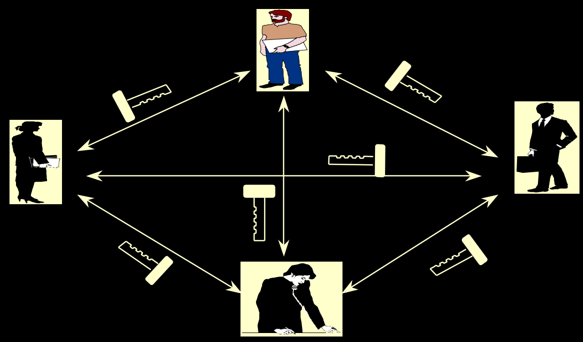
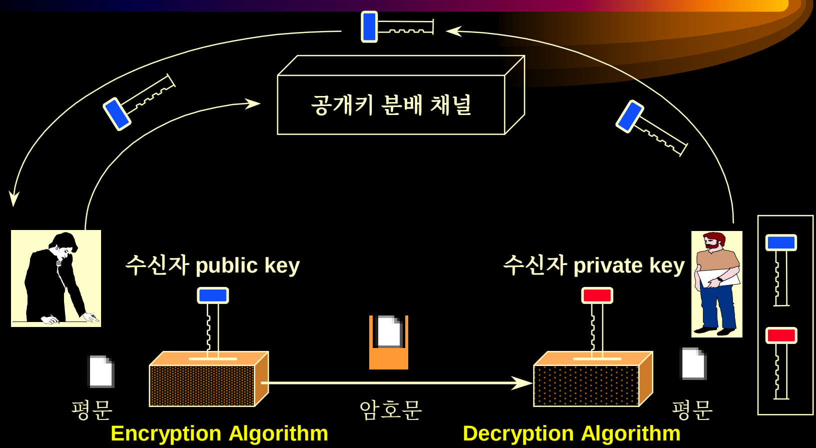
❑ 대칭형 암호의 문제점

- 대칭키 분배 문제
- 대칭키 관리 문제



❑ 공개키 암호시스템 개요



❑ One-Way Function (일 방향 함수)

- ❑ A function which is easy to compute in one direction, but difficult to invert
  - given x, $y = f(x)$ is easy
  - given y, $x = f^{-1}(y)$ is difficult
- ❑ **Integer Multiplication** vs. ***Factorization***
  - $a * b \rightarrow y$ is easy, where a and b are primes.
  - $y \rightarrow a$ and b is difficult
- ❑ **Modular Exponentiation** vs. ***Discrete Logarithm***
  - $f(x) = a^x \bmod n \rightarrow y$ is easy, Exponential Function
  - $y \rightarrow x = \log_a y \bmod (n-1)$

❑ Trapdoor One-Way Function

- ❑ One-Way Function, where some **trapdoor information** makes the function invertible
- ❑ **Power Function**, $y = f(x) = x^a \bmod n$, where **a** and **n** are given
  - $f^{-1}(y) = x$ is also *power function*, where $x = (x^a)^b \bmod n$

    $a * b \bmod \phi(n) = 1$
  - If $n = p*q$, where p and q are large primes,

    $f(x)$ is trapdoor function.
  - **Euler 정리**, $x^{\phi(n)} \bmod n = 1$, where $\phi(n) = (p-1) * (q-1)$

❑ 공개키 암호시스템 제약조건

- ❑ $m \rightarrow E_{ke}(m) = c$ is easy, but $c \rightarrow D_{kd}(m) = m$ is difficult without knowing D_{kd}.
- ❑ E_{ke} is a Trapdoor one-way Function
- ❑ D_{kd} is a Trapdoor Information

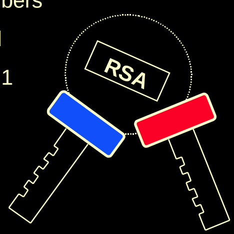
비밀성	$D_{kd}(E_{ke}(m)) = m$
인증 및 무결성	$E_{ke}(D_{kd}(m)) = m$
비밀성과 인증	$D_{kd}(E_{ke}(m)) = E_{ke}(D_{kd}(m)) = m$

- ❑ Domain of ***E*** and ***D*** should be large to avoid ***Dictionary Attack***

❑ RSA (Rivest-Shamir-Adleman)

❑ System 생성

- choose **2 primes p and q** and compute **n = p∗q**, $\phi(n) = (p-1) * (q-1)$, where p, q are 100-digit numbers
- choose ***e*** $\in [1, \phi(n)-1]$ such that $(e, \phi(n)) = 1$ and ***d*** $\in [1, \phi(n)-1]$ such that $e * d \bmod \phi(n) = 1$
- **(*e*,*n*) : public-key**
- **(*d*,*n*) : private-key**



❑ Encryption and Decryption

$m,c \in \{1,2,\ldots,n-1\}$

- **Encryption** : $c = m^e \bmod n$
- **Decryption** : $m = c^d \bmod n = m^{ed} \bmod n$, where $e * d \bmod \phi(n) = 1$

❑ RSA (Rivest-Shamir-Adleman)

❑ Example

- $p = 47$ and $q = 71$, $n = p*q = 3337$
- $(p-1)*(q-1) = 46 * 70 = 3220$, $GCD\,(e, (p-1)*(q-1)) = 1$
- Choose ***e*** at random to be 79
- ***d*** $= 79^{-1}$ mod $3220 = 1019$
- To encrypt message ***m* = 6882326879666683**

  $m_1 = 688$ $m_2 = 232$ $m_3 = 687$

  $m_4 = 966$ $m_5 = 668$ $m_6 = 003$

- $c_1 = m_1^e \bmod n = 688^{79} \bmod 3337 = 1570$
- ***c* = 1570 2756 2091 2276 2423 158**
- To decrypt, $m_1 = c_1^d \bmod n = 1570^{1019} \bmod 3337 = 688$

❑ Security of RSA

- ❑ **Factorization of *n = p*q***

  - n and e are known; $e*d \bmod (p-1)*(q-1) = 1$

- ❑ **Number Field Sieve, Quadratic Sieve, Elliptic Curve Method**
- ❑ (1983년) 69자리 10진수, Quadratic Sieve
- ❑ (1989년) 106자리 10진수, Quadratic Sieve
- ❑ (1994년) **RSA-129**, 129자리 10진수, Quadratic Sieve, 5000 mips-year

n = 114,381,625,757,888,867,669,235,779,976,146,612,010,218,296,
721,242,362,562,561,842,935,706,935,245,733,897,830,597,123,
563,958,705,058,989,075,147,599,290,026,879,543,541

- ❑ (1996년) **RSA-130**, 130자리 10진수, Number Field Sieve, 500 mips-year

❑ Security of RSA

① p와 q는 서로 크기가 거의 비슷하게 그리고 각각 10^{75} 내지는 10^{100}정도의 크기로 선정한다.

② $(p-1)$과 $(q-1)$은 각각 매우 큰 소인수를 가져야 한다.

③ $gcd(p-1, q-1)$은 작은 값이어야 한다.

❑ Security of RSA

- **Ciphertext-only Attack**

  Solving $c = m^e \bmod n$ is equivalent to taking roots modulo a composite number with unknown factorization, which is as difficult as the discrete logarithm.

- **Iterative Attack**

  Given (N, e, c), $c_1 = c^e \bmod N$, ..., $c_i = c_{i-1}^e \bmod N$
  If there is c_j in $c_1, c_2, \ldots, c_i, \ldots$ such that $c_j = c$,
  m is the same as c_{j-1} since $c_{j-1}^e \bmod N = c_j = c$.

- **Adaptively Chosen-Ciphertext Attack**

  $c = m^e \bmod n$, $x \in (0, n-1)$ $c' = cx^e \bmod n$
  $m' = (c')^d \bmod n = c^d(x^e)^d \bmod n = m \cdot x \bmod n$
  $m'x^{-1} \bmod n = m$

❑ Repeated Square-and-Multiply 반복 제곱-곱셈

```
z ← 1;
for i = k–1 downto 0 {
      z ← z² mod n;
      if e_i = 1 then z ← z · m mod n;
}
return ( z );
```

[예] 공개키 $e = 10$, 법 $n = 15$의 경우에 평문 $m = 3$에 해당하는 암호문 $c = m^e \mathit{mod}\ n = 9$는 아래과 같이 "<u>반복 제곱–곱셈</u>" 방식에 의해 계산된다. $e = 10$을 2진수로 표현하면 $(e_3\ e_2\ e_1\ e_0) = (1\ 0\ 1\ 0)$ 이다.

i	e_i	z
3	1	$1^2 \cdot 3 \ \mathit{mod}\ 15 = 3$
2	0	$3^2 \ \mathit{mod}\ 15 = 9$
1	1	$9^2 \cdot 3 \ \mathit{mod}\ 15 = 3$
0	0	$3^2 \ \mathit{mod}\ 15 = 9$

❑ Primality Testing

❑ Sieve of ERATOSTHENES

- Every positive integer > 1 has a prime divisor.
- If n is a composite, n has a prime factor not exceeding $\sqrt{n}$.

To determine if *n* is a prime, check *n* for divisibility by all primes not exceeding $\sqrt{n}$.

e.g. if n = 20 is a composite, it must have a prime factor less than $\sqrt{n}$ = 4. Since n = 20 has a prime factor 2, it is not a prime.

❑ Probabilistic Primality Testing

❑ Miller_Rabin(*n, k*) Algorithm

```
① find r and s  such that n–1 = 2^s·r ;
   for i = 1 to k  do {
②       choose a random integer a, 1 ≤ a ≤ n–2
③       y ← a^r mod n;
④       if y ≠ 1 and y ≠ n–1 then {
⑤           j ← 1;
            while ( j ≤ s–1 and y ≠ n–1 ) do {
⑥                 y ← y^2 mod n;
⑦                 j ← j+1; }
⑧           if y ≠ n–1 then return( "composite" ); } }
⑨  return( "prime" );
```

소수 $n > 2$과 홀수 r에 대해서 $n - 1 = 2^s \cdot r$이라고 하자. 이때, $gcd(a, n) = 1$을 만족하는 임의의 정수 a에 대해서 a^r $mod\ n = 1$이 성립 또는 $0 \le j \le s-1$범위내의 어떤 j에 대해서 $a^{2^j r}$ $mod\ n = -1$.

n이 홀수의 합성수임에도 불구하고 $2 \le a \le n-1$범위내의 임의의 정수 a에 대해서 a^r $mod\ n = 1$ 이 성립 또는 $0 \le j \le s-1$범위내의 어떤 j 에 대해서 $a^{2^j r}$ $mod\ n = -1$이 성립한다면 n은 실제로 소수는 아니지만 소수인 것처럼 보이는 강한 유사소수(strong pseudo-prime)가 되고, a는 n을 소수라고 잘못 판단하게 하는 강한 거짓 증거(strong liar)가 된다.

암호이론과 보안 공개키 암호

❑ Probabilistic Primality Testing

❑ Miller_Rabin(*n, k*) Algorithm

알고리즘의 결과가 *Miller_Rabin*(n, k) = "*composite*"일 경우에는 n은 분명히 합성수이다. 따라서, 이 경우에는 다른 임의의 홀수 n을 선정하여 알고리즘을 처음부터 다시 시작하게 된다. *Miller_Rabin*(n, k) = "*prime*"일 경우에는 2가지의 가능성이 존재한다.

첫째는 n이 실제로 소수인 경우와 둘째는 n이 합성수이면서도 소수로 판정되는 경우이다. 후자의 경우에는 선정된 a가 n을 소수라고 잘못 판단하게 하는 강한 거짓증거가 된다.

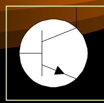
n이 홀수의 합성수이면 $2 \le a \le n-1$범위내의 정수들 중에서 기껏해야 1/4정도가 강한 거짓증거가 되기 때문에 한 개의 a에 대해서 합성수(composite) n을 소수로 판정할 확률은 *Miller–Rabin* 판정법에서는 1/4이하가 된다.

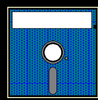
따라서, 선택된 n이 임의로 선정된 여러 개의 a 값에 대하여 *Miller–Rabin* 판정법을 통과한다면 매우 높은 확률로 n을 소수로 판정할 수가 있게 된다. 따라서, k번을 반복하여 *Miller–Rabin* 판정법을 통과한다면 n이 소수일 확률은 $1 - (1/4)^k$ 이상이 된다.

❑ Speed of RSA

- (1995) 1 Mbps with 512-bit modulus
- **Hardware 구현**, DES 보다 1000배 느림
- **Software 구현** with 8-bit public-key on SPARC II

	512-bit	768-bit	1024-bit
Encrypt	0.03sec	0.05sec	0.08sec
Decrypt	0.16sec	0.48sec	0.93sec





❑ Standard and Patent

- **RSA Data Security Inc**., (Sep. 20, 2000) in USA
- French and Australian Banks 표준
- ISO de facto standard

❑ 이산대수 문제(discrete logarithm problem)

❑ 정의

소수 p

곱셈 상의 군 Z_p*의 생성자 g

원소 $y \in Z_p^*$

$y = g^x \bmod p$를 만족하는 $1 \le x \le p-2$범위 내의 $x = \log_g y \bmod (p-1)$을 구하는 것.

❑ Algorithms

[1] *Compute* $g^0, g^1, g^2, \ldots$ until $y = g^x \bmod n$ so that x can be obtained

[2] Shanks Algorithm

[3] Pohlig-Hellman Algorithm

[4] Index-Calculus Algorithm

❑ Shanks Algorithm

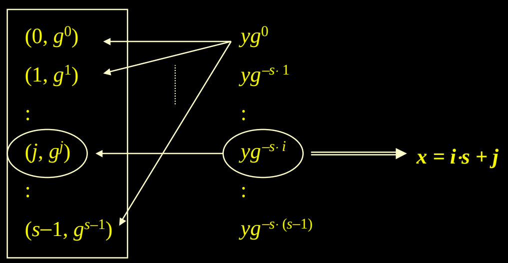
소수 p, 곱셈 상의 군 Z_p*의 생성자 g, g의 위수 $t = p\text{-}1$, $s = \lceil \sqrt{t} \rceil$

$y = g^x \bmod p$를 만족 $1 \le x \le p-2$ $x = \log_g y \bmod (p-1)$을 구하는 것

$$x = i \cdot s + j, \text{ where } i, j \in [0, s)$$

$$y = g^x = g^{is} g^j = g^{is+j}$$

$$y(g^{-s})^i = g^j$$



❑ ElGamal 공개키 암호

❑ System 생성

(i) 매우 큰 소수 p

(ii) 곱셈상의 군 Z_p*의 생성자 g

(iii) 임의의 난수 $x \in [\,1, p-2\,]$ 생성, $y = g^x \bmod p$를 계산.

❑ 공개키 k_e는 $\{\, y, g, p \,\}$, 개인키 k_d는 $\{\, x \,\}$.

❑ 암호화와 복호화

난수 $x \in [1, p-2]$ 생성 (Randomized Encryption)

$E_{ke}(\, m\,) = c = \{\, c_1, c_2 \,\} = \{\, g^x \bmod p, y^x\, m \bmod p \,\}$

$D_{kd}(\, c\,) = c_2\, c_1^{-x} \bmod p = m$

❑ Diffie-Hellman 공개키 분배

❑ **System 생성**

(i) 매우 큰 소수 p
(ii) 곱셈상의 군 Z_p^*의 생성자 g

난수 x_A — **Alice**

난수 x_B — **Bob**

Alice → Bob: $y_A = g^{x_A} \bmod p$

Bob → Alice: $y_B = g^{x_B} \bmod p$

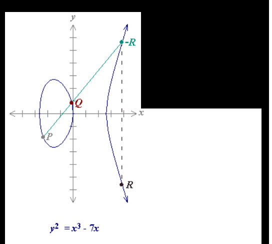
Alice: $k = (y_B)^{x_A} \bmod p = g^{x_B x_A} \bmod p$

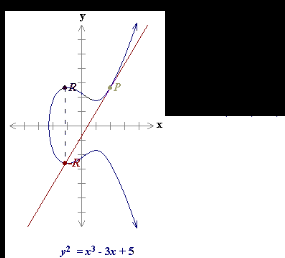
Bob: $k = (y_A)^{x_B} \bmod p = g^{x_A x_B} \bmod p$

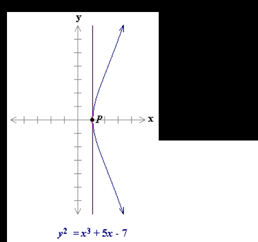
암호이론과 보안 공개키 암호

❑ Elliptic Curve (타원곡선) 공개키 암호

실수(real number)상에서의 타원곡선(elliptic curve)은 a와 b가 고정된 실수일 경우에 방정식 $y^2 = x^3 + ax + b$을 만족하는 (x, y) 점들의 집합을 의미한다.

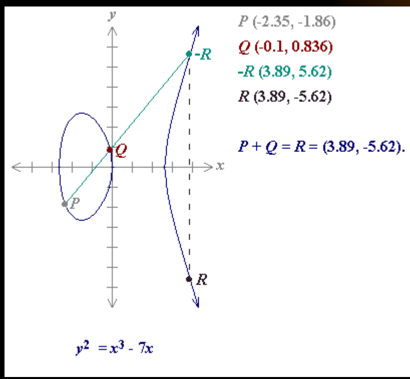






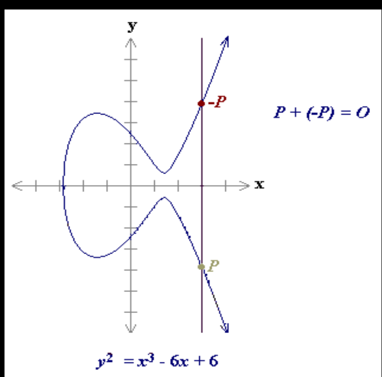
Java Animation : ecc-java1.htm

❑ Addition of point *P* and point *Q*



암호이론과 보안 공개키 암호

❑ Addition of *P* and *–P*



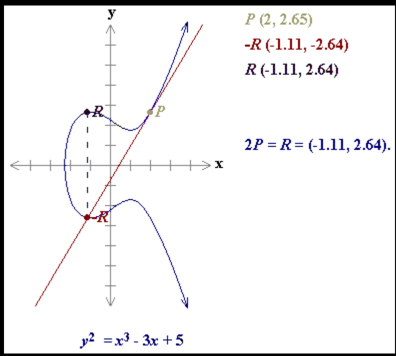
By definition, $P + (-P) = O$.

As a result of this equation,
$P + O = P$ in the elliptic curve group .

O = the additive identity of the elliptic curve group;

All elliptic curves have an additive identity.

❑ Doubling the point *P*



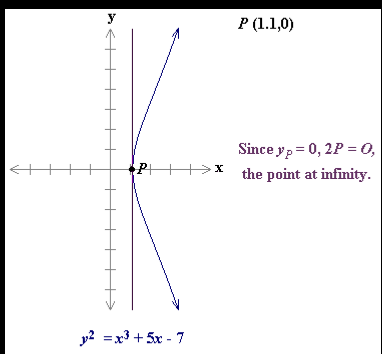
$P = (x, y)$

If $y \neq 0$, then P 에서의 접선은 정확히 타원곡선의 다른 한 점과 교차.

the law for doubling a point on an elliptic curve group :

$P + P = 2P = R$

❑ Doubling the point *P* if *y* = 0

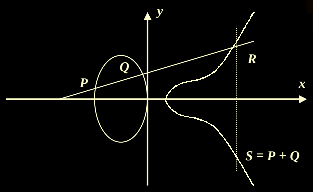


By definition, $2P = O$ for such a point P.

To find $3P$ in this situation,
one can add $2P + P$. Then,
$P + O = P$

Thus $3P = P$.
$3P = P$, $4P = O$, $5P = P$, $6P = O$, $7P = P$, …

❑ Elliptic Curve Group (타원곡선 군)



첫째, 무한대 점은 O으로 표기하며 타원곡선 위의 임의의 점 P에 대해서 $P + O = P$가 성립된다. 즉 무한대 점은 덧셈상의 항등원(identity)이 된다.

둘째는 타원곡선상의 임의의 점 P에 대해서 $P + Q = O$을 만족하는 점 Q가 존재하는데

$Q = -P$로 나타내며 뺄셈 $R-S$는 $R + (-S)$로 정의된다. 점 $-P$는 점 P의 x축에 대한 대칭점이 되기 때문에 점 P의 좌표가 (x, y)이면 점 $-P$의 좌표는 $(x, -y)$가 된다.

셋째, 타원곡선상의 임의의 점 P와 Q에 대해서 $P + Q = Q + P$가 성립한다.

넷째, 타원곡선상의 임의의 점 P, Q와 R에 대해서 $P + (Q + R) = (P + Q) + R$이 성립한다.

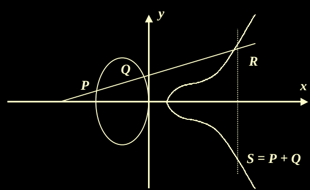
❑ Elliptic Curve (타원곡선) 공개키 암호

(x_1, y_1), (x_2, y_2), (x_3, y_3)을 점 P, Q, $S = P + Q$의 좌표. 점 $P + Q = S$ 좌표 (x_3, y_3)을 x_1, y_1, x_2, y_2로 표현.

$y = \alpha x + \beta$를 점 P와 Q를 지나는 선의 방정식,
$\alpha = (y_2 - y_1) / (x_2 - x_1)$, $\beta = y_1 - \alpha x_1$.

만약, $(\alpha x + \beta)^2 = x^3 + ax + b$이면 점 P와 Q를 지나는 선 위의 점 $(x, \alpha x + \beta)$는 타원곡선상의 점이 된다.

3차 방정식 $x^3 - (\alpha x + \beta)^2 + ax + b$의 각각의 근(root)에 대하여 한 개의 교차점이 존재하게 된다.
$(x_1, \alpha x_1 + \beta)$와 $(x_2, \alpha x_2 + \beta)$는 타원곡선상의 두 점 P와 Q이기 때문에 x_1과 x_2가 3차 방정식의 근.



$$x_1 + x_2 + x_3 = \alpha^2$$

결국 나머지 근은 $x_3 = \alpha^2 - x_1 - x_2$이 되고 점 $P + Q$의 좌표 (x_3, y_3)

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \qquad y_3 = -y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3)$$

❑ Elliptic Curve (타원곡선) 공개키 암호

$P + Q$에서 $P = Q$가 되면 α는 점 P에서의 미분 값이 되기 때문에

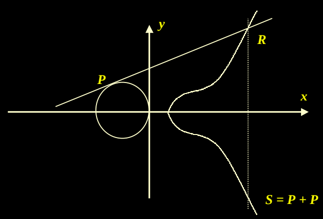
$y^2 = x^3 + ax + b$을 음함수 미분하면

$2yy' = 3x^2 + a$가 되고 점 $P = (x_1, y_1)$에서의 기울기는 $\alpha = (3x_1^2 + a) / 2y_1$이 된다.

점 $P + Q = S$의 좌표 (x_3, y_3)을 x_1, y_1, x_2, y_2를 통해서 표현하면 다음과 같다.

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1$$

$$y_3 = -y_1 + \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3)$$



❑ 유한체(finite Field) 상에서의 타원곡선

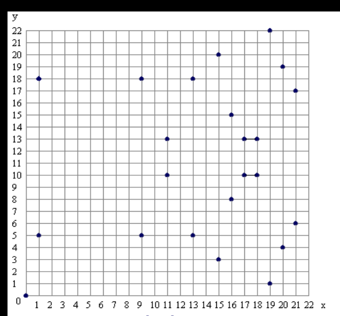
p가 소수일 때 유한 체 $GF(p)$상에서의 타원곡선(elliptic curve)은 $a, b \in GF(p)$의 경우에 방정식 $y^2 \bmod p = x^3 + ax + b \bmod p$를 만족하는 (x, y) 점들의 집합과 무한대 점 O을 의미한다. 물론, $x, y \in GF(p)$

$a = 1$ and $b = 0$일 때, 타원곡선 $y^2 = x^3 + x$.
점 (9,5)은 위의 식을 만족 because :

$y^2 \bmod p = x^3 + x \bmod p$
$5^2 \bmod 23 = 9^3 + 9 \bmod 23$
$25 \bmod 23 = 738 \bmod 23$
$2 = 2$

The 23 points which satisfy this equation are:

(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)



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

Java Animation : ecc-java2.htm

❑ 타원곡선 이산대수 문제

- $GF(p)$, where p is a prime(소수)
- 타원곡선상의 점 P의 위수(order) t : $tP = 0$을 만족하는 가장 작은 정수 t
- $GF(p)$상에서 정의된 타원곡선상의 점 Q, 위수가 t 인 점 P,

  $Q = xP$를 만족하는 정수 $x \in [0, t-1]$를 구하는 문제

  P, $2P = P + P$, $3P = P + P + P$, …를 연속적으로 계산

$$\boxed{y^2 \; mod \; 23 = x^3 + 9x + 17 \; mod \; 23,}$$

What is the discrete logarithm x of $Q = (4,5)$ to the base $P = (16,5)$? $Q = xP$

$P = (16,5)$ $2P = (20,20)$ $3P = (14,14)$ $4P = (19,20)$ $5P = (13,10)$
$6P = (7,3)$ $7P = (8,7)$ $8P = (12,17)$ $9P = (4,5)$

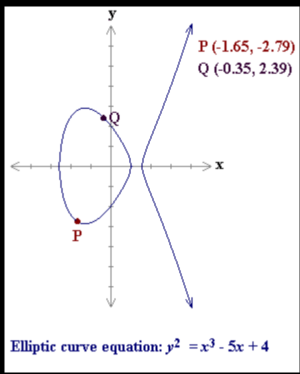
Since $9P = (4,5) = Q$, the discrete logarithm of Q to the base P is $x = 9$.

❑ 반복 2배수 -덧셈

- $2P = P + P$
- $100P = 2(\,2(\,P + 2(\,2(\,2(\,P + 2P\,)\,)\,)\,)\,)$

암호이론과 보안 공개키 암호

❑ 타원곡선 이산대수 문제



What is the discrete logarithm of Q(-0.35,2.39) to the base P(-1.65,-2.79) in the elliptic curve group $y^2 = x^3 - 5x + 4$ over real numbers?

Java Animation : ecc-java3.htm

암호이론과 보안 공개키 암호

❑ 타원곡선 공개키 암호

ElGamal 공개키 암호	타원곡선 공개키 암호
공개키 : $\{ g, p, y = g^x \bmod p \}$	$\{ G, p, Y = xG \}$
개인키 : $\{ x \}$	$\{ x \}$
암호화 : $[c_1, c_2] = [g^x \bmod p, y^x m \bmod p]$	$[C_1, C_2] = [xG, xY + M]$
복호화 : $c_2 c_1^{-x} \bmod p$	$C_2 - xC_1$

[예] $p = 11$, $a = 1$, $b = 6$인 경우에 $GF(p)$상에서 타원곡선 $y^2 = x^3 + x^2 + 6$이 정의되고 타원곡선상의 점들은 다음과 같다.

(2, 4)	(2, 7)	(3, 5)	(3, 6)	(5, 2)	(5, 9)
(7, 2)	(7, 9)	(8, 3)	(8, 8)	(10, 2)	(10, 9)

수신자의 개인키 $x = 7$, 공개키 $G = (2, 7)$, $Y = 7(2, 7) = (7, 2)$가 주어졌을 때, 송신자는 임의의 난수 $x = 3$을 선정하여 평문 $M = (10, 9)$을 암호문 $C_1 = 3(2, 7) = (8, 3)$, $C_2 = 3(7, 2) + (10, 9) = (10, 2)$로 전환한다.

❑ 타원곡선 이산대수와 소인수 분해에 소요되는 계산량

1–mips 컴퓨터가 1초에 40,000번의 타원곡선 덧셈을 수행할 수가 있다고 가정.

1–mips 컴퓨터는 1년 동안에 $(40,000)(60\times60\times24\times365) \approx 2^{40}$번의 타원곡선 덧셈을 수행.

왼쪽 표는 여러 개의 *t*값에 대해서 *Pollard*의 *rho* 알고리즘을 적용 할 때, 한 개의 타원곡선 이산대수를 계산하기 위해 소요되는 계산량을 나타낸다.

예를 들면 1,000–mips 컴퓨터 10,000대를 이용하여 $t = 2^{160}$인 경우의 타원곡선 이산대수를 구하기 위해서는 96,000–year가 소요된다

p(bits)	*t* (bits)		mips–year
163	160	2^{80}	9.6×10^{11}
191	186	2^{93}	7.9×10^{15}
239	234	2^{117}	1.6×10^{23}
359	354	2^{177}	1.5×10^{41}
431	426	2^{213}	1.0×10^{52}

n (bits)	mips–year
512	3×10^{4}
768	2×10^{8}
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

❑ ECC, RSA, DSA 안전성 비교분석