Joyce Yeo 26 Feb 2020

Elliptic Curve Cryptography (ECC)

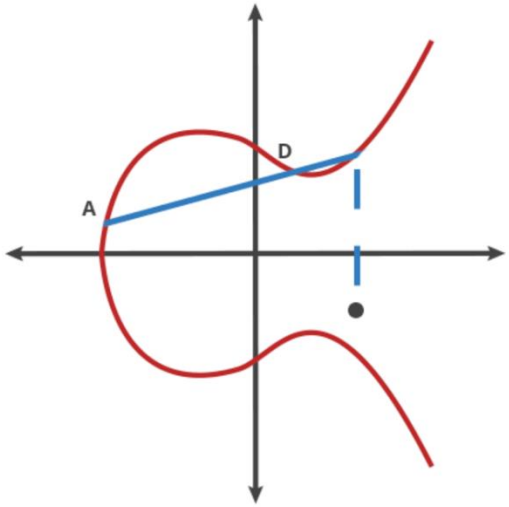
Cryptography is often based on mathematical problems. A very popular cryptographic scheme used today is RSA, which is based on the mathematically difficult problem of factoring large numbers. To be more precise, these mathematical problems are easy to compute in one direction, but difficult in the other. In RSA, multiplication of factors is easy, but factoring is difficult. However, a growing problem with RSA is that the key length needed to be secure is **2048 bits**, which means that multiplying the two primes and storing them is already challenging. With more advances in breaking RSA systems, **the key length will only increase**.

As such, cryptographers have turned to alternative areas of mathematics that share the "one-way" property – easy to compute in one direction, but difficult to reverse the computation. In the 1970s, ECC began to develop. Elliptic Curve Cryptography has gain traction because it can achieve **the same security as RSA, but with a smaller key length**. On average, the time taken to break a 2048-bit RSA key is the same time taken to break a 224-bit ECC key. By drawing an analogy to using passwords, it is clear why shorter keys are more advantageous. Imagine an attacker can crack a 2048-character password in the same time as a 224-character password, then there is no incentive to use a 2048-character password as each time you would have to enter the long password.

Elliptic Curve Mathematics

Elliptic curves have an equation of the form:

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



In the DH Mathematics, we defined group operations as taking multiplication modulo the prime p. In a similar way, an **operation** is defined in elliptic curves.

Given two points A and B on the curve, define the * operation:

A * B = the **negation** of the point resulting from the **intersection** of the curve and the straight line defined by the points A and B

(The * operation was defined for elliptic curve cryptography, because it was discovered that this * operation is one-way.)

Recall: In DH Key Exchange, we saw the logarithm is easy but discrete logarithm is hard due to the modulus. In the same way, ECC mathematicians defined a boundary box for the curve, and going over this box would mean having to "wrap around" it. This wrapping around is mathematically known as a finite field.

Example: The box is defined as x = 0 to x = 10, y = 0 to y = 10. Here are how some points would wrap around.

(x = 3, y = 1) → (x = 3, y = 1)	(x = 4, y = 22) → (x = 4, y = 2)
(x = 3, y = 11) → (x = 3, y = 1)	(x = 13, y = 22) → (x = 3, y = 2)

The ECC Assumption

ECC will take an original point, e.g. P (5, 6) and * with itself n times, where n is the secret key. Then, we get a new point $P^{*,n}$ or P * with itself n times.

ECC Assumption: Given P and $P^{*,n}$, it is difficult to calculate n.

Given the starting point P and the resulting point of P * with itself n times, it is difficult to find n. This assumption in ECC is very similar to the Discrete Logarithm DH assumption.

DH Assumption: Given g and g^x , it is difficult to calculate x.

After decades of research, no one has yet to find an efficient algorithm to break the ECC assumption. ECC is very similar to DH in that it is discrete-logarithm based.

Applying ECC to Encrypt a Message

The ECC public keys comprises the curve, the starting point P and a prime number k which defines the box boundary. The NIST keeps a list of curve equations and each curve has a number (e.g. Curve25519), to make to easier to use ECC.

Given that Alice has a message to send to Bob, Alice takes Bob's public key ($B = P^{*,b}$) and first establishes a shared secret key with Bob using the ECDH (Elliptic-Curve Diffie Hellman method), which is basically DH Key Exchange but using ECC.

Alice sends $A = P^{*,a}$ to Bob, and Bob calculates $P^{*,ab}$ by using the * operation on A b times (A * A * … *A).

Alice calculates $P^{*,ab}$ by using the * operation on B a times (B * B * … *B).

Together, they will have a shared secret $P^{*,ab}$, which can be used to encrypt their message.

Problems with ECC

As ECC is a relatively field, it is unknown if such a scheme is secure. Prime numbers have been studied for thousands of years and no known algorithm to factorise primes exists. However, ECC has only been studied in the past 50 years, raising questions if enough has been done for ECC to be secure.

Despite that, ECC is still widely used today, especially in TLS/ SSL which is used in secure HTTPS browsing. Other known applications include Bitcoin and iMessage.

As with the DH Key Exchange, ECC's security is based on the ECC assumption. If the ECC assumption is broken, then ECC is not secure. Advances in computing technology (e.g. quantum computing) could also make it easier to break ECC and DH problems.