2026/04/21, 2026 보안 기초 세미나

암호학과 네트워크 보안

- 3장 고전 대칭-키 암호, 4장 암호 수학 -

김 민 식 (ms6127@naver.com)

세종대학교 프로토콜공학연구실

목 차

- 보충
- 고전 대칭-키 암호
- 암호 수학

보충

- 활용 분야 설명 (1/2)

종류	정의	활용 분야
대칭 키 암호화	암호와와 복호화를 위해 하나의 동일한 비밀키를 사용하는 암호 방식	AES 암호화 알고리즘(Advanced Encryption Standard)
비대칭 키 암호화	서로 다른 두 개의 키(공개키, 개인키)를 사용하여 암호화와 복호화를 수행하는 방식	RSA 암호화 알고리즘(Rivest–Shamir–Adleman)
해싱	데이터를 고정된 길이의 값으로 변환하는 함수	SHA-256(Secure Hash Algorithm 256)
유클리드 알고리즘	두 정수의 최대공약수(GCD)를 구하는 알고리즘	RSA 암호에서 키 생성 시 두 소수 p, q가 서로소인지 확인하는 데 활용
확장 유클리드 알고리즘	최대공약수 $\gcd(a,b)$를 구하면서 정수 s, t를 함께 계산하는 알고리즘	RSA 암호에서 개인 키 계산 시 활용, 아핀 암호, 곱셈 암호에서 복호화 키(모듈러 역원) 계산에 활용

보충

- 활용 분야 설명 (2/2)

종류	정의	활용 분야
선형 디오판투스 방정식	$ax + by = c$ 형태의 방정식으로 정수해를 구하는 방정식	RSA 암호에서 개인 키를 구할 때 선형 디오판투스 방정식 형태로 변환하여 풀이
모듈로 연산	정수를 어떤 수 n으로 나눈 나머지를 기준으로 연산하는 방법	RSA 암호에서 매우 큰 수의 거듭제곱 연산 결과를 일정 범위 내로 제한하여 암호화 및 복호화에 활용, AES에서 유한 체 연산의 기반으로 활용하여 데이터를 안전하게 변환하는 데 사용
행렬	행렬은 수들을 직사각형 형태로 배열한 것	AES에서 MixColumns 연산이 행렬 곱셈을 기반으로 수행되어 데이터 확산(Diffusion) 효과를 제공, 오류 수정 코드에서 패리티 검사 행렬을 이용한 데이터 무결성 검증에 활용
역행렬	정방 행렬 A에 대해 A와 곱했을 때 단위 행렬(I)이 되는 행렬로	AES에서 복호화 시 MixColumns 연산의 역연산에 역행렬 개념이 활용
일변수 선형 방정식	$ax \equiv b \pmod n$의 형태를 갖는 식	암호 해독 과정에서 알려진 평문 공격 시 키를 추정하는 방정식을 세우고 푸는 데 활용
일차 연립 방정식	동일한 모듈로 n에 대해 여러 개의 선형 합동식으로 이루어진 방정식	암호 해독 과정에서 여러 개의 평문/암호문 쌍을 이용하여 키를 추정하는 연립 방정식을 세우는 데 활용

목 차

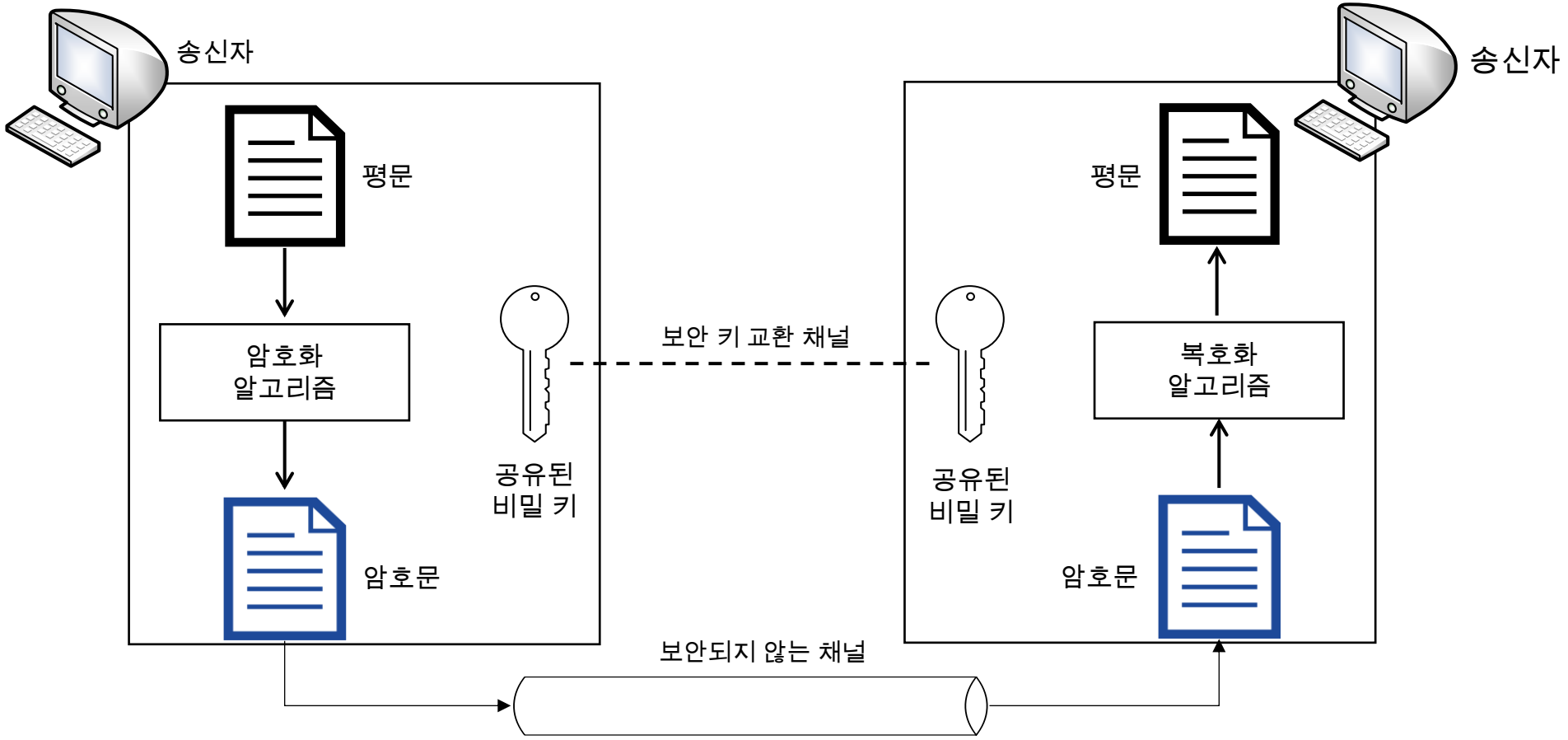
- 보충
- 고전 대칭-키 암호
- 암호 수학

개요

- 주요 용어
  - 평문(Plaintext)
    - 암호화가 적용되지 않은 원본 데이터를 의미
  - 암호문(Ciphertext)
    - 암호화 알고리즘을 통해 평문이 변환된 결과물로, 키 없이는 내용을 이해할 수 없는 형태
  - 암호 알고리즘(Encryption Algorithm)
    - 평문과 키를 입력으로 받아 암호문을 생성하는 수학적 변환 과정
  - 복호 알고리즘(Decryption Algorithm)
    - 암호화의 역과정으로, 암호문과 키를 이용해 원래의 평문을 복원하는 과정

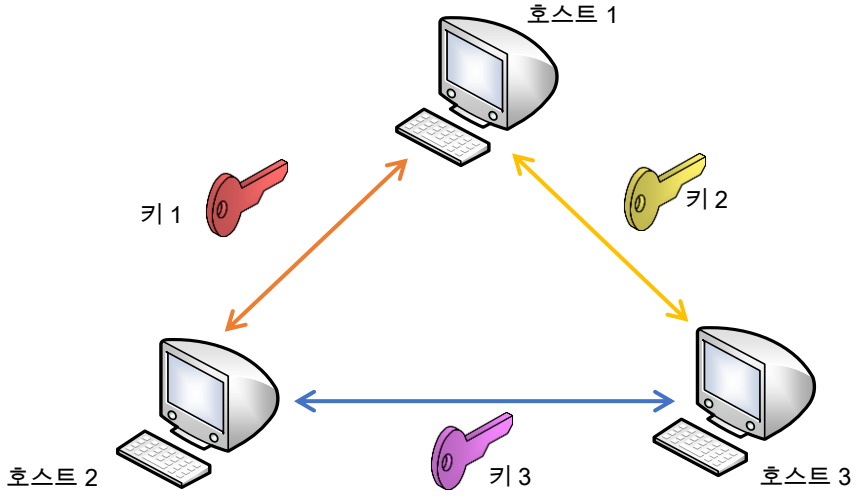
개요

- 대칭-키 암호(Symmetric-key Encipherment) (1/2)
  - 암호화와 복호화에 동일한 키를 사용하는 암호 방식



개요

- 대칭-키 암호(Symmetric-key Encipherment) (2/2)
  - 암호화 복호화 과정 모두 동일한 한 개의 키를 사용
  - 암호 알고리즘과 복호 알고리즘은 서로 역관계
    - $C = E_k(P),\ P = D_k(C)$ 이면 $D_k\big(E_k(x)\big) = E_k\big(D_k(x)\big) = x$
  - 통신을 하기위해 필요한 키는 통신 당 하나의 비밀 키 필요
    - m명의 사용자가 서로 통신을 해야 한다면 $(m \times (m-1))/2$ 개 필요
      - e.g., $m = 3$일 때 필요한 키의 개수는 3개



개요

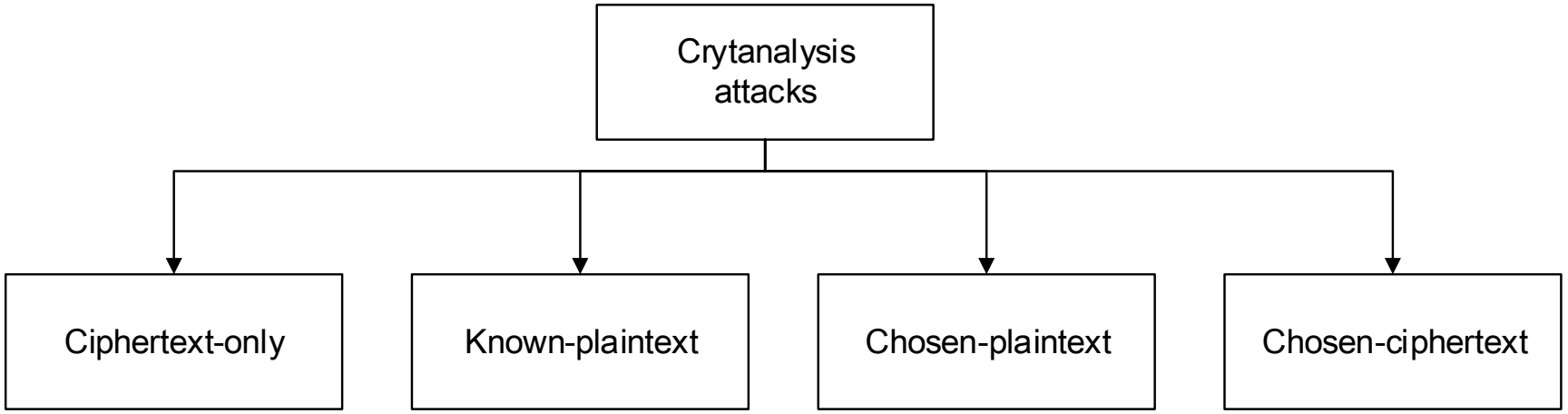
- 케르크호프스(Kerckhoff)의 원리
  - 암호 시스템의 안전성은 알고리즘이 아니라 비밀 키의 기밀성에 의존해야 함
    - 암호 알고리즘은 공개되어도 안전해야 함
    - 보안은 비밀 키의 기밀성에 의존
    - 알고리즘 은폐에 의존하지 않음
      - e.g., AES, RSA 등 현대 암호 알고리즘은 알고리즘 자체가 공개되어 있지만 비밀 키만 안전하게 유지되면 보안이 보장

개요

- 암호 해독 (1/6)
  - 암호문 또는 다양한 암호를 분석하여 비밀 키 없이 평문을 알아내거나 암호 시스템의 보안 기능을 무력화하는 기술
  - 방법
    - 전수 조사 공격(Brute-force Attack)
      - 모든 가능한 키를 하나씩 대입해보는 공격
        - e.g., PIN 번호 해킹
    - 통계적인 공격(Statistical Attack)
      - 평문의 통계적 특성(빈도 등)을 이용해 암호를 해독하는 공격
        - e.g., 영어에서 E가 가장 많이 출현한다는 통계를 이용하여 암호문에서 가장 많이 등장하는 문자를 E로 추정하는 공격패턴 공격(Pattern Attack)
      - 암호문에서 반복되는 구조나 패턴을 이용해 해독하는 공격
        - e.g., 암호문에서 반복되는 문자열을 찾아 키의 길이를 추정하는 Kasiski 테스트나, 문서 헤더 반복

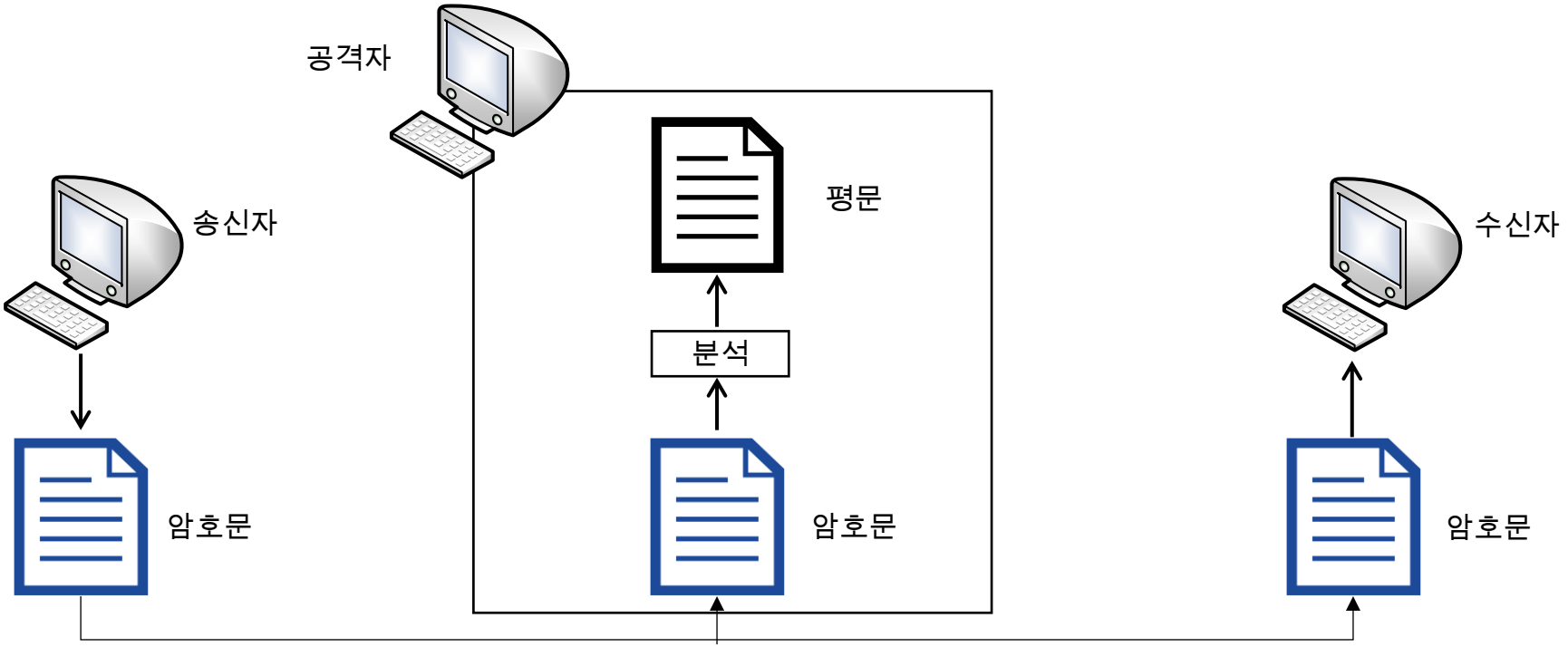
개요

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



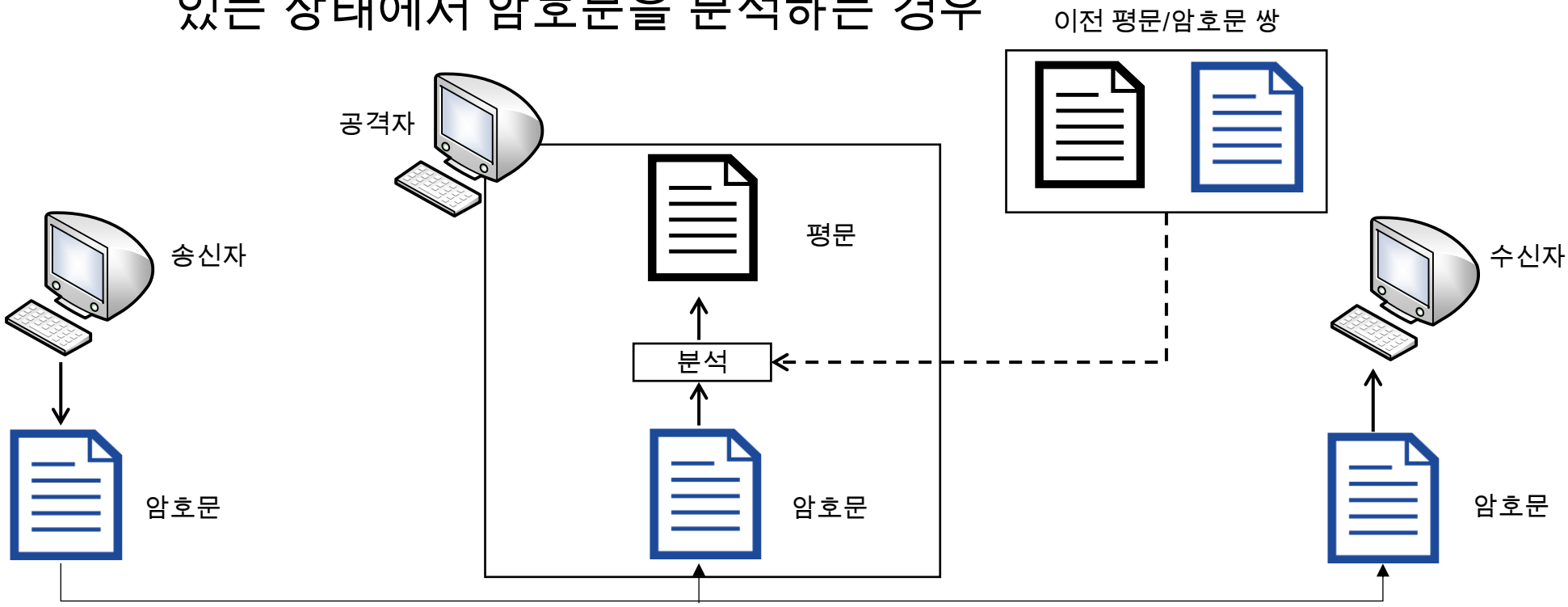
개요

- 암호 해독 (3/6)
  - 암호문 단독 공격(Ciphertext-only Attack)
    - 공격자가 암호문만 가지고 평문이나 키를 추론하는 공격
      - e.g., 빈도 분석을 통한 치환암호 해독



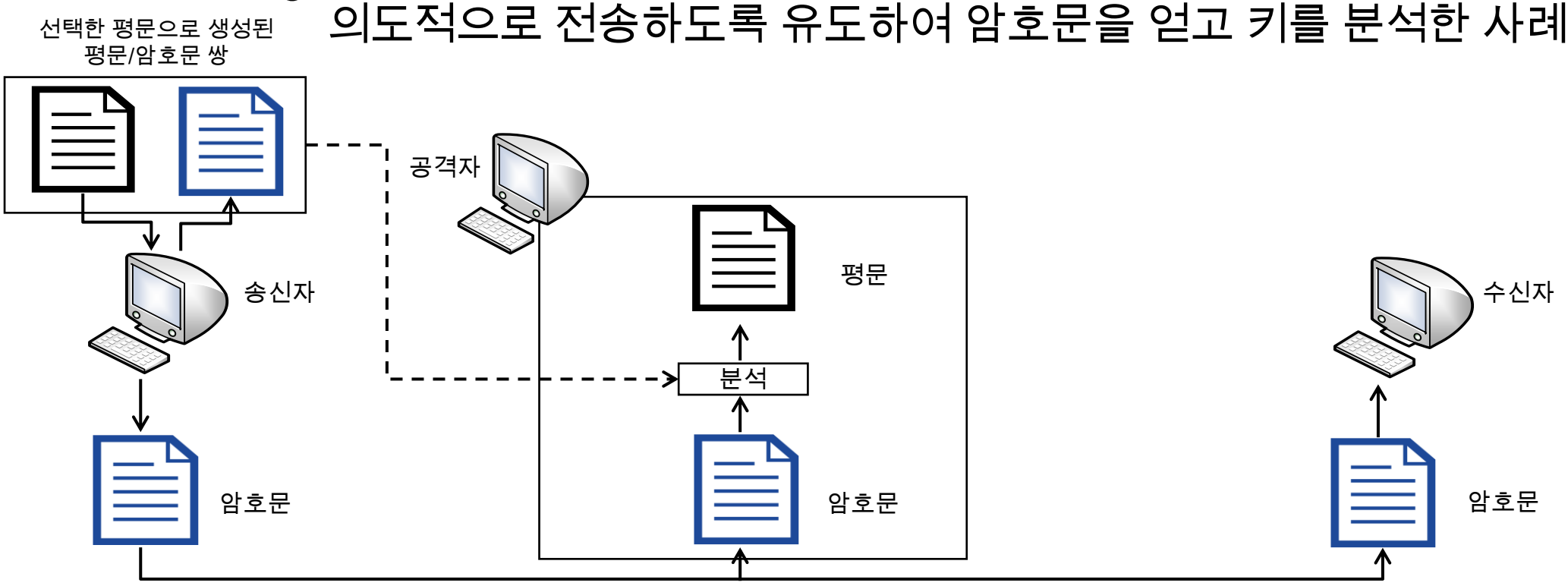
개요

- 암호 해독 (4/6)
  - 알려진 평문 공격(Known-plaintext Attack)
    - 일부 평문-암호문 쌍을 이용해 키 또는 다른 평문을 추정하는 공격
      - e.g., 공격자가 “Hello”, “Dear”와 같은 정형화된 메시지 내용을 알고 있는 상태에서 암호문을 분석하는 경우



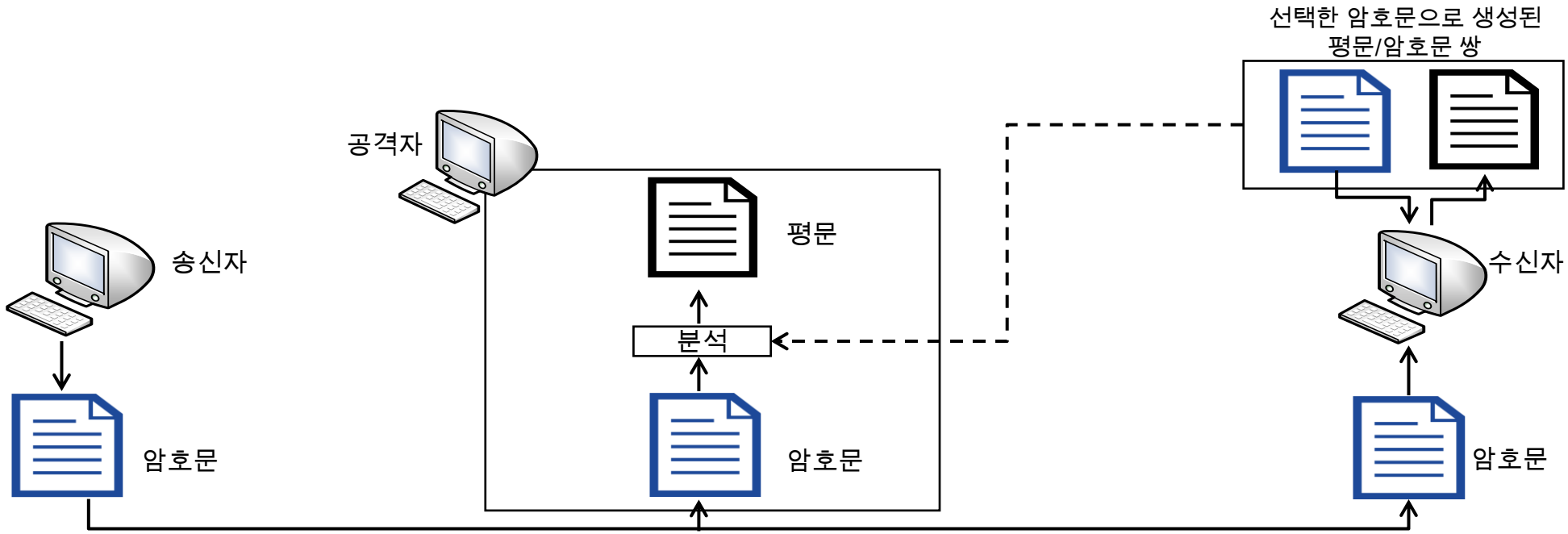
개요

- 암호 해독 (5/6)
  - 선택 평문 공격(Chosen-plaintext Attack)
    - 공격자가 임의로 선택한 평문에 대한 암호문을 얻어 이를 분석하는 공격
      - e.g., 2차 세계대전 당시 연합군이 에니그마 암호기에 특정 평문을 의도적으로 전송하도록 유도하여 암호문을 얻고 키를 분석한 사례



개요

- 암호 해독 (6/6)
  - 선택 암호문 공격(Chosen-ciphertext Attack)
    - 공격자가 원하는 암호문을 선택하여 그에 대한 복호 결과를 얻고 이를 분석하는 공격
      - e.g., 서버가 암호문을 복호화한 후 오류 메시지를 반환할 때, 공격자가 이를 이용해 평문을 추정하는 경우(패딩 오라클 공격)

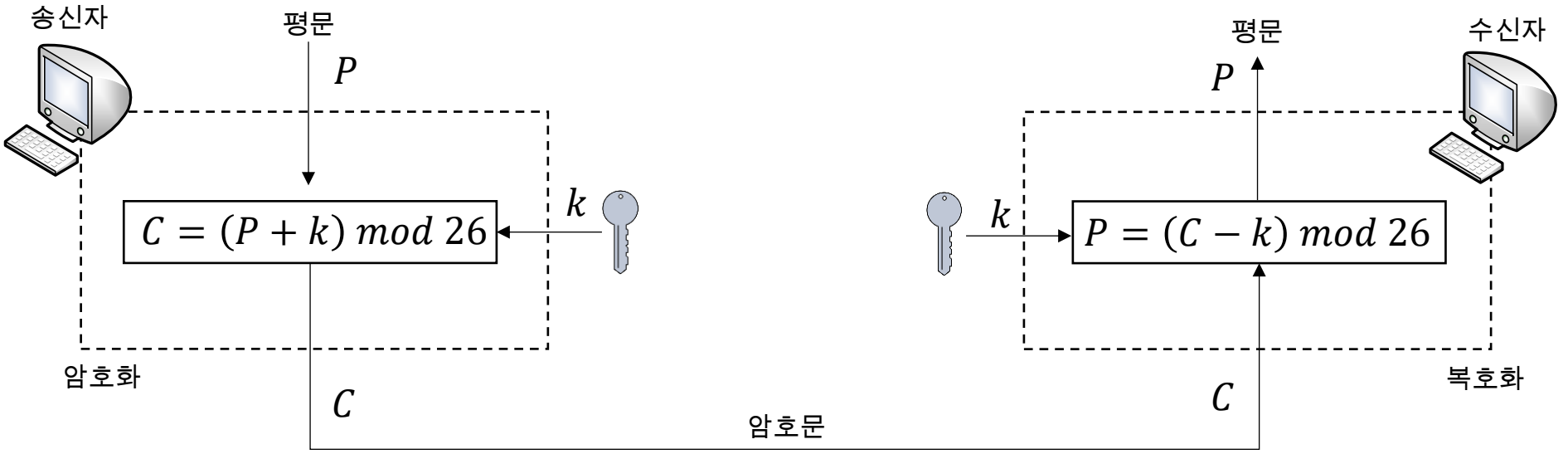


고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 평문의 각 문자를 다른 문자로 치환하여 암호문을 생성하는 암호 방식
  - 종류
    - 단일문자 암호(Monoalphabetic Ciphers)
      - 하나의 고정된 규칙을 사용하여 각 평문 문자를 하나의 암호문 문자와 일대일 대응시키는 암호 방식
    - 다중문자 암호(Polyalphabetic Ciphers)
      - 여러 개의 규칙을 사용하여 각 평문 문자를 위치나 키에 따라 서로 다른 암호문 문자와 일대다 대응시키는 암호 방식

고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (1/10)
    - 덧셈 암호(Additive Cipher)
      - 평문의 각 문자에 고정된 키 값을 더하여 암호문을 생성하고 복호화 시에는 반대로 키 값을 빼서 평문을 복원하는 방식



고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (2/10)
    - 덧셈 암호(Additive Cipher)
      - 특징
        - 이동 암호(Shift Cipher) 또는 시저 암호(Caesar Cipher)라고도 불림
        - 가능한 키의 수가 0~25로 총 26가지에 불과하여 전사 공격에 취약함
        - 단일문자 암호의 일종으로 빈도 분석 공격에도 취약함

키 15로 hello 암호화, WTAAD 복호화

$C = (P + k)\ mod\ 26, P = (C - k)\ mod\ 26$

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 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (3/10)
    - 덧셈 암호(Additive Cipher)

UVACLYFZLJBYL을 전수조사 공격을 이용하여 암호문을 어떻게 해독하는지 보여라

암호문: UVACLYFZLJBYL

		평문
$K = 1$	→	tuzbkxeykiaxk
$K = 2$	→	styajwdxjhzwj
$K = 3$	→	rsxzhvwcigyvi
$K = 4$	→	qrwygubvhfxuh
$K = 5$	→	pqvxftaugewtg
$K = 6$	→	opuwezszfdvsf
$K = 7$	→	notverysecure

$K = 7,\ P =$ "not very secure"

고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (4/10)
    - 덧셈 암호(Additive Cipher)

영어 문장에서 알파벳 출현 빈도수

문자	빈도	문자	빈도	문자	빈도	문자	빈도
E	12.7	H	6.1	W	2.4	K	0.8
T	9.1	R	6.0	F	2.2	J	0.15
A	8.2	D	4.3	G	2.0	X	0.15
O	7.5	L	4.0	Y	2.0	Q	0.1
I	7.0	C	2.8	P	1.9	Z	0.07
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

영어에서 출현 빈도수에 따라 구성된 두 문자열과 세 문자열

두 문자열 (Diagrams)	TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF
세 문자열 (Trigrams)	THE, ING, , HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH

고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (5/10)
    - 덧셈 암호(Additive Cipher)

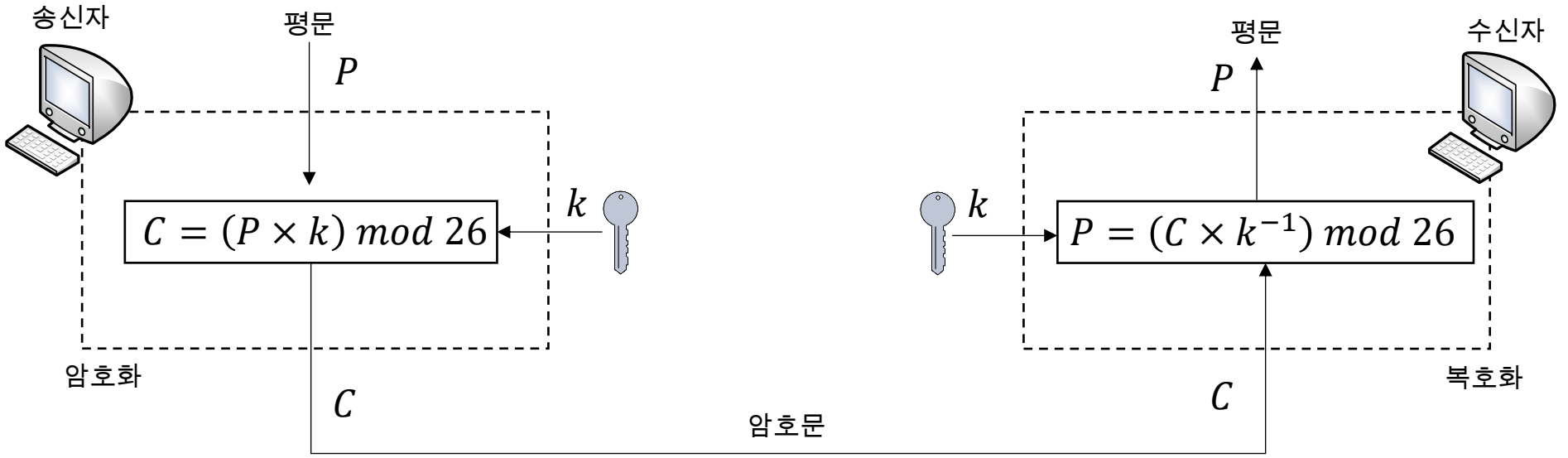
XLILSYWIGMRSAJTVSPEJSVJSYVVMPPSRHSPPEVWXMWSVSVLSSVVEBBCFI
JSVIVWIVPPIVWIGMIZIWSVVSTJJIVW 를 통계적인 공격을 이용해 평문을 구하시오

$I = 14,\ V = 13,\ S = 12$ 등이 됨
가장 큰 빈도수 14를 갖는 문자 I
I가 확률적으로 평문에서 E와 대응 따라서 $key = 4$

암호문을4로 복호화 하면
the house is now for sale for four million dollars it is
worth more hurry before the seller recieves more offers

고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (6/10)
    - 곱셈 암호(Multiplicative Ciphers)
      - 평문의 각 문자에 고정된 키 값을 곱하여 암호문을 생성하고 복호화 시에는 키의 모듈러 역원을 곱하여 평문을 복원하는 방식



고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (7/10)
    - 곱셈 암호(Multiplicative Ciphers)
      - 특징
        - 키 k는 26과 서로소인 값만 사용 가능
        - 키는 Z_{26}^*의 원소 $1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25$로 총 12가지
        - 단일문자 암호의 일종으로 빈도 분석 공격에 취약

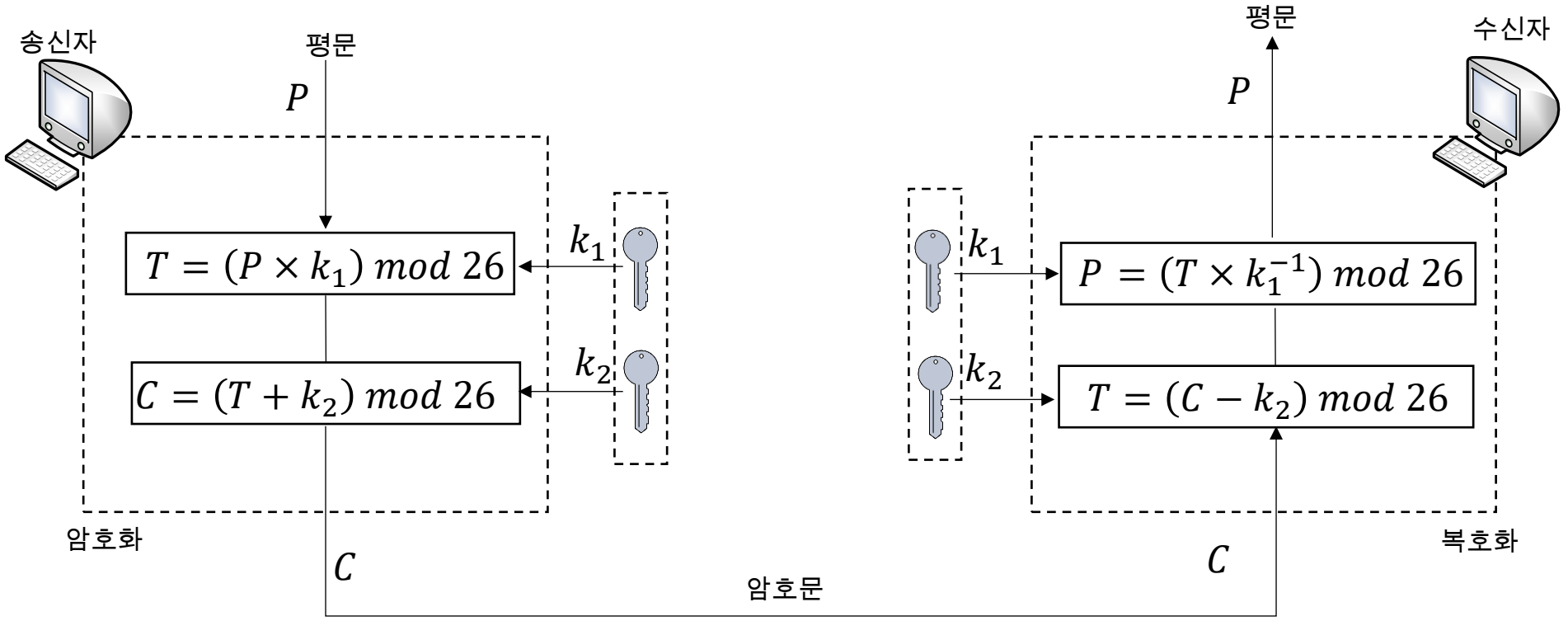
메시지 “hello”를 암호화하는데 키가 7인 곱셈 암호를 사용하시오. 암호문은 “XCZZU” 이다

$$C = (P \times k)\ mod\ 26,\ P = (C \times k^{-1})\ mod\ 26$$

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

고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (8/10)
    - 아핀 암호(Affine Cipher)
      - 덧셈 암호와 곱셈 암호를 결합한 암호 방식



고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (9/10)
    - 아핀 암호(Affine Cipher)
      - 특징
        - 첫 번째 키는 곱셈 암호에, 두 번째 키는 덧셈 암호에 사용함
        - 암호화 시 곱셈 후 덧셈 순서로 연산을 수행
        - 복호화 시 덧셈 역원을 먼저 제거한 후 곱셈 역원을 적용
        - 가능한 키 조합은 곱셈 키 12가지, 덧셈 키 26가지로 총 312가지

키 쌍이 (7,2)인 아핀 암호를 사용하여 메시지 “hello”를 암호화, “ZEBBW” 복호화 하시오

$$C = (P \times k_1 + k_2)\ mod\ 26, P = \left((C - k_2) \times k_1^{-1}\right)\ mod\ 26$$

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

고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 암호 종류 (5/10)
    - 아핀 암호(Affine Cipher)

PWUFFOGWCHFDWIWEJOUUNJORSMDWRHVCMWJUPVCCG를 선택 평문공격으로 복호화

알고리즘 1	평문: et	암호문: WC
알고리즘 2	평문: et	암호문: WF

계산 식

e → W $04 \rightarrow 22$ $(04 \times k_1 + k_2) \equiv 22 (mod\ 26)$

t → C $19 \rightarrow 02$ $(19 \times k_1 + k_2) \equiv 02 (mod\ 26)$

계산

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 22 \\ 2 \end{bmatrix} = \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix} \begin{bmatrix} 22 \\ 2 \end{bmatrix} = \begin{bmatrix} 16 \\ 10 \end{bmatrix}$$

계산 식

e → W $04 \rightarrow 22$ $(04 \times k_1 + k_2) \equiv 22 (mod\ 26)$

t → F $19 \rightarrow 05$ $(19 \times k_1 + k_2) \equiv 05 (mod\ 26)$

계산

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 22 \\ 5 \end{bmatrix} = \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix} \begin{bmatrix} 22 \\ 5 \end{bmatrix} = \begin{bmatrix} 11 \\ 4 \end{bmatrix}$$

best time of the year is spring when flowers bloom

고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 대치 암호(Monoalphabetic Substitution Ciphers) (1/2)
    - 하나의 고정된 규칙을 사용하여 각 평문 문자를 하나의 암호문 문자와 일대일 대응시키는 암호

단일문자 대치 암호에 사용되는 키의 예

평문→	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→	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

키를 사용하여 “this message is easy to encrypt but hard to find the key”를 암호화

ICFVQRVVNERFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

고전 암호의 분류

- 대치 암호(Substitution Ciphers)
  - 단일문자 대치 암호(Monoalphabetic Substitution Ciphers) (2/2)
    - 특징
      - 키는 알파벳 26개의 임의 순열로 구성되며 가능한 키의 수는 26!가지
      - 문자의 출현 빈도수를 변경시키지 않기 때문에 통계적인 공격 취약
      - 덧셈 암호, 곱셈 암호, 아핀 암호가 모두 단일문자 대치 암호의 특수한 경우에 해당

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 여러 개의 규칙을 사용하여 각 평문 문자를 위치나 키에 따라 서로 다른 암호문 문자와 일대다 대응시키는 암호 방식
  - 특징
    - 평문에서 동일한 문자는 항상 동일한 암호문 문자로 대응
    - 언어의 문자 빈도를 감추는 장점이 있어 단일 문자 빈도 해독 공격에 강함
    - 키는 부분키들로 구성된 키 수열$(k_1, k_2, k_3, \dots)$로 이루어짐
    - 평문 문자의 위치에 따라 서로 다른 부분키를 적용하여 암호문 문자를 생성

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (1/21)
    - 자동키 암호(Autokey Cipher) (1/2)
      - 키 스트림을 미리 고정하지 않고 평문 자체를 키로 활용하여 암호문을 생성하는 암호 방식

*키 스트림
암호화에 사용되는 키 값들의 연속적인 수열

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (2/21)
    - 자동키 암호(Autokey Cipher) (2/2)
      - 특징
        - 초기 키 값 하나만 사전에 공유하면 이후 키는 평문으로 자동 생성
        - 키가 평문과 같은 길이로 생성되어 키 반복 패턴이 없음
        - 평문의 일부가 노출되면 키도 함께 노출되는 취약점 존재

초기 키 값 $k_1 = 12$ 일때의 “Attack is today”의 암호화

$$P = P_1P_2P_3 \ldots, C = C_1C_2C_3 \ldots, k = (k_1, P_1, P_2, \ldots)$$
$$C_i = (P_i + k_i)\ mod\ 26, P_i = (C_i - k_i)\ mod\ 26$$

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

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (3/21)
    - 플레이페어 암호(Playfair Cipher) (1/2)
      - 평문을 두 문자씩 쌍으로 묶어 5×5 키 테이블의 규칙에 따라 각 쌍을 암호문으로 대치하는 다이그램 기반 암호 방식
        - e.g., 1차 세계대전 영국군이 전장에서 실제로 사용한 암호 방식으로, 단순하면서도 당시 기준으로 비교적 안전하여 군사 통신에 활용

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

플레이페어 비밀 키 예

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (4/21)
    - 플레이페어 암호(Playfair Cipher) (2/2)
      - 특징
        - 5×5 행렬(키 테이블)을 사용하여 암호화 수행
        - 암호화 전 두 문자가 연속하면 가짜 문자를 삽입하고, 평문 문자 개수가 홀수이면 맨 끝에 가짜 문자를 추가
        - 키 수열과 암호 수열이 동일하며, 암호문은 실제로 키 수열에 해당
        - 다중문자 암호의 요건을 만족하여 동일한 문자가 서로 다르게 암호화됨
        - 키 공간의 크기는 $25!$이고 전수조사 공격 적용하기 어려움
        - 암호화 규칙 3가지
          - 같은 행에 위치한 두 문자 → 각각 해당 행의 오른쪽 인접 문자로 대치
          - 같은 열에 위치한 두 문자 → 각각 해당 열의 아래쪽 인접 문자로 대치
          - 같은 행도 열도 아닌 경우 → 각 문자의 행에 위치하면서 상대 문자와 같은 열에 위치한 문자로 대치

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (5/21)
    - Vigenere 암호 (1/5)
      - 길이 m의 초기 비밀 키 수열을 반복하여 키 스트림을 생성하고, 평문 문자의 위치에 따라 해당 키를 적용하여 암호화하는 다중문자 암호 방식
        - 고전 암호로 현대에는 사용되지 않으며 카시스키 공격에 취약함

고전 암호의 분류

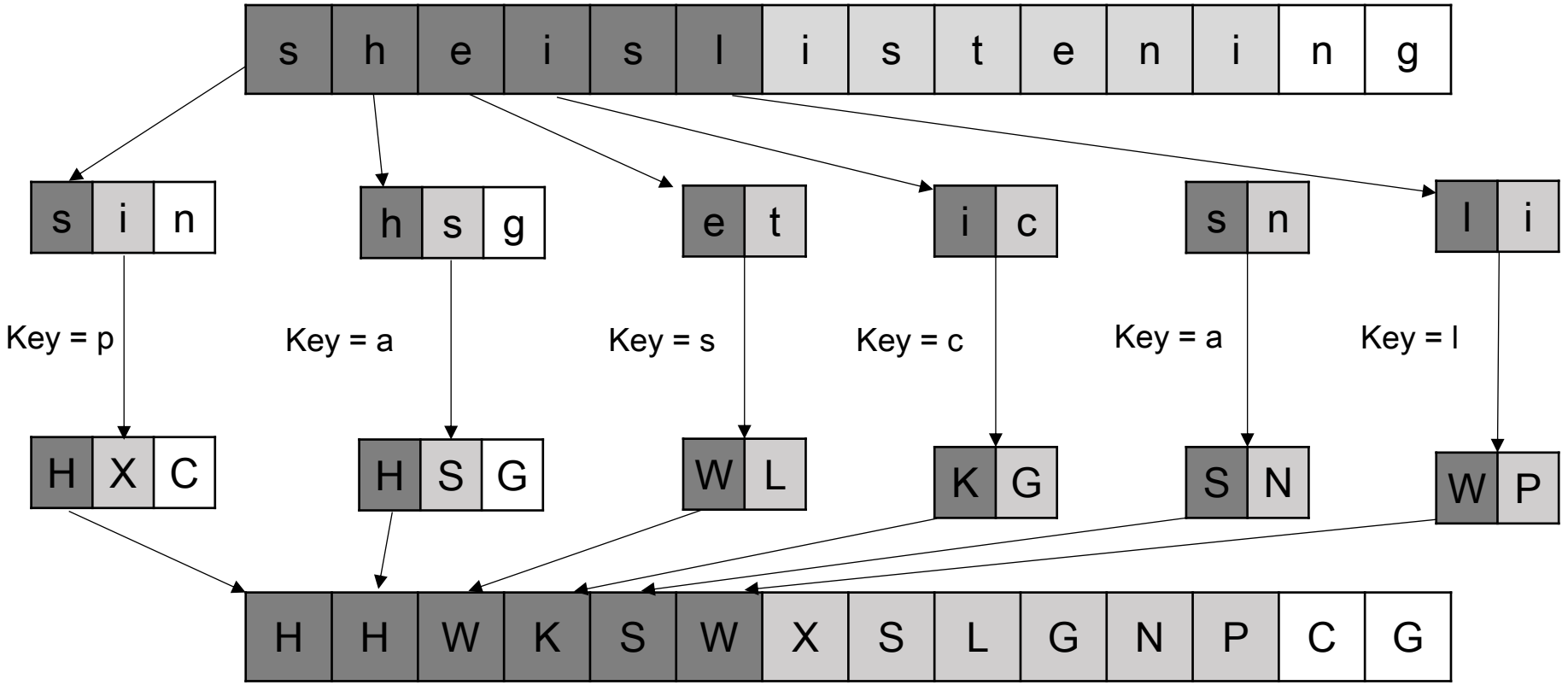
- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (6/21)
    - Vigenere 암호 (2/5)
      - 특징
        - 키 수열은 길이 $m(1 \leq m \leq 26)$인 초기 비밀 키 수열의 반복으로 구성
        - 키 수열이 평문 문자에 의존하지 않고 평문의 문자 위치에만 의존
        - 평문을 모르고도 키 수열 생성 가능
        - m개의 덧셈 암호의 조합으로 간주 가능
        - 덧셈 암호는 $m = 1$인 경우의 비즈네르 암호에 해당

"she is listening"메시지를 키 수열 $(15, 00, 18, 02, 00, 11)$로 암호화

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	E	K	S	W	X	S	L	G	N	T	C	G

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (7/21)
    - Vigenere 암호 (3/5)
      - 평문이 m개의 조각으로 나뉘어 각각 다른 키로 암호화 되는 과정



고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (8/21)
    - Vigenere 암호 (4/5)
      - 암호 해독(Kasiski 테스트)
        - 암호 해독은 키의 길이를 찾는 것과 키 자체를 찾는 두 과정으로 구성
        - 비즈네르 암호는 문자의 빈도를 보존하지 않아 단일 문자 빈도 해독이 불가능하나 Kasiski 테스트를 통해 해독 가능
      - 과정
        - 키의 길이 찾기 (Kasiski 테스트)
          1. 암호문에서 최소 3문자로 구성된 반복된 조각을 탐색
          2. 반복 조각 사이의 거리 d를 이용하여 키의 길이 m을 $d \mid m$으로 추정
          3. 반복 조각이 여럿 발견되면 $\gcd(d_1, d_2, \ldots, d_n) \mid m$으로 추정
        - 키 찾기
          1. 암호문을 m개의 조각으로 분리
          2. 각 조각에 빈도 분석 공격을 포함한 덧셈 암호 해독 방법 적용
          3. 해독된 각 조각을 조합하여 전체 평문 복원

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (9/21)
    - Vigenere 암호(5/5)
      - 암호 해독(Kasiski 테스트)

LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTHVTSGXQOVGCSVETQLTJSUMV
WVEUVLXEXEWSLGFZMVVWLGYHCUSWXQHKVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWVWJWIXGFWLT
SHGJOUEEHHVUCFVGOWICQLTJSUXGLW

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

Julius Caesar used a cryptosystem in his war,
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 cipherment

C_1 = LWGWCRAOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG
P_1 = jueuapymircneroarhtsthihytrahcieixsthcarrehe

C_2 = IGGGQHGWGKVCTSOSQSWVWFVYSHSVFSHWWFSOHCOQSL
P_2 = usssctsiswhofeaeceihcetesoecatnpntherhctecex

C_3 = OFDHURWQZKLZHGVVLUVLSZWHWKHFDUKDHVIWHUHFWLUW
P_3 = lcaerotnwhiwedssirsiirhketehretltiideatrairt

C_4 = MEVHCWILEMWVVXGETMEXLMLCXVELGMIMBWXLGEVVITX
P_4 = iardysehaisrrtcapiafpwtethecarhaesfterectpt

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (10/21)
    - Hill 암호 (1/4)
      - 평문을 같은 크기의 블록으로 나눠 $m \times m$ 정방 행렬 키를 이용한 행렬 곱셈 연산으로 블록 단위 암호화를 수행하는 다중 문자 암호 방식
        - e.g., AES의 MixColumns 연산이 힐 암호와 동일하게 행렬 곱셈 연산을 기반으로 데이터 확산 효과를 제공

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

Hill 암호의 키

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (11/21)
    - Hill 암호 (2/4)
      - 특징
        - m이 블록의 크기일 때 $m \times m$ 정방 행렬인 키를 가지며 행렬의 각 원소는 k_{ij}로 구성
        - 블록 크기를 맞추기 위해 가짜 문자(Bogus Character)를 삽입할 수 있음
        - 암호화: 평문 블록$(P_1, P_2, \ldots, P_{\mathrm{m}})$에 키 행렬 K를 곱하여 암호문 생성 $(C = PK)$
        - 복호화: 암호문 행렬에 키 행렬의 역행렬 K^{-1}을 곱하여 평문 복원 $(P = CK^{-1})$
        - 모든 정방 행렬이 역행렬을 갖는 것은 아니므로 키 선택 시 주의 필요
        - m값과 최소 m개 블록의 평문/암호문 쌍을 알고 있다면 기지 평문 공격으로 키 복구 가능

$$\overset{C}{\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1m} \\ C_{21} & C_{22} & \cdots & C_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m1} & C_{m2} & \cdots & C_{mm} \end{bmatrix}} = \overset{P}{\begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1m} \\ P_{21} & P_{22} & \cdots & P_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mm} \end{bmatrix}} \overset{K}{\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix}}$$

$$\overset{P}{\begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1m} \\ P_{21} & P_{22} & \cdots & P_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mm} \end{bmatrix}} = \overset{C}{\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1m} \\ C_{21} & C_{22} & \cdots & C_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m1} & C_{m2} & \cdots & C_{mm} \end{bmatrix}} \overset{K^{-1}}{\begin{bmatrix} k'_{11} & k'_{12} & \cdots & k'_{1m} \\ k'_{21} & k'_{22} & \cdots & k'_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ k'_{m1} & k'_{m2} & \cdots & k'_{mm} \end{bmatrix}}$$

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (12/21)
    - Hill 암호 (3/4)
      - 암호 해독
        - 전수 조사 공격
          - 키가 $m \times m$ 행렬이므로 키 공간의 크기가 $26^{(m \times m)}$으로 매우 커 전수조사 공격이 매우 어려움
          - 모든 행렬이 역행렬을 갖는 것은 아니므로 실제 키 공간은 더 작음
        - 빈도 분석 공격
          - 평문의 통계를 보존하지 않아 단일 문자, 두 문자, 세 문자 빈도 분석 적용 불가
          - 크기 m의 단어 빈도 분석은 가능하나 평문이 동일한 경우는 매우 드묾
        - 기지 평문 공격
          - Eve가 m값과 최소 m개 블록의 평문/암호문 쌍을 알고 있다면 키 복구 가능
          - $C = PK$이므로 P의 역행렬이 존재하면 $K = P^{-1}C$로 키를 구할 수 있음
          - P의 역행렬이 존재하지 않으면 추가적인 평문/암호문 쌍이 필요

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (13/21)
    - Hill 암호 (4/4)
      - 암호 해독

$m = 3$일때 평문/암호문 블록 쌍 3개로 키 행렬 구하기

$$C = P \times K, \quad K = P^{-1} \times C$$

P C

평문/암호문 쌍

$$[05 \quad 07 \quad 10] \longleftrightarrow [03 \quad 06 \quad 00]$$

$$[13 \quad 17 \quad 07] \longleftrightarrow [14 \quad 16 \quad 09]$$

$$[00 \quad 05 \quad 04] \longleftrightarrow [03 \quad 17 \quad 11]$$

K P^{-1} C

키 복구

$$\begin{bmatrix} 02 & 03 & 07 \\ 05 & 07 & 09 \\ 01 & 02 & 11 \end{bmatrix} = \begin{bmatrix} 21 & 14 & 01 \\ 00 & 08 & 25 \\ 13 & 03 & 08 \end{bmatrix} \begin{bmatrix} 03 & 06 & 00 \\ 14 & 16 & 09 \\ 03 & 17 & 11 \end{bmatrix}$$

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (14/21)
    - 일회용 패드(One-time pad)
      - 평문 문자를 키 공간에서 랜덤하게 선택된 키로 암호화하여 완벽한 보안을 제공하는 암호 방식
      - 특징
        - 키를 단 한 번만 사용하며 재사용하지 않음
        - 키는 평문과 동일한 길이의 랜덤 키 수열로 구성
        - 키가 평문과 같은 길이여야 하므로 키 배분 및 관리가 매우 어려움
        - 키를 재사용할 경우 보안성이 급격히 저하됨

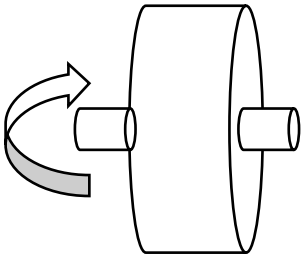
고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (15/21)
    - Rotor 암호 (1/2)
      - 회전하는 Rotor를 이용하여 평문 문자를 암호화하는 기계식 다중문자 암호 방식
      - 특징
        - 여러 개의 로터가 연결되어 각 문자를 암호화할 때마다 로터가 회전하여 서로 다른 대치 규칙을 적용
        - 첫 번째 평문 문자는 초기 설정으로 암호화되고 두 번째 문자부터 1/26 회전씩 적용하여 암호화
        - 문자 빈도를 보존하지 않아 빈도 분석 공격에 강함

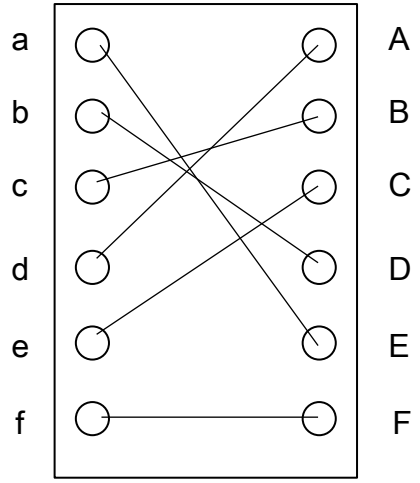
고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (16/21)
    - Rotor 암호 (2/2)

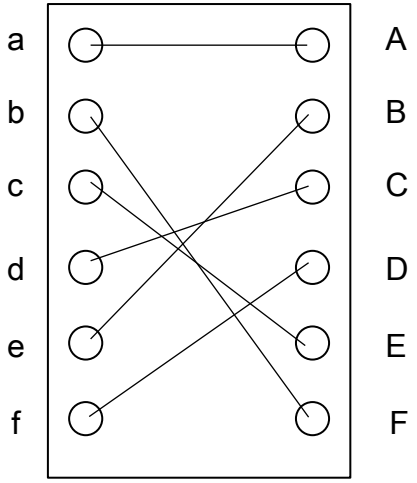
Bee와 같은 세 문자 단어는 rotor가 정지해 있으면 BAA로 암호화 되지만 rotor가 회전하면 BCA로 암호화



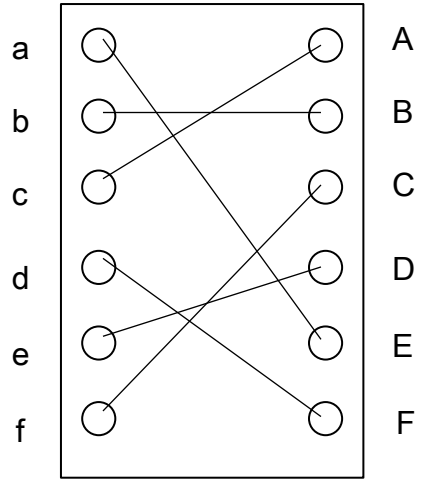
Rotor



초기 위치



첫 번째 회전



두 번째 회전

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (17/21)
    - 에니그마 기계(Enigma machine) (1/5)
      - 2차 세계대전 당시 독일군이 사용한 여러 개의 로터를 조합하여 평문 문자를 암호화하는 기계식 로터 암호 기계
      - 특징
        - 3~5개의 로터를 사용하며 로터의 조합과 초기 위치가 비밀 키로 사용
        - 각 문자를 입력할 때마다 로터가 회전하여 매번 다른 대치 규칙이 적용
        - 매우 큰 키 공간으로 수작업 해독이 사실상 불가능

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (18/21)
    - 에니그마 기계(Enigma machine) (2/5)
      - 구성 요소
        - 키보드(Keyboard)
          - 평문을 입력하는데 사용되는 26개의 키를 가짐
        - 램프보드(Lampboard)
          - 암호문 문자를 보여주는 26개의 램프를 가짐
        - 플러그보드(Plugboard)
          - 13개의 전선으로 수동으로 연결된 26개의 플러그를 가짐
          - 구성은 매일 변함
        - 사전 연결(Prewired)되어 고정된 반사경(Reflector)

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (19/21)
    - 에니그마 기계(Enigma machine) (3/5)
      - 코드북 (Code Book)
        - 에니그마 기계를 사용하기 위해 암호에 필요한 키 정보를 기록해 놓은 표로, 송신자와 수신자가 동일한 코드북을 공유하여 암호화 설정을 맞추는 방식
      - 구성 요소
        - 다섯 개의 사용 가능한 Rotor 중에 선택된 세 개의 Rotor
        - Rotor가 설치되는 순서
        - 플러그보드의 설정
        - 그날의 세 문자 코드

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (20/21)
    - 에니그마 기계(Enigma machine) (4/5)
      - 메시지 암호화 과정
        1. rotor의 시작점을 그날의 코드로 맞춤
           - e.g., "HUA"라면 각각 "H", "U", "A"로 설정함
        2. 랜덤한 세 문자 코드를 선택하고 1단계 에서의 Rotor의 초기 설정을 이용하여 문장을 암호화함
           - e.g., "ACF"를 선택 "ACFACF"를 암호화하여 "OPNABT"
        3. rotor의 시작점을 선택한 코드로 설정함
           - e.g., rotor의 시작점을"OPN"으로 설정
        4. 단계 2에서 얻은 암호화된 문자를 메시지 시작에 추가함
        5. 그 문자 코드를 포함한 메시지를 암호화 하고 암호화된 메시지를 보냄

고전 암호의 분류

- 다중문자 암호(Polyalphabetic Ciphers)
  - 다중문자 암호 종류 (21/21)
    - 에니그마 기계(Enigma machine) (5/5)
      - 메시지 복호화 과정
        1. 메시지를 받고 첫 번째 여섯 글자를 분리함
        2. Rotor의 시작점을 그날의 코드로 설정함
        3. 단계 2의 시작점을 사용하여 처음 여섯 개 글자를 복호화함
        4. 복호화된 코드의 첫 번째 반으로 Rotor의 위치를 설정함
        5. 첫 번째 여섯 글자 없이 메시지를 복호화함

전치 암호

- 전치암호(Transposition Cipher)
  - 평문의 문자 자체는 변경하지 않고 문자의 위치를 재배열하여 암호문을 생성하는 암호 방식
  - 특징
    - 문자를 다른 문자로 대치하지 않고 위치만 변경하는 방식
    - 평문의 문자 빈도가 그대로 보존되어 빈도 분석으로 전치 암호임을 식별 가능
    - 키는 문자의 재배열 규칙을 정의하며 동일한 키를 공유해야 복호화 가능
    - 빈도 분석 공격에 취약함

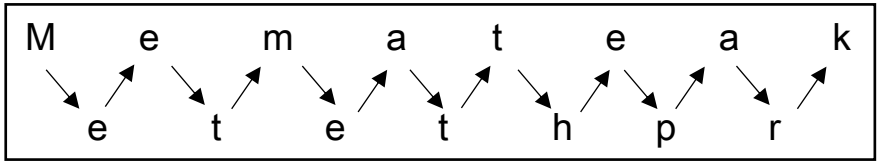
전치 암호

- 전치암호(Transposition Cipher)
  - 종류
    - 키가 없는 전치 암호
    - 키가 있는 전치 암호
    - 열 전치 암호
    - 이중 전치 암호

전치 암호

- 전치암호(Transposition Cipher)
  - 키가 없는 전치 암호
    - 미리 정해진 고정된 규칙에 따라 평문의 문자 위치를 재배열하는 방식
      - e.g., 레일 펜스 암호(Rail Fence Cipher)
  - 특징
    - 송신자와 수신자가 규칙만 공유하면 되므로 키 관리가 필요 없음
    - 규칙이 단순하여 암호 해독이 키가 있는 전치 암호보다 쉬운 편

"Meet me at the park"를 Rail Fence로 암호화



Rail Fence Cipher 예

평문을 지그재그 형태로 여러 행에 걸쳐 배열한 후 행 단위로 읽어 암호문을 생성하는 방식

전치 암호

- 전치암호(Transposition Cipher)
  - 키가 있는 전치 암호
    - 키를 이용하여 평문의 문자 재배열 순서를 결정하는 암호 방식
    - 특징
      - 평문을 블록으로 나눈 뒤 각각의 블록에서 문자를 재배열하기 위해 따로 키를 사용하는 방식
      - 키워드를 알파벳 순서로 정렬하여 열의 순서를 결정
      - 키를 모르면 재배열 순서를 알 수 없어 키가 없는 전치 암호보다 보안성이 높음

e	n	e	m	y	a	t	t	a	c	k	s	t	o	n	i	g	h	t	z

암호화↓

3	1	4	5	2
1	2	3	4	5

↑복호화

E	E	M	Y	N	T	A	A	C	T	T	K	O	N	S	H	I	T	Z	H	G

전치 암호

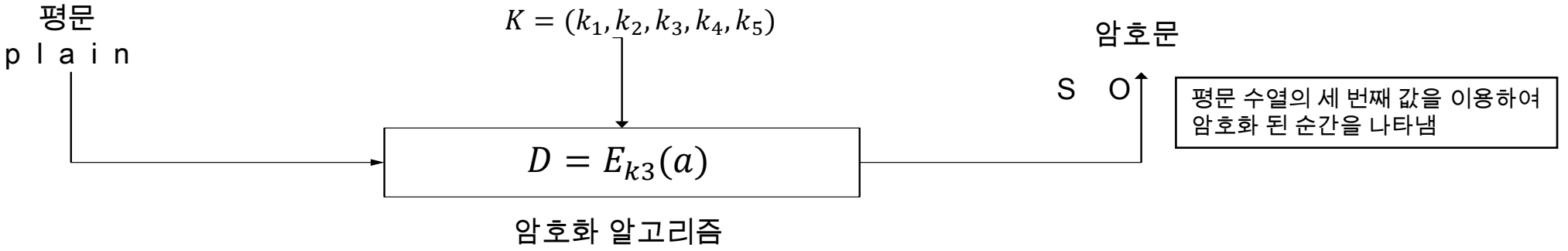
- 전치암호(Transposition Cipher)
  - 이중 전치 암호(Double Transposition Cipher)
    - 전치 암호를 두 번 반복 적용하여 보안성을 높인 암호 방식
    - 특징
      - 첫 번째 전치로 생성된 암호문을 다시 한 번 전치하여 최종 암호문을 생성
      - 단일 전치 암호에 비해 문자 배열이 더 복잡해져 해독이 어려움
      - 두 번의 전치에 동일한 키 또는 서로 다른 키를 사용할 수 있음

전치 암호

- 전치암호(Transposition Cipher)
  - 암호 해독
    - 통계적인 공격
      - 암호문에서 문자의 출현 빈도 통계를 분석하여 평문을 추론하는 공격 방식
        - e.g.,영어에서 'E'가 가장 많이 출현한다는 통계를 이용하여 암호문에서 가장 많이 등장하는 문자를 'E'로 추정
    - 전수조사 공격
      - 가능한 모든 키를 시도하여 올바른 평문을 찾아내는 공격 방식
        - e.g., 열 전치 암호에서 가능한 모든 열 순서를 시도하여 의미 있는 평문이 나올 때까지 반복
    - 패턴 공격
      - 암호문에서 반복되는 패턴이나 구조적 특징을 분석하여 평문을 추론하는 공격 방식
        - e.g., 이중 전치 암호에서 반복되는 문자 배열 패턴을 분석하여 전치 규칙을 역추론

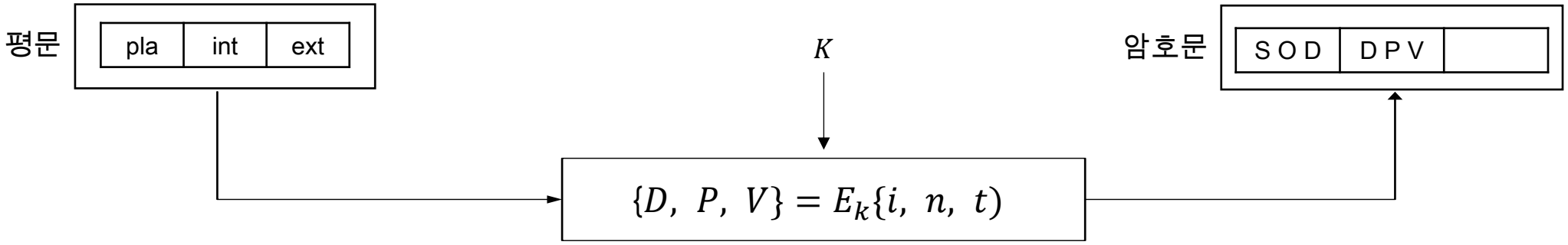
스트림 암호와 블록 암호

- 스트림 암호(Stream Cipher)
  - 이진 수열 발생기로 평문을 일련의 비트열로 취급하여 한 번에 1비트씩 암호화하는 방식
    - e.g., RC4, ChaCha20
  - 특징
    - 평문 문자는 한 번에 하나씩 암호화 알고리즘에 입력되고 암호문 문자도 한 번에 하나씩 생성함
    - 블록 암호에 비해 속도가 빠르고 구현이 간단함
    - 같은 키를 재사용하면 보안성이 급격히 저하됨



스트림 암호와 블록 암호

- 블록 암호(Block Cipher)
  - 암호문을 만들기 위해 암호 키와 알고리즘이 블록 단위의 평문에 적용되는 대칭 암호화 방식
    - e.g., AES, DES, 3DES
  - 특징
    - 평문을 고정된 크기의 블록(64비트, 128비트)으로 나누어 암호화하는 방식
    - 블록의 크기를 맞추기 위해 마지막 블록에 패딩(Padding)을 추가할 수 있음
    - 스트림 암호에 비해 속도가 느리지만 높은 보안성을 제공

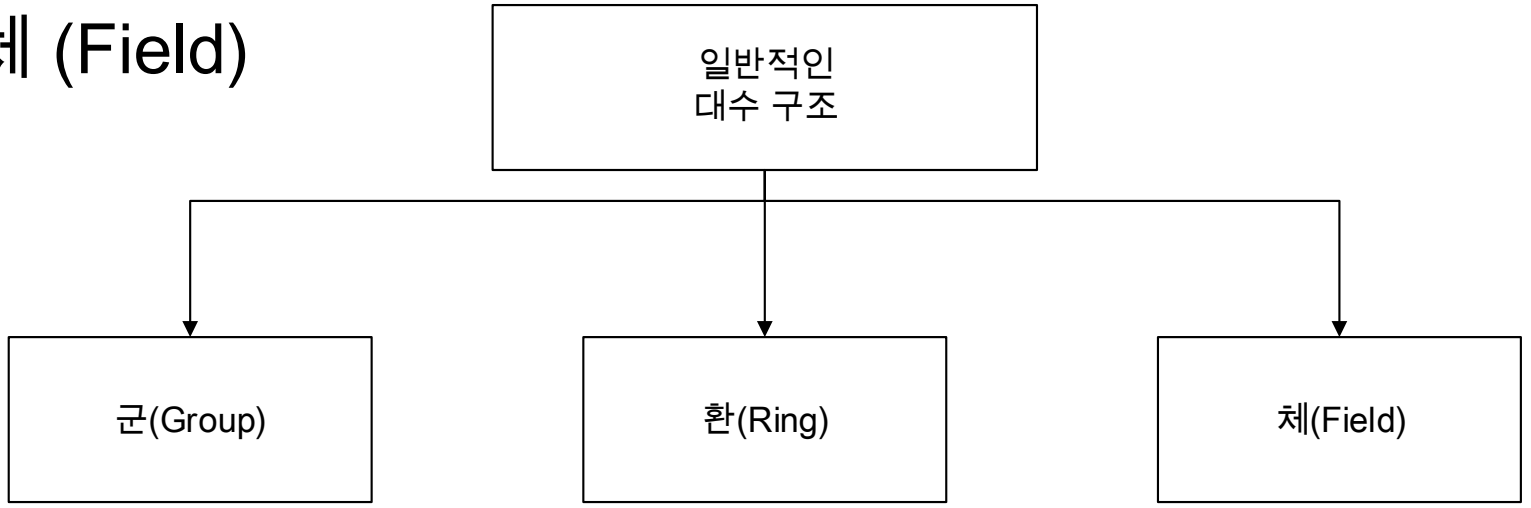


목 차

- 보충
- 고전 대칭-키 암호
- 암호 수학

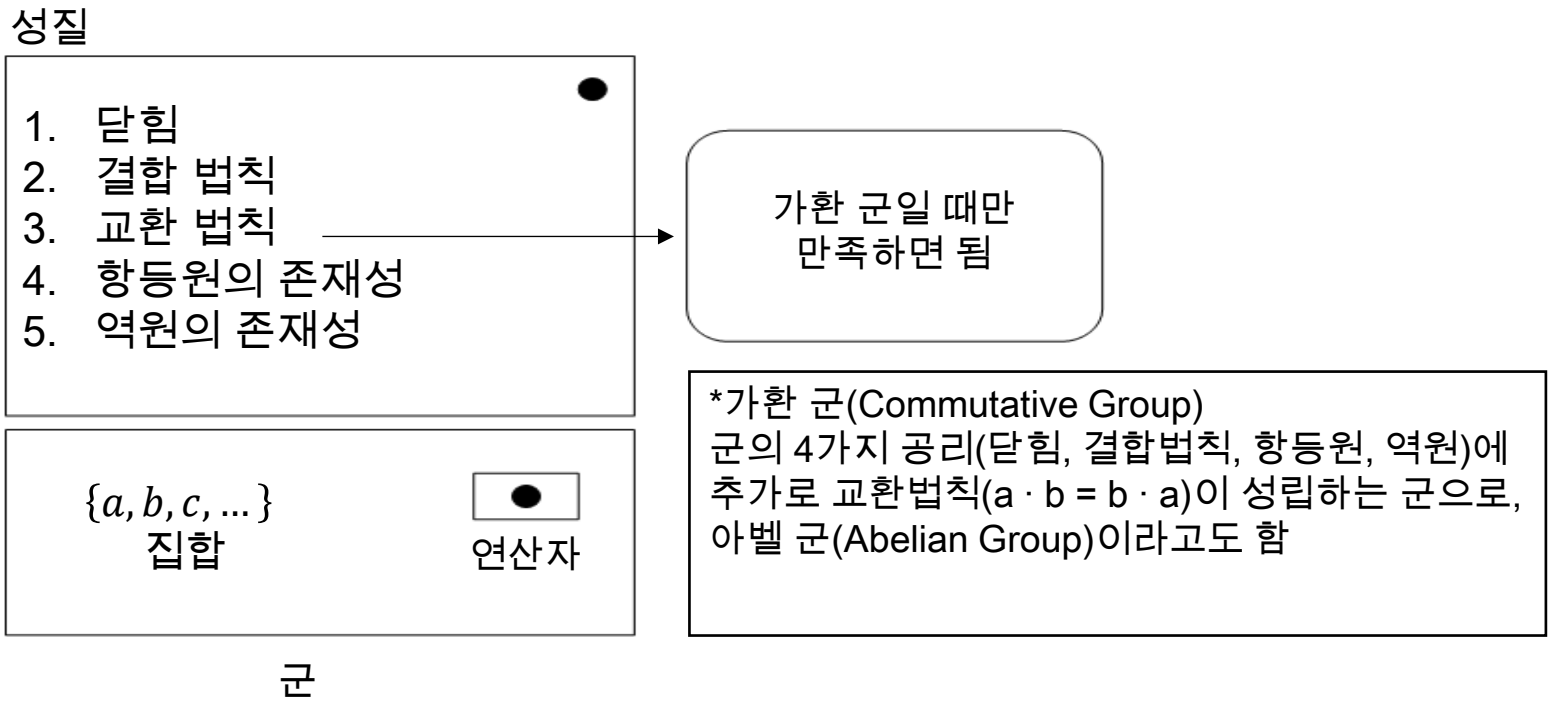
대수 구조

- 대수 구조(Algebraic Structure)
  - 집합과 그 집합 위에 정의된 연산들이 특정 규칙을 만족하는 수학적 구조
  - 종류
    - 군 (G, Group)
    - 환 (Ring)
    - 체 (Field)



대수 구조

- 대수 구조(Algebraic Structure)
  - 군(G, Groups) (1/9)
    - 하나의 이항 연산이 정의된 집합으로 닫힘, 결합법칙, 항등원 역원을 만족하는 대수 구조

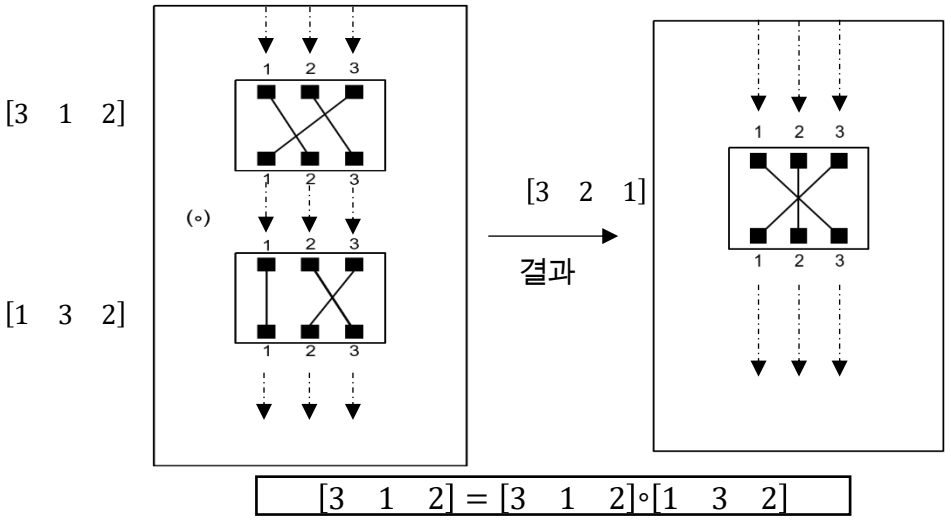


대수 구조

- 대수 구조(Algebraic Structure)
  - 군(G, Groups) (2/9)
    - 성질
      - 닫힘
        - 만약a 와 b 가 G의 원소라면 $a \bullet b$ 또한 G의 원소
      - 결합 법칙
        - 만약a 와 b 가 G의 원소라면 $(a \bullet b) \bullet c = a \bullet (b \bullet c)$를 만족
      - 교환 법칙
        - G의 임의의 두 원소 a, b 에 대하여 $a \bullet b = b \bullet a$ 를 만족
      - 항등원의 존재성
        - G의 임의의 원소 a에 대하여 $e \bullet a = a \bullet e = a$ 를 만족
      - 역원의 존재성
        - G의 임의의 원소 a에 대하여 $a \bullet a' = a' \bullet a = e$ 를 만족

대수 구조

- 대수 구조(Algebraic Structure)
  - 군(G, Groups) (3/9)
    - 치환 군(Permutation Group)
      - 집합의 원소들을 서로 다른 위치로 재배열하는 모든 치환들로 이루어진 군
      - 특징
        - n개의 원소를 가진 집합의 모든 치환으로 구성되며 원소의 수는 $n!$개
        - 모든 치환은 일대일 대응이며, 각 원소는 정확히 하나의 위치로 이동



치환 군의 연산

∘	[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]

대수 구조

- 대수 구조(Algebraic Structure)
  - 군(G, Groups) (4/9)
    - 유한 군(Finite Group)
      - 유한 개의 원소를 가지고 있는 군
    - 무한 군(Infinite Group)
      - 무한 개의 원소를 가지고 있는 군
    - 위수(Order)
      - 군의 위수 $|G|$는 군에 있는 원소의 개수

대수 구조

- 대수 구조(Algebraic Structure)
  - 군(G, Groups) (5/9)
    - 순환 부분군(Cyclic Subgroup)
      - 군의 부분군이 어떤 원소의 멱승(Power)을 사용하여 생성된 부분군
      - 특징
        - $a^n \rightarrow a \cdot a \cdot \cdots \cdot a(n번), < a >$라고 표기
        - 집합 내 중복 원소는 하나의 원소로 취급
        - $a^0 = e$
        - 모든 순환 부분군은 아벨 군(Abelian Group)에 해당

*멱승(Power)
원소에 군의 연산을 반복적으로 적용한 것

대수 구조

- 대수 구조(Algebraic Structure)
  - 군(G, Groups) (6/9)
    - 순환 부분군(Cyclic Subgroup)

$G =< Z_{10}{}^{*}, \times >$의 순환 부분군 H_1, H_2, H_3 찾는 방법

$H_1 =< \{1\}, \times >$

- $1^0 \ mod \ 10 = 1$ (이런 과정 반복)

$H_2 =< \{1,9\}, \times >$

- $9^0 \ mod \ 10 = 1$
- $9^1 \ mod \ 10 = 9$(위 원소들이 반복)

$H_3 =< \{1,3,7,9\}, \times >= G$

- $3^0 \ mod \ 10 = 1$
- $3^1 \ mod \ 10 = 3$
- $3^2 \ mod \ 10 = 9$
- $3^3 \ mod \ 10 = 7$(위 원소들이 반복)

대수 구조

- 대수 구조(Algebraic Structure)
  - 군(G, Groups) (7/9)
    - 순환 군 (Cyclic Group)
      - 그 자신이 하나의 순환 부분군인 군
      - 특징
        - 생성원 g에 대해 유한 순환 군의 원소 집합 $\{e, g^1, g^2, \dots, g^{n-1}\}$, $g^n = e$으로 표현됨
        - 유한 순환군의 경우 생성원의 차수(order)가 군의 크기와 같음
        - 모든 순환군은 아벨 군에 해당

대수 구조

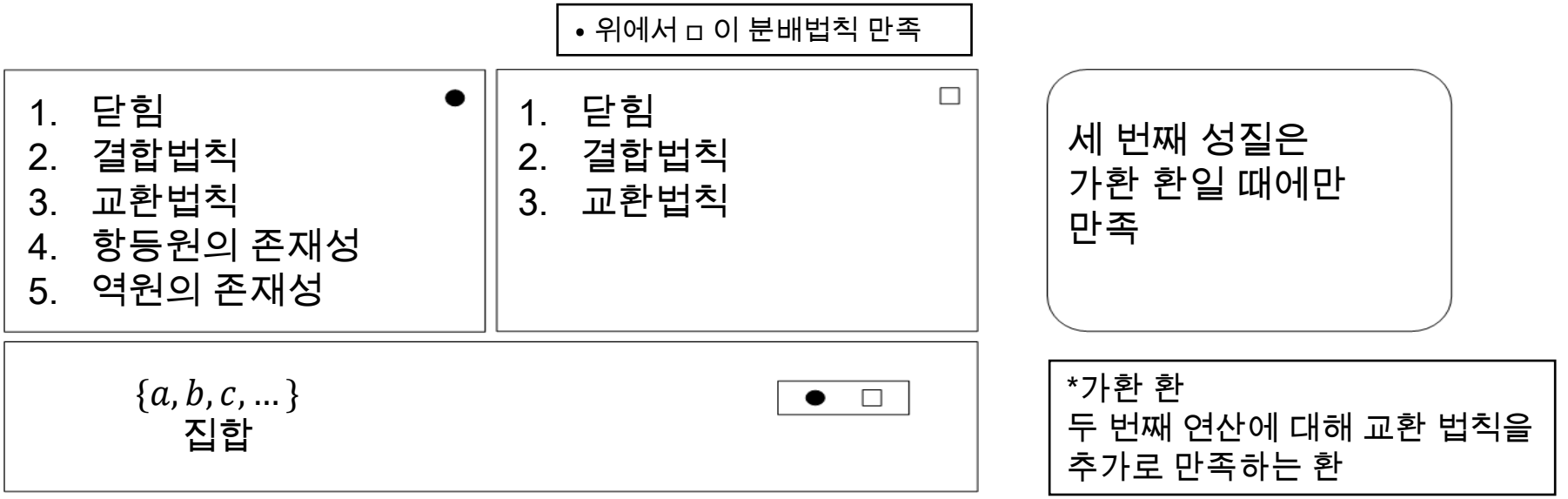
- 대수 구조(Algebraic Structure)
  - 군(G, Groups) (8/9)
    - 라그랑지 정리(Lagrange's Theorem)
      - 유한군 G의 임의의 부분군 H에 대해 H의 차수(원소의 수)는 G의 차수의 약수가 된다는 정리
      - 특징
        - 유한군 G의 부분군 H의 차수 |H|는 항상 |G|의 약수
        - $|G|$를 $|H|$로 나눈 값을 지수(Index) 라고 하며 $[G : H]$로 표기
        - 소수 차수를 가진 군은 자명한 부분군과 자기 자신 외에는 다른 부분군을 갖지 않음
        - 군의 크기가 소수이면 그 군은 반드시 순환군에 해당

대수 구조

- 대수 구조(Algebraic Structure)
  - 군(G, Groups) (9/9)
    - 원소의 위수
      - 군에서 $a^n = e$를 만족하는 가장 작은 정수 n을 원소 a의 위수라고 하며 $n = ord(a)$로 표기
        - e.g., $G =< Z_6, +>$에서 $ord(0) = 1,\ ord(1) = 6,\ ord(2) = 3,\ ord(3) = 4,\ ord(4) = 3,\ ord(5) = 6$

대수 구조

- 대수 구조(Algebraic Structure)
  - 환(Ring) (1/2)
    - 두 개의 이항 연산(덧셈과 곱셈)이 정의된 집합으로, 덧셈에 대해 아벨 군을 이루고 곱셈에 대해 결합법칙과 분배법칙을 만족하는 대수 구조
      - $R =< \{...\}, \bullet, \square >$로 표현함

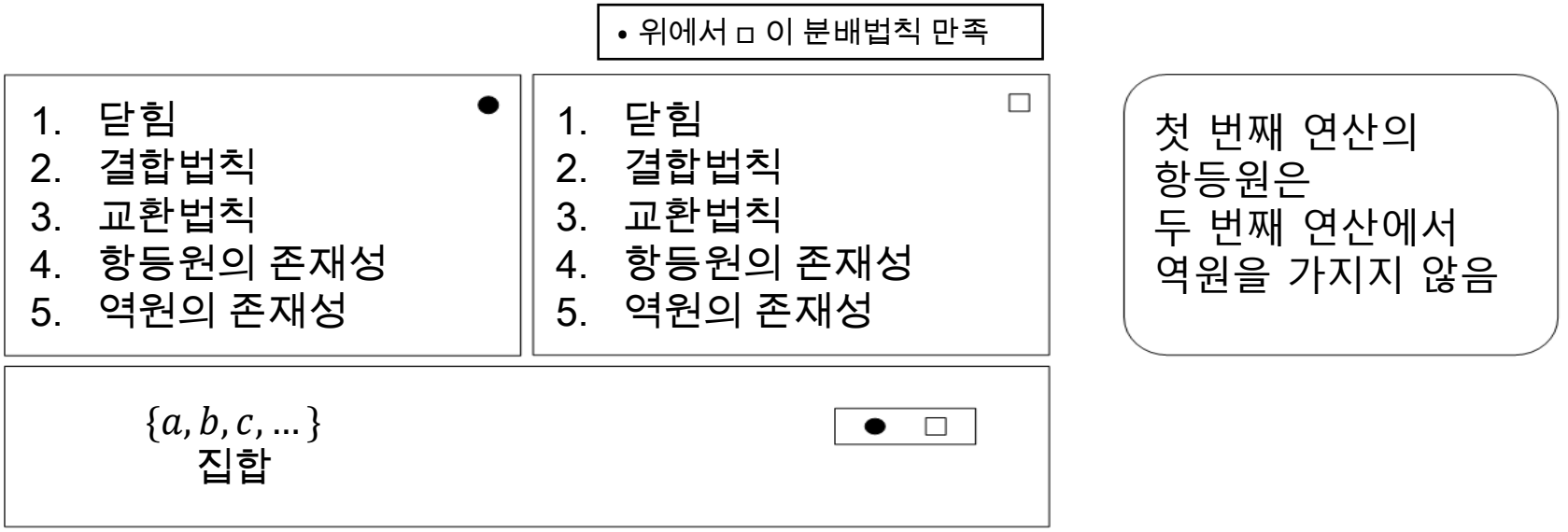


대수 구조

- 대수 구조(Algebraic Structure)
  - 환(Ring) (2/2)
    - 특징
      - 덧셈에 대해 아벨 군을 이룸 (닫힘, 결합법칙, 항등원, 역원, 교환법칙 모두 만족)
      - 곱셈에 대해 닫힘과 결합법칙을 만족하지만 항등원과 역원은 필수가 아님
      - 덧셈과 곱셈 사이에 분배법칙이 성립
      - 곱셈에 대한 교환법칙이 성립하면 교환환(Commutative Ring) 이라고 함

대수 구조

- 대수 구조(Algebraic Structure)
  - 체(Field) (1/3)
    - 두 개의 이항 연산(덧셈과 곱셈)이 정의된 집합으로, 덧셈과 곱셈 모두에 대해 아벨 군을 이루는 대수 구조
      - $F =< \{...\}, \bullet, \square >$ 로 표현함



대수 구조

- 대수 구조(Algebraic Structure)
  - 체(Field) (2/3)
    - 유한 체(Finite Field)
      - 유한 개의 원소를 갖는 체
        - 암호학에서 AES, 타원 곡선 암호(ECC), RSA 등 현대 암호의 수학적 기반으로 활용됨
      - 특징
        - 유한 체의 원소 개수는 반드시 소수 p의 거듭제곱 형태(p^n)여야 함
        - 갈루아 체(Galois Field) 라고도 하며 $GF(p^n)$으로 표기
        - 덧셈과 곱셈 모두 $mod\ p$ 연산으로 수행
        - 0을 제외한 모든 원소가 곱셈에 대한 역원을 가짐

대수 구조

- 대수 구조(Algebraic Structure)
  - 체(Field) (3/3)
    - $GF(p)$ 체
      - 소수 p에 대해 $\{0, 1, 2, \dots, p-1\}$의 p개 원소로 구성된 유한 체로, 모든 연산이 $mod\ p$로 수행되는 체
        - 암호학에서 RSA, Diffie-Hellman 키 교환 등에 활용됨
      - $GF(2)$ 체
        - 이진수 또는 비트로 표현되는 두 개의 원소만 가짐
        - 덧셈 연산은 이진수에서 사용하는 XOR 연산
        - 곱셈 연산은 이진수에서 사용하는 AND 연산
        - 덧셈과 뺄셈 연산은 같음
        - 곱셈과 나눗셈 연산은 같음

$GF(2)$

$\{0,1\}$ $+$ $\times$

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

a	0	1
$-a$	1	0

a	0	1
a^{-1}	$-$	1

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 2^n개의 원소로 구성된 유한 체로, 각 원소가 n비트 이진 다항식으로 표현되는 체
    - 암호학에서 AES, 타원 곡선 암호 등 현대 블록 암호의 핵심 수학 구조로 활용
  - 암호학에서는 사칙연산을 필요로 하므로 암호학에서 체를 주로 사용
  - 컴퓨터로 연산 시 정수는 n −비트 워드로 저장됨

$F =< \{00,01,10,11\}, \oplus, \otimes >$로 정의된 체

덧셈

$\oplus$	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

항등원 : 00

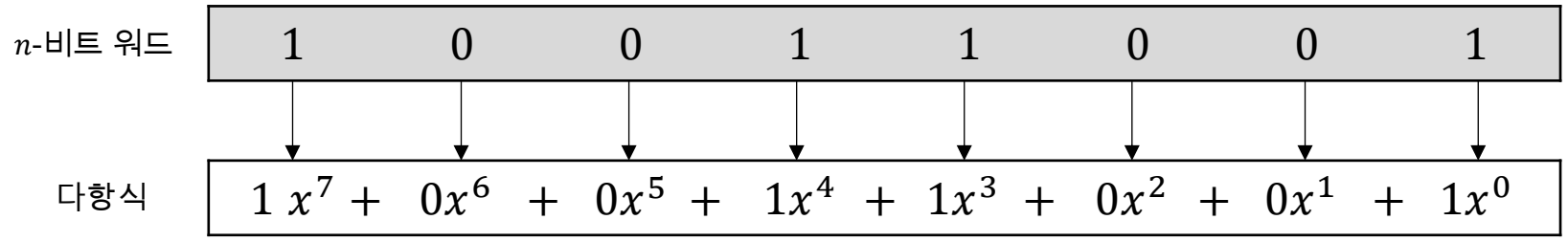
곱셈

$\otimes$	00	01	10	11
00	00	00	00	00
01	00	01	10	11
10	00	10	11	01
11	00	11	01	10

항등원 : 01

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 다항식(Polynomial) 표현 (1/7)
    - n비트 워드를 차수 $n-1$의 다항식으로 표현
    - x의 지수승은 비트의 위치를 정의
      - 가장 왼쪽은 최상위 비트, 가장 오른쪽 최하위 비트
    - 항들의 계수는 비트의 값으로 0 또는 1
    - $f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x^1 + a_0x^0$로 표현
      - e.g., 8-비트 워드(10011001)의 다항식 표현



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

계수가 0인 항은 작성하지 않음
계수가 1인 항은 계수를 생략함
$1x^0$은 1로 작성함

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 다항식(Polynomial) 표현 (2/7)
    - 연산
      - n비트 워드들을 표현하는 다항식은 두 개의 체 $GF(2)$와 $GF(2^n)$을 사용
        - 계수 연산을 위해 $GF(2)$ 체를 사용, 다항식 연산을 위해 $GF(2^n)$ 체를 사용함
    - 모듈로(Modulo)
      - 두 다항식의 곱셈은 $n-1$보다 큰 차수를 가지는 다항식을 생성할 수 있음
      - 이를 모듈로 논리에 따라 곱셈한 결과를 모듈로 다항식으로 나누고 나머지만 취하는 방식
      - $GF(2^n)$의 다항식들의 집합에 대해 차수 n의 다항식의 군은 모듈로써 정의됨

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 다항식(Polynomial) 표현 (3/7)
    - 모듈로(Modulo)
      - 기약 다항식
        - 모듈로에서 소수 다항식으로 행동하며 그 집합에 있는 어떤 다항식도 이 다항식을 나눌 수 없음
        - n보다 작은 차수를 가진 다항식으로 분해할 수 없는 다항식

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

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 다항식(Polynomial) 표현 (4/7)
    - 덧셈
      - $GF(2)$에서 계수를 갖는 다항식들에 대한 덧셈 연산
      - 덧셈은 비트 단위 XOR 연산으로 수행
      - 차수 $n-1$인 두 다항식의 합은 항상 차수 $n-1$의 다항식을 생성하여 모듈로 불필요
      - 덧셈에 대한 항등원은 0의 다항식(모든 계수가 0인 다항식)
      - 덧셈에 대한 역원은 그 원소 자신 (다항식의 덧셈과 뺄셈은 같은 연산)

$$0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0$$
$$0x^7 + 0x^6 + 0x^5 + 0x^4 + 1x^3 + 1x^2 + 0x^1 + 1x^0$$

$$0x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 0x^2 + 1x^1 + 1x^0$$

$$x^5 + x^3 + x + 1$$

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 다항식(Polynomial) 표현 (5/7)
    - 곱셈
      - 첫 번째 다항식의 각 항과 두 번째 다항식의 각 항의 곱셈의 합으로 계산함
      - 곱셈은 $n-1$보다 큰 차수를 가지는 항을 생성할 수 있어 모듈로 다항식으로 줄여야 함
      - 곱셈에 대한 항등원은 항상 1
      - 곱셈에 대한 역원은 확장 유클리드 알고리즘을 적용하여 구함
        - e.g.,기약 다항식 $(x^8+x^4+x^3+x+1)$ 가지는 $GF(2^8)$에서 $(x^5+x^2+x)\otimes(x^7+x^4+x^3+x^2+x)$의 계산

$$(x^5+x^2+x)\otimes(x^7+x^4+x^3+x^2+x) = x^5(x^7+x^4+x^3+x^2+x)+x^2(x^7+x^4+x^3+x^2+x)+x(x^7+x^4+x^3+x^2+x)$$

$$(x^5+x^2+x)\otimes(x^7+x^4+x^3+x^2+x) = x^{12}+x^9+x^8+x^7+x^6+x^5+x^4+x^3+x^8+x^5+x^4+x^3+x^2$$

$$(x^5+x^2+x)\otimes(x^7+x^4+x^3+x^2+x) = (\mathrm{x}^{12}+\mathrm{x}^7+\mathrm{x}^2)\ mod\ (x^8+x^4+x^3+x+1) = (x^5+x^3+x^2+x+1)$$

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 다항식(Polynomial) 표현 (6/7)
    - 컴퓨터 이용 곱셈
      - 나눗셈 연산 때문에 두 다항식을 곱하기 위한 프로그램 개발 시 효율성 문제가 존재
      - 이를 해결하기 위해 줄여진 다항식에 x를 반복적으로 곱하는 알고리즘을 사용
        - $(x^8 + x^4 + x^3 + x + 1)$을 기약다항식으로 갖는 $GF(2^8)$ 체에서 $P_1 = (x^5 + x^2 + x)$와 $P_2 = (x^7 + x^4 + x^3 + x^2 + x)$의 곱셈

거듭제곱	연산	새로운 결과	약분
$x^0 \otimes P_2$		$x^7 + x^4 + x^3 + x^2 + x$	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
$P_1 \times P_2 = (x^6 + x^2 + x) + (x^6 + x^3 + x^2 + x) + (x^5 + x^2 + x + 1) = x^5 + x^3 + x^2 + x + 1$			

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 다항식(Polynomial) 표현 (7/7)
    - 컴퓨터 이용 곱셈
      - 비트 알고리즘
        1. 이전 결과의 최상위 비트가 0이라면 이전 결과를 왼쪽으로 한 비트 이동
        2. 이전 결과의 최상위 비트가 1이면
           - 왼쪽으로 한 비트 이동
           - 최상위 비트를 제외하고 모듈로 다항식과 XOR 연산

거듭제곱	왼쪽 이동 연산	XOR 연산
$x^0 \otimes P_2$		10011110
$x^1 \otimes P_2$	00111100	$(00111100) \oplus (00011010) = 00100111$
$x^2 \otimes P_2$	01001110	01001110
$x^3 \otimes P_2$	10011100	10011100
$x^4 \otimes P_2$	00111000	$(00111000) \oplus (00011010) = 00100011$
$x^5 \otimes P_2$	01000110	01000110
$P_1 \otimes P_2 = (01000110) \oplus (01001110) \oplus (00100111) = 00101111$		

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 생성원 사용 (1/2)
    - $GF(2^n)$ 체의 원소를 생성원 g를 사용하여 정의하는 방식
    - 기약 다항식 $f(x)$가 정의된 체에서 체의 원소 a는 반드시 $f(a) = 0$을 만족해야 함
    - g가 체의 생성원이라면 $f(g) = 0$
    - 체의 원소들은 $\{0, g^0, g^1, g^2, \ldots, g^N\}, N = 2^n - 2$로 표기

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 생성원 사용 (2/2)

기약 다항식$f(x) = x^4 + x + 1$를 사용하여 $GF(2^4)$ 체의 원소를 생성하시오

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

$GF(2^n)$ 체

- $GF(2^n)$ 체
  - 생성원 표현
    - 덧셈에 대한 역원은 원소 자신임
      - e.g., $g^2 + g^2 = 0$
    - 곱셈에 대한 역원은 지수를 모듈로 $2^n - 1$로 계산함
      - e.g., $(g^3)^{-1} = g^{-3} = g^{12} = g^3 + g^2 + g + 1 = (1111)$
    - 덧셈과 뺄셈은 동일한 연산임
      - e.g., $g^3 - g^6 = g^3 + g^6 = g^3 + (g^3 + g^2) = g^2 = (0100)$
    - 곱셈은 지수의 덧셈 $mod\ 2^n - 1$임
      - e.g., $g^9 \times g^{11} = g^{20} = g^{20\ mod\ 15} = g^5 = g^2 + g = (0110)$
    - 나눗셈은 곱셈 역원을 사용한 곱셈임
      - e.g., $g^3 \div g^8 = g^3 \times g^{-8} = g^3 \times g^7 = g^{10} = g^2 + g + 1 = (0111)$

Thanks!

김 민 식(ms6127@naver.com)