Rabah, K. "Implementation of Elliptic curve Diffie-Hellman and EC Encryption schemes". Information technology journal, 01/2005.

[9] Rabah, K. "Implementing Secure RSA Cryptosystem Using Your Own Cryptographic JCE Provider". Journal of Applied Science, vol. 6, Issue 3, p.482-510. 2006.

[10] Robshaw, M. J. B. and Y. L. Yin. "Elliptic Curve Cryptosystems". 1997 http://www.rsasecurity.com/rsalabs/ecc/ellipticcurve.html

[11] Stallings, W. "Cryptography and Network Security: Principles and Practice, 3rd edition", Prentice Hall, New Jersey, 2003.

[12] Trappe. W and L. C. Washington "Introduction to Cryptography with Coding Theory", Prentice Hall, New Jersey, 2002.

[13] Weil, N. (). "U.S. govt.'s encryption standard cracked in record time '. Network World. 1998, http://www.networkworld.com/news0720des.html

[14] Amara, M.; Lab. LAGA, Univ. Paris-8, St. Denis, France; Siad, A. "Elliptic Curve Cryptography and its applications".

[15] Fiskiran, A.M.; Dept. of Electronics Engg. Princeton Univ., NJ, USA, Lee, R.B. "Workload characterization of elliptic curve cryptography and other network security algorithms for constrained environments".

[16] S. Maria Celestin Vigila, K. Muneeswaran. "Implementation of Text based Cryptosystem using Elliptic Curve Cryptography".

[17] Ravi Shankar Dhakar, Amit Kumar Gupta, "Modified RSA Encryption Algorithm (MREA)". Advanced Computing & Communication Technologies (ACCT), Second International Conference, 2012.

[18] Suli Wang, Ganlai Liu, "File encryption and decryption system based on RSA algorithm". Computational and Information Sciences (ICCIS), International Conference, 2011.

[19] Maryam Savari, Mohammad Montazerolzohour and Yeoh Eng Thiam, "Comparison of ECC and RSA Algorithm in Multipurpose Smart Card Application". Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), International Conference, 2012.

[20] P.R. Vijayalakshmi, K. Bommanna Raja, "Performance Analysis of RSA and ECC in Identity-Based Authenticated New Multiparty Key Agreement Protocol". Computing, Communication and Applications (ICCCA), International Conference, 2012.

[21] Kamlesh Gupta, Sanjay Silakari, "ECC over RSA for Asymmetric Encryption: A Review", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011.

[22] Arjun Kumar, Byung Gook Lee, HoonJae Lee "Secure Storage and Access of Data in Cloud Computing". ICT Convergence (ICTC), International Conference, 2012

[23] Xiao Zhang, Hong-tao Du, Jian-quan Chen, Yi Lin, Lei-jie Zeng, "Ensure Data Security in Cloud Storage". Network Computing and Information Security (NCIS), International Conference, 2011.

[24] Somani, U, Lakhani, K, Mundra, M, "Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing". Parallel Distributed and Grid Computing (PDGC), 1st International Conference, 2010.

[25] Chakraborty, T.K.; Dhami, A.; Bansal, P.; Singh, T. "Enhanced public auditability & secure data storage in cloud computing". 3rd IEEE International Advance Computing Conference (IACC), 2013.

Author Profile

Ravi Gharshi received Bachelor of Computer Science Engineering from Visvesvaraya Technological University, Belgaum, Karnataka. He is now pursuing Master of Technology in Computer Network Engineering. His research interests are security in cloud storage and data security

Suresha completed B.E in Electronics and Communication Engineering from National Institute of Engineering, Mysore and did his Master's in Computer Science and Engineering at Basveswara Engineering College, Bagalkot and got his Ph. D. in Computer and information Sciences from VTU Belgaum. He has 14 years of Industrial Experience, 16 years of Teaching Experience and is actively involved in research from past 5 years. At present he is working as a professor and PG Coordinator for part time M. Tech programs in the Department of Computer Science and Engineering at Reva Institute of Technology and Management, Bangalore. He has many publications in referred International Journals and Conferences. His areas of interest are Network Security, Ad-hoc

Networks, Wireless Communications and Securities in Wireless Sensor Networks. He is more passionate about research and currently guiding three research students. He is a life member of ISTE and CSI.