제 5 장 암호수학

공개키 암호알고리즘은 주로 정수를 이용한 수학연산을 통해 구현된다. 따라서 공개키 암호알고리즘의 원리를 이해하기 위해서는 정수론과 응용대수학에 대한 이해가 선행되어야 한다. 일반적으로 대칭 암호알고리즘에서는 XOR, 순환, 치환과 같은 비트 기반 연산을 이용하였지만 최근에 차세대 미국 대칭 암호알고리즘 표준으로 제정된 AES(Advanced Encryption Standard)에서는 이와 같은 비트 연산 대신에 수학연산을 사용하고 있다. 따라서 공개키 암호알고리즘 뿐만 아니라 대칭 암호알고리즘에서도 정수론, 응용대수학에 대한 이해가 필요하다.

5.1 정수론

5.1.1 표기법

이 장에서는 표 5.1과 같은 표기법을 사용하여 서술한다.

<표 5.1> 표기법

표기	의미
$\mathbb{Z}$	정수 집합
$\mathbb{Z}^+$	양의 정수 집합
$\mathbb{Z}_n$	n이 양의 정수일 때 0부터 $n-1$까지의 정수 집합
$\mathbb{Z}_n^*$	$\mathbb{Z}_n$의 원소 중 n과 서로 소(relatively prime, coprime)인 원소들의 집합
$\gcd(a,b)$	양의 정수 a와 b의 최대공약수
$\mathrm{lcm}(a,b)$	양의 정수 a와 b의 최소공배수
$a\|b$	정수 a는 정수 b의 약수
대문자	집합
소문자	원소

예를 들어 $\mathbb{Z}_{10} = \{0,1,2,3,4,5,6,7,8,9\}$이며, $\mathbb{Z}_{10}^* = \{1,3,7,9\}$이다.

5.1.2 기초

정수론에서 가장 기본이 되는 정리 중 하나가 다음의 나눗셈 정리이다. 어떤 정수를 또 다른 정수로 나누었을 때 그 몫과 나머지는 유일하다는 것이다.

정리 5.1 (나눗셈 정리) 정수 a와 양의 정수 b가 주어졌을 때 다음을 만족하는 유일한 정수 q와 r이 존재한다.

$$a = bq + r,\ 0 \le r < b$$

여가서 q를 몫(quotient), r를 나머지(remainder, residue)라 한다.

위 정리에서 나머지가 0일 때 b는 a의 약수(divisor)라 하며, c가 동시에 a와 b의 약수이면 c는 a와 b의 공약수(common divisor)라 한다. 공약수 중에 가장 큰 수를 최대공약수(gcd, Greatest Common Divisor)라 하며, 다음과 같이 정의된다.

정의 5.1 (최대공약수) 양의 정수 c가 a와 b의 최대공약수가 되기 위한 조건은 다음과 같다.

- 조건 1. (공약수) $c|a$, $c|b$
- 조건 2. (최대공약수) 모든 d에 대해 $d|a$이고 $d|b$이면 $d|c$이어야 한다.

Bezout는 최대공약수와 관련 다음과 같은 정리를 정의하고 있다.

정리 5.2 (Bezout의 identity) 동시에 0이 아닌 두 정수 a와 b에 대해서 $\gcd(a,b)=ax+by$를 만족하는 정수 x와 y가 존재한다.

이와 관련하여 $d=\gcd(a,b)$일 때, 어떤 정수 c가 $ax+by$로 표현되기 위한 필요충분조건은 $d|c$이다. 또 $\gcd(a,b)=1$이면 정수 a와 b를 서로 소라 한다. 최대공약수를 계산하는 가장 유명한 알고리즘은 유클리드(Euclid) 알고리즘이다. 이 알고리즘은 다음 정리를 이용한다.

정리 5.3 (유클리드 알고리즘) $a=bq+r$이면 $\gcd(a,b)=\gcd(b,r)$이다.

즉, 두 수의 최대공약수를 구할 때 가장 큰 수 대신에 다른 작은 수를 이용하여 계산할 수 있으며, 이것을 반복하여 보다 작은 수들을 이용하여 최대공약수를 구할 수 있다.

정수 중에 다른 수들과 구별되는 소수(prime)라는 수들이 있다. 소수는 1과 자신만을 약수로 가지는 수이다. 유클리드는 소수가 무수히 많이 존재한다는 것을 증명하였으며, 나눗셈 특성 때문에 n이 합성수(소수가 아닌 수)이면 n은 $\sqrt{n}$보다 작은 소인수를 가진다. 또 모든 정수는 소수를 이용하여 독특하게 표현할 수 있다. 이것과 관련된 정리는 다음과 같다.

정리 5.4 (정수론의 기본정리) 모든 정수 $n>1$은 소수의 곱으로 표현되며, 소수의 순서를 무시하면 그 표현은 유일하다.

$$n=p_1^{a_1}p_2^{a_2}...p_r^{a_r}$$

여기서 p_i는 소수이다.

5.1.3 합동식

일반 정수 연산을 암호알고리즘에서 사용하는 정수 값의 크기가 계속 커질 수 있다. 따라서 특정한 범위로 연산의 결과를 한정하고 싶으면 법(modular) 연산을 사용할 수 있으며, 법 연산과 관련된 것이 합동식이다. 합동은 어떤 기저가 되는 수에 국한된 것으로서, 두 수

가 합동(congruent)하다는 것은 다음과 같이 정의한다.

정의 5.2 (합동) 두 정수 a와 b가 법 n에서 합동이기 위해서는 $n|(a-b)$이어야 하며, 다음과 같이 표기한다.

$$a \equiv b \pmod{n}$$

예를 들어 2 ≡ 7 (mod 5)이다. 합동은 다르게 표현하면 두 수를 정해진 법으로 나누었을 때 나머지가 같다는 것을 의미한다.

법 $n \geq 1$에서 모든 정수는 $\mathbb{Z}_n$ 중 하나의 수를 표기할 수 있으며, $\mathbb{Z}_n$를 법 n의 최소 양의 잉여(least non-negative residue)라 한다. 다시 말하면 정수 $0 \leq b < n$에 대해 모든 양의 정수 a는 $a \equiv b \pmod{n}$를 만족하는 b가 존재하며, 이 때 b는 법 n에서 a의 최소 양의 잉여라 한다. 최소 양의 잉여 대신에 특정 법의 모든 원소를 최소 절대 잉여로 나타낼 수 있다. n이 홀수이면 $0, \pm 1, \pm 2, ..., \pm (n-1)/2$, n이 짝수이면 $0, \pm 1, \pm 2, ..., \pm (n-1)/2, n/2$가 최고 절대 잉여들이다. 예를 들어 5의 최소 양의 잉여 집합은 {0, 1, 2, 3, 4}이고, 5의 최소 절대 잉여 집합은 {−2, −1, 0, 1, 2}이다. 법 5에서 3은 −2와 같은 수이다. 이것은 $2+3 \equiv 2+(-2) \pmod{5}$식을 보면 쉽게 이해할 수 있다. 각 잉여류를 대표하는 정수를 정확하게 하나만 포함하는 정수의 집합을 법 n에 관한 완전잉여계(complete set of residues modulo n)라 한다. 하지만 보통 최소 양의 잉여계나 최소 절대 잉여계를 많이 사용한다. 최소 양의 잉여계는 다른 말로 표준 잉여계(standard residue system)이라 한다.

$ab \equiv 1 \pmod{n}$이면 b를 법 n에서 a의 곱셈에 대한 역원(multiplicative inverse)이라 한다. 예를 들어 법 5에서 2의 곱셈에 대한 역원 3이다. 정수 a가 법 n에서 곱셈에 대한 역원을 가지기 위한 필요충분조건은 $\gcd(a,n) = 1$이다. 법 연산을 할 때 곱셈에 대한 역원을 가지는 원소들이 매우 유용하게 사용된다. 따라서 법 n에서 역원을 가지는 잉여를 대표하는 정수를 정확하게 하나만 포함하는 정수의 집합을 기약잉여계라 한다. 예를 들어 법 10의 표준 기약잉여계는 {1,3,7,9}이다. 즉, 법 n의 표준 기약잉여계는 $\mathbb{Z}_n^*$로 표기된다.

합동식을 간소화하고 여러 합동식을 하나의 합동식으로 결합할 때 사용할 수 있는 원리는 다음과 같다.

- a, b, c와 $n > 0$이 정수일 때, $\gcd(c, n) = 1$이고 $ac \equiv bc \pmod{n}$이면 $a \equiv b \pmod{n}$이다.
- a와 b는 정수이고, $n_1, ..., n_k$는 양의 정수라 하자. 또한 $1 \leq i \neq j \leq k$에 대해 $\gcd(n_i, n_j) = 1$라 하자. 이 때 $a \equiv b \pmod{n_1}$, $a \equiv b \pmod{n_2}$, ..., $a \equiv b \pmod{n_k}$이면 $a \equiv b \pmod{n_1 n_2 ... n_k}$이다.

예를 들어 $6x \equiv 12 \pmod{5}$을 $2 \cdot 3 \cdot x \equiv 4 \cdot 3 \pmod{5}$로 다시 표현할 수 있다. 즉, $6x \equiv 12 \pmod{5}$은 $2x \equiv 4 \pmod{5}$로 바꿀 수 있다. 여기서 중요한 것은 양변에서 제거된 수가 법과 서로 소이어야 한다. $6x \equiv 12 \pmod{9}$이면 $2x \equiv 4 \pmod{9}$가 성립하지 않는다. 일차합동식은 이와 같은 방법으로 해를 찾을 수 있다.

연립합동식의 경우에는 다음과 같은 중국인 나머지 정리를 이용하여 해를 찾을 수 있다.

정리 5.4 (중국인의 나머지 정리) n_1, ..., n_k은 양의 정수라 하자. 단, $1 \leq i \neq j \leq k$에 대해

$\gcd(n_i, n_j) = 1$이다. 이 때 연립합동식

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\vdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

은 $M = n_1 n_2 \cdots n_k$을 법으로 유일한 해를 갖는다.
중국인 나머지 정리에 정의된 유일한 해는 다음과 같이 계산할 수 있다.

$$x_0 = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_k M_k y_k$$

여기서 $M_i = M/n_i$이고 y_i는 법 n_i에서 M_i의 곱셈에 대한 역원이다.

합동식들의 해를 구하기 위해서는 법 n에서 어떤 수의 곱셈에 대한 역원을 구해야 한다. 곱셈에 대한 역원은 확장 유클리드 알고리즘을 통해 구할 수 있다. 유클리드 알고리즘은 최대공약수를 구하는 알고리즘이지만 확장 유클리드 알고리즘은 최대공약수뿐만 아니라 곱셈에 대한 역원도 구할 수 있다. 물론 법 n에서 정수 a의 곱셈에 대한 역원을 구하기 위해서는 $\gcd(a,n) = 1$이어야 한다. 따라서 확장 유클리드 알고리즘을 이용할 경우 계산된 최대공약수는 1이 된다. 확장 유클리드 알고리즘은 표 5.2처럼 진행된다.

<표 5.2> 확장 유클리드 알고리즘

n	q	r_i	s_i	t_i
0		1759	1	0
1	3	550	0	1
2	5	109	1	-3
3	21	5	-5	16
4	1	4	106	-339
5		1	-111	355

즉, 법 $n(=1759)$에서 $a(=550)$의 곱셈에 대한 역원을 구하고 싶으면 다음과 같이 진행한다.

- 단계 1. $r_0 = n$, $r_1 = a$, $s_0 = 1$, $s_1 = 0$, $s_0 = 1$, $s_1 = 0$을 설정한다.
- 단계 2. $q_i = r_{i-2}/r_{i-1}$, $s_i = s_{i-2} - q_{i-1}s_{i-1}$, $t_i = t_{i-2} - q_{i-1}t_{i-1}$를 차례로 계산한다.
- 단계 3. $r_i = 1$이면 t_i가 법 n에서 정수 a의 곱셈에 대한 역원이 된다.

이와 같은 과정을 통해 법 n에서 정수 a의 곱셈에 대한 역원을 찾을 수 있는 이유는 다음과 같다. 유클리드 알고리즘에 의해 $n = aq + r$이면 $\gcd(n,a) = \gcd(a,r)$이다. 따라서 이 방법을 이용하여 최대공약수를 구하면 다음과 같다.

1759 = 550 × 3 + 109 gcd(1759, 550) =

550 = 109 × 5 + 5	gcd(550, 109) =
109 = 5 × 21 + 4	gcd(109, 5) =
5 = 4 × 1 + 1	gcd(5, 4) = 1

이것을 다시 거꾸로 다음과 같이 표현할 수 있다.

1 = 5 - 4 × 1
= 5 - (109 - 5 × 21)
= -109 + 5 × 22
= -109 + (550 - 109 × 5) × 22
= 550 × 22 - 109 × 111
= 550 × 22 - (1759 - 550 × 3) × 111
= 550 × 355 - 1759 × 111

즉, $550 \bullet 355 \equiv 1 \pmod{1579}$이다.

합동식 중에 암호기술에 많이 사용되는 특수한 합동식들이 있다. 그 중에 가장 핵심이 되는 합동식이 페르마의 작은 정리와 오일러 정리에 정의된 합동식이다. 페르마의 정리는 다음과 같다.

정리 5.5 (페르마의 작은 정리) p가 소수이고 $\gcd(a,p)=1$이면 $a^{p-1} \equiv 1 \pmod{p}$이다.

페르마의 정리는 소수가 법일 때 사용할 수 있는 합동식이다. 합성수가 법일 때에는 페르마의 정리와 유사한 오일러 정리를 사용할 수 있다. 오일러 정리에서는 오일러 함수(Euler function)를 사용하는데 이 함수는 다음과 같이 정의되며,

정의 5.3 (오일러 함수) n이 양의 정수일 때 오일러 함수는 n보다 작은 양의 정수 중 n가 서로 소인 정수의 개수로 정의된다. 즉, 다음과 같다.

$$\phi(n) = |\{a | 1 \le a < n \wedge \gcd(a,n) = 1\}|$$

오일리 정리는 다음과 같다.

정리 5.6 (오일러 정리) $n > 0$은 정수이고 $\gcd(a,n)=1$이면 $a^{\phi(n)} \equiv 1 \pmod{n}$이다.

오일러 정리를 이용하기 위해서는 오일러 함수를 계산하는 방법을 알아야 한다. 오일러 함수는 다음 세 가지 특성을 이용하여 계산한다.

- p가 소수이면 $\phi(p) = p-1$이다.
- $\gcd(m,n)=1$이면 $\phi(mn) = \phi(m)\phi(n)$이다.
- p가 소수이면 모든 i에 대해 $\phi(p^i) = p^i - p^{i-1}$이다.

모든 양의 정수는 정수론의 기본정리에 의해 소인수분해될 수 있으며, 합성수의 소인수분해를 구할 수 있으면 그것의 오일러 함수값은 위 특성들을 이용하여 쉽게 계산할 수 있다. 예를 들어 $\phi(100)=\phi(2^2 \cdot 5^2)=\phi(2^2)\phi(5^2)=(4-2)(25-5)=50$이 된다.

n이 두 개의 소수 p와 q의 곱일 때, 오일러 함수 대신에 보편 지수(universal exponent)라는 것을 사용할 수 있다. 즉, $n>0$은 정수이며, λ가 n의 보편 지수일 때 $\gcd(a,n)=1$이면 $a^{\lambda} \equiv 1 \pmod{n}$이다. n이 두 개의 소수 p와 q의 곱일 때 n의 보편 지수는 $\mathrm{lcm}(p-1,q-1)$이다. 보통 어떤 수의 보편 지수는 그 수의 오일러 함수 값보다 작거나 같다.

$\gcd(a,n)=1$인 정수 a를 법 n에서 거듭제곱을 하다 보면 결국에는 1이 된다. 이 때 $a^x \equiv 1 \pmod{n}$을 만족하는 가장 작은 양의 정수 x를 법 n에서 a의 위수(order)라 한다. 법 n에서 a의 위수를 $\mathrm{ord}_n a$로 표기한다. 위수와 관련된 특성은 다음과 같다.

- $\gcd(a,n)=1$이면 $\mathrm{ord}_n a | \phi(n)$이다.
- $\gcd(r,n)=1$이고 $\mathrm{ord}_n r=\phi(n)$이면 r를 법 n에서 원시근(primitive root)라 한다.
- $n>0$이 원시근을 가지기 위한 필요충분조건은 $n=2,\ 4,\ p^k,\ 2p^k$이다. 여기서 p는 소수이고, $k \geq 1$인 정수이다.

5.2 응용대수학

5.2.1 군

앞서 언급한 바와 같이 일반 수학연산을 사용하면 값은 무한히 증가할 수 있다. 하지만 공개키 암호알고리즘처럼 컴퓨터를 이용한 수학연산의 경우에는 그 결과가 특정한 범위 내로 한정되기를 원한다. 따라서 이와 같은 효과를 얻기 위해 보통 법 n에서의 연산을 사용한다. 법 n에서의 연산을 사용할 경우에는 피연산자는 전체 정수가 될 수 있지만 모든 정수는 0부터 $n-1$까지의 수로 매핑될 수 있으므로 전체 정수 집합보다는 $\mathbb{Z}_n$ 집합에 한정하여 연산을 수행한다고 생각할 수 있다. 이와 같은 개념을 보다 구체화하고, 이와 같은 연산의 특성을 생각하면 응용대수학에서 정의하고 있는 **군**(group), **환**(ring), **체**(field) 개념과 일치한다는 것을 알 수 있다.

군은 집합과 이항연산에 의해 정의되며, 이항연산이란 동일한 집합에서 피연산자를 두 개 취하며, 연산의 결과가 다시 그 집합으로 매핑되는 연산을 말한다. 즉, 이항연산의 피연산자 집합에 대해 닫혀있다.

정의 5.4 (군) 공집합이 아닌 집합 G 위에 다음 세 가지 조건을 만족하는 이항연산 $\circ$가 정의될 때, $\langle G, \circ \rangle$를 군이라 한다.

- **(결합법칙)** G의 임의의 원소 a, b, c에 대해 다음이 성립한다.
$$(a \circ b) \circ c = a \circ (b \circ c)$$
- **(항등원)** G의 모든 원소 a에 대해 다음이 성립하는 $e \in G$가 존재해야 한다.
$$a \circ e = e \circ a = a$$
이 때 e를 G의 단위원 또는 **항등원**(identity)이라 한다.

- (**역원**) G의 각 원소 a에 대해 다음이 성립하는 a^{-1}이 G에 존재해야 한다.

$$a \circ a^{-1} = a^{-1} \circ a = e$$

이 때 a^{-1}를 a의 **역원**(inverse)이라 한다.

따라서 군은 닫힘특성, 결합법칙, 항등원, 역원, 네 가지 특성을 가지고 있어야 한다. 군이 이 네 가지 특성에 추가적으로 교환법칙까지 성립하면 이 군을 아벨군(abelian group)이라 한다.

정의 5.5 (아벨군) $\langle G, \circ \rangle$가 군일 때 이 군의 이항연산 $\circ$가 다음을 만족하면 이 군을 아벨군 또는 가환군(commutative group)이라 한다.

- (**교환법칙**) G의 임의의 원소 a와 b에 대해 다음이 성립한다.

$$a \circ b = b \circ a$$

우리가 주로 사용하는 연산은 덧셈 또는 곱셈이다. 따라서 군의 이항연산이 덧셈 계열의 연산이면 **덧셈군**(additive group)이라 하고, 곱셈 계열의 연산일 경우에는 **곱셈군**(multiplicative group)이라 한다. 또 우리는 보통 집합의 크기가 한정되어 있는 유한집합에 정의된 군을 많이 사용한다. 이렇게 군의 집합이 유한하면 이 군을 **유한군**(finite group)이라 하고, 무한하면 무한군(infinite group)이라 한다. 이 때 군 집합의 원소의 개수 $|G|$를 군의 **위수**(order)라 하며, 군의 위수를 나타내기 위해 G_n과 같이 표기하기도 한다.

예를 들어 $\mathbb{Z}_7^* = \{1,2,3,4,5,6\}$은 법 7에서 곱셈 연산에 대해 다음에 의해 유한군을 형성함을 알 수 있다.

- 단힘 특성: 법 7에서 계산하므로 그 결과는 항상 $\mathbb{Z}_7^*$ 중 하나이다.
- 결합 법칙: 곱셈 계열의 연산은 보통 결합 법칙을 만족한다.
- 항등원: 1은 이 군의 항등원이다.
- 역원: (1,1), (2,4), (3,5), (6,6) 즉, 모든 원소는 항등원을 가진다.

보통 법 연산을 이용한 곱셈군 또는 덧셈군은 교환법칙이 성립한다. 따라서 이 군도아벨군이며, 이 군의 위수는 6이다.

군을 구성하는 집합의 부분집합이 원 군에 정의된 연산에 관하여 군을 형성하면 이 부분집합과 연산에 의해 정의되는 군을 원 군의 **부분군**(subgroup)이라 한다. 보다 정확한 정의는 다음과 같다.

정의 5.6 (부분군) $\langle G, \circ \rangle$가 군일 때 G의 부분집합 $H(\neq \varnothing)$가 이항연산 $\circ$에 관하여 군을 이루면 $\langle H, \circ \rangle$는 $\langle G, \circ \rangle$의 부분군이라 한다.

앞서 살펴본 $\mathbb{Z}_7^* = \{1,2,3,4,5,6\}$에서 $H = \{1,2,4\}$는 법 7에서 곱셈 연산에 대해 군을 이루므로 $\langle H, \times (\mathrm{mod}\, 7) \rangle$은 $\langle \mathbb{Z}_7^*, \times (\mathrm{mod}\, 7) \rangle$의 진부분군이다. 보통 군의 연산이 명확할 경우

에는 $\langle G, \circ \rangle$ 대신에 G로 군을 나타낸다.

군 G의 각 원소 a에 대해 부분군 $H = \{a^n | n \in \mathbb{Z}\}$는 a에 의해 생성된 **부분순환군**(cyclic subgroup)이라 하며, 이 군은 $\langle a \rangle$로 표기한다. 여기서 $a^0 = e$이며 a^n은 a를 피연산자로 군의 이항연산을 n번 수행한 것을 말하며, a^{-n}은 a^{-1}를 피연산자로 군의 이항연산을 n번 수행한 것을 말한다. 즉, $a^3 = a \circ a \circ a$이고, $a^{-2} = a^{-1} \circ a^{-1}$이다. $a^s, a^t \in H$에 대해 $a^s a^t = a^{s+t} \in H$이므로 이항연산에 대해 H는 닫혀 있고, $a^0 = e$이므로 항등원 H에 존재하며, 모든 원소의 역원이 존재한다. 따라서 H는 부분군이 된다. G가 유한군이면 H도 유한군이다. 그러므로 a를 피연산자로 군의 이항연산을 계속 수행하다 보면 궁극에 $a^n = e$가 되어야 한다.

$a \in G$에 대해 $G = \langle a \rangle$이면 G를 **순환군**(subgroup)이라 하며, 이 때 a를 이 군의 **생성자**(generator)라 한다. 어떤 원소가 군의 생성자가 되기 위해서는 그 원소의 위수가 군의 위수와 같아야 한다. 정수론에서 언급한 바와 같이 $n > 0$이 원시근을 가지기 위한 필요충분조건은 p가 소수이고, $k \geq 1$인 정수일 때, $n = 2, 4, p^k, 2p^k$이다. 따라서 $\mathbb{Z}_n^*$가 순환군이 되기 위해서는 $n = 2, 4, p^k, 2p^k$이어야 한다.

군 G의 원소 a의 위수가 n일 때 정수 m에 대해 a^m의 위수는 $n/\gcd(n,m)$이다. 즉, a^m들의 위수는 항수 a의 위수의 약수가 된다. 그러므로 다음 정리들이 성립한다는 것을 알 수 있다.

정리 5.7. 위수 n인 유한순환군 $G = \langle a \rangle$의 부분군은 모두 순환군이고, 그 위수는 n의 약수이다.

정리 5.8. 위수 n인 유한순환군 $G = \langle a \rangle$에 대해 d가 n의 양의 약수이면 위수가 d인 G의 부분군은 오직 하나 존재한다.

정리 5.9. 유한순환군 $G = \langle a \rangle$의 위수가 소수인 q이면 항등원 e를 제외한 G의 모든 원소는 G의 생성자가 된다.

유한순환군 $G = \langle a \rangle$의 위수가 n일 때 이 군의 모든 원소 a^i에 의해 생성되는 부분군의 위수는 $n/\gcd(n,i)$이다. 따라서 이들 부분군의 위수는 원 군의 위수의 약수임을 쉽게 알 수 있다. 또한 이 때 n이 소수 p이면 $p/\gcd(p,i) = p$이므로 항등원을 제외한 모든 원소는 원 군의 생성자가 된다.

공개키 암호알고리즘에서 많이 사용되는 군은 $\mathbb{Z}_n$와 $\mathbb{Z}_n^*$이다. 이들은 다음과 같은 특성을 가지고 있다. $\mathbb{Z}_n$은 법 n에 관한 덧셈 연산에 의해 덧셈순환군을 형성하며 그 위수는 n이다, $\mathbb{Z}_n^*$는 법 n에 관한 곱셈 연산에 의해 곱셈군을 형성하며, 이 군의 위수는 $\phi(n)$이다. 하지만 모든 $\mathbb{Z}_n^*$는 순환군이 아니다. 순환군이면 생성자가 존재하며, 생성자를 연산의 기저로 사용하면 연산의 결과를 예측할 수 있는 확률은 군의 크기에 의해 결정된다. 그런데 앞서 언급한 바와 같이 $\mathbb{Z}_n^*$가 생성자를 가지기 위한 필요충분조건은 $n = 2, 4, p^k, 2p^k$이므로

$n = 2, 4, p^k, 2p^k$이어야 $\mathbb{Z}_n^*$가 순환군이 된다. 따라서 공개키 암호알고리즘에서 가장 많이 사용하는 군은 $\mathbb{Z}_p^*$이다. 이 군은 위수가 $p-1$이며 순환군이다. 또 $d|p-1$이면 위수가 d인 $\mathbb{Z}_p^*$의 부분군 G_d가 존재하며, 이 군은 또한 순환군이다. 특히 d가 소수이면 $\mathbb{Z}_p^*$의 부분군 G_d에서 1를 제외한 모든 원소는 G_d의 생성자가 된다. 즉, 이 군의 어떤 원소를 사용하여 연산을 하더라도 그 연산의 결과를 예측할 수 있는 확률은 군의 크기에 의해 결정된다.

5.2.2 환

군은 하나의 연산에 대해서 정의되는 대수적 구조이다. 하지만 하나의 연산이 아니라 두 개의 연산에 대해 정의되는 대수적 구조가 필요할 수 있다. 환의 정의는 다음과 같다.

정의 5.7 (환) 집합 $R(\neq \varnothing)$ 위에 이항연산 덧셈 +와 곱셉 •이 정의되어 있고, 또 다음이 성립하면 $\langle R,+,\bullet\rangle$를 환이라 한다.

- **A.** $\langle R,+\rangle$는 아벨군이다.
  - **A.1** (결합법칙): $(a+b)+c = a+(b+c)$
  - **A.2** (교환법칙): $a+b = b+a$
  - **A.3** (영원): 모든 원소 $a \in R$에 대해 $a+0 = 0+a = a$가 성립하는 $0 \in R$이 존재한다.
  - **A.4** (역원): 각 원소 $a \in R$에 대해 $a+(-a) = (-a)+a = 0$이 성립하는 $-a \in R$이 존재한다.
- **M.1** (결합법칙): $(a\bullet b)\bullet c = a\bullet(b\bullet c)$
- **D.** (분배법칙): $a\bullet(b+c) = (a\bullet b)+(a\bullet c)$, $(a+b)\bullet c = (a\bullet c)+(b\bullet c)$

환은 덧셈과 곱셈 연산을 모두 고려하는 대수적 구조이지만 정의에서도 알 수 있듯이 덧셈에 대해서는 군을 형성하지만 곱셈 연산에 대해서는 군을 형성하지 않아도 된다. 위 조건에서 다음과 같이 곱셈에 대해 교환법칙이 성립하면 **가환환**(commutative ring)이라 하며, 가환환이 아닌 환을 **비가환환**(non-commutative ring)이라 한다.

- **M.2** (결합법칙): $(a\bullet b) = (b\bullet a)$

또 환 R이 다음 조건을 추가적으로 만족하면 이 환을 단위원 1을 가진 환이라 한다.

- **M.3** (단위원): 모든 원소 $a \in R$에 대해 $a\bullet 1 = 1\bullet a = a$가 성립하는 $1 \in R$이 존재한다.

M.3 조건까지 만족하면 환 R은 덧셈과 곱셈에 대한 항등원을 모두 가진다. 이를 구분하기 위해 덧셈에 대한 항등원 0은 영원이라 하고, 곱셈에 대한 항등원 1은 단위원이라고 한다. 또 영원을 제외한 모든 원소가 곱셈에 대한 역원을 가지면 환 R은 덧셈과 곱셈에 대해 모두 군을 형성하게 되며, 이와 같은 환을 나눗셈환 또는 **체**라 한다. 보다 정확히 말하면 가환환인 나눗셈환을 체라 하고, 비가환환인 나눗셈환을 **사체**(skew field)라 한다.

- **M.4** (역원): 각 원소 $a \in R$, $a \neq 0$에 대해 $a\bullet a^{-1} = a^{-1}\bullet a = 1$이 성립하는 $a^{-1} \in R$이 존

재한다.

환 R이 단위원 1을 가진 가환환으로서 다음 조건을 추가적으로 만족하면 R을 정역(integral domain)이라 한다.

- **M.4** (역원): $a \bullet b = 0$이면 $a = 0$ 또는 $b = 0$이다.

p가 소수일 때 $\mathbb{Z}_p$는 위수가 p인 체가 되며, 이 체를 $GF(p)$ 또는 F_p로 표기하고, 갈로아 체라고 한다. 또 유한체의 위수는 반드시 소수의 거듭제곱승 형태로 표현된다.

5.2.3 다항식

다항식(polynomial)을 이용하여 환을 구성할 수도 있다. 특히 다항식의 계수가 항상 특정 환 R에 속하면 이 다항식을 환 R위의 다항식이라 한다. 보다 정확한 정의는 다음과 같다.

정의 5.7 (환 R위의 다항식) R를 단위원 1을 가진 가환환이라 하고, x를 부정원이라 할 때, 다음과 같은 형태의 형식적인 무한합을 환 R위의 (x에 관한) 다항식이라 한다.

$$f(x) = a_0 + a_1 x + \cdots + a_n x^n + \cdots \quad (a_i \in R)$$

단, 유한개를 제외하고는 모든 i에 대해 $a_i = 0$이어야 한다. 여기서 a_0, a_1, $\cdots$, a_n 등을 다항식 $f(x)$의 **계수**(coefficient)라고 하며, a_0, $a_1 x$, $\cdots$, $a_n x^n$을 다항식$f(x)$의 **항**(term)이라 한다. 특히, a_0을 상수항(constant term)이라 한다.

다항식 $f(x) = a_0 + a_1 x + \cdots + a_n x^n$에서 a_0을 제외한 모든 계수가 0이면 $f(x)$를 상수다항식(constant polynomial)이라 하고, 특히 모든 계수가 0인 $f(x)$를 영다항식(zero polynomial)이라 한다. 만약 $i > n$에 대해 모든 $a_i = 0$이고 $a_n \neq 0$이면 a_n을 $f(x)$의 최고차 항의 계수(leading coefficient)라 하며, 이 다항식의 차수(degree) $\deg f(x)$를 n이라 한다. 특히, 이 때 $a_n = 1$이면 $f(x)$를 **모닉 다항식**이라 한다. 우리는 주로 모닉 다항식을 많이 사용한다.

다항식도 정수와 유사하게 사칙연산을 할 수 있다. 하지만 계수들이 환 또는 체에 속하는 환 위의 다항식을 이용한 사칙연산은 우리가 알고 있는 일반 다항식 사칙연산과 다르다.

예를 들어 $f(x) = x^3 + x^2 + 1$이고 $g(x) = x^2 + x + 1$일 때 이 두 다항식을 이용한 사칙연산의 결과는 다음과 같다.

- 덧셈: $f(x) + g(x) = x^3 + 2x^2 + x + 2$
- 뺄셈: $f(x) - g(x) = x^3 - x$
- 곱셈: $f(x) + g(x) = x^5 + 2x^4 + x^3 + 2x^2 + 2x + 1$
- 나눗셈: $f(x)/g(x)$에서 몫은 x이고 나머지는 $-x + 1$이다.

하지만 위 두 다항식이 $GF(2)$ 위의 체라 가정하자. $GF(2)$의 원소는 0과 1밖에 없다. 따라서 $GF(2)$에서 사칙연산의 결과는 다음과 같다.

- 덧셈: $f(x)+g(x)=x^3+x$
- 뺄셈: $f(x)-g(x)=x^3+x$
- 곱셈: $f(x)+g(x)=x^5+x^3+1$
- 나눗셈: $f(x)/g(x)$에서 몫은 x이고 나머지는 $x+1$이다.

이와 같은 방식으로 사칙연산을 할 경우에는 다항식들이 정수와 매우 유사한 성질을 가지게 된다. R이 가환환이면 다항식환 $R[x]$는 R 위의 모든 다항식들로 구성되는 환을 말하며, 특히 체 F위의 다항식환 $F[x]$는 다음과 같은 성질을 가지는 정역이 된다.

- (1) 임의의 $f(x), g(x) \in F[x]$에 대해 다음이 성립한다.
$$\deg f(x)g(x) = \deg f(x) + \deg g(x)$$
- (2) (나눗셈 알고리즘) 임의의 $f(x), g(x) \in F[x]$, $f(x) \neq 0$에 대해
$$g(x) = f(x)q(x) + r(x),\ r(x) = 0 \text{ 또는 } 0 \leq \deg r(x) < \deg f(x)$$
인 다항식 $q(x), r(x) \in F[x]$가 유일하게 존재한다.

체 F 위의 다항식 $p(x) \in F[x]$, $\deg p(x) \geq 1$에 대해 $f(x)|p(x)$이면 적당한 $a \in F^*$에 대하여 $f(x)=a$ 또는 $f(x)=ap(x)$일 때, $p(x)$를 $F[x]$의 **기약다항식**(irreducible polynomial)이라 한다. 반대로 체 F 위의 다항식 $p(x) \in F[x]$, $\deg p(x) \geq 1$가 기약다항식이 아닐 때, $p(x)$를 $F[x]$의 **가약다항식**(reducible polynomial)이라 한다. 따라서 모든 일차다항식은 모두 기약다항식이다. 정수와 비교하여 보면 기약다항식은 정수에서 소수와 같은 것이고, 가약다항식은 합성수와 같은 것이다. 따라서 정수가 소수로 유일하게 인수분해를 할 수 있듯이 다항식들로 모닉 기약다항식으로 유일하게 인수분해를 할 수 있다.

정리 5.9. (유일인수분해 정리) 체 F 위의 다항식환 $F[x]$에서 임의의 다항식 $f(x) \in F[x]$, $\deg f(x) \geq 1$는 다음과 같은 형태로 인수분해 된다.

$$f(x) = ap_1(x)^{e_1}p_2(x)^{e_2}\cdots p_k(x)^{e_k}$$

여기서 $a \in F^*$이고, e_i는 양의 정수이며, $p_i(x)$는 모닉 기약다항식이다. 위의 등식에서 기약다항식의 곱의 순서를 무시하면 이와 같은 인수분해는 단 한가지뿐이다.

다항식 환에서도 기약다항식을 이용하여 $\mathbb{Z}_p$와 유사한 체를 만들 수 있다. 보통 컴퓨터를 이용하여 연산을 할 경우에는 피연산자나 연산의 결과를 n비트 워드로 제한하고 싶은 경우가 많다. 특히, 암호기술에서는 더욱 그렇다. 예를 들어 3비트 데이터를 조작하고 싶을 경우에는 $\mathbb{Z}_8$의 사용을 고려해 볼 수 있다. 하지만 $\mathbb{Z}_8$은 체가 아니며 그림 5.1처럼 각 수의 등장 비율이 다르다. 하지만 기약다항식을 이용한 GF(2^3)은 체이며, 각 수의 등장비율이 같다. 따라서 최근에 이와 같은 기약다항식을 이용한 체를 활용하여 비트 조작을 한다.

F는 체이고 $f(x) \in F[x]$는 영다항식이 아닌 기약다항식일 때, $F[x]$에 있는 모든 다항식들을 $f(x)$로 나누었을 때 나머지들의 집합을 $F[x]_f$라 하면, $F[x]_f$는 체가 된다. 특히, F의

위수가 소수 p인 체이고, $f(x)$가 $F[x]$에서 n차인 기약다항식이라 하면 체 $F[x]_f$의 위수는 p^n이 된다. $p=2$이면 연산의 피연산자와 결과를 n비트 워드로 다음과 같이 제한할 수 있다.

- $f(x)=x^3+x+1$일 때 $F_2[x]_f$의 각 원소는 다음과 같이 정수로 표현할 수 있다.
  - $f(x)$의 차수는 3차이므로 $F_2[x]_f$의 위수는 $2^3=8$이며, $F_2[x]_f$의 모든 원소 $g(x)=b_2x^2+b_1x+b_0$와 같은 형태가 된다. 여기서 $b_0, b_1, b_2 \in F_2$이다.
  - 따라서 $g(x)=b_2x^2+b_1x+b_0$는 표 5.3과 같이 $b_2b_1b_0$으로 표현할 수 있다.

<표 5.3> $F_2[x]_{x^3+x+1}$의 원소에 대한 정수 표현

0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	3	1	7	5
3	0	3	6	5	7	4	1	2
4	0	4	3	7	6	2	5	1
5	0	5	1	4	2	7	3	6
6	0	6	7	1	5	3	2	4
7	0	7	5	2	1	6	4	3

<그림 5.1> $\mathbb{Z}_8$과 $F_2[x]_f, f(x)=x^3+x+1$를 이용한 곱셈표

$F_2[x]_f$에서 덧셈과 곱셈은 컴퓨터를 이용하여도 효율적으로 계산할 수 있다. 특히 두 다항식의 덧셈은 XOR 연산을 이용하면 된다.

예5.1) $F_2[x]_f$, $f(x)=x^8+x^4+x^3+x+1$의 체에서 다음 두 다항식의 더한 결과는?

$$a(x)=x^6+x^4+x^2+x+1,\ b(x)=x^7+x+1$$

답) $a(x)+b(x)=x^7+x^6+x^4+x^2$이다. $a(x)$에 대한 비트표현은 01010111이고, $b(x)$에 대한 비트표현은 10000011이다. 따라서 $a(x)+b(x)=(01010111)\oplus(10000011)=(11010100)$이다.

곱셈은 덧셈보다 조금 더 복잡하지만 여전히 효율적으로 계산할 수 있다. 예5.1의 체의 속한 $a(x)=b_7x^7+b_6x^6+b_5x^5+b_4x^4+b_3x^3+b_2x^2+b_1x+b_0$에 x를 곱하는 것을 생각하여 보자. $x \bullet a(x)$는 다음과 같다.

$$x \bullet a(x)=b_7x^8+b_6x^7+b_5x^6+b_4x^5+b_3x^4+b_2x^3+b_1x^2+b_0x$$

따라서 $b_7=0$이면 $x \bullet a(x)=b_6x^7+b_5x^6+b_4x^5+b_3x^4+b_2x^3+b_1x^2+b_0x$이며 $b_7=1$이면

$f(x)$로 나눈 나머지가 $x \bullet a(x)$가 된다. $x^8 \, mod \, f(x) = x^4 + x^3 + x + 1$이므로 $x \bullet a(x)$는 다음과 같다.

$$x \bullet a(x) = b_7 x^8 + b_6 x^7 + b_5 x^6 + b_4 x^5 + b_3 x^4 + b_2 x^3 + b_1 x^2 + b_0 x + x^4 + x^3 + x + 1$$

따라서 $x \bullet a(x)$는 다음과 같이 구할 수 있다.

$$x \bullet a(x) = \begin{cases} b_6 b_5 b_4 b_3 b_2 b_1 0 & b_7 = 0 \\ (b_6 b_5 b_4 b_3 b_2 b_1 0) \oplus (00011011) & b_7 \neq 0 \end{cases}$$

위 원리를 이용하면 $a(x) \bullet b(x)$를 오른쪽 이동 연산과 XOR 연산만을 이용하여 계산할 수 있다.

예5.2) $F_2[x]_f$, $f(x) = x^8 + x^4 + x^3 + x + 1$의 체에서 다음 두 다항식을 곱한 결과는?

$$a(x) = x^6 + x^4 + x^2 + x + 1, \; b(x) = x^7 + x + 1$$

답) $a(x) \bullet b(x) = x^7 \bullet a(x) + x \bullet a(x) + a(x)$와 같다. 이 때 $a(x)$의 7차항의 계수는 0이므로 $x \bullet a(x)$는 $a(x)$를 오른쪽 하나 이동한 것과 같다. 또 $x^7 \bullet a(x)$는 다항식 x를 7번 곱한 것과 같으므로 다음과 같이 계산할 수 있다.

- $x \bullet a(x) = 10101110$
- $x \bullet x \bullet a(x) = 01011100 \oplus 00011011 = 01000111$
- $x \bullet x \bullet x \bullet a(x) = 10001110$
- ...
- $x^7 \bullet a(x) = 00111000$

따라서 $a(x) \bullet b(x) = 00111000 \oplus 10101110 \oplus 01010111 = 11000001 = x^7 + x^6 + 1$이다.

연습문제

1. p가 소수이면 모든 $i \geq 1$에 대해 $\phi(p^i) = p^i - p^{i-1}$임을 증명하시오.
2. 다음을 구하시오.
   (1) $\phi(77)$
   (2) $\phi(231)$
3. 오일러 정리를 이용하여 $2^{98} \bmod 19$를 구하시오. (힌트. $2^{14} \equiv 16 \equiv -3 \pmod{19}$)
4. 확장 유크리드 알고리즘을 이용하여 법 4231에서 1233의 곱셈에 대한 역원을 구하시오.
5. p와 q가 서로 다른 소수이고 $n = pq$라 하자. 그러면 $\lambda = \text{lcm}\,(p-1, q-1)$에 대해 $\gcd(a,n) = 1$이면 $a^\lambda \equiv 1 \pmod{n}$임을 증명하시오.
6. $\mathbb{Z}_{13}^*$의 부분군 중 위수가 6인 부분군을 구하시오. (힌트. 2는 $\mathbb{Z}_{13}^*$의 생성자이다)
7. $\mathbb{Z}_3$ 위에서 다항식 $x^2 + 2x + 2$가 기약인지 가약다항식인지 판별하시오.
8. $m(x) = x^4 + x + 1$을 사용할 때 $F_2[x]_{m(x)}$에서 $0xD \times 0xB$를 구하시오.
9. $m(x) = x^4 + x + 1$을 사용할 때 $F_2[x]_{m(x)}$에서 $0xB$의 곱셈에 대한 역원을 구하시오.