7장 공개키 암호 알고리즘

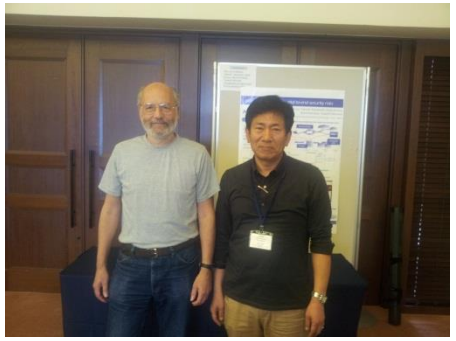
정보보호이론

Spring 2015

고려대학교
KOREA UNIVERSITY

7.1 RSA 암호

- 가장 많이 사용되고 있는 공개키 암호시스템
  - Rivest, Shamir, and Adleman 의 이름에서 RSA
  - Clifford Cocks, an English mathematician working for the UK intelligence agency, described an equivalent system in 1973, but it was mostly considered a curiosity and, as far as is publicly known, was never deployed. His discovery, however, was not revealed until 1998 due to its top-secret classification, and Rivest, Shamir, and Adleman devised RSA independently of Cocks' work.



Shamir & Lee

7.1 RSA 암호

키 생성

1. 서로 다른 두 소수 p와 q 선택 (크기가 동일한 1024비트 이상의 수로 선택) ; ($P(k \text{ is prime}) \approx \frac{2}{\ln(2^{1024})} = \frac{2}{1024 \ln(2)} \approx \frac{1}{355}$)
2. $n = p \times q$값을 계산.
3. $\varphi(n) = (p-1)(q-1)$
4. $1 < e < \varphi(n) - 1$의 범위에서 $\varphi(n)$과 서로소인 e를 선택
5. $d = e^{-1} \, mod \, \varphi(n)$ (확장 유클리드 알고리즘)

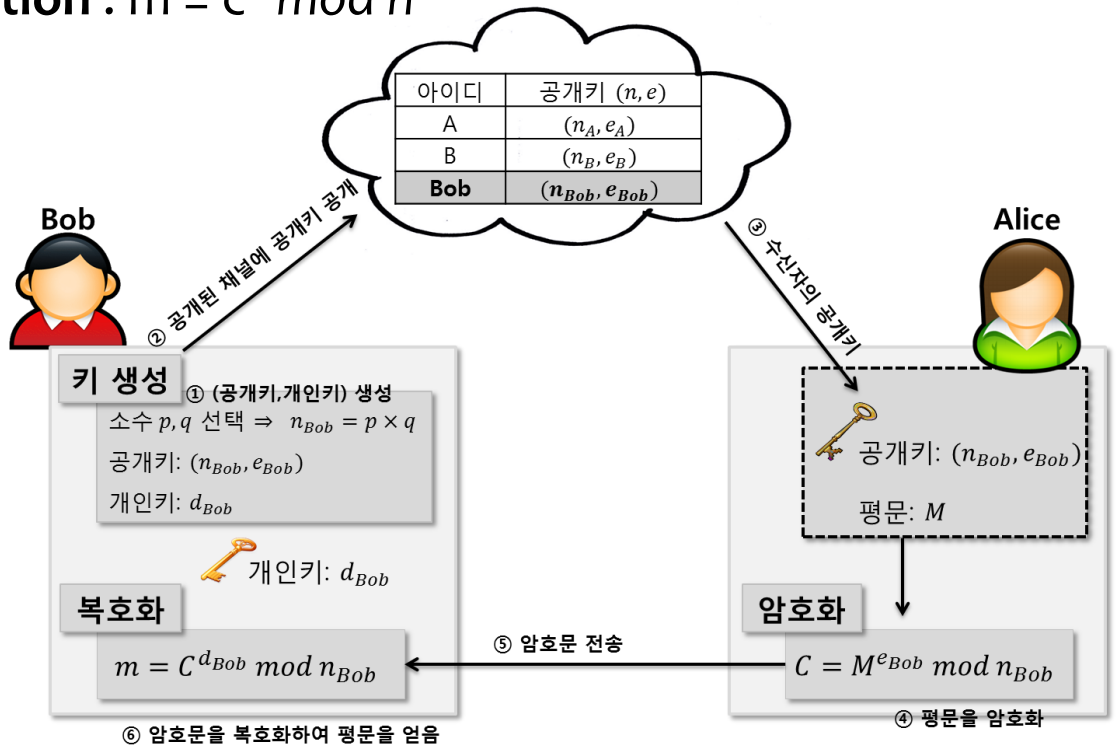
- **(e,n) : public-key**
- **(d,n) : private-key**

7.1 RSA 암호

- 예)
  1. $p = 127,\ q = 131$
  2. $n = p \times q = 127 \times 131 = 16637$
  3. $\varphi(n) = \varphi(p \times q) = \varphi(p) \times \varphi(q) = (p-1) \times (q-1) =$
     $126 \times 130 = 16380$
  4. 공개키 e는 집합 $\mathbb{Z}^*_{\varphi(n)}$에서 $gcd\big(e, \varphi(n)\big) = 1$을 만족하는 $e = 17$로 선택
  5. $d \equiv e^{-1} \equiv 17^{-1} \equiv 14453\ (mod\ 16380)$
  6. 공개키 $(\boldsymbol{n = 16637, e = 17})$, 개인키 $(\boldsymbol{d = 14453})$

7.1 RSA 암호

- 암 · 복호화
  - **Encryption** : $c = m^e \bmod n$
    - *Note m < n (for uniqueness)*
  - **Decryption** : $m = c^d \bmod n$



7.1 RSA 암호

- RSA암호의 정확성(correctness)
  - **Decryption** : $c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}$
  - $ed \equiv 1 \bmod \varphi(n) \Longrightarrow ed = k\varphi(n) + 1$

$$\therefore m^{ed} \equiv m^{k\varphi(n)+1} \equiv \left(m^{\varphi(n)}\right)^k \cdot m \pmod{n}$$

$$\equiv 1^k \cdot m \pmod{n}$$

(오일러 정리 $m^{\varphi(n)} \equiv 1 \bmod n$)

$$\equiv m \pmod{n}$$

7.1 RSA 암호

- 예
  - 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* = 688232687966668 3**
    - m_1 = 688 m_2 = 232 m_3 = 687
      m_4 = 966 m_5 = 668 m_6 = 003
    - $c_1 = m_1^{e}\ mod\ n = 688^{79}\ mod\ 3337$ = 1570
    - ***c* = 1570 2756 2091 2276 2423 158**
    - To decrypt, $m_1 = c_1^{d}\ mod\ n = 1570^{1019}\ mod\ 3337$ = 688

7.1 RSA 암호

- RSA암호의 안전성
  - **RSA 문제 :** $c \equiv m^e \pmod{n}$가 주어졌을 때 c의 e^{th} root를 구하는 문제
    - 인수분해 문제 → The RSA 문제
      - $n = p \times q$을 인수분해 → $\varphi(n)(= (p-1)(q-1))$ → e의 곱셈상의 역원 d를 계산
    - The RSA 문제 → 인수분해 문제
      - Not known yet

7.1 RSA 암호

- 효율적인 RSA 암·복호화
  - 두 소수 p와 q는 1024비트 이상의 수 → e, d > 2048 비트
    - 제곱-곱 연산 방법(Square-and-Multiply Method) : 지수가 2048 비트인 경우 2048번의 제곱과 평균 1024번의 곱셈

1. 첫 번째 방법: 공개키 e로 3, 17, 65537을 사용
   - $3 = 11_{(2)}$, $17 = 10001_{(2)}$, $65537 = 1000000000000001_{(2)}$
   - 3은 2번의 (제곱, 혹은 곱셈) 연산, 17은 5번의 연산, 65537은 17번의 연산
   - 개인키 d를 사용하는 복호화 과정을 현저하게 향상시키는 방법은 아직 존재하지 않는다

7.1 RSA 암호

- 효율적인 RSA 암·복호화
  - 2. 두 번째 방법 : CRT를 이용한 복호화
    - $m = c^d \bmod n$
      1. $m_1 = c^d \bmod p,\ m_2 = c^d \bmod q$
      2. $M = p \times q = n \rightarrow M_1 = \frac{n}{p} = q,\ M_2 = \frac{n}{q} = p$
      3. $m = (m_1 M_1^{-1} M_1 (\bmod p) + m_2 M_2 M_2^{-1} (\bmod q)) \ (\bmod M)$
    - n이 k비트인 경우 CRT를 이용하는 경우의 연산은 $k/2$ 비트 수들(즉 $\bmod p$, 혹은 $\bmod q$)의 제곱과 곱셈연산
    - 기존 복호화 과정보다 약 4배 정도 향상
    - 이 경우 p와 q를 안전하게 저장 필요

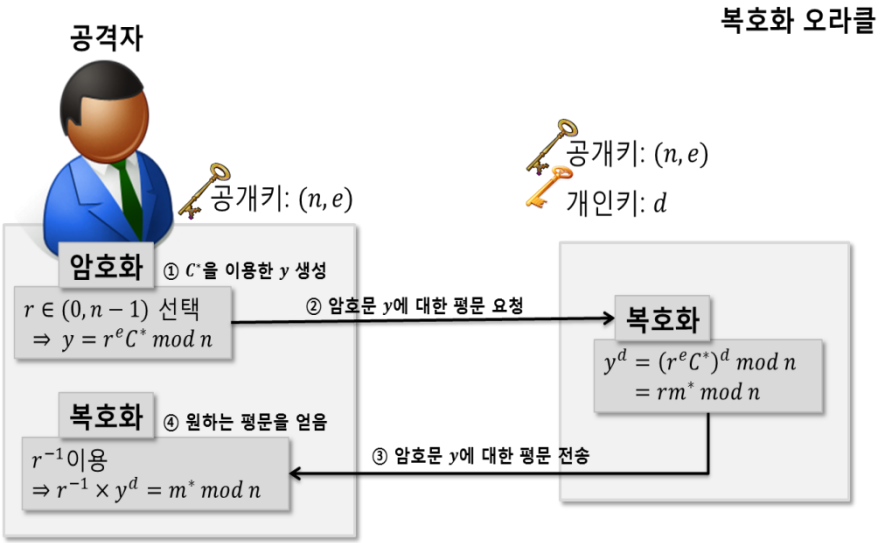
7.1 RSA 암호

- 효율적인 RSA 암·복호화
  - 2. 두 번째 방법 : CRT를 이용한 복호화
    - $n = p \times q = 127 \times 131 = 16637,\ d = 14453,\ c = 8806$
    1. $m_1 \equiv c^d \equiv 8806^{14453} \equiv 12 (\mathrm{mod}\ 127)$,
    2. $m_2 \equiv c^d \equiv 8806^{14453} \equiv 8\ (\mathrm{mod}\ 131)$
    3. $M_1 = \frac{n}{p} = q = 131,\ M_2 = \frac{n}{q} = p = 127$
    4. $M_1^{-1} \equiv 32 (\mathrm{mod}\ p),\ M_2^{-1} \equiv 98 (\mathrm{mod}\ q)$
    5. $m \equiv (m_1 M_1 (M_1^{-1} \bmod p) + m_2 M_2 (M_2^{-1} \bmod q))\ (\mathrm{mod}\ n)$
       $\equiv (12 \times 131 \times 32 + 8 \times 127 \times 98)\ (\mathrm{mod}\ 16637)$
       $\equiv (50304 + 99568)\ (\mathrm{mod}\ 16637)$
       $\equiv \mathbf{139}\ (\mathbf{mod\ 16637})$

7.1 RSA 암호

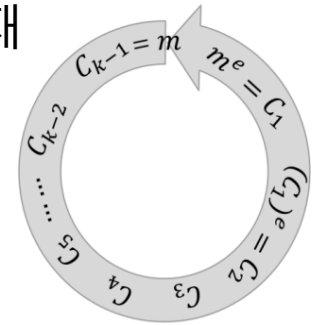
- RSA 암호에 대한 공격
  - 선택 암호문 공격(Chosen Ciphertext Attack)
    - RSA의 준동형사상(Homomorphism)의 성질을 이용
      - $(r \times m^*)^e = (r)^e \times (m^*)^e$
    - 공격자는 암호문 c^*의 평문을 목표

1. $r \in (0, n-1),\ \ y \equiv r^e c^* (\mathrm{mod}\, n)$
2. CCA를 통해 새로운 암호문 y에 대한 평문 y^d을 얻는다
3. $y^d \equiv (r^e c^*)^d \equiv r^{ed} c^{*d} \ (\mathrm{mod}\, n)$
   $\equiv r c^{*d}$
   $\equiv r(m^{*e})^d \ (\mathrm{mod}\, n)$



7.1 RSA 암호

- 암호화 지수 e에 대한 공격
  - $e = 3$ and $\{n_1, n_2, n_3\}$ relative primes, 동일한 m을 3명에게 전송하는 경우
  - 공격자는 $c_1 \equiv x(\mathrm{mod}\, n_1),\ c_2 \equiv x(\mathrm{mod}\, n_2),\ c_3 \equiv x\ (\mathrm{mod}\, n_3)$을 계산
  - CRT를 이용하여 $c^* \equiv x\ (\mathrm{mod}\, n_1 n_2 n_3)$을 계산
  - $m^3 < n_1 n_2 n_3$ 이기 때문에 $x = m^3$
- 복호화 지수 d에 대한 공격
  - **If d is revealed,** regenerate p, q, n, e, and d
- 평문 공격(Plaintext Attack)
  - **순환 공격 :** $c_1 \equiv c^e\ (\mathrm{mod}\, n),\ c_2 \equiv c_1^e \equiv (c^e)^e\ (\mathrm{mod}\, n),\ \ldots,\ c = c_k \bmod n$ → $c_{k-1} = m$
  - 암호는 평문공간에 대한 치환이기 때문에 c_k는 반드시 존재
  - 소인수분해와 동일한 복잡도



7.1 RSA 암호

- 공통 법 공격(Common Modulus Attack)
  - 공격자는 두 수신자의 동일한 메시지의 두 암호문 c_1, c_2 와 수신자의 공개키 e_1, e_2가 서로소임을 알고 있다.
    1. $re_1 + se_2 = 1$을 계산
    2. r이 음수라고 가정 (r과 s 중 하나는 반드시 음수)
    3. 확장 유클리드 알고리즘을 통해 c_1^{-1}을 계산한다.
    - $\gcd(c_1, n) = 1$ → c_1^{-1}이 존재
    - $\gcd(c_1, n) \neq 1$ → c_1이 p 또는 q의 배수(유클리드 알고리즘으로 p 또는 q를 구함)
    4. $(c_1^{-1})^{-r} \cdot c_2^{s} \equiv (m^{-e_1})^{-r} \cdot m^{e_2 \cdot s} \equiv m^{re_1 + se_2} \equiv m \pmod{n}$

  Moral: Never share a common modulus.

  - 예) $n = 35$, $e_1 = 5$와 $e_2 = 11$, $c_1 \equiv 17 \ (mod\ 35)$와 $c_2 \equiv 3 \ (mod\ 35)$
    1. $5 \times (-2) + 11 \times 1 = 1$
    2. $c_1^{-1} \equiv 33 \pmod{35}$
    3. $(c_1^{-1})^{-r} \cdot c_2^{s} \equiv 33^2 \cdot 3^1 \equiv 3267 \equiv 12 \pmod{35}$

7.1 RSA 암호

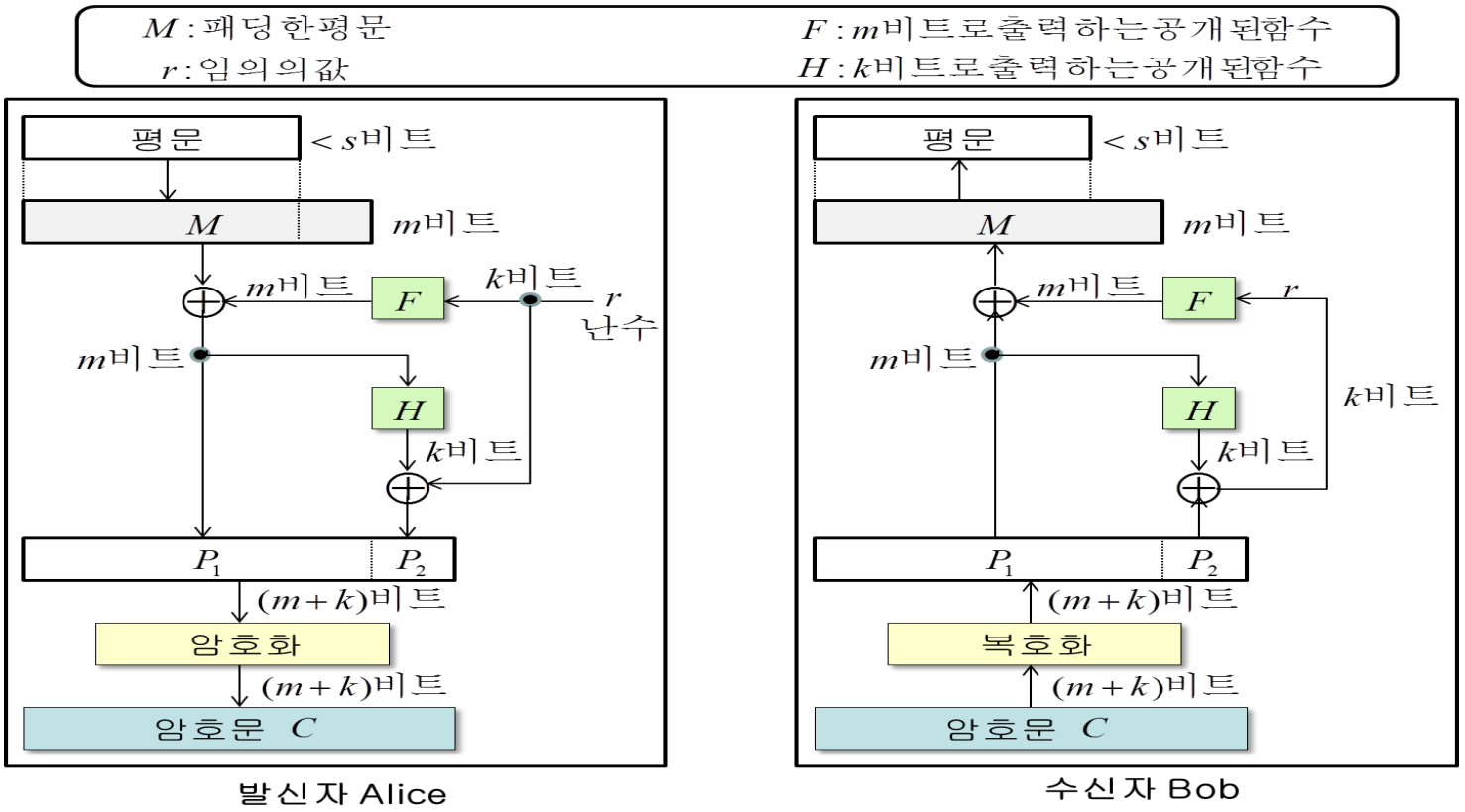
- 부채널 공격(Side Channel Attack)
  - 시간차 공격(Timing Attack)
    - 방어 방법
    1. 지수 계산 할 때 각각의 지수 계산에 동일한 시간을 걸리도록 만든다.
    2. 암호문을 복호화하기 전에 난수를 곱하는 블라인딩 기법을 사용한다.
  - 전력차 공격(Power Analysis Attack)

7.1 RSA 암호

- RSA 이용 시 권고사항
  1. n의 비트는 적어도 (서명의 경우) 2048비트가 되어야 한다.
  2. 서로 다른 두 소수 p와 q는 적어도 1024비트 이상이 되어야 한다.
  3. 서로 다른 두 소수 p와 q가 너무 가까이 있는 소수를 선택하지 않는다.
  4. $p-1$과 $q-1$은 적어도 하나의 큰 소인수를 가져야 한다.
  5. 비율 $\frac{p}{q}$가 작은 분자나 작은 분모를 갖는 유리수와 가까이 있으면 안 된다.
  6. n을 공통적으로 이용하지 않는다.
  7. 공개키 e는 $2^{16}+1=65537$을 이용하거나 혹은 65537과 가까이 있는 값을 이용한다.
  8. 만약 개인키 d가 노출되었을 경우, 수신자는 반드시 공개키 n과 e, 개인키 d를 즉시 교체해야 한다.
  9. OAEP를 이용

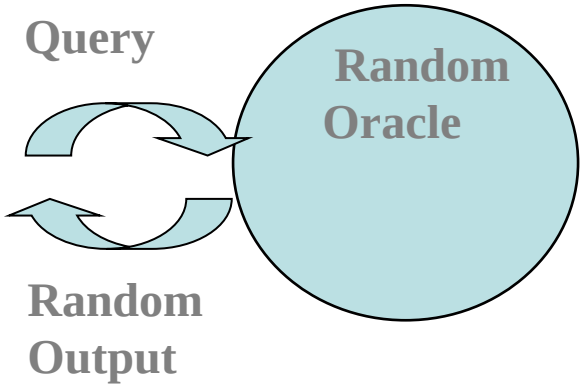
7.1 RSA 암호

- OAEP(Optimal Asymmetric Encryption Padding)
  - C=(P1 || P2)^e where P1 = (M||0^{k1})⊕F(r), P2 = H(P1)⊕ r
  - |P1| = m, |P2| =k



7.1 RSA 암호

- OAEP(Optimal Asymmetric Encryption Padding)
  - The original version of OAEP was proved in the random oracle model to be IND-CCA2 secure when OAEP is used with the RSA permutation, i.e., RSA-OAEP. An improved scheme (called OAEP+) that works with any trapdoor one-way permutation was offered by Victor Shoup. More recent work has shown that in the standard model, it is impossible to prove the IND-CCA2 security of RSA-OAEP under the assumed hardness of the RSA problem.
- random oracle model
  1. A mathematical function mapping every possible query to a random response from its output domain.
  2. Used when no known implementable function provides the mathematical properties required by the proof.



7.1 RSA 암호

- RSAES-PKCS#1(v1.5)
  - 특　　징 : 초기 PKCS#1 기술규격 공개키로 평문을 암호화하기 위한 EM(암호메시지) 구조 및 RSA연산방법이 기술되어있음
  - 보 안 성 : 1998년 Bleichenbacher는 선택된 암호문 공격을 통해 평문을 얻어낼 수 있는 확률을 높일 수 있음; 실제 암호 공간이 1/2로 줄어드는 보안 취약점이 존재
  - PKCS : Public-Key Cryptography Standard published by RSA Laboratories.

PKCS

PKCS Standards Summary

	Version	Name	Comments
PKCS #1	2.1	RSA Cryptography Standard[1]	See RFC 3447. Defines the mathematical properties and format of RSA public and private keys (ASN.1-encoded in clear-text), and the basic algorithms and encoding/padding schemes for performing RSA encryption, decryption, and producing and verifying signatures.
PKCS #2	-	*Withdrawn*	No longer active as of 2010. Covered RSA encryption of message digests; subsequently merged into PKCS #1.
PKCS #3	1.4	Diffie–Hellman Key Agreement Standard[2]	A cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel.
PKCS #4	-	*Withdrawn*	No longer active as of 2010. Covered RSA key syntax; subsequently merged into PKCS #1.
PKCS #5	2.0	Password-based Encryption Standard[3]	See RFC 2898 and PBKDF2.
PKCS #6	1.5	Extended-Certificate Syntax Standard[4]	Defines extensions to the old v1 X.509 certificate specification. Obsoleted by v3 of the same.
PKCS #7	1.5	Cryptographic Message Syntax Standard[5]	See RFC 2315. Used to sign and/or encrypt messages under a PKI. Used also for certificate dissemination (for instance as a response to a PKCS#10 message). Formed the basis for S/MIME, which is as of 2010 based on RFC 5652, an updated Cryptographic Message Syntax Standard (CMS). Often used for single sign-on.
PKCS #8	1.2	Private-Key Information Syntax Standard[6]	See RFC 5208. Used to carry private certificate keypairs (encrypted or unencrypted).
PKCS #9	2.0	Selected Attribute Types [7]	See RFC 2985. Defines selected attribute types for use in PKCS #6 extended certificates, PKCS #7 digitally signed messages, PKCS #8 private-key information, and PKCS #10 certificate-signing requests.
PKCS #10	1.7	Certification Request Standard[8]	See RFC 2986. Format of messages sent to a certification authority to request certification of a public key. See certificate signing request.
PKCS #11	2.20	Cryptographic Token Interface[9]	Also known as "Cryptoki". An API defining a generic interface to cryptographic tokens (see also Hardware Security Module). Often used in single sign-on, Public-key cryptography and disk encryption[10] systems.
PKCS #12	1.0	Personal Information Exchange Syntax Standard[11]	Defines a file format commonly used to store private keys with accompanying public key certificates, protected with a password-based symmetric key. PFX is a predecessor to PKCS#12. This container format can contain multiple embedded objects, such as multiple certificates. Usually protected/encrypted with a password. Usable as a format for the Java key store and to establish client authentication certificates in Mozilla Firefox. Usable by Apache Tomcat, but not by Apache HTTP Server.
PKCS #13	–	Elliptic Curve Cryptography Standard [12]	*(Under development as of 2011.)*[13]
PKCS #14	–	Pseudo-random Number Generation	*(Under development as of 2011.)*[13]
PKCS #15	1.1	Cryptographic Token Information Format Standard[14]	Defines a standard allowing users of cryptographic tokens to identify themselves to applications, independent of the application's Cryptoki implementation (PKCS #11) or other API. RSA has relinquished IC-card-related parts of this standard to ISO/IEC 7816-15.[15]

7.2 RABIN 암호

- RSA 암호 시스템에서 공개키 $e = 2$로 고정한 경우
  - 키생성
    1. $k \in \mathbb{Z}, 4k + 3$인 서로 다른 두 소수 p와 q를 선택
    2. $n = p \times q$
    3. (p, q) 개인키,, n 공개키
  - 암호화
    - $c \equiv m^2 \pmod{n}$

7.2 RABIN 암호

- 복호화

$$x \equiv a^{(p+1)/4} \pmod{p} \quad \text{and} \quad x \equiv -a^{(p+1)/4} \pmod{p}$$

1. 복호화 (p와 q가 $4k+3$형태임을 이용)
   - $a_1 \equiv c_1^{\frac{p+1}{4}} \pmod{p}$, $a_2 \equiv -c_1^{\frac{p+1}{4}} \pmod{p}$
   - $b_1 \equiv c_2^{\frac{q+1}{4}} \pmod{q}$, $b_2 \equiv -c_2^{\frac{q+1}{4}} \pmod{q}$
2. CRT를 이용
   - $P_1 = \mathrm{CRT}(a_1, b_1, p, q)$,
     $P_2 = \mathrm{CRT}(a_1, b_2, p, q), P_3 = \mathrm{CRT}(a_2, b_1, p, q), P_4 = \mathrm{CRT}(a_2, b_2, p, q)$
3. $\{P_1, P_2, P_3, P_4\}$ 중 하나가 평문

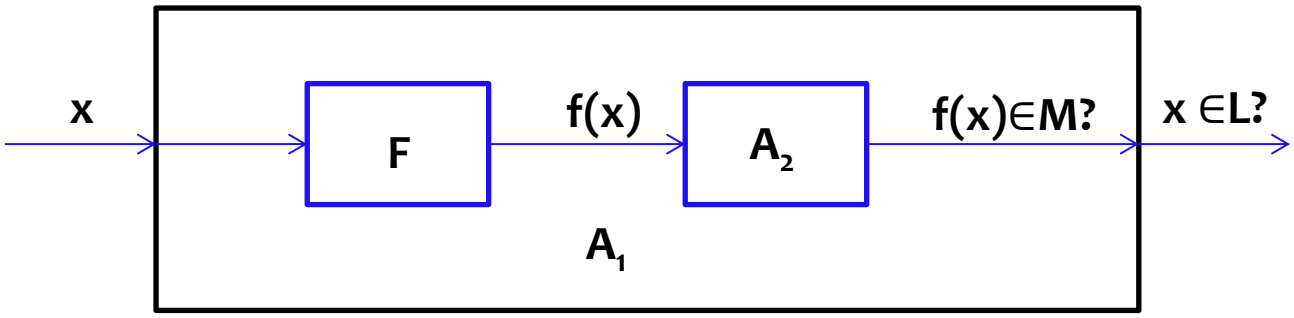
7.2 RABIN 암호 (예제)

1. 키 생성
   - $4k+3$의 형태인 $p=7$과 $q=11$
   - $n=p\times q=7\times 11=77$
   - $n=77,\ (p,q)=(7,11)$.
2. $m=10$ → $c\equiv 10^2\equiv 100\equiv 23\ (\mathrm{mod}\ 77)$
3. 복호화
   - $c_1\equiv 23\equiv 2\ (\mathrm{mod}\ 7),\ c_2\equiv 23\equiv 1\ (\mathrm{mod}\ 11)$.
   - $a_1\equiv 2^{\frac{(7+1)}{4}}\equiv 2^2\equiv 4\ (\mathrm{mod}\ 7)$,$a_2\equiv -2^{\frac{(7+1)}{4}}\equiv 2^2\equiv -4\equiv 3\ (\mathrm{mod}\ 7)$
   - $b_1\equiv 1^{\frac{(11+1)}{4}}\equiv 1^3\equiv 1\ (\mathrm{mod}\ 11)$, $b_2\equiv -1^{\frac{(11+1)}{4}}\equiv -1^3\equiv -1\ (\mathrm{mod}\ 11)$
   - CRT를 이용
     - $m_1\equiv 7\times 8\times 1+11\times 2\times 4\equiv 56+88=144\equiv 67\ (\mathrm{mod}\ 77)$
     - $m_2\equiv 7\times 8\times(-1)+11\times 2\times 4\equiv -56+88\equiv 32\ (\mathrm{mod}\ 77)$
     - $m_3\equiv 7\times 8\times(-1)+11\times 2\times(-4)\equiv -56-88\equiv -144\equiv -67\equiv 10\ (\mathrm{mod}\ 77)$
     - $m_4\equiv 7\times 8\times 1+11\times 2\times(-4)\equiv 56-88\equiv -32\equiv 45\ (\mathrm{mod}\ 77)$

7.2 RABIN 암호(안전성)

Polynomial Time Reducibility

- 만약 문제 Π1 이 (다항식 시간 안에) 문제 Π2 로 reduce된다면 , Π1은 Π2보다 어렵지는 않다는 것을 의미



- f(x) 는 문제 Π1 의 입력을 문제 Π2의 입력으로 변환하는 함수
- A_2 is an algorithm to solve Π2, i.e. to decide M
- A_1 is an algorithm to solve Π1, i.e. to decide L

7.2 RABIN 암호(안전성)

- RABIN 암호의 안전성
  - Fact 1: 정수 y와 x, 만약 $x^2 \equiv y^2 \pmod{n}$, $x \neq \pm y \pmod{n}$이면 $\gcd(x-y,n)$ 혹은 $\gcd(x+y,n)$ 는 n의 1이 아닌 인수
    - $x^2 \equiv y^2 \pmod{n}$
    - $x^2 - y^2 \equiv 0 \pmod{n}$
    - $(x+y)(x-y) \equiv 0 \pmod{n}$
    - $(x+y)(x-y) = k \times n = k \times p \times q$
    - $x \pm y \neq 0 \pmod{n}$ → $\gcd(x-y,n)$ 혹은 $\gcd(x+y,n)$은 인수
  - Fact 2: $n = pq$, $\gcd(y,n) = 1$, $x^2 \equiv y^2 \pmod{n}$, 4개의 해가 존재하며 그 중 2개는 $x = y \ (mod\ n)$과 $x = -y \ (mod\ n)$이다.
    - $n = 35, x^2 \equiv 4 \pmod{n}$ → $x = 2, 12, 23(= -12), 33\ (= -2)$

7.2 RABIN 암호

- 제곱근 문제 (Square root problem)

  Given $y \in \mathbb{Z}_n$, Find $x \in \mathbb{Z}_n$ s.t $x^2 \equiv y \pmod{n}$

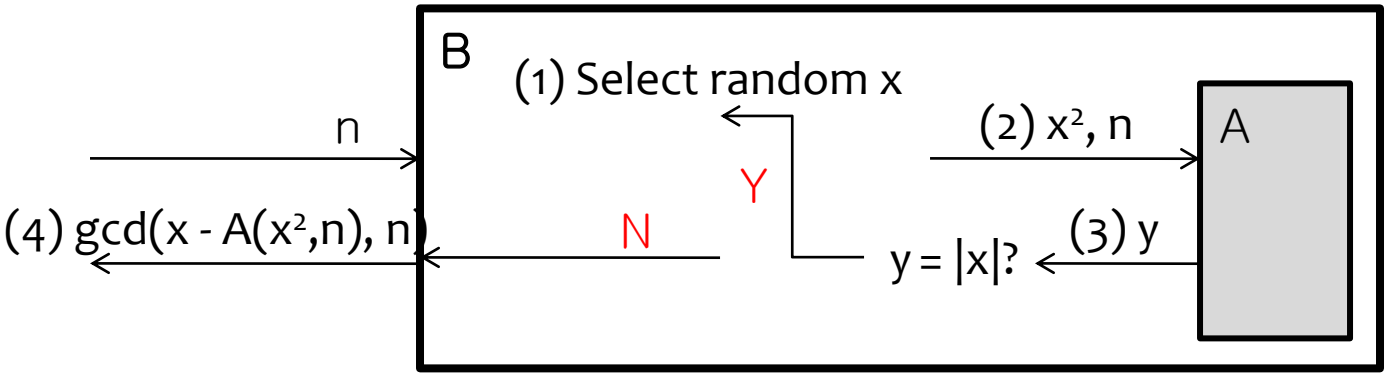
- **FACTOR** is poly-time reducible to **SQROOT (all-or-nothing security)**
  - A: SQROOT solver ; B: FACTOR solver using A

    (1) Select random x such that $\gcd(x, n) = 1$.

    (2) B calls A(x^2, n)

    (3) If A(x^2, n) = |x|, go to (1) (in average, two calls to A(x^2, n))

    (4) gcd(x - A(x^2,n), n) is a factor of n
  - Note that in average, two calls to A(x^2, n) are expected



7.3 ElGamal 암호

- 이산대수문제(Discrete Logarithm Problem, DLP) 의 어려움에 기반
  - 키생성
    1. 큰 소수 p를 선택
    2. $1 \leq d \leq p-2$ 의 범위에서 임의의 d를 선택
    3. $\mathbb{Z}_p^*$에서 원시근(Primitive Root) e_1을 선택
    4. d와 e_1을 이용해 $e_2 \equiv e_1^d \pmod{p}$ 을 계산
    5. 공개키는 (e_1, e_2, p), 개인키는 d

7.3 ElGamal 암호

- 이산대수문제(Discrete Logarithm Problem, DLP) 의 어려움에 기반
  - 암호화
    1. 임의의 값 r 을 선택
    2. $c_1 \equiv e_1^r \pmod{p}$
    3. $c_2 \equiv (m \times e_2^r) \pmod{p}$
    4. 암호문 (c_1, c_2)
  - 복호화
    1. $c_2 \times \left(c_1^d\right)^{-1} \bmod p$
       - $(c_2 \times \left(c_1^d\right)^{-1} \equiv (e_2^r \times m) \times \left(e_1^{rd}\right)^{-1} \equiv (e_1^{dr}) \times m \times \left(e_1^{rd}\right)^{-1} \equiv m \pmod{p}$

7.3 ElGamal 암호

- ElGamal 암호의 예제
  - 키생성
    1. 소수 $p = 13$
    2. $1 \le d \le 11$의 범위에서 $d = 3$
    3. $\mathbb{Z}_{13}^*$에서 원시근 $e_1 = 2$
    4. $d = 3$과 $e_1 = 2$를 이용해 $e_2 \equiv 2^3 \equiv 8 \pmod{13}$
    5. 공개키 $(e_1, e_2, p) = (2, 8, 13)$, 개인키 $d = 3$
  - 암호화
    1. 평문 $m = 11$을 선택하고, $r = 5$
    2. $c_1 \equiv 2^5 \equiv 32 \equiv 6 \pmod{13}$
    3. $c_2 \equiv 11 \times 8^5 \equiv 360{,}448 \equiv 10 \pmod{13}$
    4. 암호문 $(c_1, c_2) = (6, 10)$
  - 복호화
    1. $c_2 \times \left(c_1^d\right)^{-1} \equiv 10 \times (6^3)^{-1} \equiv 10 \times 6^9 \equiv 10 \times 5 \equiv 11 \pmod{13}$

7.3 ElGamal 암호(취약점)

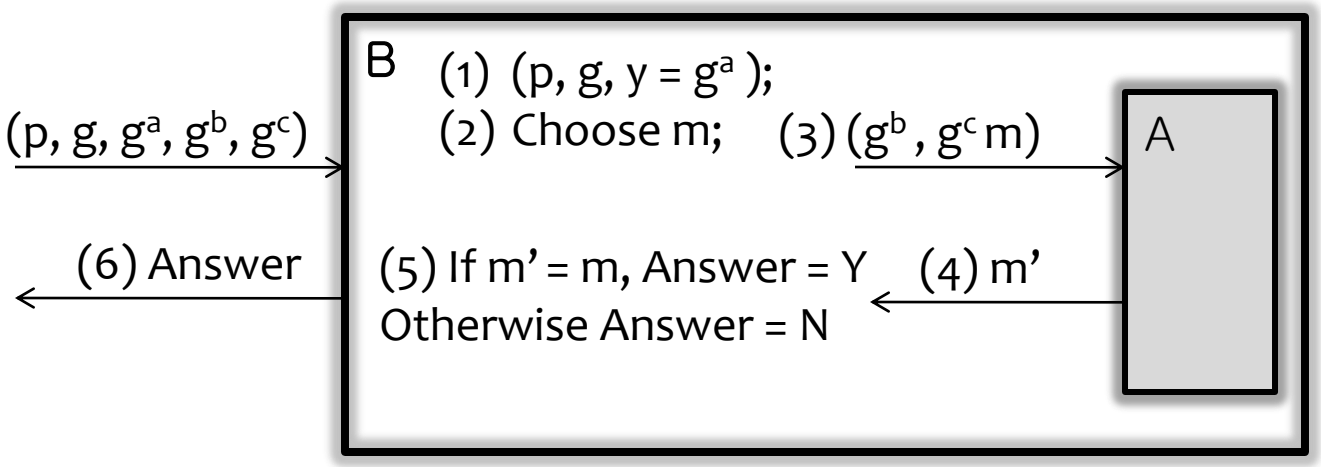
- 작은 모듈러스 공격(Low-Modulus Attack)
  - p의 값이 충분히 크지 않을 경우 전수 조사나 이산대수의 성질을 이용한 효율적인 알고리즘을 통하여 개인키 d나 임의의 값 r을 찾아낼 수 있다.
  - p는 적어도 2048 비트
- 알려진 평문 공격(Known-Plaintext Attack)
  - 평문 m에 대응하는 암호문 (c_1, c_2)을 알고 있고, 같은 r을 사용한 암호문 $c_1^* \equiv e_1^r \pmod p$, $c_2^* \equiv (m^* \times e_2^r) \pmod p$을 얻었을 때
    1. $e_2^r \equiv c_2 \times m^{-1} \pmod p$
    2. $c_2^* \times (e_2^r)^{-1} \bmod p$ ➔ m^*

7.3 ElGamal 암호(안전성)

- **The Computational Diffie-Hellman (CDH) 문제**
  - 순환군 : G of order q, 생성원 $g \in G$에 대하여, $a, b \in \{0,1, \ldots q-1\}$, (g, g^a, g^b, q), 가 주어졌을 때, $g^{ab} \bmod q \equiv ?$
- **The Decisional Diffie-Hellman (DDH)문제**
  - 순환군 : G of order q, 생성원 $g \in G$에 대하여, $a, b, c \in \{0,1, \ldots q-1\}$, (g, g^a, g^b, g^c, q), 가 주어졌을 때, $ab \equiv ? c \pmod q$
- CDH $\propto$ DL ; but DL $\propto$ CDH ?
- DDH $\propto$ CDH; CDH $\propto$ DDH?

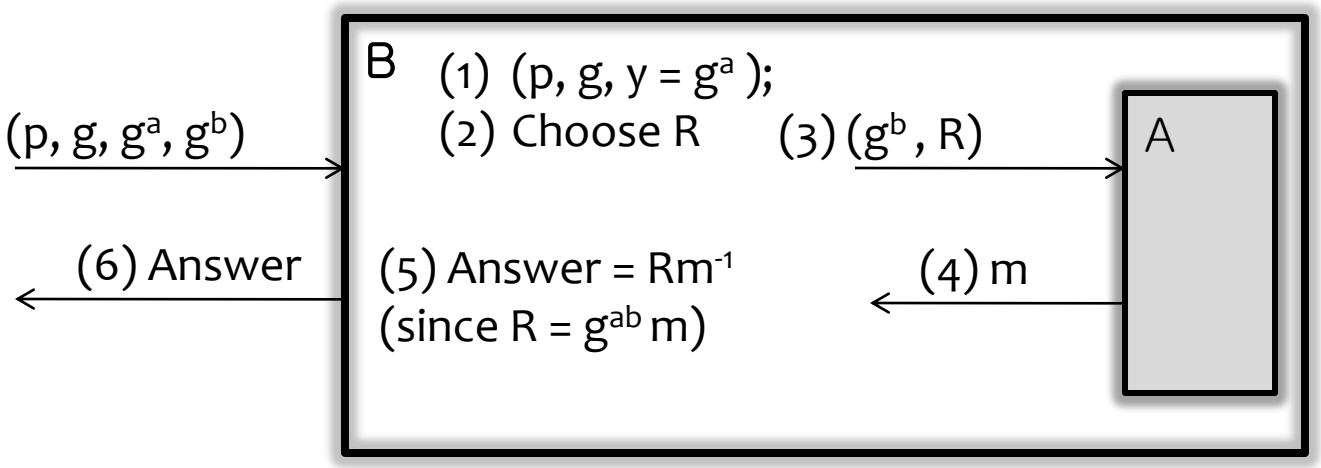
7.3 ElGamal 암호(안전성)

- **ElGamal** is secure if DDH is hard. **(all-or-nothing security)**
  - A : ElGamal Oracle B : DDH Attacker



7.3 ElGamal 암호(안전성)

- **ElGamal** is secure if CDH is hard. **(all-or-nothing security)**
  - A : ElGamal Oracle B : CDH Attacker



7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- RSA와 ElGamal은 안전한 공개키 암호시스템이지만 키의 크기가 커야 한다. 타원곡선 암호시스템(ECC)는 동일한 안전성을 유지하면서 작은 크기의 키를 사용한다.

- 타원곡선의 일반식

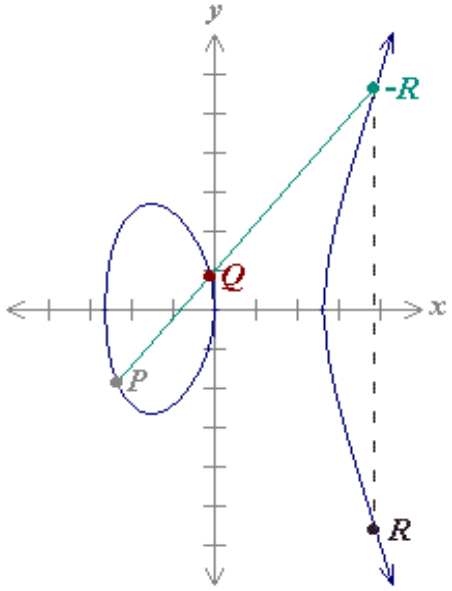
$$y^2 + b_1xy + b_2y = x^3 + a_1x^2 + a_2x + a_3$$

- 실수 상에서의 타원곡선

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

7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- Addition of point P and point Q



$y^2 = x^3 - 7x$

P (-2.35, -1.86)
Q (-0.1, 0.836)
-R (3.89, 5.62)
R (3.89, -5.62)

P + *Q* = *R* = (3.89, -5.62).

$$\lambda = (y_2 - y_1)/(x_2 - x_1)$$

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

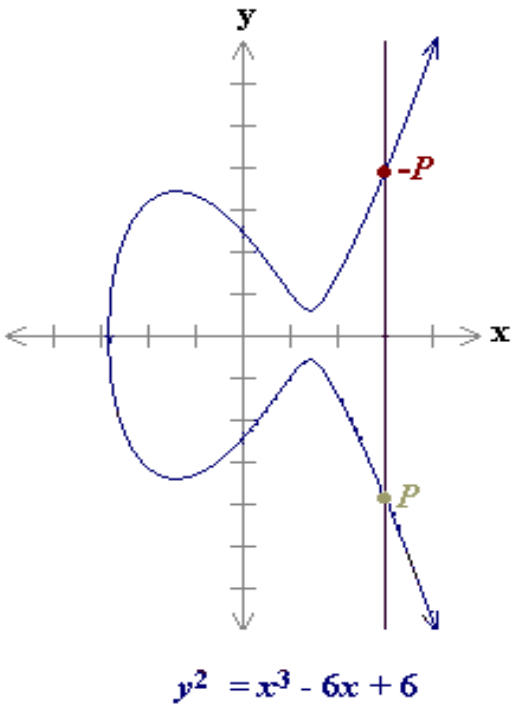
When Q = P

$$\lambda = (3x_1^2 + a)/(2y_1)$$

$$x_3 = \lambda^2 - x_1 - x_2 \qquad y_3 = \lambda\,(x_1 - x_3) - y_1$$

7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- Addition of P and -P



$P + (-P) = O$

By definition, $P + (-P) = \boldsymbol{O}$.

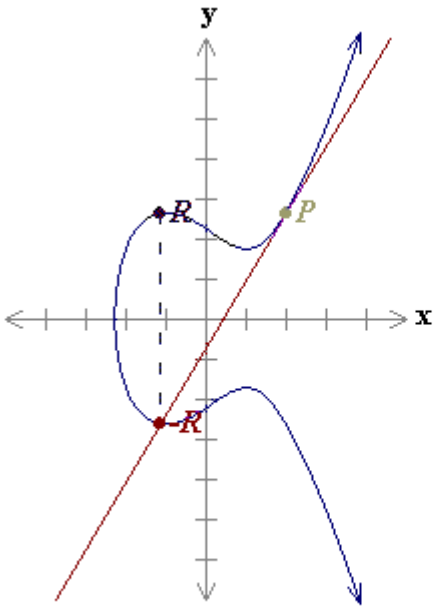
As a result of this equation,
$P + \boldsymbol{O} = P$ in the elliptic curve group .

$\boldsymbol{O}$ = the additive identity of the elliptic curve group;

All elliptic curves have an additive identity.

7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- Doubling the point P



$y^2 = x^3 - 3x + 5$

P (2, 2.65)

$-R$ (-1.11, -2.64)

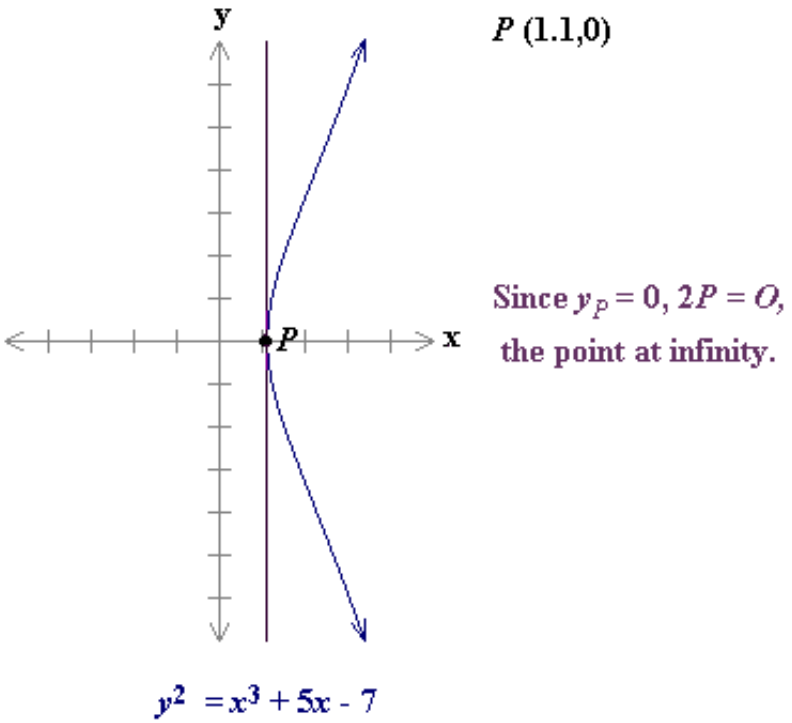
R (-1.11, 2.64)

$2P = R = (-1.11, 2.64)$.

$P = (x, y)$
If $y \neq 0$, then the tangent line at P *meets exactly one point of the curve.*
the law for doubling a point on an elliptic curve group **:**
$P + P = 2P = R$

7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- Doubling the point P when y = 0



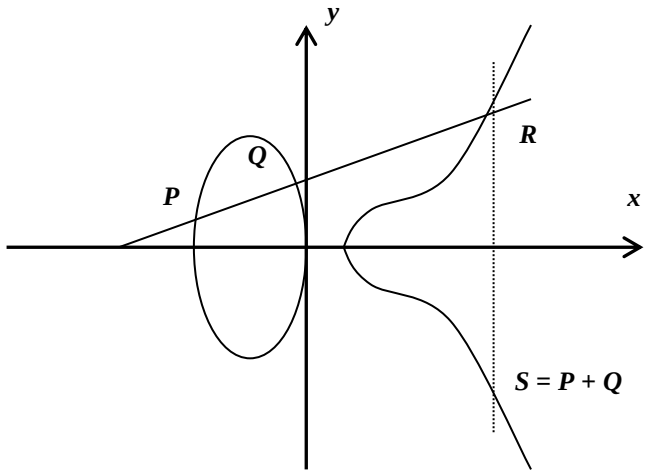
By definition, $2P = O$ for such a point P.

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

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

7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- 타원곡선 군
  1. 닫힘: (in "+" operation)
  2. 결합법칙 : P + (Q + R) = (P + Q) + R
  3. 교환법칙 : P + Q = Q + P
  4. 항등원 : P + **O** = P
  5. 역원 : P + Q = **O** : 점 (x, y)의 역원은 (x, –y)이다. (–y는 y의 덧셈상의 역원이다. 예로 p = 13이면 (4, 2)의 역원은 (4, -2) = (4, 11).



7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

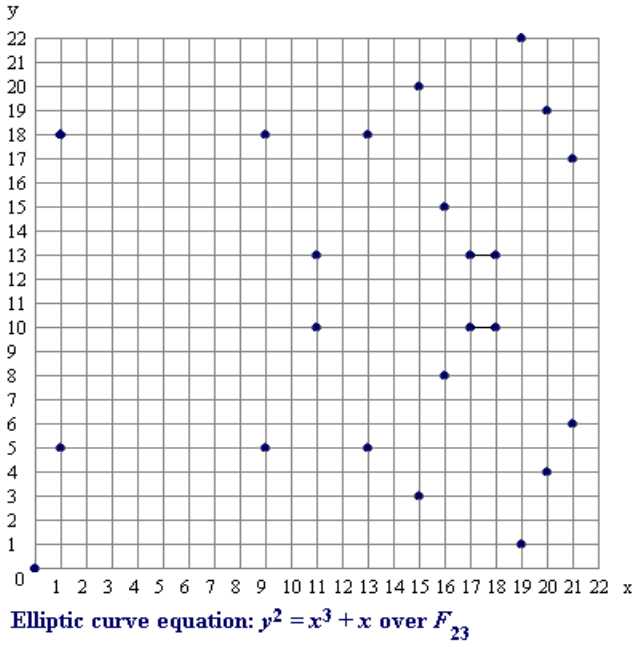
- $\mathbb{Z}_{13}$ 위에서 $y^2 = x^3 + x$ 에 정의되는 타원 곡선 군

For $a = 1$ and $b = 0$, (9,5) is on $y^2 = x^3 + x$.
because :

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

위 식을 만족하는 23 점들 :

(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}

7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- GF(p) : finite field with p elements
- Order of a point P on EC : the least integer x satisfying xP = O
- Elliptic Curve DLP
  - For two point Q and P (with order t) on EC defined in GF(p), find integer x s.t. Q = xP
- *Example : $y^2 \bmod 23 = x^3 + 9x + 17 \bmod 23$,*
  - What is the discrete logarithm *x* of *Q* = (4,5) to the base *P* = (16, 5)? *Q* = *xP*

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

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

7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- We may compute P, 2P = P + P, 3P = P + P + P, … to find x.
- Repeated doubling and multiplication
  - 2P = P + P
  - 100P = 2(2(P + 2(2(2(P + 2P)))))

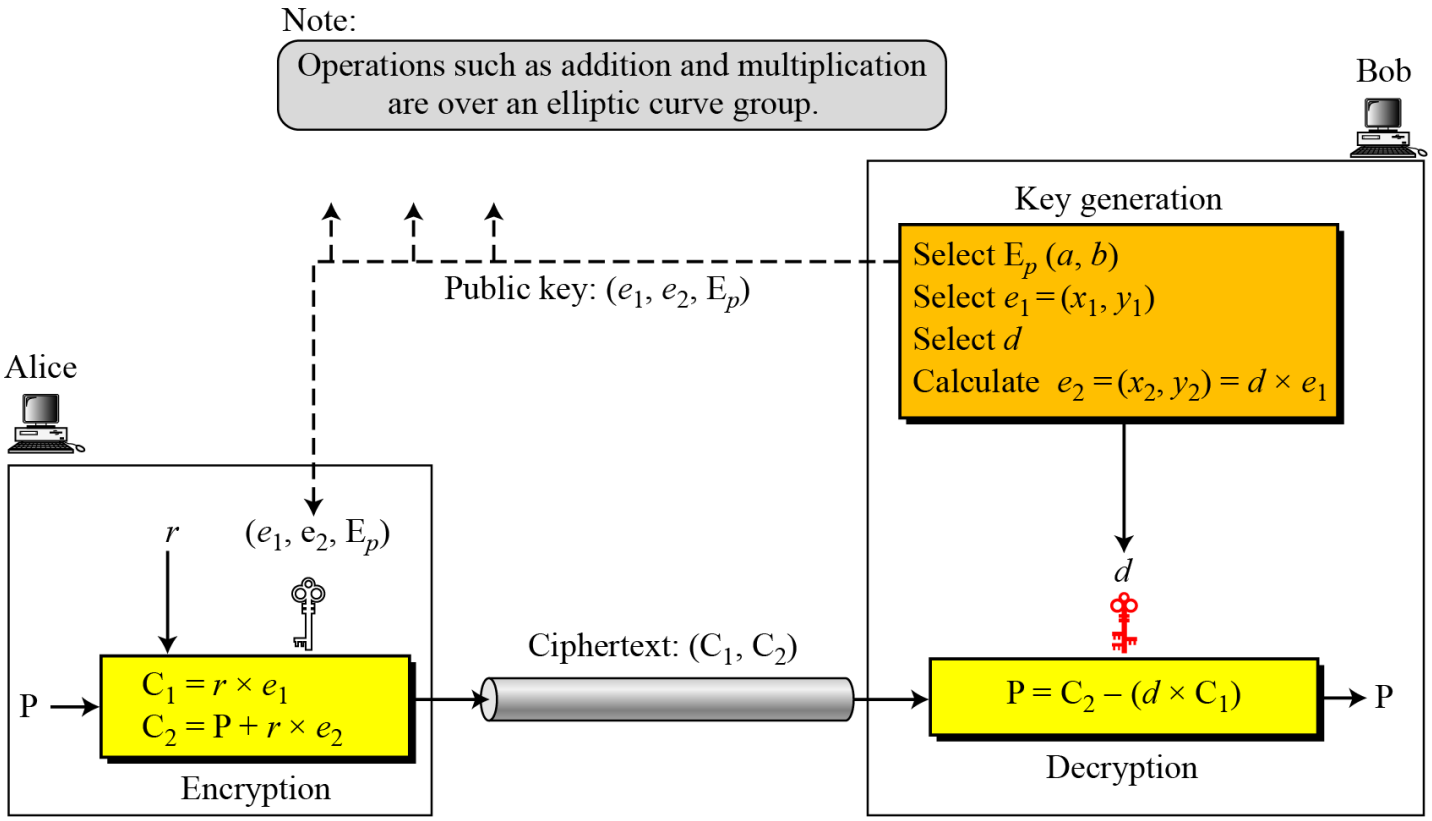
7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- ECC simulating ElGamal

	ElGamal	ECC ElGamal
Public Key	$\{e_1, p, e_2 = e_1^d \bmod p\}$	$\{e_1, p, e_2 = de_1\}$
Private Key	$\{d\}$	$\{d\}$
Enc .	$[c_1, c_2] = [e_1^r \bmod p, e_2^r m \bmod p]$	$[C_1, C_2] = [re_1, re_2 + M]$
Dec.	$c_2 c_1^{-d} \bmod p$	$C_2 - dC_1$

7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- ElGamal cryptosystem using the elliptic curve



7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- Example : For p = 11, a =1, b = 6, points on EC $y^2 = x^3 + x + 6$ defined in GF(p) are as follows:

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

When private key d = 7 , public key e_1 = (2, 7), and e_2 = 7(2, 7) = (7, 2) are given, Alice selects r = 3 randomly, and encrypts M = (10, 9) into C1 = 3(2, 7) = (8, 3), C2 = 3(7, 2) + (10, 9) = (10, 2).

7.4 타원 곡선상의 ElGamal(Elliptic Curve ElGamal)

- Key length
  - RSA : 2048 bit = ECC : 224 bit
- Enc./Dec. Efficiency
  - Computation Speed on a low power device such as Cellular Phone
    - RSA: 1 sec (2048 bit RSA exponentiation)
    - ECC: 0.09 sec (224 bit scalar multiplication)

	ECC-160	RSA-1024	ECC-192	RSA-1536	ECC-224	RSA-2048
Time (ms)	3.69	8.75	3.87	27.47	5.12	56.18
Ops/sec	271.3	114.3	258.1	36.4	195.5	17.8
Performance ratio	2.4 : 1		7.1 : 1		11 : 1	
Key-size ratio	1 : 6.4		1 : 8		1 : 9.1	