# ELLIPTIC CURVE CRYPTOGRAPHY

VAGHESAN SUNDARAM[1]

Abstract. In this paper, we explore how to utilize Elliptic Curve Cryptography to create a shared secret between two parties (Elliptic Curve Diffie-Hellman Key Exchange). The findings in this paper demonstrate ECC's security and effectiveness.

## 1. Introduction

Elliptic Curve Cryptography is a type of public key cryptography based on the algebraic structure of elliptic curves when defined over finite fields. It was proposed by Neal Koblitz and Victor S. Miller independently, who both won Levchin Prizes for the invention of ECC in 2021.

## 2. Elliptic Curves

**Definition 2.1.** We define an elliptic curve as a plane curve with the equation $y^2 = x^3 + ax + b$.

For visualizing the following definitions, we will define our curve over the real numbers. We will also define 4 operations for points on an elliptic curve.

**Definition 2.2.** We define point negation as taking a point on the curve and reflecting it over the y-axis, as shown in Figure 1. For example, for a point P on a curve, the negation of P is -P.

**Definition 2.3.** We define point addition as taking two points on a curve, finding the intersection of the line between those points and the curve, and negating the intersection point, as shown in Figure 2. For example, for points P and Q on a curve, point addition would be shown as P + Q = R, where R is the result of the addition.

*Key words and phrases.* elliptic curve, key agreement, cryptosystem, cryptography, discrete math.

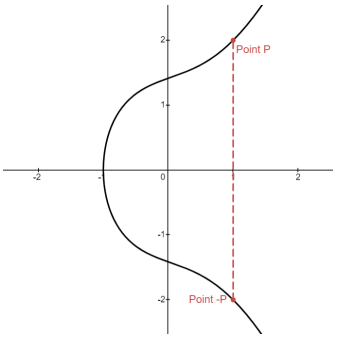


Figure 1. Point Negation of a point P on an Elliptic Curve over the Real Numbers

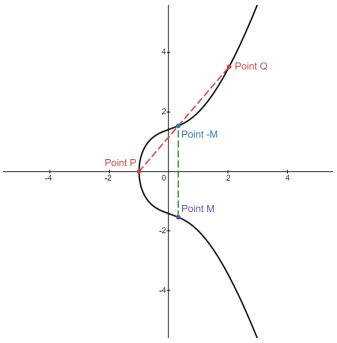


Figure 2. Point Addition of points P and Q on an Elliptic Curve over the Real Numbers

**Definition 2.4.** We define point doubling as taking a point on a curve, finding the intersection of the tangent line to the curve at that point and the curve, and negating the intersection point, as shown in Figure 3. In principle, it is the same as regular point addition, but when the two points have the same coordinates. Functionally, this is the same as doing P+P = 2P.

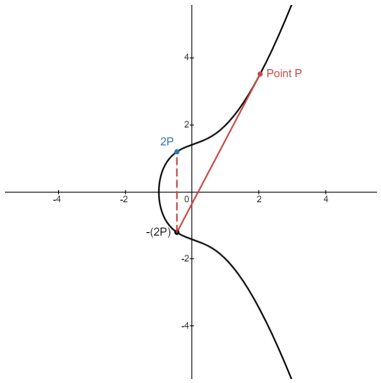


FIGURE 3. Point Doubling of a point P

**Definition 2.5.** We define point multiplication as repeated point addition, which is intuitively associative and commutative. Points can be multiplied by scalars, showing how many times the addition takes place. For example, if we take a point P on a curve, 4(P) refers to (P+P+P+P). Additionally, P + 3P = 4P, and 2(2(P)) = 4P, as demonstrated in Figures 4 and 5. Furthermore, 0(G) = a point at infinity, which we represent with O. O functions as the identity element, where any point P + O = P.

## 3. Domain Parameters

To use Elliptic Curve Cryptography, both parties need to agree on domain parameters, which are characteristics of the curve. These parameters are communicated in the form (p, a, b, G, n, h).
Elliptic curves in ECC are not defined over the real numbers. They are instead defined

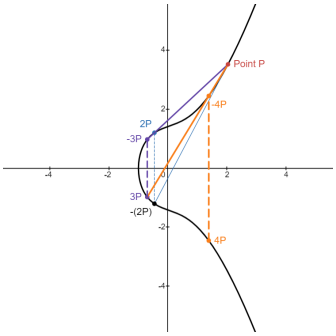


FIGURE 4. Point Multiplication of a point P via repeated addition

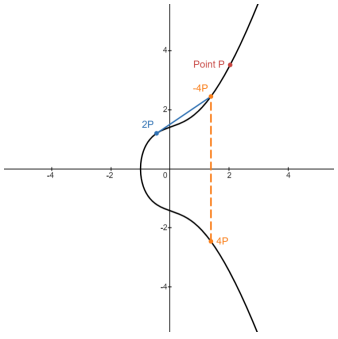


FIGURE 5. Point Multiplication of a point P via the associative property of Point Addition

over a finite field. An elliptic curve will be written in the form $y^2 = x^3 + ax + b(modp)$. Instead of having a smooth curve, we will have a set of discrete coordinates on a plane. All of these coordinates will satisfy the equation. Point negation, multiplication, addition, and doubling all still apply to the finite field. The lines drawn from points wrap around the borders of the graph.

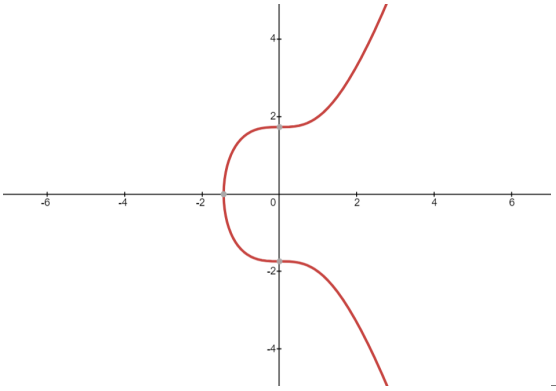


FIGURE 6. $y^2 = x^3 + 0x + 3$ over the real numbers

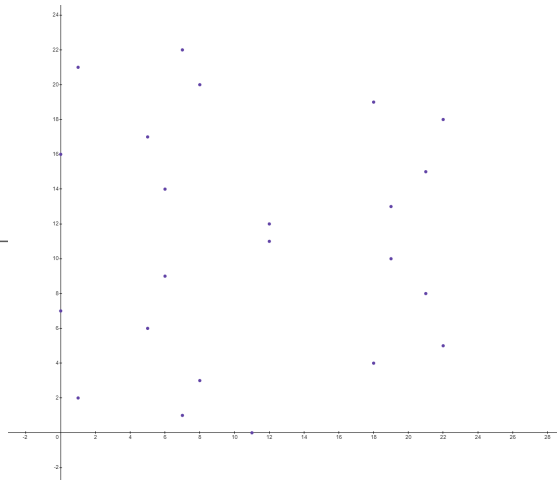

FIGURE 7. $y^2 = x^3 + 0x + 3(mod23)$ as a finite field

**Definition 3.1.** We define p as the modulus, the size of the field, which is a prime number. Only when p is prime will point addition work when defined over a finite field.

**Definition 3.2.** We define a and b as constants in the equation for the elliptic field as $y^2 = x^3 + ax + b$.

**Definition 3.3.** We define G as a point on the EC, the generator point that forms the cyclic group of EC points.

If we continually add G to itself, we will eventually find that we loop back to G.

**Definition 3.4.** We define n as the order of G. n is the smallest scalar where $n(G) =$ O.

**Definition 3.5.** We define h, the cofactor, as $n/r$. It is ideally 1, and must be less than or equal to 4.

Generally, new domain parameters are not generated for each exchange, as it is computationally lengthy. Because of this, many standard bodies, such as NIST and SECG, have published standard curves for multiple common field sizes.

## 4. Public and Private Keys

ECC is similar to many other public key cryptosystems in the way that each party will share a public key and keep secret a private key.

Suppose there are two parties; Alice and Bob. Alice and Bob have already agreed on domain parameters. Alice and Bob both generate very large random numbers for their private keys, being m and n. They calculate their public keys, being m(G) and n(G)

respectively.

By exchanging their public keys. Alice can calculate m * n(G), and Bob can calculate n * m(G). Because elliptic curve point multiplication is associative and commutative, m * n(G) = mn(G) = n * m(G). Alice and Bob now have a shared secret, being the point on the EC, mn(G).

## 5. Discussion

The security of this key exchange, known as the Elliptic Curve Diffie Hellman key exchange (ECDH) stems from the difficulty in computing an integer $m$ where $m(G) = P$, G being a generator point and P being the public key, known as the Elliptic Curve Discrete Logarithm Problem. To clarify, if Alice sent Bob $m(G)$ and kept $m$ secret, it would be difficult to calculate $m$ from $m(G)$. When $p$, $a$, and $b$ from the domain parameters are very large, this becomes extremely difficult[1].

To date, the hardest ECC scheme publicly broken to date had a 112-bit key. It was broken in July 2009 using 200 Playstation 3 consoles, and, had it been running continuously, could have finished in 3 months.

On a hypothetical quantum computer, Shor's algorithm could be used to compute discrete logarithms. However, as of June 2024, no quantum computer has the qubits or Toffoli gates necessary[3]. Elliptic Curve Cryptography seems to be weak to future quantum attacks, and may not be a viable option in the coming years.

Furthermore, there are concerns that the NSA has inserted backdoors into at least one elliptic curve-based PRNG. Edward Snowden leaked internal memos suggesting that the Dual EC DRBG standard has backdoors in it.

This is not to suggest that ECC is insecure, or less secure than other cryptographic options. It offers the same level of security as RSA with smaller keys, resulting in less computing power and network load[4]. However, care should be taken in choosing a curve, as there are many that contain weaknesses that can be exploited.

## References

1. "The Elliptic Curve Discrete Logarithm Problem." 5.2 the Elliptic Curve Discrete Logarithm Problem, Certicom, www.certicom.com/content/certicom/en/52-the-elliptic-curve-discrete-logarithm-problem.html. Accessed 10 June 2024.
2. Koblitz, Neal. "Elliptic curve cryptosystems." Mathematics of Computation, vol. 48, no. 177, Jan. 1987, p. 203, https://doi.org/10.2307/2007884.
3. Roetteler, Martin. ArXiv:1706.06752v3 [Quant-Ph] 31 Oct 2017, arxiv.org/pdf/1706.06752.pdf. Accessed 10 June 2024.
4. "Top NSA Banner." The Case for Elliptic Curve Cryptography - NSA/CSS, web.archive.org/web/20090117023500/www.nsa.gov/business/programs/elliptic_curve.shtml. Accessed 10 June 2024.

[1] Edison Academy Magnet School, Edison, NJ 08837, USA.
*Email address*: sundaramv@mcmsnj.net