개요

목차

- 용어
- Kerckhoffs's principle
- 보안 강도
- 보안의 목표
- 암호 수학 기초 소개
  - 약수, 최대공약수, 서로소, 모듈로 연산, 법 n에 대해 합동, $\boldsymbol{Z_n}$, $\boldsymbol{Z_n^*}$, $\boldsymbol{Z_p}$, $\boldsymbol{Z_p^*}$, 역원
- 현대 암호 알고리즘
- 암호방법 예
- 암호학적 해쉬함수
- 대칭키 vs. 비대칭키 암호기술
- 스트림 암호 기술
- 암호기술과 수학

용어

- Cryptology (암호학)
  - Cryptography (암호기술)
    - 평문(plaintext)을 암호화(encryption, encipherment)하여 암호문(ciphertext)을 생성하거나 암호문을 평문으로 복호화(decryption, decipherment)하는 기술
    - Symmetric-key cryptography (대칭키 암호기술)
      - Block cipher (블록 암호 기술) → DES, 3DES, AES
      - Stream cipher (스트림 암호 기술) → RC4, ChaCha20
        - 2015년 TLS에서의 RC4 사용 금함 (IETF RFC 7465)
        - 2018년 TLS 1.3에 ChaCha20 포함 (IETF RFC 8446)
    - Asymmetric-key cryptography (비대칭키 암호기술)
      - RSA, Diffie-Hellman Key Exchange, ECC
    - Cryptographic hash function (암호학적 해쉬함수)
      - MD4, MD5, SHA-1, SHA-2 (SHA-256, SHA-512, …), SHA-3
  - Cryptanalysis (암호분석)
    - 암호 알고리즘, 암호 체계 분석, 암호 해독

Kerckhoffs's principle

Reference: https://ko.wikipedia.org/wiki/케르크호프스의_원리

- Kerckhoffs's principle (커크호프스 원칙)
  - 암호시스템의 동작 방식이 공격자에게 알려진다 하더라도 문제가 되지 않도록 암호시스템을 설계해야 함
- 암호 기술 안전성 확인
  - 어떤 암호체계의 안전성을 확인하는 유일한 방법은 해당 암호 방법을 공개하고 암호분석이 진행되도록 하는 것임. 비밀로 해야 할 것은 암호시스템이 아니라 키(key)임 (Paar & Pelzl, 2010)

보안 강도

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

Reference: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf

보안강도 (비트)	안전 사용 기간	대칭키 알고리즘	공개키 알고리즘			해쉬 함수	
			이산대수 (DH, DSA 등)	인수분해 RSA	타원곡선 (ECDH, ECDSA 등)	전자서명, 충돌저항 필요 응용	메시지인증, 키유도함수, 난수생성
≤ 80	안전하지 않음	2TDEA	L=1024, N=160	RSA-1024	160	SHA-1	
112	2030년까지	3TDEA	L=2048, N=224	RSA-2048	224	SHA-224	
128	2031년 이후 계속	AES-128	L=3072, N=256	RSA-3072	256	SHA-256	SHA-1
192		AES-192	L=7680, N=384	RSA-7680	384	SHA-384	SHA-224
256		AES-256	L=15360, N=512	RSA-15360	512	SHA-512	SHA-256

- L → 공개키 크기, N → 개인키 크기
- 2TDEA → 2-key Triple Data Encryption Algorithm (K_1, K_2, K_3= K_1)
- 3TDEA → 3-key Triple Data Encryption Algorithm (K_1, K_2, K_3)
- 크기 p의 타원곡선군에 대한 best known Discrete Log 알고리즘 시간복잡도 $O(\sqrt{p})$ → 위수가 2^{160}인 군은 보안 강도 80비트 (Paar & Pelzl, 2010)

- $\log_{10}(2^{1024}) \approx 309$
- 77을 소인수분해해 보시오
- 300자리 이상 십진수(예: https://en.wikipedia.org/wiki/RSA_numbers)를 소인수분해해 보시오

보안의 목표

- 정보 및 정보시스템을 위한 주요 보안 목표 3가지(CIA triad)
  - 기밀성(**C**onfidentiality)
    - 접근에 대한 인가된 제약(authorized restrictions) 유지
      - 예) 권한 없는 접근에 대해 정보의 비밀성 유지 필요
  - 무결성(**I**ntegrity)
    - 부적절 접근(improper modification, destruction, etc.)으로부터의 보호
      - 예) 부적절한 정보 변경이 있었는지 확인 가능해야 함
  - 가용성(**A**vailability)
    - 적절한 시점에 신뢰할 수 있는 접근 및 사용을 보장

Reference: FIPS 199, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.199.pdf

암호 수학 기초: 나눗셈, 가분성

- $\mathbf{22} = 4 \times \mathbf{5} + 2$
  - 22를 5로 나누면 몫은 4이고 나머지는 2이다
    - 22를 5로 나누면 몫(quotient)은 4이고 나머지(remainder)는 2이다
- 나눗셈정리
  - 임의의 정수 a, 양의 정수 b에 대해 $\boldsymbol{a} = \boldsymbol{q} \times \boldsymbol{b} + \boldsymbol{r}\ \ (0 \le r < b)$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 → $\mathbf{5} \mid \mathbf{10}$
  - $\mathbf{11} = 2 \times \mathbf{5} + 1$ → 11은 5로 나누어지지 않음 → 표기법 → $\mathbf{5} \nmid \mathbf{11}$

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 $\mathbf{5 \mid 10}$
    - $\mathbf{0} = \boldsymbol{a} \times 0$
      - 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 $\mathbf{1} \mid \boldsymbol{a}$
    - **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, $\mathbf{gcd}(\boldsymbol{a}, \mathbf{0}) = \boldsymbol{a}$
      - $\gcd(5, 0) = 5$
    - 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

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

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

암호 수학 기초: $\boldsymbol{Z_n}$, $\boldsymbol{Z_n^*}$, $\boldsymbol{Z_p}$, $\boldsymbol{Z_p^*}$

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

- $\boldsymbol{Z_n}$
  - Set of all least residues modulo n
  - $Z_n = \{0, 1, 2, \dots, n-1\}$
    - n으로 나누었을 때 얻어지는 나머지들의 집합
  - $\boldsymbol{Z_n}$에서 n이 소수 p인 경우 $\boldsymbol{Z_p}$로 표기
    - $Z_p = \{0, 1, 2, \dots, p-1\}$
- $\boldsymbol{Z_n^*}$
  - $Z_n^* = \{x \mid 1 \le x \le n,\ \gcd(x, n) = 1\}$
    - 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
    - $|Z_n^*| = \phi(n)$
  - $\boldsymbol{Z_n^*}$에서 n이 소수 p인 경우 $\boldsymbol{Z_p^*}$로 표기
    - $Z_p^* = \{1, 2, \dots, p-1\}$

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

$Z_6^* = \{1, 5\}$		
	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
  - 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	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

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

×	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

현대 암호 알고리즘

- 대칭키(symmetric-key) 암호 기술
  - 송수신측이 동일한 비밀키로 암호화 및 복호화
  - 구분
    - 블록 암호
      - 대치(substitution) 기반 암호 방법
        - 평문 내 심볼을 다른 심볼로 변경하는 방식으로 암호화
      - 전치(transposition) 기반 암호 방법
        - 평문 내 심볼들의 위치를 변경하는 방식으로 암호화
    - 스트림 암호 기술
- 비대칭키(asymmetric-key) 암호 (공개키(public-key) 암호) 기술
  - 송수신측이 서로 다른 키들(공개키, 개인키)을 이용하여 암호화 및 복호화
  - 구분
    - 정수 인수분해 문제 기반
      - RSA
    - 이산대수 문제 기반
      - Diffie-Hellman Key Exchange
    - 타원곡선 기반
      - Elliptic Curve Diffie-Hellman(ECDH) Key Exchange

대치 기반 암호 방법 예: Caesar cipher

평문 심볼	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
암호문 심볼	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

평문 → 암호문

YES → 암호화 → BHV

대치 기반 암호 방법 예: Additive cipher

Additive cipher (shift cipher, Caesar cipher)

- 평문에 키를 더하여 암호문 생성하고, 암호문에서 키를 감하여 평문 복원

평문 심볼	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
대응 수	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

키 k → [평문 내 각 문자에 대응하는 수 P를 아래 수식의 C로 변환 $C = (P + k)\ mod\ 26$]

평문 → [평문 내 각 문자에 대응하는 수 P를 아래 수식의 C로 변환 $C = (P + k)\ mod\ 26$] → 암호문 - - - - → 암호문 → [암호문 내 각 문자에 대응하는 수 C를 아래 수식의 P로 변환 $P = (C - k)\ mod\ 26$] → 평문

키 k → [암호문 내 각 문자에 대응하는 수 C를 아래 수식의 P로 변환 $P = (C - k)\ mod\ 26$]

- 평문, 암호문, 키의 표현을 위해 $Z_{26} = \{0, 1, 2, \dots, 25\}$의 원소 사용
- 가능한 키의 개수 26개
- 법 26에 대해 $-k$ 는 키 k의 덧셈의 역원

Key=5

평문: YES (240418) → 암호화 → 암호문: DJX (030923)

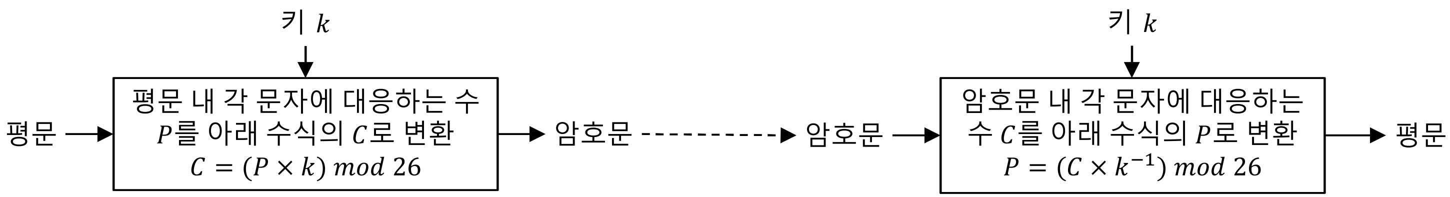
- 법 26에 대한 키 5의 덧셈의 역원 -5는 21 ($(5 + 21)\ mod\ 26 = 0$)
- Y → 24 → 암호화 → $(24 + 5)\ mod\ 26 = 03$ → D
- D → 03 → 복호화 → $(03 + (-5))\ mod\ 26 = (03 + 21)\ mod\ 26 = 24$ → Y

대치 기반 암호 방법 예: Multiplicative cipher

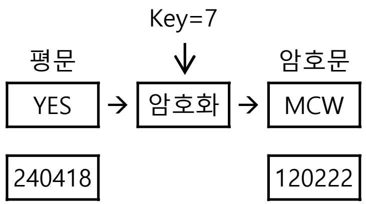
Multiplicative cipher

- 평문에 키를 곱하여 암호문 생성하고, 암호문을 키로 나누어 평문을 복원

평문 심볼	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
대응 수	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25



- 평문, 암호문 표현을 위해 $Z_{26} = \{0, 1, 2, \ldots, 25\}$의 원소 사용
- 키 표현을 위해 $Z_{26}^{*} = \{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$의 12개 원소 사용
- 암호화와 복호화가 서로의 역이 되기 위해 곱셈의 역원이 존재하는 수만을 키로 사용 가능
- 법 26에 대해 k^{-1}은 키 k의 곱셈의 역원



- 법 26에 대한 키 7의 곱셈의 역원 7^{-1}는 15 ($(7 \times 15)\ mod\ 26 = 1$)
- Y → 24 → 암호화 → $(24 \times 7)\ mod\ 26 = 12$ → M
- M → 12 → 복호화 →$(12 \times 7^{-1})\ mod\ 26 = (12 \times 15)\ mod\ 26 = 24$ → Y

대치 기반 암호 방법 예: Affine cipher

Affine cipher

- Additive cipher와 multiplicative cipher의 결합

평문 심볼	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
대응 수	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

키 k_1 키 k_2

평문 → [평문 내 각 문자에 대응하는 수 P를 아래 수식의 C로 변환 $C = ((P \times k_1) + k_2)\ mod\ 26$] → 암호문 ----------→ 암호문 → [암호문 내 각 문자에 대응하는 수 C를 아래 수식의 P로 변환 $P = ((C - k_2) \times k_1^{-1})\ mod\ 26$] → 평문

키 k_1 키 k_2

- 평문, 암호문 표현을 위해 $Z_{26} = \{0, 1, 2, \ldots, 25\}$의 원소 사용
- 키 k_1 표현을 위해 $Z_{26}^* = \{1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 15,\ 17,\ 19,\ 21,\ 23,\ 25\}$의 12개 원소 사용
- 키 k_2 표현을 위해 $Z_{26} = \{0, 1, 2, \ldots, 25\}$의 원소 사용
- 암호화와 복호화가 서로의 역이 되기 위해 곱셈의 역원이 존재하는 수만을 키로 사용 가능
- 법 26에 대해 $-k_2$는 키 k_2의 덧셈의 역원, k_1^{-1}은 키 k_1의 곱셈의 역원
- Affine cipher는 $k_1 = 1$ 인 경우 additive cipher가 되고, $k_2 = 0$ 인 경우 multiplicative cipher가 됨

$k_1 = 7$
$k_2 = 5$

평문 [YES] [240418] → [암호화] → 암호문 [RHB] [170701]

- 법 26에 대한 키 7의 곱셈의 역원 7^{-1}는 15 $((7 \times 15)\ mod\ 26 = 1)$
- 법 26에 대한 키 5의 덧셈의 역원 -5는 21 $((5 + 21)\ mod\ 26 = 0)$
- Y → 24 → 암호화 → $(24 \times 7 + 5)\ mod\ 26 = 17$ → R
- R → 17 → 복호화 →$((17 - 5) \times 7^{-1})\ mod\ 26 = ((17 + 21) \times 15)\ mod\ 26 = 24$ → Y

대치 기반 암호 방법 예: Monoalphabetic cipher

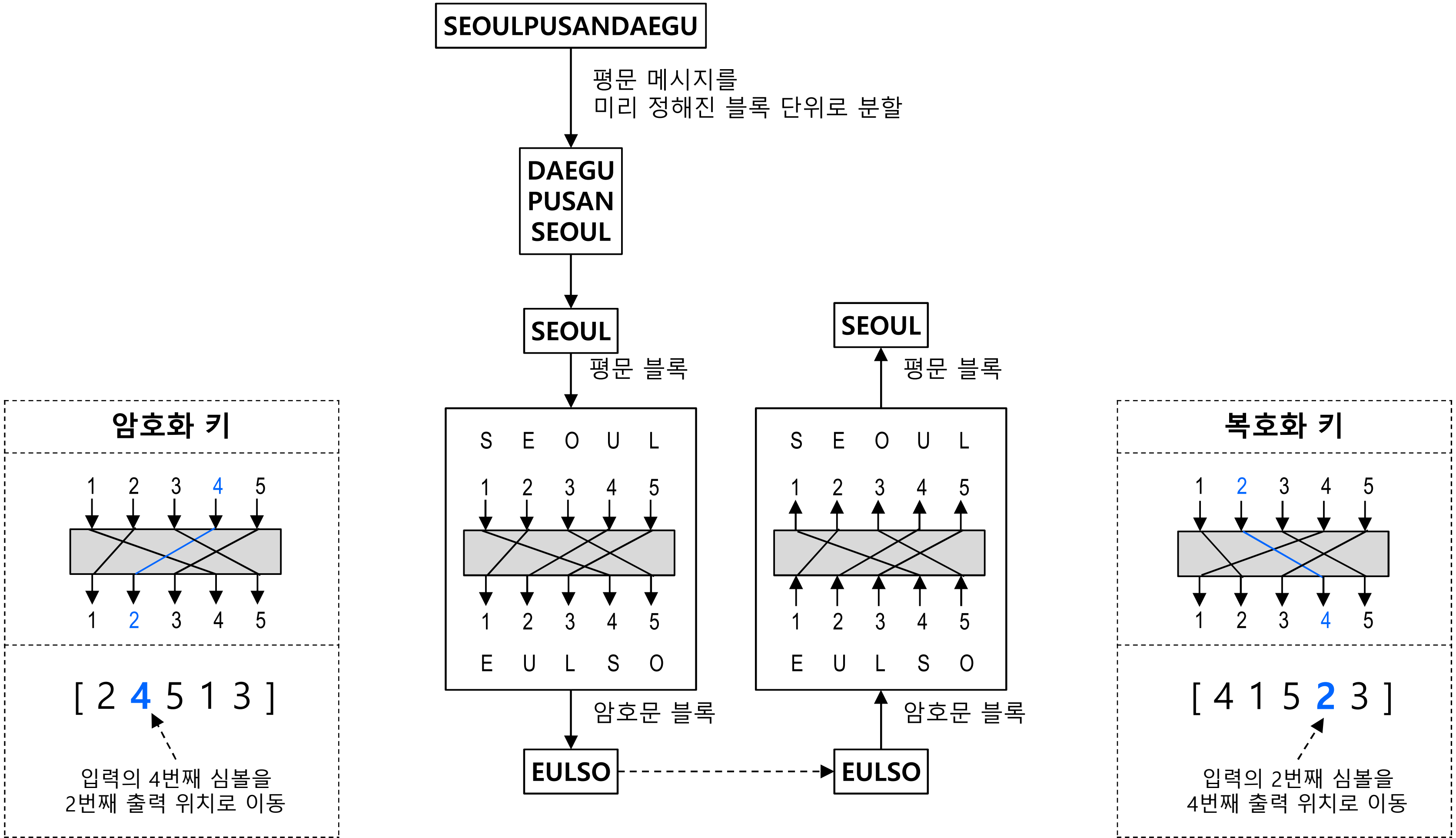
Monoalphabetic substitution cipher

- Additive, multiplicative, affine cipher들은 키 공간의 크기가 작아 키 전수조사 공격에 취약
- 평문 내 출현 가능한 각 심볼에 대해 대응하는 암호문 심볼의 변환 테이블을 키로 사용하면 알파벳 대문자 집합의 경우 키 공간의 크기가 $26! \, (\approx 4 \times 10^{26} \approx 2^{88})$이 되어 전수조사 공격을 어렵게 만들 수 있음. 초당 2^{40}개 키 검사 가능한 컴퓨터 사용하더라도 대략 890만년($\frac{2^{48}}{31536000}$) 소요
- 그러나 평문 내 동일 문자의 빈도수가 암호문 내에서도 유지되므로 통계분석공격에 취약 → 예) 영문에 대한 암호문의 경우 암호문 내 가장 많이 출현하는 심볼은 영문자 E에 대응하는 것으로 추정할 수 있음

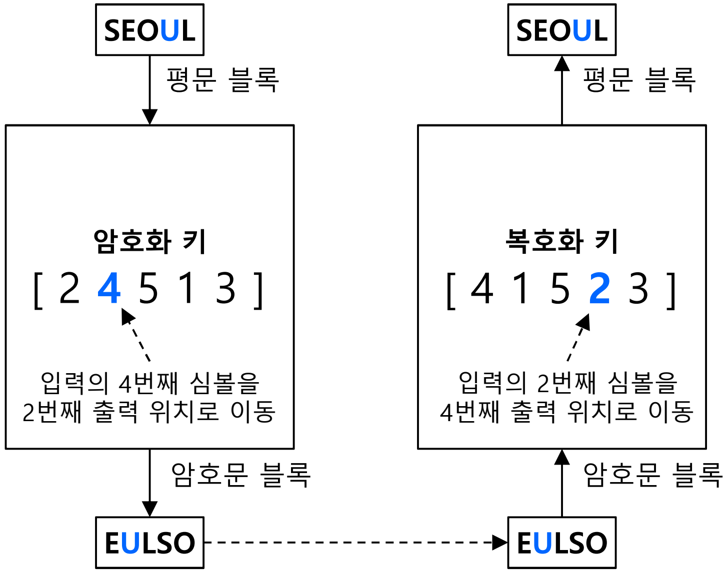
평문 심볼	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
암호문 심볼	H	O	E	A	X	L	W	C	N	K	G	Y	D	R	J	M	B	U	Z	P	V	T	F	S	Q	I

← 26! 개 중 하나

전치 기반 암호 방법 예 (1/2)



전치 기반 암호 방법 예 (2/2)



전치 암호화 키에 대한 복호화 키 생성

```
import java.util.Arrays;

public class Test {
    public static void main(String[] args) {
        int encKey[]= {2,4,5,1,3}; // 첫 심볼 인덱스를 1로 가정
        int decKey[]=getDecKey(encKey);
        System.out.println("암호화 키: "+Arrays.toString(encKey));
        System.out.println("복호화 키: "+Arrays.toString(decKey));
    }
    private static int[] getDecKey(int[] encKey) {
        int decKey[]=new int[encKey.length];
        for (int i = 0; i < encKey.length; i++) {
            decKey[ encKey[i] - 1 ] = i+1 ;
        }
        return decKey;
    }
}
```

현대 암호 알고리즘

- 대칭키(symmetric-key) 암호 기술
  - 송수신측이 동일한 비밀키로 암호화 및 복호화
  - 구분
    - 블록 암호
      - 대치(substitution) 기반 암호 방법
        - 평문 내 심볼을 다른 심볼로 변경하는 방식으로 암호화
      - 전치(transposition) 기반 암호 방법
        - 평문 내 심볼들의 위치를 변경하는 방식으로 암호화
    - 스트림 암호 기술
- 비대칭키(asymmetric-key) 암호 (공개키(public-key) 암호) 기술
  - 송수신측이 서로 다른 키들(공개키, 개인키)을 이용하여 암호화 및 복호화
  - 구분
    - 정수 인수분해 문제 기반
      - RSA
    - 이산대수 문제 기반
      - Diffie-Hellman Key Exchange
    - 타원곡선 기반
      - Elliptic Curve Diffie-Hellman(ECDH) Key Exchange

암호학적 해쉬 함수: 응용

- 암호학적 해쉬함수
  - 임의 크기 입력에 대해 (암호학적 특성을 갖는) 고정 크기 출력 생성
  - 응용
    - 메시지 인증 (Message authentication)
      - 수신 메시지가 송신측이 보낸 (변경되지 않은 진짜) 메시지인지 확인
    - 디지털 서명 (Digital signatures)
      - 메시지의 해쉬 값에 대해 송신측 개인키로 디지털 서명 수행
    - 패스워드 파일
      - 사용자 패스워드의 해쉬 값을 패스워드 파일에 저장
      - 로그인 시 입력된 패스워드의 해쉬 값을 이전 해쉬 값과 비교
    - 침입 탐지, 바이러스 탐지
      - 시스템 내 각 파일의 해쉬 값 계산하여 별도 보관
      - 이후 각 파일의 해쉬 값 재계산하여 이전 해쉬 값과 비교

A.txt

Seoul

B.txt

SEoul

C.txt

한국 미국 독일 영국 서울 부산 울산 제주

Windows 명령프롬프트에서 실행

certutil -hashfile A.txt MD5	fd38499c5c04df42d1d78807aa4b7d7d
certutil -hashfile B.txt MD5	85a744acf3a1d6fe4968211cba8d691b
certutil -hashfile C.txt MD5	39fceb046e18549f802ecdc8105fe858

대칭키 암호 기술 vs. 비대칭키 암호 기술 (1/2)

대칭키 암호 기술

- 송신자와 수신자 간 공유하는 비밀키(secret key) 존재
- 송신측에서는 비밀키로 암호화하고, 수신측에서는 동일한 비밀키로 복호화함

비대칭키 암호 기술

- 개인키(private key)와 공개키(public key)의 쌍이 존재하며 개인키는 비밀로 유지하고 공개키는 공개되어 있음
- 활용법 1 → 송신측은 수신측 공개키로 암호화하고, 수신측은 자신의 개인키로 복호화함 (암호화)
- 활용법 2 → 송신측은 자신의 비밀키로 암호화하고, 수신측은 송신측의 공개키로 복호화함 (전자서명)
- 활용법 3 → 대칭키 암호기술에서 사용되는 비밀키 설정 (키 구축)

대칭키 암호 기술 vs. 비대칭키 암호 기술 (2/2)

대칭키 암호 기술

- 키 분배 문제 존재 → 송수신측 간 안전한 채널을 통해 비밀키가 공유되어야 함
- 키의 개수 → 임의의 쌍방 간 암호화 전송이 필요한 사용자 n명에 대해 총 $\frac{n(n-1)}{2}$개 키 필요
- 키 구축 기능 제공 불가
- 부인불가(non-repudiation) 기능 제공 불가 → 앨리스는 자신이 작성하고 AES로 암호화하여 ABC전자에 보낸 제품 주문서에 대해 그 주문서는 ABC전자가 작성한 것이라고 주장하면서 자신이 해당 주문서를 작성하고 전송한 사실을 부인(repudiation)할 수 있음 (앨리스와 ABC전자는 동일한 비밀키를 알고 있음)
- 3DES, AES

비대칭키 암호 기술

- 키 구축 기능 제공 → 디피-헬만 키 교환 등
- 부인불가(non-repudiation) 기능 제공 → 자신의 개인키로 암호화한 메시지의 송신측은 해당 메시지의 생성을 부인할 수 없음
- 비대칭키 암호화는 대칭키 암호화에 비해 대략 100~1000배 느림 (Paar & Pelzl, 2010) → 대량 메시지 암호화에 활용 어려움
- RSA, Diffie-Hellman Key Exchange

부인불가

- 메시지 작성 측이 해당 메시지를 작성한 사실을 부인할 수 없는 상황

XOR 기반 암호

A XOR B		
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

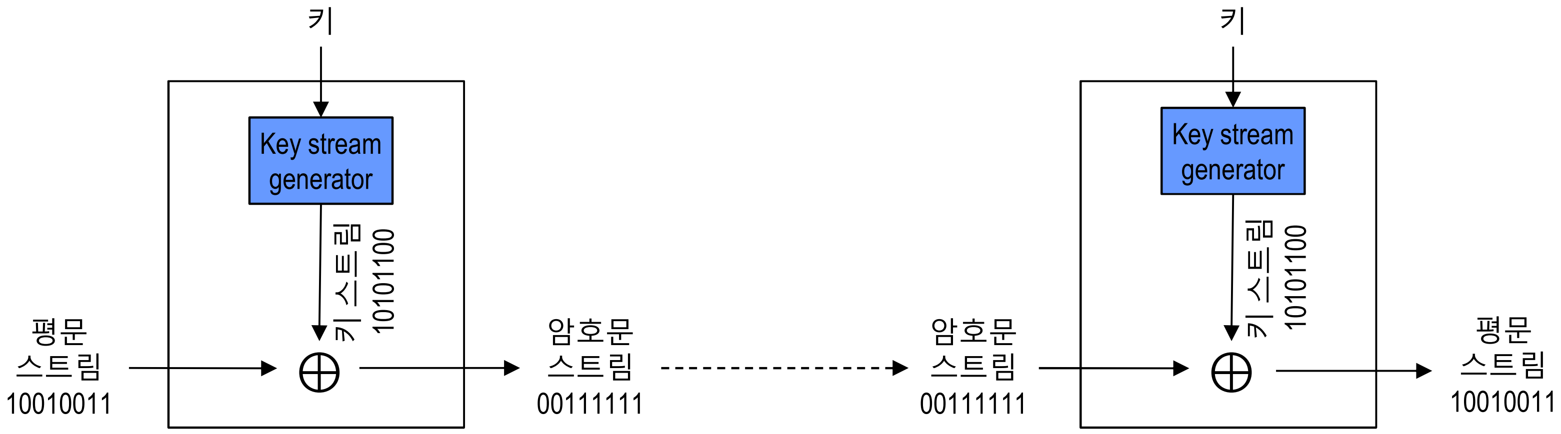
	0	1	0	1	평문
$\oplus$	0	0	1	1	키
	0	1	1	0	암호문

	0	1	1	0	암호문
$\oplus$	0	0	1	1	키
	0	1	0	1	평문

스트림 암호 기술

스트림 암호 기술

- 일반적으로 한 번에 평문 한 바이트씩 암호화 수행 (비트 단위 혹은 한 바이트보다 큰 단위로도 설계 가능)
- Key stream generator의 출력인 키 스트림과 평문 스트림을 XOR하여 암호문 스트림 생성
- 일반적으로 블록 암호 시스템보다 빠르게 동작
- RC4, A5/1, ChaCha20



	10010011	평문 스트림
⊕	**10101100**	**키 스트림**
	00111111	암호문 스트림

	00111111	암호문 스트림
⊕	**10101100**	**키 스트림**
	10010011	평문 스트림

암호 기술과 수학

- 암호 수학
  - 나눗셈과 나머지
  - 최대공약수
  - 서로소
  - 유클리드 알고리즘, 확장 유클리드 알고리즘
  - 모듈러 연산
  - 법 n에 대해 합동
  - 군, 환, 체, 유한체, 갈루아체, 다항식
  - 역원
  - 소수
  - 오일러 ϕ함수
  - 오일러 정리
  - 순환군
  - 이산대수문제
  - ...

암호 기술과 수학

AES

- 군(Group), 환(Ring), 체(Field)
- 유한체, 갈루아체(Galois field)
- $GF(2),\ GF(2^n)$
- 다항식 덧셈
- 다항식 곱셈
- 곱셈에 대한, 다항식의 역원

RSA

• 큰 두 소수 $p,\ q$ 찾기 • $n = p \times q$ 계산 • $\phi(n) = (p-1)\ \times (q-1)$ 계산 • e 선택 $(\gcd(\phi(n), e) = 1,\ 1 < e < \phi(n))$ • d 계산 $((e \times d)\ mod\ \phi(n)\ = 1)$ • 공개키 (e, n), 개인키 (d, n) • 평문 T에 대해 $(T < n)$, 암호문 $C = T^e\ mod\ n$ 생성 • 암호문 C에 대해, 평문 $T = C^d\ mod\ n$ 복원	• 두 소수 $p = 3,\ q = 11$ 선택 • $n = p \times q = 3 \times 11 = 33$ 계산 • $\phi(n) = (p-1)\ \times (q-1)\ = 20$ 계산 • $e = 3$ 선택 $(\gcd(\phi(n), e) = 1,\ 1 < e < \phi(n))$ • $d = 7$ 계산 $((e \times d)\ mod\ \phi(n)\ = 1)$ • 공개키 $(e = 3, n = 33)$, 개인키 $(d = 7, n = 33)$ • 평문 $T = 4$에 대해 $(T < n)$, 암호문 $C = T^e\ mod\ n =\ 4^3\ mod\ 33 = 31$ 생성 • 암호문 $C = 31$에 대해, 평문 $T = C^d\ mod\ n =\ 31^7\ mod\ 33 = 4$ 복원

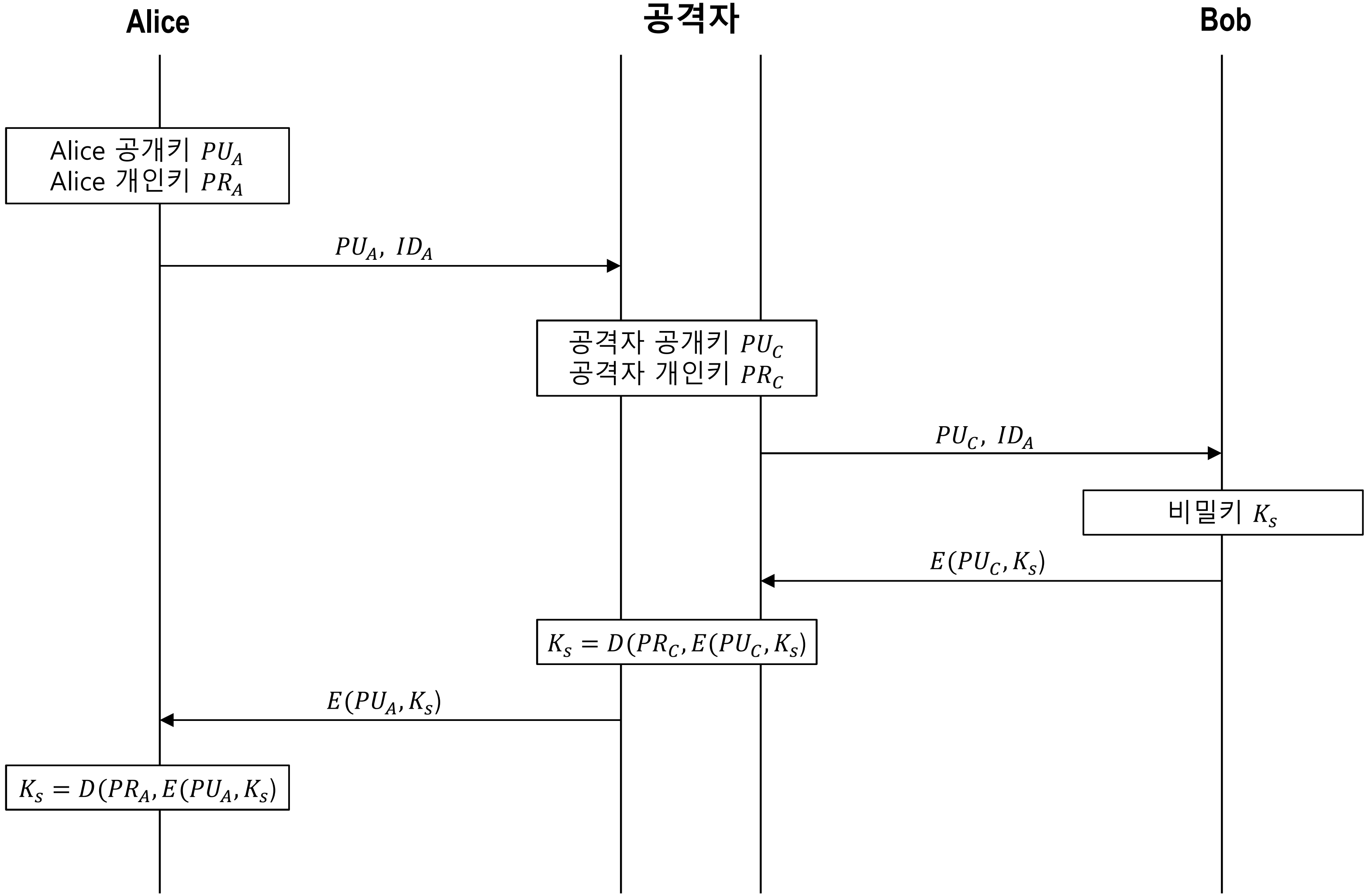
- 모듈러 연산
- 소수정리
- 오일러 totient function
- 최대공약수
- 곱셈의 역원
- 모듈러 지수승의 효율적 계산, square-and-multiply algo.
- 오일러 정리

Diffie-Hellman Key Exchange (순환군, 원시근, 모듈러 연산, 이산대수문제)

Alice	Bob
큰 소수 p와 원시근 α는 공개되어 있음 비밀키 a를 선택 $(a < p)$ 공개키 $A = \alpha^a\ mod\ p$를 계산 후 Bob에게 송신 대칭키 $K = B^a\ mod\ p$를 획득	큰 소수 p와 원시근 α는 공개되어 있음 비밀키 b를 선택 $(b < p)$ 공개키 $B = \alpha^b\ mod\ p$를 계산 후 Alice에게 송신 대칭키 $K = A^b\ mod\ p$를 획득
소수 $p = 29$, 원시근 $\alpha = 2$ 공개되어 있음 비밀키 $a = 5$를 선택 $(a < p)$ 공개키$A = \alpha^a\ mod\ p = 2^5\ mod\ 29 = 3$을 계산 후 Bob에게 송신 대칭키 $K = B^a\ mod\ p = 7^5\ mod\ 29 = 16$을 획득	소수 $p = 29$, 원시근 $\alpha = 2$ 공개되어 있음 비밀키 $b = 12$를 선택 $(b < p)$ 공개키$B = \alpha^b\ mod\ p = 2^{12}\ mod\ 29 = 7$을 계산 후 Alice에게 송신 대칭키$K = A^b\ mod\ p = 3^{12}\ mod\ 29 = 16$을 획득

Man-In-The-Middle Attack (MITM attack, 중간자 공격) 예

Reference: (Stallings 2014. p429)



Question

- 송수신 양측이 동일한 대칭키를 어떻게 확보할 것인가
  - 공개키 암호기술 활용
- 공개키는 다른 안전 장치 없이 단순히 공개하면 되는가
  - Alice의 공개키라고 수신한 공개키가 진짜 Alice의 공개키가 맞는지 어떻게 확신할 수 있는가
  - 공개키기반구조(Public-Key Infrastructure, PKI)
    - 공개키에 대해 제3자(인증기관, Certificate Authority)가 인증한 인증서 필요
- 키 안전 확보 이후에는 변경 없이 계속 사용해도 되는가
  - 동일한 키 K를 사용하여 지난 1년간 전송한 모든 암호문을 공격자가 보관해 두었다고 할 때, 해당 키 K를 공격자가 알게 되는 경우 지난 1년간 전송한 모든 암호문 내용이 해독됨

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