2010-1학기 현대암호학

5장 참고자료

정수론

박 종 혁

Tel: 970-6702
Email: jhpark1@snut.ac.kr

목 차

1. 소솟수와 서로소

- 약수(divisors)
  - 어떤 수 m에 대해서 a=mb라면
    - b≠0 일때 b는 a를 나눈다=> 약수
    - 여기서 a, b, m은 정수
    - 나눗셈에서 나머지가 없을 때 약수라고 함
  - 예) 24의 양의 약수
    - 1, 2, 3, 4, 6, 8, 12, 24

- 약수(계속)
  - 만약 a|1 이라면, 그때 a=±1
  - 만약 a|b 이고 b|a 라면, 그때 a=±b
  - 어떠한 0이 아닌 b도 0을 나눈다
  - 만약 b|g이고 b|h라면, 그때 임의의 m과 n에 대해 b|(mg+nh)
    - 만약 b|g라면, 그때 g는 어떠한 정수 g1에 대해 g=b*g1을 만족
    - 만약 b|h라면, 그때 h는 어떠한 정수 h1에 대해 h=b*h1을 만족
    - 또한

      mg+nh = mbg1 + nbh1 = b*(mg1+nh1)

      => b는 mg+mh 를 나눈다

- 약수(계속)
  - 예) b=7; g=14; h=63; m=3; n=2
  - 만약 b|g이고 b|h라면, 그때 임의의 m과 n에 대해 b|(mg+nh)
    - 7|14와 7|63에 대해서 7|(3*14+2*63)
    - mg+nh = mbg1 + nbh1 = b*(mg1+nh1)
    - g = b*g1, g1= g/b = 2
    - h = b*h1, h1= h/b = 9
    - (3*14+2*63) = 7*(3*2+2*9)계산 가능
    - 이 계산은 7|7*(3*2+2*9))가 성립함

- 솟수(prime numbers)
  - 정수 p>1이 만약 단지 약수들로서 ±1과 ±p만을 가진다면 p는 솟수이다.
  - 266페이지 표 7.1 200미만의 솟수 참조
  - 어떠한 정수 a>1은 다음과 같은 유일한 방법으로 인수분해 가능
    - $a = p_1^{\alpha 1} p_2^{\alpha 2} \ldots p_t^{\alpha t}$
    - $p_t > p_{t-1} > \ldots > p_1$ 는 솟수, $\alpha_i > 0$
    - 예) 91
      - 2, 3, 4, 5, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89
      - 7*13
    - $10164 = 7*11^2 *12$

- 솟수(계속)
  - 만약 P가 모든 솟수들의 집합이라면 어떤 양의 정수는 다음의 형태에서처럼 유일하게 표시될 수 있다

    $a = \pi\ p^{ap}$, 여기서 $ap >= 0$

    - a의 어떤 특별한 값에 있어서 멱 지수 ap의 대부분은 0
    - 예) $3600 => 2^4 * 3^2 * 5^2$

- 솟수(계속)
  - 정수 12
    - {a2 = 2, a3 = 1} => 2^2*3^1
  - 정수 18
    - {a2 = 1, a3 = 2} => 2^1*3^2
  - 두수의 곱셈은 해당하는 멱 지수를 더하는 것과 같다
    - $12*18 = (2^2*3^1)*(2^1*3^2) = (2^3*3^3) = 216$
  - 이러한 솟수들의 관점에서 a|b의 의미는???

- 솟수(계속)
  - 이러한 솟수들의 관점에서 a|b의 의미는???
    - p^k 형태를 가지는 어떠한 정수는 단지 그것보다 작거나 같은 지수를 가지는 솟수 p^j, 단 j<=k 에 의해 나누어 질 수 있다
    - 즉, 모든 p에 대해 a|b -> $a_p <= b_p$
    - 예) a=12, b=36

      $12 = 2^2*3^1$ $36 = 2^2*3^2$
      - $a_2 = 2 = b_2$
      - $a_3 = 1<=2 = b_3$

- 서로소(relatively prime number)
  - 어떤 두 수가 공통적인 소인수를 갖지 못할 때 두 수를 서로소 라고 한다.
  - 공통적인 소인수 => 최대 공약수 => GCD
  - 양의 정수 c가 다음의 조건을 만족한다면 c는 a와 b의 최대 공약수
    - c는 a와 b의 약수
    - a와 b에 대한 어떠한 약수는 c의 약수
  - GCD(a,b)= max[k, 이때 k는 k|a이고 k|b]

- 서로소(계속)
  - 암호학에서 요구하는 최대 공약수는 양수
    - GCD(a,b)=GCD(a,−b)=GCD(−a,b)=GCD(−a,−b)=GCD(|a|,|b|)
    - GCD(60,24)=GCD(60,−24)=12
    - 모든 0이 아닌 정수들은 0을 나누기 때문에 GCD(a,0)=|a|
  - 모든 정수의 솟수 표현법을 이용하면 최대 공약수 결정이 쉬워짐
    - $300 = 2^2*3^1*5^2$ $18 = 2^1*3^2$
    - $GCD(300,18) = 2^1*3^1*5^0 = 6$

- 서로소(계속)
  - 8과 15는 서로 소인가?
    - 8은 약수로 1, 2, 4, 8 가짐
    - 15는 약수로 1, 3, 5, 15 가짐

    => 동일한 약수를 가지지 못하므로 서로 소임

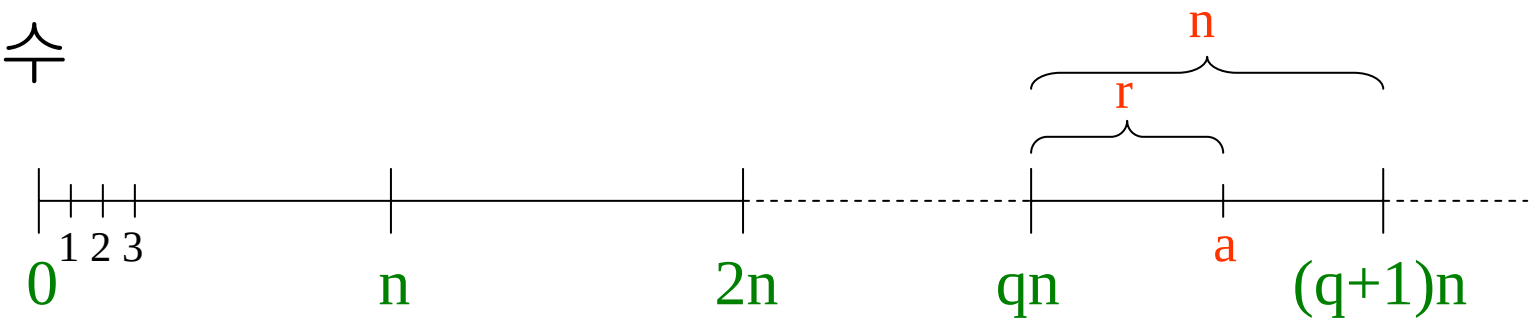
2. 모듈러 연산

- 어떤 양의 정수 n과 어떤 정수 a가 주어지고, 만약 a를 n으로 나눈다면 다음과 같은 관계를 가지는 몫 q와 나머지 r을 얻는다

$$a = qn + r \qquad 0 <= r < n$$

$$q = \lfloor a/n \rfloor \qquad a \equiv r \bmod n$$

=> 여기서 $\lfloor x \rfloor$ 는 x보다 작거나 또는 같은 가장 큰 정수



- 예제)
  - a = qn + r
  - a=11, n=7

    11 = 1*7 + 4 = 4 mod 7

    4 = 11 mod 7

    11 ≡ 4 mod 7

  - a= −11, n=7

    −11 = 2*7 − 3 = 3 mod 7

    3 = −11 mod 7

    −11 ≡ 3 mod 7

- 합동
  - (a mod n) = (b mod n), 두 정수 a와 b는 modulo n에 대해 합동
  - $a \equiv b \bmod n$ 으로 표기
  - $a \equiv 0 \bmod n$ 이라면 그때 $n|a$ 이다

- 모듈러 연산자의 특성
  1. 만약 $n|(a-b)$ 라면, $a \equiv b \bmod n$
  2. (a mod n) = (b mod n)은 $a \equiv b \bmod n$
  3. $a \equiv b \bmod n$은 $b \equiv a \bmod n$ 을 의미
  4. $a \equiv b \bmod n$과 $b \equiv c \bmod n$ 은 $a \equiv c \bmod n$ 을 의미

1. 만약 n|(a-b) 라면, $a \equiv b \bmod n$

   n = 5, a = 23, b = 8

   23 – 8 = 15 = 5*3 이기 때문에 $23 \equiv 8 \bmod 5$

모듈러 산술 연산

- mod n 연산
  - 정수들의 범위 => {0, 1, ..., (n-1)}으로 표현가능
  - 즉, 이러한 집합의 범위 내에서 산술 연산이 가능

  - 모듈러 연산의 특징

1. [(a mod n)+(b mod n)] mod n = (a+b) mod n
2. [(a mod n)-(b mod n)] mod n = (a-b) mod n
3. [(a mod n)*(b mod n)] mod n = (a*b) mod n

- 예제) a = 11, b = 15, n = 8

1. $(a + b) \bmod n = ((11 \bmod 8) + (15 \bmod 8)) \bmod 8$
   $= 3 + 7 \bmod 8$
   $= 10 \bmod 8 = 2$
2. $(a - b) \bmod n = ((11 \bmod 8) - (15 \bmod 8)) \bmod 8$
   $= 3 - 7 \bmod 8$
   $= -4 \bmod 8 = 4$
3. $(a * b) \bmod n = ((11 \bmod 8) * (15 \bmod 8)) \bmod 8$
   $= 3 * 7 \bmod 8$
   $= 21 \bmod 8 = 5$

- 지수 연산 => 곱셈의 반복으로 수행가능
- 예) $11^7 \bmod 13$

$11^2 = 121 = 4 \bmod 13$

$11^4 = 4^2 = 3 \bmod 13$

$11^7 = 11*4*3 = 2 \bmod 13$

- 모듈러 연산의 속성
  - 교환법칙
    - (w+x) mod n = (x+w) mod n
    - (w*x) mod n = (x*w) mod n
  - 결합법칙
    - [(w+x)+y] mod n = [w+(x+y)] mod n
    - [(w*x)*y] mod n = [w*(x*y)] mod n
  - 분배법칙
    - [w*(x+y)] mod n = [(w*x)+(w*y)] mod n
  - 항등원
    - (0+w) mod n = w mod n
    - (1*w) mod n = w mod n
  - 덧셈에 대한 역원 (−w)

3. 페르마와 오일러의 정리

- 페르마 정리(Fermat Theorem)

  만약 p가 소수라면 a는 p에 의해 나누어지지 않는 양의 정수이다. 그때

  $$a^{p-1} \equiv 1 \bmod p$$

  가 성립한다

- 예제) a=7, p= 19
  - $a^{p-1} \equiv 1 \bmod p$
  - $7^2 = 49 \equiv 11 \bmod 19$
  - $7^4 = 121 \equiv 7 \bmod 19$
  - $7^8 = 49 \equiv 11 \bmod 19$
  - $7^{16} = 121 \equiv 7 \bmod 19$
  - $a^{p-1} = 7^{18} = 7^{16} * 7^2 \equiv 7*11 \equiv 1 \bmod 19$

- 페르마의 다른 유용한 형태

  만약 p가 솟수이고 a가 양의 정수라면

  $$a^p \equiv a \bmod p$$

  가 성립한다

- 예) a=3, p=5, $3^5 = 243 \equiv 3 \bmod 5$

- 예) a=10, p=5, $10^5 = 100000 \equiv 10 \bmod 5 \equiv 0 \bmod 5$

오일러 정리

- 오일러의 Totient 함수

  정수론에서 오일러의 totient 함수는 φ(n)라고 표기된다

  φ(n): n보다 작고 n과 서로 소인 양의 정수의 개수

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
φ(n)	1	1	2	2	4	2	6	4	6	4	10	4	12	6

오일러 정리

- 오일러 함수의 특성
  - $\varphi(1) = 1$
  - 솟수 n에 대해서
    $\varphi(n) = n-1$
  - 솟수 p와 q에 대해서 n=pq
    $\varphi(n) = \varphi(pq) = \varphi(p)* \varphi(q) = (p-1)(q-1)$

오일러 정리

- 오일러 정리

  서로소인 모든 a와 n에 대한 관계를 나타낸다

  $a^{\varphi(n)} \equiv 1 \bmod n$

  $a=3;\ n=10;\ \varphi(10) = 4;\ 3^4 = 81 \equiv 1 \bmod 10$

  $a=2;\ n=11;\ \varphi(11) = 10;\ 3^{10} = 1024 \equiv 1 \bmod 11$

- 오일러 정리의 추가적인 특성
  - $a^{\varphi(n)+1} \equiv a \bmod n$

4. 이산대수

- 다음과 같은 등식을 고려해 보자

  $y = g^x \bmod p$

  g, x 그리고 p가 주어진다면, y를 계산하는 것은 쉬운 문제이다. 최악의 경우, 반복적인 곱셈과정을 x번 수행해야만 하며, 효율적인 계산을 위한 알고리즘이 존재한다.

  그러나 y, g 그리고 p가 주어진다고 하더라도 x를 계산하는 것은 매우 어려운 문제이다. 그 어려움은 RSA알고리즘을 풀기 위해 요구되는 인수분해의 어려움과 같은 어려움을 가진다.

RSA 알고리즘 예

암호화와 복호화

: 평문 메시지 M = 19 일 경우

- 암호문 : $19^5 = 66 \bmod 119 \Rightarrow 66$
- 복호문 : $66^{77} = 19 \bmod 119 \Rightarrow 19$