

**ELLIPTYCAL CURVE CRYPTOGRAPHY**

In Subject: Discrete Mathematics

*by*

Rahul Dewani (22211545)

Piyush Deshmukh (22311415)

Neeraj Koulgi (22311455)

Spandan Kewte (22311588)

Department of Artificial Intelligence and Data Science VIIT

**2020-2021**

**Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Topic** | | **Page No.** |
|  |  | |  |
| **Chapter-1** | **Introduction** | | 2 |
|  | 1.1 | Introduction |  |
|  | 1.2 | Requirements |  |
|  | 1.3 | Design & Problem Statement |  |
|  | 1.4 | Proposed work |  |
| **Chapter-2** | **Methodology** | | 4 |
|  | 2.1 | Datasets |  |
|  | 2.2 | Approach |  |
|  | 2.3 | Platform and Technology |  |
|  | 2.3 | Outcomes & Use cases |  |
|  | 2.4 | Challenges |  |
| **Chapter-3** | **Conclusion** | | 6 |
|  | **References** | |  |

**Chapter 1: Introduction**

* 1. **Introduction**

**What is Elliptic Curve Cryptography (ECC)?**

Elliptic Curve Cryptography (ECC) is a type of public-key cryptography based on the algebraic structure of elliptic curves over finite fields. The security of ECC relies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is computationally harder than similar problems in other cryptosystems like RSA or DSA..

**IMPORTANCE OF Elliptic Curve Cryptography (ECC)**

**Higher Security with Smaller Keys**

* **Efficiency**: ECC provides the same level of security as traditional algorithms like RSA and Diffie-Hellman but with significantly smaller key sizes. For example, a 256-bit ECC key offers equivalent security to a 3072-bit RSA key.
* **Smaller key sizes** mean:
  + Less storage and memory usage.
  + Faster computations.
  + Reduced bandwidth for data transmissions.

**2. Performance for Modern Applications**

* **Faster Cryptographic Operations**: ECC is more efficient for tasks like encryption, decryption, and digital signatures, making it ideal for devices with limited processing power, such as smartphones, smart cards, and IoT devices.
* **Better Performance in Key Agreement Protocols**: ECC is widely used in secure key exchange protocols, like the Elliptic Curve Diffie-Hellman (ECDH) algorithm, because of its speed and reduced computational overhead compared to older algorithms.

**3. Energy Efficiency**

* ECC’s smaller key sizes require less computational power, making it highly energy-efficient. This is critical in resource-constrained environments like IoT devices, mobile applications, and embedded systems.

**4. Security Longevity**

* **Quantum Resistance (to some extent)**: While no classical cryptographic system is fully resistant to quantum computing attacks, ECC with appropriately chosen curves may have some quantum-resilient features, though it still requires post-quantum cryptography for future-proofing.
* With RSA requiring much larger keys to maintain security as technology advances, ECC’s smaller key sizes ensure that systems can remain secure for the foreseeable future without dramatically increasing key lengths.

**5. Widespread Adoption**

* **Standards**: ECC has become a part of many widely recognized standards, including those set by the National Institute of Standards and Technology (NIST), the International Organization for Standardization (ISO), and the National Security Agency (NSA) for classified communications.
* **TLS and SSL**: ECC is now a critical part of securing internet communications, often implemented in TLS (Transport Layer Security) protocols for websites and online services.

**6. Scalability**

* As networks, devices, and users grow exponentially, ECC scales well with modern infrastructure due to its minimal resource demands. It is suitable for secure communications in large-scale systems such as blockchain networks, cryptocurrencies, and financial institutions.

**7. Digital Signatures**

* **ECDSA (Elliptic Curve Digital Signature Algorithm)** is a widely used elliptic curve-based algorithm for creating digital signatures. It is more secure and efficient than traditional digital signature algorithms like RSA.
* ECC-based digital signatures are used in secure email (PGP/GPG), authentication protocols, and secure software updates.

**Requirements:**

* **Mathematical Foundation of ECC:** ECC relies on the properties of elliptic curves over finite fields. The general form of an elliptic curve is y2=x3+ax+by^2 = x^3 + ax + by2=x3+ax+b, where aaa and bbb are constants that satisfy a specific set of conditions to ensure a well-defined curve. The points on the curve represent possible keys in the cryptographic system.
* **Finite Fields and Modular Arithmetic:** Elliptic curve operations are typically defined over finite fields, meaning that arithmetic on the curve is performed modulo a prime number. These operations form the basis of ECC, and they are crucial to the generation of public and private keys.
* **Security Parameters:** ECC offers stronger security at smaller key sizes compared to RSA and Diffie-Hellman (DH). For instance, a 256-bit ECC key provides equivalent security to a 2048-bit RSA key. This is an important consideration in environments with constrained resources, such as embedded systems.
* **Cryptographic Primitives:** ECC is built on fundamental cryptographic principles like one-way functions and the discrete logarithm problem (ECDLP). The security of ECC rests on the difficulty of solving the ECDLP, which is significantly harder than the factorization problem that RSA is based on.

**Problem Statement:**

As the digital world grows, traditional encryption methods face scalability challenges. The increasing computational power available to attackers, combined with the exponential growth of data, has exposed the limitations of algorithms like RSA and Diffie-Hellman. ECC addresses the need for a lightweight cryptographic system that can offer the same level of security as RSA with smaller key sizes, making it ideal for applications with limited processing power and memory, such as mobile devices, IoT, and blockchain technologies.

**Design of ECC :**

In ECC, the core operations include point addition and scalar multiplication on the elliptic curve. These operations are used to derive public and private keys, perform encryption and decryption, and generate and verify digital signatures. The design of ECC systems must ensure that these operations are efficient and secure against potential attacks, such as side-channel attacks.

**Proposed Work:**

 **Objective:** The objective of this project is to demonstrate the advantages of ECC over traditional cryptographic methods like RSA, focusing on key generation, encryption, and decryption processes, as well as simulating real-world use cases such as secure communication and digital signatures.

 **ECC’s Role in Modern Cryptography:** ECC’s efficiency makes it highly suitable for modern applications requiring secure communication without sacrificing performance. It is increasingly used in SSL/TLS protocols, email encryption, and secure mobile communications. The project aims to highlight these applications and show how ECC can be implemented and tested in a practical setting.

 **Proposed Contribution:** The proposed work also includes a simulation of an attack on an ECC-based system, demonstrating the system's robustness against attempts to compromise data by exhausting computational resources.

**Methodology:**

 **Understanding the Mathematical Foundation**:  
The foundation of ECC lies in the algebraic structure of elliptic curves. An elliptic curve is represented by the equation y2=x3+ax+by^2 = x^3 + ax + by2=x3+ax+b, where aaa and bbb are constants that satisfy the condition 4a3+27b2≠04a^3 + 27b^2 \neq 04a3+27b2=0 to ensure that the curve has no singularities. These curves exhibit unique geometric properties that are leveraged in cryptographic systems.

 **Elliptic Curves over Finite Fields**:  
In ECC, elliptic curves are defined over finite fields, meaning that all operations are performed modulo a prime number ppp (in prime fields) or over binary fields (in special cases). Finite field arithmetic ensures that the number of points on the curve is finite and manageable for cryptographic purposes.

 **Point Addition and Doubling**:  
The core operations on elliptic curves are point addition and point doubling. Given two points on the curve PPP and QQQ, point addition produces a third point RRR on the curve. Point doubling is the special case where P=QP = QP=Q, which also yields another point on the curve. These operations are fundamental in ECC for generating public keys and performing encryption.

 **Scalar Multiplication**:  
Scalar multiplication involves multiplying a point PPP on the elliptic curve by a scalar kkk, producing a new point kPkPkP. This operation is computationally intensive and forms the backbone of ECC. It is used to generate public keys from private keys and is the basis of cryptographic strength, relying on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP).

 **Encryption Using ECC**:  
To encrypt a message, ECC represents the message as a point on the elliptic curve, referred to as MMM. A random number kkk is selected, and two points are generated: kGkGkG and M+kQM + kQM+kQ, where QQQ is the recipient’s public key. The ciphertext consists of these two points. This ensures that only the intended recipient, who possesses the private key, can decrypt the message.

 **Decryption Process**:  
Decryption in ECC involves using the recipient’s private key ddd to recover the original message. The recipient computes kGkGkG using the first point of the ciphertext and then uses their private key to compute d(kG)d(kG)d(kG). Subtracting this value from the second point M+kQM + kQM+kQ of the ciphertext gives the original message point MMM.

 **Elliptic Curve Diffie-Hellman (ECDH)**:  
ECDH is a key exchange protocol that allows two parties to securely share a secret key over an insecure channel. Both parties generate their own private-public key pairs and exchange public keys. Using scalar multiplication, they can compute a shared secret that no eavesdropper can derive, as it would require solving the ECDLP.

 **Elliptic Curve Digital Signature Algorithm (ECDSA)**:  
ECDSA is an ECC-based signature scheme used for verifying the authenticity of a message. It involves generating a signature using the sender’s private key and verifying it using the sender’s public key. The use of elliptic curves ensures that the signature is both secure and computationally efficient, making it ideal for digital certificates and secure communications.

 **Performance and Security Considerations**:  
ECC provides strong security with smaller key sizes compared to RSA. For example, a 256-bit ECC key is as secure as a 3072-bit RSA key. This efficiency makes ECC suitable for environments with limited computational power, such as mobile devices and IoT. The method also includes ensuring that elliptic curve parameters are chosen correctly to prevent vulnerabilities, such as weak curves that could be exploited by attackers.

**Visualization of Elliptic Curve Cryptography (ECC) :**

To visualize Elliptic Curve Cryptography (ECC) effectively, here are some of the best graphs to represent the key concepts. Each graph highlights a critical aspect of ECC:

### 1. \*Plot of the Elliptic Curve\*

- \*Purpose\*: Shows the shape of the elliptic curve for a chosen equation like \( y^2 = x^3 + ax + b \).

- \*Graph\*:

- A 2D plot of the elliptic curve in the real number domain.

- Example curve: \( y^2 = x^3 - 4x + 1 \).

- The graph shows the smooth, looping shape of the curve.

- \*Key Takeaway\*: This visual gives an intuitive understanding of the geometric structure of elliptic curves.

### 2. \*Point Addition on the Elliptic Curve\*

- \*Purpose: Illustrates the \*\*point addition\* operation, fundamental to ECC.

- \*Graph\*:

- Plot the curve and two points, \(P\) and \(Q\), on the curve.

- Draw a straight line connecting \(P\) and \(Q\), showing the intersection with the curve at point \(R\).

- Reflect \(R\) across the x-axis to show the result \(P + Q = (x\_3, -y\_3)\).

- \*Key Takeaway\*: This graph demonstrates how elliptic curve operations work geometrically, a key insight into how ECC performs encryption.

### 3. \*Scalar Multiplication Visualization\*

- \*Purpose: Shows how \*\*scalar multiplication\* works on elliptic curves.

- \*Graph\*:

- Start with a single point \(P\) on the curve.

- Repeatedly add \(P\) to itself and show the new points \(2P, 3P, 4P, \dots, nP\).

- This graph would show the trajectory of point multiplication, which forms the foundation of key generation in ECC.

- \*Key Takeaway\*: The process of scalar multiplication is visualized step by step, which is useful for understanding how encryption keys are created.

The Nearest Neighbor Algorithm, while relatively simple, offers a pragmatic solution to the Traveling Salesman Problem. The visual representation of the algorithm's execution on the map enhances the understanding of the optimization process. Challenges in optimizing the tour and the algorithm's efficiency are acknowledged. Further exploration and refinement of TSP algorithms remain intriguing avenues for future work.