Elliptic Curve Cryptography for Securing Encoded Media Storage on Cloud

Ashok Patel
Computer Engineering,
K. J. Somaiya College of Engineering, Mumbai, India

Sheetal Thakkar
Computer Engineering,
K. J. Somaiya College of Engineering, Mumbai, India

Saloni Parekh
Computer Engineering,
K. J. Somaiya College of Engineering, Mumbai, India

ABSTRACT
With the increasing rate of adoption of cloud services for storage of personal files, there is a need of protecting the data, especially those in media format to avoid intrusion of external attackers as well as the cloud storage providers, preventing them from using it otherwise. This paper briefly describes the concept of base64 encoding and its use along with Elliptic Curve Cryptography (ECC) as a crypto-system, to store media files on cloud in an encrypted digital form. The paper also includes a comparative study of ECC and RSA.

General Terms
Cryptology, Computer and Information Security, Cloud

Keywords
Cyber Security, Cryptography, ECC, RSA, Encryption, Encoding, Base64

1. INTRODUCTION
With an exponential increase in the growth of internet, the world is moving towards the usage of cloud even for basic computing. Cloud is more prominently used for storage purposes as cloud provides wide storage space, reliability, availability along with cost reduction. With the advantages of using of cloud, comes a plethora of issues and data becomes more and more vulnerable to security threats. Hence, security becomes a major concern when it comes to utility of cloud services, especially cloud storage. Most of cloud service providers do not provide sufficient security in terms of storing the data in an encrypted manner. The data on cloud is exposed to even more vulnerability when the cloud services are synced with smart phones, tablets and laptops. If these devices are lost or stolen – this astoundingly happens quite often – the data is potentially exposed to malpractices and other cyber attacks.[1] Besides, when data is encrypted in cloud, the key used for encryption lies somewhere in the same cloud and even the algorithms used for encryption are not sufficiently strong. This is because encryption requires processing power and may also increase the cost of storage. [2] While in some cases, this data is analyzed and used by the cloud service providers to understand their users and provide better experience for further use. Some clouds only encrypt certain information like the username and the password. While looking at all this, security for media may get compromised totally. Even media files may contain sensitive data that needs to be protected from, and including the provider itself. While most of the files, are encrypted using top-notch security measures, including algorithms such as AES(Advanced Encryption Standard), DH(Diffie-Hellman) with signatures generated by implementing equally strong RSA(Rivest-Shamir-Adleman) algorithm. The mentioned strategies are strong enough to keep our cloud storage safe from external attacks. However, there is always a threat to individual files that get stored over the cloud, as cloud services don't prefer encrypting each file individually. This is done for many reasons, one being reduced processing and response times, and the other being the data made available to the provider, on which the it thrives. This preference comes at the cost of an exposed loophole that makes all our individual files over the cloud vulnerable. Thus, a solution to this very issue has been proposed by making use of Elliptic Curve Cryptography, a modern cryptography solution based on the ECDLP - Elliptic Curve Discrete Logarithm Problem. Despite almost three decades of research, mathematicians still haven't found an algorithm to solve this problem that improves upon the naive approach [3].

2. LITERATURE SURVEY
As a part of the case study, a survey was conducted among individuals that gave us an insight into the general notion of cyber security pertaining to cloud storage services and cryptography. About 78% of the sample population is aware of the concept of cryptography. It also reveals that more than 80% of people feel that the data stored and accessed by their device(s) is not secure. 95.7% individuals believe that handheld devices like mobile phones, tablets and PDAs are gaining prominence, but only 37% are satisfied with the security the device provides. 82.2% of people say that there's a need to protect data on cloud but 89.1% of individuals have not heard of Elliptic Curve Cryptography. Although, a majority of them do feel that the security will get strong if we use it with some algebraic structure or function. Prevailing implementations and products that provide a solution to cloud storage's viable security are either too costly, or use trivial crypto systems that are easily prone to backdoor attacks. Thus, we conclude that there is a need of awareness on employing some security measures for cloud storage among the masses, as well as increasing the use of newer standards such as ECC for efficient cryptosystems. The results inspired the proposed implementation of ECC for improving cloud security.

3. ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM
Extensive studies have been done on Elliptic curves and their properties for over 150 years. The difficulty in backtracking a system based on this ECDLP can be understood with the help of an elliptic curve. An elliptic curve is the set of all points in 2D region that satisfy an equation in two variables with degree two in one of the variables and three in the other. Thus, a general elliptic curve can be represented by the following equation:

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

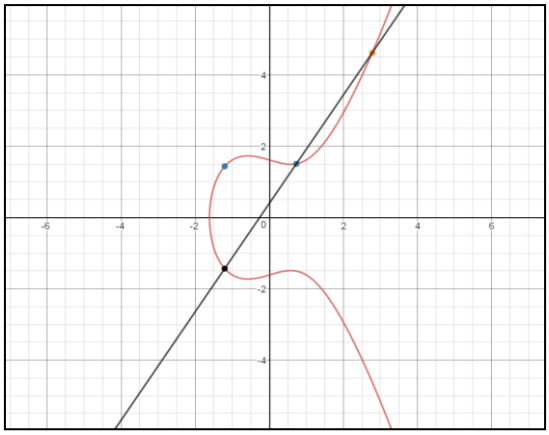

Fig 1: Sample Elliptic Curve

Substituting a = -1 and b = 2.6, the curve obtained is as shown in figure 1. Each choice of the numbers a and b yields a different elliptic curve, but should satisfy the condition that $4a^2 + 27b^2! = 0$. One of the properties of any elliptic curve is horizontal symmetry, i.e. any point on the curve can be mirrored and obtained on the curve itself with respect to x-axis. Another important property of such a curve is that every non-vertical line that passes through the curve intersects the curve in no more than 3 points. Thus, when a point is 'dotted' with itself, we get a second point on the curve, which when again dotted with the first point, gives the final point that lies on the straight line and intersects the curve. Continuing the process, and dotting every subsequent point with the first point, we can have 'n' moves to arrive at a particular final point on the curve. The beauty of this property lies in the fact that, even if we know the first and the final point, one cannot exactly determine the number of times dotting has taken place to reach that final point, which is equal to the number 'n'. For sufficiently large value of 'n', elliptic curves give rise to a very good trapdoor function that can be used for encryption strategies.

An actual implementation of elliptic curves in cryptography for security purposes is made possible by restricting the value co-ordinates of points to a whole number. An elliptic curve cryptosystem can be defined by picking a prime number as a maximum, a curve equation, and a public point on the curve. A private key is a number priv, and a public key is the public point dotted with itself priv times. Computing the private key from the public key in this kind of cryptosystem is called the elliptic curve discrete logarithm function. This turns out to be the trapdoor function used for Elliptic Curve Cryptography [3].

Thus, unlike other cryptosystems that rely on factoring of substantially large numbers for a firm trapdoor function, ECC is capable of providing equivalent or better security for the same or less size of numbers. This makes elliptic curve cryptosystems harder to break than its counterparts - RSA and Diffie Hellman.

4. MEDIA ENCRYPTION USING ECC

As a performance measure, to compare the efficiency of ECC and RSA algorithms, signature generation was taken into consideration by both the solutions. A signature is a string of characters or numbers that is derived from the original message and appended to the encrypted message before transmitting. This signature is used by the receiver to verify the authenticity of the message that has been received. The implementation for ECC Signature generation was made possible by using the java.security and java.math.BigInteger packages, that provide classes and interfaces for key specifications/algorithm parameter specifications [4] and manipulating numbers that don't fit in any of the primitive data types [5] respectively. Similarly, RSA Signature generation was made possible with the help of Java's inbuilt java.security.Signature package. The ECC implementation makes use of KeyPairGenerator class from java.security package to generate sect163k1 key pair on every run. The text (message) for which the signature is to be generated was to be so chosen that it replicates real life usage of these algorithms.

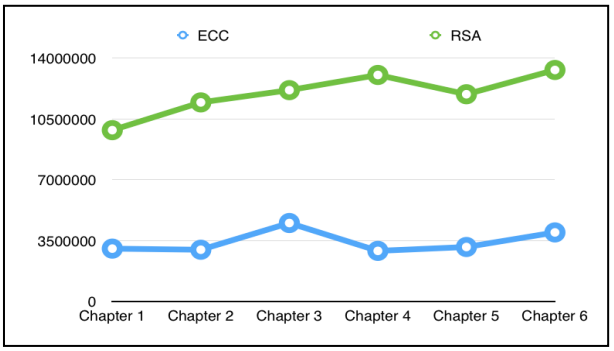

Fig 2: Signature generation times of ECC and RSA

Hence, for every run, a chapter from the book Pride and Prejudice by Jane Austen available as text format [6]. Each chapter consists of varying number of thousands of characters in the range 3000 to 11000 and a .txt file size ranging between 2KB to 11KB, and total time taken by each implementation to generate a hashed signature was noted down. As ECC has better efficiency in signing messages, the difference in times noted down was quite substantial, an average of 28.68% increase in speed by implementation of ECC for the same text file. The difference can be studied from the graph shown in figures 2 and 3.

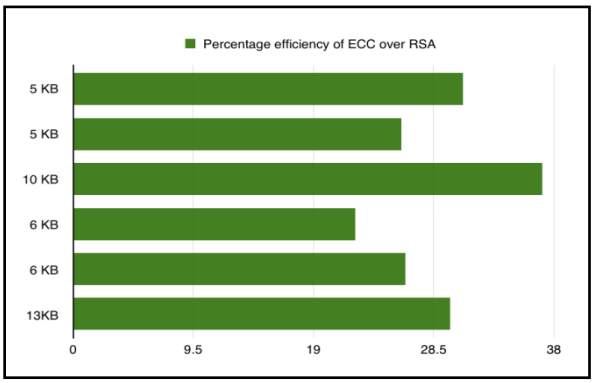

Fig 3: Percentage efficiency of ECC over RSA

As per the data from the literature survey, 75% of the sample population that took the survey believes that prevalent cloud storage services are secure and have no access to files stored over cloud. This is in stark contrast with the actuality of services provided by these cloud storage options, especially those that survive on user data. For one of such services, data in the content the user uploads is very crucial as it is used to target ads to the user anywhere on the service. It may not use the data in one's files directly for its own use, but can use it to get a better understanding of the user groups it is servicing [7]. This can serve as backdoor, leaving personal but potentially critical user information getting revealed out of consent. There have been many instances of this happening in

the past, and can happen even more number of times given the rate of adoption of cloud storage for personal usage, especially media files such as images, videos as well as audio. A firm crypto system such as ECC can be adopted to indirectly protect these file types, by first converting them into text format by encoding, and then deploying the algorithm on the text files so obtained before uploading over cloud. A system, in which only the owner can have access to its files without the service provider or a hacker getting access to the individual files, even after the storage getting hampered can thus be built to ensure security. The question that may arise at this point is - What are the setbacks of converting media files before uploading? To analyze probable drawbacks of this process, 6 sample satellite images [8] with file sizes in the range 40 to 100 MB were encoded into text format using base64 encoding. This conversion results in an increased file size, with percentage increase in each sample image as seen in figure 4. These text formatted files can now be encrypted using ECC and thus protecting confidential data stored over commercial cloud service.

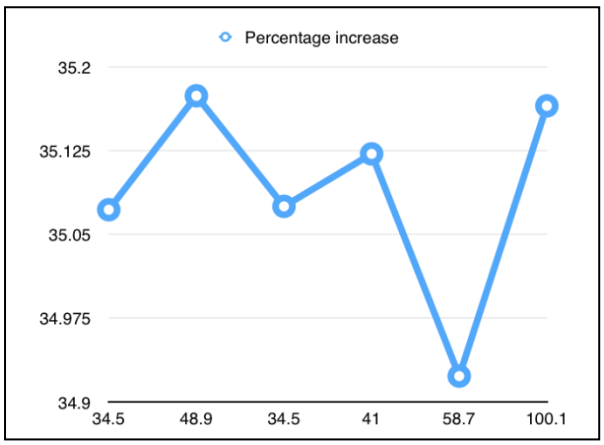

Fig 4: Percentage increase in file size (megabytes)

5. COMPARISON OF ECC AND RSA

With increasing vulnerability of data to potential cyber-attacks, modern cryptographic mechanisms make comparatively extensive use of asymmetric algorithms i.e. public key encryption. Asymmetric cryptographic algorithms use a pair of public key and private key for encryption-decryption. RSA gains its security by factoring the product of two large prime numbers whereas ECC is based on complex algebraic structures of ECDLP. It is comparatively difficult as well as time consuming to factor a large integer number composed of two or more large prime factors. Security of a cryptographic algorithm can be evaluated when its logical complexity is combined with its key size. In today's technical era, at least 128 bit of security is essential which can be achieved by 128-bit AES keys, 256-ECC key, and 3072-bit RSA keys. This means that ECC can provide same level of security as that of RSA with a smaller key size. Also, when the key size increases exponentially it becomes difficult to implement an algorithm. Hence, for RSA to gain deep level of security the implementation may become challenging especially in embedded systems [9]. Also, in ECC the time required to generate key grows linearly with the key size while in RSA it grows exponentially [10]. Considering the above factors, it can be stated that ECC has lesser computational time when compared to RSA. In ECC, it is assumed that finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is infeasible [11]. ECC involves complex but smaller computations. Basically, ECC leads to faster computation and lesser memory consumption. As a powerful computational problem refers to strong cryptography, it is difficult to break ECC when compared RSA and Diffie-Hellman. The implementation of ECC is considered to be quite complex and tricky. ECC is expected to be more vulnerable to attacks based on Shor's algorithm, based on quantum computing [11].

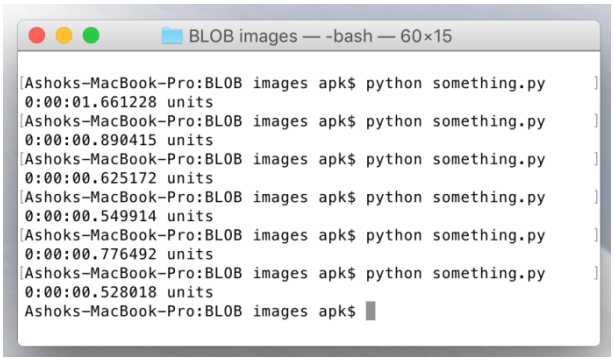

Fig 5: Calculating time elapsed for encoding

6. CONCLUSION

As inferred from the results, the protection of files stored over the cloud can be improvised using better asymmetric cryptosystems such as ECC, that provide equal or better security using smaller keys. The integrity of a user's data needs to be maintained and protected from external attacks. Thus, proposed combination of base64encoding of media files and further encryption using ECC standards can serve the purpose without compromising on expected security levels in prevalent cloud storage services, which is not implemented at the moment for the sake of data collection and accessibility.

7. REFERENCES

[1] All about Google drive encryption – Sookasa https://www.sookasa.com/resources/google-drive-encryption/

[2] Cloud storage encryption – TechTarget http://searchcloudstorage.techtarget.com/definition/cloud-storage-encryption

[3] Nick Sullivan, A primer on Elliptic Curve Cryptography http://arstechnica.com/security/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/2/

[4] https://docs.oracle.com/javase/7/docs/api/java/security/spec/package-summary.html

[5] http://www.geeksforgeeks.org/biginteger-class-in-java/

[6] http://www.kellynch.com/e-texts/PrideandPrejudice.php

[7] The Google Drive FAQ – CNET https://www.cnet.com/news/the-google-drive-faq/

[8] http://effigis.com/solutions/satellite-images/satellite-image-samples/

[9] Kerry Maletsky, Atmel, 'RSA vs ECC Comparison for Embedded Systems.' Available - http://www.atmel.com/images/atmel-8951-cryptoauth-rsa-ecc-comparison-embedded-systems-whitepaper.pdf

[10] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.7139&rep=rep1&type=pdf

[11] Elliptic Curve Cryptography – Wikipedia https://en.wikipedia.org/wiki/Elliptic_curve_cryptography