Elliptic Curve Cryptography

Cryptography

School of Engineering and Technology
CQUniversity Australia

Prepared by Steven Gordon on 23 Dec 2021,
elliptic.tex, r1949

Contents

Elliptic Curve (definition)

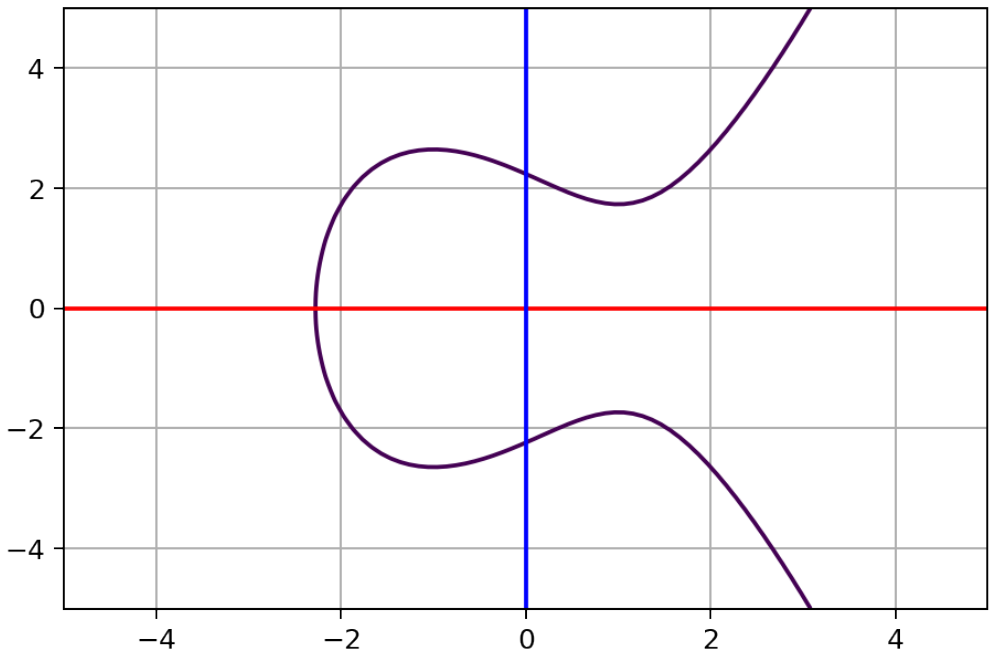
An elliptic curve is defined by:

$$y^2 = x^3 + ax + b$$

(with some constraints of constants a and b)

The constraints on a and b specify the relationship between the values, i.e. you cannot necessarily choose any values. We will not go into that detail here.

Elliptic Curve for $y^2 = x^3 - 3x + 5$



Credit: Generated based on MIT Licensed code by Fang-Pen Lin

The figure on slide 4 shows an example elliptic curve where $a = -3$ and $b = 5$, plotted for x values from -4 to 4. An elliptic curve always mirrors itself about the horizontal (red) axis.

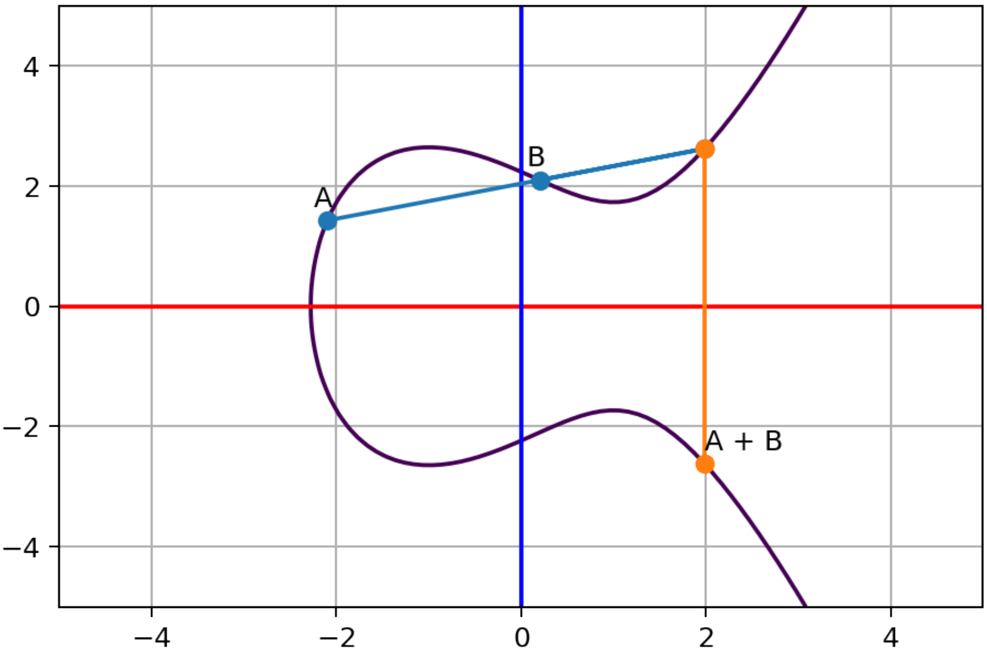
Addition Operation with an Elliptic Curve (definition)

Select two points on the curve, A and B, and draw a straight line through them. The line will intersect with the curve at a third point, R (and no other points). The horizontal inverse of point R, is defined as the addition of A and B.

$$A + B = -R$$

See the following figure for an example of this concept. Note the points, A, B, R and -R are just (x, y) coordinates.

Addition Operation on Elliptic Curve

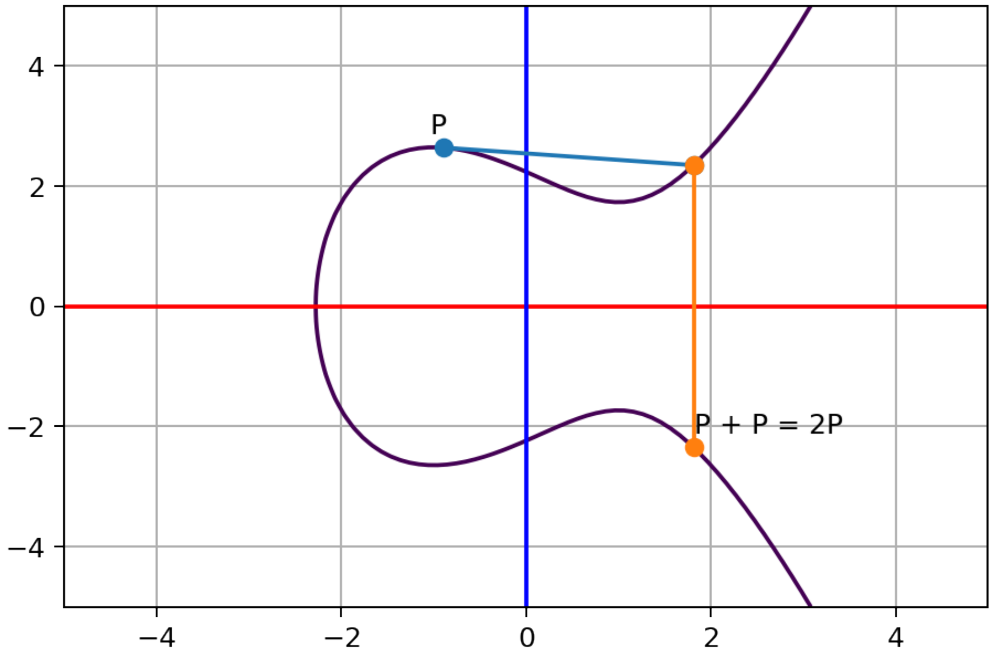


Credit: Generated based on MIT Licensed code by Fang-Pen Lin

The figure on slide 6 shows the concept of addition. Adding the points A and B results in the point shown as A+B. There is always a third point that intersects the curve on the line between A and B, and there is always an inverse of this point.

Note that we could continue the addition. For example, with A+B, add another point C, to arrive at a new point A+B+C. And so on.

Self Addition on Elliptic Curve

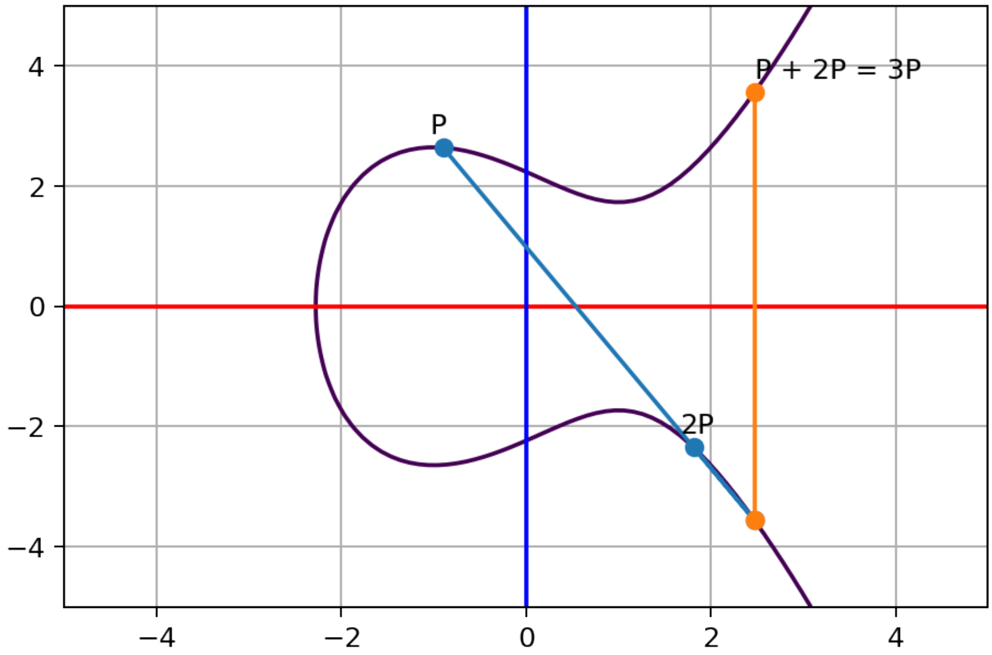


Credit: Generated based on MIT Licensed code by Fang-Pen Lin

The figure on slide 7 shows the self addition of point P. When adding a single point P to itself, the line that intersects P is chosen as the line tangent to P. So P+P = 2P.

We can continue to add P.

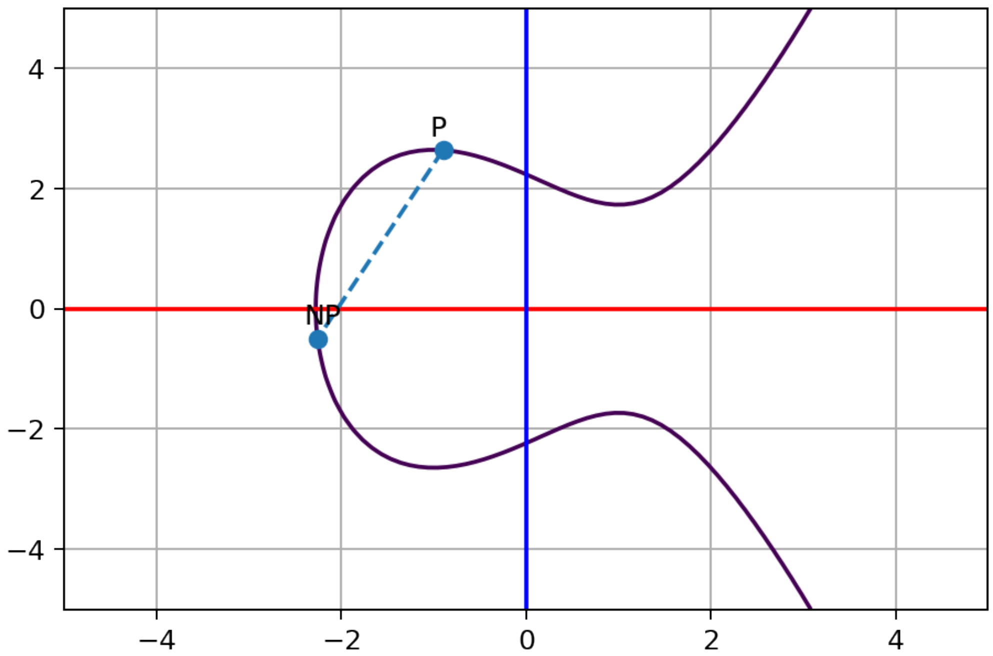
P + 2P on Elliptic Curve



Credit: Generated based on MIT Licensed code by Fang-Pen Lin

The figure on slide 8 shows P + 2P = 3P. Then we can add P again to get 4P and so on.

NP on Elliptic Curve



Credit: Generated based on MIT Licensed code by Fang-Pen Lin

The figure on slide 9 shows NP. In this example N=13. That is, we start with point P, and add P twelve times, resulting in the point 13P.

How is Point Addition used in Elliptic Curve Cryptography?

- User chooses a point P (global public parameter)
- User chooses a large, random N (private key)
- User calculates NP (public key)
  - Easy, since there is a shortcut (described shortly)
- Challenge for attacker: given NP, find N
  - Computationally hard for large N

As with other public key systems, elliptic curve cryptography relies on the fact that it is easy for the user to generate the public and private key, but practically impossible for an attacker to find the private key from the public key.

Why is that the case? So far we said NP is found by adding P $N-1$ times, that is, takes $N-1$ addition operations. So an attacker could simply start with P, and keep adding P until they get an answer of NP. Now the know how many additions, i.e. the private value N.

However if N is large enough the attackers method will be practically impossible. And for the user to generate NP when they know N, there is a shortcut that is practically achievable.

Shortcut for Calculating NP

- Assume N is large, e.g. 256-bit random number
- Naive point addition: $P + P + P + P + \ldots + P + P$ ($2^{256} - 1$ additions)
- Shortcut algorithm for point addition:
  - Calculate P, $P + P = 2P = 2^1P$, $2P + 2P = 4P = 2^2P$, $4P + 4P = 8P = 2^3P$, ..., $2^{255}P$ (255 additions)
  - Write N as binary expansion, e.g.:
    - $N = 233 = 2^7 + 2^6 + 2^5 + 2^3 + 2^0$
    - $NP = 2^7P + 2^6P + 2^5P + 2^3P + 2^0P$
    - In this example, there are 4 point additions
    - Maximum number of point additions for 256-bit N is 255
  - Calculate NP using the binary expansion
  - Maximum number of point additions for 256-bit N: 255 + 255 = 510

In summary, knowing the b-bit value N, the user needs to perform about $2 \times b$ point additions. This is easy. But the attacker, who doesn't know N, must perform about 2^b point additions, which is practically impossible.

Elliptic Curve with Modular Arithmetic

- The above discussed a normal elliptic curve
- But to ensure all values contained within finite coordinate space, modular arithmetic is used
- $y^2 \bmod p = (x^3 + ax + b) \bmod p$
- p is a prime number

The figures and examples given previously shown an elliptic curve without modular arithmetic. But in elliptic curve cryptography, modular arithmetic occurs. The same principles, and reasoning why it is hard for the attacker, still apply. The plots of the elliptic curve in modular arithmetic look different however—they now have distinct points in a finite coordinate space. Search online for examples.

Contents

Overview of Elliptic Curve Cryptography

Applications of Elliptic Curve Cryptography

Elliptic Curve Cryptography in OpenSSL

Applications of ECC

- Secret key exchange, e.g. ECDH, ECMQV
- Digital signatures, e.g. ECDSA, EC-KCDSA
- Public key encryption, e.g. ECIES, PSEC

The most common applications are for secret key exchange, especially with ECDH, and digital signatures with ECDSA. We will look at ECDH in the following.

Elliptic Curve Diffie-Hellman Key Exchange (algorithm)

Assume users A and B have EC key pairs: $PU_A = NP$, $PR_A = N$, $PU_B = MP$, $PR_B = M$.

1. User A calculates secret $S_A = N \cdot PU_B = NMP$ using shortcut point addition.
2. User B calculates secret $S_B = M \cdot PU_A = MNP$ using shortcut point addition.

Diffie-Hellman key exchange can be used using ECC so that both users obtain a shared secret over an insecure channel. Users agree on a public point P. They generate their own keypairs, where the private key is some large random number, and the public key is that number times P. Note that in the key generation, each user can use the shortcut to calculate NP or MP.

Assume the users exchange public keys. They then use their own private key multiplied by the other's public key. Again, the shortcut point addition can be used. Both will arrive at the same point (coordinate), i.e. $NMP = MNP$. This is the shared secret.

An attacker that knows the public keys and initial point P has to find either N or M. If those numbers are large enough, this is practically impossible.

Choosing Parameters for ECC

- Parameters for ECC are usually standardised
  - Base point, P (also referred to as generator, G)
  - Curve parameters, a and b
  - Prime, p
  - Other parameters also included
- Common curves (see also `https://safecurves.cr.yp.to/`):
  - NIST FIPS 186: P-256, P-384 and 13 others
  - SECG: secp160k1, secp160r1, . . . (NIST curves are a subset)
  - ANSI X9.62: prime192, prime256, . . .
  - Other curves: Curve25519, Brainpool

SECG in SEC 2 defined a large set of curves. The NIST curves were a subset of the SEC 2 curves. NSA Suite B curves are a subset of NIST curves.

Contents

Overview of Elliptic Curve Cryptography

Applications of Elliptic Curve Cryptography

Elliptic Curve Cryptography in OpenSSL