Bitcoin

Beyza Kaya

Abstract

This paper investigates the key mathematical concepts that make Bitcoin secure. It focuses on how Bitcoin uses elliptic curve cryptography (ECC), modular arithmetic, and the discrete logarithm problem (DLP) to protect transactions. The paper explains these ideas in simple terms, breaking down the calculations, theories, and methods that form the foundation of Bitcoin's security system.

1 Introduction

Bitcoin is a decentralized digital currency that was introduced by Satoshi Nakamoto in 2009. It uses elliptic curve cryptography (ECC) to ensure the integrity and confidentiality of transactions. The specific curve used is secp256k1, defined over a finite field. This curve allows for secure key generation, digital signatures, and the verification of transactions.

The core of Bitcoin's security lies in the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). This problem underpins the cryptographic strength of Bitcoin, making it resistant to attacks with current computational capabilities.

Modular arithmetic plays a crucial role in elliptic curve operations. It ensures that all calculations remain within the defined finite field, maintaining the integrity of cryptographic processes.

With the advent of quantum computing, traditional cryptographic methods, including those used in Bitcoin, face potential threats. This paper explores the mathematical foundations of Bitcoin's cryptography.

2 Elliptic Curve Cryptography in Bitcoin

Elliptic curves are algebraic structures defined by the equation:

$$E : y^2 \equiv x^3 + ax + b \pmod{p} \tag{1}$$

where a and b are constants that must satisfy the condition for non-singularity:

$$4a^3 + 27b^2 \not\equiv 0 \pmod{p} \tag{2}$$

This condition ensures the curve has no cusps or self-intersections, which is crucial for maintaining the security properties of the curve.

In Bitcoin, the elliptic curve used is secp256k1, defined by the equation:

$$y^2 \equiv x^3 + 7 \pmod{p} \tag{3}$$

where p is a large prime number:

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1 \tag{4}$$

For the curve $y^2 = x^3 + 7$, we check the discriminant Δ:

$$\Delta = -16(4a^3 + 27b^2) \tag{5}$$

For $a = 0$ and $b = 7$:

$$\Delta = -16(0 + 27 \times 7^2) = -16 \times 1323 \neq 0 \tag{6}$$

Thus, the curve is non-singular, which is critical for its application in cryptography.

Given two distinct points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, their sum $R = P + Q = (x_3, y_3)$ on the curve is computed using the slope formula:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod{p} \tag{7}$$

The x-coordinate of the resulting point R is:

$$x_3 = \lambda^2 - x_1 - x_2 \pmod{p} \tag{8}$$

The y-coordinate y_3 is:

$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p} \tag{9}$$

For $P = Q$, the slope λ becomes:

$$\lambda = \frac{3x_1^2 + a}{2y_1} \pmod{p} \tag{10}$$

The coordinates of the resulting point $R = 2P = (x_3, y_3)$ are:

$$x_3 = \lambda^2 - 2x_1 \pmod{p} \tag{11}$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p} \tag{12}$$

3 Modular Arithmetic in Bitcoin

Modular arithmetic is central to all cryptographic operations in Bitcoin, ensuring operations remain within a finite field F_p. The basic operations are:

$$(a + b) \pmod{p} = (a + b) - p \left\lfloor \frac{a + b}{p} \right\rfloor \tag{13}$$

$$(a - b) \pmod{p} = (a - b) + p \left\lfloor \frac{b - a}{p} \right\rfloor \tag{14}$$

$$(a \times b) \pmod{p} = (a \times b) - p \left\lfloor \frac{a \times b}{p} \right\rfloor \tag{15}$$

$$a^{-1}p \quad is\ found\ by\ solving \quad ax \equiv 1 \pmod{p} \tag{16}$$

Modular exponentiation, a fundamental operation in public-key cryptography, is defined as:

$$c = a^b \pmod{p} \tag{17}$$

Modular exponentiation can be efficiently computed using the method of exponentiation by squaring. For an exponent $b = \sum_{i=0}^{n-1} b_i 2^i$, we compute:

$$c = \left(\prod_{i=0}^{n-1} (a^{2^i})^{b_i} \right) \pmod{p} \tag{18}$$

This method reduces the complexity from $O(b)$ to $O(\log b)$, making it feasible for large exponents.

4 Elliptic Curve Discrete Logarithm Problem (ECDLP)

The security of Bitcoin's elliptic curve cryptography relies on the intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP). The problem is formally stated as:

$$Given\ P,\ Q\ on\ E(F_p),\ find\ k\ such\ that\ Q = kP \tag{19}$$

The best-known algorithm for solving the ECDLP is Pollard's Rho algorithm, which has a time complexity of $O(\sqrt{n})$, where n is the order of the group generated by P. Given the large size of p, typically 256 bits, $\sqrt{p}$ is prohibitively large, making the ECDLP infeasible to solve with current computing power.

5 Key Generation and Digital Signatures in Bitcoin

Bitcoin uses the Elliptic Curve Digital Signature Algorithm (ECDSA) for secure transactions.

A private key k is a large random integer, and the corresponding public key Q is computed as:

$$Q = kG \tag{20}$$

where G is the base point on the elliptic curve $E(F_p)$.

The signature on a message m involves selecting a random integer r and computing:

$$R = rG \tag{21}$$

$$r = (R_x) \pmod n \tag{22}$$

$$s = r^{-1}(H(m) + kr) \pmod n \tag{23}$$

The verification process involves computing:

$$u_1 = s^{-1}H(m) \pmod n \tag{24}$$

$$u_2 = s^{-1}r \pmod n \tag{25}$$

$$R' = u_1G + u_2Q \tag{26}$$

The signature is valid if $R'_x \equiv r \pmod n$, ensuring that the message was signed by the holder of the private key k.

Proof of Work (PoW) Mechanism

Proof of Work (PoW) in Bitcoin is like a game where miners compete to solve a puzzle. The puzzle involves finding a special number called a "nonce." When this nonce is combined with the block's data and passed through a cryptographic function (like a digital lock), it produces a unique code (called a hash). The goal is to find a nonce that makes the hash start with a certain number of zeros, which is really hard to do and takes a lot of tries. The first miner to find this winning nonce gets to add the new block to the blockchain and earns a reward.

$$H(x) = SHA - 256(x)$$

where H is the cryptographic hash function producing a 256-bit output.

Find a nonce n such that:

$$H(Block\ Data \| n) < T$$

where T is the target threshold.

$$T = \frac{2^{256}}{D}$$

where D is the difficulty level.

$$P(valid\ nonce) = \frac{T}{2^{256}} = \frac{1}{D}$$

$$E[trials] = D$$

$$D_{new} = D_{current} \times \frac{t_{actual}}{t_{target}}$$

where t_{actual} is the actual time for recent blocks, and t_{target} is typically 600 seconds.

$$t_{attack} \approx k \times t_{target} \times \frac{1}{attacker's\ hash\ power\ fraction}$$

where k is the number of blocks to re-mine.

6 Quantum Computing and ECC

Quantum computing poses a significant threat to elliptic curve cryptography, particularly through Shor's algorithm, which can solve the ECDLP in polynomial time.

Shor's algorithm reduces the complexity of solving ECDLP from $O(\sqrt{n})$ to $O(\log n)$.

Given an elliptic curve E over a finite field F_p, a point $P \in E(F_p)$, and a point $Q \in E(F_p$, find the integer k such that:

$$Q = kP$$

The classical approach, using Pollard's Rho algorithm, solves the ECDLP with a time complexity:

$$T_{classical} = O(\sqrt{n})$$

where n is the order of the group generated by P. For $n \approx 2^{256}$:

$$T_{classical} = O(2^{128})$$

Shor's algorithm, implemented on a quantum computer, reduces the time complexity of solving the ECDLP to:

$$T_{quantum} = O((\log n)^3)$$

For $n \approx 2^{256}$:

$$T_{quantum} = O((\log 2^{256})^3) = O(256^3) = O(2^{24})$$

Thus:

$$T_{classical} = O(2^{128})$$

$$T_{quantum} = O(2^{24})$$

This reduction from $O(2^{128})$ to $O(2^{24})$ illustrates the substantial decrease in computational difficulty provided by quantum algorithms.

7 Conclusion

This paper provided a detailed mathematical analysis of Bitcoin's cryptographic foundation and explored elliptic curve cryptography (ECC), focusing on the secp256k1 curve, alongside discussing key operations like point addition and doubling. Modular arithmetic and the discrete logarithm problem (DLP) were also examined, highlighting their roles in Bitcoin's security. In summary, Bitcoin's security relies on complex mathematical operations that remain effective against classical attacks.

8 References

References

[1] N. Koblitz, *A Course in Number Theory and Cryptography*, Springer-Verlag, New York, 1987.

[2] J. H. Silverman, *A Friendly Introduction to Number Theory*, Prentice-Hall, New Jersey, 1997.

[3] J. H. Silverman, J. T. Tate, *Rational Points on Elliptic Curves*, Springer, 1994.

[4] T. R. Shemanske, *Modern Cryptography and Elliptic Curves*, American Mathematical Society, 2017.

[5] Q. ShenTu, J. Yu, “A Blind-Mixing Scheme for Bitcoin based on an Elliptic Curve Cryptography Blind Digital Signature Algorithm,” 2015.

[6] E. S. Jiménez-Ayala, J. Medina, “The Mathematics Behind Cryptocurrencies and Blockchain,” 2019.