공개키 암호 알고리즘(RSA)

유튜브 주소

한성대학교
HANSUNG UNIVERSITY

CryptoCraft LAB

RSA 개요

공개키 알고리즘을 위한 핵심 정수론

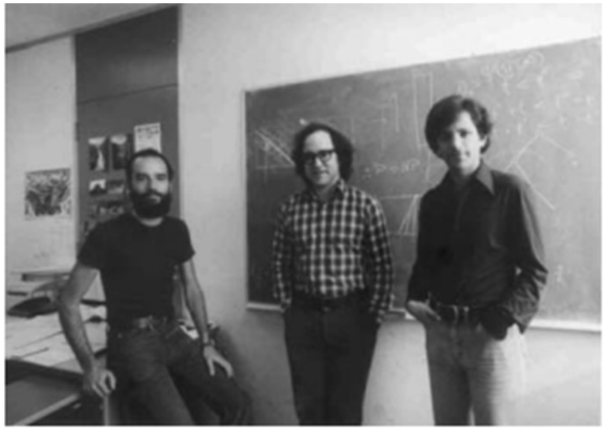
오일러의 파이 함수 및 페르마 소정리

RSA의 암호화 및 복호화

RSA의 개요

RSA

- 1977년 Ronald Rivest, Adi Shamir, Leonard Adleman등 3명의 수학자에 의해 개발.
- 공개키 및 비대칭키 알고리즘으로 두 개의 큰 소수(보통 140자리 이상)를 이용.
- 암호화 및 복화화는 정수환 $Z_n = \{0, 1, \ldots, n-1\}$에서 실행되며 모듈러 연산이 중요한 역할.
- RSA가 갖는 전사서명 기능은 인증을 요구하는 전자 상거래에 등에 광범위하게 활용중.



공개키 알고리즘을 위한 정수론

중요한 공개키 알고리즘

비대칭 알고리즘은 **일방향 함수(One-way Function)**에 기반을 둠.

Definition 6.1.1 One-Way Function $f(\cdot)$
- $y = f(x)$의 계산은 수월함(Computationally **Easy**).
- $x = f^{-1}(y)$의 계산은 실행 불가능함(Computationally **Infeasible**).

공개키 알고리즘 계열

- **정수의 인수분해**

  ex) RSA

- **이산 대수 :** 정수 a, y, m에 대하여, $a^x \equiv y \bmod m$인 x를 찾는 문제

  ex) Diffie-Hellman Key Exchange, DSA 등

- **타원 곡선 :** 이산대수의 일반화

  ex) Elliptic Curve Diffie-Hellman키 교환, Elliptic Curve Digital Signature Algorithm(ECDSA) 등

공개키 알고리즘을 위한 핵심 정수론

유클리드 호제법(Euclidean Algorithm)

- $\gcd(r_0, r_1)$: 두 양의 정수 r_0와 r_1 의 최대 공약수

  Example

  $r_0 = 84$ = 2x2x3x7, $r_1 = 30$ = 2x3x5에 대해, gcd(84,30) = 2x3 = 6

- $\gcd(r_0, r_1) = \gcd(r_0 - r_1, r_1) = \gcd(r_0 - 2r_1, r_1) = \cdots = \gcd(r_0 - mr_1, r_1)$ for $(r_0 - mr_1) > 0$ 이 성립함.

  -> 두 수의 gcd를 찾는 문제는 보다 작은 두 수의 gcd를 찾는 문제로 축소됨.

  -> 이를 반복하여 처리하여, 최종적으로 $\gcd(r_l, 0) = r_l$ 이 얻고자 하는 해답이 됨.

  Example

  $r_0 = 27$, r_1 =21의 gcd(27,21)는? **-> gcd(27,21)** = gcd(1x21+6,21) = gcd(21,6)

  gcd(21,6) = gcd(3x6+3,6) = gcd(6,3)

  gcd(6,3) = gcd(2x3+0,3) **= gcd(3,0) = 3**

공개키 알고리즘을 위한 핵심 정수론

확장 유클리즈 호제법(EEA, Extended Euclideam Algorithm)

EEA는 $r_1 \bmod r_0$의 모듈러 역원을 계산함.

EEA는 정수 s, t 및 gcd를 계산함: $\gcd(r_0, r_1) = s \times r_0 + \underbrace{t \times r_1}$

$$= s \cdot r_0 + t \cdot r_1 = 1$$

$$= s \cdot 0 + t \cdot r_1 \equiv 1 \bmod r_0$$

$$r_1 \cdot t \equiv 1 \bmod r_0$$

모듈러 역원의 정의와 비교하면, t 는 $r_1 \bmod r_0$ 의 역원임. 역원이 존재하기 위해서는 $\gcd(r_0, r_1) = 1$임.

Example

12 mod 67의 모듈러 역원(즉, $12^{-1} \bmod 67$)를 구하시오.
$\gcd(67, 12) = 1$.
EEA를 적용하면, $\gcd(67, 12) = 1 = s \cdot 67 + t \cdot 12$.
$r_0 = 67$, $r_1 = 12$로부터 출발.
표로부터 $-5 \cdot 67 + 28 \cdot 12 = 1$을 얻음.
즉, 28이 12 mod 67의 역원임($12^{-1} \equiv 28 \bmod 67$).
$28 \cdot 12 = 336 \equiv 1 \bmod 67$로부터 검증됨.

i	q_{i-1}	r_i	s_i	t_i
2	5	7	1	-5
3	1	5	-1	6
4	1	2	2	-11
5	2	1	-5	**28**

오일러의 파이 함수 및 페르마 소정리

정수환 $\boldsymbol{Z_m} = \{\mathbf{0}, \mathbf{1}, \dots, \boldsymbol{m} - \mathbf{1}\}$에 대해 집합 내에서 m과 서로소인 정수가 얼마나 많은가?

=> Answer: 오일러의 파이 함수$\boldsymbol{\Phi(m)}$

Example

$m = 6$, $\boldsymbol{Z_6} = \{0,1,2,3,4,5\}$ 및 $m = 5$, $\boldsymbol{Z_5} = \{0,1,2,3,4\}$에 대해

$\gcd(0,6) = 6$	$\gcd(0,5) = 5$
$\gcd(1,6) = 1 \;\star$	$\gcd(1,5) = 1 \;\star$
$\gcd(2,6) = 2$	$\gcd(2,5) = 1 \;\star$
$\gcd(3,6) = 3$	$\gcd(3,5) = 1 \;\star$
$\gcd(4,6) = 2$	$\gcd(4,5) = 1 \;\star$
$\gcd(5,6) = 1 \;\star$	

따라서, $\Phi(6) = 2$, $\Phi(5) = 4$.

집합 내에서 모든 정수에 대해 gcd를 계산하는 것이 큰 수 m에 대해서 매우 느림.

오일러의 파이 함수 및 페르마 소정리

정수 m이 m = $p_1^{e_1} \cdot p_2^{e_2} \cdot \dots \cdot p_n^{e_n}$ 의 형태로 표현 가능하고, p_i는 소수, e_i는 **양의 정수일 때**

$\boldsymbol{\Phi(m) = \prod_{i=1}^{n}(p_i^{e_i} - p_i^{e_i-1})}$**의 형태로 구해질 수 있음.**

특히, $e_i = 1$ 인 경우 파이 함수 계산이 간단해짐.

예를 들어, m = p x q인 경우, $\Phi(m) = (p-1) \cdot (q-1)$.

Example

$m = 240 = 16 \cdot 15 = 2^4 \cdot 3 \cdot 5 = p_1^{e_1} \cdot p_2^{e_2} \cdot p_3^{e_3}$로 부터

$\Phi(240) = (2^4 - 2^3) \cdot (3^1 - 3^0) \cdot (5^1 - 5^0) = 8 \cdot 2 \cdot 4 = 64.$

Important

임의의 정수 m의 인수분해가 알려지면, $\boldsymbol{\Phi(m)}$ 를 구하는 것은 계산적으로 쉬움.

그렇지 않은 경우, 큰 수 m에 대한 $\boldsymbol{\Phi(m)}$을 구하는 것은 계산적으로 실행 불가능함.

오일러의 파이 함수 및 페르마 소정리

페르마 소정리

정수 a와 소수 p에 대해, $a^p \equiv a \bmod p$

위의 정의로부터 $a^{p-1} \equiv 1 \bmod p$임을 알 수 있음.

모듈러 역원을 구하는데 활용 가능.

즉, $a^{-1} \equiv a^{p-2} \bmod p$ 로 부터 a의 모듈러 역원 a^{p-2}를 구할 수 있음.

Example

p = 7, a = 2에 대해 a의 역원은 $a^{p-2} = 2^5 = 32 \equiv 4 \bmod 7$.

검증: 2 x 4 = 8 $\equiv 1 \bmod 7$ → 페르마 소정리는 모듈러 소수 p에 대해서만 성립함을 주의.

오일러의 파이 함수 및 페르마 소정리

오일러 정리

gcd(a, m) = 1인 정수 a, m에 대해(즉, 서로소인 a,m에 대해) $a^{\Phi(m)} \equiv 1 \bmod m$

Example

$m = 12,\ a = 5$에 대해
오일러 파이 함수를 계산함.

$$\Phi(12) = \Phi(2^2 \cdot 3) = (2^2 - 2^1) \cdot (3^1 - 3^0) = 4$$

이로부터, 오일러 정리를 검증하면 $5^{\Phi(12)} = 5^4 = 625 \equiv 1 \bmod 12$임을 알 수 있음.

p가 소수인 경우, $\Phi(p) = (p^1 - p^0) = p - 1$이고, $a^{\Phi(p)} = a^{p-1} \equiv 1 \bmod p$임.

=> **페르마 소정리는 오일러 정리의 특별한 경우임.**

RSA의 암호화 및 복호화

RSA 암호화 및 복호화는 정수환 $Z_n = \{0, 1, \ldots, n-1\}$에서 실행되며, 모듈러 연산이 중요한 역할을 함

RSA 암호화(Encryption)

- 공개키(Public-Key) $(n, e) = k_{pub}$와 평문(Plaintext) x에 대해
- 암호화 함수: $y = e_{k_{pub}}(x) \equiv x^e \bmod n$, with $x, y \in \boldsymbol{Z}_n$.

RSA 복호화(Decryption)

- 개인키(Private-Key) $d = k_{pr}$과 암호문(Ciphertext) y에 대해
- 복호화 함수: $x = d_{k_{pr}}(y) \equiv y^d \bmod n$, with $x, y \in \boldsymbol{Z}_n$.

실제로 x, y, n and d는 매우 큰 정수임

RSA의 안전성은 공개키(n, e)에 대해 d를 유도하기 어렵다는 사실에 기반을 둠.

RSA의 암호화 및 복호화

RSA 키 생성 알고리즘

출력(Output): Public-Key $k_{pub} = (n, e)$, Private-Key $k_{pr} = d$

1. 두 개의 큰 소수 p와 q를 선택함.
2. $n = p \cdot q$를 계산함.
3. $\Phi(n) = (p-1) \cdot (q-1)$을 계산함.
4. $\gcd(e, \Phi(n)) = 1$인 공개 지수 $e \in \{1, 2, \ldots, \Phi(n) - 1\}$을 선택함.
5. $d \cdot e \equiv 1 \bmod \Phi(n)$인 개인키 d를 계산함.

주목할 점

1. 두 개의 큰 소수 p와 q를 선택하는 것이 쉽지 않음
2. $\gcd(e, \Phi(n)) = 1$ 은 e의 역원이 존재하고, 그로부터 개인키 d가 항상 존재하는 것을 보장.
3. 확장 유클리드 호제법(EEA)를 이용하여 d와 e를 계산함.

RSA의 암호화 및 복호화

주목할 점

4. 실제로 먼저 $0 < e < \Phi(n)$이 되도록 공개 지수 e를 선택함. 단, $\gcd(e, \Phi(n)) = 1$를 만족해야 함.
5. n과 e를 가지는 EEA를 적용하여 $\gcd(\Phi(n), e) = s \cdot \Phi(n) + t \cdot e$의 관계식을 얻음.
6. $\gcd(e, \Phi(n)) = 1$이면 e는 유효한 공개 지수임을 알 수 있으며, 또한 EEA를 통해 계산된 t가 e의 역원이라는 사실도 알 수 있음. 즉, $d \equiv e^{-1} \equiv t \bmod \Phi(n)$.
7. EEA의 계수 s는 계산될 필요가 없음.

Example

Alice | **Bob**

message $x = 4$

Bob:
1. choose $p = 3$ and $q = 11$
2. $n = p \cdot q = 33$
3. $\Phi(n) = (3-1)(11-1) = 20$
4. choose $e = 3$
5. $d \equiv e^{-1} \equiv 7 \bmod 20$

$$\xleftarrow{k_{pub}=(33,3)}$$

Alice: $y = x^e \equiv 4^3 \equiv 31 \bmod 33$

$$\xrightarrow{y=31}$$

Bob: $y^d = 31^7 \equiv 4 = x \bmod 33$

$d \cdot e = 7 \cdot 3 \equiv 1 \bmod \Phi(n)$라는 조건을 만족함을 알 수 있음.

Q & A