암호수학 1

암호 수학 기초: 나눗셈, 가분성

- $\mathbf{22} = 4 \times \mathbf{5} + 2$
  - 22를 5로 나누면 몫은 4이고 나머지는 2이다
    - 22(dividend)를 5(divisor)로 나누면 몫(quotient)은 4이고 나머지(remainder)는 2이다
- 나눗셈정리
  - 임의의 정수 a, 양의 정수 b에 대해 $\boldsymbol{a = q \times b + r}$ $(0 \leq r < b)$을 만족하는 정수 q와 r이 유일하게 존재한다
    - q와 r은 a를 b로 나눈 몫과 나머지
- 가분성(divisibility)
  - $\mathbf{10} = 2 \times \mathbf{5}$ → 10은 5로 나누어짐 → 표기법 → $\mathbf{5 \mid 10}$
  - $\mathbf{11} = 2 \times \mathbf{5} + 1$ → 11은 5로 나누어지지 않음 → 표기법 → $\mathbf{5 \nmid 11}$

Reference: 수리암호학개론, 김명환, 2019

암호 수학 기초: 약수, 배수

- 약수(divisor, factor), 배수(multiple)
  - 두 정수 $a(\neq 0),\ b$에 대해 $a \mid b$일 때
    - 즉, $b = aq$인 정수 q가 존재할 때
    - a는 b의 약수(divisor)
    - b는 a의 배수(multiple)
  - 예
    - $\mathbf{10} = \mathbf{5} \times 2$
      - 5는 10을 (나머지 없이) 나누는 수이다. 5는 10의 약수이며 10은 5의 배수이다. 즉 $\mathbf{5} \mid \mathbf{10}$
    - $\mathbf{0} = \boldsymbol{a} \times 0$
      - 0이 아닌 임의의 정수 a는 0을 나눈다(0으로 나누는 것은 정의되지 않음). 0이 아닌 모든 정수는 0의 약수이다. 즉 $\boldsymbol{a} \mid \mathbf{0}$
    - $\boldsymbol{a} = \mathbf{1} \times a$
      - 1은 모든 정수의 약수이다. 즉 $\mathbf{1} \mid \boldsymbol{a}$
    - $\mathbf{2}$의 모든 약수는 $1,\ -1,\ 2,\ -2$
      - 음의 약수도 가능함

Reference: 수리암호학개론, 김명환, 2019

암호 수학 기초: 최대공약수, 서로소

- 최대공약수(Greatest Common Divisor, gcd)
  - 두 정수 a, b의 공약수 중 가장 큰 것
    - 두 정수 a, b 중 어느 하나는 0이 아니어야 최대공약수 존재 가능
      - a, b 모두 0인 경우 0 아닌 모든 정수가 공약수가 되므로 gcd 선택 불가
    - 두 정수 a, b의 최대공약수를 $\gcd(a, b)$로 표기
    - $\gcd(a, b) = \gcd(-a, b) = \gcd(a, -b) = \gcd(-a, -b) = \gcd(|a|, |b|)$
    - 양수 a에 대해, $\mathbf{gcd(a, 0) = a}$
      - $\gcd(5, 0)=5$
    - 36과 27의 최대공약수 $\gcd(36, 27) = 9$
      - 36의 (양의) 약수: 1, 2, 3, 4, 6, **9**, 12, 18, 36
      - 27의 (양의) 약수: 1, 3, **9**, 27
- 서로소 (coprime, relatively prime)
  - 최대공약수가 1인 두 정수는 서로소(coprime)이다
  - 1 외의 다른 공약수를 갖지 않는 두 정수는 서로소
    - 6과 8은 서로소가 아니다
      - $6 = 2 \cdot 3,\ 8 = 2 \cdot 2 \cdot 2$ → 1 외의 공약수 2를 가짐
    - 16과 27은 서로소이다
      - $16 = 2 \cdot 2 \cdot 2 \cdot 2,\ 27 = 3 \cdot 3 \cdot 3$ → 1 외의 공약수 없음
      - 16과 27은 둘 다 소수가 아니지만 서로소인 관계에 있음

Reference: 수리암호학개론, 김명환, 2019

암호 수학 기초: 모듈로 연산

- 모듈로(modulo) 연산자
  - 임의의 정수 a, 양의 정수 n에 대해 아래 모듈로 연산자 mod는 a를 n으로 나눈 나머지를 계산한다

$$\boldsymbol{a \bmod n = r}$$

  - 예
    - $17 \bmod 3 = 2$
    - $15 \bmod 3 = 0$
    - $3 \bmod 5 = 3$
    - $-3 \bmod 5 = 2$
      - $-3 = (-1) \cdot 5 + 2$
      - $(-3 + 5)\ mod\ 5 = 2$

암호 수학 기초: 법 n에 대해 합동

Reference: (정민석, 2017; 최은미, 2019)

- 법 n에 대해 합동(congruence)
  - 정수 a, b, n에 대해 $n \mid (a-b)$ 인 경우 $a,\ b$는 **법 n에 대해 합동**이라고 하며 다음과 같이 표기

$$\boldsymbol{a \equiv b\ (mod\ n)}$$

    - n으로 나눈 나머지가 같다는 의미
      - $(a\ mod\ n) = (b\ mod\ n)$
    - $n \nmid (a-b)$이면 $a \not\equiv b(mod\ n)$로 표기

(Reference: 최은미, 정수와 암호론, 2019)
$a = q_1 n + r_1$, $b = q_2 n + r_2$라고 하면,
- $a \equiv b(mod\ n)$이면 $n \mid (a-b)$이므로 $n \mid (r_1 - r_2)$로부터 $(r_1 - r_2)$가 n 의 배수이나 n보다 클 수 없으므로 $(r_1 - r_2) = 0$이 되어 a, b의 n으로 나눈 나머지는 같다
- a, b의 n으로 나눈 나머지가 같다면 $(a-b) = (q_1 - q_2)n$ 이므로 $a \equiv b(mod\ n)$가 성립

  - 예
    - $20 \equiv 14\ (mod\ 3)$ ← 여기서의 mod는 congruence relation
      - $20 = 6 \cdot 3 + 2,\quad 14 = 4 \cdot 3 + 2 \rightarrow (20-14) = (6-4) \cdot 3 + 0 \rightarrow 3 \mid (20-14)$
      - $(20\ mod\ 3) = (14\ mod\ 3)$
    - $15 \equiv 0\ (mod\ 3)$
    - $-3 \equiv 2\ (mod\ 5)$
    - $-7 \equiv -4 \equiv -1 \equiv 2 \equiv 5 \equiv 8 \equiv 11 (mod\ 3)$

여기서의 두 mod는 이항 연산자

소수, 합성수, 소인수분해, 서로소

Reference: (Forouzan, 2008; 최은미, 2019)

양의 정수	1	• 양의 약수 1개 • 1은 소수가 아니다
	소수	• 1보다 큰 정수 중, 1과 자신 외의 다른 약수를 갖지 않는 수 • 양의 약수 2개 • 예) 2, 3, 5, 7, 11, 13, 17, …
	합성수	• 1보다 큰 정수 중 소수가 아닌 수 • 양의 약수 3개 이상 • 예) 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, …

소인수분해 (prime factorization)
- 1보다 큰 정수를 소수들의 곱으로 표현하는 것
- 예) $6 = 2 \times 3, \quad 8 = 2 \times 2 \times 2, \quad 60 = 2 \times 2 \times 3 \times 5$

산술의 기본 정리 (fundamental theorem of arithmetic)
- 1보다 큰 임의의 정수는 (소수들의 곱의 순서를 무시할 때) 소수들의 곱으로 유일하게 표현된다
- 예) $6 = 2 \times 3, \quad 8 = 2 \times 2 \times 2, \quad 60 = 2 \times 2 \times 3 \times 5$

서로소
- 최대공약수가 1인 두 정수는 서로소(coprime)이다. 즉 $\gcd(a, b) = 1$인 두 정수 a , b는 서로소
- 1은 모든 정수와 서로소이다
- 소수 p에 대해, $1, 2, \ldots, p-1$의 각 수는 p와 서로소이다

암호 수학 기초: Z_n, Z_n^*, Z_p, Z_p^*

Reference: (이민섭, 2008; 최은미, 2019; Forouzan, 2008)

- Z_n
  - Set of all least residues modulo n
  - $Z_n = \{0, 1, 2, \ldots, n-1\}$
    - n으로 나누었을 때 얻어지는 나머지들의 집합
  - Z_n에서 n이 소수 p인 경우 Z_p로 표기
    - $Z_p = \{0, 1, 2, \ldots, p-1\}$
- Z_n^*
  - $Z_n^* = \{x \mid 1 \le x \le n,\ \gcd(x, n) = 1\}$
    - n 이하 양의 정수 중 n과 서로소인 수들의 집합
      - 최대공약수(Greatest common divisor, gcd)가 1인 두 정수는 서로소(coprime)이다
        - 1 외의 다른 공약수를 갖지 않는 두 정수는 서로소
    - $|Z_n^*| = \phi(n)$
  - Z_n^*에서 n이 소수 p인 경우 Z_p^*로 표기
    - $Z_p^* = \{1, 2, \ldots, p-1\}$

$Z_6 = \{0, 1, 2, 3, 4, 5\}$	$Z_6^* = \{1, 5\}$
$Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$	$Z_7^* = \{1, 2, 3, 4, 5, 6\}$
$Z_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$	$Z_{10}^* = \{1, 3, 7, 9\}$

$Z_6^* = \{1, 5\}$		
	양의 약수만 나열	6과의 최대공약수
0	모든 양수	gcd(0, 6)=6
1	**1**	**gcd(1, 6)=1**
2	1, 2	gcd(2, 6)=2
3	1, 3	gcd(3, 6)=3
4	1, 2, 4	gcd(4, 6)=2
5	**1, 5**	**gcd(5, 6)=1**
6	1, 2, 3, 6	

암호 수학 기초: 역원

- $mod\ n$ 연산에서의 역원(inverse)
  - 덧셈의 역원(additive inverse)
    - Z_n에서 $a + b \equiv 0\ (mod\ n)$이면 a, b는 서로에 대해 덧셈의 역원
    - Z_n 내 각 정수는 덧셈의 유일한 역원 존재
      - $Z_4 = \{0, 1, 2, 3\}$
        - $0 + 0 \equiv 0\ (mod\ 4), 1 + 3 \equiv 0\ (mod\ 4), 2 + 2 \equiv 0\ (mod\ 4)$
        - 덧셈에 대해 0의 역원은 0, 1의 역원은 3, 3의 역원은 1, 2의 역원은 2

$Z_4 = \{0, 1, 2, 3\}$				
+	0	1	2	3
0	**0**	1	2	3
1	1	2	3	**0**
2	2	3	**0**	1
3	3	**0**	1	2

  - 곱셈의 역원(multiplicative inverse)
    - Z_n에서 $a \times b \equiv 1\ (mod\ n)$이면 a, b는 서로에 대해 곱셈의 역원
    - Z_n 내 각 정수는 곱셈의 역원을 가질 수도 있고 가지지 않을 수도 있음
      - $Z_4 = \{0, 1, 2, 3\}$
        - $1 \times 1 \equiv 1\ (mod\ 4), 3 \times 3 \equiv 1\ (mod\ 4)$
        - 곱셈에 대해 $1, 3$의 역원만 존재 (1의 역원은 1, 3의 역원은 3)
      - $Z_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
        - $1 \times 1 \equiv 1\ (mod\ 10), 3 \times 7 \equiv 1\ (mod\ 10), 9 \times 9 \equiv 1\ (mod\ 10)$
        - 곱셈에 대해 $1, 3, 7, 9$의 역원만 존재
    - Z_n^* 내 각 정수는 곱셈의 역원 존재
      - $Z_4^* = \{1, 3\}$
      - $Z_{10}^* = \{1, 3, 7, 9\}$
      - $Z_7^* = \{1, 2, 3, 4, 5, 6\}$

$Z_4 = \{0, 1, 2, 3\}$				
×	0	1	2	3
0	0	0	0	0
1	0	**1**	2	3
2	0	2	0	2
3	0	3	2	**1**

유클리드 알고리즘

- 유클리드 알고리즘 (Euclidean algorithm)
  - 두 양의 정수 a, b의 최대공약수(gcd)를 구하는 알고리즘
    - $\gcd(a, b) = \gcd(b, r)$
      - r은 a를 b로 나눈 나머지
    - 양수 a에 대해, $\gcd(a, 0) = a$
  - 예
    - $\gcd(36, 28) = \gcd(28,8) = \gcd(8,4) = \gcd(4,0) = 4$
    - $\gcd(24, 30) = \gcd(30,24) = \gcd(24,6) = \gcd(6,0) = 6$
      - $\gcd(a, b)$에서 a, b의 대소 순서 무관

a=qb+r일 때, gcd(a,b)=d1, gcd(b,r)=d2라고 하면,
- d1 | a, d1 | b이므로 d1 | (a-qb)임. 즉 d1 | r
- d1은 b, r의 공약수이므로 $d1 \leq d2$
- d2는 d2 | b, d2 | r이므로 d2 | (qb+r)임. 즉 d2 | a
- d2는 a, b의 공약수이므로 $d2 \leq d1$
- $d1 \leq d2$이고 $d2 \leq d1$이므로 $d2 = d1$

q	a	b	r		
1	36	28	8	36 =1 x 28 + 8	gcd(36,28)
3	28	8	4	28 = 3 x 8 + 4	gcd(28,8)
2	8	4	0	8 = 2 x 4 + 0	gcd(8,4)
	4	0			gcd(4,0) = 4

q	a	b	r
0	24	30	24
1	30	24	6
4	24	6	0
	6	0	

Reference: https://en.wikipedia.org/wiki/Euclidean_algorithm

유클리드 알고리즘

```
def gcd(a,b):
    r1,r2=a,b
    while r2>0:
        q = r1 // r2
        r = r1 - q*r2
        r1 = r2
        r2 = r
    return r1

print(gcd(36,28))
print(gcd(24,30))
```

q	r1	r2	r
1	36	28	8
3	28	8	4
2	8	4	0
	4	0	

q	r1	r2	r
0	24	30	24
1	30	24	6
4	24	6	0
	6	0	

Reference: https://en.wikipedia.org/wiki/Euclidean_algorithm

베주 항등식

- 베주 항등식(Bézout's identity)
  - 두 정수 a, b에 대해 다음 항등식을 만족하는 정수 s, t가 존재한다
    - $\gcd(a, b) = sa + tb$
      - 그러한 s, t가 유일하게 결정되는 것은 아님
        - $\gcd(15,6) = 3$

$$= 15 \cdot (1) + 6 \cdot (-2)$$
$$= 15 \cdot (1) + 6 \cdot (-2) + 15 \cdot 6 - 15 \cdot 6$$
$$= 15 \times (7) + 6 \times (-17)$$

      - a, b의 최대공약수를 a, b의 배수들의 합으로 표현 가능

Reference: https://en.wikipedia.org/wiki/Bézout's identity
참고: 수리암호학개론, 김명환, 2019

유클리드 알고리즘과 베주 항등식

q	a	b	r		
1	36	28	8	36 =1 x 28 + 8	8 = 36 - 1 x 28
3	28	8	4	28 = 3 x 8 + 4	4 = 28 - 3 x 8
2	8	4	0	8 = 2 x 4 + 0	
	4	0			

a, b의 최대공약수를 a, b의 일차결합으로 표현 가능

$$\begin{aligned}&\gcd(36,28) = 4\\&= 28 - 3 \cdot 8\\&= 28 - 3 \cdot (36 - 1 \cdot 28)\\&= -3 \cdot 36 + 4 \cdot 28\\&= s \cdot a + t \cdot b\end{aligned}$$

Reference: https://en.wikipedia.org/wiki/Bézout's identity
참고: 수리암호학개론, 김명환, 2019

유클리드 알고리즘과 베주 항등식

Reference: (William Stallings, 2014)

r_i는 a , b의 일차결합으로 표현가능

q	a	b	r						
						$r_{-1} = a$	$r_{-1} = a \cdot 1 + b \cdot 0$	$s_{-1} = 1$	$t_{-1} = 0$
						$r_0 = b$	$r_0 = a \cdot 0 + b \cdot 1$	$s_0 = 0$	$t_0 = 1$
1	56	34	22	$22 = 56 - 1 \cdot 34$	$r_1 = a - q_1 \cdot b$	$r_1 = r_{-1} - q_1 \cdot r_0$	$r_1 = a \cdot s_1 + b \cdot t_1$	$s_1 = s_{-1} - q_1 \cdot s_0$	$t_1 = t_{-1} - q_1 \cdot t_0$
1	34	22	12	$12 = 34 - 1 \cdot 22$	$r_2 = b - q_2 \cdot r_1$	$r_2 = r_0 - q_2 \cdot r_1$	$r_2 = a \cdot s_2 + b \cdot t_2$	$s_2 = s_0 - q_2 \cdot s_1$	$t_2 = t_0 - q_2 \cdot t_1$
1	22	12	10	$10 = 22 - 1 \cdot 12$	$r_3 = r_1 - q_3 \cdot r_2$	$r_3 = r_1 - q_3 \cdot r_2$	$r_3 = a \cdot s_3 + b \cdot t_3$	$s_3 = s_1 - q_3 \cdot s_2$	$t_3 = t_1 - q_3 \cdot t_2$
1	12	10	2	$2 = 12 - 1 \cdot 10$	$r_4 = r_2 - q_4 \cdot r_3$	$r_4 = r_2 - q_4 \cdot r_3$	$r_4 = a \cdot s_4 + b \cdot t_4$	$s_4 = s_2 - q_4 \cdot s_3$	$t_4 = t_2 - q_4 \cdot t_3$
5	10	2	0	$0 = 10 - 5 \cdot 2$	$r_5 = r_3 - q_5 \cdot r_4$	$r_5 = r_3 - q_5 \cdot r_4$	$r_5 = a \cdot s_5 + b \cdot t_5$	$s_5 = s_3 - q_5 \cdot s_4$	$t_5 = t_3 - q_5 \cdot t_4$
	2	0							

- q_i → r_{i-2}를 r_{i-1}로 나눈 몫
- 이전 계산된 2개 나머지들로부터 새로운 나머지를 계산 → $r_i = r_{i-2} - q_i \cdot r_{i-1}$
- 이전 계산된 2개 s 값들로부터 새로운 s 값을 계산 → $s_i = s_{i-2} - q_i \cdot s_{i-1}$
- 이전 계산된 2개 t 값들로부터 새로운 t 값을 계산 → $t_i = t_{i-2} - q_i \cdot t_{i-1}$
- 초기값 → $r_{-1} = a, r_0 = b, s_{-1} = 1, s_0 = 0, t_{-1} = 0, t_0 = 1$

$$
\begin{aligned}
& r_5 = r_3 - q_5 \cdot r_4 \\
& = (a \cdot s_3 + b \cdot t_3) - q_5(a \cdot s_4 + b \cdot t_4) \\
& = a(s_3 - q_5 \cdot s_4) + b(t_3 - q_5 \cdot t_4) \\
& = a \cdot s_5 + b \cdot t_5
\end{aligned}
$$

$r_5 = r_3 - q_5 \cdot r_4$

$s_5 = s_3 - q_5 \cdot s_4$

$t_5 = t_3 - q_5 \cdot t_4$

확장 유클리드 알고리즘

- 초기값 → $r_{-1} = a = 56, r_0 = b = 34,\ \ s_{-1} = 1, s_0 = 0,\ \ t_{-1} = 0, t_0 = 1$

q	r1	r2	r	s1	s2	s	t1	t2	t
1	56	34	22	1	0	1	0	1	-1
1	34	22	12	0	1	-1	1	-1	2
1	22	12	10	1	-1	2	-1	2	-3
1	12	10	2	-1	2	-3	2	-3	5
5	10	2	0	2	-3	17	-3	5	-28
	2	0		**-3**	17		**5**	-28	

- $s = s1 - q \cdot s2$ → $-3 = (-1) - 1 \cdot 2$
- $t = t1 - q \cdot t2$ → $5 = 2 - 1 \cdot (-3)$

확장유클리드알고리즘
- $\gcd(56,34) = 2 = 56 \cdot (-3) + 34 \cdot (5)$
- 최대공약수 및 베주항등식의 계수까지 계산 가능

Reference: (William Stallings, 2014)

확장 유클리드 알고리즘

```
def egcd(a,b):
    r1,r2=a,b
    s1,s2=1,0
    t1,t2=0,1
    while r2>0:
        q = r1 // r2
        r = r1 - q*r2;   r1 = r2;   r2 = r;
        s = s1 - q*s2;   s1 = s2;   s2 = s;
        t = t1 - q*t2;   t1 = t2;   t2 = t;
    return (r1,s1,t1)

print(egcd(56,34))
```

Reference: https://en.wikipedia.org/wiki/Euclidean_algorithm

곱셈의 역원 존재 판단

Reference: (정민석, 암호수학, 2017)

- 곱셈의 역원 존재 판단
  - 양의 정수 n과 정수 a에 대해 다음 두 명제는 동치이다
    - 법 n에 대해 a의 곱셈의 역원이 존재한다
    - $\gcd(n, a) = 1$
      - n, a는 서로소(coprime)

Reference: (정민석, 암호수학, 2017)
- a의 곱셈의 역원 t가 존재한다면 $at \equiv 1 \ (mod\ n)$이므로 $n \mid (at - 1)$로부터 $ns = at - 1$이고 $n(-s) + at = 1$이므로 $\gcd(n, a) = 1$이 성립
- $\gcd(n, a) = 1$인 경우 적당한 정수 s, t 에 대해 $ns + at = 1$이고 $n(-s) = at - 1$이므로 $n \mid (at - 1)$이 되어 $at \equiv 1 \ (mod\ n)$이 성립하므로 a는 곱셈의 역원 t를 가짐

곱셈의 역원과 확장 유클리드 알고리즘

Reference: (정민석, 암호수학, 2017)

- 확장 유클리드 알고리즘으로 곱셈의 역원 찾기
  - 법 n과 정수 a 에 대해 $\gcd(n, a) = 1$인 경우 곱셈의 역원 찾기
    - n, a에 대해 확장 유클리드 알고리즘 적용하여 다음 수식의 s, t 결정하면 a의 곱셈의 역원은 t임
      - $\gcd(n, a) = 1 = ns + at$
        - $1 = ns + at$로부터 $at - 1$이 n의 배수임. 즉 $n \mid at - 1$임
        - 따라서 $at \equiv 1 \ (mod\ n)$이므로 법 n에 대한 a의 곱셈의 역원은 t임
        - 또한, 법 a에 대한 n의 곱셈의 역원은 s임
  - 예
    - 법 10에 대해 7의 곱셈의 역원 3 구하기
      - $\gcd(10,7) = 1 = (-2) \cdot 10 + 3 \cdot 7$

q	r1	r2	r	s1	s2	s	t1	t2	t
1	10	7	3	1	0	1	0	1	-1
2	7	3	1	0	1	-2	1	-1	3
3	3	1	0	1	-2	7	-1	3	-10
	1	0		**-2**	7		**3**	-10	

역원만 필요하다면 s의 값 구하는 과정 불필요

q	r1	r2	r	t1	t2	t
1	10	7	3	0	1	-1
2	7	3	1	1	-1	3
3	3	1	0	-1	3	-10
	1	0		**3**	-10	

- 법 10에 대한 7의 곱셈의 역원은 아래 식을 만족하는 $\boldsymbol{t}$를 구하는 것

$$\boldsymbol{t \cdot 7 \equiv 1 \ (mod\ 10)}$$
$$\boldsymbol{10 \mid (1 - t \cdot 7)}$$
$$\boldsymbol{s \cdot 10 = 1 - t \cdot 7}$$
$$\boldsymbol{s \cdot 10 + t \cdot 7 = 1}$$

곱셈의 역원과 확장 유클리드 알고리즘

Reference: (정민석, 암호수학, 2017)

법 10에 대해 9의 곱셈의 역원 구하기

- $\gcd(10,9) = 1 = 10 \cdot 1 + 9 \cdot (-1)$
- 법 10에 대한 9의 역원을 10으로 나눈 나머지로 표현하면 9임 $(9 \equiv -1 \ (mod\ 10))$

q	r1	r2	r	s1	s2	s	t1	t2	t
1	10	9	1	1	0	1	0	1	-1
9	9	1	0	0	1	-9	1	-1	10
	1	0		**1**	-9		**-1**	10	

역원만 필요하다면 s의 값 구하는 과정 불필요

q	r1	r2	r	t1	t2	t
1	10	9	1	0	1	-1
9	9	1	0	1	-1	10
	1	0		**-1**	10	

References

- Behrouz A. Forouzan, Cryptography and Network Security, McGraw-Hill, 2008
- William Stallings, Cryptography and Network Security: Principles and Practice, Sixth Edition, Prentice Hall, 2014
- Christof Paar, Jan Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, Springer, 2010
- 김명환, 수리암호학개론, 2019
- 정민석, 암호수학, 경문사, 2017
- 최은미, 정수와 암호론, 북스힐, 2019
- 이민섭, 정수론과 암호론, 교우사, 2008
- Kevin S. McCurley, The Discrete Logarithm Problem, Proceedings of Symposia in Applied Mathematics, Vol 42, 1990