Scribe: Cryptography and Network Security (Class.35.B)

Shubham Mishra

12-Nov-2020

1 Fields of usage of ECC

Elliptic Curve Cryptography (ECC) is used analogously to other public-key cryptosystems, like RSA and El-Gamal. Hence we have:

- Elliptic key Diffie-Helman Key Exchange (ECDH)
- Elliptic key Digital Signature Algorithm (ECDSA)

along with the usual message encryption.

2 Generic Procedure of ECC

In any elliptic curve cryptosystem the following things are known in public:

- The curve equation: $y^2 = x^3 + ax + b$, more specifically the values a and b,
- A prime p such that we consider points in the curve that belong to $\mathbb{Z}_p \times \mathbb{Z}_p$,
- A base point $B \in \mathbb{Z}_p \times \mathbb{Z}_p$ on the curve such that it can generate the whole elliptic curve group,

The **private key** in this context is an integer x selected from the interval $[1, p-1]$.

The **public key**, $Q = xB$ where xB means *point addition* of the base point B to itself x times. Note that, as the elliptic curve Discrete Log problem (ECDLP) is believed to be hard, it is difficult to find x given Q and B.

2.1 Encryption

Suppose Alice wants to send a message to Bob.

Alice and Bob decide on a base point B.

For Alice, private key = a and public key, $P_A = aB$. For Bob, private key = b and public key, $P_B = bB$.

Alice takes plaintext message, M and encodes it onto a point P_M from the elliptic group.

From this point onwards, the process is similar to El Gamal Cryptosystem.

Alice chooses a **random** integer $k \in [1, p-1]$.

The ciphertext is also a point on the curve given by $P_C = [(kB), (P_M + kP_B)]$

Here the term kP_B works as a **blinding factor** and the term kB is used as a hint to decrypt the blinded message.

2.2 Decryption

Bob has access to its own private key b. He takes the x-coordiante (or 1^{st} element) of P_C and calculates its scalar product with b.

We have, $b(kB) = k(bB) = kP_B$ since the scalar multiplication is commutative in the scalars.

Now Bob computes $(P_M + kP_B) - b(kB) = P_M + kP_B - kP_B = P_M$.

Then using the same encoding scheme as Alice used, Bob gets back the original message M from P_M.

2.3 Encoding

Here we give an example of a type of encoding.

We need to map the plaintext alphabet onto the curve $y^2 = x^3 + ax + b$. For simplicity let us assume that the aplhabet consists of numbers 0 to 9 and English lowercase alphabets a to z (represented by numbers from 10 to 35). For example, 'b' will be encoded as $m = 11$.

Now, we shall declare a public parameter, $k = 20$.

Then for every character in the alphabet m we shall compute $x = mk + i$. We will put this value in the curve and aim to get an integral value of y in $\mathbb{Z}_p$. For that, we vary i from 1 to $k - 1$.

And thus m is encoded as (x, y).

The decoding scheme is simply, $m = \lfloor \frac{x-1}{k} \rfloor$.

Lemma: A valid integral value for y will always exist with high probability.

Proof: Notice that we are actually solving a quadratic residue problem. We know that there are $\frac{p-1}{2}$ quadratic residues in $\mathbb{Z}_p$ and they are uniformly distributed. So the probability of finding a solution is $\frac{1}{2}$. So the probability that a solution will not exist in the first $k-1$ numbers is $(\frac{1}{2})^{k-1}$ which can be made arbitrarily small. Typically $k = 20$ is good enough. *(Q.E.D)*

These encoding schemes are fixed by standardising bodies like FIPS.

2.4 Elliptic Curve Diffie-Helman Key Exchange

The key exchange scheme is analogous to that used with RSA or El-Gamal.

Suppose Alice and Bob wants to send a message. The base point B is agreed upon.

Alice sends to Bob, aB where a is the private key of Alice. Bob sends to Alice, bB where b is the private key of Bob.

Alice and Bob can now both calculate abB as they have the knowledge of a and b respectively, beforehand. But no Evesdropper can find abB solely from the information of aB and bB since Elliptic Curve Discrete Log Problem (ECDLP) is believed to be hard.

3 Why ECC is used?

The strength of a cryptosystem lies on the hardness of the underlying problem. ECDLP is much more difficult than normal Discrete Log Problem. This result in shorter key sizes.

3.1 A comparison of key sizes

The below table gives the NIST Recommended Security Bit levels of RSA and ECC. [1]

Security Bit Level	RSA	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

3.2 How hard is ECDLP?

Consider 2 lists generated by choosing random integers: $j_1, j_2, ..., j_r$ and $k_1, k_2, ..., k_r$ where the numbers are between 1 and p.

Now let us consider two points on the curve P and Q. Now consider the lists:

List L_1: $j_1P, j_2P, ..., j_rP$

List L_2: $k_1P + Q, k_2P + Q, ..., k_rP + Q$

Any collision between the 2 lists will imply $j_u P = k_v P + Q$ for some u and v. Thus $Q = (j_u - k_v)P$

If $r = O(p^{1/2})$, by Birthday paradox, we have a high probability of finding a collision.

The fastest known algorithm for ECDLP takes $O(p^{1/2})$ time.[2] However, for normal DLP, it is $O(p^{1/4})$.

So ECDLP is harder than DLP in F_p.

3.3 Applications of ECC

ECC, due to its simplicity and smaller key sizes, is becoming increasingly popular in the fields of IoT and other low energy devices.

4 Conclusion

In today's lecture, we learned about ECC and its various applications.

References

[1] Dilip Kumar Yadav Dindayal Mahto. Rsa and ecc: A comparative analysis, 2017. URL https://www.ripublication.com/ijaer17/ijaerv12n19_140.pdf. [Online; accessed 17-November-2020].

[2] Steven Galbraith. Algorithms for the ecdlp, 2015. URL https://www.math.u-bordeaux.fr/~aenge/ecc2015/documents/galbraith.pdf. [Online; accessed 17-November-2020].