Information Security

#12. 공개키 암호2 - Elliptic Curve Cryptosystems

참고자료

1. **CPE5021, Advanced Network Security from Univ. Manash**
2. **https://medium.com/coinmonks/the-wonderful-world-of-elliptic-curve-cryptography-b7784acdef50**
3. **Bill's Security site.com, https://asecuritysite.com/**

Howon Kim

2022.4

정보보호 및 사물지능 연구실
Information Security & AIoT (AI of Things)
블록체인 플랫폼 연구센터
BLOCKCHAIN PLATFORM RESEARCH CENTER
부산대학교 융합보안대학원
PNU GRADUATE SCHOOL OF CONVERGENCE SECURITY

Agenda

- Background
  - Group
  - Field
- Principle of public key systems
  - Discrete Logarithm Problem (DLP)
- Elliptic Curve Cryptography
- ECC in practice

Group

- Definition: a set S together with a binary operation $(\oplus)$ defined on S *is a group if* $(S,\oplus)$ satisfies the following properties:

- *Closure*: $a,b \in S \Rightarrow a \oplus b \in S.$
- *Identity*: $\exists e \in S \text{ s.t. } e \oplus a = a \oplus e = a \quad \forall a \in S.$
- *Associativity*: $\forall a,b,c \in S, \ (a \oplus b) \oplus c = a \oplus (b \oplus c).$
- *Inverses*:

For each $a \in S, \ \exists! b \in S \text{ s.t. } a \oplus b = b \oplus a = e.$

If a group S satisfies the *commutative law* $a \oplus b = b \oplus a \ \ \forall a,b \in S$ the group is said to be **Abelian**.

Field

- Definition: a set F with two operations (+,*) *defined* on it is a field if it satisfies the following criteria:

  - (F; +) is an Abelian group;

  - (F \ {0}, *) is an Abelian group, where {0} is the identity of addition and zero of multiplication;

  - distributive: $\forall\ \boldsymbol{x; y; z} \in \mathrm{F}$

$$x * (y + z) = x * y + x * z$$

$$(x + y) * z = x * z + y * z$$

Elliptic Curve Cryptography (ECC)

RSA

- Security of popular RSA is provided by the intractability of the *integer factoring problem (IFP)*
- To ensure security of RSA-based systems, current key sizes must be a minimum of 1024 bits.
- As computing power increases, larger key sizes will be needed to guarantee security of such systems. This will result in
  - higher computational costs
  - more space requirement for keys
  - reduction in scalability

Elliptic curve cryptosystem (ECC)

- Why ECC?
  - There are other public key cryptographic systems. However, we choose to study ECC because
    - The sub-exponential algorithm of breaking ECC has not been found, that is : ECC is not less secure than RSA or some other public key crypto algorithms.
    - ECC with smaller key size can achieve the same security as RSA or some other crypto algorithms. Hence ECC is more efficient for secure wireless applications.
    - High scalability.
    - More potential due to EC theory (rich theory with many alternatives).

Elliptic curve cryptosystem (ECC)

Symmetric key size (in bits)	Example algorithm	DLP key size for equivalent security (p in bits)	RSA key size for equivalent security (n in bits)	ECC key size for equivalent security (n in bits)	Key size ratio of RSA to ECC (approx)
56	-	512	512	112	5:1
80	SKIPJACK2 2	1024	1024	160	6:1
112	Triple DES	2048	2048	224	9:1
128	AES-128	3072	3072	256	12:1
192	AES-192	7680	7680	384	20:1
256	AES-256	15360	15360	512	30:1

RSA and ECC challenges

Year	Number of decimal digits	Number of bits	MIPS Years	Calendar Time to Solution	Method (year method developed)
1994	**129**	**429**	**5000**	**8 months, using 1600 computers**	**Quadratic Sieve (1984)**
1995	**119**	**395**	**250**		
1996	**130**	**432**	**750**		**General Number Field Sieve (1989)**
1999	**140**	**466**	**2000**		
1999	**155**	**512**	**8000**	**3.7 months**	**General Number Field Sieve (1989)**

Progress in Integer Factorisation (Certicom 1997)

RSA and ECC challenges

RSA Security organisation sponsors a challenge for solving the integer factorisation problem IFP or DLP (to break RSA), while Certicom corporation sponsors a challenge for solving the EC DLP (to break ECC)

Date Solved	Bits	Details	MIPS Years
September, 1999	**97**	**740 computers, 130 billion EC operations.**	**16,000**
April, 2000	**108**	**9,500 computers.**	**≈ 400,000**
November, 2002	**109**	**10,000 computers for 549 days.**	**-**

Progress in solving ECDLP Certicom (2002)

Discrete Logarithm Problem (DLP)

- For a group G,

Given group elements, α, β

find an integer x such that $\beta = \alpha^x$

x is called the *discrete log* of β to the base α.

- It is easy to compute β

- It is hard to find x, knowing α and β

DLP - Example

- If $a^b = c$, then $\log_a c = b$
- Example:
  - $2^3 = 8 \Leftrightarrow \log_2 8 = 3$
  - $10^3 = 1000 \Leftrightarrow \log_{10} 1000 = 3$
- Computing a^b and $\log_a c$ are both easy for real numbers.
- However, when working with field such as (Zp,mod), it is easy to calculate $c = a^b \bmod p$, but given *c*, *a* and *p* it is very difficult to find *b*.
- Given an integer *n* it is hard to find two integers *p*, *q* such that $n = p \bullet q$ (factorisation problem as in RSA)

Real Elliptic Curves

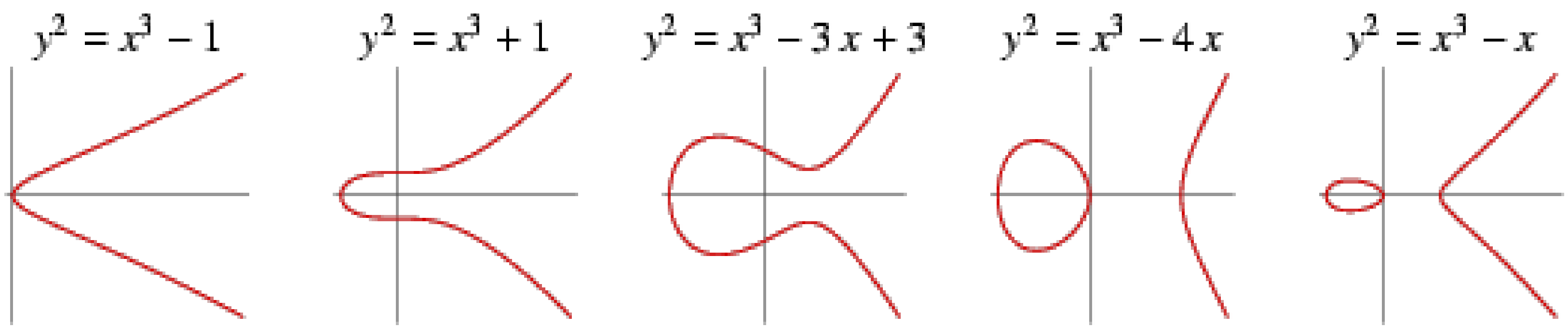
- An elliptic curve is defined by an equation in two variables x & y, with coefficients:
  - $y^2 + \boldsymbol{a}xy + \boldsymbol{b}y = x^3 + \boldsymbol{c}x^2 + \boldsymbol{d}x + \boldsymbol{e}$ ***(general form)***
- Consider a cubic elliptic curve of form
  - $y^2 = x^3 + \mathbf{a}x + \boldsymbol{b}$; where x,y,*a*,*b* are all real numbers. Eg.
    - $y^2 = x^3 + x + \mathbf{1}$.
    - $y^2 = x^3 + 2x + \mathbf{6}$.

General form of Elliptic Curves

- An *elliptic curve* is a plane curve defined by an equation of the form

$$y^2 = x^3 + ax + b$$

Examples



Weierstrass Equation

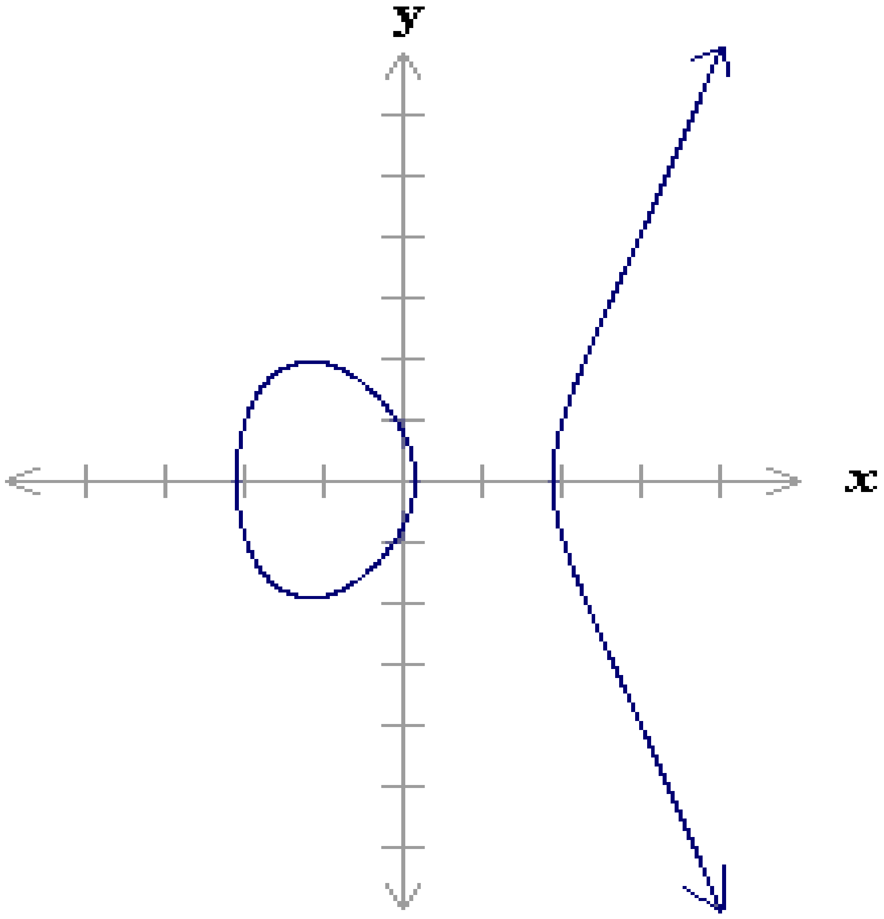
- A two variable equation F(x,y)=0, forms a curve in the plane. We are seeking geometric arithmetic methods to find solutions

- Generalized Weierstrass Equation of elliptic curves:

$$y^2 + a_1xy + a_3y = x^2 + a_2x^2 + a_4x + a_6$$

Here, A, B, x and y all belong to a field of say rational numbers, complex numbers, finite fields (F_p) or Galois Fields ($GF(2^n)$).

Example of EC



$$y^2 = x^3 - 4x + 0.67$$

Elliptic curve over real number

- Let's consider the equation:

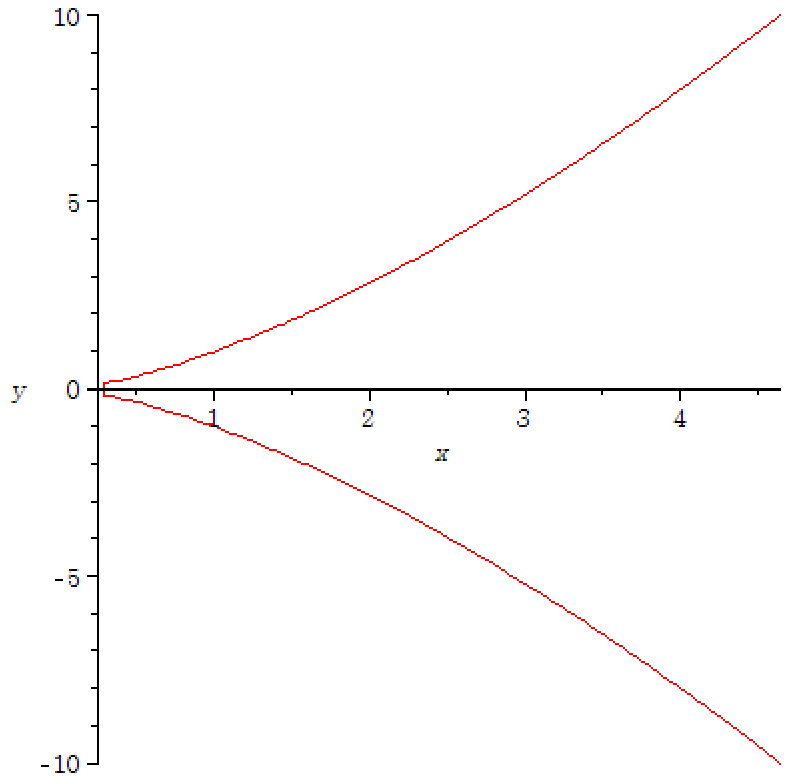
  $y^2 = x^3 + ax + b$, where x, y, a and b are real numbers, where $4a^3 + 27b^2 \neq 0$ – condition for distinct single roots (smooth curve).

  판별식 **(Discriminant)**

- All (x,y) points satisfying above equation along with an infinite point ***O*** and addition operation (+), form a group.
  - ***O* is the identity of the group.**
  - (+) is group addition operation

Elliptic curve over real number

- If $4a^3 + 27b^2 = 0$, then it has single root ! (plot of the $y^2 = x^3$)
  - *We should not use this curve*



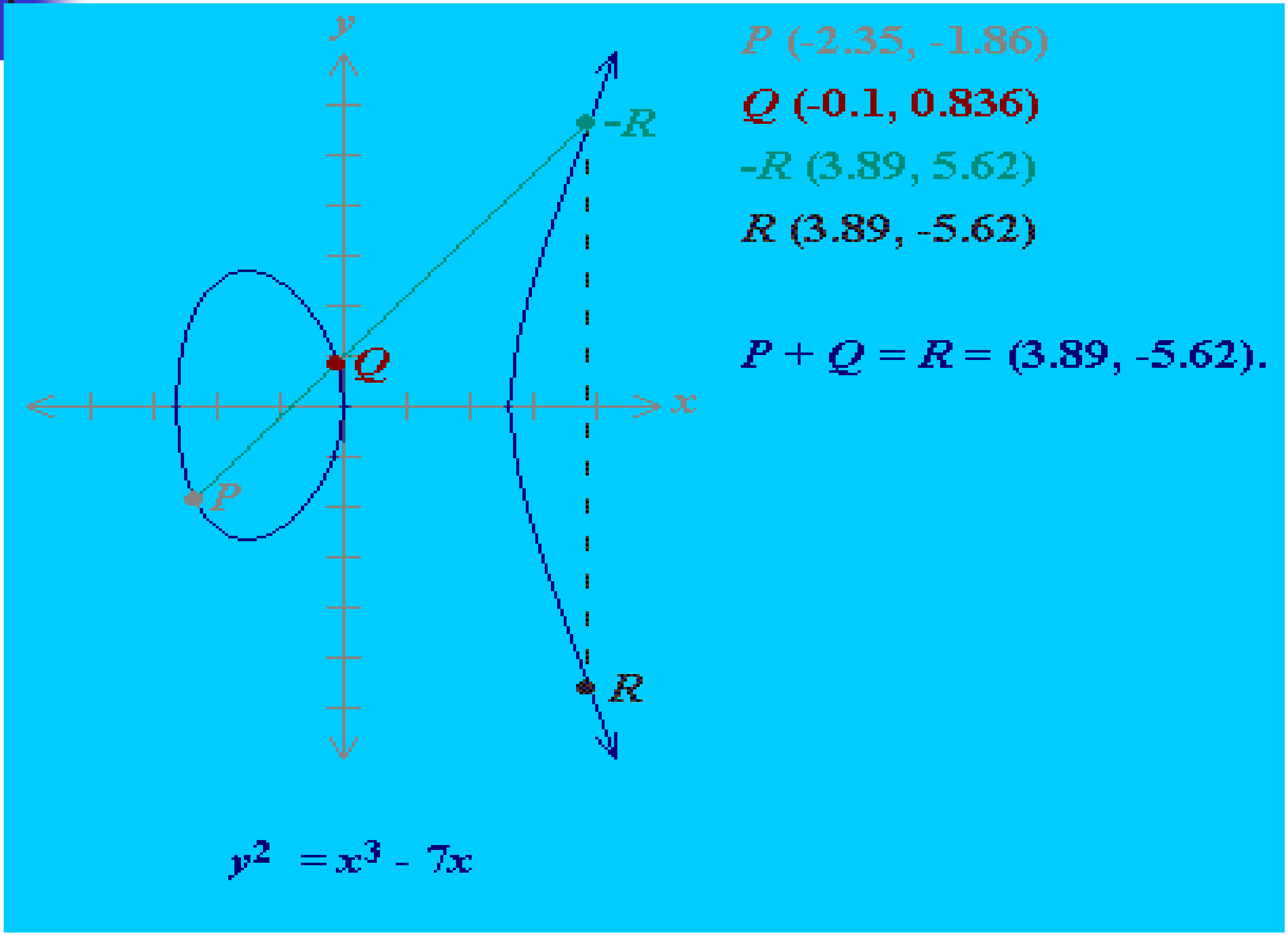
EC over a group (G,+) – E(G,+)

An EC over a group (G,+) is defined with the following:

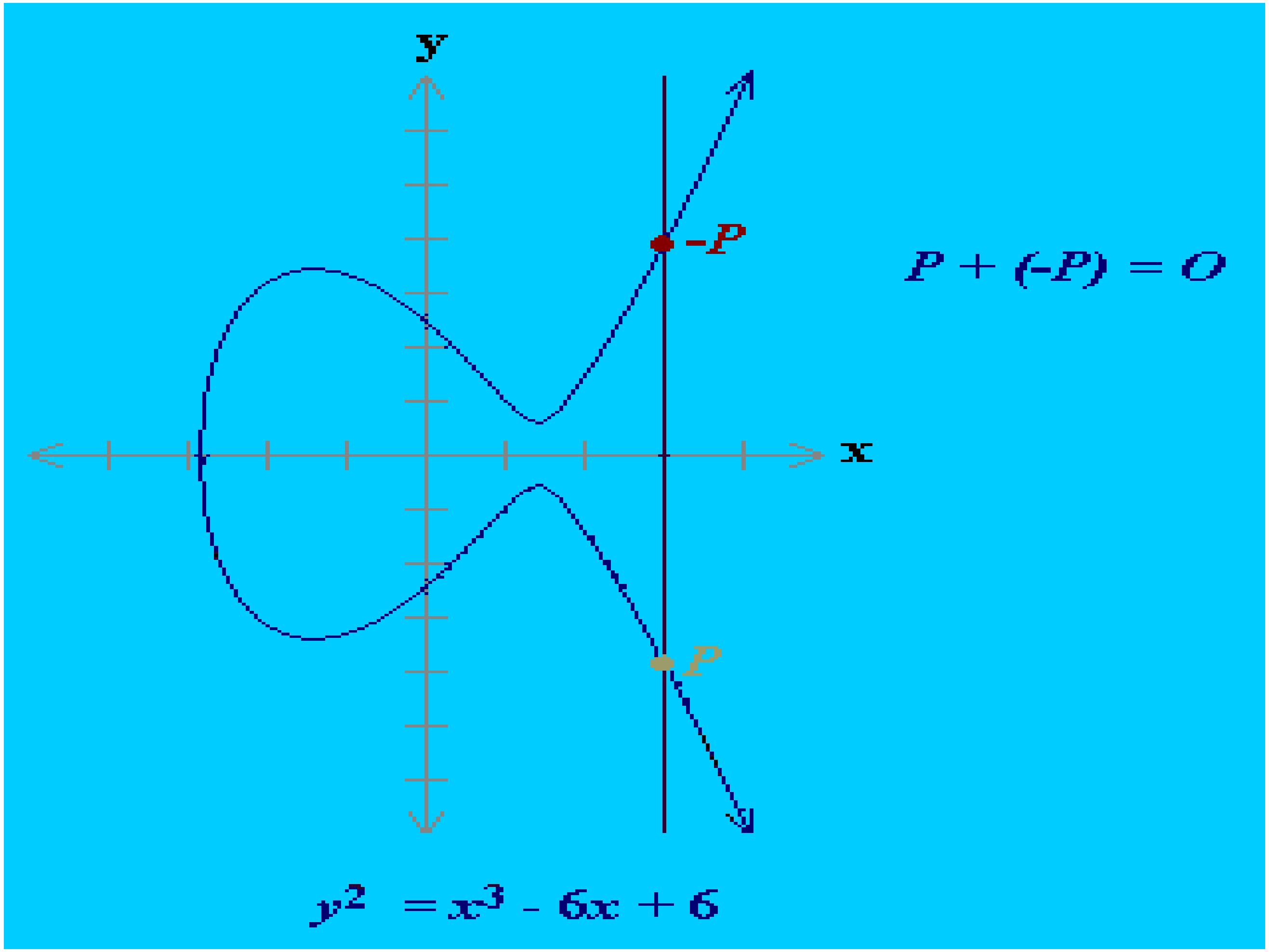
1. Addition (+) : If P and Q are distinct, and $P \neq -Q$, define P+Q as follows:
   - Draw a line through P and Q, then the line will intersect with the curve, the intersected point is denoted as –R, and define P+Q=R.
2. *For every P, define* $P + (-P) = O$
3. *If* $P=(x,0)$, *then* $P+P = O$, (a vertical line)

   Otherwise, draw a tangent(touch!) line through P, the intersected point is defined as –R, then P+P =2P =R.

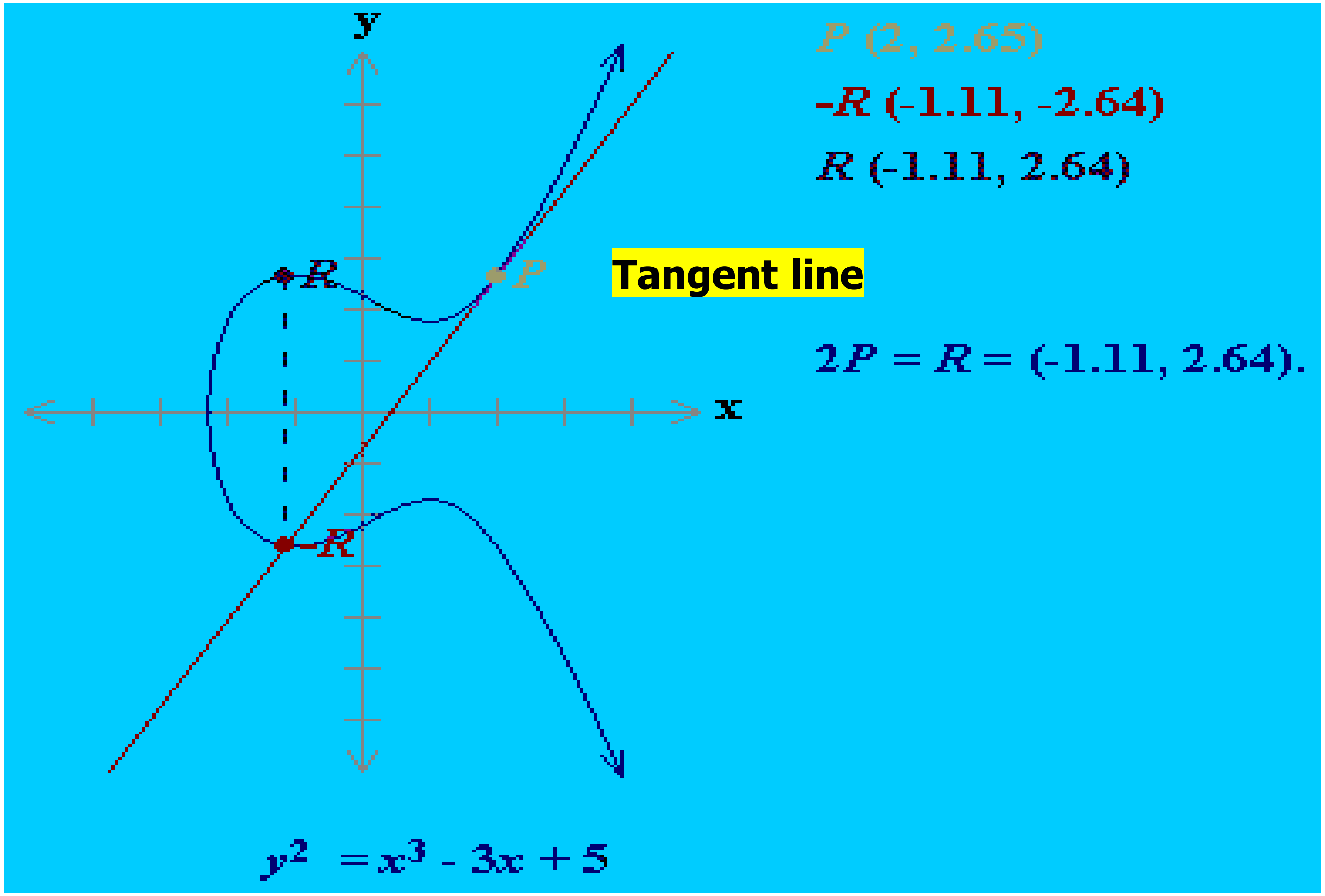
Definition of P+Q = R



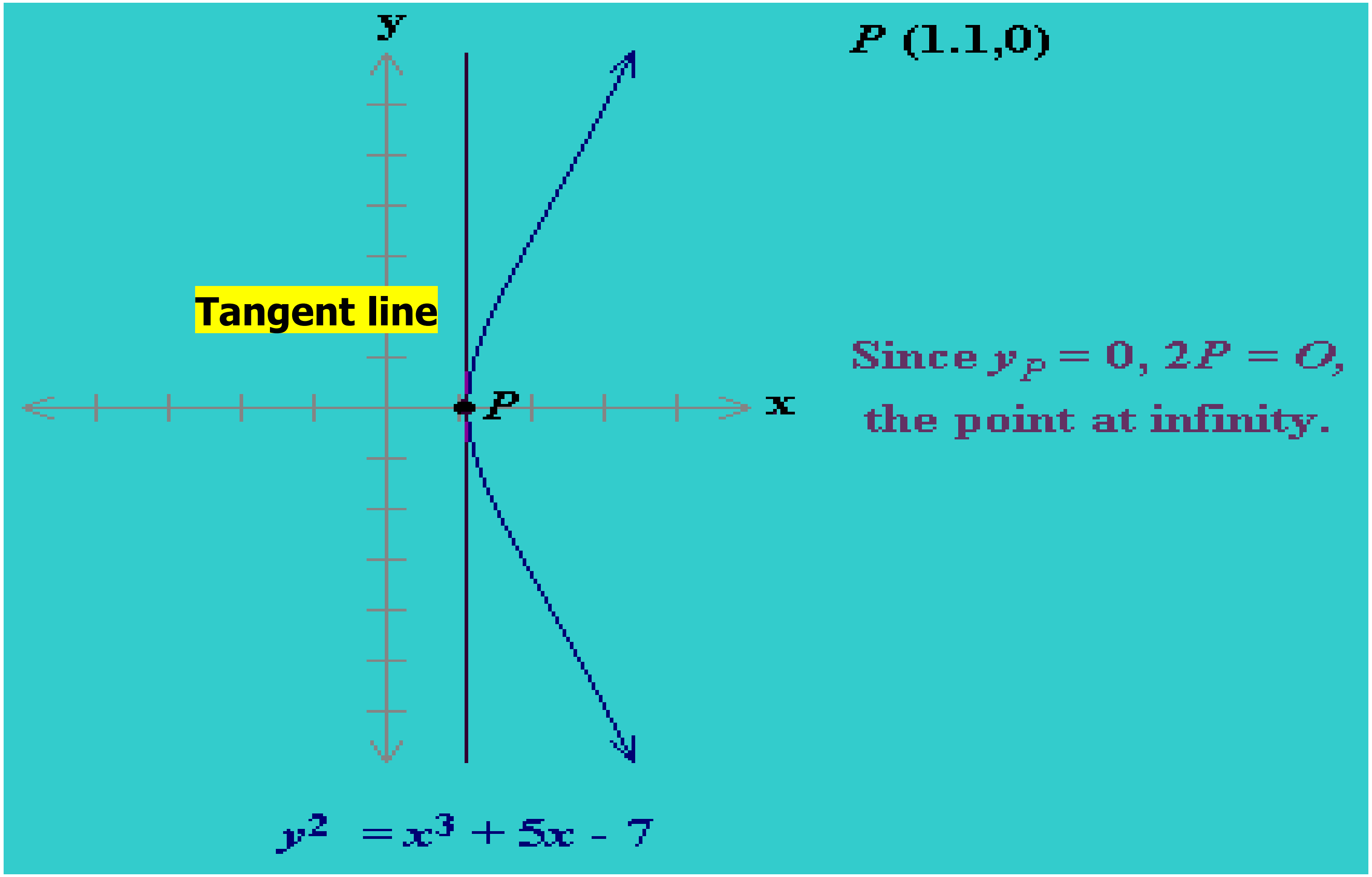
Definition of P+(-P)



Definition of P+P (where y!=0)



Definition of P+P (where y=0)



Elliptic Curve : An Algebraic Approach

1. Adding distinct points P and Q (1)
   When $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ and $P \neq Q$, $P \neq -Q$,
   $P + Q = R(x_R, y_R)$ with $x_R = s^2 - x_P - x_Q$ and $y_R = s(x_P - x_R) - y_P$
   where $s = (y_P - y_Q) / (x_P - x_Q)$

2. Doubling the point P (2)
   When y_P is not O,
   $2P = R(x_R, y_R)$ with $x_R = s^2 - 2x_P$ and $y_R = s(x_P - x_R) - y_P$
   where $s = (3x_P^2 + a) / (2y_P)$

3. $P + (-P) = O$ (3)

4. If $P = (x_P, y_P)$ and $y_P = 0$, then $P + P = 2P = O$ (4)

Elliptic Curve : An Algebraic Approach

- Let $P(x_1,y_1)$, $Q(x_2,y_2)$, $R(x_3,y_3)=P+Q$ are on the EC
- We can represent the $R(=P+Q)$ using the P, Q. Also, the following holds. $-R = (x_3, -y_3)$
- In the case of $P=(x_1,y_1),\quad Q=(x_2,y_2),\quad x_2 \neq x_1$
  - The gradient of line (k) PQ is $k = \dfrac{y_1-y_2}{x_1-x_2}$
  - So the equation of the line PQ is $y = k(x-x_1)+y_1$
  - We apply this equation to the EC equation.
  - EC equation: $y^2 = x^3 + ax + b$
  - Then we get $[k(x-x_1)+y_1]^2 = x^3 + ax + b$
  - Also we get

$$x^3 - k^2x^2 + (2k^2x_1 - 2ky_1 + a)x + b - k^2x_1{}^2 + 2kx_1y_1 - y_1{}^2 = 0$$

$ax^3+bx^2+cx+d=0(a\neq 0)$의 세 근을 α,β,γ라 하면

$\alpha+\beta+\gamma=-\dfrac{b}{a}$

Elliptic Curve : An Algebraic Approach

- In the equation, we know the P (x_1, y_1), Q (x_2, y_2) are the roots of the equation.
- So we can think three roots (x_1, x_2, x_3) satisfy the following conditions:

$$x_1 + x_2 + x_3 = k^2$$

$$x_3 = k^2 - (x_1 + x_2),$$

$$-y_3 = k(x_3 - x_1) + y_1 = kx_3 + (y_1 - kx_1) = y_1 + k(x_3 - x_1)$$

- Now we the $R = P + Q$ can be expressed as follows:

$$x_3 = k^2 - (x_1 + x_2) = \boxed{\left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 - (x_1 + x_2)}$$

$$y_3 = -y_1 + k(x_1 - x_3)$$

$$= \boxed{-y_1 + \left(\frac{y_1 - y_2}{x_1 - x_2}\right)(x_1 - x_3)}$$

  - The case of P!=Q

Elliptic Curve : An Algebraic Approach

- There are another equations for the case of P = Q.
  etc.

Finite Elliptic Curves on discrete Fields

- Cryptography works with finite field and Elliptic curve cryptography uses curves whose variables and coefficients are finite
- There are two commonly used ECC families:
  - prime curves $E_p(a,b)$ defined over Z_p
    - use modulo with a prime number p
    - efficient in software
  - binary curves $E_{2m}(a,b)$ defined over $GF(2^n)$
    - use polynomials with binary coefficients
    - efficient in hardware

Elliptic Curve Groups over Z_p

- $(Z_p, \text{mod}) = \{0,1,\ldots,p\text{-}1\}$ is a group
  - Where p is a prime number
- Define the elliptic curve
  - $y^2 = x^3 + \mathbf{a}x + \mathbf{b} \bmod p$
  - Where **a** and **b** are in Z_p, and x, y are also in Z_p.
  - $(4a^3 + 27b^2 \pmod p) \neq 0$.

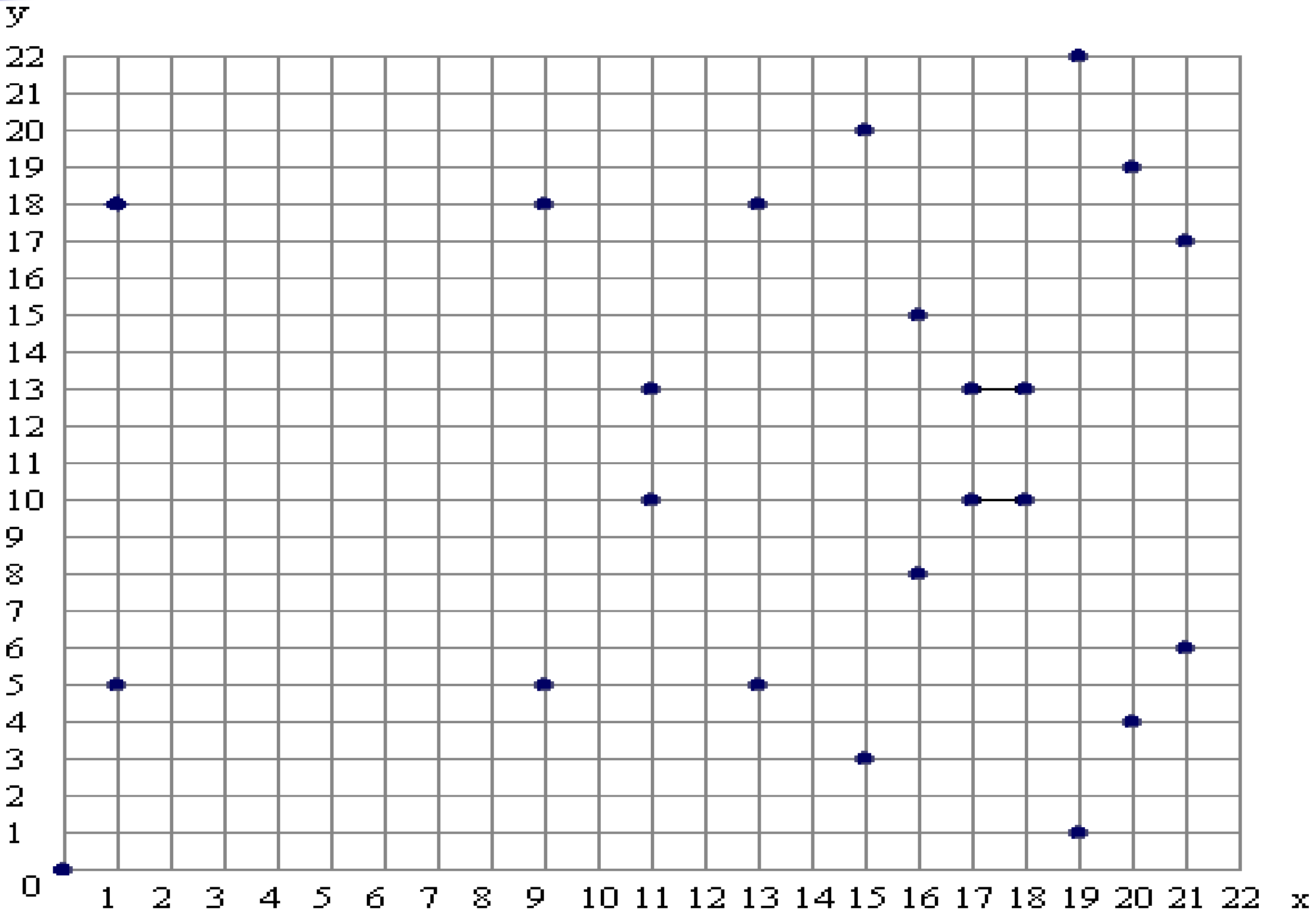
EC over (Z_p, mod)- examples

- p=11, Zp=Z_{11}. $y^2 = x^3 + x + 6 \pmod{11}$
  - E(Z_{11}, mod) = {(2,4),(2,7), (3,5),(3,6), (5,2),(5,9), (7,2),(7,9), (8,3),(8,8), (10,2),(10,9)}

- p=23, Zp=Z_{23}. $y^2 = x^3 + x \pmod{23}$
  - E(Z_{23}, mod) = {(0,0), (1,5), (1,18), (9,5), (9,18), (11,10), (11,13), (13,5), (13,18), (15,3), (15,20), (16,8), (16,15), (17,10), (17,13), (18,10), (18,13) (19,1),(19,22), (20,4), (20,19), (21,6), (21,17)}

- p=23, Zp=Z_{23}. $y^2 = x^3 + x + 1 \pmod{23}$
  - E(Z_{23},mod) = { (0,1), (0,22), (1,7), (1,16), (3,10), (3,13), (4,0), (5,4), (5,19), (6,4), (6,19), (7,11), (7,12), (9,7), (9,16), (11,3), (11,20), (12,4), (12,19), (13,7), (13,16), (17,3), (17,20), (18,3), (18,20), (19,5), (19,18)}

$y^2 = x^3 + x \bmod 23$



Elliptic curve equation: $y^2 = x^3 + x$ over F_{23}

Operations on $E(Z_{11}, mod)$

- Consider the $E(Z_{11}, mod)$:

Let P and Q on $E(Z_{11}, mod)$

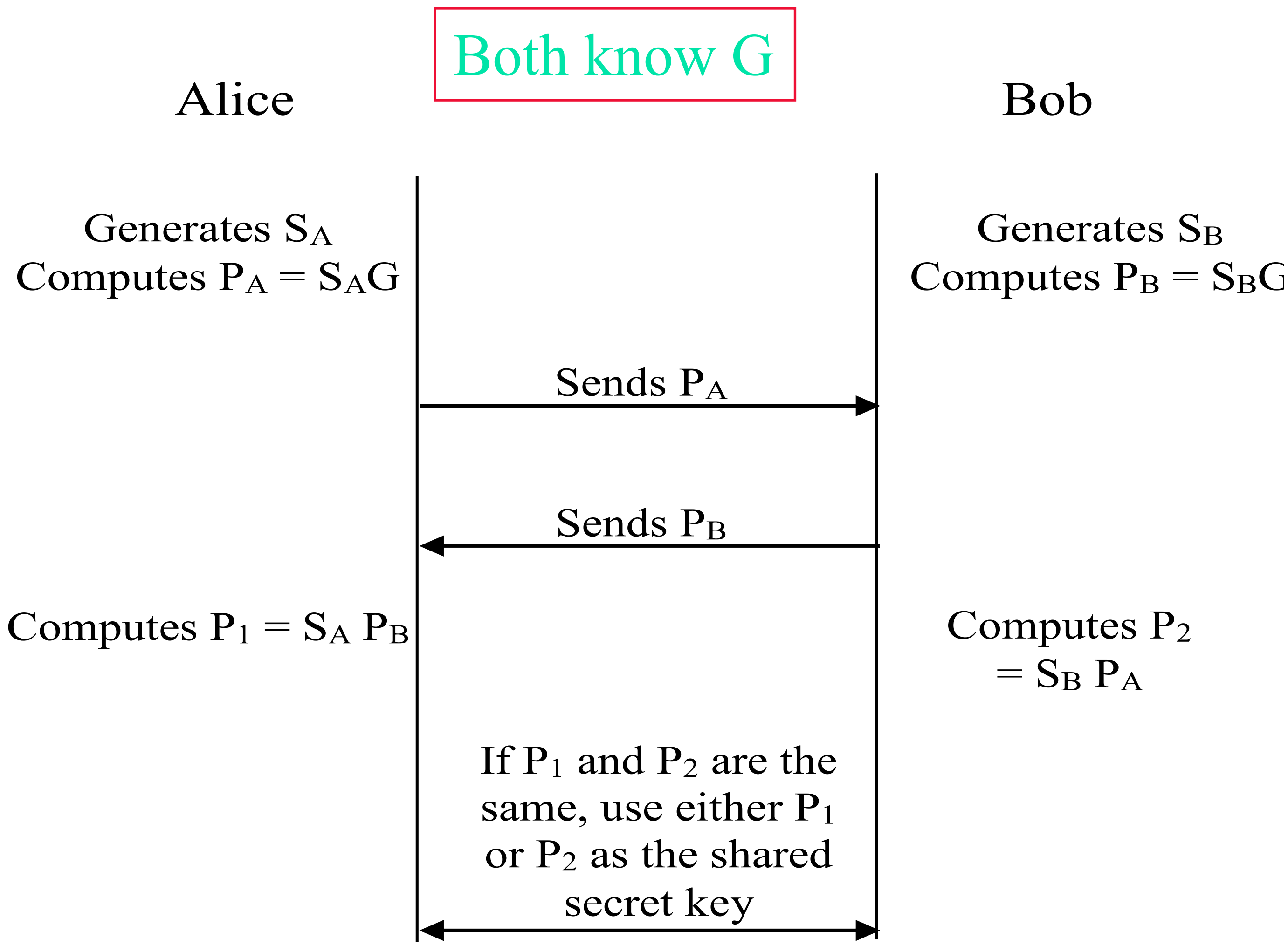
1. P = **(10,2)** and Q= **(5,2) then** P + Q **= (10,2) + (5,2) = (7,9).**
2. P = **(2,7);** P + P **= (5,2).**
3. P = **(2,7); -P = (2,-7); P + -P =?**

ECC system (general approach)

- General steps to construct an EC cryptosystem
  1. Selects an underlying field F
  2. Implementing arithmetic operations in F
  3. Selecting an appropriate EC over F to form E(F)
  4. Implementing EC operations in group E(F)
  5. Choose a protocol
  6. Implement ECC based on the chosen protocol.

In real applications, we have to use only "the recommended curves for higher security". For example, the NIST recommended curves such as ECC P256R1 are secure.
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
실제 응용에서는 표준 기술만 사용해야 함 !!!

Diffie-Hellman Key Exchange Protocol



Diffie Hellman over ECC

- Alice chooses a random ***a*** and compute **a**P $\in$ E
- Bob chooses a random ***b*** and compute **b**P $\in$ E
- Alice and Bob exchange the computed values
- Alice, from **b**P and a can compute S = **ab**P
- Bob, from **a**P and b can compute S = **ab**P

Simple implementation of ECC

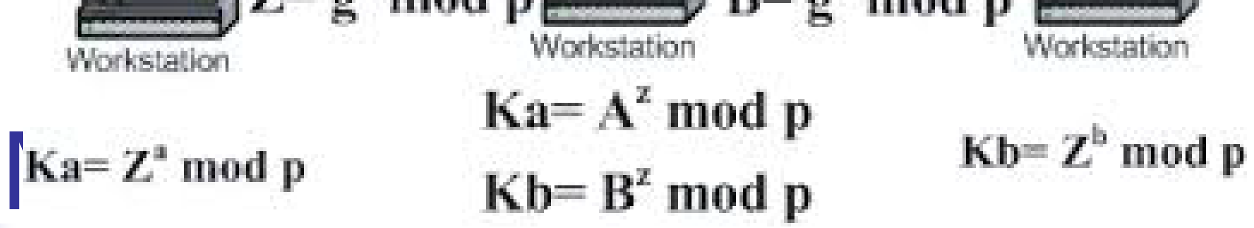
- Simple steps to construct an EC cryptosystem
  - Select an underlying field F and generate a random curve (e.g: $y^2 = x^3 + \boldsymbol{a}x + \boldsymbol{b}$) – store values of ***a*** and ***b***

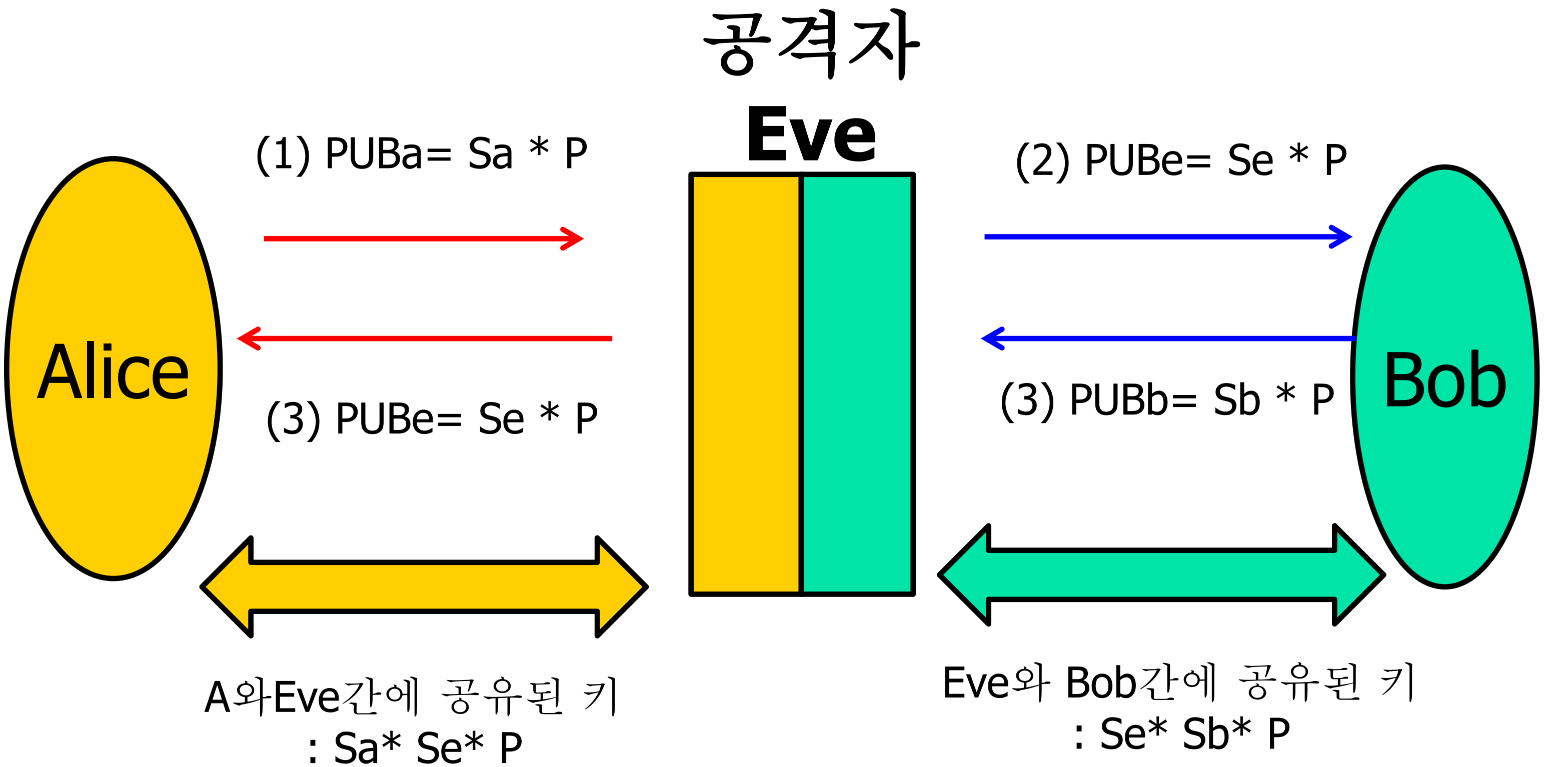
    (should declare data structures to store curve and point parameters prior this)
  - Find the base point *g* (generator) as public point (Everyone knows this point)
  - Compute shared secret key using Diffie Hellman over ECC
  - Compute public keys:
    - Alice chooses a random number as a secret key Sa and computes her public key Pa = Sa* g
    - Bob chooses a random number Sb as his secret key and computes his public key Pb = Sb* g

    (Both Alice and Bob can now compute the shared key Sb*Sa*g)

  - Embed message *m* onto a point, M(x,y), of the curve using Koblitz's method
  - Encrypt and decrypt
    - Alice encrypts the message M(x,y): (Pa, Sa* Pb + M) and sends it to Bob.
    - Bob decrypts the message by computing Pa*Sb and then

      M + Sa*Pb - Pa*Sb = M + Sa*Sb*g – Sa*g*Sb = M





Sa : A의 비밀키
PUBa : A의 공개키

Se: E의 비밀키
PUBe : E의 공개키

Sb : B의 비밀키
PUBb : B의 공개키

Basic operations of ECC

Complex operations on points of an Elliptic Curve

Level 1

- scalar multiplication: $k \cdot P = \underbrace{P + P + \ldots + P}_{k \text{ times}}$

Basic operations on points of an Elliptic Curve

Level 2

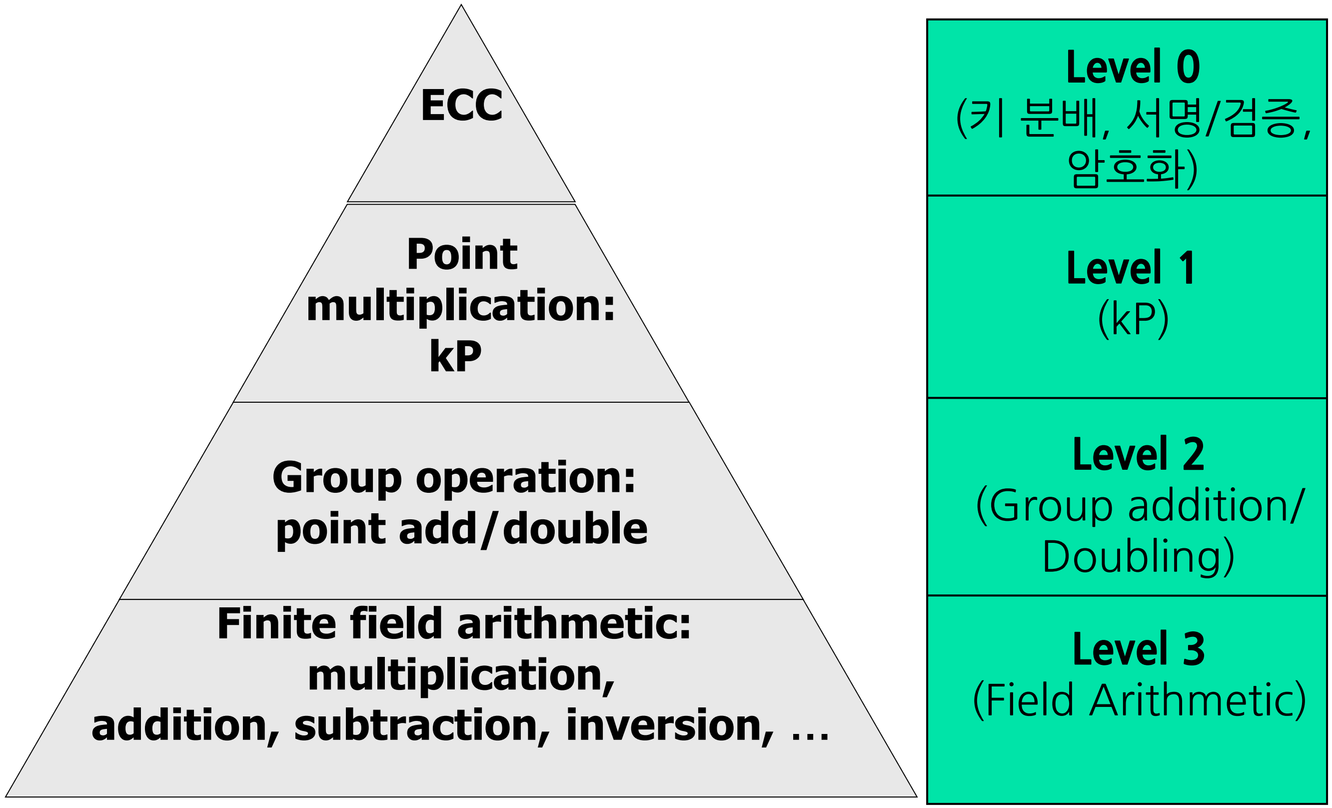
- addition of points: $P + Q$
- doubling a point: $2P$
- projective to affine coordinate: P2A

Basic operations in Galois Field $GF(2^m)$ or Z_p

Level 3

- addition and subtraction (xor): $x+y$, $x-y$
- multiplication, squaring: $x \cdot y$, x^2
- inversion: x^{-1}

ECC operations: Hierarchy



Scalar multiplication

- Basic crypto operation of an ECC.
- Series of point addition and doubling.
- Binary method due to no pre-computation phase .
- Faster processing when using signed representation of the scalar value.

Scalar Multiplication: MSB first

- Require $k=(k_{n-1},k_{n-2},\ldots,k_0)_2$, $k_{n-1}=1$
- Compute $Q=kP$
  - $Q=P$
  - For $i=n-2$ to 0
    - $Q=2Q$ (doubling)
    - If $k_i=1$ then
      - $Q=Q+P$ (addition)
    - End if
  - End for
  - Return Q

참고 논문: Cryptology and Network Security: 9th International Conference, CANS 2010, p.186

Example

- **Compute 7P:**
  - $7=(111)_2$
  - 7P=2(2(P)+P)+P=> 2 iterations are required
  - Principle: First double and then add (accumulate)
- **Compute 6P:**
  - $6=(110)_2$
  - 6P=2(2(P)+P)

Embedding plaintext messages as points on an Elliptic Curve

- In order to build an ECC, there must be an accurate and efficient way for embedding a ciphertext message on an EC.
  - There is no known deterministic algorithm for embedding message units as points on an elliptic curve.
  - However, there is a probabilistic method that can be used for embedding message units as points on an elliptic curve.
  - See Koblitz's proposal of representing a message unit as a point on an EC.

Embed a message *m*

- Suppose p is prime with p mod 4 = 3
- Pick k so that $1/2^k$ is small
- Let *m* be the message and allows *m* < (p-k)/k
- For j=0, …, k-1
- Set $x_j = m*k + j$; $w_j = x^3 + a\,x_j + b$; $z_j = w_j^{((p+1)/4)}$
- If ($z_j^2 = w_j$) then (x_j , z_j) is the point to encode *m*
- If no, j works then FAIL with Prob. <= $1/2^k$
- If *m* is embedded as M(x,y) then *m* = [x/k]

The method proposed by Koblitz represents a message as a point on an elliptic curve. Suppose E is an elliptic curve given by $y^2 = x^3 + Ax + B$ over a field F_q where q is a large prime. Using the following steps to map a message m to a point on the curve.

1. Treat m as an element in F_q and let x have the value of m.
2. Compute $\alpha = x^3 + Ax + B \, mod \, q$.
3. Find the square root β of α mod q.
   (a) Compute $\delta = \alpha^{(q-1)/2} \, mod \, q$.
   (b) If $\delta \neq 1$, set $x = x + 1$, goto Step 2.
   (c) Compute the square root β using one of the following methods.
      i. If $q \equiv 3 \, mod \, 4$, compute $u = (q-3)/4$ and set $\beta = \alpha^u \, mod \, q$.
      ii. If $q \equiv 5 \, mod \, 8$, compute $u = (q-5)/8$, $\gamma = (2\alpha)^u \, mod \, q$, $i = 2\alpha\gamma^2 \, mod \, q$ and set $\beta = \alpha\gamma(i-1) \, mod \, q$.
      iii. If $q \equiv 1 \, mod \, 4$, please refer to [1].
4. If the right-most bit of β equals to $x \, mod \, 2$, then set $y = \beta$. Otherwise, set $y = q - \beta$.
5. Output the point (x, y).

A의 square root를 계산하는 것은 결국 msg m이 curve y^2=m^3+… 상의 points인지를 check하는 목적임

Embed a message *m*

- **Koblitz's message embedding example**

- $y^2 = x^3 + x + 6 \mod 11$ (we use E(Z11))
  p= 11 mod 4 = 3
- k=3;

- // if j=2, let m=2
- x2 =2*3 + 2 = 8
- w2 = 8^3 + 8 + 6 = 9
- z2 = w2^(12/4)=w2^3 = 9^3 mod 11 = 3
- if(z^2 == w2) → 3^2 == 9 yes.
- (x2,z2)=(8,3)은 m을 EC 상에 인코딩한 point 값

- // if j=1, let m = 2
- x1 = 2 * 3 + 1 = 7
- w1 = 7^3 + 7 + 6 mod 11 = 4
- z1 = 4^3 mod 11 = 9
- If (z1^2 == w1?) 9^2 mod 11 → 4 .. yes.
- (x1,z1)=(7,9)은 m을 EC에 인코딩한 point 값
-

- // if j=0, let m=2
- X0 = 2*3=6
- w0 = 6^2 + 6 + 6 = 48 mod 11 = 4
- z0 = 4^3 = 64 mod 11 = 9
- If (9^2 == 4)? 4 → yes
- (x0,z0)=(6,8)은 m을 EC상에 인코딩한 point 값

Point Compression

- An elliptic curve point $P=(x,y)$ can be represented by its x-coordinate and an additional bit.
- This is because, given x, the elliptic curve equation becomes quadratic in y. The quadratic equation has at most two solutions, so one bit is sufficient to specify y (the additional bit is not required when Char(F)=2 and P has odd order [7]). For example, for the case of Fp, we have

$$Y^2 = x^3 + a*x + b = x(x^2 + a) + b$$

- Therefore, given x and an additional bit, y can be obtained at the cost of 1M + 1S + 1SR, where M, S, and SR denote the cost of a field multiplication, a field squaring, and a square root operation, respectively.
- When the square root operation is computationally expensive, this point compression is not practical. In this case, the elliptic curve points are typically represented by both their x-coordinate and y-coordinate

, : IEEE Trans. On Computers, Vol.56, No.3, March 2007, Reference "Double Point Compression with Applications to Speeding up Random Point Multiplication"

[7] G. Seroussi, "Compact Representation of Elliptic Curve Points over IF2n ," Technical Report No. HPL-98-94R1, Hewlett-Packard Laboratories, 1998.

Some references

- http://beast.csse.monash.edu.acu/cpe5021
- http://www.certicom.com/index.php?action=res,ecc_faq (good introduction papers)
- http://cnscenter.future.co.kr/crypto/algorithm/ecc.html (more materials)
- http://www.cs.mdx.ac.uk/staffpages/m_cheng/link/ecc_simple.pdf (good introduction for students)
- http://www.secg.org/collateral/sec1_final.pdf SEC1: Elliptic Curve Cryptography
- http://www.secg.org/collateral/sec2_final.pdf SEC2: Recommended Elliptic Curve Domain Parameters

Next…

- We will study on Message Authentication and Hash Functions…

Q&A