Elliptic Curve Cryptosystems

Elliptic Curve - General Equation

Set of solutions \((x, y)\) to the equation

\[
y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6
\]

where \(x, y \in K\)

\(a_1, a_2, a_3, a_4, a_6 \in K\)

Values of \(a_i\) limited by constraints specific to the field \(K\)

\(K\) is a field

+ a special point called \(the \ point \ at \ infinity \ \mathbf{O}\)

Three Classes of Elliptic Curves

Elliptic curves built over

- \(K = GF(p)\)
- \(K = GF(2^m)\)

Arithmetic operations present in many libraries

Polynomial basis representation

Normal basis representation

Fast in hardware

Compact in hardware
Elliptic Curve over GF(p)

Set of solutions \((x, y)\) to the equation

\[ y^2 = x^3 + ax + b \]

where \(x, y \in \text{GF}(p)\)

\(a, b \in \text{GF}(p)\) \(\quad 4a^3 + 27b^2 \neq 0 \pmod{p} \)

+ a special point called the point at infinity \(O\)

Example: Elliptic curve \(y^2 = x^3 + x + 1\) over GF(23)

\[
\begin{align*}
(0, 1) & \quad (6, 4) & \quad (12, 19) \\
(0, 22) & \quad (6, 19) & \quad (13, 7) \\
(1, 7) & \quad (7, 11) & \quad (13, 16) \\
(1, 16) & \quad (7, 12) & \quad (17, 3) \\
(3, 10) & \quad (9, 7) & \quad (17, 20) \\
(3, 13) & \quad (9, 16) & \quad (18, 3) \\
(4, 0) & \quad (11, 3) & \quad (18, 20) \\
(5, 4) & \quad (11, 20) & \quad (19, 5) \\
(5, 19) & \quad (12, 4) & \quad (19, 18) \\
O & & \quad \text{28 points}
\end{align*}
\]

Generating a point of an elliptic curve (1)

1. Choose \(x\)
   e.g., \(x = 3\)

2. Compute \(z = y^2 = x^3 + ax + b\)
   e.g., \(z = 3^3 + 1 \cdot 3 + 1 \pmod{23} = 8\)

3. If \(z = 0\), then \(y=0\) and there is only one point, \((x,0)\), with the given \(x\) coordinate
Generating a point of an elliptic curve (2)

Otherwise

4. Verify whether there exists \( y \) such that \( z = y^2 \pmod{p} \) using Euler’s criterion, i.e., check whether
\[ e^{y^{p-1}/2} = 1 \pmod{p} \]
(if this is the case \( z \) is called a quadratic residue mod \( p \))

e.g., \( 8^{(23-1)/2} \pmod{23} = 8^{11} \pmod{23} = (8^8 \pmod{23})(8^2 \pmod{23})(8^1 \pmod{23}) \pmod{23} = 4 \cdot 18 \cdot 8 \pmod{23} = 1 \)

If Euler’s criterion is not met (i.e., \( e^{y^{p-1}/2} \neq 1 \pmod{p} \)), then there is no point of the given elliptic curve with the given \( x \) coordinate.

Generating a point of an elliptic curve (3)

Otherwise

5. If Euler’s criterion is met, then there are two points with a given \( x \) coordinate
\((x, y_1)\) and \((x, y_2)\)

If \( p \equiv 3 \pmod{4} \) then
\[ y_1 \text{ and } y_2 \text{ can be computed from the equation} \]
\[ y_1 = e^{x^{p+1}/4} \pmod{p} \]
\[ y_2 = e^{x^{p+1}/4} \pmod{p} = p - e^{x^{p+1}/4} \pmod{p} = \]

E.g., \( 23 \equiv 3 \pmod{4} \)
\[ y_1 = 8^{23+1}/4 \pmod{23} = 8^8 \pmod{23} = 13 \]
\[ y_2 = -13 = 23 - 13 = 10 \]

Addition of two points on the elliptic curve over \( \text{GF}(p) \) (1)

\[ P = (x_1, y_1) \quad Q = (x_2, y_2) \]
\[ R = P + Q = (x_3, y_3) \]

Case 1:
\[ P + O = O + P = P \]

Case 2:
\[ x_3 = x_1 \text{ and } y_3 = -y_1 \]
\[ P + Q = O \]
\[ Q = -P \]
Addition of two points on the elliptic curve over GF(p) \hspace{1cm} (2)

Case 3:
\[ x_3 = \lambda^2 - x_1 - x_2 \]
\[ y_3 = \lambda (x_1 - x_3) - y_1 \]

where

Case 3a: if $P \neq Q$
\[ \lambda = \frac{y_2 - y_1}{x_2 - x_1} = (y_2 - y_1)(x_2 - x_1)^{-1} \]

Case 3b: if $P = Q$
\[ \lambda = \frac{3x_1^2 + a}{2y_1} = (3x_1^2 + a)(2y_1)^{-1} \]

Example: Addition of points on the elliptic curve $y^2 = x^3 + x + 6$ over GF(11)

$P = (2, 7)$

$2P = P + P = (2, 7) + (2, 7)$

$\lambda = (3 \cdot 2^2 + 1)(2 \cdot 7)^{-1} \mod 11 = 2 \cdot 3^4 \mod 11 = 2 \cdot 4 \mod 11 = 8$

$x_3 = 8^2 - 2 - 2 \mod 11 = 9 - 2 - 2 \mod 11 = 5$

$y_3 = 8(2 - 5) - 7 \mod 11 = 9 - 7 \mod 11 = 2$

$2P = (5, 2)$

Example: Addition of points on the elliptic curve $y^2 = x^3 + x + 6$ over GF(11)

$P = (2, 7)$ \hspace{1cm} 2P = (5, 2)$

$3P = P + 2P = (2, 7) + (5, 2)$

$\lambda = (2 - 7)(5 - 2)^{-1} \mod 11 = 6 \cdot 3 \mod 11 = 6 \cdot 4 \mod 11 = 2$

$x_5 = 2^2 - 2 - 5 \mod 11 = 4 - 2 - 5 \mod 11 = 8$

$y_5 = 2(2 - 8) - 7 \mod 11 = 10 - 7 \mod 11 = 3$

$3P = (8, 3)$
Scalar multiples of P

<table>
<thead>
<tr>
<th>P</th>
<th>7P</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2,7)</td>
<td>(7,2)</td>
</tr>
<tr>
<td>(5,2)</td>
<td>(3,5)</td>
</tr>
<tr>
<td>(8,3)</td>
<td>(10,9)</td>
</tr>
<tr>
<td>(10,2)</td>
<td>(8,8)</td>
</tr>
<tr>
<td>(3,6)</td>
<td>(5,9)</td>
</tr>
<tr>
<td>(7,9)</td>
<td>(2,4)</td>
</tr>
<tr>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

Number of points on the curve = 13

P is a generator of the group of points on the elliptic curve.

Number of points on the curve $\# E(GF(p))$

- order of an elliptic curve
- cardinality of an elliptic curve

Hasse’s Theorem

$$ p + 1 - 2\sqrt{p} \leq \# E(GF(p)) \leq p + 1 + 2\sqrt{p} $$

e.g.,

order of a curve over GF(11)

$$ 11 + 1 - 2\sqrt{11} \leq \# E(GF(11)) \leq 11 + 1 + 2\sqrt{11} $$

$$ 5.7 \leq \# E(GF(11)) \leq 18.63 $$

order of the curve $y^2 = x^3 + x + 6$ over GF(11) = 13

Number of points on the curve $\# E(GF(p))$

Exact number $\# E(GF(p))$ can be computed using

School’s algorithm

Complexity: $(\log p)^8$

To prevent the Pohlig-Hellman method of computing elliptic curve discrete logarithm:

$\# E(GF(p))$ must have a large prime divisor

“Large” currently means $\sim 10^{40}$
Exponentiation: \[ y = a^e \mod n \]

Right-to-left binary exponentiation

\[
e = (e_{L-1}, e_{L-2}, \ldots, e_1, e_0)_2
\]

\[
y = 1; \\
s = a, \\
\text{for } i = 0 \text{ to } L-1 \\
\{
\text{if } (e_i = 1) \\
y = y \cdot s \mod n; \\
s = s^2 \mod n;
\}
\]

Left-to-right binary exponentiation

\[
y = 1; \\
\text{for } i = L-1 \text{ down to } 0 \\
\{
\text{if } (e_i = 1) \\
y = y \cdot a \mod n;
\}
\]

Scalar Multiplication: \[ Y = k \cdot P \]

Right-to-left binary scalar multiplication

\[
k = (k_{L-1}, k_{L-2}, \ldots, k_1, k_0)_2
\]

\[
Y = O; \\
S = P; \\
\text{for } i = 0 \text{ to } L-1 \\
\{
\text{if } (k_i = 1) \\
Y = Y + S; \\
S = 2S;
\}
\]

Left-to-right binary scalar multiplication

\[
Y = O; \\
\text{for } i = L-1 \text{ down to } 0 \\
\{
Y = 2Y; \\
\text{if } (k_i = 1) \\
Y = Y + P;
\}
\]

Diffie-Hellman

Alice \[ g \cdot \text{generator of } Z_p^* \]

A’s private key: \( x_A \)

A’s public key: \( y_A = g^{x_A} \)

Bob

B’s private key: \( x_B \)

B’s public key: \( y_B = g^{x_B} \)

Secret derivation

\[
z_{AB} = y_B^{x_A} = g^{x_B x_A} \\
\]

Secret derivation

\[
z_{BA} = y_A^{x_B} = g^{x_A x_B} \\
\]
**Elliptic Curve Diffie-Hellman**

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>A’s private key: $x_A$</td>
<td>B’s private key: $x_B$</td>
</tr>
<tr>
<td>A’s public key: $Q_A = x_A P$</td>
<td>B’s public key: $Q_B = x_B P$</td>
</tr>
</tbody>
</table>

Secret derivation

$Z_{AB} = x_A Q_B = x_A (x_B P)$

$Z_{BA} = x_B Q_A = x_B (x_A P)$

---

**Digital Signature Algorithm**

**System parameters**

- May be shared by a group of users or belong to a single user; known to everybody

- $q$ - 160-bit prime
- $p$ - $L$-bit prime, such that $q | p-1$
  where $L = 1024 + 64k$

- $g = h^{(p-1)/q} \mod p$
  where $1 < h < p-1$, such that $g > 1$

  From Fermat’s theorem
  
  $g^q \mod p = h^{p-1} \mod p = 1$

  $g$ - generator of the cyclic group of order $q$
  in $\mathbb{Z}_p^*$

---

**Elliptic Curve Digital Signature Algorithm ECDSA**

**System parameters**

- May be shared by a group of users or belong to a single user; known to everybody

- $E$ - elliptic curve over $\mathbb{F}(p)$ or $\mathbb{F}(2^m)$

- $P$ - point of order $q$ on the elliptic curve $E$
**Digital Signature Algorithm**

*Public and private key*

**Private key**

\[ x \text{ - arbitrary 160 bit number} \quad 0 < x < q \]

**Public key**

\[ y = g^x \mod p \quad 0 < y < p \]

\[ L \text{ - bit number} \]

---

**Elliptic Curve Digital Signature Algorithm**

*Public and private key*

**Private key**

\[ x \text{ - arbitrary number} \quad 0 < x < q \]

**Public key**

\[ Y = xP \]

---

**DSA: Signature generation**

1. Choose random message private key \( 1 < k < q \) (secret, different for each message)

2. Compute message public key \( r = (g^k \mod p) \mod q \)

3. Compute hash value

4. Compute \( x = k^{-1} \cdot (\text{SHA}(M) + x \cdot r) \mod q \)

\[ \text{SGN}(M) = r \| x \text{  160 bit 160 bit 40 bytes} \]
ECDSA: Signature generation

1. Choose random message private key $1 < k < q$
   (secret, different for each message)

2. Compute message public key
   \[ R = kP \]
   \[ r : x\text{-coordinate of } R \]

4. Compute
   \[ s = k^{-1} (\text{SHA}(M) + r) \mod q \]
   \[ \text{SGN}(M) = r \| s \]

DSA: Signature verification

1. Compute hash value
   \[ [\text{SGN}(M)]' \]

2. Compute
   \[ w = (s')^{-1} \mod q \]

3. Compute
   \[ u_1 = \text{SHA}(M') w \mod q \]

4. Compute
   \[ u_2 = r' w \mod q \]

5. Compute
   \[ v = ((u_1)^e - u_2^e) \mod p \mod q \]

6. Compare
   \[ \text{Signature invalid} \quad \text{N} \quad \text{v} \quad \text{Y} \quad \text{Signature valid} \]

ECDSA: Signature verification

1. Compute hash value
   \[ [\text{SGN}(M)]' \]

2. Compute
   \[ w = (s')^{-1} \mod q \]

3. Compute
   \[ u_1 = \text{SHA}(M') w \mod q \]

4. Compute
   \[ u_2 = r' w \mod q \]

5. Compute
   \[ v = u_1 P + u_2 Y \]
   \[ v \text{ is the x-coordinate of } V \]

6. Compare
   \[ \text{Signature invalid} \quad \text{N} \quad \text{v} \quad \text{Y} \quad \text{Signature valid} \]
El-Gamal Encryption

System parameters
May be shared by a group of users or belong to a single user; known to everybody

- \( p \) - prime
- \( g \) - generator of the group \( \mathbb{Z}_p^* \)

Elliptic Curve El-Gamal Encryption

System parameters
May be shared by a group of users or belong to a single user; known to everybody

- \( E \) - elliptic curve over \( \text{GF}(p) \) or \( \text{GF}(2^m) \)
- \( \mathbf{P} \) - generator of the group of points on the elliptic curve

El-Gamal Encryption

Public and private key

Private key

- \( x \) - arbitrary number \( 1 \leq x \leq p-2 \)

Public key

- \( y = g^x \mod p \) \( 0 < y < p \)
Elliptic Curve El-Gamal Encryption

Public and private key

Private key

- \( x \) - arbitrary number, \( 1 \leq x \leq \#E(GF(q)) - 1 \)

Public key

\( Y = xP \)

---

El-Gamal: Encryption

1. Choose random \( m \) - message private key, \( 1 \leq k \leq p - 2 \), relatively prime with \( p \) - 1 (secret, different for each message)

2. Compute

- \( r = g^k \mod p \)

3. Compute

- \( c = y^k \cdot M \mod p \)

\( C(M) = r \parallel c \)

---

Elliptic Curve El-Gamal: Encryption

1. Choose random \( m \) - message private key, \( 1 \leq k \leq \#E(GF(q)) - 1 \), (secret, different for each message)

2. Compute

- \( R = kP \)

3. Compute

- \( M = (m, n) \)

- \( n \) - y-coordinate corresponding to the x-coordinate \( m \)

\( C(m) = R \parallel C \)
El-Gamal: Decryption

\[ r \cdot C \quad C(M) \]

\[ M = c \cdot (r^x)^{-1} \mod p \]

Justification:

\[ c \cdot (r^x)^{-1} \mod p = y^k \cdot M \cdot ((g^k)^x)^{-1} = y^k \cdot M \cdot (y^x)^{-1} = \]
\[ = y^k \cdot M \cdot (y^x)^{-1} = M \]

Elliptic Curve El-Gamal: Decryption

\[ R \quad C \quad C(m) \]

\[ M = C \cdot x R \]

m: x-coordinate of M

Justification:

\[ C \cdot x R = (k \cdot Y + M) \cdot x R = (k \cdot Y + M) \cdot x \cdot k \cdot P = \]
\[ = (k \cdot Y + M) \cdot (x \cdot P) = k \cdot Y + M \cdot x \cdot Y = M \]

Menezes-Vanstone Elliptic Curve Cryptosystem

System parameters

May be shared by a group of users or belong to a single user;
known to everybody

E - elliptic curve over GF(p) or GF(2^m)

P - generator of the group of points
on the elliptic curve
Menezes-Vanstone Elliptic Curve Cryptosystem

**Public and private key**

**Private key**

- $x$ - arbitrary number
  
  $1 \leq x \leq \#\text{E}(\text{GF}(q)) - 1$

**Public key**

- $Y = xP$

---

Menezes-Vanstone Cryptosystem: Encryption

1. Choose random message private key $1 \leq k \leq \#\text{E}(\text{GF}(q)) - 1$, (secret, different for each message)

2. Compute message public key $R = kP$

3. Form message block: $(m_1, m_2)$

4. Compute

   $C = kY = (c_1, c_2)$

5. Compute

   $y_1 = c_1 m_1$
   
   $y_2 = c_2 m_2$

   $C(m_1, m_2) = R || y_1 y_2$

---

Menezes Vanstone Cryptosystem : Decryption

<table>
<thead>
<tr>
<th>R</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$C(m_1, m_2)$</th>
</tr>
</thead>
</table>

$C = xR = (c_1, c_2)$

$m_1 = c_1^{-1} y_1$

$m_2 = c_2^{-1} y_2$

**Justification:**

$xR = xkP = k(xP) = kY = C$