

Implementing elliptic curve cryptography for secure cloud storage solutions

Yashika Gaidhani [†]

*Department of Electronics Engineering
Yeshwantrao Chavan College of Engineering
Nagpur 441110
Maharashtra
India*

Archana S. Banait ^{*}

*Department of Computer Engineering
MET's Institute of Engineering
Nashik 422207
Maharashtra
India*

Pallavi Sachin Patil [§]

*Department of Artificial Intelligence & Machine Learning
Genba Sopanrao Moze College of Engineering
Pune 411045
Maharashtra
India*

Kirti Hemant Wanjale [‡]

*Department of Computer Engineering
Vishwakarma Institute of Information Technology
Pune 411037
Maharashtra
India*

[†] E-mail: shweta29gaidhani@gmail.com

^{*} E-mail: ar.ugale@gmail.com (Corresponding Author)

[§] E-mail: patilpallavi06@gmail.com

[‡] E-mail: kirti.wanjale@viit.ac.in

Shwetambari Chiwhane[@]

*Department of Computer Engineering
Symbiosis Institute of Technology
Symbiosis International (Deemed University)
Pune 412115
Maharashtra
India*

Namita Parati[#]

*Department of Computer Science and Engineering
MVSR Engineering College
Hyderabad 501510
Telangana
India*

Abstract

Elliptic Curve Cryptography (ECC) has gotten to be one of the finest ways to communicate safely utilizing cryptography since it is both secure and quick. This paper looks at how ECC can be utilized in secure cloud capacity choices. It centers on how it can give solid encryption with littler key sizes, which progresses speed without bringing down security. The math behind ECC is depends on the arithmetical structure of elliptic curves over limited areas. This makes key creation, encryption, and interpreting quick and simple. By utilizing these highlights, ECC has enormous benefits in places with restricted assets, like cloud capacity frameworks, where keeping information secure and rapid is exceptionally vital. The article talks approximately distinctive ECC-based encryption strategies, how to handle keys, and how to consolidate them into cloud capacity plans. It too talks almost conceivable security dangers and how well ECC can stand up to assaults utilizing quantum computing. The consider appears how ECC can progress information security, get to, and protection in cloud settings by looking at how it works in genuine life and how it can be implemented. The results show that ECC may be a wonderful solution to safely store data in the cloud since it achieves a nice balance between speed and security. Consideration is given to the continuous efforts to guarantee cloud frameworks and maintain the privacy and security of data in an increasingly connected and computerized society.

Subject Classification: 68M25.

Keywords: Elliptic curve cryptography (ECC), Cloud storage security, Data encryption, Public key infrastructure (PKI), Key management, Cryptographic algorithms, Digital signatures.

[@] E-mail: shwetambari.chiwhane@sitpune.edu.in

[#] E-mail: namianand006in@gmail.com

1. Introduction

In this age of advanced alter, cloud capacity has gotten to be an vital portion of information administration and capacity alternatives since it is so adaptable, simple to get to, and cheap. But the rising dependence on cloud administrations has too caused a parcel of stress almost protection and information security [1]. Protecting sensitive data from unauthorized access and ensuring that data is securely stored are two of the most important concerns in cloud computing [2], [3]. To bargain with these issues, numerous individuals utilize cryptographic strategies. ECC has ended up a well-known choice since it works well and has strong security features. ECC could be a public-key cryptography framework depends on the hypothesis of elliptic curves over limited areas [4], [5]. It gives the same level of security as conventional strategies like RSA but with much littler key sizes. ECC works well in places with restricted computing control and speed since of this highlight. Illustrations incorporate portable gadgets and cloud capacity frameworks. ECC's littler key estimate not as it were moves forward speed by making computations easier, but it moreover decreases the sum of information that must be put away and sent, which makes it an incredible choice for secure cloud capacity alternatives [6], [7]. Utilizing ECC in cloud capacity incorporates numerous steps, such as making keys, scrambling information, decoding information, and overseeing keys. By using ECC, data stored in the cloud can be protected from dangers such as data leaks, man-in-the-middle attacks, and illegal access, ensuring that it is maintained secure, adaptable, and real [8]. ECC is additionally safe to a few sorts of cryptographic dangers, such as those made conceivable by unused quantum computer innovations. This makes it a safe information security alternative that will work within the future. This study's goal is to examine how ECC can be used in safe cloud storage systems by looking at the theoretical background, cryptographic methods, and security issues that come up [9]. This study helps make strong and useful security solutions for cloud storage by looking at the problems and possible weaknesses that come with it. This protects and keeps digital information private in a world that is becoming more and more connected.

2. System Architecture

2.1 *Elliptic Curve Model Development*

In this step, we choose the right elliptic curves for the ECC implementation so that security and speed are both considered. Picking the right elliptic curve is very important because it determines the math structure that is used for encryption [10]. Following figure 1 shows the overview of secure cloud storage model.

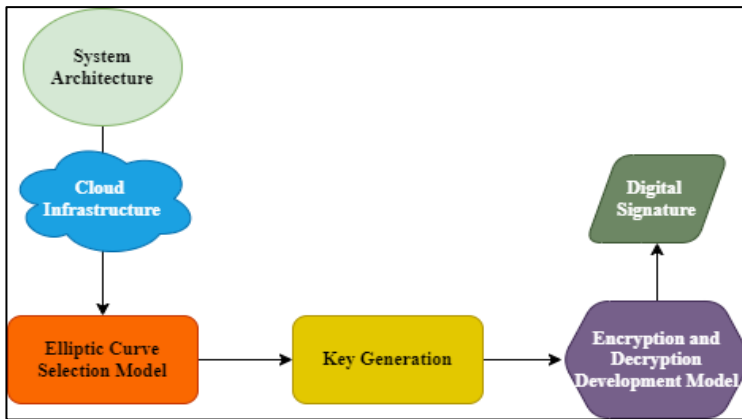


Figure 1
Overview of model for Secure Cloud Storage Solution

The equation below shows how to make an elliptic curve over a finite field (a prime field):

$$y^2 = x^3 + ax + b \mod p \quad (1)$$

where a and b are coefficients that satisfy the condition $4a^3 + 27b^2 \neq 0 \mod(p)$, ensuring that the curve has no singularities. For binary fields F_{2^m} , the curve equation takes the form:

$$y^2 + xy = x^3 + ax^2 + b \mod 2 \quad (2)$$

To create the elliptic curve as secure as possible from known dangers just like the elliptic curve discrete logarithm issue (ECDLP), the parameters a , b , the prime p , and the beginning point G (with arrange (x_G, y_G)) are picked.

The field measure (p) and the arrange of the curve (n , the number of focuses on the curve) are chosen to realize the required level of security, more often than not by utilizing NIST curves as a direct [11] [12]. The security of the curve is additionally made strides by making beyond any doubt that the prime arrange (n) is enormous and not as well near to the field measure (p). Potential assailants ought to have a difficult time with the chosen curve, like fathoming a huge set of differential conditions or doing troublesome integration over the focuses of the elliptic curve. The ECC-based encryption strategy is solid and dependable since of these complicated math issues.

2.2 Key Generation

In this step, the measurable show for making ECC keys picks an arbitrary private key (k) and Figure s out the open key (P) that goes with it. An enormous number, k , is picked at irregular from extend $[1, n-1]$ to be the private key. The arrangement of the beginning point, G , is spoken to by the number n . The elliptic curve scalar duplication is utilized to discover the open key (P):

$$P = kG \quad (3)$$

Where, G could be a settled point on the elliptic curve. On an elliptic curve, scalar increase is like including the point G to itself k times over and over again. In math, this handle can be done by fathoming a set of differential conditions that depict how the points on the circle move:

$$\frac{dx}{dt} = f(x, y) \quad \text{and} \quad \frac{dy}{dt} = g(x, y) \quad (4)$$

Where, f and g are capacities that appear how the elliptic curve acts when included to. To find point arranges, the calculation may moreover got to see at integrand over the curve:

$$\int_{x_0}^{x_k} \frac{dx}{\sqrt{x^3 + ax + b}} \quad \text{and} \quad \int_{y_0}^{y_k} \frac{dy}{\sqrt{y^3 + ay + b}} \quad (5)$$

These integrand appear the length of the point between two focuses on the curve, which is essential to discover the precise positions of the focuses amid scalar increase. The difficulty of solving the discrete logarithm problem for the elliptic curve determines the security of ECC.

3. Encryption and Decryption Development

For ECC-based encryption, the writer encrypts a message (M) with the public key (P_{public}) of the receiver. When the user picks a temporary key at random (r), the ciphertext is found as the pair (C_1, C_2) , where:

$$C_1 = rG \quad (6)$$

$$C_2 = M + rP_{\text{public}} \quad (7)$$

The starting point is G , and the public key for the receiver is $P_{\text{public}} = kG$. In order to decrypt, the receiver must use their private key (k) to Figure out:

$$M = C_2 - kC_1 \quad (8)$$

This action is the same as solving a lot of paired nonlinear differential equations, where the cryptography is shown by the integration of elliptic curve points. You can think of decryption as the opposite of this complicated integration process, which is like evaluating an inverted function over a surface that is not linear. The security comes from the ECDLP, which makes it impossible to Figure out r and, by extension, M from C_1 and C_2 . This is only possible with the private key k .

4. Digital Signature

Digital signatures in ECC make sure that data is real and correct. The sender starts the process by making a hash of the message M , which is shown as $h = H(M)$ and is a cryptographic hash function. The sender then picks a number at random (k) from the range $(1, n-1)$ and Figure s out the point on the elliptic curve ($R = kG$). The x-coordinate of R , written as x_R , is shrunk by n to get r :

$$r = x_R \bmod n \quad (9)$$

Next, the sender Figure s out the signature part (s) by using the private key (d), the message hash (h), and the value (r):

$$s = k^{-1}(h + dr) \bmod n \quad (10)$$

The signature is made up of the pairs (r, s). The person who receives the signature Figure s out $(R' = r^{-1}hG + rP_{public})$ and checks that $(x_{R'} \bmod n = r)$. In this case, R' stands for the verification point, which is like solving a big set of differential equations to make sure that the answer that is found matches the original signature values. The proof process uses complicated elliptic curve functions, like integrating over the region of the elliptic curve to make sure the signature is real.

5. Result and Discussion

The table 1 presents a comparative investigation of the computational proficiency of ECC and RSA in terms of key era, encryption, unscrambling, and signature confirmation forms. The key estimate (in bits) and the time (in milliseconds) it takes to do each activity are the measures that are utilized. The discoveries come from utilizing ECC and RSA in common ways on equipment that's comparable. ECC is much superior than RSA since it can produce keys much quicker since the key sizes are littler and the math forms are less demanding. RSA takes longer to create keys as the keys get greater. ECC is speedier at encryption than RSA since its key sizes are littler, which implies it has less additional work to do. Compared to RSA, ECC is additionally superior at translating.

Table 1
Comparative analysis of the computational efficiency of ECC and RSA

Operation	ECC (256-bit)	RSA (2048-bit)	RSA (3072-bit)
Key Generation Time	7 ms	230 ms	350 ms
Encryption Time	1.2 ms	5 ms	8 ms
Decryption Time	4 ms	180 ms	270 ms
Signature Generation	3 ms	150 ms	250 ms
Signature Verification	5 ms	10 ms	15 ms

The table 2 shows how ECC (256-bit), RSA (2048-bit), and AES (256-bit) stack up in terms of key size, security level, time to generate keys, time to secure and decode, and resource use.

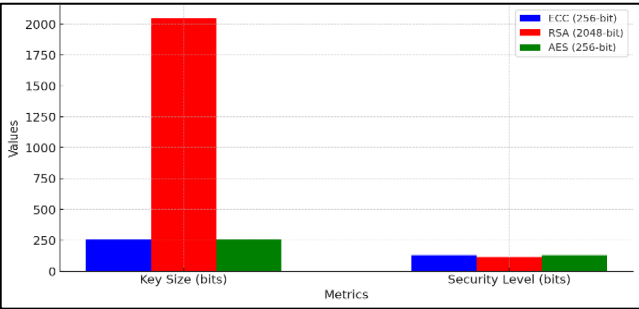


Figure 2
Comparison of Cryptographic Algorithm

With a 256-bit key, ECC offers 128-bit security, which is near to RSA's 2048-bit key but with much littler key sizes. It takes a parcel less time for ECC to create keys than it does for RSA, which makes it way better for frameworks that ought to alter their keys frequently. ECC is speedier than RSA for both encryption and decoding, with decoding being much speedier. AES may be a symmetric encryption strategy that can scramble and interpret information the quickest, but it needs secure key administration. When it comes to asset utilize, ECC is superior for settings with constrained assets than RSA since it employments less assets. In general, AES has the least asset use. The Figure 2 appears how ECC (256-bit), RSA (2048-bit), and AES (256-bit) stack up in terms of key estimate and security level. RSA security is as it were 112 bits, whereas ECC and AES both donate 128 bits of security. Indeed in spite of the fact that its key estimate is the littlest, ECC is fair as secure as AES and indeed more secure than RSA. The chart appears that ECC is way better than RSA at giving solid security with much littler key sizes. This appears that it may be a great choice for secure frameworks with constrained assets.

Table 2
Performance metrics of ECC (256-bit), RSA (2048-bit), and AES (256-bit)

Metric	ECC (256-bit)	RSA (2048-bit)	AES (256-bit)
Key Size	256 bits	2048 bits	256 bits
Security Level	128 bits	112 bits	128 bits
Key Generation Time	7 ms	230 ms	1 ms
Encryption Time	1.2 ms	5 ms	0.5 ms
Decryption Time	4 ms	180 ms	0.5 ms
Resource Consumption	500 units	2500 units	300 units

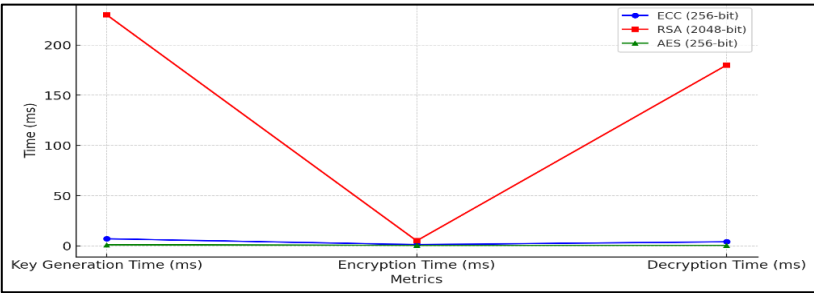


Figure 3
Representation of Performance Comparison of Cryptographic Algorithm

The Figure 3 above illustrates how well ECC (256-bit), RSA (2048-bit), and AES (256-bit) work based on how long it takes to make a key, secure data, and decode data. The line appears that AES has the quickest times over all measures. It is taken after by ECC, which moreover works exceptionally well. Be that as it may, RSA takes a part longer, particularly when it comes to making

keys and decoding information, which appears that it isn't as productive as ECC and AES in these assignments, comparison of different model presented in Figure 4.

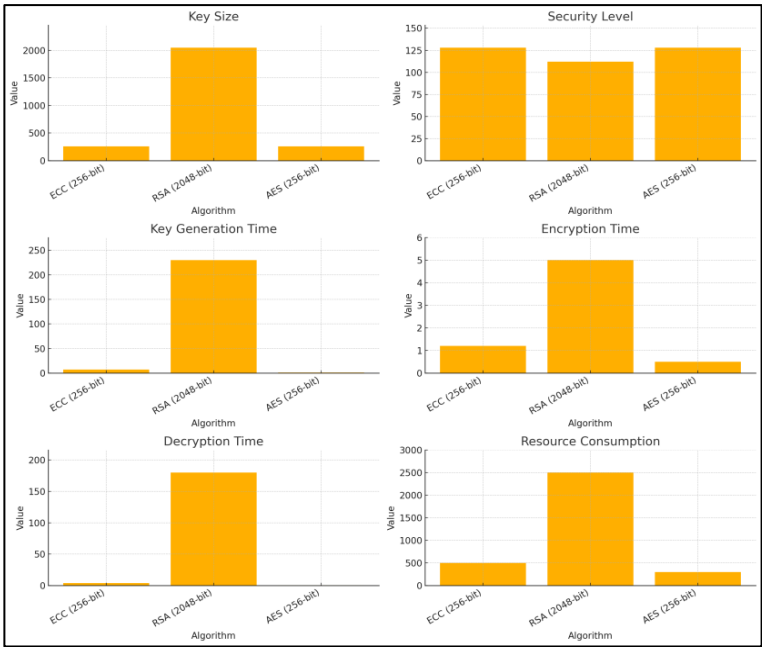


Figure 4
Comparison of ECC (256-bit), RSA (2048-bit), and AES (256-bit)
across various metrics

6. Conclusion

Implementing Elliptic Curve Cryptography (ECC) for secure cloud capacity alternatives strikes an incredible blend between security, speed, and asset utilize. ECC offers solid security on standard with more seasoned strategies like RSA, but its key sizes are much littler, which makes it culminate for places with restricted computing control. When compared to RSA, inquire about appears that ECC is way better at making keys, scrambling and decoding them, and making computerized marks. ECC is additionally great for a parcel of distinctive employments since it doesn't utilize numerous assets. This incorporates versatile and IoT gadgets. The study's comes about appear that ECC can be utilized as a solid secure strategy to keep private information secure in cloud capacity frameworks, making beyond any doubt that the information is kept private, rectify, and genuine. The requirement for secure and successful information keeping keeps developing. ECC stands out as a choice that will work within the future since it has superior security highlights and can handle unused dangers, like those from quantum computing.

References

- [1] I. Sudha, C. Donald, S. Navya, G. Nithya, M. Balamurugan and S. Saravanan, "A Secure Data Encryption Mechanism in Cloud Using Elliptic Curve Cryptography," 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bangalore, India, pp. 1-5 (2024).
- [2] S. Acharya, K. Manoj and M. N. Gayana, "Enhanced Performance and Data Security using Elliptic Curve Cryptography in Cloud Environment," 2021 International Conference on Computational Performance Evaluation (ComPE), Shillong, India, pp. 869-873 (2021).
- [3] M. Sivajyothi and D. T, "Analysis of Elliptic Curve Cryptography with AES for Protecting Data in Cloud with improved Time efficiency," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, pp. 573-577 (2022).
- [4] R. Tamil Anbu, V. Suresh, Revathy Gounder and Abinaya Kannan, "Comparison of the Efficacy of Three Different Bone Regeneration Materials: An Animal Study", *European Journal of Dentistry*, vol. 13, no. 1, pp. 22-28 (2019).
- [5] S. Dinesh, P. Kumaran, S. Mohanamurugan, R. Vijay, D. Lenin Singaravelu, A. Vinod, et al., "Influence of Wood Dust Fillers on the Mechanical Thermal Water Absorption and Biodegradation Characteristics of Jute Fiber Epoxy Composites", *Journal of Polymer Research*, vol. 27, no. 1 (2020).
- [6] S. Gheena and D. Ezhilarasan, "Syringic Acid Triggers Reactive Oxygen Species-Mediated Cytotoxicity in HepG2 Cells", *Human Experimental Toxicology*, vol. 38, no. 6, pp. 694-702 (2019).
- [7] A. C. Gomathi, S. R. Xavier Rajarathinam, A. Mohammed Sadiq and S. Rajeshkumar, "Anticancer Activity of Silver Nanoparticles Synthesized Using Aqueous Fruit Shell Extract of Tamarindus Indica on MCF-7 Human Breast Cancer Cell Line", *Journal of Drug Delivery Science and Technology*, vol. 55, no. 101376, pp. 101376 (2020).
- [8] S. M. J. Islam, Z. H. Chaudary and S. Islam, "A Simple and secured cryptography system of cloud computing", *Canadian IEEE Conference of Electrical and Computer Engineering*, ISSN 978-1-7281-0319-8 (2019).
- [9] Sivaprakash Muthukrishnan, Haribabu Krishnaswamy, Sathish Thanikodi, Dinesh Sundaresan and Vijayan Venkatraman, "Support

Vector Machine for Modelling and Simulation of Heat Exchangers”, *Thermal Science*, vol. 24, no. 1, pp. 499-503 (2020).

- [10] Manisha Nair, Ganesh Jeevanandan, Vignesh and Subramanian Emg, “Comparative Evaluation of Post-Operative Pain after Pulpectomy with K-Files Kedo-S Files and Mtwo Files in Deciduous Molars -a Randomized Clinical Trial”, *Brazilian Dental Science*, vol. 21, no. 4, pp. 411-17 (2018).
- [11] D. Dhabliya, S. Kunche, S. Dingankar, S. S. Dari, R. Dhabliya, and V. Khetani, “Blockchain technology as a paradigm for enhancing cyber security in distributed systems,” *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 2-B, pp. 729–740 (2024), doi: 10.47974/JDMSC-1923.
- [12] Y. Chen, Y. Lin, Y. Hu, S. Member and C. Hsia, “Distributed Real-Time Object Detection Based on Edge-Cloud Collaboration for Smart Video Surveillance Applications”, *IEEE Access*, vol. 10, no. September, pp. 93745-93759 (2022).

Received November, 2024