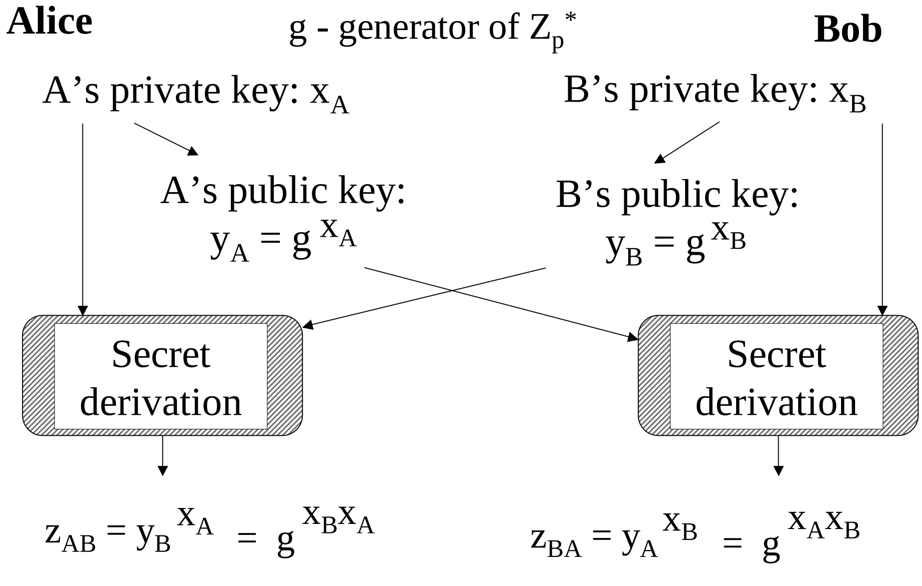
ECE297:11 Lecture 18

Implementation of public key cryptosystems

Diffie-Hellman



Alice

g - generator of Z_p^*

Bob

A's private key: x_A

B's private key: x_B

A's public key: $y_A = g^{x_A}$

B's public key: $y_B = g^{x_B}$

Secret derivation

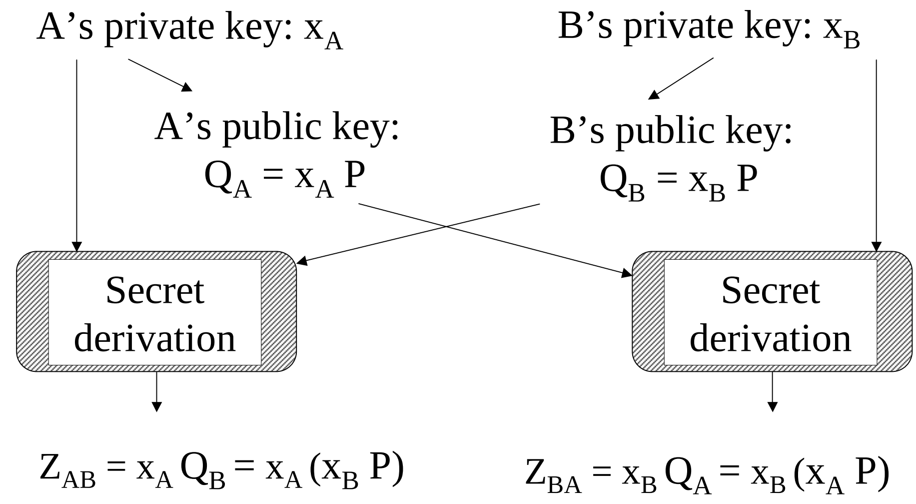
Secret derivation

$z_{AB} = y_B^{x_A} = g^{x_B x_A}$

$z_{BA} = y_A^{x_B} = g^{x_A x_B}$

Elliptic Curve Diffie-Hellman

Alice P - generator of E(GF(q)) **Bob**



A's private key: x_A

B's private key: x_B

A's public key:
$Q_A = x_A P$

B's public key:
$Q_B = x_B P$

Secret derivation

Secret derivation

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

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

Exponentiation: $y = a^e \bmod n$

Right-to-left binary exponentiation

Left-to-right binary exponentiation

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

```
$y$ = 1;
$s$ = $a$;
for  $i$=0 to L-1
  {
    if ($e_i$ == 1)
      $y = y \cdot s \bmod n$;
   $s = s^2 \bmod n$;
  }
```

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

Scalar Multiplication: $Y = k \cdot P$

Right-to-left binary scalar multiplication

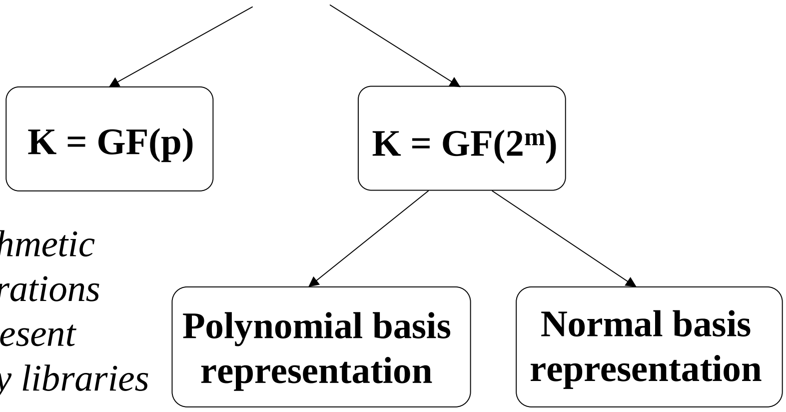
Left-to-right binary scalar multiplication

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

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

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

Three Classes of Elliptic Curves



Elliptic curves built over

K = GF(p)

K = GF(2^m)

Polynomial basis representation

Normal basis representation

Arithmetic operations present in many libraries

Fast in hardware

Compact in hardware

Elliptic Curve over GF(p)

Set of solutions (x, y) to the equation

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

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

Elliptic Curve over GF(2^n)

Non-supersingular

Set of solutions (x, y) to the equation

$$\boldsymbol{y^2 + xy = x^3 + a_2\,x^2 + a_6}$$

where $x, y \in$ GF(2^n)

$a_2 \in \{0,1\}$, $a_6 \in$ GF(2^n)

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

Elliptic Curve over GF(2^n)

Supersingular

Set of solutions (x, y) to the equation

$$y^2 + a_3 y = x^3 + a_4 x + a_6$$

where $x, y \in$ GF(2^n)

a_3 , a_4 , $a_6 \in$ GF(2^n), $a_3 \neq 0$

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

MOV (Menezes-Okamoto-Vanstone) attack

- The ***elliptic curve discrete logarithm problem*** **on E(GF(q))** can be reduced to the ***logarithm problem*** **over GF(q^k)**
- The ***logarithm problem*** **over GF(q^k)** can be solved in subexponential time using the **index calculus method**
- Value of k
  - small (< 7) for supersingular curves
  - large for non-supersingular curves
- **Non-supersingular curves more suitable for cryptographic transformations**

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

$$\mathbf{P = (x_1, y_1) \qquad Q = (x_2, y_2)}$$

$$\mathbf{R = P + Q = (x_3, y_3)}$$

Case 1:

$$P + O = O + P = P$$

Case 2:

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

$$P + Q = O$$

$$Q = -P$$

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 $P = Q$

$$\lambda = \frac{3x_1^2 + a}{2y_1} = (3x_1^2 + a)(2y_1)^{-1}$$

Addition of two points on the elliptic curve over GF(p)   (3)

Case 3a: if $P \neq Q$

2 multiplications in GF(p)
1 squaring in GF(p)
1 inverse in GF(p)
6 subtractions in GF(p)

Case 3b: if $P = Q$

2 multiplications in GF(p)
2 squarings in GF(p)
1 inverse in GF(p)
6 additions/subtractions in GF(p)

Addition of two points on the non-supersingular elliptic curve over GF(2^n)

$$\mathbf{P = (x_1, y_1) \qquad Q = (x_2, y_2)}$$

$$\mathbf{R = P + Q = (x_3, y_3)}$$

Case 1:

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

Case 2:

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

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

$$Q = -P$$

Addition of two points on the non-supersingular elliptic curve over GF(2^n)

Case 3a: if $P \neq Q$

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a_2$$
$$y_3 = \lambda (x_1 - x_3) - y_1$$

where

$$\lambda = \frac{y_1 + y_2}{x_1 + x_2} = (y_1 + y_2)(x_1 + x_2)^{-1}$$

Number of field operations:

1 inversion in GF(2^n)
2 multiplications in GF(2^n)
1 squaring in GF(2^n)

Addition of two points on the non-supersingular elliptic curve over GF(2^n)

Case 3b: if $P = Q$

$$x_3 = a_6 (x_1^{-1})^2 + x_1^2$$
$$y_3 = x_1^2 + (x_1 + y_1 x_1^{-1}) x_3 + x_3$$

Number of field operations:

1 inversion in GF(2^n)
3 multiplications in GF(2^n)
2 squarings in GF(2^n)

Notation

a	Multiplicand	$a_{k-1}a_{k-2} \, . \, . \, . \; a_1 \; a_0$
x	Multiplier	$x_{k-1}x_{k-2} \, . \, . \, . \; x_1 \; x_0$
p	Product (a · x)	$p_{2k-1}p_{2k-2} \, . \, . \, . \; p_2 \; p_1 \; p_0$

Basic Multiplication Equations

$$p = a \cdot x \qquad\qquad x = \sum_{i=0}^{k-1} x_i \cdot 2^i$$

$$p = a \cdot x = \sum_{i=0}^{k-1} x_i \cdot 2^i =$$

$$= x_0a2^0 + x_1a2^1 + x_2a2^2 + \ldots + x_{k-1}a2^{k-1}$$

Shift/Add Algorithms

Right-shift algorithm

$$p = a \cdot x = x_0a2^0 + x_1a2^1 + x_2a2^2 + \dots + x_{k-1}a2^{k-1} =$$

$$= (...((0 + \underbrace{x_0a2^k)/2 + x_1a2^k)/2 + ... + x_{k-1}a2^k)/2}_{\text{k times}} =$$

$$p^{(0)} = 0$$

$$p^{(j+1)} = (p^{(j)} + x_j\, a\, 2^k) / 2 \qquad j=0..k-1$$

$$p = p^{(k)}$$

Shift/Add Algorithms

Right-shift algorithm: multiply-add

$$p^{(0)} = y2^k$$

$$p^{(j+1)} = (p^{(j)} + x_j\, a\, 2^k) / 2 \qquad j=0..k-1$$

$$p = p^{(k)}$$

$$= (...((y2^k + \underbrace{x_0a2^k)/2 + x_1a2^k)/2 + ... + x_{k-1}a2^k)/2}_{\text{k times}} =$$

$$= y + x_0a2^0 + x_1a2^1 + x_2a2^2 + \dots + x_{k-1}a2^{k-1} = y + a \cdot x$$

Notation

z	Dividend	$z_{2k-1}z_{2k-2} \dots z_2\ z_1\ z_0$
d	Divisor	$d_{k-1}d_{k-2} \dots d_1\ d_0$
q	Quotient	$q_{k-1}q_{k-2} \dots q_1\ q_0$
s	Remainder (s = z - dq)	$s_{k-1}s_{k-2} \dots s_1\ s_0$

Basic Equations of Division

$$z = \mathbf{q}\, d + \mathbf{s}$$

$$s < d$$

Unsigned Integer Division Overflow

Condition for no overflow:

$$z = q\ d + s < (2^k\text{-}1)\ d + d = d\ 2^k$$

$$z = z_H\ 2^k + z_L < d\ 2^k$$

$$z_H < d$$

Sequential Integer Division
Basic Equations

$$s^{(0)} = z$$

$$s^{(j)} = 2\ s^{(j-1)} - q_{k-j}\ (2^k\ d)$$

$$s^{(k)} = 2^k\ s$$

Sequential Integer Division Justification

$s^{(1)} = 2\ z - q_{k-1}\ (2^k\ d)$
$s^{(2)} = 2(2\ z - q_{k-1}\ (2^k\ d)) - q_{k-2}\ (2^k\ d)$
$s^{(3)} = 2(2(2\ z - q_{k-1}\ (2^k\ d)) - q_{k-2}\ (2^k\ d)) - q_{k-3}\ (2^k\ d)$

. . . . . .

$s^{(k)} = 2(. . . 2(2(2\ z - q_{k-1}\ (2^k\ d)) - q_{k-2}\ (2^k\ d)) - q_{k-3}\ (2^k\ d) . . .$
$\quad - q_0\ (2^k\ d) =$
$= 2^k\ z - (2^k\ d)\ (q_{k-1}\ 2^{k-1} + q_{k-2}\ 2^{k-2} + q_{k-3}\ 2^{k-3} + \ldots + q_0 2^0) =$
$= 2^k\ z - (2^k\ d)\ q = 2^k\ (z - d\ q) = 2^k\ s$

Montgomery Modular Multiplication (1)

C = A · B mod M A, B, M – k-bit numbers

Integer domain **Montgomery domain**

$A \longrightarrow A' = A \cdot 2^k \bmod M$

$B \longrightarrow B' = B \cdot 2^k \bmod M$

$C' = MP(A', B', M) =$
$= A' \cdot B' \cdot 2^{-k} \bmod M =$
$= (A \cdot 2^k) \cdot (B \cdot 2^k) \cdot 2^{-k} \bmod M =$
$= A \cdot B \cdot 2^k \bmod M$

$C = A \cdot B \longleftarrow C' = C \cdot 2^k \bmod M$

Montgomery Modular Multiplication (2)

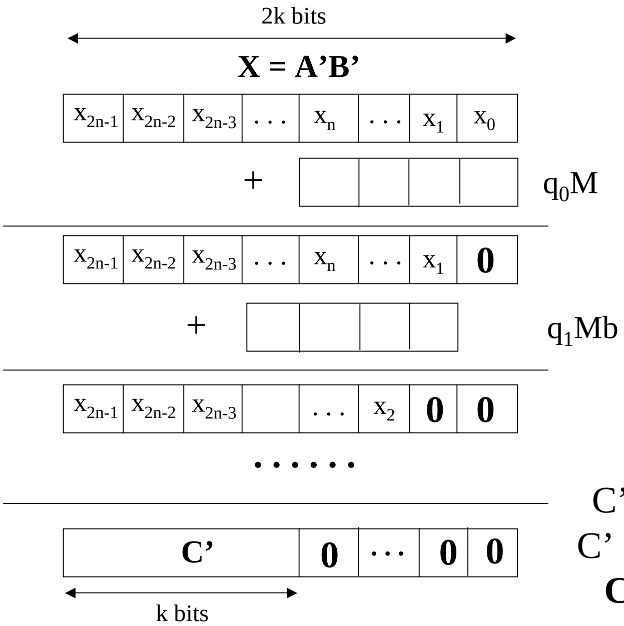
$$\mathbf{A} \longrightarrow \mathbf{A'}$$

$$\mathbf{A' = MP(A, 2^{2k} \text{ mod } M, M)}$$

$$\mathbf{C} \longleftarrow \mathbf{C'}$$

$$\mathbf{C = MP(C', 1, M)}$$

Montgomery Modular Multiplication (3)



$$C'\,2^k = X + zM$$

$$C'\,2^k \equiv X = A'B'$$

$$\mathbf{C' \equiv A'B'\,2^{-k}}$$