

# RSA시스템의 기반 수학과 이해

EUnS

2019년 6월 8일

## 제 1 장

$d, m, n$ ,이 어떤 정수일 때, 다음이 성립한다.

1.  $d$ 가  $m$ 과  $n$ 의 공약수일때,  $m + n$ 도  $d$ 의 배수이다.
2.  $d$ 가  $m$ 과  $n$ 의 공약수일때,  $m - n$ 도  $d$ 의 배수이다.

이에 대한 증명은 간단합니다.

$m = dq_1, n = dq_2$  ( $q_1, q_2$ 는 어떤 정수)

$m + n = d(q_1 + q_2), m - n = d(q_1 - q_2)$

참고로  $d$ 가  $n$ 의 약수(인수)일 때  $d \mid n$ 으로 표시합니다.

$m$ 과  $n$ 의 최대공약수는  $\gcd(m, n)$ 이라고 합니다.

$r$ 이  $a$ 를  $b$ 로 나눈 나머지라면  $r = a \bmod b$ 입니다.

이를써서 위 명제를 다시 적으면  $d \mid n, d \mid m \longrightarrow d \mid (m + n), d \mid (m - n)$

$a, b, z$ 를 양의 정수라 하면, 다음이 성립한다.

$$ab \bmod z = [(a \bmod z)(b \bmod z)] \bmod z$$

$w = ab \bmod z, x = a \bmod z, y = b \bmod z$ 라 하자. 다음이 성립하는  $q_1$ 이 존재한다.

$$ab = q_1 z + w \iff w = ab - q_1 z$$

마찬가지로 다음을 만족시키는  $q_2$ 와  $q_3$ 가 존재한다.

$$\begin{aligned} a &= q_2 z + x, b = q_3 z + y \\ w &= ab - q_1 z = (q_2 z + x)(q_3 z + y) - q_1 z \\ &= (q_2 q_3 z + q_2 y + q_3 x - q_1)z + xy \\ &= qz + xy \end{aligned}$$

여기서  $q = q_2 q_3 z + q_2 y + q_3 x - q_1$ 이므로

$$xy = -qz + w$$

즉  $w$ 는  $xy$ 를  $z$ 로 나눌 때의 나머지이다. 그러므로  $w = xy \bmod z$ 가 되고 이는 다음과 같이 변환된다.

$$ab \bmod z = [(a \bmod z)(b \bmod z)] \bmod z$$

이는 큰수를 인수분해해서 작은값으로 나눠서 큰수를 다루는 부담을 덜어주지만 지수승에 대해서도 응용이 가능하다. 이를 이용해서  $a^{29} \bmod z$ 를 계산하는 절차를 예시로 들어보겠다.  $a^{29}$ 는 다음과 같은 순서로 계산한다.

$$a, a^5 = a \cdot a^4, a^{13} = a^5 \cdot a^8, a^{29} = a^{13} \cdot a^{16}$$

$a^{29} \bmod z$ 는 다음과 같은 순서로 계산한다.

$$a \bmod z, a^5 \bmod z, a^{13} \bmod z, a^{29} \bmod z$$

$$\begin{aligned}
a^2 \bmod z &= [(a \bmod z)(a \bmod z)] \bmod z \\
a^4 \bmod z &= [(a^2 \bmod z)(a^2 \bmod z)] \bmod z \\
a^8 \bmod z &= [(a^4 \bmod z)(a^4 \bmod z)] \bmod z \\
a^{16} \bmod z &= [(a^8 \bmod z)(a^8 \bmod z)] \bmod z \\
a^5 \bmod z &= [(a \bmod z)(a^4 \bmod z)] \bmod z \\
a^{13} \bmod z &= [(a^5 \bmod z)(a^8 \bmod z)] \bmod z \\
a^{29} \bmod z &= [(a^{13} \bmod z)(a^{16} \bmod z)] \bmod z
\end{aligned}$$

## 제 2 장

### 1 유클리드 호제법(Euclidean algorithm)

$a$ 가 음이 아닌 정수이고,  $b$ 가 양의 정수이며  $r = a \bmod b$ 이면 다음이 성립한다.

$$\gcd(a, b) = \gcd(b, r)$$

수식이 익숙하지 않은 분을 위해 풀어서 설명하자면,  $a$ 가 음이 아닌 정수이고,  $b$ 가 양의 정수이며,  $r$ 이  $a$ 를  $b$ 로 나눈 나머지라면  $a$ 와  $b$ 의 최대공약수는  $b$ 와  $r$ 의 최대공약수와 같다.

뭐 어쨌든, 증명을 하자면  $a = bq + r$  ( $0 \leq r < b, q$ 는 어떤 정수)인데,  $c$ 를  $a$ 와  $b$ 의 공약수라 하면,  $c$ 는  $bq$ 의 약수인 것은 자명합니다.  $a$  또한  $c$ 의 약수이므로  $c$ 는  $a - bq (= r)$ 의 약수입니다. 따라서  $c$ 는  $b$ 와  $r$ 의 공약수입니다. 반대로  $c'$ 가  $b$ 와  $r$ 의 공약수이면,  $c'$ 는  $bq + r (= a)$ 의 약수가 되고 따라서  $a$ 와  $b$ 의 공약수가 됩니다. 따라서  $a$ 와  $b$ 의 공약수 집합이  $b$ 와  $r$ 의 공약수 집합과 같으므로  $\gcd(a, b) = \gcd(b, r)$ 이 성립합니다.

유클리드 알고리즘의 의의는 나머지 연산만을 이용해서 뱅뱅이 돌리면 어떻게 됐든지 간에 최대공약수를 기계적으로 구할수있다는 것에 있다.  $\gcd(a, b) = \gcd(b, r)$ 에서  $b, r$ 을 새로운  $a, b$ 로서 값을 넣어서 연속적으로 계산을 하면 언젠가  $b$ 가 0이 되는 순간이 오는데, 이때  $a$ 가 처음  $a, b$ 의 최대 공약수가 되는것이다.

## 2 확장된 유클리드 알고리즘(Extended Euclidean algorithm)

확장된 유클리드 알고리즘은 다음의 명제에 대해서  $s$ 와  $t$ 를 효율적으로 구하는 방법에 대한 것이다.

$a$ 와  $b$ 가 음이 아니고 동시에 0이 아닌 정수라 하면 다음을 만족시키는 정수  $s$ 와  $t$ 가 존재한다.

$$\gcd(a, b) = s \cdot a + t \cdot b$$

<sup>a</sup>선형 디오판토스 방정식 그중에서도 베주 항등식이라고도 한다.

### 2.1 베주의 항등식

$ax + by = \gcd(x, y)$ 인  $a, b$ 가 존재한다.

일반해는  $\gcd(x, y)$ 의 배수이다.

증명

집합  $S = \{m | m = ax + by, x \in \mathbf{Z}, y \in \mathbf{Z}\}$ 를 생각해 보면, 이 집합  $S$ 는  $S \subset \mathbf{Z}$ ,  $S \subset \emptyset$  ( $x, y$ 를 원소로 가짐을 알 수 있다.) 자연수의 정렬성으로부터 최소가 되는 원소  $d$ 가 존재한다.  $\alpha \in S \Rightarrow \alpha = qd + r (0 \leq r < d \because \text{나머지정리})$  만약  $d \nmid \alpha$  일때,  $r > 0$ ,  $r = \alpha - qd$ , ( $\alpha, d \in S$ )  $\alpha = a_1x + b_1y, d = a_2x + b_2y$ 라 하면.  $r = (a_1 - a_2q)x + (b_1 - qb_2)y \in S$   $0 < r < d$ 인  $r$ 에 대해  $d$ 가 최소라는 가정이 모순이다.  $\therefore r = 0, d | \alpha (\forall \alpha \in S)$   $d | x, d | y \cdots d$ 는  $x, y$ 의 공약수  $\gcd(x, y) = k$ 라 할때,  $d = akx'' + bky'' = k(ax'' + by'')$   $k | d$ 에서  $k = d$

### 2.2 활용

이미 증명되어있는 유클리드 알고리즘의 흐름을 통해서 예시로 이해 해보자.

$a = 273$ ,  $b = 110$ 으로 하는  $\gcd(273, 110)$ 을 구해봅시다.

$$r = 273 \bmod 110 = 53 \cdots 1$$

$a = 110, b = 53$ 으로 지정

$$r = 110 \bmod 53 = 4 \cdots 2$$

$a = 53, b = 4$ 로 지정

$$r = 53 \bmod 4 = 1 \cdots 3$$

$a = 4, b = 1$ 로 지정

$$r = 4 \bmod 1 = 0 \cdots 4$$

$r = 0$ 이므로  $\gcd(273, 110)$ 은 최대공약수로 1을 가진다. 여기서 4 식으로 되돌아가면 이는 다음과 같이 쓸 수 있다.

$$1 = 53 - 4 \cdot 13$$

계속 역순으로 뒤집어 올라가자 3

$$4 = 110 - 53 \cdot 2$$

이를 처음의 식에 대입하면

$$1 = 53 - (110 - 53 \cdot 2) \cdot 13 = 27 \cdot 53 - 13 \cdot 110$$

2

$$53 = 273 - 110 \cdot 2$$

이 식을 다시 대입하면

$$1 = 27 \cdot 53 - 13 \cdot 110 = 27 \cdot (273 - 110 \cdot 2) - 13 \cdot 110 = 27 \cdot 273 - 67 \cdot 110$$

따라서  $s = 27, t = -67$ 로서 성립하는 값을 찾았다.

### 3 나머지 연산에서 곱셈에 대한 역원 (modular multiplicative inverse) <sup>1</sup>

$\gcd(n, \phi) = 1$ 인 두 정수  $n > 0, \phi > 1$ 가 있다고 하자.<sup>2</sup>  $n \cdot s \bmod \phi = 1$ 을 만족시키는  $s$ 를  $n \bmod \phi$ 의 역원(inverse) 이라고 한다.

이  $s$ 를 효율적으로 구하는 방법에 대해서 논해보자

$\gcd(n, \phi) = 1$ 은 확장된 유클리드 알고리즘을 이용하여  $s' \cdot n + t \cdot \phi = 1$ 이 되는  $s'$ 과  $t'$ 을 구할수있다.  $n \cdot s' = -t' \cdot \phi + 1$ 이 되고  $\phi > 1$ 이므로 1이 나머지가 된다.  $n \cdot s' \bmod \phi = 1$ 에서  $s = s' \bmod \phi$ 라 하면  $0 \leq s < \phi$ 가 되며  $s \neq 0$ 이다.

위 식을 변형하면  $s' = q \cdot \phi + s$  가되며 이를 만족하는 정수 q가 존재한다.

따라서

<sup>1</sup>역원: a와 연산자에 대해 연산결과가 항등원(= 1)이 되는 유일한 원소 b를 a의 역원이라한다.

<sup>2</sup>한 마디로 n과  $\phi$ 는 서로소이다.

$$n \cdot s = ns' - \phi nq = -t'\phi + 1 - \phi nq = \phi(-t' - nq) + 1$$

따라서  $n \cdot s \bmod \phi = 1$ 이 된다.

## 제 3 장

### 4 오일러의 $\varphi$ 함수(Euler's phi (totient) function)

양의 정수  $n$ 에 대해서  $\varphi(n)$ 은 1부터  $n$ 까지의 양의 정수 중에  $n$ 과 서로소인 것의 개수를 나타내는 함수이다.

$\varphi(n)$ 은 다음의 성질이 있다.

- 소수  $p$ 에 대해서  $\varphi(p) = p - 1$
- $m, n$ 이 서로소인 정수일 때, 다음이 성립한다.

$$\varphi(mn) = \varphi(m)\varphi(n)$$

성질이 몇 개 더 있지만 RSA에서 필요한것만을 다루기위해서 생략하였다. 첫번째 성질은 어찌보면 당연하다  $p$ 는 소수이니 자기 자신을 제외한 모든 수와 서로 소이다 (여기서 1도 세야한다.)

두번째 성질은 두수의 곱  $mn$ 은 각각  $m$ 에대해서 나뉘지는 수가  $n$ 개이고  $n$ 에 대해서 나뉘지는 수가  $m$ 개 이며  $mn$ 으로 나뉘지는 수가 한 개이므로  $mn - \frac{mn}{m} - \frac{mn}{n} + \frac{mn}{mn} = mn - m - n + 1 = (m - 1)(n - 1) = \varphi(m)\varphi(n)$ 가 된다.

### 5 오일러 정리(Euler's theorem)<sup>3</sup>

임의의 정수  $a$ 와  $n$ 이 서로소일 때, 다음이 성립한다.

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

#### 5.1 증명

정수  $n$ 에 대해서 1부터  $n$ 까지의 양의 정수 중에  $n$ 과 서로소인 것의 집합을 생각해보자. 그러면 이는 집합

<sup>3</sup>페르마의 소정리는 오일러 정리에서의 특수한 경우이다.



$$A = \{r_1, r_2, r_3, \dots, r_{\varphi(n)}\}^4$$

으로 나타낼 수 있다. 이 집합은  $A$ 라하고 이 각 원소에  $n$ 과 서로소인  $a$ 를 곱한 집합을  $B$ 집합이라 하자.

$$B = \{ar_1, ar_2, ar_3, \dots, ar_{\varphi(n)}\}$$

확실한건  $B$ 에 있는 모든 원소는  $n$ 과 서로소인 것이다. 그럼  $B$ 집합의 각 원소를  $\text{mod } n$ 에 대해 계산한 것을 생각해보자. 이는 각 원소의 나머지가  $a$ 를 곱하기전 값과 같은지는 모르지만  $\varphi(n)$ 개에 대해서 각각 일대일대응이 가능 한다는것을 알수있다. <sup>5</sup> 따라서  $A$ 의 모든 원소를 곱한 값에  $\text{mod } n$ 을 한것과  $B$ 의 모든 원소를 곱한 값에  $\text{mod } n$ 을 한 값은 같다.

$$ar_1 \cdot ar_2 \cdot ar_3 \cdots ar_{\varphi(n)} \equiv r_1 \cdot r_2 \cdot r_3 \cdots r_{\varphi(n)} \pmod{n}$$

$$a^{\varphi(n)} \text{ mod } n = 1$$

---

<sup>4</sup>이러한 집합을 기약잉여계라고 부른다. 또한 집합  $A$ 의 원소의 갯수는  $\varphi(n)$ 이다.

<sup>5</sup>실제 증명은 귀류법을 통해서 증명할수있다.  $ar_i \equiv ar_j \pmod{n}$  인  $1 \leq i < j \leq \varphi(n)$  이 존재한다고 가정해보자.

## 제 4 장

### 6 RSA 공개 키 암호 시스템<sup>6</sup> (RSA public-key cryptosystem)

이 알고리즘은 보안 기법중 하나로 가장 흔한 예시로서는 공인인증서가 있다.

$$A \longrightarrow B$$

$A$ 가  $B$ 에게 숫자를 하나 보낸다고 생각 해보자.  $A$ 에게는 공개키가 필요하며  $B$ 에게는 개인키가 있어야한다. 공개키는 누가 가져도 상관없는 키이며 개인키는 절대로 노출되어서는 안되는 키이다.

$A$ 는  $B$ 에게  $a$ 를 보낼때 공개키를 이용하여  $a$ 를  $c$ 로 암호화 하여 보내며  $B$ 는  $c$ 를 공개키와 개인키를 이용하여  $a$ 로 복호화하여 읽는 방식이다.

#### 6.1 공개키, 암호키 생성

두 개의 소수  $p, q$ 를 선택하여  $z = pq$ 를 계산한다.<sup>7</sup> 그 후  $\phi = (p-1)(q-1)$ 을 계산하고  $\gcd(n, \phi) = 1$ 인 정수  $n$ 을 선택한다. 그후  $z$ 와  $n$ 을 공개한다.  $ns \bmod \phi = 1$ 이고  $0 < s < \phi$ 를 만족시키는  $s$ 를 생성하여  $s$ 를 개인키로 사용한다.<sup>8</sup>

#### 6.2 단계

$A$ 가  $B$ 에게 정수  $a(0 \leq a \leq z-1)$ 를 보내기 위해서  $A$ 는  $c = a^n \bmod z$ 를 계산하여  $c$ 를 보낸다.<sup>9</sup>  $B$ 는  $c^s \bmod z$ 를 계산하면 이 값이  $a$ 이다.

---

<sup>6</sup>로널드 라이베스트(Ron Rivest), 아디 샤미르(Adi Shamir), 레너드 애들먼(Leonard Adleman)이 세명의 이름 앞글자를 따서 지었다.

<sup>7</sup>그 후  $p, q$ 는 버린다. 가지고 있어봤자 개인키가 풀리는 취약점이 될수가있다.

<sup>8</sup> $s$ 는 위에서 언급한 나머지 연산에서 곱셈에 대한 역원을 구하는 방법으로 효율적으로 구할수 있다.

<sup>9</sup> $c$ 를 효율적으로 구하는 방법 또한 위에서 다루었다.

### 6.3 복호화 과정

$$ns \bmod \phi = 1 \iff ns = b\varphi(n) + 1 (b \text{는 어떤 상수})$$

$$\begin{aligned} c^s \bmod z &= (a^n \bmod z)^s \bmod z = (a^n)^s \bmod z \\ &= a^{ns} \bmod z = a^{b\varphi(n)+1} \bmod z = (a^{\varphi(n)} \bmod z)^b a \bmod z = a^{10} \end{aligned}$$

### 6.4 이게 과연 안전한가?

기본적으로 RSA를 풀기위한 시간복잡도는 np문제와 연결되는데 결론만 말하면 안전하다는 것이다.

이를 구하기위한 해결방법은 결과적으로 소인수분해와 직결되는데 소인수분해를 다항시간내에하는 알고리즘은 개발되지 않았다.

---

<sup>10</sup>오일러정리 사용

참고 저서: 이산 수학(Discrete Mathematics) (저자:Richard Johnsonbaugh)  
암호학과 네트워크 보안(cryptography and network security) (저자: Behrouz  
A. Forouzan)  
<https://blog.naver.com/mindo1103/221234421987>  
<https://blog.naver.com/mindo1103/221234421626>