Public Key Cryptography: Elliptic Curve Cryptography (ECC) - Part 2

Dr. Ashok Kumar Das

IEEE Senior Member
Associate Professor
Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

E-mail: *ashok.das@iiit.ac.in*
URL: `http://www.iiit.ac.in/people/faculty/ashokkdas`
`https://sites.google.com/view/iitkgpakdas/`

Elliptic Curve Cryptography Key Exchange Protocol (ECC Key Exchange)

Elliptic Curve Cryptography Key Exchange Protocol (ECC Key Exchange)

- Pick a large integer q, where $q = p$; p being a prime, or $q = 2^m$, for some positive integer m, and the elliptic curve parameters a ansd b for the elliptic curves:

$$\begin{aligned} y^2 &= x^3 + ax + b \pmod{p} \text{ in } GF(p); \\ y^2 + xy &= x^3 + ax^2 + b \pmod{p} \text{ in } GF(2^m). \end{aligned}$$

- Pick a base point $G = (x, y)$ in $E_q(a, b)$ whose order is a very large value n, that is, $nG = \mathcal{O}$.
- $E_q(a, b)$ and G are parameters of the cryptosystem known to all participants.

Elliptic Curve Cryptography Key Exchange Protocol (ECC Key Exchange)

A key exchange between two users A and B can be accomplished as follows:

- A selects an integer n_A, where $n_A < n$. A's private key is n_A.
  A generates a public key $P_A = n_A G$; the public key is a point in $E_q(a, b)$.
- B similarly selects a private key n_B, where $n_B < n$. B's private key is n_B.
  B generates a public key $P_B = n_B G$.
- A generates the secret key $K_{A,B} = n_A P_B$.
- B generates the secret key $K_{B,A} = n_B P_A$.

ECC Key Exchange Protocol (continued...)

Summary

User A	User B
1. Select private n_A	
2. Calculate public P_A	
3. $P_A = n_A G \longrightarrow$	
	1. Select private n_B
	2. Calculate public P_B
	3. $\longleftarrow P_B = n_B G$
4. $K_{A,B} = n_A P_B$	4. $K_{B,A} = n_B P_A$

ECC Key Exchange Protocol (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Correctness Proof

$$
\begin{aligned}
K_{A,B} &= n_A \times P_B \text{ [User A]} \\
&= n_A \times (n_B \times G) \\
&= (n_A \times n_B) \times G \\
&= (n_B \times n_A) \times G \\
&= n_B \times (n_A \times G) \\
&= n_B \times P_A \\
&= K_{B,A} \text{ [User B]}
\end{aligned}
$$

ECC Key Exchange Protocol (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Problem [ECC Key Exchange]

Users A and B use the ECC key exchange technique with a common prime $q = 211$ and an elliptic curve $E_q(a, b)$, where $a = 0$ and $b = -4$. Let $G = (2, 2)$ a base point on $E_q(a, b)$.

(a) If user A has private key $n_A = 121$, what is the A's public key P_A?

Solution: $P_A = n_A.G = 121.(2, 2) = (115, 48)$.

(b) If user B has private key $n_B = 203$, what is the B's public key P_B?

Solution: $P_B = n_B.G = 203.(2, 2) = (130, 203)$.

(c) What is the secret shared key?

Solution: $K_{A,B} = n_A.P_B = 121.(185, 178) = (161, 69)$, by user A.

$K_{B,A} = n_B.P_A = 203.(67, 106) = (161, 69)$, by user B.

ECC Key Exchange Protocol (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Man-in-the-middle attack on ECC key exchange protocol???

ECC Key Exchange Protocol (continued...)

Man-in-the-middle attack on ECC key exchange protocol

User A	Adversary $\mathcal{C}$	User B
1. Select private n_A		
2. Calculate public P_A	**Intercept & block P_A**	
3. $P_A = n_A G \longrightarrow$	1. Select private n_C	
	2. Calculate public P_C	
	3. $P_C = n_C G \longrightarrow$	
	$\longleftarrow P_C = n_C G$	
		1. Select private n_B
		2. Calculate public P_B
	Intercept & block P_B	3. $\longleftarrow P_B = n_B G$
	4. $K_1 = n_C P_A$	
4. $K_1 = n_A P_C$	$K_2 = n_C P_B$	4. $K_2 = n_B P_C$

Elliptic Curve Encryption/Decryption

Elliptic Curve Encryption/Decryption

ECC Encryption

- The first task in this system is to encode the plaintext message m to be sent as an x-y point P_m in $E_q(a, b)$. (For example, Koblitz method (Available at `http://zoo.cs.yale.edu/classes/cs467/2012s/lectures/ln13.pdf`)).
- It is the point P_m that will be encrypted as a ciphertext and subsequently decrypted.
- As with the ECC key exchange system, an encryption/decryption system requires a base point G and an elliptic curve $E_q(a, b)$.
- Let user A's private-public key pair $(KR_a, KU_a) = (n_A, P_A)$ and user B's private-public key pair $(KR_b, KU_b) = (n_B, P_B)$

Elliptic Curve Encryption/Decryption

ECC Encryption

- To encrypt and send a plaintext message (encoded) P_m to user B, user A proceeds as follows:
  - A chooses a random positive integer k.
  - A produces the ciphertext C_m consisting of the pair of points

$$\begin{aligned} C_m &= E_{PU_b}(P_m) \\ &= E_{P_B}(P_m) \\ &= \{C_1, C_2\} \\ &= \{kG, P_m + kP_B\}, \end{aligned}$$

where $P_B = n_B G$.

Elliptic Curve Encryption/Decryption

ECC Decryption

- To decrypt the ciphertext C_m, user B proceeds as follows:
  - B uses its own private key $KR_b = n_B$.
  - The plaintext P_m is recovered as

$$\begin{aligned} P_m &= D_{PR_b}(C_m) \\ &= D_{n_B}(C_m) \\ &= C_2 - n_B C_1 \\ &= P_m + kP_B - n_B(kG) \\ &= P_m + k(n_B G) - n_B(kG) \\ &= P_m. \end{aligned}$$

ECC Encryption/Decryption (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Problem [ECC Encryption/Decryption]

Suppose two users A and B rely on the ECC key cryptosystem. An elliptic curve cryptosystem operates on the curve $y^2 = x^3 + ax + b \pmod{q}$ with the parameters $E_{11}(1, 6)$, and the base point $G = (2, 7)$. Assume that B's private key is $n_B = 7$.

(a) Find B's public key P_B.

(b) Determine the ciphertext C_m, if the user A sends the message $P_m = (10, 9)$ with the random value $k = 3$.

ECC Encryption/Decryption (Continued...)

Solution:

- (a) $P_B = n_B G = 7.(2,7) = (7,2)$.
- (b) $C_m = (C_1, C_2)$, where

$$\begin{aligned} C_1 &= kG \\ &= 3.(2,7) \\ &= (8,3), \end{aligned}$$

and

$$\begin{aligned} C_2 &= P_m + kP_B \\ &= (10,9) + 3.(7,2) \\ &= (10,9) + (3,5) \\ &= (10,2). \end{aligned}$$

ECC Encryption/Decryption (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Online Demo on ECC Encryption/Decryption

- Generating private/public keys pair for User A (Alice) and User B (Bob)
- Encrypting a message
- Decryping a message

`https://8gwifi.org/ecfunctions.jsp`

Elliptic Curve Digital Signature Algorithm (ECDSA)

Digital Signatures

Signature Schemes

- A *signature scheme* is a five-tuple $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, where the following conditions are satisfied:
- 1. $\mathcal{P}$ is a finite set of possible messages;
- 2. $\mathcal{A}$ is a finite set of possible signatures;
- 3. $\mathcal{K}$, the key space, is a finite set of possible keys;
- 4. For each $k \in \mathcal{K}$, there is a signing algorithm $sig_k \in \mathcal{S}$ and a corresponding verification algorithm $ver_k \in \mathcal{V}$. Each $sig_k : \mathcal{P} \to \mathcal{A}$ and $ver_k : \mathcal{P} \times \mathcal{A} \to \{true, false\}$ are functions such that the following equation is satisfied for every message $x \in \mathcal{P}$ and for every signature $y \in \mathcal{A}$:
  $ver_k(x, y) =$ true, if $y = sig_k(x)$,
  $ver_k(x, y) =$ false, if $y \neq sig_k(x)$.
- The pair (x, y) with $x \in \mathcal{P}$ and $y \in \mathcal{A}$ is called a *signed message*.

Elliptic Curve Digital Signature Algorithm

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Key generation

The domain parameters for the Elliptic Curve Digital Signature Algorithm (ECDSA) consist of a suitably chosen elliptic cure $E_p(a, b)$ defined over a finite field $GF(p)$, and a base point $G \in E_p(a, b)$ with order n. Each entity $\mathcal{A}$ does the following:

- Step 1. Select a random or pseudorandom integer k in the interval $[1, n-1]$.
- Step 2. Compute $Q = kG$.
- Step 3. $\mathcal{A}$'s public key is Q; $\mathcal{A}$'s private key is k.

Elliptic Curve Digital Signature Algorithm

Signature generation

To sign a message, say m, an entity $\mathcal{A}$ with domain parameters $D = (p, n, Q, G, E_p(a, b), h(\cdot))$, where $h(\cdot)$ is a secure hash function and associated key pair (k, Q) does the following steps:

- Step 1. Select a random or pseudorandom integer l, with $1 \leq l \leq n - 1$.
- Step 2. Compute $lG = (x_1, y_1)$ and $r = x_1 \mod n$. If $r = 0$ then go to step 1.
- Step 3. Compute $e = h(m)$ and $s = l^{-1}(e + kr) \mod n$. If $s = 0$ then go to step 1.
- Step 4. $\mathcal{A}$'s signature for the message m is (r, s).

Elliptic Curve Digital Signature Algorithm

Signature verification

In order to verify $\mathcal{A}$'s signature (r, s) on m, the verifier $\mathcal{B}$ obtains an authentic copy of $\mathcal{A}$'s domain parameters D and associated public key Q. Then $\mathcal{B}$ does the following steps:

- Step 1. Verify that r and s are integers in the interval $[1, n-1]$.
- Step 2. Compute $e = h(m)$.
- Step 3. Compute $w = s^{-1} \mod n$, $u_1 = ew \mod n$, $u_2 = rw \mod n$ and $X = u_1 G + u_2 Q$. If $X = \mathcal{O}$, then reject the signature. Otherwise, compute $v = x_1 \mod n$, where $X = (x_1, y_1)$.
- Step 4. Accept the signature if and only if $v = r$.

Elliptic Curve Digital Signature Algorithm

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Online Demo on ECDSA

- Generating EC keypair
- Signing and verifying ECDSA signature

`https://kjur.github.io/jsrsasign/sample/sample-ecdsa.html`

Elliptic Curve Cryptography (ECC)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Table: Comparison of security strengths

Security strength (in bits)	Symmetric-key algorithm	IFC (e.g. RSA) (in bits)	ECC (e.g. ECDSA) (in bits)
≤ 80	2TDEA	$k = 1024$	$f = 160$-223
112	3TDEA	$k = 2048$	$f = 224$-255
128	AES-128	$k = 3072$	$f = 256$-383
192	AES-192	$k = 7680$	$f = 384$-511
256	AES-256	$k = 15360$	$f = 512+$

Note: 2TDEA (2DES): Two-key Triple Data Encryption Algorithm; 3TDEA (3DES): Three-key Triple Data Encryption Algorithm; AES-128, AES-192, AES-256: Advanced Encryption Standard algorithms with key sizes 128 bits, 192 bits, 256 bits, respectively; IFC: Integer Factorization Cryptography; ECC: Elliptic Curve Cryptography; ECDSA: Elliptic Curve Digital Signature Algorithm; k: the size of the modulus for integer factorization algorithms; f: the size (order) of the base point for an elliptic curve.

Elliptic Curve Cryptography (ECC)

Security of ECC

- The security of ECC depends on how difficult it to determine the discrete logarithm k given kP and P. This is referred to as the "elliptic curve discrete logarithm problem (ECDLP)". The fasted known technique for taking the elliptic curve logarithm is known as the Pollard rho method, which is exponential in time complexity.
- Considerably smaller key size can be used for ECC compared to RSA (for example, the security of 160-bit ECC is equivalent to that for 1024-bit RSA).
- There is a computational advantage to using ECC with a shorter key length than a comparably secure RSA.

Elliptic Curve Cryptography (ECC)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Comparison of security issues between RSA and ECC

RSA	ECC
Hard problem: $n = p \times q$, finding $d \equiv e^{-1} \pmod{\phi(n)}$ is a hard problem given n and e.	Hard problem: $Q = kP$, where P and $Q \in E_p(a, b)$ are known, to compute k is a hard problem.