ECE297:11 Lecture 15

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

$x, y \in K$

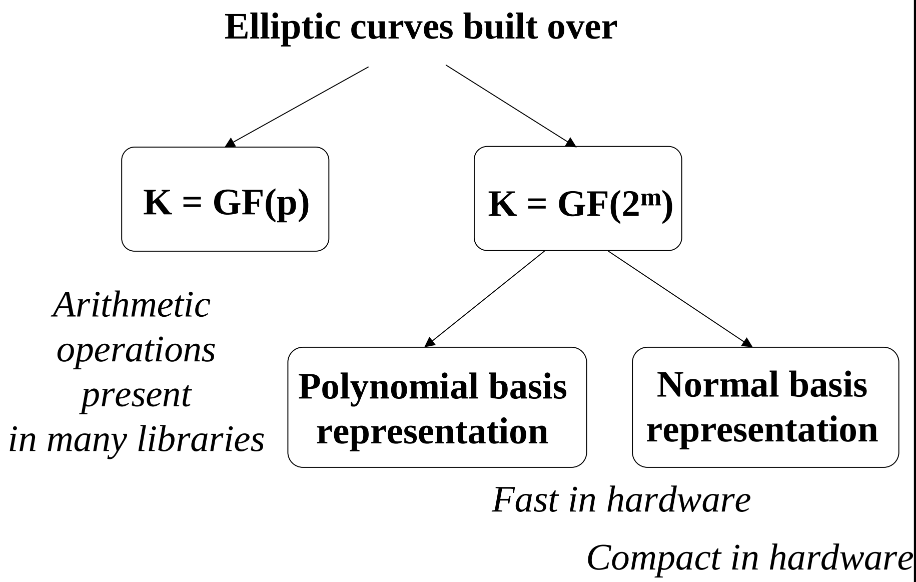
$a_1, a_2, a_3, a_4, a_5, a_6 \in K$

K is a field

Values of a_i limited by constraints specific to the field K

+ a special point called *the point at infinity* O

Three Classes of Elliptic Curves



Elliptic Curve over GF(p)

Set of solutions (x, y) to the equation

$$y^2 = x^3 + a\,x + b$$

where $x, y \in$ GF(p)

$a, b \in$ GF(p) $4a^3 + 27\,b^2 \not\equiv 0 \pmod{p}$

+ a special point called *the point at infinity* O

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

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)
		O

28 points

Generating a point of an elliptic curve (1)

1. Choose x

e.g., $x=3$

2. Compute $z = y^2 = x^3 + a x + b$

e.g., $z = 3^3 + 1 \cdot 3 + 1 \pmod{23} = 8$

3. If z = 0, then y=0 and there is only <u>one point</u>, (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., $8^{(23-1)/2} \pmod{23} = 8^{11} \bmod 23 =$

$= (8^8 \bmod 23)(8^2 \bmod 23)(8^1 \bmod 23) \pmod{23} =$

$= 4 \cdot 18 \cdot 8 \pmod{23} = 1$

If Euler's criterion is not met (i.e., $z^{(p-1)/2} \neq 1 \pmod{p}$), then there is <u>no point</u> 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 <u>two points</u> with a given x coordinate

$$(x, y_1) \text{ and } (x, y_2)$$

If $p \equiv 3 \pmod{4}$ then

y_1 and y_2 can be computed from the equation

$$y_1 = +z^{(p+1)/4} \pmod{p}$$

$$y_2 = -z^{(p+1)/4} \pmod{p} \equiv p - z^{(p+1)/4} \pmod{p} =$$

$$= p - y_1$$

E.g., $23 \equiv 3 \bmod 4$

$y_1 = 8^{(23+1)/4} \bmod 23 = 8^6 \bmod 23 = 13$

$y_2 = -13 \equiv 23 - 13 = 10$

Addition of two points on the elliptic curve over GF(p) (1)

$P = (x_1, y_1)$ $Q = (x_2, y_2)$

$R = P + Q = (x_3, y_3)$

Case 1:

$$P + \mathcal{O} = \mathcal{O} + P = P$$

Case 2:

$$x_2 = x_1 \text{ and } y_2 = -y_1$$

$$P + Q = \mathcal{O}$$

$$Q = -P$$

Addition of two points on the elliptic curve over GF(p) (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)

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

$x_3 = 8^2 - 2 - 2 \bmod 11 = 9 - 2 - 2 \bmod 11 = 5$
$y_3 = 8\,(2 - 5) - 7 \bmod 11 = 9 - 7 \bmod 11 = 2$

2P = (5, 2)

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

P = (2, 7) 2P = (5, 2)

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

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

$x_3 = 2^2 - 2 - 5 \bmod 11 = 4 - 2 - 5 \bmod 11 = 8$
$y_3 = 2\,(2 - 8) - 7 \bmod 11 = 10 - 7 \bmod 11 = 3$

3P = (8, 3)

Scalar multiples of P

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)
	13P = $\mathcal{O}$

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)

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

$$5.37 \leq \#E(GF(11)) \leq 18.63$$

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

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: $y = a^e \bmod n$

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

Right-to-left binary exponentiation

```
y = 1;
s = a;
for i=0 to L-1
  {
    if (e_i == 1)
      y = y · s mod n;
    s = s^2 mod n;
  }
```

Left-to-right binary exponentiation

```
y = 1;
for i=L-1 downto 0
  {
   y = y^2 mod n;
    if (e_i == 1)
      y = y · a mod n;
  }
```

Scalar Multiplication: $Y = k \cdot P$

Right-to-left binary scalar multiplication

Left-to-right binary scalar multiplication

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

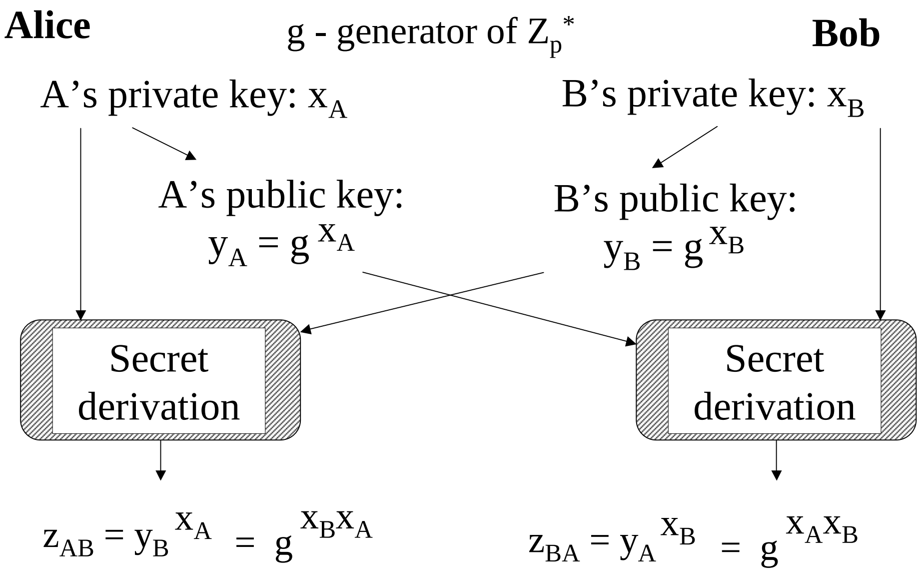
Right-to-left:

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

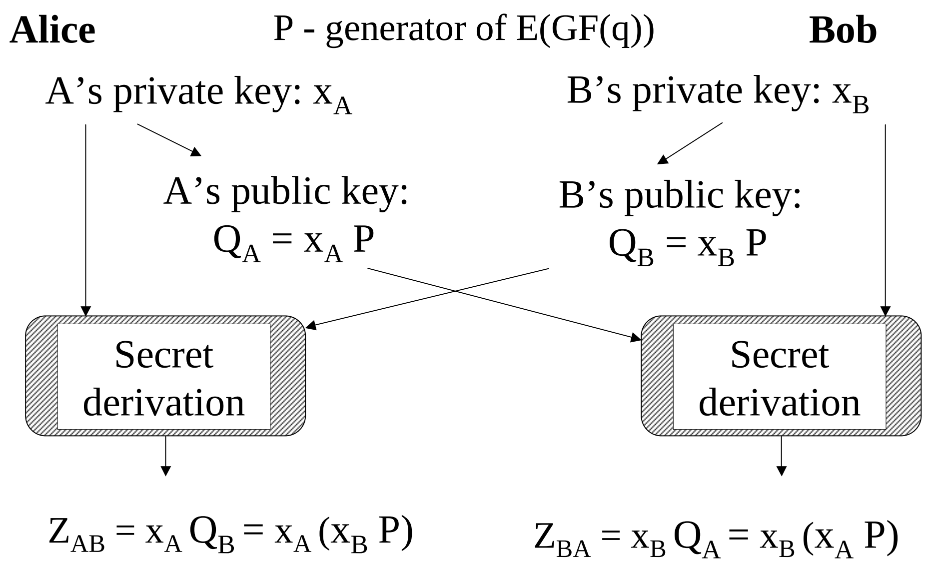
Left-to-right:

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

Diffie-Hellman



Elliptic Curve Diffie-Hellman



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
$g^q \bmod p = h^{p-1} \bmod p = 1$
g - generator of the cyclic group of order q in Zp*

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

$y = g^x \bmod p$ $0 < y < p$

L - bit number

Elliptic Curve Digital Signature Algorithm

Public and private key

Private key

x - arbitrary number $0 < x < q$

Public key

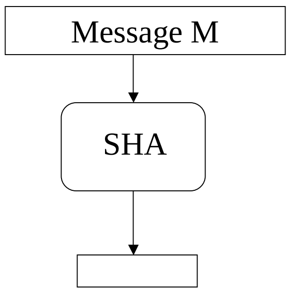
$Y = x P$

DSA: Signature generation

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

2. Compute
message public key
$\boldsymbol{r} = (g^k \bmod p) \bmod q$

3. Compute *hash value*



SHA(M)

4. Compute

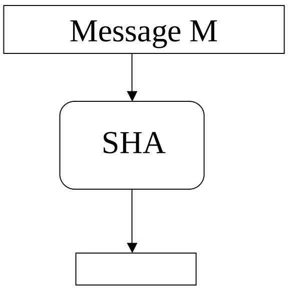
$$\boldsymbol{s} = k^{-1}\,(\mathrm{SHA}(M) + x \cdot r) \bmod q$$

$$\mathrm{SGN}(M) = r \parallel s$$

160 bit 160 bit 40 bytes

ECDSA: Signature generation

3. Compute *hash value*



SHA(M)

1. Choose random *message private key* $1 < \boldsymbol{k} < q$ (secret, different for each message)

2. Compute *message public key*
   $\mathbf{R} = k\,P$
   $\boldsymbol{r}$: x-coordinate of R

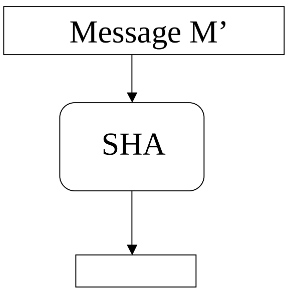
4. Compute

$$\boldsymbol{s} = k^{-1}\,(\mathrm{SHA(M)} + x \cdot r) \bmod q$$

$$\mathrm{SGN(M)} = r \parallel s$$

DSA: Signature verification

1. Compute *hash value*



SHA(M’)

r’	*s’*

[SGN(M)]’

2. Compute

$$w = (s')^{-1} \bmod q$$

3. Compute

$$u1 = \mathrm{SHA(M')} \cdot w \bmod q$$

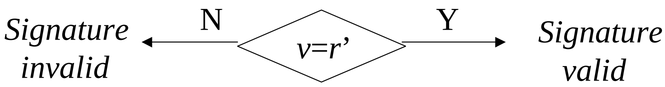
4. Compute

$$u2 = r' \cdot w \bmod q$$

5. Compute

$$v = ((g^{u1} \cdot y^{u2}) \bmod p) \bmod q$$

6. Compare

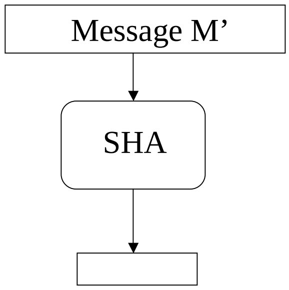


ECDSA: Signature verification

1. Compute *hash value*

r'	s'

[SGN(M)]'



SHA(M')

2. Compute

$w = (s')^{-1} \bmod q$

3. Compute

$u_1 = \mathrm{SHA}(M') \cdot w \bmod q$

4. Compute

$u_2 = r' \cdot w \bmod q$

5. Compute $V = u_1 P + u_2 Y$ v is the x-coordinate of V

6. Compare

Signature invalid ← N — $v = r'$ — Y → *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 Zp*

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 - arbitrary number $1 \le x \le p-2$

Public key

$y = g^x \bmod p$ $0 < y < p$

Elliptic Curve El-Gamal Encryption

Public and private key

Private key

x - arbitrary number $1 \le x \le \#E(GF(q))-1$

Public key

$Y = x P$

El-Gamal: Encryption

1. Choose random
message private key $1 \le \boldsymbol{k} \le p-2$,
relatively prime with p-1
(secret, different for each message)

2. Compute
message public key
$\boldsymbol{r} = g^k \bmod p$

3. Compute

$\boldsymbol{c} = y^k \cdot M \bmod p$

$C(M) = \boldsymbol{r} \,||\, \boldsymbol{c}$

Elliptic Curve El-Gamal: Encryption

1. Choose random

message private key $1 \le \boldsymbol{k} \le \#E(GF(q))-1$,
(secret, different for each message)

2. Compute
message public key
$\mathbf{R} = k\,P$

3. Compute

$\boldsymbol{C} = k\,Y + M \bmod p$

$C(m) = \boldsymbol{R} \,||\, \boldsymbol{C}$

3. Compute

$\mathbf{M} = (m, n)$

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

El-Gamal: Decryption

r	*c*

C(*M*)

$M = c \cdot (r^x)^{-1} \bmod p$

Justification:

$c \cdot (r^x)^{-1} \bmod p = y^k \cdot M \cdot ((g^k)^x)^{-1} = y^k \cdot M \cdot ((g^x)^k)^{-1} =$
$= y^k \cdot M \cdot (y^k)^{-1} = M$

Elliptic Curve El-Gamal: Decryption

R	*C*

C(m)

$M = C - x R$

m: x-coordinate of M

Justification:

$C - x R = (k Y + M) - x R = (k Y + M) - x k P =$
$= (k Y + M) - k (x P) = k Y + M - k Y = M$

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 $1 \le x \le \#E(GF(q))-1$

Public key

$Y = x P$

Menezes-Vanstone Cryptosystem: Encryption

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

2. Compute
message public key
$\mathbf{R} = k P$

3. Form message block:
(m_1, m_2)

4. Compute

$\boldsymbol{C} = k Y = (c_1, c_2)$

5. Compute

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

$C(m_1, m_2) = \boldsymbol{R} \,||\, \boldsymbol{y_1, y_2}$

Menezes Vanstone Cryptosystem : Decryption

R	y_1	y_2

$C(m_1, m_2)$

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

$m_1 = c_1^{-1} y_1$
$m_2 = c_2^{-1} y_2$

Justification:

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