Elliptic Curve Cryptosystems

Elliptic Curve - General Equation

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

\[
y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6
\]

where \(x, y \in K\)

\(a_1, a_2, a_3, a_4, a_5, a_6 \in K\) \hspace{1cm} \text{Values of } a_i \text{ limited by constraints specific to the field } K

\(K\) is a field

\(+ a\text{ a special point called } the\ point\ at\ infinity\ \mathcal{O}\)
Three Classes of Elliptic Curves

Elliptic curves built over

\[ K = \text{GF}(p) \]

\[ K = \text{GF}(2^m) \]

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 \textit{the point at infinity} \(O\)
Example: Elliptic curve $y^2 = x^3 + x + 1$ over GF(23)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 1)</td>
<td>(6, 4)</td>
<td>(12, 19)</td>
</tr>
<tr>
<td>(0, 22)</td>
<td>(6, 19)</td>
<td>(13, 7)</td>
</tr>
<tr>
<td>(1, 7)</td>
<td>(7, 11)</td>
<td>(13, 16)</td>
</tr>
<tr>
<td>(1, 16)</td>
<td>(7, 12)</td>
<td>(17, 3)</td>
</tr>
<tr>
<td>(3, 10)</td>
<td>(9, 7)</td>
<td>(17, 20)</td>
</tr>
<tr>
<td>(3, 13)</td>
<td>(9, 16)</td>
<td>(18, 3)</td>
</tr>
<tr>
<td>(4, 0)</td>
<td>(11, 3)</td>
<td>(18, 20)</td>
</tr>
<tr>
<td>(5, 4)</td>
<td>(11, 20)</td>
<td>(19, 5)</td>
</tr>
<tr>
<td>(5, 19)</td>
<td>(12, 4)</td>
<td>(19, 18)</td>
</tr>
</tbody>
</table>

28 points

Generating a point of an elliptic curve (1)

1. Choose $x$
eq 3

2. Compute $z = y^2 = x^3 + ax + b$
eq 3 + 1.3 + 1 (mod 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
   \[ z^{(p-1)/2} = 1 \pmod{p} \]
   (if this is the case \( z \) is called a quadratic residue mod \( p \))
   e.g., \( 8^{(23-1)/2} \pmod{23} = 8^{11} \pmod{23} = 8 \pmod{23} = 8 \)
   \[ \equiv 1 \pmod{23} \]

If Euler’s criterion is not met (i.e., \( z^{(p-1)/2} \neq 1 \pmod{p} \),
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
   \( y_1 \) and \( y_2 \) can be computed from the equation
   \[ y_1 = \pm z^{(p+1)/4} \pmod{p} \]
   \[ y_2 = -z^{(p+1)/4} \pmod{p} \equiv p - z^{(p+1)/4} \pmod{p} = p - y_1 \]
   E.g., \( 23 \equiv 3 \pmod{4} \)
   \[ y_1 = 8^{(23+1)/4} \pmod{23} = 8^6 \pmod{23} = 13 \]
   \[ y_2 = -13 \equiv 23 - 13 = 10 \]
Addition of two points on the elliptic curve over GF(p) \hspace{1cm} (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:
\[ x_2 = x_1 \quad \text{and} \quad y_2 = -y_1 \]
\[ P + Q = O \]
\[ Q = -P \]

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: \hspace{1cm} \text{if} \hspace{0.5cm} P \neq Q
\[ \lambda = \frac{y_2 - y_1}{x_2 - x_1} = (y_2 - y_1) \frac{1}{(x_2 - x_1)} \]

Case 3b: \hspace{1cm} \text{if} \hspace{0.5cm} P = Q
\[ \lambda = \frac{3x_1^2 + a}{2y_1} = (3x_1^2 + a) \frac{1}{(2y_1)} \]
Example: Addition of points on the elliptic curve
\[ y^2 = x^3 + x + 6 \] over GF(11)

\[ P = (2, 7) \]

\[ 2P = P + P = (2, 7) + (2, 7) \]

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

\[ x_3 = 8^2 - 2 - 2 \mod 11 = 9 - 2 - 2 \mod 11 = 5 \]
\[ y_3 = 8 \cdot (2 - 5) - 7 \mod 11 = 9 - 7 \mod 11 = 2 \]

\[ 2P = (5, 2) \]

---

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

\[ P = (2, 7) \quad 2P = (5, 2) \]

\[ 3P = P + 2P = (2, 7) + (5, 2) \]

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

\[ x_3 = 2^2 - 2 - 5 \mod 11 = 4 - 2 - 5 \mod 11 = 8 \]
\[ y_3 = 2 \cdot (2 - 8) - 7 \mod 11 = 10 - 7 \mod 11 = 3 \]

\[ 3P = (8, 3) \]
Scalar multiples of P

\[
\begin{align*}
P &= (2, 7) & 7P &= (7, 2) \\
2P &= (5, 2) & 8P &= (3, 5) \\
3P &= (8, 3) & 9P &= (10, 9) \\
4P &= (10, 2) & 10P &= (8, 8) \\
5P &= (3, 6) & 11P &= (5, 9) \\
6P &= (7, 9) & 12P &= (2, 4) \\
7P &= (7, 2) & 13P &= \mathcal{O}
\end{align*}
\]

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 \( \#E(\text{GF}(p)) \)

= order of an elliptic curve

= cardinality of an elliptic curve

Hasse’s Theorem

\[
p + 1 - 2\sqrt{p} \leq \#E(\text{GF}(p)) \leq p + 1 + 2\sqrt{p}
\]

e.g.,

order of a curve over GF(11)

\[
11 + 1 - 2\sqrt{11} \leq \#E(\text{GF}(11)) \leq 11 + 1 + 2\sqrt{11}
\]

\[
5.37 \leq \#E(\text{GF}(11)) \leq 18.63
\]

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 Schoof’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

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

\[
\begin{align*}
y &= 1; \\
s &= a; \\
&\text{for } i=0 \text{ to } L-1 \\
&\quad \{ \\
&\quad \quad \text{if } (e_i = 1) \\
&\quad \quad \quad y = y \cdot s \mod n; \\
&\quad \quad \quad s = s^2 \mod n; \\
&\quad \quad \} \\
\end{align*}
\]

Left-to-right binary exponentiation

\[
\begin{align*}
y &= 1; \\
&\text{for } i=L-1 \text{ downto } 0 \\
&\quad \{ \\
&\quad \quad y = y^2 \mod n; \\
&\quad \quad \text{if } (e_i = 1) \\
&\quad \quad \quad y = y \cdot a \mod n; \\
&\quad \quad \} \\
\end{align*}
\]
Scalar Multiplication:  \[ Y = k \cdot P \]

Right-to-left binary scalar multiplication

Left-to-right binary scalar multiplication

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

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

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

Diffie-Hellman

Alice

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

A’s private key: \( x_A \)

A’s public key:

\[ y_A = g^{x_A} \]

Secret derivation

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

Bob

B’s private key: \( x_B \)

B’s public key:

\[ y_B = g^{x_B} \]

Secret derivation

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

**Alice**
- P - generator of E(GF(q))
- A’s private key: $x_A$
- A’s public key: $Q_A = x_A P$

**Bob**
- B’s private key: $x_B$
- B’s public key: $Q_B = x_B P$

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 $L = 1024 + 64 \cdot k$

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

From Fermat’s theorem

- $g^h \mod p = h^{p-1} \mod p = 1$
- g - generator of the cyclic group of order q in $Z_p^*$
**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 GF(p) or GF(2\(^m\))
- **P** - point of order **q** on the elliptic curve **E**

---

**Digital Signature Algorithm**

*Public and private key*

**Private key**

- x - arbitrary 160 bit number  
  \(0 < x < q\)

**Public key**

- \(y = g^x \mod p\)  
  \(0 < y < p\)  
  L - bit number
Elliptic Curve Digital Signature Algorithm

Public and private key

Private key

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

Public key

\[ Y = x \, P \]

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

\[ s = k^{-1} (\text{SHA}(M) + x \cdot r) \mod q \]

\[ \text{SGN}(M) = r \| s \]

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
   \( R = kP \)
   \( r : x \)-coordinate of \( R \)

3. Compute hash value

   - Message \( M \)
   - SHA
   - SHA(\( M \))

4. Compute

   \[ s = k^{-1}(SHA(M) + x \cdot r) \mod q \]

   \[ SGN(M) = r || s \]

**DSA: Signature verification**

1. Compute hash value

   - Message \( M' \)
   - SHA
   - SHA(\( M' \))

2. Compute

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

3. Compute

   \[ u1 = SHA(M') \cdot w \mod q \]

4. Compute

   \[ u2 = r' \cdot w \mod q \]

5. Compute

   \[ v = ((g^{u1} \cdot y^{u2}) \mod p) \mod q \]

6. Compare

   - Signature invalid
   - Signature valid
**ECDSA: Signature verification**

1. Compute hash value

   Message M'  
   \[
   \text{SHA} \rightarrow \text{SHA}(M')
   \]

2. Compute

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

3. Compute

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

4. Compute

   \[u_2 = r' \cdot w \mod q\]

5. Compute

   \[V = u_1 P + u_2 Y\]

6. Compare

   \[\text{Signature invalid} \quad \text{N} \quad \text{v} = r' \quad \text{Y} \quad \text{Signature valid}\]

**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 \(Z_p^*\)
### 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 GF(p) or GF(2^m)
- **P** - generator of the group of points on the elliptic curve

### El-Gamal Encryption

**Public and private key**

**Private key**

\[ x \text{ - arbitrary number} \quad 1 \leq x \leq p-2 \]

**Public key**

\[ y = g^x \mod p \quad 0 < y < p \]
**Elliptic Curve El-Gamal Encryption**

*Public and private key*

**Private key**

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

**Public key**

\[ Y = x \cdot P \]

---

**El-Gamal: Encryption**

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

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

3. Compute
   
   \[ c = y^k \cdot M \mod p \]
   
   \( C(M) = r \parallel c \)
**Elliptic Curve El-Gamal: Encryption**

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

2. Compute

   *message public key*
   
   $R = kP$

3. Compute

   $M = (m, n)$

3. Compute

   $m$ - *message*
   
   $n$ - *y-coordinate*  
   corresponding to the x-coordinate $m$

   $C = kY + M \mod p$

   $C(m) = R \parallel C$

**El-Gamal: Decryption**

\[
\begin{array}{c|c}
 r & c \\
\end{array} \\
C(M)
\]

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

**Justification:**

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

\[= y^k \cdot M \cdot (y^k)^{-1} = M\]
Elliptic Curve El-Gamal: Decryption

\[
\begin{array}{ccc}
R & C & C(m) \\
\end{array}
\]

\[M = C - x \cdot R\]

\[m: \text{x-coordinate of } M\]

Justification:

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

Menezes-Vanstone Elliptic Curve Cryptosystem

System parameters

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

\[E - \text{elliptic curve over } \text{GF}(p) \text{ or } \text{GF}(2^m)\]

\[P - \text{generator of the group of points on the elliptic curve}\]
Menezes-Vanstone Elliptic Curve Cryptosystem

Public and private key

Private key

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

Public key

\[ Y = xP \]

Menezes-Vanstone Cryptosystem: Encryption

1. Choose random
   
   message private key \(1 \leq k \leq \#E(GF(q))-1\),
   
   (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

   \[ y_1 = c_1m_1 \]
   \[ y_2 = c_2m_2 \]

   \[ C(m_1, m_2) = R \parallel y_1y_2 \]
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>

$C = x R = (c_1, c_2)$

$m_1 = c_1^{-1} y_1$

$m_2 = c_2^{-1} y_2$

**Justification:**

$x R = x k P = k (x P) = k Y = C$