

암호학과 네트워크 보안

- 9장 암호수학(2) -

김민식 (ms6127@naver.com)

세종대학교 프로토콜공학연구실

목 차

- 보충
- 중국인의 나머지 정리
- 2차 합동
- 지수와 로그

보충

- AES 보안성 (1/2)

1. 큰 키 공간(128, 192, 256 비트)

- 현재 컴퓨터 성능 기준으로, 128/192/256비트 키 모두
전수조사 공격이 어려움

2. 키 확장 과정의 혼돈과 확산

- SubWord의 S-Box 연산으로 라운드 키 생성 과정에 혼돈을 부여함
- RotWord는 바이트 위치를 순환 이동시켜 키 바이트가 서로 다른 위치로 전달되도록 하여 확산을 제공함
- Rcon은 라운드마다 다른 고정 상수를 적용하여 라운드 키 간의 반복 패턴과 대칭성을 방지함
- 이를 통해 라운드 키 간의 관계 분석을 어렵게 만들어 보안성을 높임

- AES 보안성 (2/2)

- 3. 라운드 함수의 혼돈과 확산

- SubBytes는 평문, 키, 암호문 사이의 관계를 비선형적으로 만들어 혼돈을 제공함
 - ShiftRows와 MixColumns를 통해 한 바이트의 변화가 여러 바이트로 퍼지도록 하여 확산을 제공함
 - 이를 통해 입력의 작은 변화가 암호문 전체에 영향을 미치게 하여 평문과 암호문 간의 관계 분석을 어렵게 만듦

목 차

- 보충
- 중국인의 나머지 정리
- 2차 합동
- 지수와 로그

중국인의 나머지 정리

- 정리

- 모듈로 값들이 서로소만 된다면 연립 합동 방정식이 유일한 해를 갖는다는 것을 보인 정리
 - 연립 합동 방정식 ($0 \leq x < M$)

$$\begin{aligned}x &\equiv a_1 \pmod{m_1} \\x &\equiv a_2 \pmod{m_2} \\&\vdots \\x &\equiv a_k \pmod{m_k}\end{aligned}$$

- 적용 사례

- RSA에서는 $n = p \times q$ 구조를 이용하여 큰 모듈러 n 에서 바로 계산하지 않고 $\pmod{p}$ 와 $\pmod{q}$ 로 나누어 복호화 연산을 수행하고 각각의 결과를 중국인의 나머지 정리로 결합하여 원래 $\pmod{n}$ 에서의 결과를 복원함

중국인의 나머지 정리

• 방법

1. 공통 모듈로로 사용할 $M = m_1 \times m_2 \times \cdots \times m_k$ 을 구함
2. $M_1 = M/m_1, M_2 = M/m_2, \cdots, M_k = M/m_k$ 를 구함
3. 각 M_i 의 m_i 에 대한 곱셈 역원 M_i^{-1} 을 구함
4. 각 값을 공식에 대입하여 최종 해 x 를 구함
$$x \equiv (a_1 \times M_1 \times M_1^{-1}) + (a_2 \times M_2 \times M_2^{-1}) + (a_k \times M_k \times M_k^{-1}) \pmod{M}$$

*연립 합동 방정식:

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

중국인의 나머지 정리

• 예제 9.1

다음 합동방정식의 해를 구하라

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 2 \pmod{7}$$

• 풀이

1. $M = m_1 \times m_2 \times m_3 = 3 \times 5 \times 7 = 105$

2. $M_1 = M/m_1 = 105/3 = 35$

$$M_2 = M/m_2 = 105/5 = 21$$

$$M_3 = M/m_3 = 105/7 = 15$$

3. 곱에 대한 역원은 $35 \times M_1^{-1} \equiv 1 \pmod{3} \rightarrow M_1^{-1} = 2$

$$21 \times M_2^{-1} \equiv 1 \pmod{5} \rightarrow M_2^{-1} = 1$$

$$15 \times M_3^{-1} \equiv 1 \pmod{7} \rightarrow M_3^{-1} = 1$$

4. $x \equiv (2 \times 35 \times 2) + (3 \times 21 \times 1) + (2 \times 15 \times 1) \pmod{105} \equiv 23 \pmod{105}$

중국인의 나머지 정리

• 예제 9.2 (1/2)

사용하는 시스템의 능력이 100보다 작은 수만 입력받을 수 있다고 가정하고 $x = 123$, $y = 334$ 일 때, $z = x + y$ 를 중국인의 나머지 정리를 이용하여 구하라

• 풀이

- $x \equiv 24 \pmod{99}$ $y \equiv 37 \pmod{99}$
 $x \equiv 25 \pmod{98}$ $y \equiv 40 \pmod{98}$
 $x \equiv 26 \pmod{97}$ $y \equiv 43 \pmod{97}$
- $x + y \equiv 61 \pmod{99} \rightarrow z \equiv 61 \pmod{99}$
 $x + y \equiv 65 \pmod{98} \rightarrow z \equiv 65 \pmod{98}$
 $x + y \equiv 69 \pmod{97} \rightarrow z \equiv 69 \pmod{97}$

중국인의 나머지 정리

• 예제 9.2 (2/2)

사용하는 시스템의 능력이 100보다 작은 수만 입력받을 수 있다고 가정하고 $x = 123$, $y = 334$ 일 때, $z = x + y$ 를 중국인의 나머지 정리를 이용하여 구하라

• 풀이

1. $M = 99 \times 98 \times 97 = 941094$

2. $M_1 = 941094/99 = 9506$

$$M_2 = 941094/98 = 9603$$

$$M_3 = 941094/97 = 9702$$

3. $9506 \times M_1^{-1} \equiv 1 \pmod{99} \rightarrow M_1^{-1} = 50$

$$9603 \times M_2^{-1} \equiv 1 \pmod{98} \rightarrow M_2^{-1} = 97$$

$$9702 \times M_3^{-1} \equiv 1 \pmod{97} \rightarrow M_3^{-1} = 49$$

4. $z \equiv (61 \times 9506 \times 50) + (65 \times 9603 \times 97) + (69 \times 9702 \times 49) \pmod{941094} \equiv 122342677 \pmod{941094} \equiv 457 \pmod{941094}$
 $\therefore z = 457$

목 차

- 보충
- 중국인의 나머지 정리
- 2차 합동
- 지수와 로그

2차 합동

- 정의

- $a_2x^2 + a_1x + a_0 \equiv 0 \pmod{n}$ 의 형태를 갖는 합동 방정식
 - 암호수학에서는 주로 $x^2 \equiv a \pmod{n}$ 형태를 가짐

- 적용 사례

- Rabin 공개키 암호의 복호화 과정에서는 암호문 c 가 주어졌을 때 $x^2 \equiv c \pmod{n}$ 을 만족하는 모듈러 제곱근 x 를 찾아야 하므로, 2차 합동 개념이 직접적으로 사용됨

2차 합동

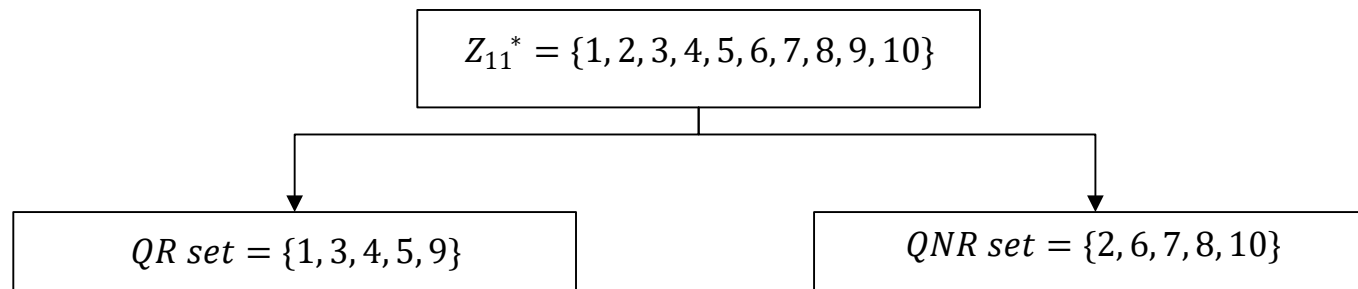
- 모듈로가 소수인 2차 합동 (1/8)
 - 2차 잉여(QR, Quadratic Residue)
 - $x^2 \equiv a \pmod{p}$ 에서 해 x 가 존재하는 a
 - e.g., $x^2 \equiv 3 \pmod{13} \rightarrow x = 4$ 에서, $16 \equiv 3 \pmod{13}$, $x = 9$ 에서 $81 \equiv 3 \pmod{13}$ 3은 $\pmod{13}$ 에서 2차 잉여임
 - 2차 비잉여(QNR, Quadratic Nonresidue)
 - $x^2 \equiv a \pmod{p}$ 에서 해 x 가 존재하지 않는 a
 - e.g., $x^2 \equiv 5 \pmod{13} \rightarrow$ 해가 존재하지 않음
5는 $\pmod{13}$ 에서 2차 비잉여임
- 암호 기법에서 주로 사용하는 $\mathbb{Z}_p^*$ 에서는 전체 $(p - 1)$ 개의 원소 중 절반은 2차 잉여이고, 나머지는 2차 비잉여임

2차 합동

- 모듈로가 소수인 2차 합동 (2/8)
- 예제 9.3

Z_{11}^* 에는 10개의 원소가 있다. 2차 잉여와 2차 비잉여를 구분해서 나타내시오.

- 풀이
 - $1^2 \equiv 1 \pmod{11}$, $2^2 \equiv 4 \pmod{11}$, $3^2 \equiv 9 \pmod{11}$,
 $4^2 \equiv 5 \pmod{11}$, $5^2 \equiv 3 \pmod{11}$
제공 결과로 나오는 값은 1, 3, 4, 5, 9이므로 2차 잉여임
 - 나머지 원소 2, 6, 7, 8, 10들은 어떤 값을 제공해도 나오지 않으므로 2차 비잉여임



2차 합동

- 모듈로가 소수인 2차 합동 (3/8)
- 오일러 판정 기준(Euler's Criterion)
 - a. 만약 $a^{(p-1)/2} \equiv 1 \pmod{p}$ 이면 모듈로 p 로 a 는 2차 잉여임
 - e.g., $p = 13, a = 3$ 일 때 $3^{(13-1)/2} \equiv 3^6 \pmod{13} \equiv 729 \pmod{13} \equiv 1 \pmod{13}$, 3은 $\pmod{13}$ 에서 2차 잉여
 - b. 만약 $a^{(p-1)/2} \equiv -1 \pmod{p}$ 이면 모듈로 p 로 a 는 2차 비잉여임
 - e.g., $p = 13, a = 5$ 일 때 $5^{(13-1)/2} \equiv 5^6 \pmod{13} \equiv 15625 \pmod{13} \equiv 12 \pmod{13} \equiv -1 \pmod{13}$, 5는 $\pmod{13}$ 에서 2차 비잉여

2차 합동

- 모듈로가 소수인 2차 합동 (4/8)

- 방정식 풀이

1. 오일러 판정 기준으로 해의 존재 여부를 먼저 확인함

- $a^{(p-1)/2} \equiv 1 \pmod p \rightarrow$ 해 존재
 $a^{(p-1)/2} \equiv -1 \pmod p \rightarrow$ 해 없음

2. 해가 존재하고 $p = 4k + 3$ 형태이면 $x \equiv \pm a^{(p+1)/4} \pmod p$ 로 해를 구함

- $p = 4k + 3$ 형태가 아니면 $(p+1)/4$ 이 정수가 아니므로 공식을 사용할 수 없어 $p-1 = Q \cdot 2^s$ 구조를 활용해 제곱근 후보를 보정하는 Tonelli-Shanks 알고리즘으로 해를 구함

2차 합동

- 모듈로가 소수인 2차 합동 (5/8)

- 예제 9.4

다음 2차 합동 방정식의 해를 구하시오.

- a. $x^2 \equiv 3 \pmod{23}$
- b. $x^2 \equiv 2 \pmod{11}$
- c. $x^2 \equiv 7 \pmod{19}$

- 풀이

- a. 3은 Z_{23}^* 에서 $3^{11} \equiv 1 \pmod{23}$ 이므로 QR이고 23은 $4k + 3$ 형태이므로 해는 $x \equiv \pm 3^6 \pmod{23} \equiv \pm 16 \pmod{23}$ 임
- b. 2는 Z_{11}^* 에서 $2^5 \equiv -1 \pmod{11}$ 이므로 QNR 이고 해가 존재하지 않음
- c. 7은 Z_{19}^* 에서 $7^9 \equiv 1 \pmod{19}$ 이므로 QR이고 19는 $4k + 3$ 형태이므로 해는 $x \equiv \pm 7^5 \pmod{19} \equiv \pm 11 \pmod{19}$ 임

2차 합동

- 모듈로가 소수인 2차 합동 (6/8)

- Tonelli-Shanks 알고리즘

- 정의

- 2차 비잉여 z 로부터 보정값을 생성하고, 제곱근 후보를 반복적으로 수정하면서 $x^2 \equiv a \pmod{p}$ 의 해를 구하는 알고리즘

- 구조

$$p - 1 = Q \cdot 2^S$$

- $p - 1$ 을 홀수 부분 Q 와 2의 거듭제곱 부분 2^S 로 분해하는 구조를 이용하여 제곱근 후보를 반복적으로 보정함

2차 합동

• 모듈로가 소수인 2차 합동 (7/8)

• Tonelli-Shanks 알고리즘

• 동작과정

1. $p - 1 = Q \cdot 2^S$ 형태로 나눔
2. 2차 비잉여 $z^{(p-1)/2} \equiv -1 \pmod p$ 을 만족하는 z 값을 고름
3. 변수를 $M = S, c = z^Q \pmod p, t = a^Q, R = a^{(Q+1)/2} \pmod p$ 로 초기화함
4. $t \equiv 1 \pmod p$ 이면 $R^2 \equiv a \pmod p$ 가 성립함
5. 만약 $t \neq 1$ 이면 $t^{2^i} \equiv 1 \pmod p$ 을 만족하는 가장 작은 i 를 찾고 보정 값 $b = c^{2^{M-i-1}} \pmod p$ 을 계산하고 변수 t 가 1이 될 때까지 $M = i, c = b^2, t = tb^2, R = Rb$ 를 반복함
6. $t \equiv 1 \pmod p$ 이 되면 알고리즘이 종료되고 이때 R 이 하나의 제곱근이고, 다른 하나는 $p - R$ 임
7. 따라서 최종 해는 $x \equiv R, p - R \pmod p$ 임

z : 제곱근 후보를 보정하기 위해 선택하는 2차 비잉여 보조 값

Q : $Q \cdot 2^S$ 형태의 홀수 부분

M : 현재 보정 단계 수

$c = z^Q \pmod p$: 보정에 사용하는 값

R : 현재 제곱근 후보

t : R 이 제곱근인지 확인하는 보조값

2차 합동

- 모듈로가 소수인 2차 합동 (8/8)

- Tonelli-Shanks 알고리즘

- 예제 9.5

Tonelli-Shanks 알고리즘을 이용하여 다음 2차 합동방정식의 해를 구하라.

$$x^2 \equiv 5 \pmod{41}$$

- 풀이

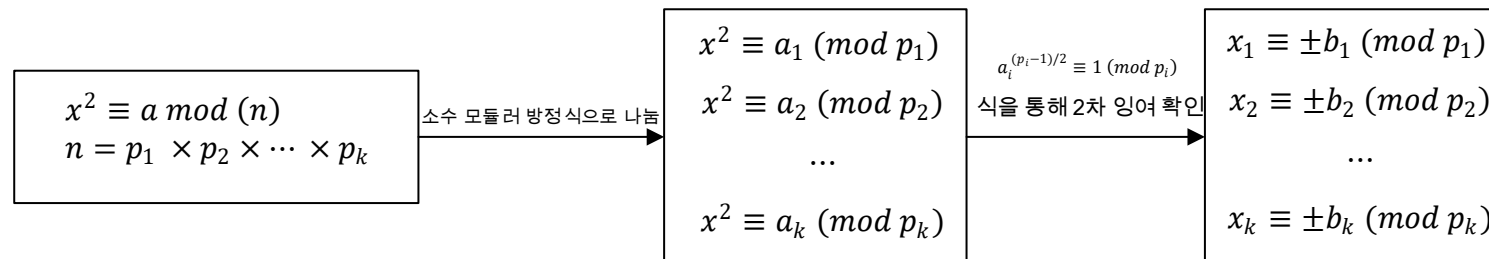
1. $p - 1 = 40 = 5 \cdot 2^3$ ($Q = 5, S = 3$)
2. $z = 3$ 선택 $3^{20} \equiv -1 \pmod{41}$ 이므로 2차 비잉여
3. $M = 3, c = 3^5 \pmod{41} = 38, t = 5^5 \pmod{41} = 9, R = 5^3 \pmod{41} = 2$ 로 초기값 설정
4. $t^{(2^2)} \equiv 1 \pmod{41}$ 이 되는 가장 작은 $i = 2$ 를 찾고 $b = c^{2^{3-2-1}} = c^1 = 38$ 이후 갱신하면 $M = 2, c = b^2 \pmod{41} = 9, t = tb^2 \pmod{41} = 40, R = Rb \pmod{41} = 35$
5. $t \neq 1$ 이므로 1이 될 때까지 다시 반복함
 $t^{2^1} \equiv 1 \pmod{41}$ 이므로 $i = 1, b = c^{2^{2-1-1}} = c^1 = 9$ 갱신하면
 $M = 1, c = b^2 \pmod{41} = 40, t = tb^2 \pmod{41} = 1, R = Rb \pmod{41} = 28$
6. $t = 1$ 이므로 종료 최종 해는 $x \equiv 28, 13 \pmod{41}$

2차 합동

• 모듈로가 합성수인 2차 합동 (1/2)

• 방정식 풀이

1. 합성수 n 을 소인수분해함: $n = p_1 \times p_2 \times \cdots \times p_k$
2. 원래 방정식을 각 소수 모듈러에 대한 방정식으로 나눔:
 $x^2 \equiv a_1 \pmod{p_1} \quad x^2 \equiv a_2 \pmod{p_2} \quad \cdots \quad x^2 \equiv a_k \pmod{p_k}$
3. 각 소수 모듈러에서 a_i 가 2차 잉여인지 확인함:
 $a_i^{(p_i-1)/2} \equiv 1 \pmod{p_i}$
4. 각 소수 모듈러 p_i 에서 제곱근 x_i 를 구하고
 $x_i \equiv \pm b_i \pmod{p_i}$ 형태로 정리하여 연립 합동식 후보를 만듦
5. 각 연립 합동식을 중국인의 나머지 정리로 결합하여 최종 $\pmod{n}$ 의 해를 구함



2차 합동

- 모듈로가 합성수인 2차 합동 (2/2)

- 예제 9.5

77 = 7 × 11임을 이용하여 $x^2 \equiv 36 \pmod{77}$ 의 해를 구하라

- 풀이

- $x^2 \equiv 36 \pmod{7} \equiv 1 \pmod{7}$ 과 $x^2 \equiv 36 \pmod{11} \equiv 3 \pmod{11}$ 로 나눔
- 오일러 판정 기준으로 $1^3 \equiv 1 \pmod{7}$ 이고 $3^5 \equiv 1 \pmod{11}$ 이므로 해가 존재함
- 7과 11은 $4k + 3$ 형태이고 $x \equiv \pm a^{(p+1)/4} \pmod{p}$ 에 적용하면
$$x \equiv \pm 1^{(7+1)/4} \pmod{7} \equiv \pm 1^2 \pmod{7}$$
$$x \equiv \pm 3^{(11+1)/4} \pmod{11} \equiv \pm 3^3 \pmod{11} \equiv \pm 5 \pmod{11}$$
- Set 1: $x \equiv 1 \pmod{7}$ Set 2: $x \equiv 1 \pmod{7}$
 $x \equiv 5 \pmod{11}$ $x \equiv 6 \pmod{11}$
Set 3: $x \equiv 6 \pmod{7}$ Set 4: $x \equiv 6 \pmod{7}$
 $x \equiv 5 \pmod{11}$ $x \equiv 6 \pmod{11}$
- 중국인의 나머지 정리를 통해서 구하면
Set 1: $x \equiv 71 \pmod{77} \equiv -6 \pmod{77}$, Set 2: $x \equiv 50 \pmod{77} \equiv -27 \pmod{77}$
Set 3: $x \equiv 27 \pmod{77}$, Set 4: $x \equiv 6 \pmod{77}$
- 각각의 연립방정식의 해는 $x = \pm 6 \pmod{77}, \pm 27 \pmod{77}$ 임

목 차

- 보충
- 중국인의 나머지 정리
- 2차 합동
- **지수와 로그**

지수와 로그

- 지수(Exponentiation)

- 정의

$$y = a^x$$

- x 를 지수라고 부르고 a 를 밑수라고 하고 밑수를 x 번 곱해야 되는 것을 의미함

- 적용 사례

- RSA 암호의 암호화·복호화 과정에서 $c^d \bmod n$, $m^e \bmod n$ 형태의 모듈러 지수계산이 사용됨
- Diffie-Hellman 키 교환에서는 각 사용자가 $g^a \bmod p$, $g^b \bmod p$ 형태의 값을 계산하여 공개하고, 이를 이용해 공통 비밀키를 생성함

지수와 로그

- 고속 지수 계산 (1/4)
 - 제곱-곱 방법(Square-and-Multiply Method)을 사용함
 - 제곱과 곱셈을 모두 사용함
 - 지수를 n_b 비트(x_0 에서 x_{n_b-1})를 갖는 2진수로 간주함

$$x = x_{n_b-1} \times 2^{n_b-1} + x_{n_b-2} \times 2^{n_b-2} + \dots + x_2 \times 2^2 + x_1 \times 2^1 + x_0 \times 2^0$$

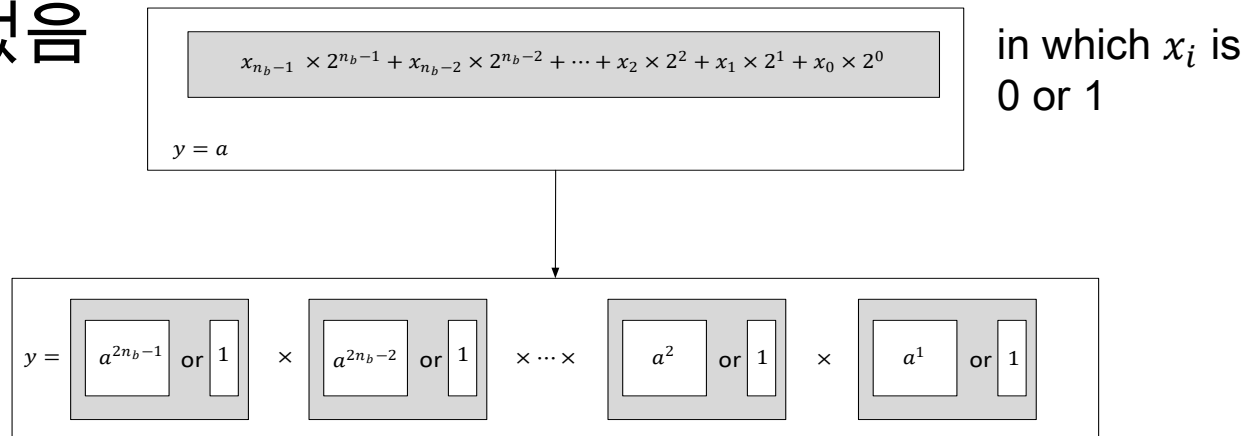
- e.g., $x = 22 = (10110)_2$ 일 때 $22 = 2^4 \times 1 + 2^3 \times 0 + 2^2 \times 1 + 2^1 \times 1 + 2^0 \times 0$ 로 표현 가능함

지수와 로그

• 고속 지수 계산 (2/4)

• 계산 과정

1. 지수 x 의 비트 x_i 를 확인하고 해당 비트가 1이면 현재 결과 값 y 에 a 를 곱함
2. 다음 비트 계산에 사용하기 위해 a 를 제곱함
3. 1~2 과정을 지수 x 의 비트 수인 n_b 번 반복함
4. 마지막 반복 이후에는 더 이상 다음 비트가 없으므로 제공 과정이 필요 없음



Example:

$$y = a^9 = a^{1001_2} = a^8 \times 1 \times 1 \times a$$

지수와 로그

- 고속 지수 계산 (3/4)
- 알고리즘

```
Square_and_Multiply ( $a, x, n$ )
{
   $y \leftarrow 1$ 
  for ( $i \leftarrow 0$  to  $n_b - 1$ )           //  $n_b$ 는  $x$ 의 비트 수이다.
  {
    if( $x_i = 1$ )   $y \leftarrow a \times y \bmod n$       //비트가 1인 경우만 곱한다.

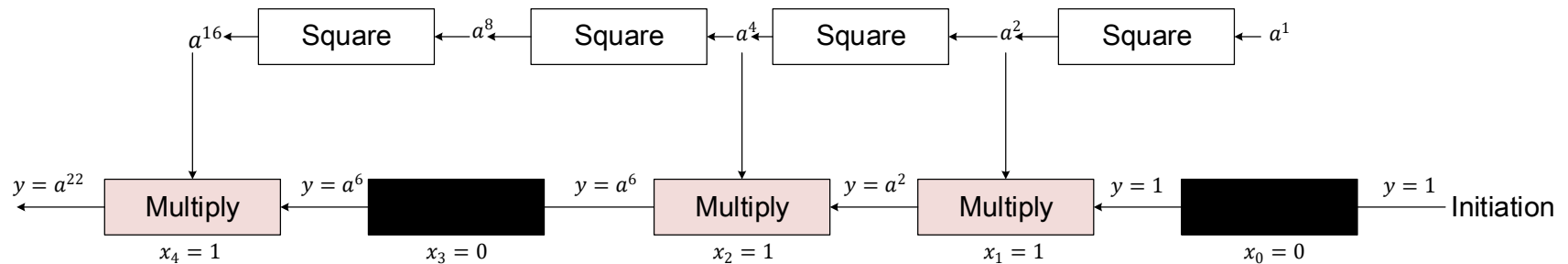
     $a \leftarrow a^2 \bmod n$                         //마지막 반복에서는 제곱이 필요 없다.
  }
  return  $y$ 
}
```

- 복잡도
 - 비트 연산 복잡도는 $O(n_b)$ 임

지수와 로그

- 고속 지수계산 (4/4)
- 예제 9.6

$y = a^x$ 에서 x 가 $22=(10110)_2$ 일 때 절차를 보여라



• 풀이

- $x = 22 = (10110)_2$ 이므로 각 비트를 확인하여 비트가 1인 위치의 a 의 거듭제곱만 선택해 곱하여 $a^{22} = a^{16} \cdot a^4 \cdot a^2$ 로 계산됨
- 검은색 사각형은 해당 비트가 0이라서 곱셈을 하지않고 지나가는 단계를 의미함

지수와 로그

- 로그(Logarithm)

- 정의

$$x = \log_a y$$

- 지수의 역연산으로, $\log_a y = x$ 는 $a^x = y$ 를 만족하는 지수 x 를 의미함 ($a > 0, a \neq 1, y > 0$)

- 적용 사례

- Diffie-Hellman에서는 개인키 x 로부터 공개값 $g^x \bmod p$ 를 생성하고, 공격자가 이 공개값으로부터 x 를 찾기 어렵다는 이산 로그 문제를 보안 기반으로 사용함

지수와 로그

- 전부 찾기(Exhaustive Search)

- 정의

- 가능한 지수 x 를 하나씩 대입하여 $y \equiv a^x \bmod n$ 을 만족하는 값을 찾는 방법

- 알고리즘

```
Modular_Logarithm ( $a, y, n$ )  
{  
  for ( $x = 1$  to  $n - 1$ )  
  {  
    if( $y \equiv a^x \bmod n$ ) return  $x$   
  }  
  return failure  
}
```

- 복잡도

- 비트 연산 복잡도는 $O(2^{n_b})$ 임

지수와 로그

- 이산 대수(Discrete Logarithm)

- 정의

- $y \equiv a^x \pmod{n}$ 에서 a 를 몇 번 곱해야 y 와 같은 나머지가 되는지를 나타내는 지수 x

- 이산 대수를 이해하기 위한 성질

- 유한 곱셈 군(Finite Multiplicative Group)
 - 군의 크기(Group Order)
 - 원소의 위수(Order of an Element)
 - 오일러 정리(Euler's Theorem)
 - 원시근(Primitive Root)
 - 순환 군(Cyclic Group)

지수와 로그

- 이산 대수를 이해하기 위한 성질 (1/9)
- 유한 곱셈 군(Finite Multiplicative Group)
 - 유한 곱셈군은 곱셈 연산에 대해 닫혀 있는 유한한 원소들의 집합임
 - $G = \langle \mathbb{Z}_n^*, \times \rangle$ 형태의 군에서 $\mathbb{Z}_n^*$ 는 n 과 서로소인 $n - 1$ 까지의 정수들로 구성되고 항등원은 1임
 - 모듈로 값 n 이 소수 p 일 때는 $G = \langle \mathbb{Z}_p^*, \times \rangle$ 가 됨

지수와 로그

- 이산 대수를 이해하기 위한 성질 (2/9)

- 군의 크기

- 군에 속하는 원소의 개수를 의미하며 $|G|$ 로 나타냄
- $G = \langle \mathbb{Z}_n^*, \times \rangle$ 의 크기는 $\varphi(n)$ 임
- 예제 9.7

군 $G = \langle \mathbb{Z}_{21}^*, \times \rangle$ 의 크기는 얼마인가?

- 풀이

- $|G| = \varphi(21) = \varphi(3) \times \varphi(7) = 2 \times 6 = 12$ 이고
이 군은 12개의 원소인 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20으로 이루어짐

지수와 로그

• 이산 대수를 이해하기 위한 성질 (3/9)

• 원소의 위수

- $G = \langle \mathbb{Z}_n^*, \times \rangle$ 에서 원소 a 의 위수는 $a^i \equiv e \pmod{n}$ 이 되는 정수 중에서 가장 작은 정수 i 로 정의되고 항등원 e 의 위수는 1임
- 예제 9.8

군 $G = \langle \mathbb{Z}_{10}^*, \times \rangle$ 의 모든 원소에 대한 위수를 계산하시오.

• 풀이

- 이 군은 $\varphi(10) = 4$ 개의 원소 1, 3, 5, 7을 가짐
- 라그랑주 정리를 이용하여 군의 크기를 나누면 4를 나누는 수는 오직 1, 2, 4이므로 원소의 위수를 계산하기 위해 3가지 수에 대해서만 거듭제곱수를 계산함

라그랑주 정리: 유한군에서 부분군의 크기는 전체 군의 크기를 나눈다는 정리

- $1^1 \equiv 1 \pmod{10} ; \rightarrow \text{ord}(1) = 1$
- $3^1 \equiv 3 \pmod{10} ; 3^2 \equiv 9 \pmod{10} ; 3^4 \equiv 1 \pmod{10} \rightarrow \text{ord}(3) = 4$
- $7^1 \equiv 7 \pmod{10} ; 7^2 \equiv 9 \pmod{10} ; 7^4 \equiv 1 \pmod{10} \rightarrow \text{ord}(7) = 4$
- $9^1 \equiv 9 \pmod{10} ; 9^2 \equiv 1 \pmod{10} ; \rightarrow \text{ord}(9) = 2$

지수와 로그

• 이산 대수를 이해하기 위한 성질 (4/9)

• 오일러 정리

- a 가 군 $G = \langle \mathbb{Z}_n^*, \times \rangle$ 의 원소라면 $a^{\varphi(n)} \equiv 1 \pmod n$ 임

- $i = \varphi(n)$ 일 경우 $a^i \equiv 1 \pmod n$ 이 성립됨

• 예제 9.9

군 $G = \langle \mathbb{Z}_8^*, \times \rangle$ 에 대한 $a^i \equiv 1 \pmod 8$ 의 값을 나타내시오.

$\varphi(8) = 4$ 이고 원소는 1, 3, 5, 7 이라는 점에 유의

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$
$a = 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$
$a = 3$	$x: 3$	$x: 1$	$x: 3$	$x: 1$	$x: 3$	$x: 1$	$x: 3$
$a = 5$	$x: 5$	$x: 1$	$x: 5$	$x: 1$	$x: 5$	$x: 1$	$x: 5$
$a = 7$	$x: 7$	$x: 1$	$x: 7$	$x: 1$	$x: 7$	$x: 1$	$x: 7$

• 풀이

- 음영으로 처리된 부분은 오일러 정리를 적용한 결과이며 $i = \varphi(8) = 4$ 이고 모든 a 에 대해서 그 결과는 $x = 1$ 임
- 어떤 원소 a 를 계속 거듭제곱했을 때, $a^i \equiv 1 \pmod n$ 이 처음 성립하는 i 값이 그 원소의 위수임
- 각 원소의 위수는 $\text{ord}(1) = 1, \text{ord}(3) = 2, \text{ord}(5) = 2, \text{ord}(7) = 2$ 임

지수와 로그

- 이산 대수를 이해하기 위한 성질 (5/9)

- 원시근(Primitive Root) (1/4)

- $G = \langle \mathbb{Z}_n^*, \times \rangle$ 에서 위수의 군의 크기 $|G|$ 와 같은 원소를 의미
 - $\text{ord}(a) = |G|$ 이면 a 를 군의 원시근이라고 함

- 예제 9.10

군 $G = \langle \mathbb{Z}_8^*, \times \rangle$ 에 원시근을 구하시오.

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$
$a = 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$
$a = 3$	$x: 3$	$x: 1$	$x: 3$	$x: 1$	$x: 3$	$x: 1$	$x: 3$
$a = 5$	$x: 5$	$x: 1$	$x: 5$	$x: 1$	$x: 5$	$x: 1$	$x: 5$
$a = 7$	$x: 7$	$x: 1$	$x: 7$	$x: 1$	$x: 7$	$x: 1$	$x: 7$

- 풀이

- 어떠한 원소도 위수가 $\varphi(8) = 4$ 가 되지 않고 위수가 모두 4보다 작으므로 원시근을 가지지 않음

지수와 로그

- 이산 대수를 이해하기 위한 성질 (6/9)
- 원시근(Primitive Root) (2/4)
 - 예제 9.11

군 $G = \langle \mathbb{Z}_7^*, \times \rangle$ 에 대한 $a^i \equiv x \pmod{7}$ 의 값을 계산해 놓은 표이다. 이 군에 원시근을 구하시오

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
$a = 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$	$x: 1$
$a = 2$	$x: 2$	$x: 4$	$x: 1$	$x: 2$	$x: 4$	$x: 1$
원시근 $\longrightarrow$ $a = 3$	$x: 3$	$x: 2$	$x: 6$	$x: 4$	$x: 5$	$x: 1$
$a = 4$	$x: 4$	$x: 2$	$x: 1$	$x: 4$	$x: 2$	$x: 1$
원시근 $\longrightarrow$ $a = 5$	$x: 5$	$x: 4$	$x: 6$	$x: 2$	$x: 3$	$x: 1$
$a = 6$	$x: 6$	$x: 1$	$x: 6$	$x: 1$	$x: 6$	$x: 1$

- 풀이

- 군 크기는 $\varphi(7) = 6$ 이고 위수를 살펴보면 $\text{ord}(1) = 1, \text{ord}(2) = 3, \text{ord}(3) = 6, \text{ord}(4) = 3, \text{ord}(5) = 6, \text{ord}(6) = 2$ 임
- 두 원소 3과 5만이 $i = \varphi(n) = 6$ 에서 위수 6을 가지므로 군은 2개의 원시근 3과 5를 가짐

지수와 로그

- 이산 대수를 이해하기 위한 성질 (7/9)

- 원시근(Primitive Root) (3/4)

- 예제 9.12

p : 2가 아닌 소수
 t : 소수 p 가 몇 번 거듭제곱 되었는지를 나타내는 지수

$n = 17, 20, 38, 50$ 중 어떤 n 에 대해서 군 $G = \langle \mathbb{Z}_n^*, \times \rangle$ 가 원시근을 갖는가?

- 풀이

- 원시근은 $\mathbb{Z}_n^*$ 전체를 생성하는 원소이므로 $\mathbb{Z}_n^*$ 가 순환군일 때만 존재함
 - 정수론에서 $\mathbb{Z}_n^*$ 가 순환군이 되는 경우는 $n = 2, 4, p^t, 2p^t$ 형태뿐임
- a. $G = \langle \mathbb{Z}_{17}^*, \times \rangle$ 에서 원시근을 가짐
 - $17 = 17^1$ ($p = 17, t = 1$)이기 때문임
- b. $G = \langle \mathbb{Z}_{20}^*, \times \rangle$ 은 원시근을 가지지 않음
 - $20 = 4 \times 5$ 이기 때문에 조건에 없음
- c. $G = \langle \mathbb{Z}_{38}^*, \times \rangle$ 에서 원시근을 가짐
 - $38 = 2 \times 19^1$ ($p = 19, t = 1$)이기 때문임
- d. $G = \langle \mathbb{Z}_{50}^*, \times \rangle$ 에서 원시근을 가짐
 - $50 = 2 \times 5^2$ ($p = 5, t = 2$)이기 때문임

지수와 로그

- 이산 대수를 이해하기 위한 성질 (8/9)

- 원시근(Primitive Root) (4/4)

- 성질

만약 군 $G = \langle \mathbb{Z}_n^*, \times \rangle$ 이 원시근을 갖는다면, 원시근 개수는 $\varphi(\varphi(n))$ 개다.

- 성질에 관한 질문

1. 주어진 원소 a 와 $G = \langle \mathbb{Z}_n^*, \times \rangle$ 에 대해서 a 가 G 의 원시근인지 아닌지를 어떻게 알아낼 수 있는가?
 - 주어진 원소 a 가 원시근인지 확인하기 위해서는 $\varphi(n)$ 을 구하고, $\text{ord}(a) = \varphi(n)$ 인지 점검해야 함
2. 주어진 군 $G = \langle \mathbb{Z}_n^*, \times \rangle$ 에 대해서 G 의 모든 원시근을 알아낼 수 있을까?
 - 원시근의 개수는 $\varphi(\varphi(n))$ 으로 알 수 있지만, 실제 어떤 원소들이 원시근인지는 각 원소의 위수를 확인해야 함
3. 주어진 군 $G = \langle \mathbb{Z}_n^*, \times \rangle$ 에 대해서 G 의 한 원시근을 어떻게 선택할 것인가?
 - $\text{ord}(a) = \varphi(n)$ 을 만족하는 원소를 찾고, 그 중 하나를 원시근으로 선택함

지수와 로그

- 이산 대수를 이해하기 위한 성질 (9/9)

- 순환 군(Cyclic Group)

- 군 $G = \langle Z_n^*, \times \rangle$ 가 원시군을 갖는 군을 말함
- 만약 g 가 이 군의 한 원시군이라면 군 전체 집합 Z_n^* 을 $Z_n^* = \{g^1, g^2, g^3, \dots, g^{\varphi(n)}\} \bmod n$ 으로 나타낼 수 있음
 - 이때 g 를 원시근 또는 생성자라고 함
- 예제 9.13

군 $G = \langle Z_{10}^*, \times \rangle$ 는 $\varphi(10) = 4$ 이고 $\varphi(\varphi(10)) = 2$ 이므로 두 개의 원시근을 갖는다. 이 두 개의 원시근은 3과 7이라는 것을 구해낼 수 있다. 이 두 개의 원시근이 각각 어떻게 전체 군의 집합을 생성해내는지 알아보자.

- 풀이

- $g = 3 \rightarrow g^1 \bmod 10 = 3, g^2 \bmod 10 = 9, g^3 \bmod 10 = 7, g^4 \bmod 10 = 1, \dots$
 $g = 7 \rightarrow g^1 \bmod 10 = 7, g^2 \bmod 10 = 9, g^3 \bmod 10 = 3, g^4 \bmod 10 = 1, \dots$
- $Z_{10}^* = \{1, 3, 7, 9\}$ 이고 3과 7은 Z_{10}^* 전체를 생성하므로 원시근임

지수와 로그

- 이산 대수의 아이디어

- 군 $G = \langle \mathbb{Z}_p^*, \times \rangle$ 의 성질

1. 이 군의 원소는 1부터 $p-1$ 까지의 모든 수임
2. 이 군은 항상 원시군을 가짐
3. 이 군은 순환 군이고 원소들은 g^x 를 사용하여 생성해 낼 수 있음
 - x 는 1부터 $\varphi(n) = p - 1$ 안의 정수임
4. 원시군은 로그의 밑수로 생각할 수 있음
 - 만약 군이 k 개의 원시군을 가진다고 한다면 k 개의 서로 다른 밑수를 가지고 계산이 가능함
 - 집합 y 에 대해 $x = \log_g y$ 라고 한다면 g 를 밑수로 갖는 y 의 로그인 새로운 원소 x 가 존재함
 - 이와 같은 로그를 이산 대수라고 함
 - 밑수 g 를 나타내기 위해서 $L_g y$ 라고 표현함

지수와 로그

- 이산 대수의 도표화 (1/2)

- y 가 주어졌을 때 $y = a^x \bmod n$ 과 같은 문제에서 x 를 풀기 위해 서로 다른 밑수별로 Z_p^* 에 대한 표를 사용함
 - e.g., 군 $G = \langle Z_7^*, \times \rangle$ 에 대한 이산 대수 표

y	1	2	3	4	5	6
$x = L_3 y$	6	2	1	4	5	3
$x = L_5 y$	6	4	5	2	1	3

- 같은 y 값이라도 밑수 g 에 따라 이산 로그값 x 가 달라질 수 있음

지수와 로그

- 이산 대수의 도표화 (2/2)

- 예제 9.14

다음경우 각각에 대해 x 를 구하시오.

a. $4 \equiv 3^x \text{ mod } 7$

b. $6 \equiv 5^x \text{ mod } 7$

- 풀이

y	1	2	3	4	5	6
$x = L_3 y$	6	2	1	4	5	3
$x = L_5 y$	6	4	5	2	1	3

a. $4 \equiv 3^x \text{ mod } 7 \rightarrow x = L_3 4 \text{ mod } 7 = 4 \text{ mod } 7$

b. $6 \equiv 5^x \text{ mod } 7 \rightarrow x = L_5 6 \text{ mod } 7 = 3 \text{ mod } 7$

지수와 로그

- 이산 대수 복잡도

- $y \equiv a^x \bmod n$ 에서 y 를 구하는 지수계산은 빠르지만, 주어진 y 로부터 지수 x 를 찾는 것은 가능한 x 를 하나씩 확인해야 하므로 n 이 커질수록 계산이 어려움
- 이산 로그 문제의 복잡도는 소인수분해 문제와 유사하게 큰 수에서 효율적으로 풀기 어려운 수준이므로 공개키 암호의 보안 기반으로 사용됨
 - 단순 소인수 분해의 복잡도 $O(2^{n_b/2})$

지수와 로그

- 전통적인 로그와 이산 대수의 비교
 - 이산 대수는 전통적인 로그 같은 성질을 가짐
 - 모듈로 값으로 n 대신에 $\varphi(n)$ 을 사용함

전통적인 로그	이산 대수
$\log_a 1 = 0$	$L_g 1 \equiv 0 \pmod{\varphi(n)}$
$\log_a(x \times y) = \log_a x + \log_a y$	$L_g(x \times y) \equiv (L_q x + L_q y) \pmod{\varphi(n)}$
$\log_a x^k = k \times \log_a x$	$L_g x^k \equiv k \times L_q x \pmod{\varphi(n)}$

- 이산 대수는 전통적인 로그처럼 곱셈과 거듭제곱에 대한 유사한 성질을 가지지만, 모듈러 환경에서 계산되기 때문에 $\varphi(n)$ 을 기준으로 연산됨

Thanks!

김민식(ms6127@naver.com)