Elliptic curve cryptography is a public key algorithm that is used today to protect data. It is based on the mathematics of elliptic curves, where a and b are constants and defined by,

y^2 = x^3 + ax + b

History

The elliptic curve was invented in the mid-1980s by mathematicians, Koblitz and Miller. The idea was to use the structure of elliptic curves to create a public key system, as a similar replacement to the standard RSA.

Over time, the Elliptic curve has become increasingly popular, especially in the areas of digital signatures or software keys. The US National Security Agency has even recommended the Elliptic curve for many protocols and standards, like SSL/TLS and the Advanced Encryption Standard (AES).

Encryption Process

In the Elliptic curve, a public key is generated by multiplying a point on the elliptic curve by a private key. This new point becomes the public key.

To encrypt a message using the Elliptic curve, the sender needs the recipient's public key. The sender uses that public key to generate a shared key, which is then used to encrypt the message. Which can be sent to the recipient, using their private key to decrypt.

Here is an example, of Elliptic curve cryptography:  
  
>Suppose Clark wants to send a message to Drew using elliptic curve cryptography.

>Drew has already generated his public and private keys following:

-Drew selects both,

-an elliptic curve E and a base point P on the curve.

-a random integer d, 1 < d < n as a private key.

n = order of the base point P.

and computes the public key,

Q = d \* P.

>Drew shares his public key Q with Clark.

>Clark is now able encrypt his message.

-Clark selects a random integer k between 1 and n-1.

and computes the point,

R = k \* P on the elliptic curve E.

and computes the shared secret key, S.

S = k \* Q = k \* d \* P.

>Clark encodes his message M into a point on the curve E.

Let C be the encoded message.

-Clark computes the ciphertext (R, C {XOR} H(S)),

H = cryptographic hash function.

>Clark sends the ciphertext to Drew, who uses his private key to decrypt the message.

>Drew is now able to compute the shared secret key and

-Shared secret key - S = d \* R.

-encrypted message S' = k \* S = k \* d \* R = k \* d \* P.

>Drew is now able to decrypt, getting the original method.

M = C {XOR} H(S').

Drew is the only one who can compute the shared secret S, since he knows his private key d, obtaining the plaintext message M.

Security

One interesting element of the Elliptic curve is that it requires smaller key sizes and faster computations compared to RSA. This makes the Elliptic curve more efficient and suitable for resource-constrained environments, such as mobile or handheld technology. The elliptic curve is also resistant to common attacks like brute force attacks. The elliptic curve can be seen in electronic commerce, and digital rights management and It is also used in smart cards, where small key sizes and efficient computations are necessary.

One concern with Elliptic curve cryptography in the future is quantum attacks, which use quantum computers to break the encryption. It is important to use proper key sizes and implementation techniques. The National Institute of Standards has published many guidelines for the use of the Elliptic curve, including recommendations for key sizes (128, 192, and 256 bits) and other parameters.