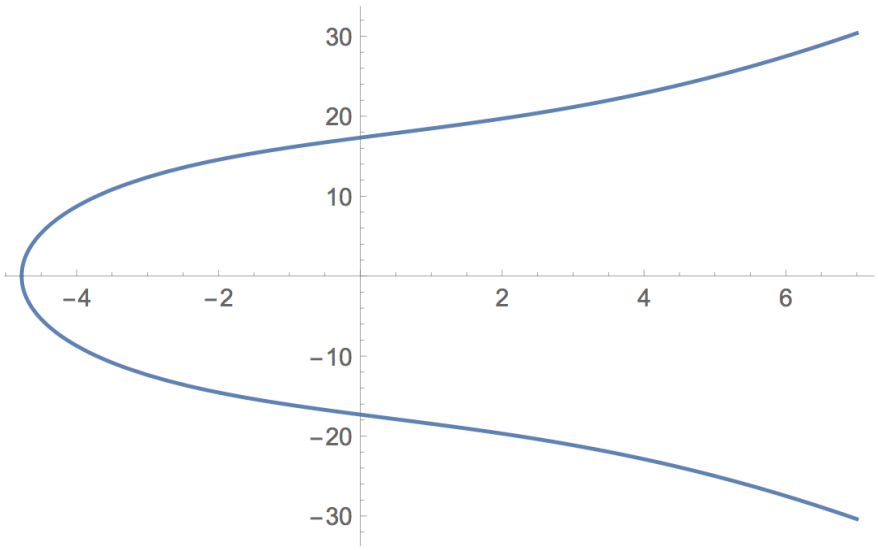
Elliptic curve cryptography

Elliptic curve cryptography is based on the finite field congruence of elliptic curves such as

$$x^3 + ax + b \equiv y^2 \,(\mathrm{mod}\, p)$$

where $(\mathrm{mod}\, p)$ is modulo prime p and $\equiv$ is read "is congruent to". An elliptic curve is pictured below.



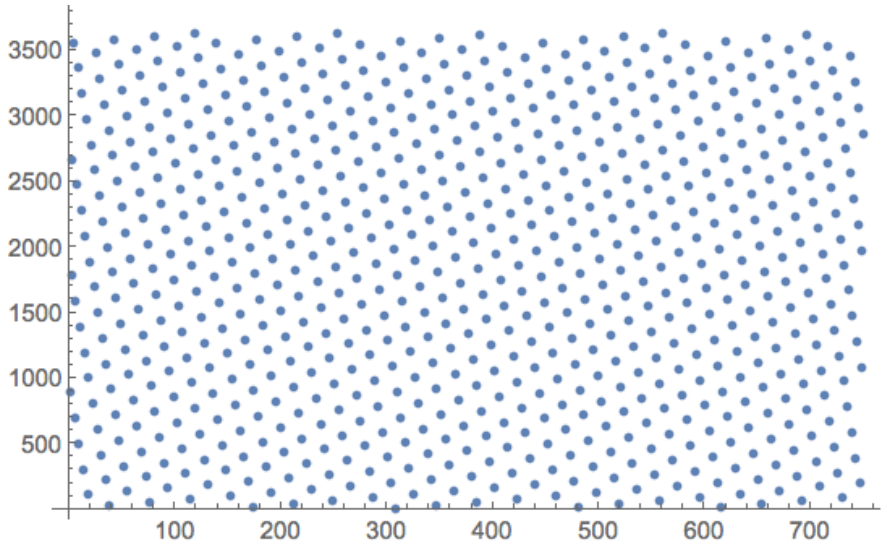
Elliptic curve

Once a base solution is found, say $G = \{x_0, y_0\}$, congruence (modulo p) is maintained for all integer powers up to $n \leq p - 1$.

$$\left(x_0^3 + ax_0 + b\right)^j \equiv \left(y_0^2\right)^j \,(\mathrm{mod}\, p)\,, \text{ for } j = 1, 2, \ldots, n$$

This produces a cloud such as depicted below. As is the case, for the cardinal

elliptic curve, the cloud has an axis of symmetry.



Elliptic curve cloud

Security is driven by the computational complexity of the discrete logarithm problem (though mathematically unproven). That is, it is easy to compute K from $G^d \equiv K \pmod{p}$ but very difficult to determine d from K and G.

First, we describe an encryption scheme then we present an example.

1 Encryption scheme

1. The parties select a common encoding scheme and common knowledge parameters: a, b, k, n, p.

2. Find a base solution or generator to the elliptic curve $G = \{x_0, y_0\}$ such that

$$x^3 + ax + b \equiv y^2 \pmod{p}$$

3. The receiver randomly generates a private key $d < n$, then distributes a public key $G^d \equiv K \pmod{p}$.

4. The sender randomly generates a private key $r < n$, encodes the message m as $em = m * k + j$, where $j = 1, 2, \ldots$ such that y is found that fits on the elliptic curve, and encrypts two quantities: That is, try $j = 1$, if y is found then stop, otherwise try $j = 2$, and so on. This step is the most time-intensive and the sender finds the smallest value j and shares it with the receiver. Alternatively, the sender chooses k such that $j = 1$ and this is understood by the receiver. The sender's encoding is $G^r \equiv C_1 \pmod{p}$ and $K^r + em \equiv C_2 \pmod{p}$. The probability of failure to find a square for y is approximately $\left(\frac{1}{2}\right)^k$.

5. The receiver decrypts the message by $G^{dr} + em - G^{dr} \equiv C_2 - C_1^d \pmod{p}$ to recover em. If the em doesn't reside on the elliptic curve the receiver requests the character be resent. Otherwise, the receiver recovers the message as $(em_x - j) \times \left(k^{-1}\right) \equiv m \pmod{p}$ where em_x refers to the x coordinate of em.

6. The receiver decodes the message.

2 Example

1. Suppose the encoding scheme adopted is "a" $= 11$, "b" $= 12, \ldots,$ "z" $= 36, \ldots$ (might include upper case and punctuation, etc.). Common parameters are $a = -1, b = 188, k = 20, n = 727, p = 751$.

2. Then, a generator is $G = \{45, 97\}$.

3. The receiver's (Bob) private (randomly generated) key is $d = 258$. Then, the receiver's public key is

$$G^d \equiv K \pmod{p} = \{189, 745\}$$

4. Suppose the message is "b" and $m = 12$ is encoded as $em = \{12 * 20 + 1, y\} = \{241, 230\}$. The sender (Alice) randomly generates private key $r = 483$ and encrypts the message

$$\begin{aligned} G^r &\equiv C_1 \pmod{p} = \{490, 271\} \\ K^r + em &\equiv C_2 \pmod{p} = \{487, 162\} \end{aligned}$$

5. The receiver (Bob) decrypts the message by $C_2 - C_1^d \equiv em \pmod{p}$

$$= \{241, 230\}$$

$$(241 - 1) \times \left(20^{-1}\right) \equiv m \pmod{p} = 12$$

6. The receiver (Bob) checks the consistency of the decrypted message as a point on the elliptic curve and, if coherent, decodes the message "b". If the decrypted message is not coherent, new private keys, d and r, are randomly generated and possibly new common parameters a, b, k, n, p, and G then Alice sends the message again.

The example is a bit misleading as an eavesdropper (Eve) could easily employ a brute force search and recover the private keys thus breaking the encryption in this simple example. However, if n and p ($n \leq p - 1$) are chosen to be large, say, 10^{50} or 10^{100}. Searching for the private keys, d and/or r, randomly drawn from $[1, \ldots, n]$ is akin to searching for the proverbial needle in a haystack and likely requires an extraordinary amount of computing time.