2026/11/11, 2026 보안 기초 세미나

암호학과 네트워크 보안

-3장 고전 대칭-키 암호, 4장 암호 수학-

서 진 우(jinwoo@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 고전 대칭-키 암호
- 암호 수학

목 차

- 고전 대칭-키 암호
- 암호 수학

고전 대칭-키 암호

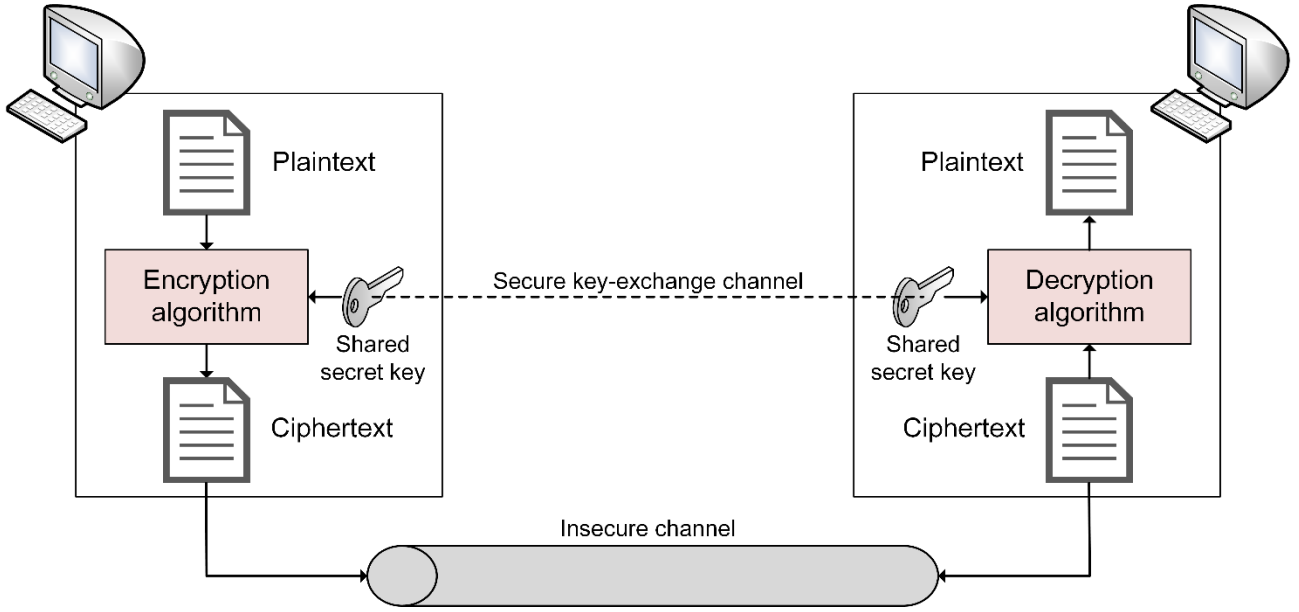
- 개요 (1/10)

*채널(Channel)
송/수신자간 통신 경로

  - 대칭-키(Symmetric-Key) 암호 (1/3)
    - 정의
      - 암호화 키와 복호화 키가 동일한 방식의 암호화 기술
    - 주요 용어
      - 평문(Plaintext)
        - 송신자가 수신자에게 전달하는 본래의 메시지
      - 암호문(Ciphertext)
        - *채널을 통해 보내는 메시지
      - 암호(Cipher)
        - 정보를 안전하게 전달하기 위한 암/복호화 알고리즘
      - 대칭-키(Symmetric-Key)
        - 암/복호화를 위해 송/수신자 사이에 공유된 동일한 비밀 키

고전 대칭-키 암호

- 개요 (2/10)
  - 대칭-키(Symmetric-Key) 암호 (2/3)
    - 원리
      1. 송/수신자가 안전한 채널을 통해 비밀 키를 사전 공유
      2. 송신자가 평문과 비밀 키로 암호문 생성
      3. 암호문 전송
      4. 수신자는 동일한 비밀 키로 평문 복원



고전 대칭-키 암호

- 개요 (3/10)
  - 대칭-키(Symmetric-Key) 암호 (3/3)
    - 특징
      - 양방향 동일한 키 사용
      - 키 공유를 위한 별도의 안전한 통신 채널 필요
        - 직접 전달
          - e.g., 대면 또는 물리적 매체를 통한 키를 교환하는 방식
        - 제 3자 이용
          - e.g., 신뢰할 수 있는 *KDC가 키를 분배하는 방식
        - 비대칭-키 암호를 이용한 임시 키 생성
          - e.g., *HTTPS에서 공개키 방식으로 키를 교환하는 방식
      - 암호화 알고리즘과 복호화 알고리즘이 서로 역관계
        - $D_K(E_K(x)) = E_K(D_K(x)) = x$
      - 키 관리 부담
        - m명이 상호 통신 시 $m(m-1)/2$개의 키 필요

*KDC(Key Distribution Center)
용자들 간의 암호 통신을 위한 키를 안전하게 생성하고 분배하는 중앙 시스템
*HTTPS(Hypertext Transfer Protocol Secure)
웹 브라우저와 웹사이트 간의 통신을 암호화하여 안전하게 데이터를 주고받기 위한 보안 프로토콜

고전 대칭-키 암호

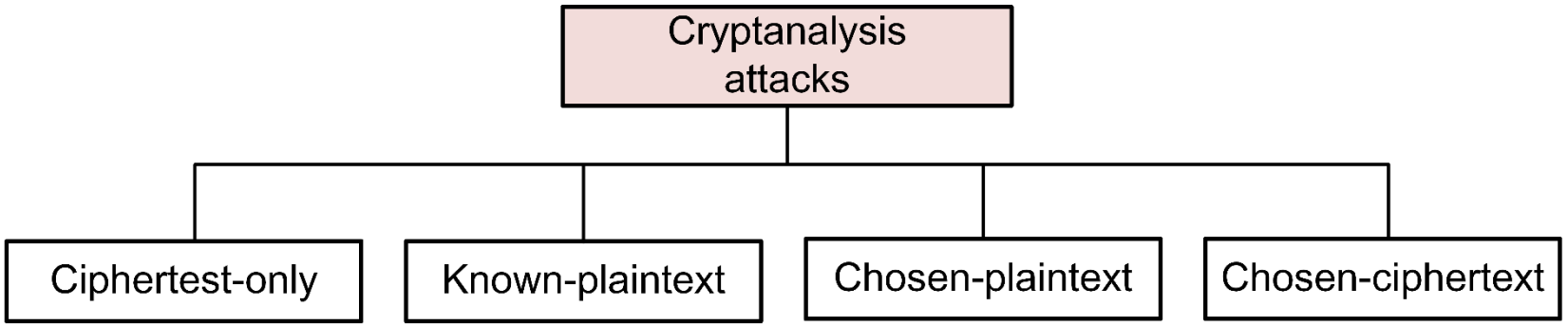
- 개요 (4/10)
  - Kerckhoff의 원리
    - 정의
      - 암호 시스템의 안전성은 암호 알고리즘의 기밀성이 아니라 비밀키의 기밀성에 의존해야 한다는 원리
    - 특징
      - 공격자가 암/복호화 알고리즘을 알고 있다는 전제
      - 비밀키의 기밀성에 의존
      - 알고리즘을 비공개로 유지하는 것보다 비밀키를 안전하게 관리하는 것이 중요

고전 대칭-키 암호

- 개요 (5/10)
  - 암호 해독 (1/6)
    - 정의
      - 암호를 해독하여 평문, 암호문, 키를 유추하는 행위
    - 공격 기법
      - 전수조사 공격(Brute-force Attack)
        - 가능한 모든 키를 하나씩 대입하는 공격
      - 통계적인 공격(Statistical Attack)
        - 평문 언어의 고유한 특징(Inherent Characteristics of the Plaintext Language)으로부터 정보를 얻어 해당 특성을 이용하는 공격
      - 패턴 공격(Pattern Attack)
        - 암호문에서 반복되는 구조를 이용하는 공격

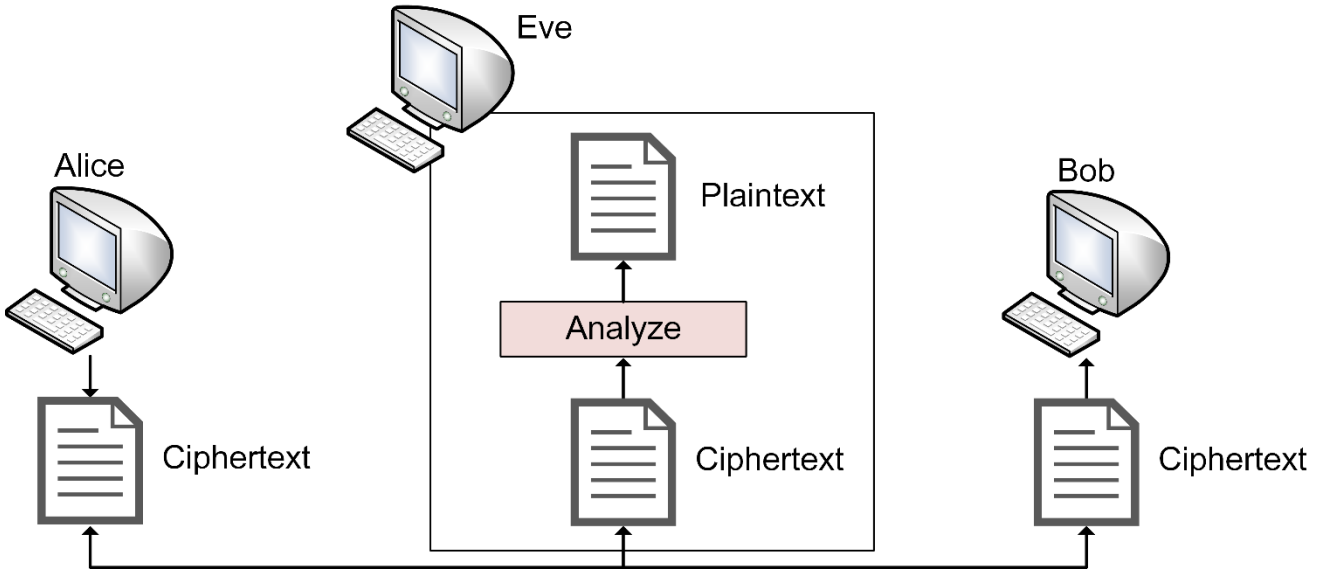
고전 대칭-키 암호

- 개요 (6/10)
  - 암호 해독 (2/6)
    - 공격 모델
      - 암호문 단독 공격(Ciphertext-only Attack)
      - 알려진 평문 공격(Known-plaintext Attack)
      - 선택 평문 공격(Chosen-plaintext Attack)
      - 선택 암호문 공격(Chosen-ciphertext Attack)



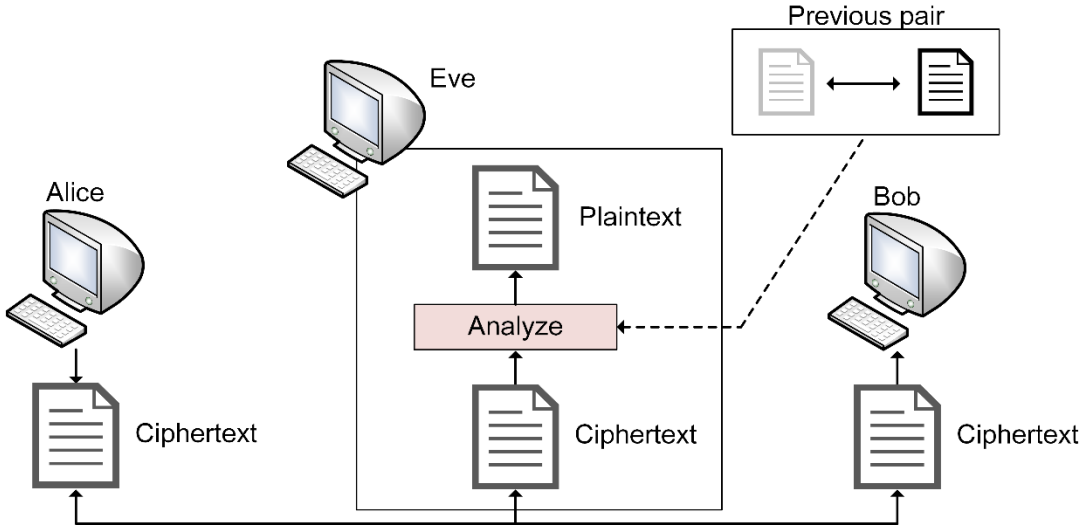
고전 대칭-키 암호

- 개요 (7/10)
  - 암호 해독 (3/6)
    - 암호문 단독 공격(Ciphertext-only Attack)
      - 정의
        - 암호문을 얻어 대응되는 평문 또는 키를 탐색하는 공격
      - 특징
        - 공격자는 알고리즘을 알고 있으며, 암호문 도청 가능 상황 가정
        - 공격자가 가진 정보가 가장 적은 공격



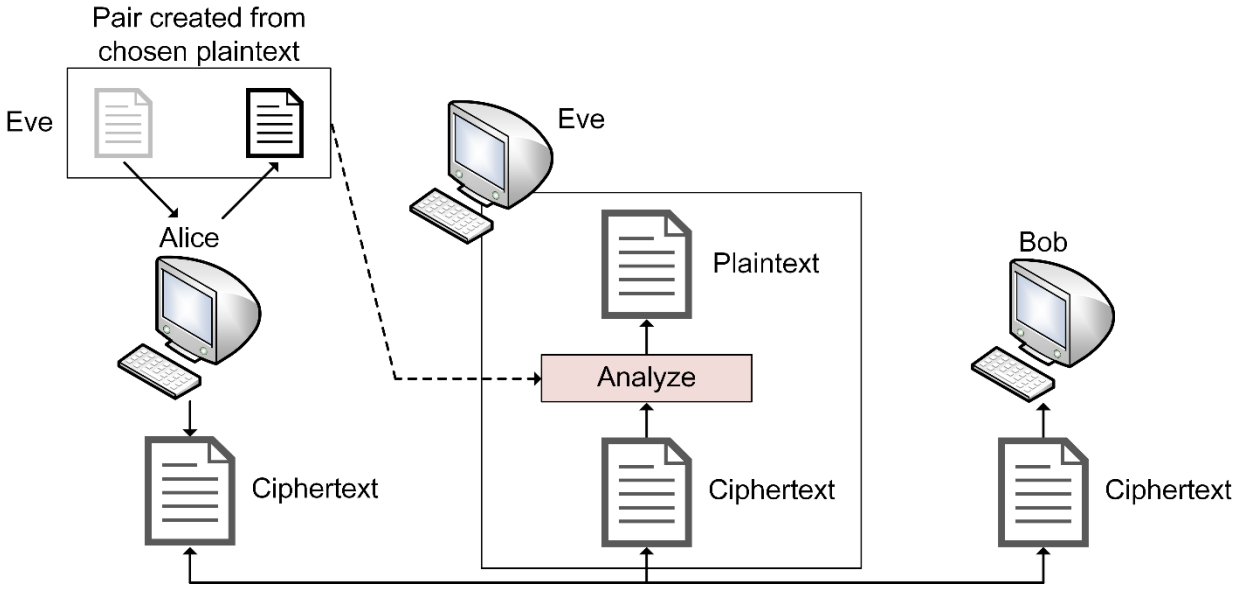
고전 대칭-키 암호

- 개요 (8/10)
  - 암호 해독 (4/6)
    - 알려진 평문 공격(Known-plaintext Attack)
      - 정의
        - 평문/암호문 쌍을 이용하여 새로운 암호문의 평문 또는 키를 탐색하는 공격
      - 특징
        - 공격자는 이미 노출된 평문/암호문 쌍만 확보 가능한 상황 가정
        - 평문/암호문 쌍이 노출된 경우에만 성립하여 암호문 단독 공격보다 적용 상황이 제한적인 공격



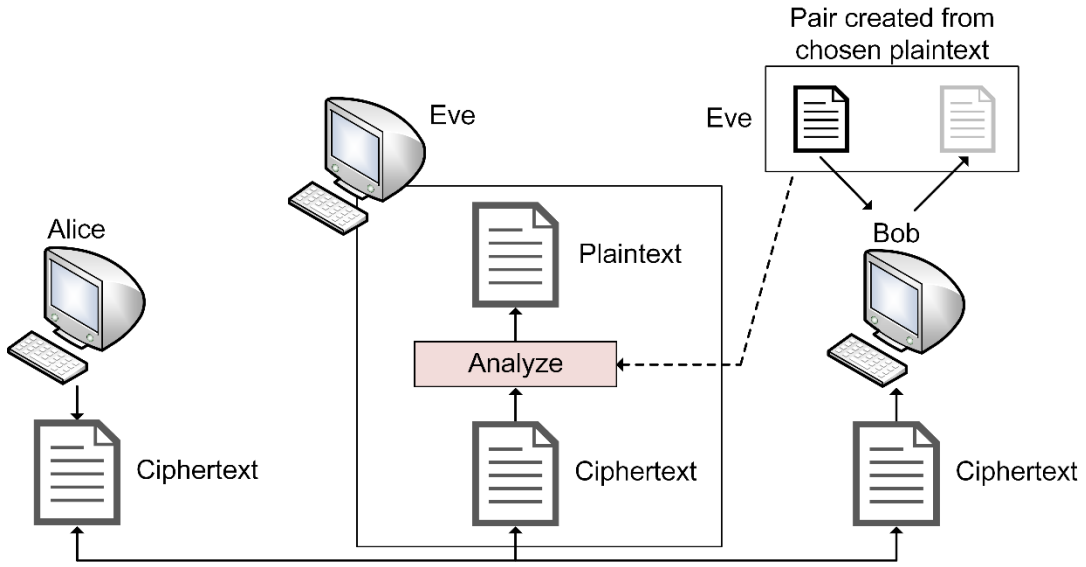
고전 대칭-키 암호

- 개요 (9/10)
  - 암호 해독 (5/6)
    - 선택 평문 공격(Chosen-plaintext Attack)
      - 정의
        - 공격자가 선택한 평문의 암호문을 얻어 키를 탐색하는 공격
      - 특징
        - 공격자는 원하는 평문을 직접 암호화할 수 있는 상황 가정
        - 분석에 유리한 입력을 골라 넣을 수 있어 알려진 평문 공격보다 해독이 쉬운 공격



고전 대칭-키 암호

- 개요 (10/10)
  - 암호 해독 (6/6)
    - 선택 암호문 공격(Chosen-ciphertext Attack)
      - 정의
        - 공격자가 선택한 암호문의 평문을 얻어 키를 탐색하는 공격
      - 특징
        - 공격자는 원하는 암호문을 직접 복호화할 수 있는 상황 가정
        - 공격자가 가진 정보가 가장 많은 공격



고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (1/36)
  - 정의
    - 하나의 기호를 다른 기호로 대체하는 방식
  - 예시
    - 알파벳일 경우
      - $A \rightarrow D, T \rightarrow Z$
    - 숫자일 경우
      - $2 \rightarrow 6, 3 \rightarrow 7$
  - 종류
    - 단일문자 암호(Monoalphabetic Ciphers)
    - 다중문자 암호(Polyalphabetic Ciphers)

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (2/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (1/12)
    - 정의
      - 평문에서 하나의 문자 혹은 기호가 위치와 관계없이 같은 문자 혹은 기호로 대체되는 암호
    - 특징
      - 평문의 어떤 글자와 암호문에 대응되는 글자는 일-대-일 대응 관계
        - e.g., 평문에서 A가 항상 D로 대체될 경우, 모든 A는 D로 대체
    - 예제 3.1

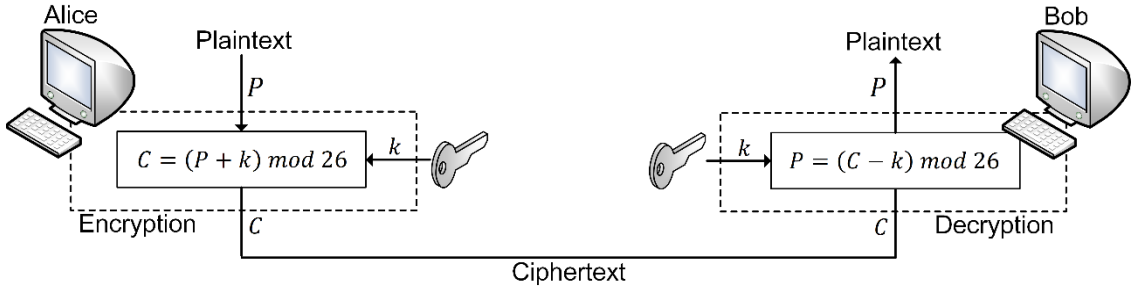
다음 평문과 암호문에 대해 단일문자 암호인지 아닌지 판별하여라.

1. 평문: hello, 암호문: KHOOR
2. 평문: hello, 암호문: ABNZF

- 풀이
  1. 알파벳 I가 O로 모두 암호화하였기 때문에, 단일문자일 가능성이 높음
  2. 첫 번째 알파벳 I는 N, 두 번째 알파벳 I는 Z로 암호화되었으므로, 단일문자 암호가 아님

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (3/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (2/12)
    - 덧셈 암호 (1/4)
      - 정의
        - 평문 문자에 키를 더해 암호문 문자를 생성하는 암호
      - 특징
        - 가장 간단한 단일문자 암호임
        - 이동 암호(Shift Cipher) 혹은 시저 암호(Caeser Cipher)라고 불림
        - $C = (P + k) \, mod \, 26, \;\; P = (C - k) \, mod \, 26$
        - 평문, 암호문, 키는 Z_{26}의 원소임



Plaintext →	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
Ciphertext →	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
Value →	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

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (4/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (3/12)
    - 덧셈 암호 (2/4)
      - 특징
        - 키 공간이 작고 26개의 키가 존재하기 때문에, 전수조사 공격을 이용한 암호문 단독 공격에 취약
        - 공격자가 긴 암호문을 가지고 있는 경우, 통계적인 공격에 취약
      - 예제 3.2

키가 15인 덧셈 암호를 이용하여 “hello” 암호화 및 “WTAAD”의 복호화를 해보아라.

- 풀이

평문	암호화	암호문	암호문	복호화	평문
h = 07	$(07 + 15)\ mod\ 26$	22 = W	W = 22	$(22 - 15)\ mod\ 26$	07 = h
e = 04	$(04 + 15)\ mod\ 26$	19 = T	T = 19	$(19 - 15)\ mod\ 26$	04 = e
l = 11	$(11 + 15)\ mod\ 26$	00 = A	A = 00	$(00 - 15)\ mod\ 26$	11 = l
l = 11	$(11 + 15)\ mod\ 26$	00 = A	A = 00	$(00 - 15)\ mod\ 26$	11 = l
o = 14	$(14 + 15)\ mod\ 26$	03 = D	D = 03	$(03 - 15)\ mod\ 26$	14 = o

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (5/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (4/12)
    - 덧셈 암호 (3/4)
      - 예제 3.3

암호문 “UVACLYFZLJBYL”에 대해 전수조사 공격을 이용하여 암호문을 해독하라.

- 풀이
  - 키 값을 1부터 7까지 1씩 증가시키며 복호화 진행

키 값	평문
1	tuzbkxeykiaxk
2	styajwdxjhzwj
3	rsxzhvwcigyvi
4	qrwygubvhfxuh
5	pqvxftaugewtg
6	opuwezszfdvsf
7	**notverysecure**

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (6/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (5/12)
    - 덧셈 암호 (4/4)
      - 예제 3.4

"XLILSYWIGMRSAJTVSPEJSVJSYVVMPPSRHSPPEVWXMWSVSVLSSVVEBBCFIJSVIVWIVPPIVWIGMIZIWSVVSTJJIVW"에 대해 통계적인 공격을 이용하여 암호문을 해독하라.

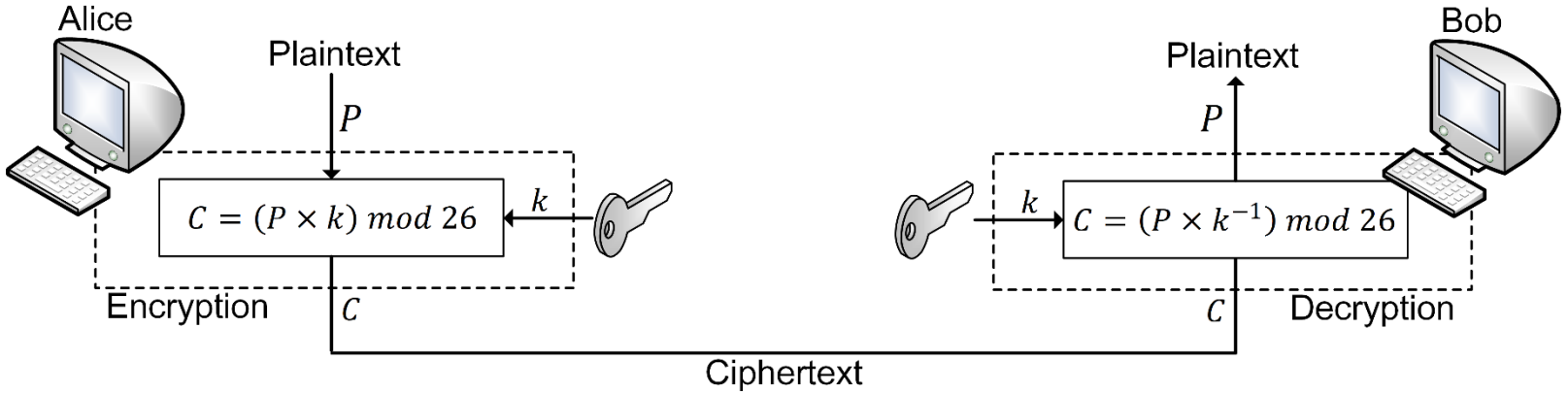
- 풀이
  - 암호문에서 암호문 문자에 대한 빈도 분석 해볼 결과 I가 14번으로 가장 많이 등장하므로 Key=4임

∴ "the house is now for sale for four million dollars it is worth more hurry before the seller recieves more offers"

암호문 문자	빈도
I	14
V	13
S	12

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (7/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (6/12)
    - 곱셈 암호 (1/2)
      - 정의
        - 평문 문자에 키를 곱해 암호문 문자를 생성하는 암호
      - 특징
        - 복호화를 위해 키는 곱셈 역원이 존재하는 Z_{26}^*의 원소
        - $C = (P \times k) \bmod 26,\ P = (C \times k^{-1}) \bmod 26$
        - 키 공간 12개로 전수조사 공격에 취약
        - 문자 출현 빈도가 유지되어 통계적인 공격에 취약



고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (8/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (7/12)
    - 곱셈 암호 (2/2)
      - 예제 3.5

키 7로 “hello” 암호화, “XCZZU” 복호화를 진행하시오.

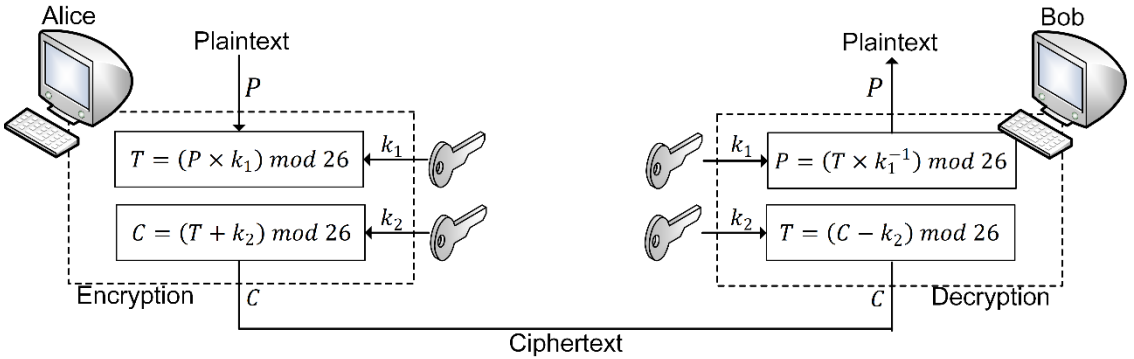
- 풀이

평문	암호화	암호문
h = 07	$(07 \times 7)\ mod\ 26$	23 = X
e = 04	$(04 \times 7)\ mod\ 26$	02 = C
l = 11	$(11 \times 7)\ mod\ 26$	25 = Z
l = 11	$(11 \times 7)\ mod\ 26$	25 = Z
o = 14	$(14 \times 7)\ mod\ 26$	20 = U

암호문	복호화	평문
X = 23	$(23 \times 7^{-1})\ mod\ 26 = (23 \times 15)\ mod\ 26$	07 = h
C = 02	$(02 \times 7^{-1})\ mod\ 26 = (02 \times 15)\ mod\ 26$	04 = e
Z = 25	$(25 \times 7^{-1})\ mod\ 26 = (25 \times 15)\ mod\ 26$	11 = l
Z = 25	$(25 \times 7^{-1})\ mod\ 26 = (25 \times 15)\ mod\ 26$	11 = l
U = 20	$(20 \times 7^{-1})\ mod\ 26 = (20 \times 15)\ mod\ 26$	14 = o

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (9/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (8/12)
    - 아핀 암호 (1/4)
      - 정의
        - 두 개의 키를 사용하고, 덧셈 암호와 곱셈 암호를 결합한 암호
      - 특징
        - 곱셈 암호 적용 후 덧셈 암호 적용
        - 첫 번째 키는 Z_{26}^*의 원소로 곱셈 암호에 사용, 두 번째 키는 Z_{26}의 원소로 덧셈 암호에 사용
        - 암호 알고리즘 $C = (P \times k_1 + k_2)\ mod\ 26$
        - 복호 알고리즘 $P = \left((C - k_2) \times k_1^{-1}\right)\ mod\ 26$



고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (10/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (9/12)
    - 아핀 암호 (2/4)
      - 특징
        - 복호화 시 덧셈 역원으로 뺄셈 연산 후 곱셈 역원으로 곱셈 연산
        - 덧셈/곱셈 암호를 모두 이용하기 때문에, 전수조사/통계적인 공격 적용 가능
      - 예제 3.7

키 (7,2)로 “hello” 암호화, “ZEBBW” 복호화를 진행하시오.

- 풀이

평문	암호화	암호문
h = 07	$(07 \times 7 + 2)\ mod\ 26$	25 = **Z**
e = 04	$(04 \times 7 + 2)\ mod\ 26$	04 = **E**
l = 11	$(11 \times 7 + 2)\ mod\ 26$	01 = **B**
l = 11	$(11 \times 7 + 2)\ mod\ 26$	01 = **B**
o = 14	$(14 \times 7 + 2)\ mod\ 26$	22 = **W**

암호문	복호화	평문
Z = 25	$((25-2) \times 7^{-1})\ mod\ 26 = ((25-2) \times 15)\ mod\ 26$	07 = **h**
E = 04	$((04-2) \times 7^{-1})\ mod\ 26 = ((04-2) \times 15)\ mod\ 26$	04 = **e**
B = 01	$((01-2) \times 7^{-1})\ mod\ 26 = ((01-2) \times 15)\ mod\ 26$	11 = **l**
B = 01	$((01-2) \times 7^{-1})\ mod\ 26 = ((01-2) \times 15)\ mod\ 26$	11 = **l**
W = 22	$((22-2) \times 7^{-1})\ mod\ 26 = ((22-2) \times 15)\ mod\ 26$	14 = **o**

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (11/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (10/12)
    - 아핀 암호 (3/4)
      - 예제 3.8 (1/2)

"PWUFFOGWCHFDWIWEJOUUNJORSMDWRHVCMWJUPVCCG"를 선택 평문 공격으로 복호화를 진행하시오. 이때 공격자는 다음 2가지 알고리즘을 알고 있다고 가정하자.

알고리즘	평문	암호문
1번	et	WC
2번	et	WF

- 풀이
  - 알고리즘 2을 사용할 경우
    - e(04)는 W(22)이므로 $(04 \times k_1 + k_2) \equiv 22 (mod\ 26)$
    - t(19)는 C(02) 이므로 $(19 \times k_1 + k_2) \equiv 02 (mod\ 26)$
    - $\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 22 \\ 2 \end{bmatrix} = \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix} \begin{bmatrix} 22 \\ 2 \end{bmatrix} = \begin{bmatrix} 16 \\ 10 \end{bmatrix}$

$\therefore k_1 = 16, k_2 = 10$

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (12/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (11/12)
    - 아핀 암호 (4/4)
      - 예제 3.8 (2/2)

"PWUFFOGWCHFDWIWEJOUUNJORSMDWRHVCMWJUPVCCG"를 선택 평문 공격으로 복호화를 진행하시오. 이때 공격자는 다음 2가지 알고리즘을 알고 있다고 가정하자.

알고리즘	평문	암호문
1번	et	WC
2번	et	WF

- 풀이
  - 알고리즘 2을 사용할 경우
    - e(04)는 W(22)이므로 $(04 \times k_1 + k_2) \equiv 22 (mod\ 26)$
    - t(19)는 F(05) 이므로 $(19 \times k_1 + k_2) \equiv 05 (mod\ 26)$
    - $\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 22 \\ 5 \end{bmatrix} = \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix} \begin{bmatrix} 22 \\ 5 \end{bmatrix} = \begin{bmatrix} 11 \\ 4 \end{bmatrix}$

$\therefore k_1 = 11, k_2 = 4$

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (13/36)
  - 단일문자 암호(Monoalphabetic Ciphers) (12/12)
    - 단일문자 대치 암호
      - 정의
        - 평문 문자와 암호문 문자를 대응시켜 대치하는 암호
      - 특징
        - 키 공간 크기는 $26!\ (\approx 4 \times 10^{26})$으로 덧셈/곱셈/아핀 암호에 비해 상대적으로 큰 키 공간
        - 문자의 출현 빈도수를 변경시키지 않기 때문에 문자의 출현 빈도수를 이용한 통계적인 공격 적용 가능
      - 예제 3.9

아래 표를 사용하여 “this message is easy to encrypt but hard to find the key”를 암호화하시오.

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

- 풀이

  ∴ ICFVQRVVNERFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (14/36)
  - 다중문자 암호(Polyalphabetic Ciphers) (1/23)
    - 정의
      - 각 문자가 다른 대치를 가지는 다중문자 대치를 사용하는 암호
    - 특징
      - 평문 문자와 암호문 문자와의 관계는 일 대 다 대응
      - 언어의 문자 빈도를 감추브로 단일 문자 빈도 해독 사용 불가
    - 키
      - 평문 문자의 위치에 따라 정해지는 부분키들로 구성된 키 수열
        - $k = (k_1, k_2, k_3, \dots)$이며 k_i는 평문의 i번 째 문자를 암호화 시 사용되는 부분키
    - 종류
      - 자동키 암호, 플레이페어 암호, Vigenere 암호, Hill 암호, 일회용 패드, Rotor 암호, 에니그마 기계

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (15/36)
  - 다중문자 암호(Polyalphabetic Ciphers) (2/23)
    - 자동키 암호 (1/2)
      - 정의
        - 암호화 과정 중 부분키가 평문으로부터 자동으로 생성되는 다중문자 암호
      - 특징
        - 첫 번째 부분키는 비밀리에 합의한 사전에 정의된 값이지만, 두 번째 부분키부터는 직전 평문 문자의 값(0~25)
        - $P = P_1P_2P_3 \ldots$, $C = C_1C_2C_3 \ldots$, $k = (k_1, P_1, P_2, \ldots)$
        - 암호화 $C_i = (P_i + k_i)\ mod\ 26$, 복호화 $P_i = (C_i - k_i)\ mod\ 26$
        - 평문의 단일문자 빈도 통계를 감추지만 첫 번째 부분키가 1~25의 25개 값 중 하나라 전수조사 공격에 취약
        - 언어의 특징을 감출 뿐만 아니라 키 공간이 큰 다중문자 암호가 필요

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (16/36)
  - 다중문자 암호(Polyalphabetic Ciphers) (3/23)
    - 자동키 암호 (2/2)
      - 예제 3.10

초기 키 값 $k_1 = 12$ 일 때의 “attack is today”의 암호화 해보아라.

- 풀이

평문	a	t	t	a	c	k	i	s	t	o	d	a	y
P 값	00	19	19	00	02	10	08	18	19	14	03	00	24
키 수열	12	00	19	19	00	02	10	08	18	19	14	03	00
C 값	12	19	12	19	02	12	18	00	11	7	17	03	24
암호문	M	T	M	T	C	M	S	A	L	H	R	D	Y

      - 예제 3.11

“UBHOZVPQWGWLV”를 전수조사 공격으로 복호화 해보아라.

- 풀이

키 값	평문
1	tizpklemkwalk
2	sjyqjmdnjxzmj
3	rkxrincoiyyni

키 값	평문
4	qlwshobphzxoh
5	pmvtgpagawpg
6	onuufqzrfbvqf

키 값	평문
7	notverysecure

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (17/36)
  - 다중문자 암호(Polyalphabetic Cipher) (4/23)
    - 플레이페어 암호(Playfair Cipher) (1/3)
      - 정의
        - 5×5 행렬로 배열한 25개의 알파벳 문자를 비밀 키로 사용하는 다중문자 암호
      - 특징
        - 평문을 두 문자씩 묶어서 암호화
        - 문자 I와 J는 동일한 문자로 간주
      - 키
        - 행렬에서 문자의 배열을 바꿔 서로 다른 많은 비밀 키를 생성
          - e.g., 오른쪽 위 모서리부터 대각선 방향으로 문자를 배치한 아래의 비밀 키

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

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (18/36)
  - 다중문자 암호(Polyalphabetic Cipher) (5/23)
    - 플레이페어 암호(Playfair Cipher) (2/3)
      - 전처리
        - 한 쌍의 두 문자가 같으면 둘을 분리하기 위해 가짜 문자(Bogus Letter)를 삽입
        - 가짜 문자 삽입 후 평문 문자의 개수가 홀수이면 맨 끝에 가짜 문자를 추가
      - 암호화를 위한 세 가지 법칙
        1. 두 문자가 키의 같은 행에 있으면 각 문자의 오른쪽 문자로 암호화
        2. 두 문자가 같은 열에 있으면 각 문자의 아래 문자로 암호화
        3. 같은 행이나 열에 있지 않으면 각 문자의 행에 있으면서 상대 문자와 같은 열에 있는 문자로 암호화
           - 행이나 열의 마지막 문자이면 그 행이나 열의 첫 번째 문자를 사용
      - 키 수열
        - 키는 두 개씩 생성되는 부분키들의 수열이며 키 수열과 암호 수열이 같음
        - $k = [(k_1, k_2), (k_3, k_4), \dots]$, 암호화 $C_i = k_i$, 복호화 $P_i = k_i$
        - 각 평문 문자에 대한 부분키는 쌍을 이루는 다음이나 이전 문자에 의존
          - e.g., "hello"에서 l은 쌍에 따라 Q와 B로 다르게 암호화

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (19/36)
  - 다중문자 암호(Polyalphabetic Cipher) (6/23)
    - 플레이페어 암호(Playfair Cipher) (3/3)
      - 암호 해독
        - 키 공간의 크기가 25!이므로 전수조사 공격은 매우 어려움
        - 단일 문자 빈도는 숨기지만 두 문자열의 빈도는 일정 부분 보존
        - 두 문자열 빈도 테스트에 기반한 암호문 단독 공격으로 키를 찾을 수 있음
      - 예제 3.11

플레이페어 암호로 “hello”를 암호화하시오.

- 풀이
  - 두 개의 l 사이에 가짜 문자 x를 삽입하여 he, lx, lo로 묶음
  - he는 같은 행이므로 각 문자의 오른쪽 문자인 EC
  - lx는 같은 열이므로 각 문자의 아래 문자인 QZ
  - lo는 같은 행·열이 아니므로 각 문자의 행에서 상대 문자의 열에 있는 BX

  ∴ “ECQZBX”

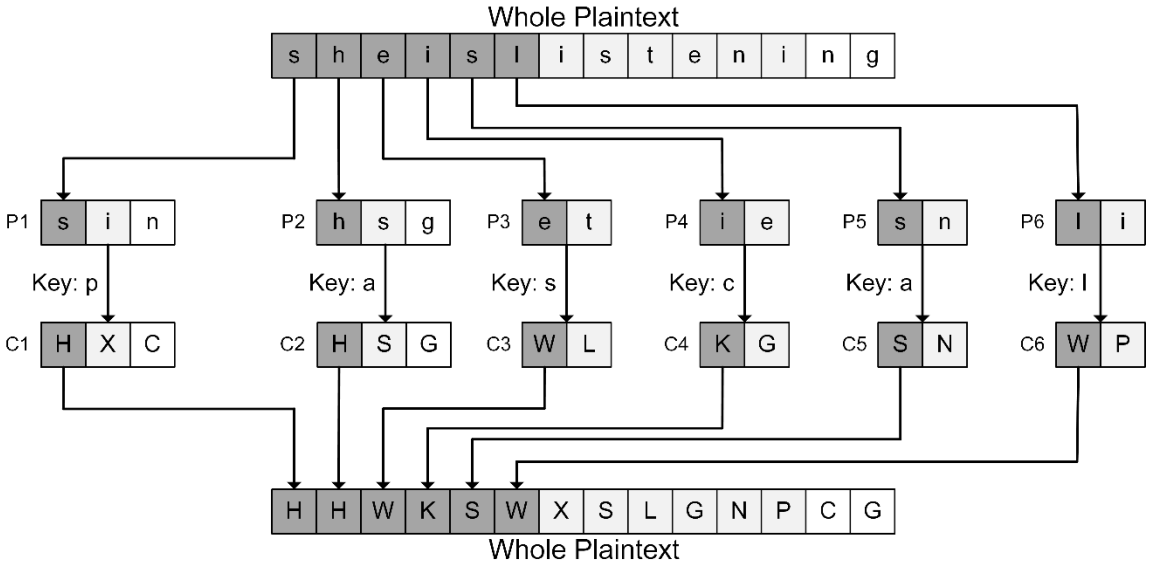
L	G	D	B	A
Q	M	H	E	C
U	R	N	I/J	F
X	V	S	O	K
Z	Y	W	T	P

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (20/36)
  - 다중문자 암호(Polyalphabetic Cipher) (7/23)
    - Vigenere 암호 (1/3)
      - 정의
        - 길이가 $m(1 \leq m \leq 26)$인 초기 비밀 키 수열을 반복하여 키 수열로 사용하는 다중문자 암호
      - 특징
        - 키 수열이 평문 문자에 의존하지 않고 평문 문자의 위치에만 의존
        - 평문을 모르고도 키 수열을 생성할 수 있음
      - 표기
        - 초기 비밀 키 $(k_1, k_2, \dots, k_m)$일 때 $k = [(k_1, \dots, k_m), (k_1, \dots, k_m), \dots]$
        - 암호화 $C_i = P_i + k_i$, 복호화 $P_i = C_i - k_i$
      - 덧셈 암호와의 관계
        - Vigenere 암호는 m개의 덧셈 암호의 조합
        - 평문을 m개의 조각으로 나누고 각 조각을 서로 다른 키의 덧셈 암호로 암호화
        - 덧셈 암호는 $m = 1$인 Vigenere 암호

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (21/36)
  - 다중문자 암호(Polyalphabetic Cipher) (8/23)
    - Vigenere 암호 (2/3)
      - Vigenere 표
        - 첫 번째 행은 평문 문자, 첫 번째 열은 키 문자, 나머지는 암호문 문자
        - 평문 문자의 열과 키 문자의 행이 교차하는 지점이 암호문 문자
          - e.g., 평문 문자 s와 키 문자 P의 교차점은 H



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

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (22/36)
  - 다중문자 암호(Polyalphabetic Cipher) (9/23)
    - Vigenere 암호 (3/3)
      - 예제 3.12

키워드 “PASCAL”을 사용하는 Vigenere 암호로 “She is listening”을 암호화하시오.

- 풀이
  - 키워드를 수치로 바꾸면 초기 키 수열은 (15, 0, 18, 2, 0, 11)
  - 키 수열을 평문 길이만큼 반복하여 각 평문 문자에 더하고 mod 26 연산

평문	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P 값	18	07	04	08	18	11	08	18	19	04	13	08	13	06
키 수열	15	00	18	02	00	11	15	00	18	02	00	11	15	00
C 값	07	07	22	10	18	22	23	18	11	06	13	19	02	06
암호문	H	H	W	K	S	W	X	S	L	G	N	T	C	G

∴ “HHWKSWXSLGNTCG”

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (23/36)
  - 다중문자 암호(Polyalphabetic Cipher) (10/23)
    - Vigenere 암호의 암호 해독 (1/3)
      - 방법
        - 키의 길이를 찾은 뒤 키를 찾는 두 단계로 해독
      - 키의 길이 찾기
        - Kasiski 테스트로 암호문에서 3문자 이상 반복되는 조각을 찾음
        - 두 조각 사이의 거리 d는 키 길이 m의 배수로 추정
        - 반복 조각의 거리가 $d_1, d_2, \ldots, d_n$이면 m은 이들의 최대공약수의 약수
      - 키 찾기
        - 암호문을 m개의 조각으로 나누고 각 조각에 덧셈 암호 해독 방법을 적용
        - 전체 암호문은 단일 문자 빈도를 보존하지 않지만 각 조각은 보존

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (24/36)
  - 다중문자 암호(Polyalphabetic Cipher) (11/23)
    - Vigenere 암호의 암호 해독 (2/3)
      - 예제 3.13 (1/2)

다음 암호문을 Kasiski 테스트를 이용하여 해독하시오.

LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLW
PCZVGTHVTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFZM
VVWLGYHCUSWXQHKVGSHEEVFLCFDGVSUMPHKIRZDMPHHB
VWVWJWIXGFWLTSHGJOUEEHHVUCFVGOWICQLTJSUXGLW

- 풀이
  - 3문자 반복 조각과 두 조각 사이의 거리를 구함
  - 거리 100, 48, 60, 8의 최대공약수가 4이므로 키 길이는 4의 배수
  - $m = 4$로 가정하고 암호문을 네 조각으로 나눔
    - C_1은 1, 5, 9, …번째 문자로 구성

문자열	첫 번째 인덱스	두 번째 인덱스	거리
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60
MPH	119	127	8

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (25/36)
  - 다중문자 암호(Polyalphabetic Cipher) (12/23)
    - Vigenere 암호의 암호 해독 (3/3)
      - 예제 3.13 (2/2)

다음 암호문을 Kasiski 테스트를 이용하여 해독하시오.

LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLW
PCZVGTHVTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFZM
VVWLGYHCUSWXQHKVGSHEEVFLCFDGVSUMPHKIRZDMPHHB
VWVWJWIXGFWLTSHGJOUEEHHVUCFVGOWICQLTJSUXGLW

- 풀이
  - 각 조각에 통계적인 공격을 적용하면 키는 (2, 14, 3, 4)
    - e.g., C_1의 최빈 문자 G를 평문 e에 대응시키면 키는 2
  - 각 조각을 복호화한 뒤 한 문자씩 끼워 넣어 전체 평문을 구성
  - 평문이 말이 되지 않으면 다른 m으로 다시 시도

  ∴ 키 “code”, 평문 “Julius Caesar used a cryptosystem in his wars, which is now referred to as Caesar cipher. It is an additive cipher with the key set to three. Each character in the plaintext is shifted three characters to create ciphertext”

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (26/36)
  - 다중문자 암호(Polyalphabetic Cipher) (13/23)
    - Hill 암호 (1/3)
      - 정의
        - 평문을 크기 m인 블록으로 나누고 $m \times m$ 정방 행렬 키로 블록 단위로 암호화하는 다중문자 암호
      - 특징
        - 블록의 각 문자가 블록 내 다른 문자들의 암호화에 영향을 주므로 블록 암호
      - 키
        - 각 원소가 k_{ij}인 $m \times m$ 정방 행렬 K
        - Z_{26}에서 곱셈에 대한 역행렬을 갖는 행렬이어야 복호화 가능

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix}$$

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (27/36)
  - 다중문자 암호(Polyalphabetic Cipher) (14/23)
    - Hill 암호 (2/3)
      - 암호화
        - $C_1 = P_1k_{11} + P_2k_{21} + \cdots + P_mk_{m1}$
        - $C_2 = P_1k_{12} + P_2k_{22} + \cdots + P_mk_{m2}$
        - $C_m = P_1k_{1m} + P_2k_{2m} + \cdots + P_mk_{mm}$
          - e.g., C_1은 블록 $(P_1, P_2, \ldots, P_m)$의 모든 평문 문자에 의존
      - 행렬 표현
        - 블록이 l개이면 평문은 $l \times m$ 행렬 P
        - 암호화 $C = PK$, 복호화 $P = CK^{-1}$

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (28/36)
  - 다중문자 암호(Polyalphabetic Cipher) (15/23)
    - Hill 암호 (3/3)
      - 예제 3.14

오른쪽 그림의 4×4 키 행렬을 사용하는 Hill 암호로 “code is ready”를 암호화하시오.

- 풀이
  - 띄어쓰기를 없애고 4문자 블록으로 나눈 뒤 마지막 블록에 가짜 문자 z를 삽입
  - 각 문자를 수치로 바꿔 3×4 평문 행렬을 만듦
  - 평문 행렬에 키 행렬을 곱하고 mod 26 연산

  ∴ “OHKNIHGHFISS”

$$\overset{C}{\begin{bmatrix}14 & 07 & 10 & 13\\ 08 & 07 & 06 & 11\\ 11 & 08 & 18 & 18\end{bmatrix}} = \overset{P}{\begin{bmatrix}02 & 14 & 03 & 04\\ 08 & 18 & 17 & 04\\ 00 & 03 & 24 & 25\end{bmatrix}} \overset{K}{\begin{bmatrix}09 & 07 & 11 & 13\\ 04 & 07 & 05 & 06\\ 02 & 21 & 14 & 09\\ 03 & 23 & 21 & 08\end{bmatrix}}$$

Encryption

$$\overset{P}{\begin{bmatrix}02 & 14 & 03 & 04\\ 08 & 18 & 17 & 04\\ 00 & 03 & 24 & 25\end{bmatrix}} = \overset{C}{\begin{bmatrix}14 & 07 & 10 & 13\\ 08 & 07 & 06 & 11\\ 11 & 08 & 18 & 18\end{bmatrix}} \overset{K^{-1}}{\begin{bmatrix}02 & 15 & 22 & 03\\ 15 & 00 & 19 & 03\\ 09 & 09 & 03 & 11\\ 17 & 00 & 04 & 07\end{bmatrix}}$$

Decryption

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (29/36)
  - 다중문자 암호(Polyalphabetic Cipher) (16/23)
    - Hill 암호의 암호 해독 (1/2)
      - 암호문 단독 공격
        - 키 공간의 크기가 $26^{m \times m}$에 가까워 전수조사 공격이 매우 어려움
        - 평문의 단일 문자, 두 문자, 세 문자 빈도를 보존하지 않아 빈도 분석을 적용할 수 없음
      - 알려진 평문 공격
        - 공격자가 m과 서로 다른 m개 블록의 평문/암호문 쌍을 알면 적용 가능
        - 쌍으로부터 $m \times m$ 행렬 P와 C를 만들고 $C = PK$에서 $K = P^{-1}C$로 키를 계산
        - P의 역행렬이 없으면 다른 m개의 쌍을 사용

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (30/36)
  - 다중문자 암호(Polyalphabetic Cipher) (17/23)
    - Hill 암호의 암호 해독 (2/2)
      - 예제 3.15

$m = 3$인 Hill 암호에서 다음 세 쌍의 평문/암호문 블록을 가로챘을 때 키 행렬을 구하시오.

$$[05\ \ 07\ \ 10] \longleftrightarrow [03\ \ 06\ \ 00]$$
$$[13\ \ 17\ \ 07] \longleftrightarrow [14\ \ 16\ \ 09]$$
$$[00\ \ 05\ \ 04] \longleftrightarrow [03\ \ 17\ \ 11]$$
$$P \qquad\qquad C$$

- 풀이
  - 각 쌍을 행으로 하여 3×3 행렬 P와 C를 만듦
  - P의 역행렬 P^{-1}을 Z_{26}에서 계산
    - P^{-1}의 행은 (21, 14, 01), (00, 08, 25), (13, 03, 08)
  - $K = P^{-1}C$를 계산

$$\therefore K\text{의 행렬은 } \begin{bmatrix} 02 & 03 & 07 \\ 05 & 07 & 09 \\ 01 & 02 & 11 \end{bmatrix}$$

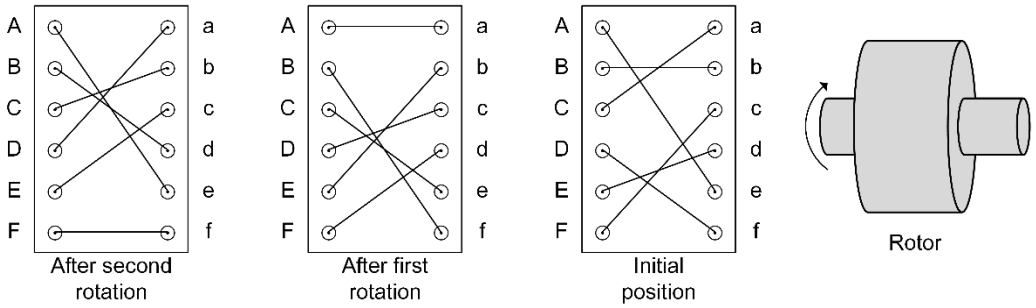
$$\underset{K}{\begin{bmatrix} 02 & 03 & 07 \\ 05 & 07 & 09 \\ 01 & 02 & 11 \end{bmatrix}} = \underset{P^{-1}}{\begin{bmatrix} 21 & 14 & 01 \\ 00 & 08 & 25 \\ 13 & 03 & 08 \end{bmatrix}} \underset{C}{\begin{bmatrix} 03 & 06 & 00 \\ 14 & 16 & 09 \\ 03 & 17 & 11 \end{bmatrix}}$$

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (31/36)
  - 다중문자 암호(Polyalphabetic Cipher) (18/23)
    - 일회용 패드(One-time Pad)
      - 정의
        - 평문과 같은 길이의 키를 완전히 랜덤하게 선택하여 사용하는 암호
      - 원리
        - 각 평문 문자를 키 공간에서 랜덤하게 선택한 키로 암호화하면 완벽한 보안(Perfect Secrecy)을 얻음
          - e.g., 덧셈 암호에서 각 문자의 키를 00~25 중 랜덤하게 04, 02, 21, …로 선택
      - 특징
        - 메시지마다 다른 랜덤 키를 사용하므로 암호문 단독 공격을 포함한 모든 공격이 불가능
        - 메시지마다 새로운 키를 수신자에게 전달해야 하므로 상업적으로 구현하기 어려움

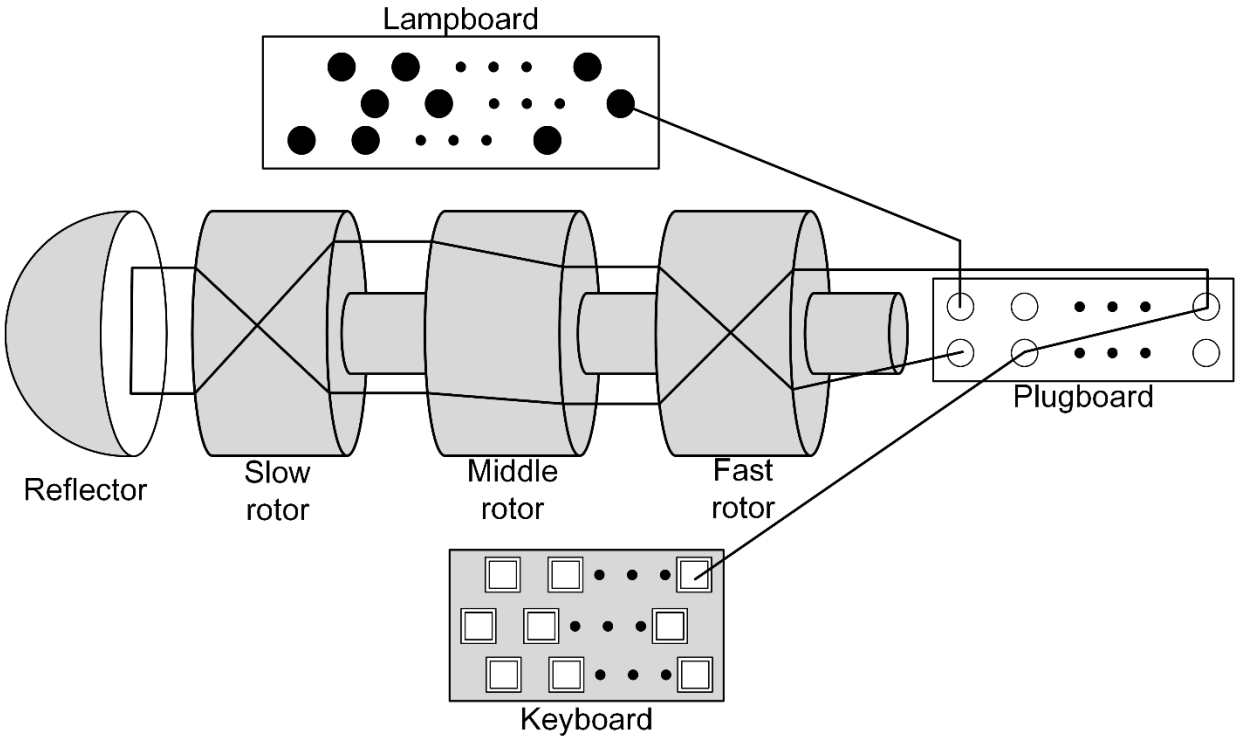
고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (32/36)
  - 다중문자 암호(Polyalphabetic Cipher) (19/23)
    - Rotor 암호
      - 정의
        - 평문 문자를 하나 암호화할 때마다 평문과 암호문 문자의 대응을 바꾸는 단일 알파벳 치환 기반 암호
      - 특징
        - 키는 rotor의 초기 설정
        - 문자 빈도를 보존하지 않음
        - 첫 번째 평문 문자는 초기 설정으로, 이후 문자는 rotor를 1/26 회전시킨 뒤 암호화
          - e.g., 그림의 6문자 rotor에서 "bee"는 정지 시 "BAA", 회전 시 "BCA"
        - 26!개의 가능한 대응 중 초기 대응을 찾아야 하므로 전수조사 공격에 안전
        - 단일 알파벳 치환 암호보다 통계적인 공격에 훨씬 강함



고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (33/36)
  - 다중문자 암호(Polyalphabetic Cipher) (20/23)
    - 에니그마 기계(Enigma Machine) (1/4)
      - 정의
        - rotor 암호의 원리를 기반으로 한 암호 기계



고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (34/36)
  - 다중문자 암호(Polyalphabetic Cipher) (21/23)
    - 에니그마 기계(Enigma Machine) (2/4)
      - 구성 요소
        - 키보드 - 암호화 시 평문, 복호화 시 암호문을 입력하는 26개의 키
        - 램프보드 - 암호화 시 암호문, 복호화 시 평문 문자를 보여주는 26개의 램프
        - 플러그보드 - 13개의 전선으로 수동 연결된 26개의 플러그이며 구성이 매일 변함
        - rotor - 5개 중 매일 3개를 선택하며 빠른 rotor는 문자마다 1/26 회전
        - 중간·느린 rotor는 앞 rotor가 한 바퀴 돌 때마다 1/26 회전
        - 반사경(Reflector) - 사전 연결되어 고정
      - 코드북(Code Book)
        - 매일의 설정을 제공
          - e.g., 사용할 rotor 3개와 설치 순서, 플러그보드 설정, 그날의 세 문자 코드

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (35/36)
  - 다중문자 암호(Polyalphabetic Cipher) (22/23)
    - 에니그마 기계(Enigma Machine) (3/4)
      - 암호화 과정
        1. rotor의 시작점을 그날의 코드로 설정
           - e.g., 코드가 “HUA”이면 rotor를 각각 H, U, A로 초기화
        2. 랜덤한 세 문자 코드를 “선택하고 반복한 코드를 단계 1의 설정으로 암호화
           - e.g., “ACF”를 선택하면 “ACFACF”를 암호화하여 “OPNABT”
        3. rotor의 시작점을 암호화된 코드의 앞 절반인 설정
           - e.g., Rotor의 시작점을 “OPN”으로 설정
        4. 암호화된 6문자를 메시지의 시작에 추가
           - e.g., “OPNABT”를 메시지 시작에 추가
        5. 6문자 코드를 포함한 메시지를 암호화하여 전송
           - e.g., “OPNABTQCNPO”를 전송

고전 대칭-키 암호

- 대치 암호(Substitution Cipher) (36/36)
  - 다중문자 암호(Polyalphabetic Cipher) (23/23)
    - 에니그마 기계(Enigma Machine) (4/4)
      - 복호화 과정
        1. 메시지를 받고 처음 6문자를 분리
           - e.g., 받은 메시지에서 처음 6문자 “OPNABT”를 분리
        2. rotor의 시작점을 그날의 코드로 설정
           - e.g., rotor를 “HUA”로 설정
        3. 단계 2의 설정으로 처음 6문자를 복호화
           - e.g., “OPNABT”를 복호화하여 “ACFACF”를 계산
        4. 복호화된 코드의 앞 절반으로 rotor의 위치를 설정
           - e.g., rotor를 “ACF”로 설정
        5. 처음 6문자를 제외한 메시지를 복호화
           - e.g., 나머지 메시지 “QCNPO”를 복호화

고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (1/15)
  - 정의
    - 기호를 다른 기호로 대체하지 않고 기호의 위치를 재정렬하는 암호
  - 특징
    - 문자 자체가 바뀌지 않아 단일 문자 출현 빈도가 그대로 유지
      - 두/세 문자열 빈도는 보존되지 않으므로, 단일 빈도만 보존되면 전치 암호로 판별 가능
    - 통계적 공격, 전수조사 공격, 패턴 공격에 취약
  - 종류
    - 키가 없는 전치 암호
    - 키가 있는 전치 암호
    - 두 가지 방법의 결합

고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (2/15)
  - 키가 없는 전치 암호 (1/2)
    - 정의
      - 키 없이 정해진 기록 방향과 읽기 방향만으로 문자를 재정렬하는 암호
    - 특징
      - 열 순서로 기록 후 행 순서로 전송
      - 행 순서로 기록 후 열 순서로 전송
      - 사용 방식이나 열 개수만 알 경우, 해독 가능
    - 예시
      - Rail Fence 암호
        - 평문은 지그재그 패턴으로 두 열에 배열되고, 암호문은 행 순서의 패턴으로 읽혀져 생성
        - 첫 번째 행 다음 두 번째 행을 보냄으로써 암호문 생성
        - “meet me at the park”의 경우 “MEMATEAKETETHPR”
        - $\begin{array}{ccccccccccccccc} M & & E & & M & & A & & T & & E & & A & & K \\ & \searrow\nearrow & & \searrow\nearrow & & \searrow\nearrow & & \searrow\nearrow & & \searrow\nearrow & & \searrow\nearrow & & \searrow\nearrow & \\ & E & & T & & E & & T & & H & & P & & R & \end{array}$

고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (3/15)
  - 키가 없는 전치 암호 (2/2)
    - 예제 3.16

"Meet me at the park"를 행 순서로 표에 기록 후 열 순서로 전달해보아라.

- 풀이
  - 행 순서로 표에 표기
    - $\begin{matrix} m & e & e & t \\ m & e & a & t \\ t & h & e & p \\ a & r & k & \end{matrix}$
  - 이후 열 순서로 전달
    - MMTAEEHREAEKTTP

∴ MMTAEEHREAEKTTP

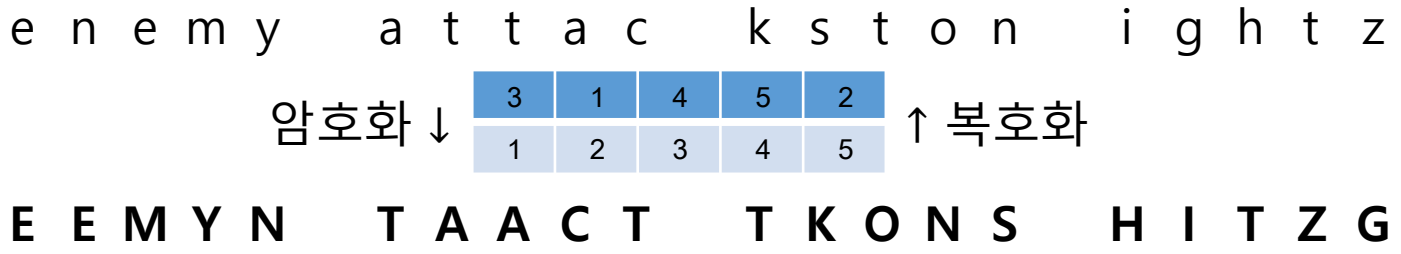
    - 예제 3.17

$$\begin{matrix} 01 & 02 & 03 & 04 & 05 & 06 & 07 & 08 & 09 & 10 & 11 & 12 & 13 & 14 & 15 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \text{의 규칙을 찾아라.} \\ 01 & 05 & 09 & 13 & 02 & 06 & 10 & 13 & 03 & 07 & 11 & 15 & 04 & 08 & 12 \end{matrix}$$

- 풀이
  - $(01, 05, 09, 13), (02, 06, 10, 14)$와 같이 4씩 증가하는 규칙

고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (4/15)
  - 키가 있는 전치 암호
    - 정의
      - 사전에 정한 키를 이용하여 문자를 재배열하는 암호
    - 특징
      - 평문을 블록으로 나눈 뒤 각각의 블록에서 문자를 치환하기 위해 따로 키를 사용
      - 암/복호화 시 문자를 어떻게 치환할지 나타내는 치환 키를 사용
      - 블록 크기를 맞추기 위해 가짜 문자(Bogus Charater)를 삽입 가능
    - 예시
      - “Enemy attacks tonight”을 암호화 시, “EEMYN TAACT TKONS HITZG”



고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (5/15)
  - 두 가지 접근법의 결합 (1/6)
    - 개요
      - 가장 최신의 전치 암호는 더 좋은 섞기(Scrambling)을 위하여 두 가지 방법 결합
    - 단계
      1. 평문을 표에 행 순서로 기록
      2. 열을 재배열하여 치환
      3. 새로운 표를 열 순서로 읽기
    - 특징
      - 1단계와 3단계는 키가 없이 전체적으로 재배열
      - 2단계는 블록단위로 키를 이용하여 재배열
    - 명칭
      - 키가 있는 열 전치 암호(Key Columnar Transposition Ciphers0 또는 열 전치 암호(Columnar Transposition Ciphers)

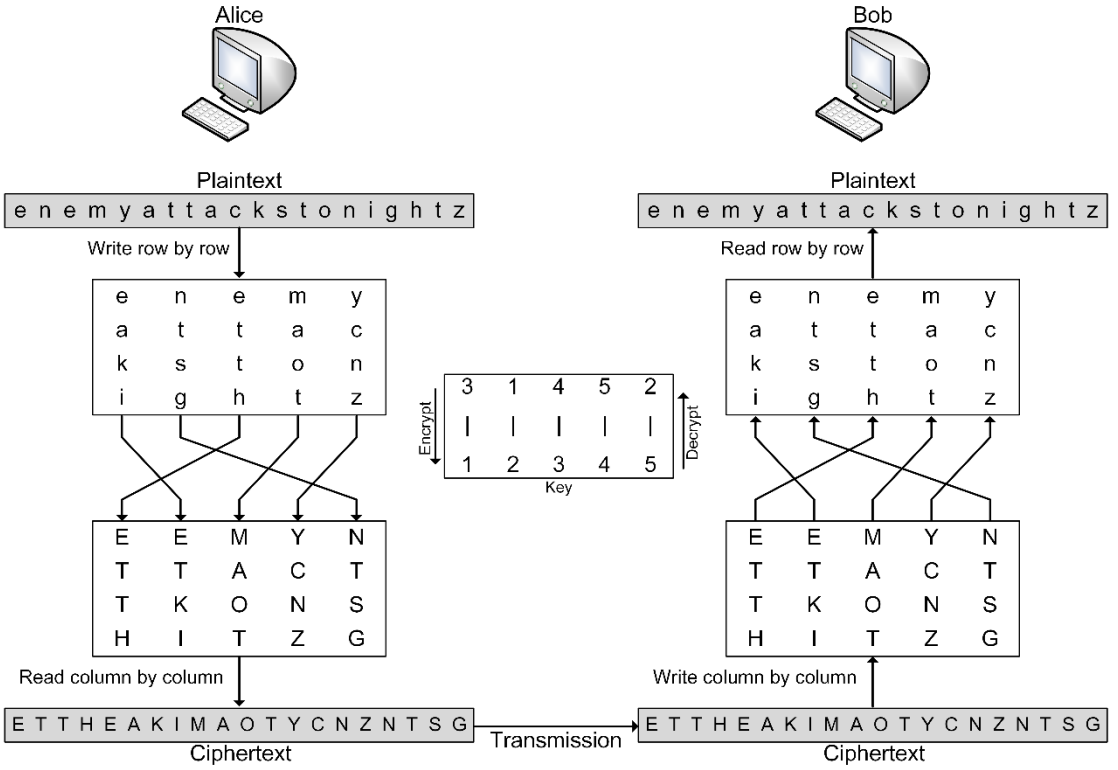
고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (6/15)
  - 두 가지 접근법의 결합 (2/6)
    - 예제 3.18

키 (3 1 4 5 2)를 사용하여 “Enemy attacks tonight”를 결합된 방법으로 암호화하시오.

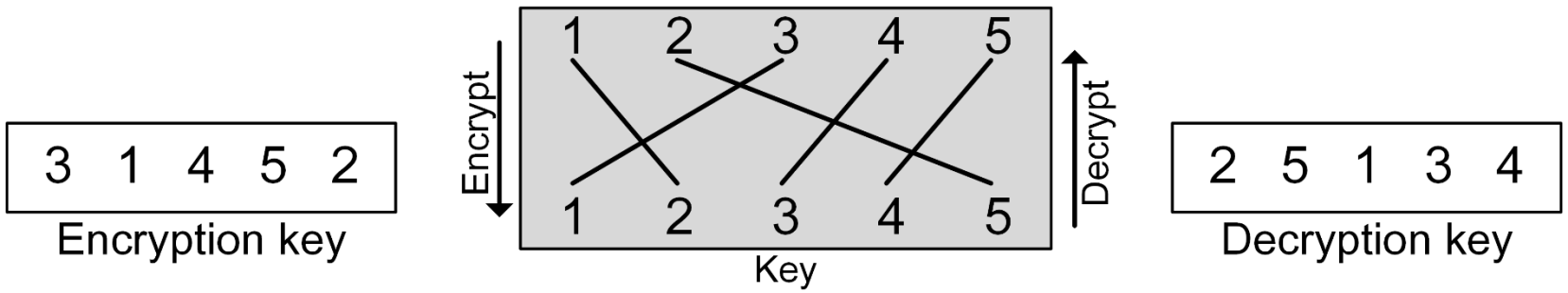
- 풀이
  - 평문을 5열 표에 행 순서로 기록
  - 키에 따라 열을 치환
  - 치환된 표를 열 순서로 읽기

  ∴ “ETTHEAKIMAOTYCNZNTSG”



고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (7/15)
  - 두 가지 접근법의 결합 (3/6)
    - 키 (1/2)
      - 두 개의 키
        - 암호화에 사용하는 키와 복호화에 사용하는 키를 따로 생성
      - 키 표현
        - 각 열에 대해 하나의 값을 가지는 표에 저장
        - 값은 소스의 열 번호, 값의 위치는 대상의 열 번호
          - e.g., 암호화 키 (3 1 4 5 2)는 소스의 세 번째 열이 대상의 첫 번째 열
          - e.g., 복호화 키 (2 5 1 3 4)는 소스의 두 번째 열이 대상의 첫 번째 열



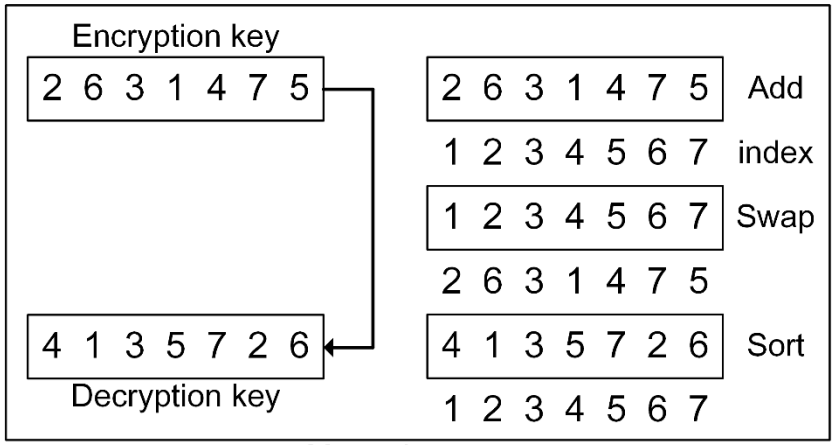
고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (8/15)
  - 두 가지 접근법의 결합 (4/6)
    - 키 (2/2)
      - 키 변환
        - 암호화 키로부터 복호화 키를 구하는 과정

1. 인덱스들을 키 테이블에 기록
2. 내용과 인덱스를 교환
3. 인덱스에 따라 쌍들을 정렬

- e.g., 암호화 키 (2 6 3 1 4 7 5)로부터 복호화 키 (4 1 3 5 7 2 6)

- 알고리즘으로는 각 인덱스에 대해 DecKey[EncKey[index]] ← index를 수행




```
Given: EncKey [index]
index ← 1
    while (index ≤ Column)
    {
    DecKey[EncKey[index]] ← index
    index ← index + 1
    }
Return : DecKey[index]
```

고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (9/15)
  - 두 가지 접근법의 결합 (5/6)
    - 행렬의 사용 (1/2)
      - 개요
        - 암/복호화 과정을 위해 행렬 이용 가능
        - 키 행렬을 전치하면 역행렬 계산 없이 복호화 키 행렬을 얻는 것이 가능
      - 표현
        - 평문과 암호문은 $l \times m$ 행렬, 키는 $m \times m$ *치환 행렬
        - 치환 행렬의 각 행과 열은 하나만 1이고 나머지는 0
      - 암·복호화
        - 암호화는 평문 행렬에 키 행렬을 곱 수행
        - 복호화는 암호문 행렬에 키 행렬의 역행렬을 곱 수행

*치환 행렬(Permutation Matrix)
행이나 열의 순서를 재배치(치환)하는 데 사용되는
정사각 이진 행렬로 0과 1로 구성된 행렬

고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (10/15)
  - 두 가지 접근법의 결합 (6/6)
    - 행렬의 사용 (2/2)
      - 예시
        - 4×5 평문 행렬에 5×5 암호화 키를 곱하면 4×5 암호문 행렬
        - 행렬 곱셈은 열 치환만 수행하고 표에 쓰고 읽는 과정은 알고리즘의 나머지 부분에서 수행

$$\begin{bmatrix} 04 & 13 & 04 & 12 & 24 \\ 00 & 19 & 19 & 00 & 02 \\ 10 & 18 & 19 & 14 & 13 \\ 08 & 06 & 07 & 19 & 25 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 04 & 04 & 12 & 24 & 13 \\ 19 & 00 & 00 & 02 & 19 \\ 19 & 10 & 14 & 13 & 18 \\ 07 & 08 & 19 & 25 & 06 \end{bmatrix}$$

3 1 4 5 2

Plaintext Decryption key Ciphertext

고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (11/15)
  - 전치 암호의 암호 해독 (1/3)
    - 통계적인 공격
      - 문자를 재정렬만 하므로 단일 문자 빈도 분석이 적용 가능
      - 두 문자열·세 문자열의 빈도는 보존하지 않아 이를 이용한 분석은 불가능
      - 단일 문자 빈도만 보존하는 암호문은 전치 암호일 가능성이 높음
    - 전수조사 공격
      - 가능한 키의 개수는 $1! + 2! + 3! + \cdots + L!$로 매우 큼
        - L은 암호문의 길이
      - 열의 개수가 L을 나누므로 열의 개수를 먼저 추측
        - 한 개의 열이나 한 개의 행인 경우는 제외

고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (12/15)
  - 전치 암호의 암호 해독 (2/3)
    - 예제 3.19

키가 있는 전치 암호로 만든 암호문 “EEMYNTAACTTKONSHITZG”를 전수조사 공격으로 해독하시오.

- 풀이
  - $L = 20$이므로 열의 개수는 2, 4, 5, 10, 20 중 하나
  - 열의 개수별로 가능한 치환을 시도
    - 2개는 (2 1) 치환 결과 “ee ym tn aa tc kt no hs ti gz”로 의미 없음
    - 4개는 치환이 없는 경우를 뺀 23개 치환 모두 의미 없음
    - 5개는 (2 5 1 3 4) 치환에서 “enemyattackstonightz”를 얻음
  - 마지막 가짜 문자 z를 없애고 띄어쓰기를 함

∴ “enemy attacks tonight”

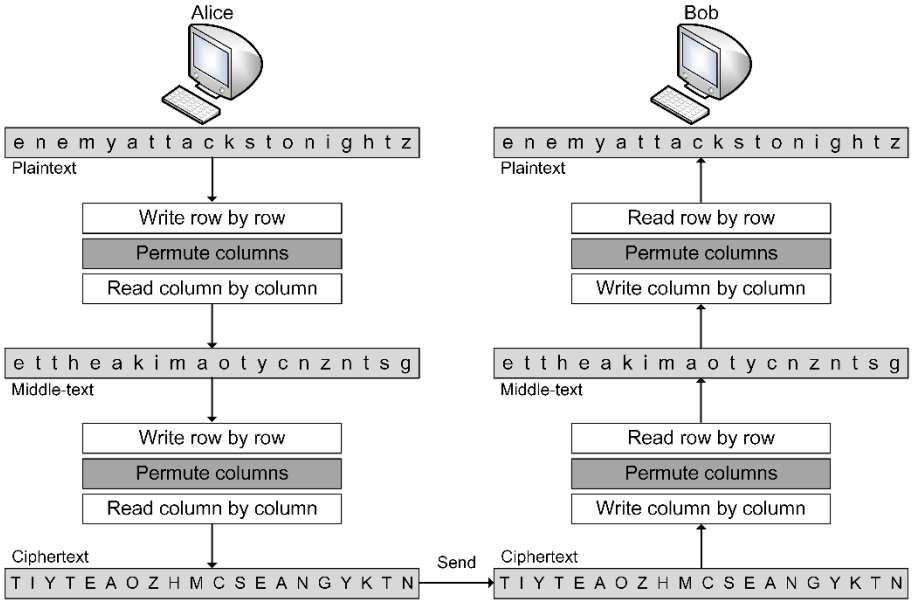
고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (13/15)
  - 전치 암호의 암호 해독 (3/3)
    - 패턴 공격(Pattern Attack)
      - 키가 있는 전치 암호의 암호문은 반복되는 패턴을 가짐
      - 결합된 방법의 암호문 “ETTHEAKIMAOTYCNZNTSG”의 각 문자가 온 평문 위치는 아래 표와 같음
        - (03, 08, 13, 18), (01, 06, 11, 16) 등 5개 그룹에서 인접한 두 수의 차이는 5
      - 열의 개수를 알면 암호문을 그룹으로 나누고 치환하여 평문을 찾을 수 있음

암호문 위치	1	2	3	4	5	6	7	8	9	0	11	12	13	14	15	16	17	18	19	20
평문 위치	03	08	13	18	01	06	11	16	04	09	14	19	05	10	15	20	02	07	12	17

고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (14/15)
  - 이중 전치 암호 (1/2)
    - 정의
      - 결합된 방법의 암호화, 복호화 알고리즘을 두 번 반복하는 암호
    - 특징
      - 단일 문자 빈도 분석은 가능하지만 패턴 공격은 훨씬 더 어려움
      - 각 단계에 다른 키를 쓸 수 있지만 보통 같은 키를 사용



고전 대칭-키 암호

- 전치 암호(Transposition Cipher) (15/15)
  - 이중 전치 암호 (2/2)
    - 예제 3.20

키 (3 1 4 5 2)를 사용하는 이중 전치 암호로 “enemyattackstonightz”를 암호화하시오.

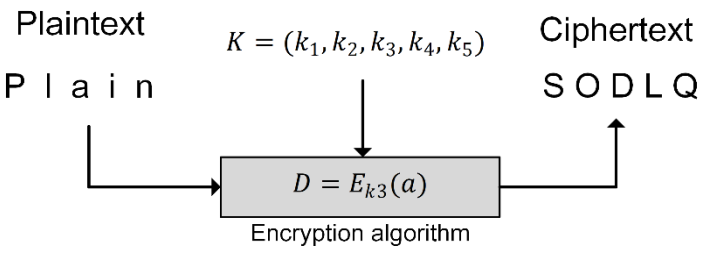
- 풀이
  - 결합된 방법으로 한 번 암호화하면 “ettheakimaotycnzntsg”
  - 같은 키로 한 번 더 암호화
  - 각 암호문 문자가 온 평문 위치에 반복되는 패턴이 없음

암호문 위치	1	2	3	4	5	6	7	8	9	0	11	12	13	14	15	16	17	18	19	20
평문 위치	13	16	05	07	03	06	14	20	18	04	10	12	01	09	15	17	08	11	19	02

∴ “TIYTEAOZHMCSEANGTKTN”

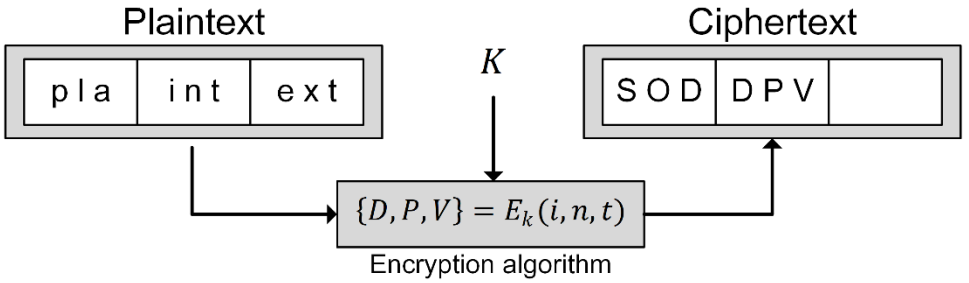
고전 대칭-키 암호

- 스트림 암호와 블록 암호 (1/3)
  - 스트림 암호(Stream Cipher)
    - 정의
      - 평문을 한 비트 또는 한 바이트 단위로 암호화하는 암호
    - 특징
      - 평문 기호마다 키 수열의 대응되는 값을 적용하여 암호문 기호를 하나씩 생성
        - e.g., a일 경우 k_3
      - 키 수열은 사전에 정의된 수열이나 알고리즘으로 생성
        - 평문/암호문/이전 키 값에 의해 결정 가능
      - 키 수열이 평문 문자 위치에 따라 달라질 경우, 단일문자 암호이며 위치와 무관할 경우, 다중문자 암호로 구분
        - e.g., 덧셈 암호, 단일문자 대치 암호는 단일문자 스트림 암호
        - e.g., Vigenere 암호, 자동키 암호는 다중문자 스트림 암호



고전 대칭-키 암호

- 스트림 암호와 블록 암호 (2/3)
  - 블록 암호(block Cipher)
    - 정의
      - 평문을 크기 $m(m > 1)$의 블록 단위로 묶어 한 번에 암호화하는 암호
    - 특징
      - 키가 여러 값으로 구성되어도 하나의 키로 블록 전체를 암호화
        - e.g., Hill 암호는 $m \times m$ 행렬 키를 단일 키로 사용
      - 암호문 블록의 각 문자가 평문 블록 전체에 의해 결정
        - e.g., 플레이페어 암호는 $m = 2$로 두 문자를 함께 암호화
      - 평문 블록 전체가 암호문에 영향을 주므로 모든 블록 암호는 다중문자 암호



고전 대칭-키 암호

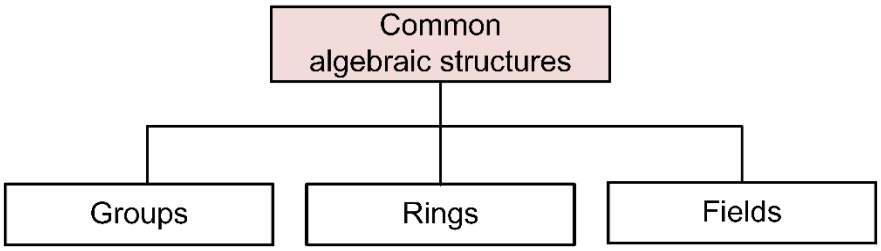
- 스트림 암호와 블록 암호(3/3)
  - 조합
    - 실제 평문 블록은 개별적으로 암호화되지만, 키 수열을 사용하여 블록 단위로 전체 메시지 암호화
      - 개별 블록으로 볼 경우 블록 암호이지만, 각 블록을 한 개의 단위로 생각하고 전체 메시지를 볼 경우 스트림 암호
    - 각 블록은 암호화 중이거나 이전에 생성된 다른 키 사용

목 차

- 보안 개요
- 암호 수학

암호 수학

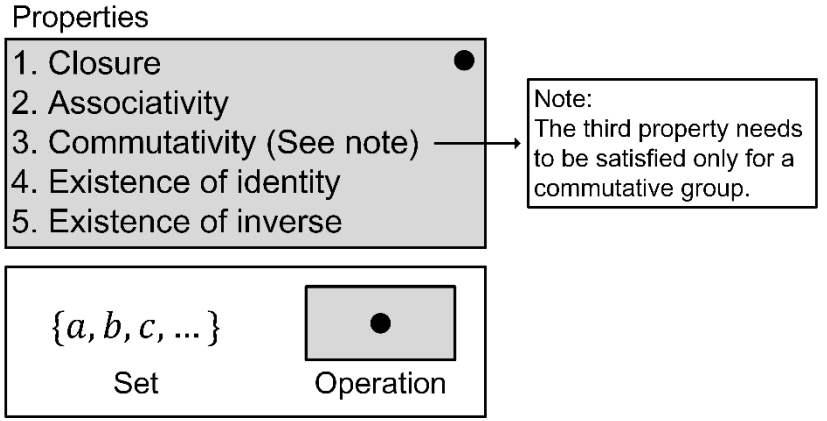
- 대수 구조(Algebraic Structure) (1/26)
  - 정의
    - 집합과 그 집합의 원소들에 적용되는 연산들을 통틀어 이르는 구조
  - 필요성
    - 현대 대칭-키 블록 암호는 n-비트 워드 단위로 덧셈, 곱셈 등의 연산 수행
    - 연산이 정의된 집합의 성질을 알아야 암호의 동작 이해 및 분석 가능
      - e.g., 곱셈 암호는 키가 Z_{26}에서 곱셈에 대한 역원을 가져야 복호화 가능
  - 종류
    - 군(Group), 환(Ring), 체(Field)



암호 수학

- 대수 구조(Algebraic Structure) (2/26)
  - 군(Group) (1/15)
    - 정의
      - 네 가지 성질(닫힘, 결합 법칙, 항등원의 존재성, 역원의 존재성)을 만족하는 *이항 연산 "•"이 정의된 원소들의 집합

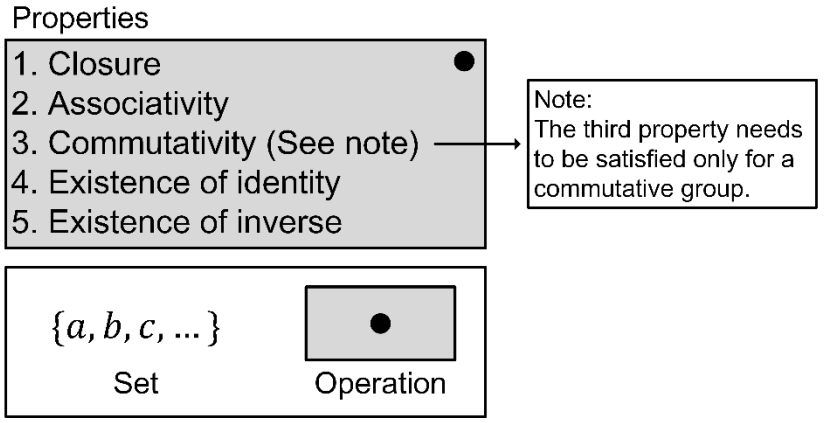
*이항 연산(Binary Operation)
두 개의 입력 값에 대하여 하나의 결과 값을 산출하는 연산



    - 표기
      - $G = < \{\ldots\}, \bullet >$
    - 분류
      - 가환 군(Commutative Group)
        - 네 가지 성질에 교환 법칙을 추가로 만족하는 군
        - 아벨 군(Abelian Group)이라고도 함
      - 비가환 군(Non-commutative Group)
        - 교환 법칙을 만족하지 않는 군
          - e.g., 치환 군은 합성 순서에 따라 결과가 달라짐

암호 수학

- 대수 구조(Algebraic Structure) (3/26)
  - 군(Group) (2/15)
    - 성질
      - 닫힘(Closure)
        - $a, b \in G$이면 $a \bullet b \in G$ (연산 결과가 집합 내의 원소)
      - 결합 법칙(Associativity)
        - $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
      - 교환 법칙(Commutativity)
        - $a \bullet b = b \bullet a$ (가환 군에서만 필요)



      - 항등원의 존재성(Existence of Identity)
        - 모든 $a \in G$에 대해 $e \bullet a = a \bullet e = a$를 만족하는 항등원 e 존재
      - 역원의 존재성(Existence of Inverse)
        - 모든 $a \in G$에 대해 $a \bullet a' = a' \bullet a = e$를 만족하는 역원 a' 존재

암호 수학

- 대수 구조(Algebraic Structure) (4/26)
  - 군(Group) (3/15)
    - 특징
      - 역원이 존재하므로 새로운 원소 추가 없이 역연산도 사용 가능
        - e.g., 덧셈이 정의된 군은 뺄셈, 곱셈이 정의된 군은 나눗셈 사용 가능
      - 덧셈/뺄셈 또는 곱셈/나눗셈 중 한 쌍만 사용 가능
        - 두 쌍을 모두 사용하려면 체(Field) 필요
      - 원소가 숫자가 아니어도, 성질을 만족하는 연산이 정의되면 군
        - e.g., 문자의 집합, 치환의 집합
    - 예제 4.1

덧셈 연산이 정의된 잉여류(Residue Integers)의 집합 $G =< Z_n, + >$가 가환 군인지 확인하고, 항등원과 원소 a의 역원을 구하시오.

- 풀이
  - 닫힘 - 두 원소의 합을 mod n 연산하면 결과도 Z_n의 원소
  - 결합 법칙
    - $(a + b) + c \equiv a + (b + c) \ (mod\ n)$
  - 교환 법칙 - $a + b \equiv b + a \ (mod\ n)$
  - 항등원 - $a + 0 \equiv a$이므로 0
  - 역원 - $a + (n - a) \equiv 0 \ (mod\ n)$이므로 $n - a$

$\therefore$ 다섯 가지 성질을 모두 만족하는 가환 군이며 항등원은 0, a의 역원은 $n - a$

암호 수학

- 대수 구조(Algebraic Structure) (5/26)
  - 군(Group) (4/15)
    - 예제 4.2

덧셈 연산이 정의된 잉여류(Residue Integers)의 집합 $G =< Z_n, + >$가 가환 군인지 확인하고, 항등원과 원소 a의 역원을 구하시오.

- 풀이
  - Z_{10}^*는 10과 서로소인 원소의 집합 $\{1, 3, 7, 9\}$
  - 닫힘
    - 오른쪽 표의 모든 결과가 Z_{10}^*의 원소
  - 결합·교환 법칙
    - 정수 곱셈의 성질이 $mod\ 10$에서도 유지
  - 항등원
    - $a \times 1 \equiv a$이므로 1
  - 역원
    - $a \times b \equiv 1\ (mod\ 10)$인 b를 확장 유클리드 알고리즘으로 계산

×	1	3	7	9
1	1	3	7	9
3	3	9	1	7
7	7	1	9	3
8	9	7	3	1

$\therefore$ 가환 군이며 항등원은 1, 역원 쌍은 $(1,1),(3,7),(7,3),(9,9)$

- 대수 구조(Algebraic Structure) (6/26)
  - 군(Group) (5/15)
    - 예제 4.3

덧셈 연산이 정의된 잉여류(Residue Integers)의 집합 $G =< Z_n, + >$가 가환 군인지 확인하고, 항등원과 원소 a의 역원을 구하시오.

- 풀이
  - 닫힘
    - 표의 모든 결과가 $\{a, b, c, d\}$의 원소
  - 결합 법칙
    - a, b, c, d를 $0, 1, 2, 3$에 대응시키면 $< Z_4, + >$와 같은 표이므로 만족
  - 교환 법칙
    - 표가 대각선 기준으로 대칭이므로 만족
  - 항등원
    - a 행과 a 열의 결과가 원래 원소와 같으므로 a
  - 역원
    - 결과가 a인 칸을 찾으면 $a \bullet a, b \bullet d, c \bullet c$

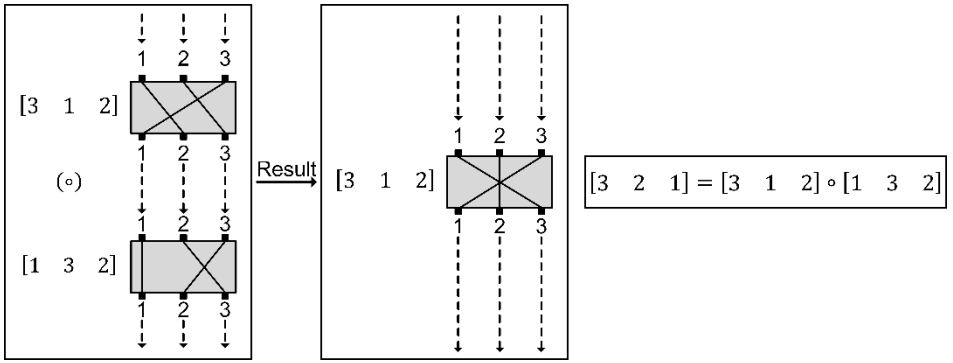
$\bullet$	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

$\therefore$ 가환 군이며 항등원은 a, 역원 쌍은 $(a, a), (b, d), (c, c)$

암호 수학

- 대수 구조(Algebraic Structure) (7/26)
  - 군(Group) (6/15)
    - 치환 군(Permutation Group) (1/2)
      - 정의
        - 치환들의 집합과 두 치환의 *합성 연산 “∘”으로 이루어진 군
      - 특징
        - 입력 값 3개에 대해 $3! = 6$개의 치환 존재
        - 합성은 오른쪽에서 왼쪽 순서로 수행
          - e.g., $[3\ 1\ 2] \circ [1\ 3\ 2]$는 $[1\ 3\ 2]$ 수행 후 $[3\ 1\ 2]$를 수행하여 $[3\ 2\ 1]$
        - 교환 법칙을 만족하지 않는 비가환 군
        - 치환을 연속으로 수행해도 같은 결과를 내는 단일 치환이 존재
          - 암호의 안전성이 강화되지 않음을 의미

*합성(Composition)
하나의 치환을 수행한 뒤 다음 치환을 수행하는 연산



암호 수학

- 대수 구조(Algebraic Structure) (8/26)
  - 군(Group) (7/17)
    - 치환 군(Permutation Group) (2/2)
      - 예시
        - 세 개의 입력 값에 대한 6개 치환의 집합과 합성 연산 ∘ (아래 표)
        - 닫힘, 결합 법칙, 항등원 [1 2 3], 역원을 만족하므로 군
          - e.g., [2 3 1]과 [3 1 2]는 서로 역원
        - 교환 법칙을 만족하지 않으므로 비가환 군
          - e.g., [1 3 2] ∘ [2 1 3] = [3 1 2], [2 1 3] ∘ [1 3 2] = [2 3 1]

∘	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 2 3]	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 3 2]	[1 3 2]	[1 2 3]	[2 3 1]	[2 1 3]	[3 2 1]	[3 1 2]
[2 1 3]	[2 1 3]	[3 1 2]	[1 2 3]	[3 2 1]	[1 3 2]	[2 3 1]
[2 3 1]	[2 3 1]	[3 2 1]	[1 3 2]	[3 1 2]	[1 2 3]	[2 1 3]
[3 1 2]	[3 1 2]	[2 1 3]	[3 2 1]	[1 2 3]	[2 3 1]	[1 3 2]
[3 2 1]	[3 2 1]	[2 3 1]	[3 1 2]	[1 3 2]	[2 1 3]	[1 2 3]

암호 수학

- 대수 구조(Algebraic Structure) (9/26)
  - 군(Group) (8/15)
    - 유한 군(Finite Group)
      - 유한 개의 원소를 갖는 군
      - e.g., $G =< Z_n, + >$, $G =< Z_n^*, \times >$
    - 무한 군(Infinite Group)
      - 무한 개의 원소를 갖는 군
      - e.g., $G =< Z, + >$
    - 군의 위수(Order of a Group)
      - 군에 있는 원소의 개수
      - $|G|$로 표기
      - 유한 군의 위수는 유한, 무한 군의 위수는 무한
      - e.g., $G =< Z_6, + >$의 위수는 $|G| = 6$

암호 수학

- 대수 구조(Algebraic Structure) (10/26)
  - 군(Group) (9/15)
    - 부분군(Subgroup)
      - 정의
        - 군 $G =< S, \bullet >$에서 S의 공집합이 아닌 부분집합 T에 대해 $H =< T, \bullet >$가 같은 연산으로 군이 될 때, H를 G의 부분군이라 함
      - 성질
        - a, b가 두 군의 원소이면 $a \bullet b$도 두 군의 원소임
        - 군과 부분군은 같은 항등원을 가짐
        - a가 두 군의 원소이면 a의 역원도 두 군의 원소임
        - G의 항등원만으로 만든 $H =< \{e\}, \bullet >$는 G의 부분군임
        - 모든 군은 자기 자신의 부분군임
      - 예시
        - Z_{10}은 Z_{12}의 부분집합이지만, $H =< Z_{10}, + >$는 $G =< Z_{12}, + >$의 부분군이 아님
        - H는 모듈로 10 덧셈, G는 모듈로 12 덧셈으로 연산이 다르기 때문임
          - e.g., $7 + 5$의 결과가 H에서는 2, G에서는 0

암호 수학

- 대수 구조(Algebraic Structure) (11/26)
  - 군(Group) (10/15)
    - 순환 부분군(Cyclic Subgroup) (1/3)
      - 정의
        - 군의 한 원소의 *멱승으로 생성되는 부분군
      - 표기
        - 원소 a로 생성된 순환 부분군은 $\langle a \rangle$
        - $a^n = a \bullet a \bullet \cdots \bullet a$ (n번), $a^0 = e$
      - 특징
        - 중복된 원소는 하나로 취급하며, 원소가 반복되면 생성 중지
        - 덧셈 군에서 a^n은 a에 n을 곱한 값
      - 예시
        - $G =< Z_{10}{}^*, \times >$의 순환 부분군 H_1, H_2, H_3

*멱승(Power)
원소에 군의 연산을 반복적으로 적용한 것

$H_1 =< \{1\}, \times >$	$H_2 =< \{1,9\}, \times >$	$H_3 =< \{1,3,7,9\}, \times >= G$
$1^0 \bmod 10 = 1$	$9^0 \bmod 10 = 1$	$3^0 \bmod 10 = 1$
⋮	$9^1 \bmod 10 = 9$	$3^1 \bmod 10 = 3$
	⋮	⋮

암호 수학

- 대수 구조(Algebraic Structure) (12/26)
  - 군(Group) (11/15)
    - 순환 부분군(Cyclic Subgroup) (2/3)
      - 예제 4.4

군 $G =< Z_6, + >$로부터 생성되는 순환 부분군을 모두 구하시오.

- 풀이
  - 각 원소의 멱승을 모듈로 6 덧셈으로 계산
    - 덧셈 군이므로 $a^n = n \times a \bmod 6$
  - 원소가 반복되면 계산 중지
  - 4와 5는 새로운 부분군을 생성하지 않음

$\therefore H_1, H_2, H_3, H_4$

원소	멱승 계산 $(a^0, a^1, \ldots)$	생성된 순환 부분군
0	0	$H_1 =< \{0\}, + >$
1	0,1,2,3,4,5	$H_4 = G$
2	0,2,4	$H_2 =< \{0,2,4\}, + >$
3	0,3	$H_3 =< \{0,3\}, + >$
4	0,4,2	H_2
5	0,5,4,3,2,1	$H_4 = G$

- 대수 구조(Algebraic Structure) (13/26)
  - 군(Group) (12/15)
    - 순환 부분군(Cyclic Subgroup) (3/3)
      - 예제 4.5

군 $G =< Z_{10}^{*}, \times >$로부터 생성되는 순환 부분군을 모두 구하시오.

- 풀이
  - $Z_{10}^{*} = \{1, 3, 7, 9\}$
  - 각 원소의 멱승을 모듈로 10 곱셈으로 계산
  - 원소가 반복되면 계산 중지
  - 7은 새로운 부분군을 생성하지 않음

  $\therefore H_1, H_2, H_3$

원소	멱승 계산 $(a^0, a^1, \ldots)$	생성된 순환 부분군
1	1	$H_1 =< \{1\}, \times >$
3	1, 3, 9, 7	$H_3 = G$
7	1, 7, 9, 3	$H_3 = G$
9	1, 9	$H_2 =< \{1, 9\}, \times >$

암호 수학

- 대수 구조(Algebraic Structure) (14/26)
  - 군(Group) (13/15)
    - 순환 군(Cyclic Group)
      - 정의
        - 그 자신이 하나의 순환 부분군인 군
      - 특징
        - 군 전체를 생성하는 원소를 *생성원이라 함
        - 생성원이 g인 유한 순환 군의 원소는 $\{e, g, g^2, \ldots, g^{n-1}\}$이며, $g^n = e$
        - 하나의 순환 군은 여러 개의 생성원을 가질 수 있음
          - e.g., $G =< Z_6, + >$의 생성원은 1과 5, $G =< Z_{10}^*, \times >$의 생성원은 3과 7
      - 예시
      - $G =< Z_3, + >$

*생성원(Generator)
순환 군에서 멱승으로 군 전체를 생성하는 원소

$g = 1$	$g = 2$
$g^0 \bmod 3 = e = 0$	$g^0 \bmod 3 = e = 0$
$g^1 \bmod 3 = 1$	$g^1 \bmod 3 = 2$
⋮	⋮

암호 수학

- 대수 구조(Algebraic Structure) (15/26)
  - 군(Group) (14/15)
    - 라그랑지 정리(Lagrange's Theorem)
      - 정리
        - G가 군이고 H가 G의 부분군이면, $|H|$는 $|G|$를 나눔
          - e.g., $G =< Z_6, + >$의 위수는 6이고, 순환 부분군들의 위수는 1, 3, 2, 6으로 모두 6의 약수
      - 활용
        - 군의 위수만으로 가능한 부분군의 위수 결정 가능
          - e.g., $G =< Z_{17}, + >$의 위수 17의 약수는 1과 17뿐이므로, 부분군은 $H_1 =< \{0\}, + >$와 $H_2 = G$뿐

- 대수 구조(Algebraic Structure) (16/26)
  - 군(Group) (15/15)
    - 원소의 위수(Order of an Element)
      - 정의
        - 군에서 $a^n = e$를 만족하는 가장 작은 양의 정수 n
      - 특징
        - $ord(a)$
        - 원소 a가 생성하는 순환 부분군의 위수와 같음
        - 라그랑지 정리에 의해 $ord(a)$는 $|G|$의 약수
      - 예제 4.6

다음 군에서 각 원소의 위수를 구하시오.

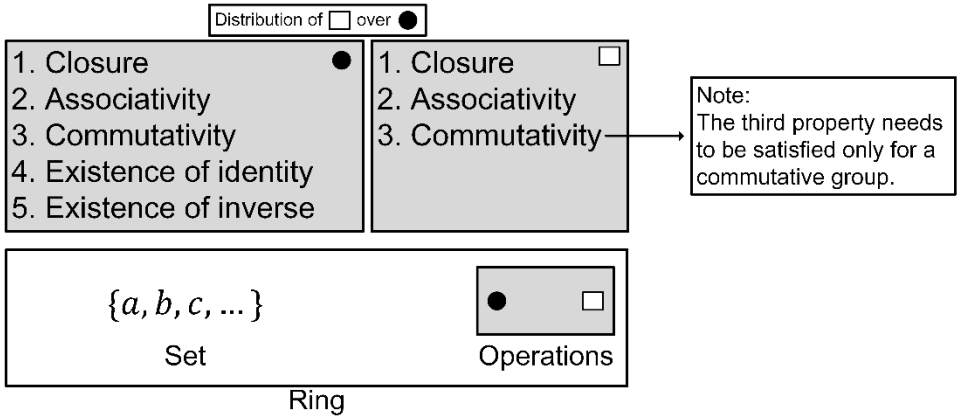
1. $G =< Z_6, + >$
2. $G =< Z_{10}^*, \times >$

- 풀이
  - 각 원소가 생성하는 순환 부분군의 원소 개수 확인
  - (1) $ord(0) = 1, ord(1) = 6, ord(2) = 3, ord(3) = 2, ord(4) = 3, ord(5) = 6$
  - (2) $ord(1) = 1, ord(3) = 4, ord(7) = 4, ord(9) = 2$

  $\therefore$ 모든 원소의 위수가 $|G|$의 약수 (라그랑지 정리)

암호 수학

- 대수 구조(Algebraic Structure) (17/26)
  - 환(Ring) (1/3)
    - 정의
      - 두 개의 연산이 정의된 대수 구조
    - 표기
      - $R = < \{...\}, \bullet, \square >$
    - 성질
      - 첫 번째 연산 •
        - 가환 군의 다섯 가지 성질을 모두 만족
      - 두 번째 연산 □
        - 닫힘, 결합 법칙만 만족
        - 교환 법칙은 가환 환일 때만 만족
      - 분배 법칙(Distributivity)
        - □는 • 위에서 분배 법칙 만족
        - $a \square (b \bullet c) = (a \square b) \bullet (a \square c)$
        - $(a \bullet b) \square c = (a \square c) \bullet (b \square c)$



암호 수학

- 대수 구조(Algebraic Structure) (18/26)
  - 환(Ring) (2/3)
    - 분류
      - 가환 환(Commutative Ring)
        - 두 번째 연산에 대해 교환 법칙을 추가로 만족하는 환
          - e.g., $R =< Z, +, \times >$
      - 비가환 환(Non-commutative Ring)
        - 두 번째 연산에 대해 교환 법칙을 만족하지 않는 환
          - e.g., 2×2 정수 행렬의 집합과 행렬 덧셈·곱셈 ($A \times B \neq B \times A$인 경우 존재)
    - 특징
      - 두 번째 연산은 항등원과 역원이 없을 가능성 존재
      - 덧셈/뺄셈 쌍과 곱셈은 사용 가능하지만, 나눗셈은 사용 불가

- 대수 구조(Algebraic Structure) (19/26)
  - 환(Ring) (3/3)
    - 예시
      - $R =< Z, +, \times >$는 가환 환
        - 덧셈은 가환 군의 다섯 가지 성질을 모두 만족 (항등원 0, a의 역원 $-a$)
        - 곱셈은 닫힘, 결합 법칙, 교환 법칙을 만족하지만 곱셈 역원은 없음
          - e.g., 2의 곱셈 역원 $1/2 \notin Z$
        - 곱셈은 덧셈 위에서 분배 법칙 만족
          - e.g., $5 \times (3 + 2) = (5 \times 3) + (5 \times 2) = 25$
        - 나눗셈은 결과가 집합 밖의 원소를 만들 수 있어 사용 불가
          - e.g., $12 \div 5 = 2.4 \notin Z$

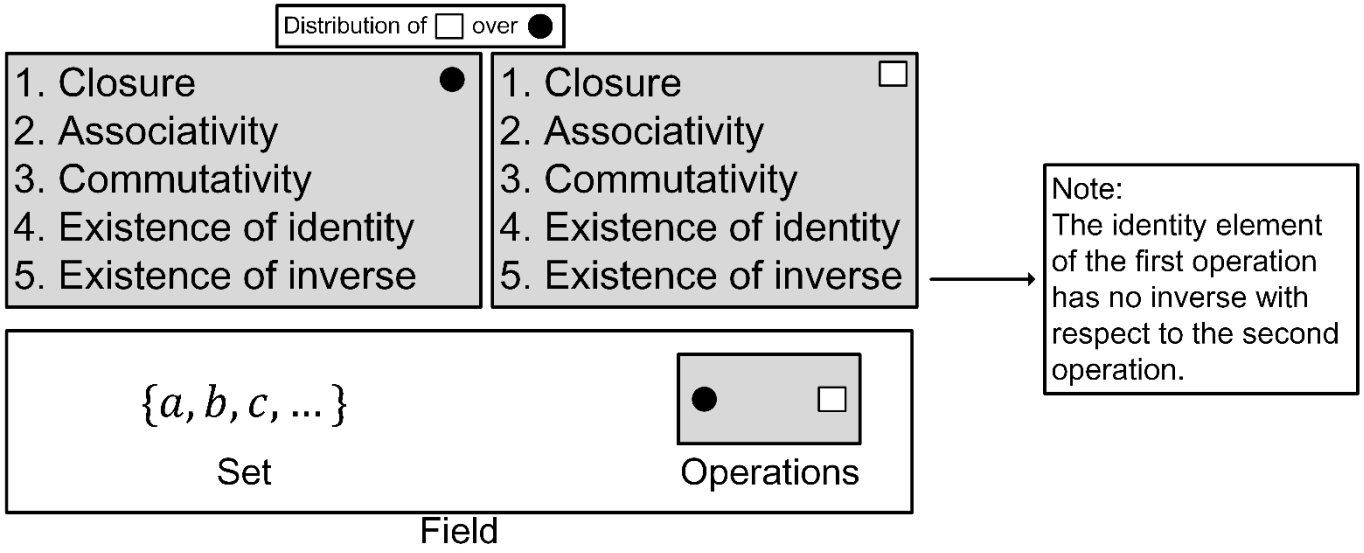
암호 수학

- 대수 구조(Algebraic Structure) (20/26)
  - 체(Field) (1/6)

*0원소(Zero Element)
첫 번째 연산의 항등원으로, 덧셈에서는 0

    - 정의
      - 두 번째 연산에서 첫 번째 연산의 항등원(*0원소)을 제외한 모든 원소가 역원을 갖는 가환 환
    - 표기
      - $F =< \{...\}, \bullet, \square >$
    - 특징
      - 덧셈/뺄셈, 곱셈/나눗셈 두 쌍을 모두 사용 가능
    - 성질
      - 첫 번째 연산
        - 가환 군의 다섯 가지 성질을 모두 만족
      - 두 번째 연산
        - 0원소의 역원이 없는 것을 제외하고 다섯 가지 성질을 모두 만족
      - 분배 법칙
        - 두 번째 연산은 첫 번째 연산 위에서 분배 법칙 만족

- 대수 구조(Algebraic Structure) (21/26)
  - 체(Field) (2/6)
    - 체의 구조

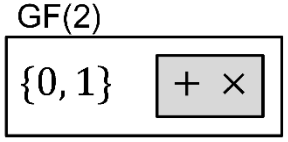


암호 수학

- 대수 구조(Algebraic Structure) (22/26)
  - 체(Field) (3/6)
    - 유한 체(Finite Field)
      - 정의
        - 유한 개의 원소를 갖는 체
      - 표기
        - $GF(p^n)$
      - 특징
        - 갈로아 체(Galois Field)라고 하며, 유한 체의 원소 개수가 p^n개를 가짐
          - p는 소수, n은 양의 정수
        - 컴퓨터는 유한한 비트로 값을 표현하므로, 유한 체는 암호에서 매우 중요한 구조임
        - p^n개의 원소를 가지지 않을 경우 원소들의 곱셈 역원을 보장할 수 없어 체의 성질을 만족할 수 없음
          - e.g., $F =< Z_5, +, \times >$은 $\{0,1,2,3,4\}$의 5개 원소를 가지고 두 연산이 체의 성질을 만족
          - e.g., $F =< Z_{10}, +, \times >$은 $\{0,1,2,3,4,5,6,7,8,9\}$의 10개 원소를 가지고 $2,4,5,6,8$ 원소에 대한 곱셈 역원이 존재하지 않으므로 체라고 볼 수 없음

암호 수학

- 대수 구조(Algebraic Structure) (23/26)
  - 체(Field) (4/6)
    - $GF(p)$ 체 (1/2)
      - 정의
        - $n = 1$인 갈로아 체로, 집합 $Z_p = \{0, 1, \dots, p-1\}$과 덧셈·곱셈으로 구성
      - 특징
        - 모든 원소는 덧셈 역원을 가지며, 0이 아닌 모든 원소는 곱셈 역원을 가짐
        - p가 소수이므로 0이 아닌 모든 a에 대해 $gcd(a, p) = 1$이 성립하기 때문
        - 덧셈/뺄셈, 곱셈/나눗셈 모두 사용 가능
      - 예시
        - $GF(2)$는 집합 $\{0, 1\}$과 덧셈·곱셈으로 구성된 체로, 암호에서 가장 널리 사용
        - 덧셈은 XOR 연산, 곱셈은 AND 연산
        - 덧셈과 뺄셈은 같은 연산(XOR), 곱셈과 나눗셈은 같은 연산(AND)



+	0	1
0	0	1
1	1	0

Addition

×	0	1
0	0	1
1	1	0

Multiplication

a	0	1
$-a$	1	0

a	0	1
$-a$	1	0

Inverses

암호 수학

- 대수 구조(Algebraic Structure) (24/26)
  - 체(Field) (5/6)
    - $GF(p)$ 체 (2/2)
      - 예시
        - $GF(5)$는 Z_5 위에 덧셈과 곱셈을 정의한 체
        - 덧셈 역원 쌍은 $(0,0),(1,4),(2,3)$
        - 곱셈 역원 쌍은 $(1,1),(2,3),(4,4)$이며, 0은 곱셈 역원 없음
        - 곱셈표에서 곱이 1인 칸으로 곱셈 역원을 바로 확인 가능

GF(5)

$\{0,1,2,3,4\}$ $+$ $\times$

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Addition

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Multiplication

Additive Inverses

a	0	1	2	3	4
$-a$	0	4	3	2	1

a	0	1	2	3	4
a^{-1}	-	1	3	2	4

Multiplicative Inverses

- 대수 구조(Algebraic Structure) (25/26)
  - 체(Field) (6/6)
    - $GF(p^n)$ 체
      - 정의
        - 원소가 p^n개($n > 1$)인 유한 체
      - 특징
        - $GF(p)$와 함께 암호에서 널리 사용
        - Z, Z_n, Z_n^*, Z_p와 덧셈·곱셈으로는 $GF(p^n)$을 만들 수 없음
          - e.g., Z_4에서 2는 곱셈 역원이 없으므로($gcd(2,4) = 2$) 체가 아님
        - 새로운 집합과 새로운 연산의 정의 필요 → $GF(2^n)$ 체

암호 수학

- 대수 구조(Algebraic Structure) (26/26)
  - 세 대수 구조 비교
    - 군은 하나의 연산 쌍
    - 환은 하나의 연산 쌍과 하나의 연산
    - 체는 두 개의 연산 쌍 지원

대수 구조	지원 연산	대표 정수 집합
군(Group)	(+ −) 또는 (× ÷)	Z_n 또는 Z_n^*
환(Ring)	(+ −)와 (×)	Z
체(Field)	(+ −)와 (× ÷)	Z_p

암호 수학

- $GF(2^n)$ 체 (1/20)
  - 개요 (1/2)
    - 필요성
      - 암호에서는 덧셈, 뺄셈, 곱셈, 나눗셈이 모두 필요하므로 체 사용
      - 컴퓨터는 양의 정수를 n-비트 워드로 저장 (보통 n=8, 16, 32, 64)
        - 표현 범위는 $0 \sim 2^n - 1$이므로, 모듈로 2^n 연산
      - 2^n은 소수가 아니므로 Z_{2^n}은 체가 아님
        - e.g., Z_8에서 $2, 4, 6$은 곱셈 역원이 없음
    - 해결 방법
      - 2^n보다 작은 가장 큰 소수 p로 $GF(p)$ 사용
        - $p \sim 2^n - 1$의 값을 사용할 수 없어 비효율적
          - e.g., $n = 4$이면 $p = 13$이므로 13, 14, 15 사용 불가
      - 원소가 2^n개인 $GF(2^n)$ 체 사용
        - 모든 n-비트 워드를 원소로 사용 가능
          - e.g., $n = 3$이면 $\{000, 001, 010, 011, 100, 101, 110, 111\}$
        - 일반적인 사칙연산을 적용할 수 없으므로, 체의 성질을 만족하는 새로운 덧셈·곱셈 정의 필요

암호 수학

- $GF(2^n)$ 체 (2/20)
  - 개요 (2/2)
    - 예시
      - $GF(2^2)$는 2-비트 워드의 집합 $\{00, 01, 10, 11\}$에 덧셈·곱셈을 정의한 체
      - 각 워드는 자기 자신이 덧셈 역원
        - e.g., $10 \oplus 10 = 00$
      - 00을 제외한 모든 워드가 곱셈 역원을 가지며, 역원 쌍은 $(01, 01), (10, 11)$
      - 덧셈과 곱셈 규칙은 다항식 형태로 정의

Addition

⊕	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

Identity: 00

Multiplication

⊗	00	01	10	11
00	00	00	00	00
01	00	01	10	11
10	00	10	11	01
11	00	11	01	10

Identity: 01

암호 수학

- $GF(2^n)$ 체 (3/20)
  - 다항식(Polynomial) (1/14)
    - 다항식 표현 (1/2)
      - 정의
        - n-비트 워드를 차수 $n-1$의 다항식으로 표현한 것
      - 표기
        - $f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x^1 + a_0x^0$
        - x^i는 i번째 항(Term), a_i는 i번째 항의 계수(Coefficient)
      - 필요성
        - 비트 워드에 대한 연산 규칙을 직접 정의하는 것보다 다항식 연산으로 정의하는 것이 간단
      - 규칙
        - x의 지수는 비트의 위치 (가장 오른쪽 비트는 x^0, 가장 왼쪽 비트는 x^{n-1})
        - 계수는 비트의 값으로, 0 또는 1
        - 계수가 0인 항은 생략, 계수 1은 생략, x^0은 1로 표기

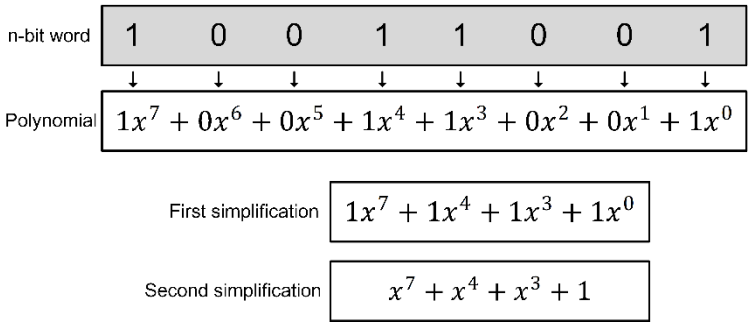
암호 수학

- $GF(2^n)$ 체 (4/20)
  - 다항식(Polynomial) (2/14)
    - 다항식 표현 (2/2)
      - 예제 4.7

8-비트 워드 10011001을 다항식으로 표현하시오.

- 풀이
  - $1x^7 + 0x^6 + 0x^5 + 1x^4 + 1x^3 + 0x^2 + 0x^1 + 1x^0$
  - 계수가 0인 항과 계수 1 생략

  $\therefore x^7 + x^4 + x^3 + 1$



      - 예제 4.8

다항식 $x^5 + x^2 + x$를 8-비트 워드로 표현하시오.

- 풀이
  - $n = 8$이므로 차수 7까지 생략된 항을 채움
    - $0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0$

  $\therefore 00100110$

암호 수학

- $GF(2^n)$ 체 (5/20)
  - 다항식(Polynomial) (3/14)
    - 연산
      - 계수에 대한 연산과 다항식에 대한 연산으로 구성되며, 두 개의 체 사용
        - 계수 연산은 $GF(2)$에서 수행 (계수는 0 또는 1)
        - 다항식 연산은 $GF(2^n)$에서 수행
  - 모듈로(Modulus) (1/2)
    - 필요성
      - 두 다항식의 덧셈 결과는 항상 차수 $n-1$ 이하
      - 두 다항식의 곱셈 결과는 차수가 $n-1$을 초과할 수 있음
      - 곱셈 결과를 모듈로 다항식으로 나눈 나머지를 사용하여 차수를 줄임

암호 수학

- $GF(2^n)$ 체 (6/20)
  - 다항식(Polynomial) (4/14)
    - 모듈로(Modulus) (2/2)
      - 기약 다항식(Irreducible Polynomial)
        - 차수가 n보다 작은 다항식들의 곱으로 분해되지 않는 차수 n의 다항식
          - e.g., $x^2 + 1 = (x + 1)(x + 1)$이므로 기약 다항식이 아님 ($GF(2)$에서 $2x = 0$)
        - 정수의 소수와 같은 역할을 하여 소수 다항식(Prime Polynomial)이라고도 함
        - 차수마다 하나 이상 존재하므로, $GF(2^n)$을 정의할 때 사용할 기약 다항식을 반드시 명시

차수	기약 다항식
1	$(x + 1), (x)$
2	$(x^2 + x + 1)$
3	$(x^3 + x^2 + 1), (x^3 + x + 1)$
4	$(x^4 + x^3 + x^2 + x + 1), (x^4 + x^3 + 1), (x^4 + x + 1)$
5	$(x^5 + x^2 + 1), (x^5 + x^3 + x^2 + x + 1), (x^5 + x^4 + x^3 + x + 1),$ $(x^5 + x^4 + x^3 + x^2 + 1), (x^5 + x^4 + x^2 + x + 1)$

암호 수학

- $GF(2^n)$ 체 (7/20)
  - 다항식(Polynomial) (5/14)
    - 덧셈 (1/2)
      - 방법
        - 같은 차수 항의 계수를 $GF(2)$에서 더함 (결과의 차수가 $n-1$ 이하이므로 모듈로 불필요)
        - 손쉬운 방법 - 공통 항은 삭제, 일치하지 않는 항은 그대로 유지
        - 비트 형태에서는 두 워드의 비트별 XOR 연산과 동일
      - 덧셈에 대한 항등원
        - 모든 계수가 0인 0 다항식
      - 덧셈에 대한 역원
        - 다항식 자기 자신이므로, 덧셈과 뺄셈은 같은 연산

암호 수학

- $GF(2^n)$ 체 (8/20)
  - 다항식(Polynomial) (6/14)
    - 덧셈 (2/2)
      - 예제 4.9

$GF(2^8)$에서 $(x^5 + x^2 + x) \oplus (x^3 + x^2 + 1)$을 계산하시오.

- 풀이
  - 생략된 항을 채운 뒤 같은 차수의 계수를 더함
    - $0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0$
    - $0x^7 + 0x^6 + 0x^5 + 0x^4 + 1x^3 + 1x^2 + 0x^1 + 1x^0$

  ∴ 공통 항 x^2은 삭제되고 나머지 항은 유지되어 $x^5 + x^3 + x + 1$

      - 예제 4.10

예제 4.17을 비트 단위 XOR 연산으로 계산하시오.

- 풀이
  - $x^5 + x^2 + x$는 00100110, $x^3 + x^2 + 1$은 00001101

  ∴ $00100110 \oplus 00001101 = 00101011$ $(x^5 + x^3 + x + 1)$

암호 수학

- $GF(2^n)$ 체 (9/20)
  - 다항식(Polynomial) (7/14)
    - 곱셈 (1/4)
      - 방법
        - 첫 번째 다항식의 각 항과 두 번째 다항식의 각 항을 곱한 결과의 합
        - 계수의 곱셈은 $GF(2)$에서 수행하며, $x^i \otimes x^j = x^{i+j}$
        - 같은 차수의 두 항은 상쇄 (e.g., $x^9 + x^9 = 0$)
        - 결과의 차수가 $n-1$보다 크면 모듈로 다항식으로 나눈 나머지 사용
      - 곱셈에 대한 항등원
        - 1 (e.g., $GF(2^8)$에서 00000001)
      - 곱셈에 대한 역원
        - 모듈로와 다항식에 확장 유클리드 알고리즘 적용 (정수의 경우와 동일)

암호 수학

- $GF(2^n)$ 체 (10/20)
  - 다항식(Polynomial) (8/14)
    - 곱셈 (2/4)
      - 예제 4.11

기약 다항식 $(x^8 + x^4 + x^3 + x + 1)$을 갖는 $GF(2^8)$에서
$(x^5 + x^2 + x) \otimes (x^7 + x^4 + x^3 + x^2 + x)$를 계산하시오.

- 풀이
  - 일반적인 다항식 곱셈 후 같은 차수의 두 항은 삭제
    - $P_1 \otimes P_2 = x^5(x^7 + x^4 + x^3 + x^2 + x) + x^2(x^7 + x^4 + x^3 + x^2 + x)$
      $+x(x^7 + x^4 + x^3 + x^2 + x)$
    - $P_1 \otimes P_2 = x^{12} + x^9 + x^8 + x^7 + x^6 + x^9 + x^6 + x^5 + x^4 + x^3 + x^8 + x^5 + x^4 + x^3 + x^2$
    - $P_1 \otimes P_2 = x^{12} + x^7 + x^2$
  - 차수 12의 결과를 차수 8의 모듈로로 나눈 나머지 사용
    - $(x^{12} + x^7 + x^2) mod\ (x^8 + x^4 + x^3 + x + 1)$
      $= x^5 + x^3 + x^2 + x + 1$

$\therefore x^5 + x^3 + x^2 + x + 1$

$$
\begin{array}{r|l}
 & x^4 + 1 \\
\hline
x^8 + x^4 + x^3 + x + 1 & x^{12} + x^7 + x^2 \\
 & x^{12} + x^8 + x^7 + x^5 + x^4 \\
\hline
 & x^8 + x^5 + x^4 + x^2 \\
 & x^8 + x^4 + x^3 + x + 1 \\
\hline
\text{Remainder} & x^5 + x^3 + x^2 + x + 1
\end{array}
$$

- $GF(2^n)$ 체 (11/20)
  - 다항식(Polynomial) (9/14)
    - 곱셈 (3/4)
      - 예제 4.12

$GF(2^4)$에서 모듈로 $(x^4 + x + 1)$에 대한 $(x^2 + 1)$의 곱셈 역원을 구하시오.

- 풀이
  - 확장 유클리드 알고리즘 사용 (r_1은 모듈로, r_2는 $x^2 + 1$, $t_1 = 0$, $t_2 = 1$로 시작)

q	r_1	r_2	r	t_1	t_2	t
$(x^2 + 1)$	$(x^4 + x + 1)$	$(x^2 + 1)$	(x)	(0)	(1)	$(x^2 + 1)$
(x)	$(x^2 + 1)$	(x)	(1)	(1)	$(x^2 + 1)$	$(x^3 + x + 1)$
(x)	(x)	(1)	(0)	$(x^2 + 1)$	$(x^3 + x + 1)$	(0)
	(1)	(0)		$(x^3 + x + 1)$	(0)	

$\therefore (x^2 + 1)^{-1} = x^3 + x + 1$

- $GF(2^n)$ 체 (12/20)
  - 다항식(Polynomial) (10/14)
    - 곱셈 (4/4)
      - 예제 4.13

$GF(2^8)$에서 모듈로 $(x^8 + x^4 + x^3 + x + 1)$에 대한 (x^5)의 곱셈 역원을 구하시오.

- 풀이
  - 확장 유클리드 알고리즘 사용
    - r_1은 모듈로, r_2는 x^5, $t_1 = 0$, $t_2 = 1$로 시작

q	r_1	r_2	r	t_1	t_2	t
(x^3)	$(x^8 + x^4 + x^3 + x + 1)$	(x^5)	$(x^4 + x^3 + x + 1)$	(0)	(1)	(x^3)
$(x + 1)$	(x^5)	$(x^4 + x^3 + x + 1)$	$(x^3 + x^2 + 1)$	(1)	(x^3)	$(x^4 + x^3 + 1)$
(x)	$(x^4 + x^3 + x + 1)$	$(x^3 + x^2 + 1)$	(1)	(x^3)	$(x^4 + x^3 + 1)$	$(x^5 + x^4 + x^3 + x)$
$(x^3 + x^2 + 1)$	$(x^3 + x^2 + 1)$	(1)	(0)	$(x^4 + x^3 + 1)$	$(x^5 + x^4 + x^3 + x)$	(0)
	(1)	(0)		$(x^5 + x^4 + x^3 + x)$	(0)	

$$\therefore (x^5)^{-1} = x^5 + x^4 + x^3 + x$$

암호 수학

- $GF(2^n)$ 체 (13/20)
  - 다항식(Polynomial) (11/14)
    - 컴퓨터를 이용한 곱셈 (1/3)
      - 필요성
        - 다항식 나눗셈은 프로그램으로 구현하기에 비효율적
        - x를 반복적으로 곱하는 방식으로 계산 (e.g., $x^2 \otimes P_2 = x \otimes (x \otimes P_2)$)
      - 동작 원리 ($x^i \otimes P_2$를 이전 결과 $x^{i-1} \otimes P_2$로부터 계산)
        1. 이전 결과의 최상위 비트가 0이면 왼쪽으로 한 비트 이동
        2. 이전 결과의 최상위 비트가 1이면 왼쪽으로 한 비트 이동한 뒤, 최상위 비트를 제외한 모듈로 다항식과 XOR
      - 장점
        - x의 곱셈은 한 비트 왼쪽 이동(Shift)으로 계산
        - 최고 차수가 $n-1$일 때만 XOR 한 번으로 축소(Reduction)
        - 차수 $n-1$인 두 다항식의 곱셈에 최대 $n-1$번 이동, $2n$번 XOR 필요

암호 수학

- $GF(2^n)$ 체 (14/20)
  - 다항식(Polynomial) (12/14)
    - 컴퓨터를 이용한 곱셈 (2/3)
      - 예제 4.14

기약 다항식 $(x^8 + x^4 + x^3 + x + 1)$을 갖는 $GF(2^8)$에서 $P_1 = (x^5 + x^2 + x)$와 $P_2 = (x^7 + x^4 + x^3 + x^2 + x)$의 곱셈을 x를 반복적으로 곱하는 방법으로 구하시오.

- 풀이
  - P_2에 $x^0 \sim x^5$를 차례로 곱한 부분 결과 계산 (이전 결과에 x를 곱함)
  - P_1의 항 x^5, x^2, x에 해당하는 부분 결과를 더함

멱승	연산	새로운 결과	축소
$x^0 \otimes P_2$		$x^7 + x^4 + x^3 + x^2 + x$	No
$x^1 \otimes P_2$	$x \otimes (x^7 + x^4 + x^3 + x^2 + x)$	$x^5 + x^2 + x + 1$	Yes
$x^2 \otimes P_2$	$x \otimes (x^5 + x^2 + x + 1)$	$x^6 + x^3 + x^2 + x$	No
$x^3 \otimes P_2$	$x \otimes (x^6 + x^3 + x^2 + x)$	$x^7 + x^4 + x^3 + x^2$	No
$x^4 \otimes P_2$	$x \otimes (x^7 + x^4 + x^3 + x^2)$	$x^5 + x + 1$	Yes
$x^5 \otimes P_2$	$x \otimes (x^5 + x + 1)$	$x^6 + x^2 + x$	No

$\therefore P_1 \otimes P_2 = (x^6 + x^2 + x) + (x^6 + x^3 + x^2 + x) + (x^5 + x^2 + x + 1) = x^5 + x^3 + x^2 + x + 1$

암호 수학

- $GF(2^n)$ 체 (15/20)
  - 다항식(Polynomial) (13/14)
    - 컴퓨터를 이용한 곱셈 (3/3)
      - 예제 4.15

예제 4.22를 8-비트 형식으로 계산하시오.

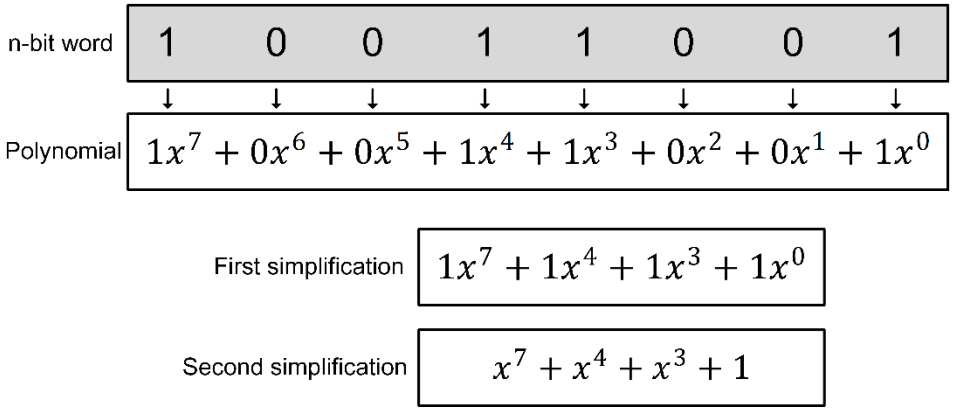
- 풀이
  - $P_1 = 00100110$, $P_2 = 10011110$, 모듈로 $= 100011011$ (9비트)
  - 축소 시 최상위 비트를 제외한 00011011과 XOR

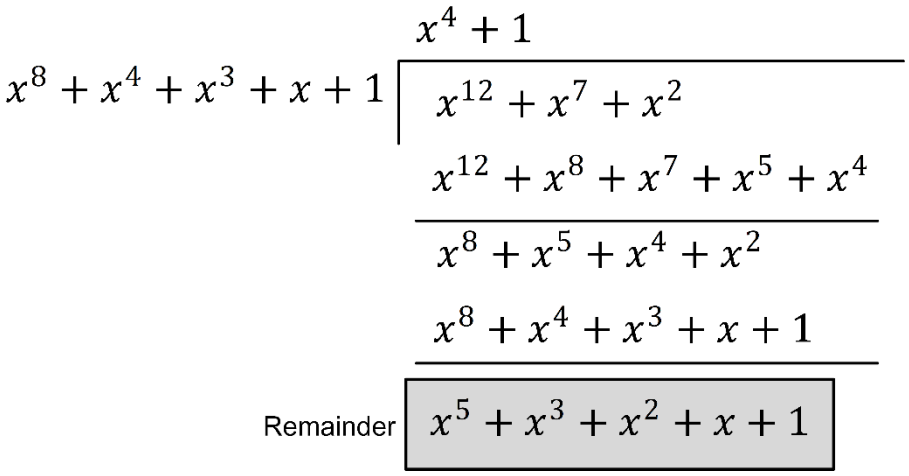
멱승	왼쪽 이동	XOR
$x^0 \otimes P_2$		10011110
$x^1 \otimes P_2$	00111100	$(00111100) \oplus (00011011) = 00100111$
$x^2 \otimes P_2$	01001110	01001110
$x^3 \otimes P_2$	10011100	10011100
$x^4 \otimes P_2$	00111000	$(00111000) \oplus (00011011) = 00100011$
$x^5 \otimes P_2$	01000110	01000110

$\therefore P_1 \otimes P_2 = (00100111) \oplus (01001110) \oplus (01000110) = 00101111$
왼쪽 이동 5번과 XOR 4번으로 계산

암호 수학

- $GF(2^n)$ 체 (16/20)
  - 다항식(Polynomial) (14/14)
    - 예시
      - 기약 다항식 $(x^3 + x^2 + 1)$을 사용하는 $GF(2^3)$ 체
        - 원소는 8개의 3-비트 워드$(000 \sim 111)$
        - 덧셈 역원은 각 원소 자기 자신 (덧셈표의 대각선이 모두 000)
        - 곱셈 역원 쌍은 $(001, 001), (010, 110), (011, 100), (101, 111)$
        - 차수 3인 기약 다항식은 $(x^3 + x^2 + 1), (x^3 + x + 1)$ 두 개이며, 모듈로에 따라 곱셈표가 달라짐





암호 수학

- $GF(2^n)$ 체 (17/20)
  - 생성원(Generator)의 사용 (1/4)
    - 정의
      - $GF(2^n)$의 원소를 생성원 g의 멱승으로 표현하는 방법
    - 특징
      - 기약 다항식 $f(x)$로 정의된 체에서 생성원 g는 $f(g) = 0$을 만족
      - 체의 원소는 $\{0, g^0, g^1, g^2, \ldots, g^N\}$이며, $N = 2^n - 2$
      - $f(g) = 0$의 관계식으로 높은 차수의 멱승을 다항식 나눗셈 없이 축소
      - 곱셈이 멱승의 덧셈이 되므로 곱셈과 역원 계산이 간단

암호 수학

- $GF(2^n)$ 체 (18/20)
  - 생성원(Generator)의 사용 (2/4)
    - 예제 4.16

기약 다항식 $f(x) = x^4 + x + 1$을 사용하여 $GF(2^4)$ 체의 원소들을 생성하시오.

- 풀이
  - $g^0 \sim g^3$은 $1, x, x^2, x^3$의 4-비트 표현이므로 그대로 사용
  - $f(g) = g^4 + g + 1 = 0$이므로 $g^4 = g + 1$
  - $g^4 \sim g^{14}$는 $g^4 = g + 1$을 반복 적용하여 $1, g, g^2, g^3$의 조합으로 축소
    - e.g., $g^{12} = g(g^{11}) = g(g^3 + g^2 + g) = g^4 + g^3 + g^2 = g^3 + g^2 + g + 1$
  - 축소된 다항식의 각 항을 비트 위치로 변환
    - e.g., $g^{14} = g^3 + 1 \rightarrow 1001$

$\therefore$ 16개 원소 생성

원소	다항식	비트	원소	다항식	비트
0	0	0000	g^7	$g^3 + g + 1$	1011
g^0	1	0001	g^8	$g^2 + 1$	0101
g^1	g	0010	g^9	$g^3 + g$	1010
g^2	g^2	0100	g^{10}	$g^2 + g + 1$	0111
g^3	g^3	1000	g^{11}	$g^3 + g^2 + g$	1110
g^4	$g + 1$	0011	g^{12}	$g^3 + g^2 + g + 1$	1111
g^5	$g^2 + g$	0110	g^{13}	$g^3 + g^2 + 1$	1101
g^6	$g^3 + g^2$	1100	g^{14}	$g^3 + 1$	1001

암호 수학

- $GF(2^n)$ 체 (19/20)
  - 생성원(Generator)의 사용 (3/4)
    - 역원
      - 덧셈에 대한 역원
        - 각 원소 자기 자신 (e.g., $-g^3 = g^3$)
      - 곱셈에 대한 역원
        - $(g^i)^{-1} = g^{-i} = g^{(2^n-1)-i}$
          - e.g., $GF(2^4)$에서 $(g^3)^{-1} = g^{-3} = g^{12} = g^3 + g^2 + g + 1 \rightarrow 1111$
            $(g^3 \times g^{12} = g^{15} = g^0 = 1)$
    - 연산
      - 덧셈과 뺄셈
        - 같은 연산이며, 같은 항 두 개는 상쇄
      - 곱셈
        - 멱승을 모듈로 $2^n - 1$로 더함
      - 나눗셈
        - 곱셈 역원과의 곱셈

암호 수학

- $GF(2^n)$ 체 (20/20)
  - 생성원(Generator)의 사용 (4/4)
    - 예제 4.17

$GF(2^4)$에서 다음 연산을 계산하시오.

1. $g^3 + g^{12} + g^7$
2. $g^3 - g^6$
3. $g^9 \times g^{11}$
4. g^3 / g^8

- 풀이
  - (1) $g^3 + (g^3 + g^2 + g + 1) + (g^3 + g + 1) = g^3 + g^2 \rightarrow 1100$
  - (2) $g^3 + g^6 = g^3 + (g^3 + g^2) = g^2 \rightarrow 0100$
  - (3) $g^{20} = g^{20\ mod\ 15} = g^5 = g^2 + g \rightarrow 0110$
  - (4) $g^3 \times g^{-8} = g^3 \times g^7 = g^{10} = g^2 + g + 1 \rightarrow 0111$

Thanks!

서 진 우(jinwoo@pel.sejong.ac.kr)