2025/07/11, 2025 보안 기초 세미나

암호학과 네트워크 보안

-3장 고전 대칭-키 암호, 4장 암호 수학-

이 성 현(dltjdgus2166@naver.com)

세종대학교 프로토콜공학연구실

목 차

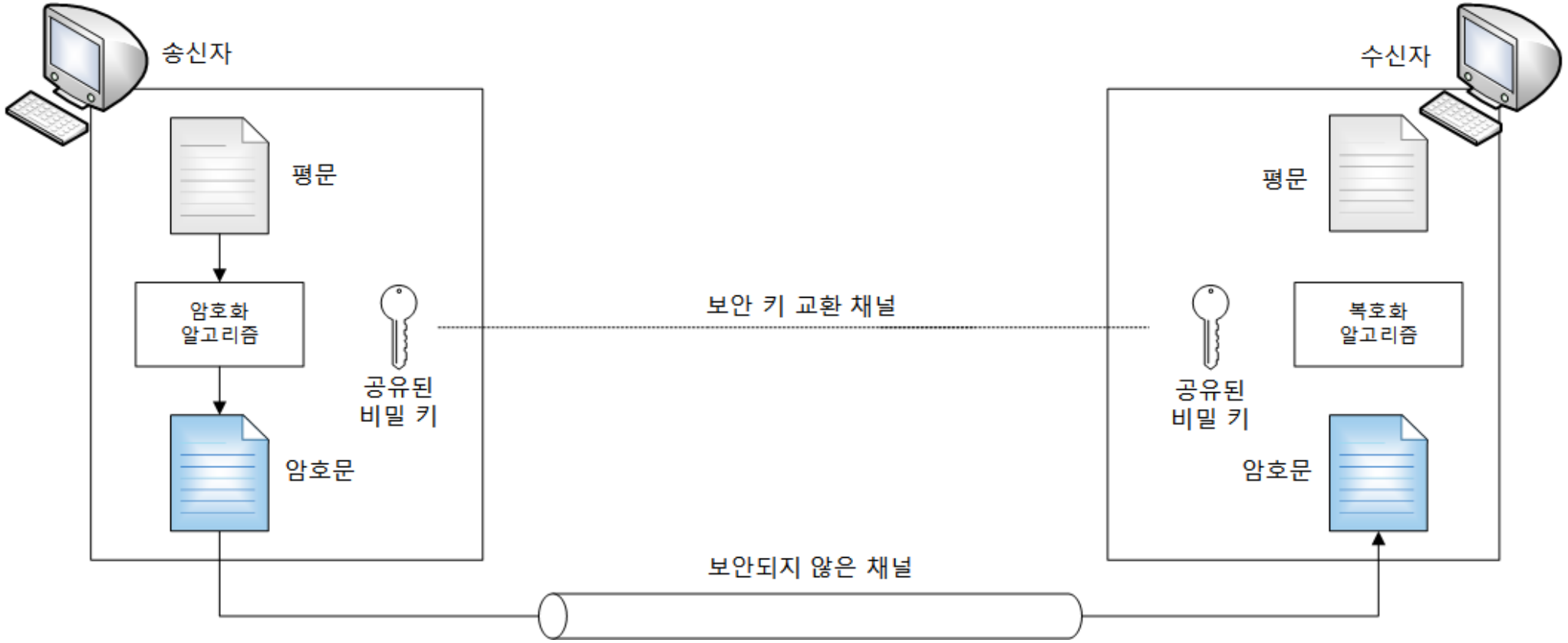
- 고전 대칭-키 암호
  - 개요
  - 대치 암호(Substitution Ciphers)
  - 전치 암호(Transposition Cipher)
  - 스트림 암호(Stream Cipher)와 블록 암호(Block Cipher)
- 암호 수학
  - 대수 구조
  - $GF(2^n)$ 체

개요

- 주요 용어
  - 평문(Plaintext)
    - 암호화 되기 전의 원래 메시지
  - 암호문(Ciphertext)
    - 암호화 알고리즘(Encryption Algorithm)을 통해 암호화된 메시지
  - 암호(Cipher)
    - 정보를 안전하게 전달하기 위해 암/복호화하는 알고리즘
  - 대칭-키(Symmetric-Key)
    - 암/복호화를 위해 송신자와 수신자 사이에 공유된 비밀 정보

개요

- 대칭-키 암호(Symmetric-key Encipherment) (1/3)
  - 정의
    - 암/복호화 과정에서 동일한 비밀 키를 사용하는 암호



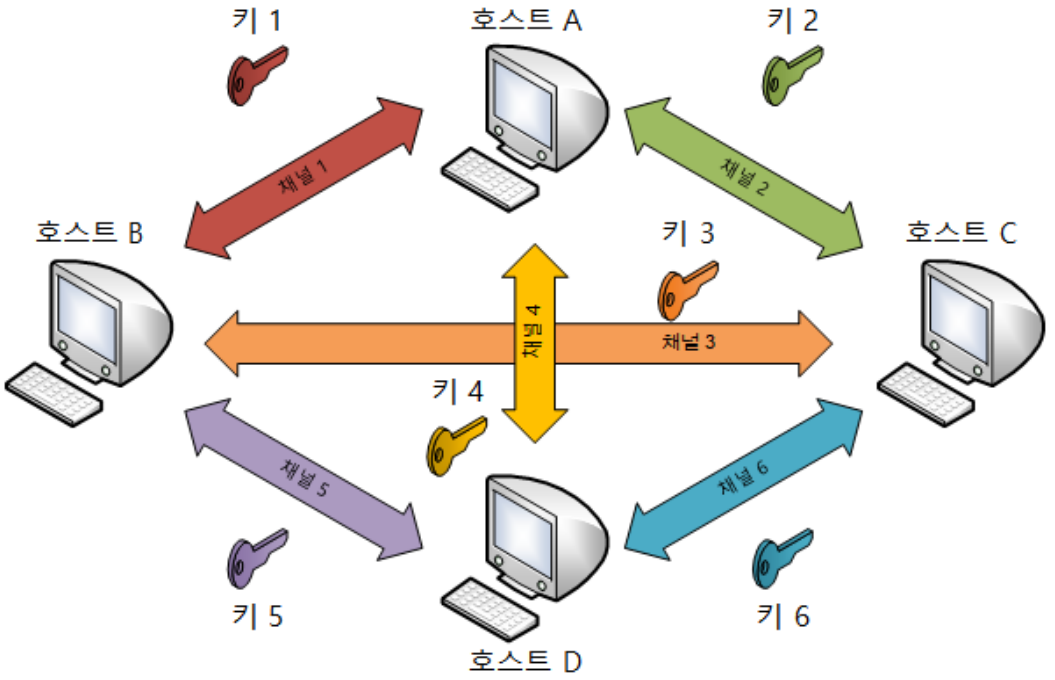
개요

- 대칭-키 암호(Symmetric-key Encipherment) (2/3)
  - 특징 (1/2)
    - 암호화 알고리즘과 복호화 알고리즘이 서로 역관계
      - $C = E_k(P),\ P = D_k(C),\ D\big(E_k(P)\big) = P,\ E(D_k(C)) = C$
    - 비밀 키 공유를 위해 안전한 또 다른 채널이 필요
      - 비밀 키 공유 방법
        - 직접 개인적으로 교환
          - e.g., 이동식 디스크를 이용한 물리적으로 교환
        - 제3자(Third Party)를 통해 공유
          - e.g., KDC (Key Distribution Center) 등을 중간 기관으로 이용
        - 비대칭-키 암호로 임시 비밀 키 생성
          - e.g., TLS (Transport Layer Security) 핸드 셰이크

표기	설명
P	평문
C	암호문
k	키
$E_k(x)$	암호화
$D_k(x)$	복호화

개요

- 대칭-키 암호(Symmetric-key Encipherment) (3/3)
  - 특징 (2/2)
    - 통신을 위해 필요한 키는 통신 당 하나
      - m명의 사용자가 서로 통신할 때 필요한 키의 개수 $(m \times (m-1))/2$개
        - e.g., $m = 4$일 때 필요한 키의 개수는 6개



개요

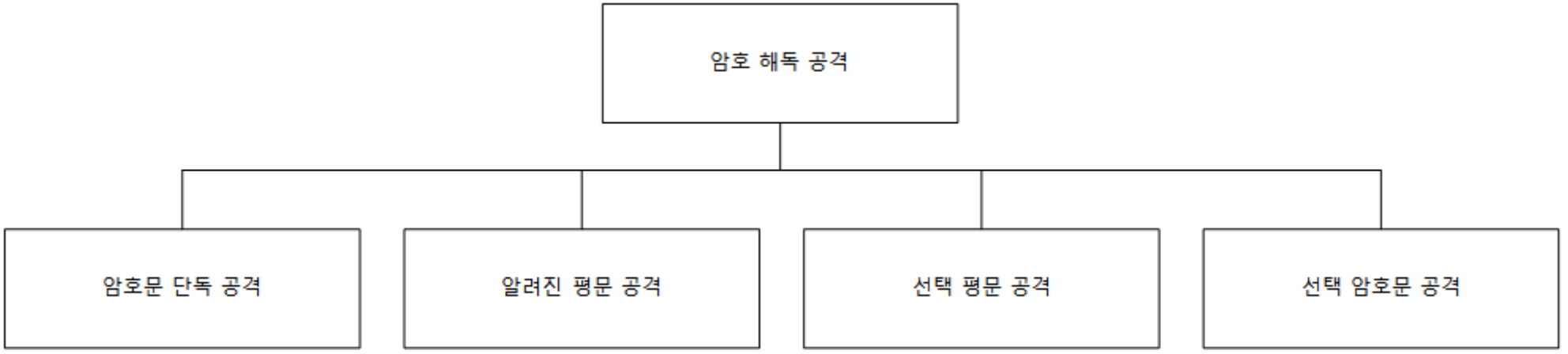
- 케르크호프스(Kerckhoff)의 원리
  - 정의
    - 암호의 안전성은 키의 기밀성에만 의존해야 한다는 원칙
  - 특징
    - 암/복호화 알고리즘 공개 허용
    - 키의 관리와 기밀성 유지가 중요
    - 암호 안전성 유지에 유리

개요

- 암호 해독 (1/6)
  - 정의
    - 암호를 분석하여 암호문을 복호화하거나 키를 유추하는 행위
  - 방법
    - 전수조사 공격(Brute-force Attack)
      - 가능한 키 값을 하나씩 시도하여 키를 찾아내어 암호를 해독하는 방법
    - 통계적인 공격(Statistical Attack)
      - 평문 언어의 통계적 특성과 암호문 간의 상관관계를 통해 암호를 해독하는 방법
    - 패턴 공격(Pattern Attack)
      - 암호문에 존재하는 규칙적인 패턴을 분석하여 암호를 해독하는 방법

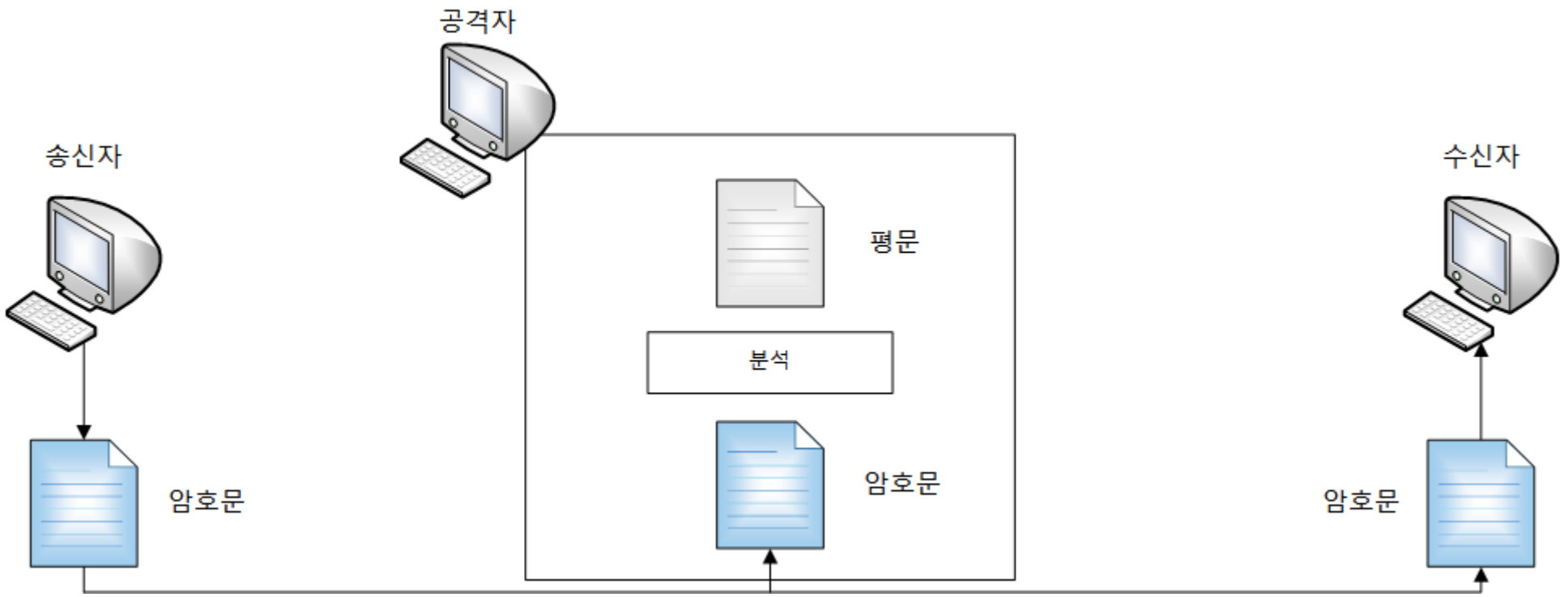
개요

- 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)



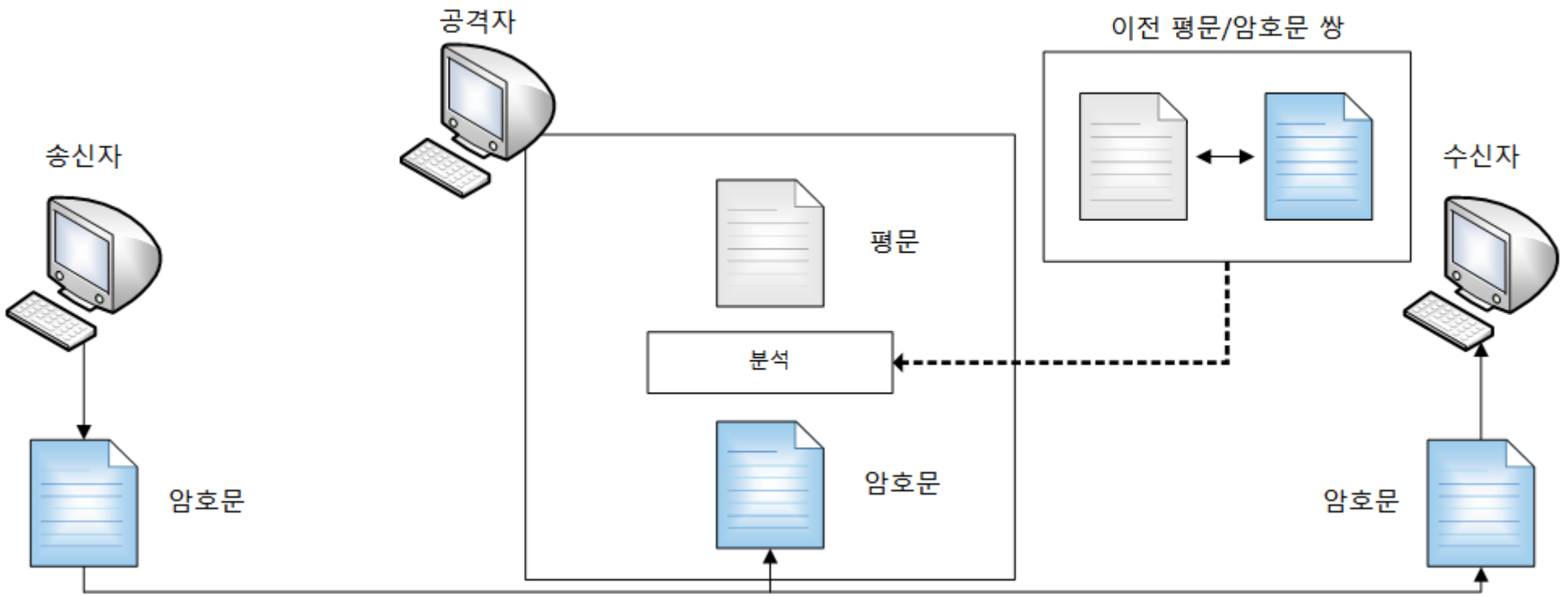
개요

- 암호 해독 (3/6)
  - 암호문 단독 공격(Ciphertext-only Attack)
    - 암호문만을 이용하여 평문이나 키를 유추하는 공격 방식



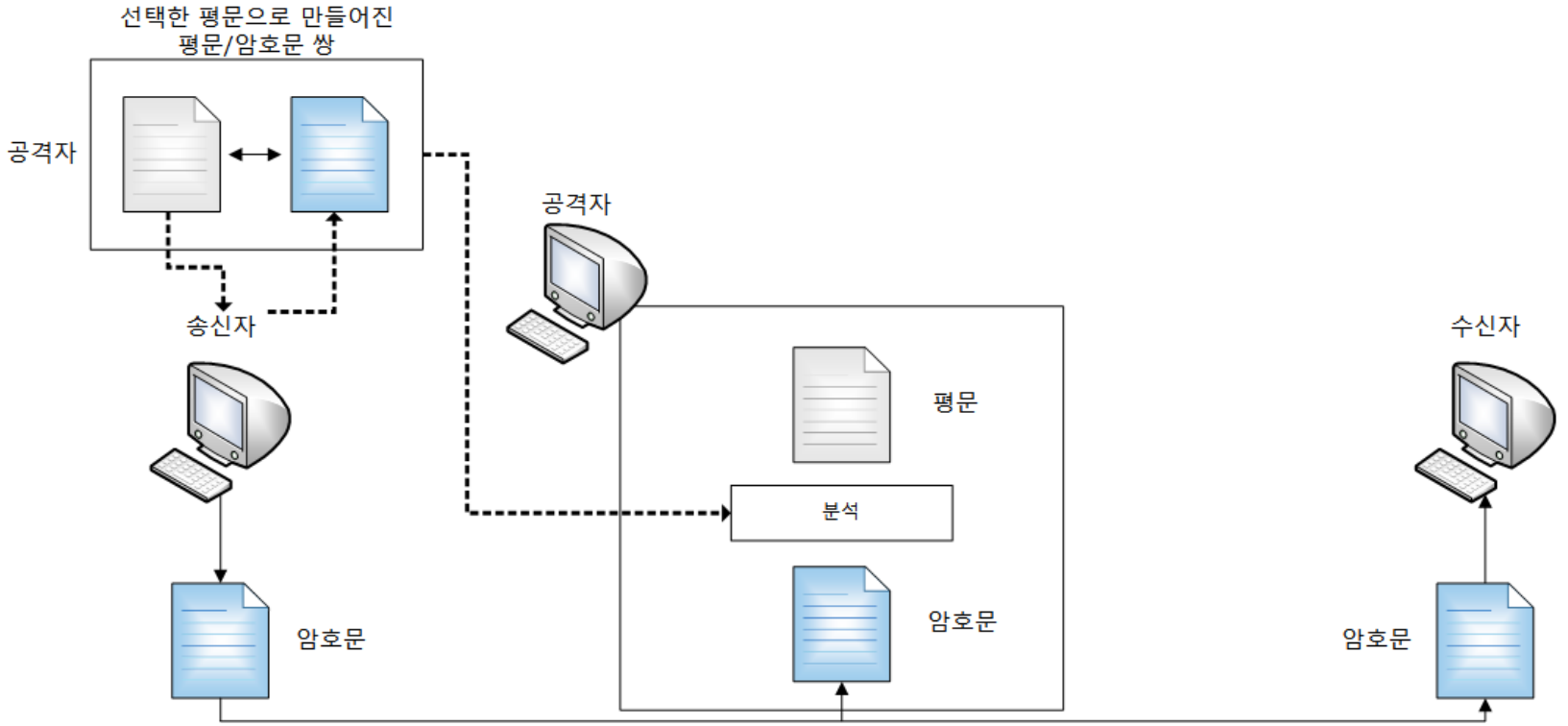
개요

- 암호 해독 (4/6)
  - 알려진 평문 공격(Known-plaintext Attack)
    - 노출된 평문/암호문 쌍을 이용하여 평문이나 키를 유추하는 공격 방식



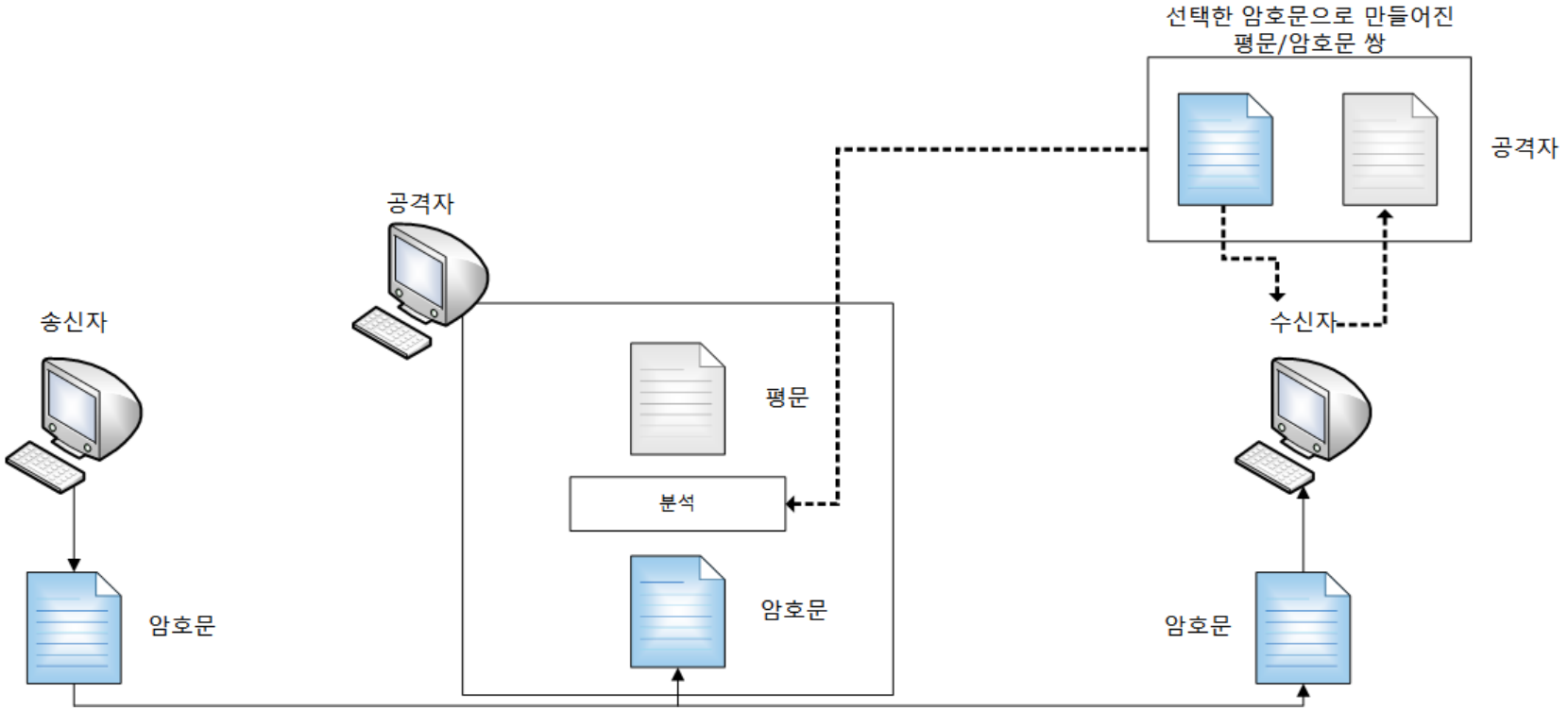
개요

- 암호 해독 (5/6)
  - 선택 평문 공격(Chosen-plaintext Attack)
    - 공격자가 송신자의 컴퓨터에 접속하여 직접 정한 평문을 암호화하여 얻은 평문/암호문 쌍을 이용하여 평문이나 키를 유추하는 공격 방식



개요

- 암호 해독 (6/6)
  - 선택 암호문 공격(Chosen-ciphertext Attack)
    - 공격자가 수신자의 컴퓨터에 접속하여 직접 정한 암호문을 복호화하여 얻은 평문/암호문 쌍을 이용하여 평문이나 키를 유추하는 공격 방식



대치 암호(Substitution Ciphers)

- 대치 암호(Substitution Ciphers)
  - 정의
    - 하나의 문자를 다른 문자로 변경하는 암호
  - 종류
    - 단일문자 암호(Monoalphabetic Ciphers)
      - 평문 문자와 암호문 문자 간 일대일 대응 관계인 암호
    - 다중문자 암호(Polyalphabetic Ciphers)
      - 평문 문자와 암호문 문자 간 일대다 대응 관계인 암호

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

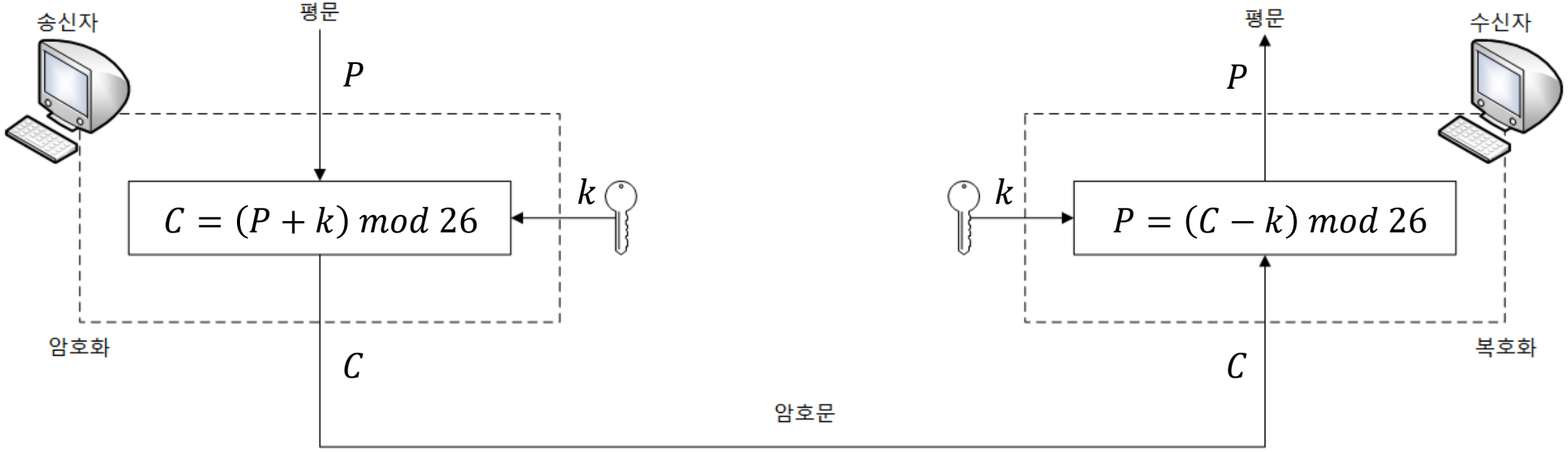
< 암호에 대한 문자 표현 >

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 정의
    - 평문에서 하나의 문자가 위치와 상관 없이 항상 같은 문자로 대체되는 암호
  - 특징
    - 평문의 한 문자와 그에 대응되는 암호문의 한 문자는 일대일 대응 관계임
    - 문자를 하나씩 개별적으로 암호화함
    - 문자의 빈도수를 감추지 않기 때문에 통계적인 공격에 취약함

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 덧셈 암호(Additive Cipher) (1/4)
    - 정의
      - 평문에 키를 더하는 암호화 알고리즘과 암호문을 키만큼 빼는 복호화 알고리즘을 가진 암호



대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 덧셈 암호(Additive Cipher) (2/4)
    - 특징 (1/3)
      - 이동 암호(Shift Cipher) 또는 시저 암호(Caesar Cipher)라고도 불림
      - $C = (P + k)\ mod\ 26$
      - $P = (C - k)\ mod\ 26$

키 15로 “hello” 암호화, “WTAAD” 복호화

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
∴ “hello” → “WTAAD”			∴ “WTAAD” → “hello”		

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 덧셈 암호(Additive Cipher) (3/4)
    - 특징 (2/3)
      - 키 공간이 작아서 전수조사 공격(Brute-force Attack)에 취약함

"UVACLYFZLJBYL"를 전수조사 공격으로 복호화

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

$\therefore k = 7, P =$ "not very secure"

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 덧셈 암호(Additive Cipher) (4/4)
    - 특징 (3/3)
      - 암호문이 길어지면 통계적인 공격에도 취약함

"XLILSYWIGMRSAJTVSPEJSVJSYVVMPPSRHSPPEVWXMWSVSVLSSVVEBB CFIJSVIVWIVPPIVWIGMIZIWSVVSTJJIVW"를 통계적인 공격으로 복호화

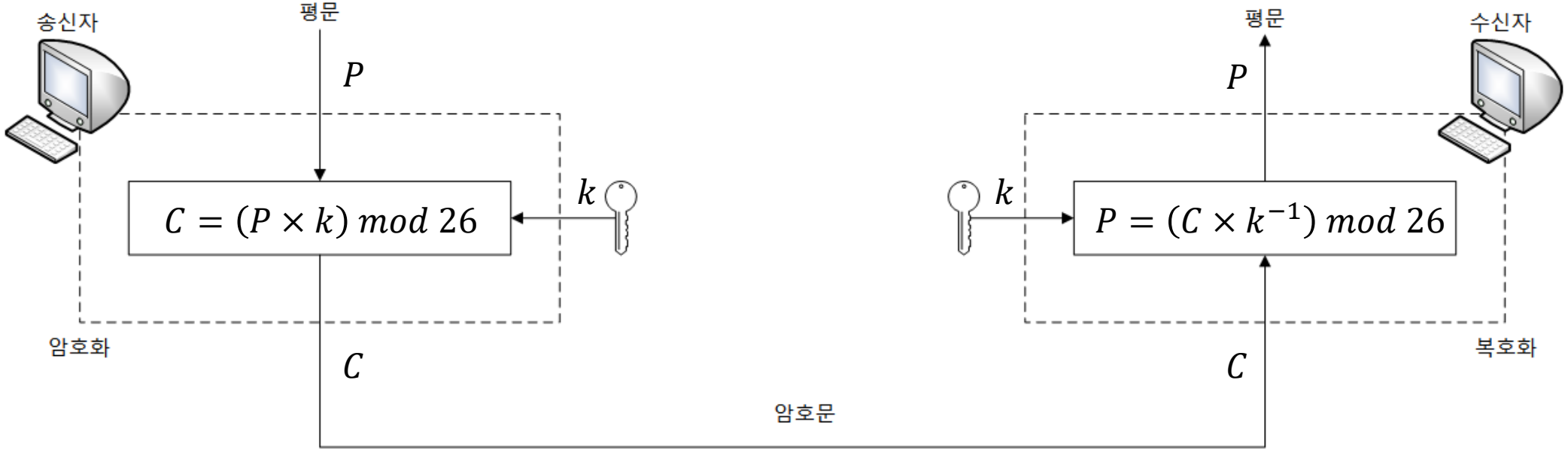
암호문 문자	빈도
I	14
V	13
S	12

∴ I를 e로 가정하면 키 값은 4, 암호문을 4로 복호화하면
the house is now for sale for four million dollars it is worth more hurry before the seller recieves more offers

부록#1 알파벳 출현 빈도수 표 참고

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 곱셈 암호(Multiplicative Ciphers) (1/3)
    - 정의
      - 평문에 키를 곱하는 암호화 알고리즘과 암호문에 키의 곱셈 역원을 곱하는 복호화 알고리즘을 가진 암호



대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 곱셈 암호(Multiplicative Ciphers) (2/3)
    - 특징 (1/2)
      - 암호화와 복호화가 서로 역함수 관계에 있음을 보장하기 위해 키는 집합$Z_{26}{}^{*}$의 원소이어야함
        - $Z_{26}{}^{*} = \{1,3,5,7,9,11,15,17,19,21,23,25\}$
      - $C = (P \times k)\ mod\ 26$
      - $P = (C \times k^{-1})\ mod\ 26$

키 7로 “hello” 암호화, “XCZZU” 복호화

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
∴ “hello” → “XCZZU”			∴ “XCZZU” → “hello”		

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 곱셈 암호(Multiplicative Ciphers) (3/3)
    - 특징 (2/2)
      - 덧셈 암호에 비해 키 공간이 작아서 전수조사 공격에 취약함

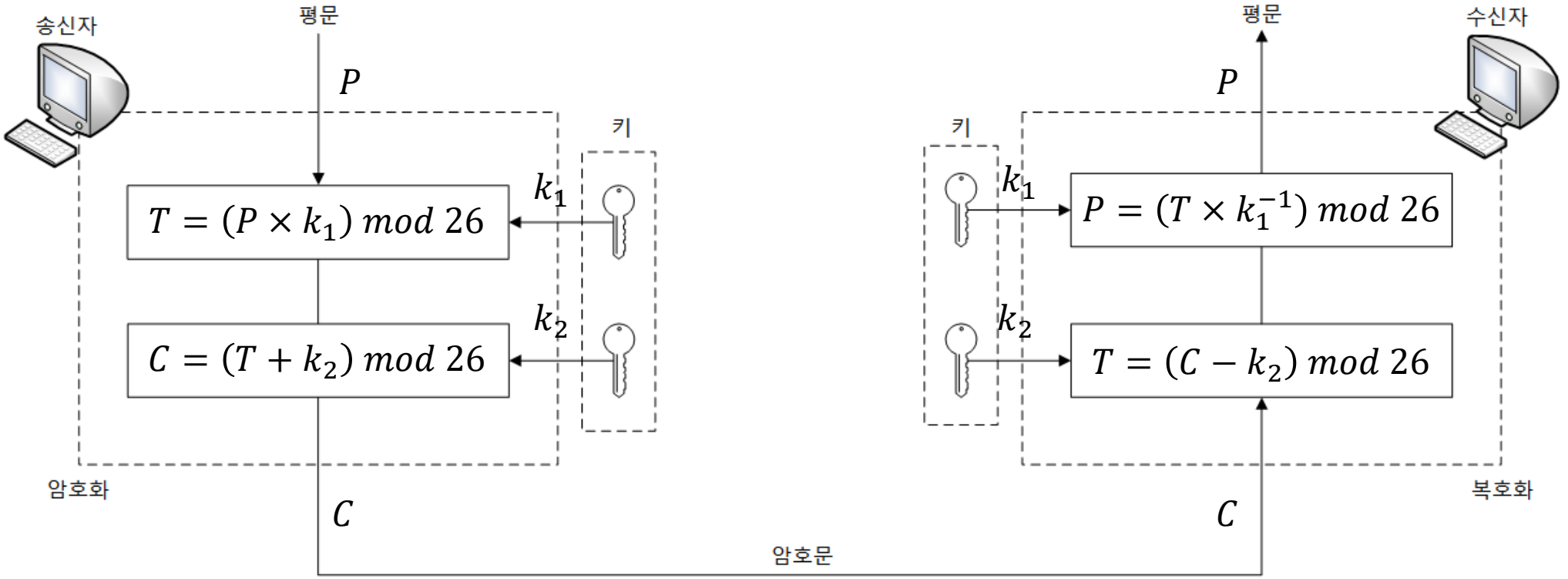
"NUDRCPMWCOKPC"를 전수조사 공격으로 복호화

키 값	키 값의 역원	평문
1	1	nudrcpmwcokpc
3	9	nwbxsfeqswmfs
5	21	neltqdsuqicdq
7	15	notverysecure

$\therefore k = 7, P =$ "not very secure"

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 아핀 암호(Affine Cipher) (1/3)
    - 정의
      - 덧셈 암호와 곱셈 암호를 결합하여 한 쌍의 키를 쓰는 암호



대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 아핀 암호(Affine Cipher) (2/3)
    - 특징
      - 첫 번째 키는 곱셈 암호에, 두 번째 키는 덧셈 암호에 사용함
      - 곱셈 암호를 먼저 적용한 후 덧셈 암호를 사용함
      - 복호화 시에는 덧셈 역원을 먼저 빼준 후 곱셈 역원을 곱함
      - $C = (P \times k_1 + k_2)\ mod\ 26$
      - $P = \left((C - k_2) \times k_1^{-1}\right)\ mod\ 26$

키 (7,2)로 “hello” 암호화, “ZEBBW” 복호화

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

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 아핀 암호(Affine Cipher) (3/3)
    - 특징
      - 덧셈 암호, 곱셈 암호와 마찬가지로 노출된 평문/암호문 쌍이 있으면 키가 쉽게 유추됨

"PWUFFOGWCHFDWIWEJOUUNJORSMDWRHVCMWJUPVCCG"를 선택 평문 공격으로 복호화

평문	선택 알고리즘	암호문
et	알고리즘 1	WC
	알고리즘 2	WF

평문→암호문	계산식	계산	k_1	k_2
e(04)→W(22)	$(04 \times k_1 + k_2) \equiv 22 (mod\ 26)$	$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 22 \\ 2 \end{bmatrix} = \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix} \begin{bmatrix} 22 \\ 2 \end{bmatrix} = \begin{bmatrix} 16 \\ 10 \end{bmatrix}$	16	10
t(19)→C(02)	$(19 \times k_1 + k_2) \equiv 02 (mod\ 26)$			

평문→암호문	계산식	계산	k_1	k_2
e(04)→W(22)	$(04 \times k_1 + k_2) \equiv 22 (mod\ 26)$	$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 22 \\ 5 \end{bmatrix} = \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix} \begin{bmatrix} 22 \\ 5 \end{bmatrix} = \begin{bmatrix} 11 \\ 4 \end{bmatrix}$	11	4
t(19)→F(05)	$(19 \times k_1 + k_2) \equiv 05 (mod\ 26)$			

∴ 복호화된 문장 : best time of the year is spring when flowers bloom

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 단일문자 대치 암호(Monoalphabetic Substitution Ciphers) (1/2)
    - 정의
      - 각 문자를 다른 문자에 일대일 대응시켜 대치하는 암호

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

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

∴ ICFVQRVVNERFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

대치 암호

- 단일문자 암호(Monoalphabetic Ciphers)
  - 단일문자 대치 암호(Monoalphabetic Substitution Ciphers) (2/2)
    - 특징
      - 가능한 키 공간이 26!로, 덧셈 암호, 곱셈 암호 등에 비해 상대적으로 키 공간이 큼
      - 문자의 출현 빈도수를 변경시키지 않기 때문에 통계적인 공격은 여전히 유효함

"ICFVQRVVNERFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS"에 통계적인 공격 시도

암호문 문자	빈도
I	6
R	6
V	5

∴ I 또는 R을 e로 가정하고 대입 가능

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 정의
    - 하나의 평문 문자가 암호화될 때, 상황에 따라 서로 다른 문자로 대치되는 암호
  - 특징
    - 평문 문자와 그 문자의 위치에 따라 암호문 문자를 생성함
    - 평문 문자와 암호문 문자와의 관계는 일대다 대응임
    - 키는 i번째 문자를 암호화하는 부분키 k_i로 이루어진 키 수열임
    - 언어의 문자 빈도를 감출 수 있음

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 자동키 암호(Autokey Cipher) (1/2)
    - 정의
      - 초기 키 값 이후의 키 값을 평문 문자를 통해 생성하는 암호
    - 특징 (1/2)
      - 첫 번째 부분키는 사전에 정의된 값이며 두 번째 부분키부터는 이전 번째 평문 문자의 값을 가짐
        - $P = P_1P_2P_3 \ldots, C = C_1C_2C_3 \ldots, k = (k_1, P_1, P_2, \ldots)$
        - $C_i = (P_i + k_i)\ mod\ 26, P_i = (C_i - k_i)\ mod\ 26$

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

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

∴ “attack is today” → “MTMTCMSALHRDY”

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 자동키 암호(Autokey Cipher) (2/2)
    - 특징 (2/2)
      - 단일문자의 빈도 통계를 감출 수 있지만 키 공간이 26으로 전수조사 공격에 취약함

"UBHOZVPQWGWLV"를 전수조사 공격으로 복호화

키 값	평문
1	tizpklemkwalk
2	sjyqjmdnjxzmj
3	rkxrincoiyyni
4	qlwshobphzxoh
5	pmvtgpagawpg
6	onuufqzrfbvqf
7	notverysecure

$\therefore k = 7, P =$ "not very secure"

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 플레이페어 암호(Playfair Cipher) (1/2)
    - 정의
      - 평문을 2글자의 문자 쌍으로 나누어 5×5 알파벳 키 테이블을 기반으로 각 문자 쌍을 규칙에 따라 대치하는 암호
    - 특징
      - 키 수열이 평문의 문자의 위치에 영향을 받음
      - 암호화하기 전 두 문자가 연속하여 같으면 둘을 분리하기 위해 가짜 문자(Bogus Letter)를 삽입함
      - 평문 문자의 개수가 홀수이면 문자의 개수를 짝수로 만들기 위해 맨 끝에 추가적인 가짜 문자를 삽입함
      - 키 테이블 크기와 문자 개수를 맞추기 위해서 주로 I와 J를 같은 문자로 취급하여 키 테이블을 구성함
      - 키 공간의 크기가 25!로, 전수조사 공격을 적용하기 어려움
      - 단일 문자 빈도를 숨기지만 두 문자열의 빈도는 보존됨

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 플레이페어 암호(Playfair Cipher) (2/2)
    - 암호화 규칙
      1. 한 쌍으로 된 두 문자가 비밀 키의 같은 행에 위치하면 각 문자에 대응되는 암호 문자는 같은 행의 오른쪽에 인접하는 문자
         - e.g., (h,e) → (E,C)
      2. 한 쌍으로 된 두 문자가 비밀 키의 같은 열에 위치하면 각 문자에 대응되는 암호 문자는 같은 열에서 그 아래에 위치한 문자
         - e.g., (l,x) → (Q,Z)
      3. 한 쌍으로 된 두 문자가 비밀 키의 같은 행이나 열에 위치하지 않으면 각 문자에 대응되는 암호 문자는 그 자신의 행에서 다른 문자와 같은 열에 위치한 문자
         - e.g., (l,o) → (B,X)

∴ “hello” → “helxlo” → “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

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - Vigenere 암호 (1/4)
    - 정의
      - 반복되는 키를 이용하여 문자를 대치하는 암호
    - 특징 (1/4)
      - 키 수열은 길이가 $m(1 \leq m \leq 26)$인 초기 비밀 키 수열의 반복
      - 평문 문자에 의존하지 않고 평문 문자의 위치에만 의존하므로 수열이 평문을 몰라도 생성될 수 있음

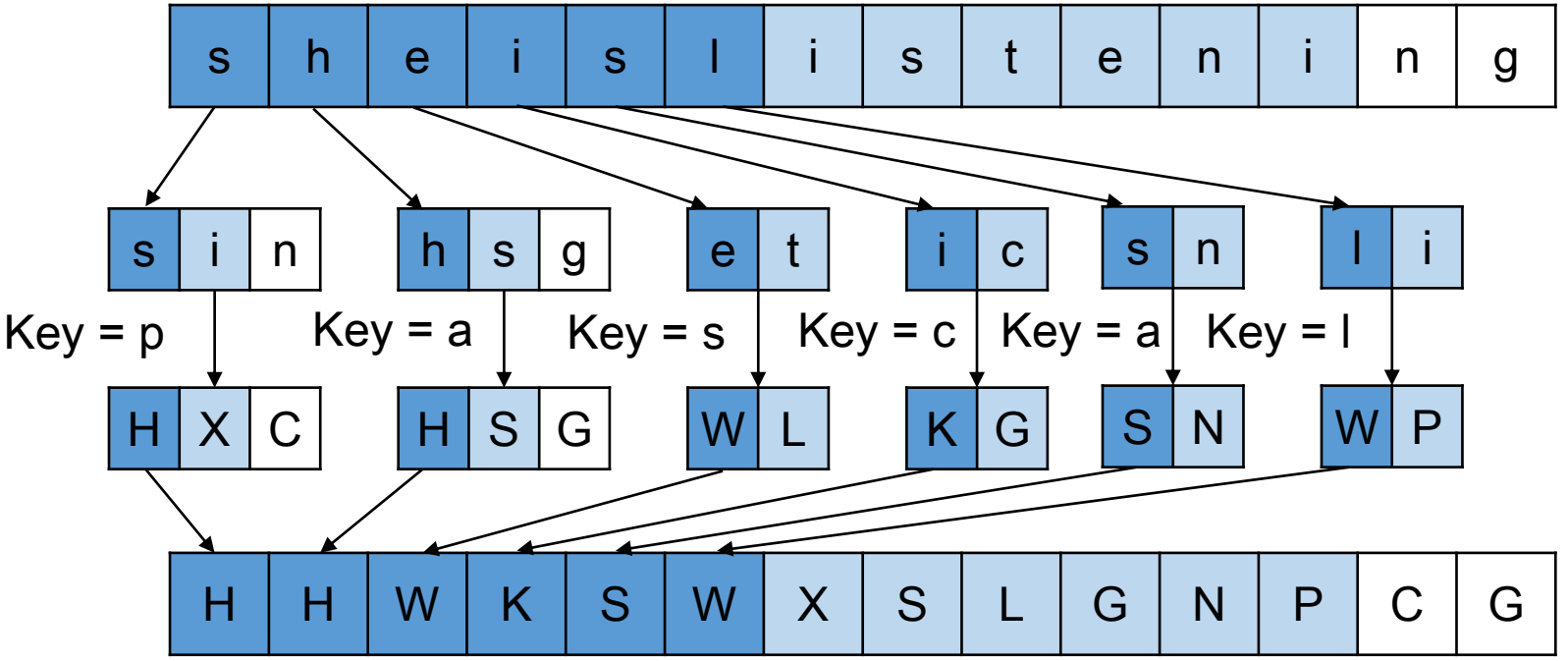
"she is listening"을 초기 키 값 15로 암호화

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

∴ "she is listening" → "HHWKSWXSLGNTCG"

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - Vigenere 암호 (2/4)
    - 특징 (2/4)
      - Vigenere 암호는 m개의 덧셈 암호의 조합으로 볼 수 있음



부록#2 Vigenere 표 참고

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - Vigenere 암호 (3/4)
    - 특징 (3/4)
      - 문자의 빈도를 보존하지 않음
        - Kasiski 테스트 (1/2)
          - 정의
            - 암호문에서 반복되는 문자 조각을 찾아 그 반복 간격의 거리를 계산하고 그 거리들의 공약수를 통해 키 길이를 추정하는 방법
          - 과정
            1. 암호문에서 최소 세 문자로 구성된 반복된 원문 조각 두 개를 찾고 그 사이의 거리가 d라고 가정할 때 키의 길이 m에 대해 $d \mid m$이라고 추정함
            2. 거리가 $d_1, d_2, \ldots, d_n$인 더 많은 반복된 조각이 발견되면 $gcd(d_1, d_2, \ldots, d_n) \mid m$
            3. 키의 길이를 찾은 후, 암호문을 m개의 다른 부분으로 나누고 전수조사 공격이나 통계적인 공격을 실행함

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - Vigenere 암호 (4/4)
    - 특징 (4/4)
      - 문자의 빈도를 보존하지 않음
        - Kasiski 테스트 (2/2)

"LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTHVTSGXQOVGCSVETQLTJSUMVWVEUVLXEXEWSLGFZMVVWLGYHCUSWX QHKVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWVWJWIXGFWLTSHGJOUEEHHVUCFVGOWICQLTJSUXGLW"를 복호화

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

$\therefore 4 \mid m$

$C_1 =$ LWGWCRAOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG
$P_1 =$ jueuapymircneroarhtsthihytrahcieixsthcarrehe
$C_2 =$ IGGGQHGWGKVCTSOSQSWVWFVYSHSVFSHWWFSOHCOQSL
$P_2 =$ usssctsiswhofeaeceihcetesoecatnpntherhctecex
$C_3 =$ OFDHURWQZKLZHGVVLUVLSZWHWKHFDUKDHVIWHUHFWLUW
$P_3 =$ lcaerotnwhiwedssirsiirhketehretltiideatrairt
$C_4 =$ MEVHCWILEMWVVXGETMEXLMLCXVELGMIMBWXLGEVVITX
$P_4 =$ iardysehaisrrtcapiafpwtethecarhaesfterectpt

$\therefore P =$ 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.

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - Hill 암호 (1/3)
    - 정의
      - 평문을 고정된 크기의 블록으로 나눠 키 행렬을 통해 대치하는 암호
    - 특징 (1/3)
      - 키는 m이 블록의 크기일 때, $m \times m$ 정방 행렬임
      - 각 암호문 문자가 블록에 속한 모든 평문 문자에 의존함
      - 평문 블록에서 m개의 문자를 $P_1, P_2, \dots, P_m$ 이라면 암호문 블록에서 연관된 문자는 $C_1, C_2, \dots, C_m$임
        - $C_1 = P_1k_{11} + P_2k_{21} + \cdots + P_mk_{m1}$
          $C_2 = P_1k_{12} + P_2k_{22} + \cdots + P_mk_{m2}$
          ...
          $C_m = P_1k_{1m} + P_2k_{2m} + \cdots + P_mk_{mm}$

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

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - Hill 암호 (2/3)
    - 특징 (2/3)
      - 키 행렬은 곱셈에 대한 역원을 가져야 함

Hill 암호의 암호화 및 복호화

$$\overset{C}{\begin{bmatrix}15 & 14 & 07\\ 20 & 15 & 22\end{bmatrix}} = \overset{P}{\begin{bmatrix}00 & 02 & 19\\ 13 & 14 & 22\end{bmatrix}} \overset{K}{\begin{bmatrix}06 & 13 & 20\\ 24 & 16 & 17\\ 01 & 10 & 15\end{bmatrix}}$$

$$\overset{P}{\begin{bmatrix}00 & 02 & 19\\ 13 & 14 & 22\end{bmatrix}} = \overset{C}{\begin{bmatrix}15 & 14 & 07\\ 20 & 15 & 22\end{bmatrix}} \overset{K^{-1}}{\begin{bmatrix}08 & 21 & 21\\ 05 & 08 & 12\\ 10 & 21 & 08\end{bmatrix}}$$

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - Hill 암호 (3/3)
    - 특징 (3/3)
      - 키 공간의 크기가 커서 전수조사 공격이 어려움
      - 평문의 통계를 보존하지 않으므로 통계적인 공격도 어려움
      - m값과 최소 m개 블록에 대한 평문/암호문 쌍을 알고 있다면 암호에 대해 알려진 평문 공격을 할 수 있음

$m = 3$일때 평문/암호문 블록 쌍 3개로 키 행렬 구하기

$$\begin{matrix} [05 & 07 & 10] \\ [13 & 17 & 07] \\ [00 & 05 & 04] \end{matrix} \longleftrightarrow \begin{matrix} [03 & 06 & 00] \\ [14 & 16 & 09] \\ [03 & 17 & 11] \end{matrix}$$

$$P \qquad\qquad C$$

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

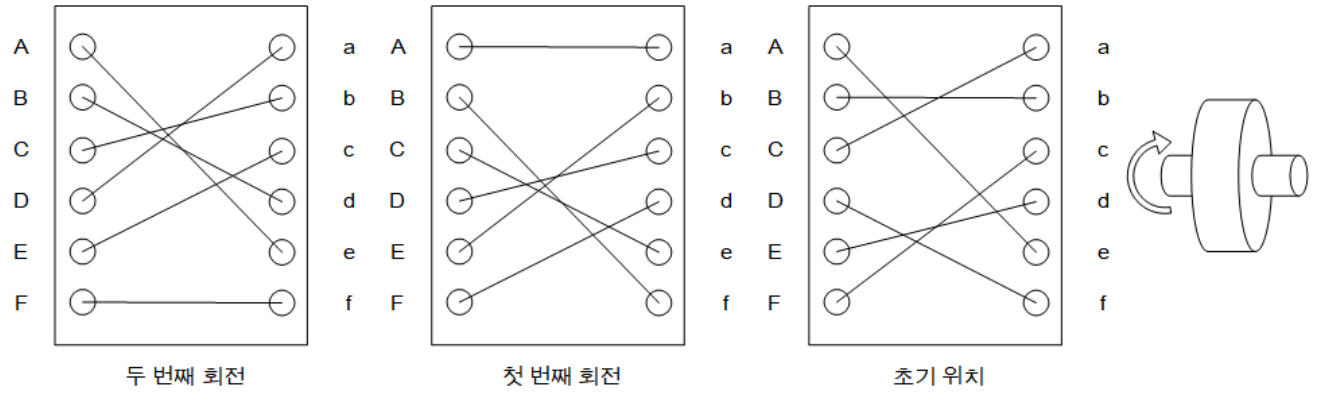
$$K \qquad\qquad P^{-1} \qquad\qquad C$$

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 일회용 패드(One-time pad)
    - 정의
      - 평문과 동일한 길이의 무작위로 정해진 키를 이용하여 대치하는 암호
    - 특징
      - 안정성을 위해 키는 한 번 사용 후 절대 재사용되지 않음
      - 매번 새로운 키가 갱신되어 상대에게 전해야하기 때문에 상업적으로 구현되기는 어려움
        - e.g., 군 통신과 같이 기밀성이 중요한 통신에 주로 사용됨

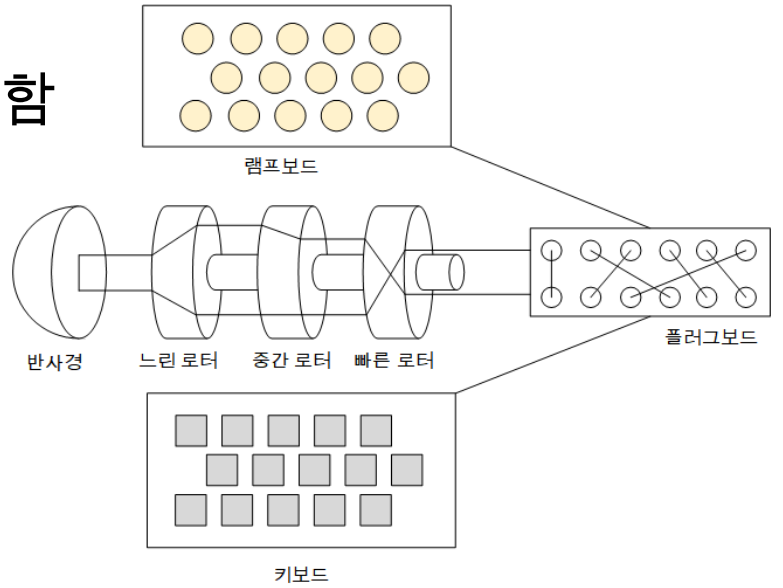
대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - Rotor 암호
    - 정의
      - 회전하는 Rotor를 이용해 평문 문자를 대치하는 암호
    - 특징
      - 첫 번째 평문 문자가 초기 설정을 사용하여 암호화되고 두 번째 문자가 첫 번째 회전 이후 암호화되며 나머지도 이와 같은 방법으로 암호화됨
      - Rotor가 회전하면서 대치 방식이 바뀌기 때문에 같은 문자여도 다르게 대치될 수 있음
        - e.g., “aaa” → “CAD”



대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 에니그마 기계(Enigma machine) (1/5)
    - 정의
      - 여러 개의 회전하는 Rotor와 반사경(Reflector), 플러그보드(Plugboard)를 통해 평문을 대치하는 암호
    - 특징
      - Sherbius에 의해 최초로 고안되었으나 독일군에 의해 변형되어 제2차 세계대전에서 사용됨
      - Rotor 암호의 원리를 기반으로 함



대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 에니그마 기계(Enigma machine) (2/5)
    - 구성 요소
      - 키보드(Keyboard)
        - 문자를 입력하는데 사용되는 26개의 키를 가짐
      - 램프보드(Lampboard)
        - 문자를 보여주는 26개의 램프를 가짐
      - 플러그보드(Plugboard)
        - 13개의 전선으로 수동으로 연결된 26개의 플러그를 가짐
        - 구성은 매일 변함
      - 평문을 암호화 시 쓸 Rotor
        - 다섯 개의 사용가능한 Rotor로부터 매일 세 개의 Rotor를 선택함
        - 각 로터의 속도는 앞선 Rotor의 1/26
      - 사전 연결(Prewired)되어 고정된 반사경(Reflector)

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 에니그마 기계(Enigma machine) (3/5)
    - 코드북 (Code Book)
      - 정의
        - 에니그마 기계의 설정 값을 기록한 책자
      - 구성 요소
        - 다섯 개의 사용 가능한 Rotor 중에 선택된 세 개의 Rotor
        - Rotor가 설치되는 순서
        - 플러그보드의 설정
        - 그날의 세 문자 코드

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 에니그마 기계(Enigma machine) (4/5)
    - 암호화 과정
      1. Rotor의 시작점을 그날의 코드로 맞춤
         - e.g., “HUA”라면 각각 “H”, “U”, “A”로 설정함
      2. 랜덤한 세 문자 코드를 선택하고 단계 1에서의 Rotor의 초기 설정을 이용하여 문장을 암호화함
         - e.g., “ACF”를 선택해 “ACFACF”를 암호화화여 “OPNABT” 얻음
      3. Rotor의 시작점을 선택한 코드로 설정함
         - e.g., Rotor의 시작점을 “ACF”으로 설정함
      4. 암호화할 메시지를 해당 코드로 암호화함
         - e.g., “HELLO”를 암호화하여 “QCNPO”를 얻음
      5. 단계 2에서 얻은 코드를 보낼 메시지 앞에 추가하여 보냄
         - e.g., “OPNABTQCNPO”를 보냄

대치 암호

- 다중문자 암호(Polyalphabetic Ciphers)
  - 에니그마 기계(Enigma machine) (5/5)
    - 복호화 과정
      1. 메시지를 받고 첫 번째 여섯 글자를 분리함
         - e.g., “OPNABTQCNPO”에서 “OPNABT”를 떼어냄
      2. Rotor의 시작점을 그날의 코드로 설정함
         - e.g., 각각 “H”, “U”, “A”로 설정함
      3. 단계 2의 시작점을 사용하여 처음 여섯 개 글자를 복호화함
         - e.g., “OPNABT”를 복호화하여 “ACFACF”를 얻음
      4. 복호화된 코드의 첫 번째 반으로 Rotor의 위치를 설정함
         - e.g., 각각 “A”, “C”, “F”로 설정함
      5. 첫 번째 여섯 글자 없이 메시지를 복호화함
         - e.g., “QCNPO”를 복호화하여 “HELLO”를 얻음

전치 암호(Transposition Cipher)

- 정의
  - 평문의 문자 순서를 일정한 규칙에 따라 재배열하는 암호
- 특징
  - 문자의 위치만 바뀌고, 값은 유지됨
  - 문자의 빈도수가 유지되어 통계적인 공격에 취약함
- 종류
  - 키가 없는 전치 암호
  - 키가 있는 전치 암호
  - 열 전치 암호
  - 이중 전치 암호

전치 암호

- 키가 없는 전치 암호 (1/2)
  - 정의
    - 키 없이 문자를 고정된 규칙을 통해 항상 같은 방식으로 재배열하는 암호
  - 특징
    - 고정된 규칙에 의해 항상 같은 방식으로 전치되어 규칙만 알고 있다면 누구나 암/복호화 할 수 있음
    - 반복되는 규칙으로 암호화될 경우 암호문에 패턴이 생겨 패턴 공격에 취약함

m	e	e	t
m	e	a	t
t	h	e	p
a	r	k	

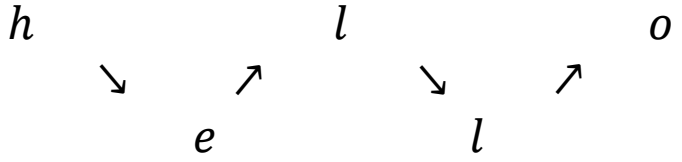
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
01	05	09	13	02	06	10	13	03	07	11	15	04	08	12

“meet at the park” → “MMTAEEHREAEKTTP”

전치 암호

- 키가 없는 전치 암호 (2/2)
  - Rail Fence
    - 정의
      - 문자를 지그재그로 배치하고 행 순서로 읽어 문자를 재배열하는 암호
    - 특징
      - 암호에 사용할 Rail 수를 송신자와 수신자가 사전에 정의하면 키가 있는 전치 암호처럼 동작할 수도 있음
      - 복호화 시에는 암호문을 절반으로 나눠 두 행에 배치하고 지그재그로 읽음

"hello"를 Rail Fence로 암호화



∴ "hello" → "HLOEL"

전치 암호

- 키가 있는 전치 암호
  - 정의
    - 사전에 합의된 키를 이용하여 문자를 재배열하는 암호
  - 특징
    - 평문을 블록으로 나눈 뒤 각각의 블록에서 문자를 치환하기 위해 따로 키를 사용함
    - 암/복호화 시 문자를 어떻게 치환할지 나타내는 치환 키를 사용함
    - 블록 크기를 맞추기 위해 가짜 문자(Bogus Charater)를 삽입할 수 있음

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 G

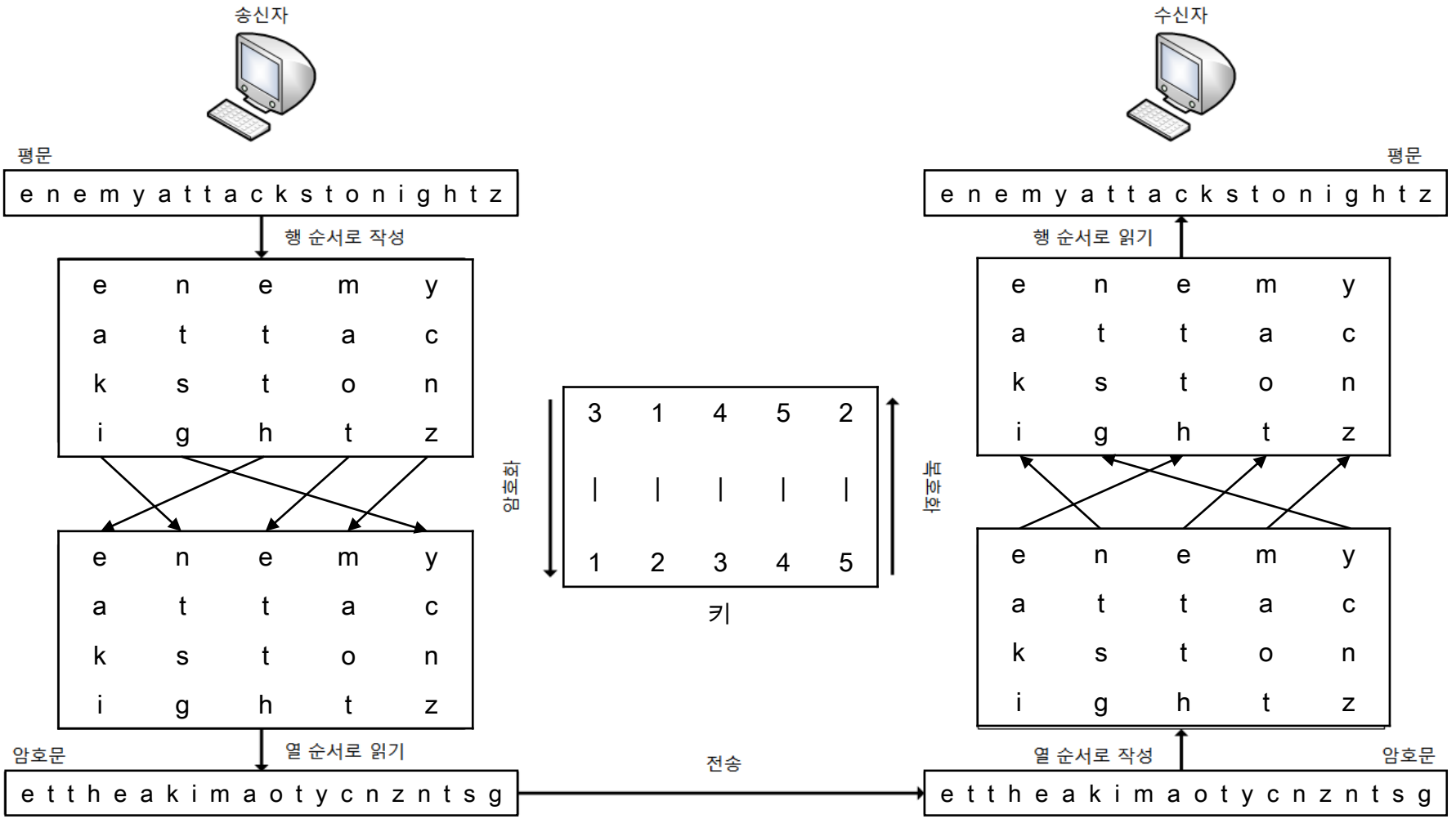
∴ “enemy attacks tonightz” → “EEMYNTAACTTKONSHITZG”

전치 암호

- 열 전치 암호(Columnar Transposition Ciphers) (1/2)
  - 정의
    - 문자를 행렬 형태로 배치한 후 규칙이나 키에 따라 재배열하고 열 순서로 읽어서 재배열하는 암호
  - 특징
    - 키가 없는 열 전치 암호와 키가 있는 열 전치 암호로 분류함
    - 일반적으로 키가 있는 열 전치 암호를 의미함
    - 블록의 크기를 맞추기 위해서 가짜 문자(Bogus Character)가 삽입될 수 있음
    - 암호문에 특정한 패턴이 나타날 수 있음

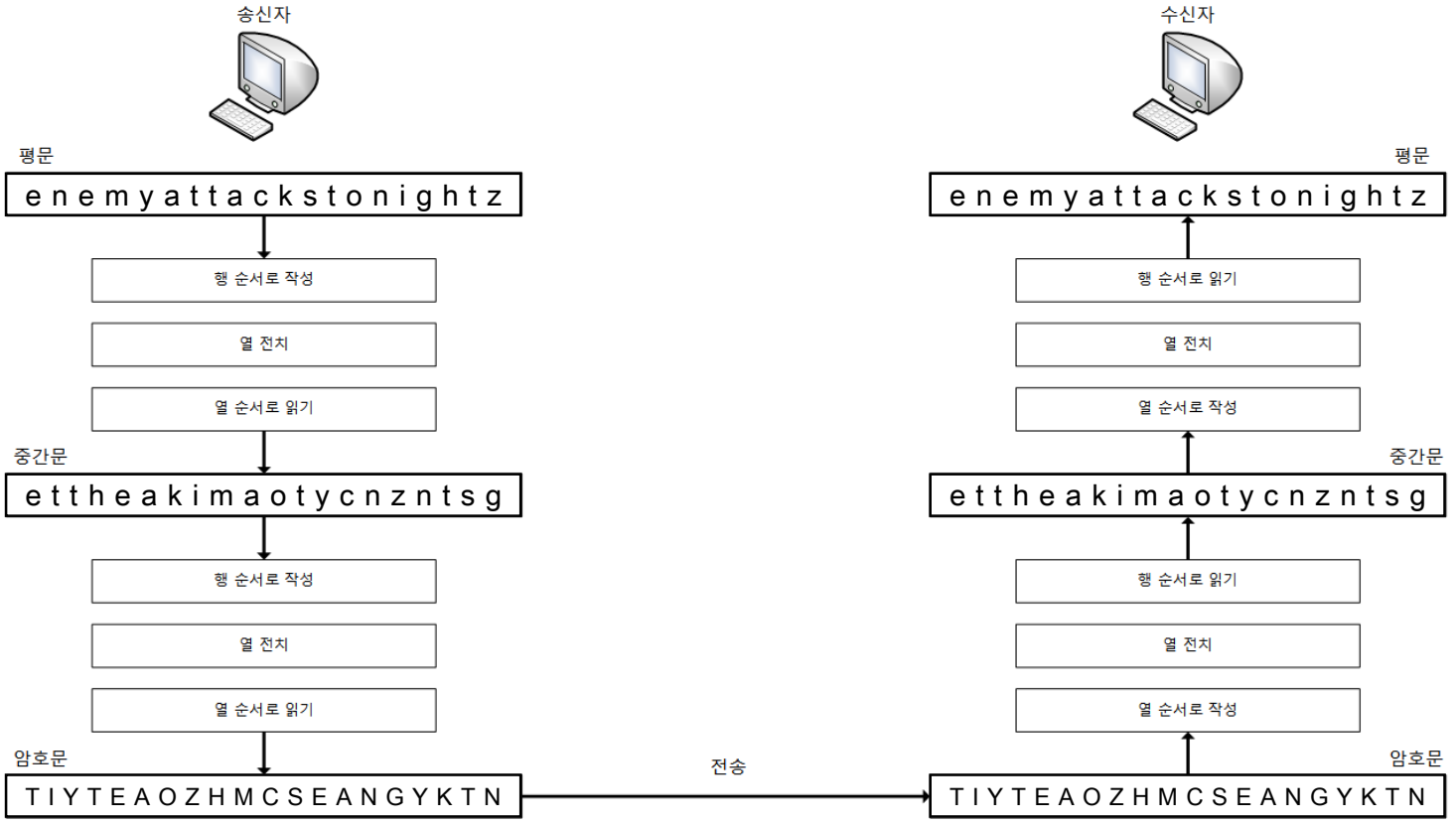
전치 암호

- 열 전치 암호(Columnar Transposition Ciphers) (2/2)



전치 암호

- 이중 전치 암호(Double Transposition Cipher) (1/2)
  - 정의
    - 전치를 두 번 연속으로 적용한 암호

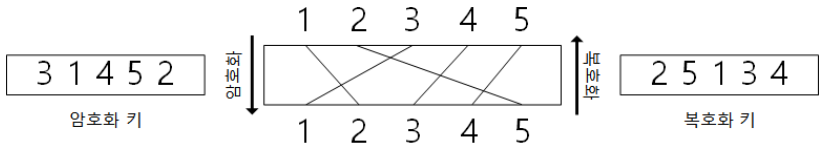


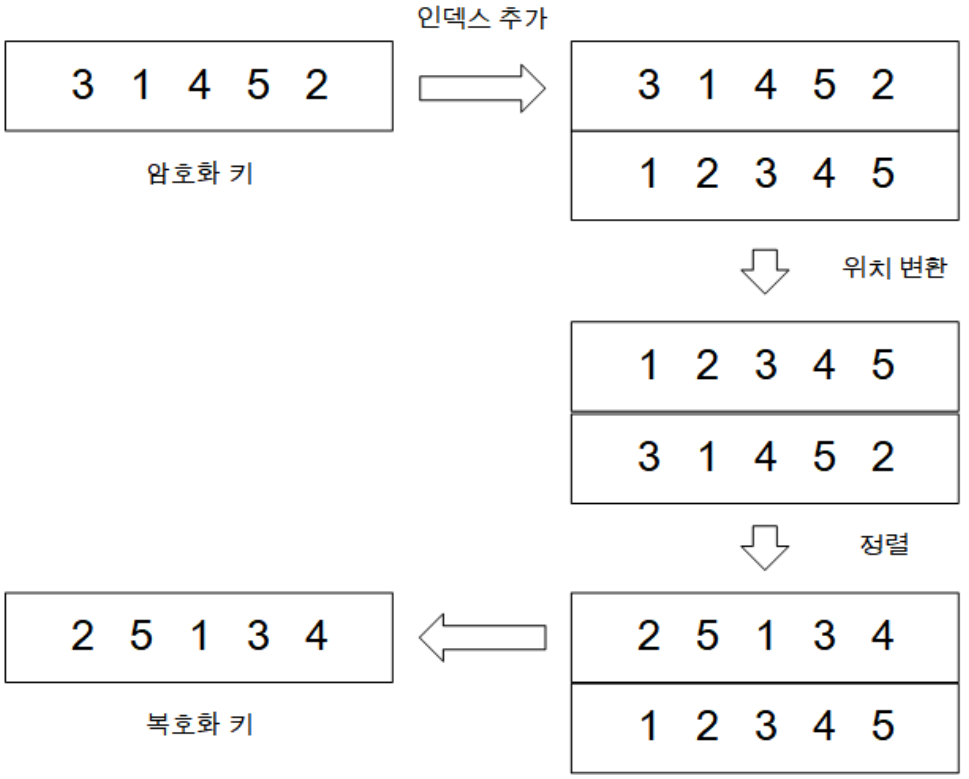
전치 암호

- 이중 전치 암호(Double Transposition Cipher) (2/2)
  - 특징
    - 일반적으로 전치마다 다른 키를 이용함
    - 같은 키를 사용해서 전치할 수도 있음
    - 이중 전치 이후 암호문이 가지고 있던 패턴이 사라짐

전치 암호

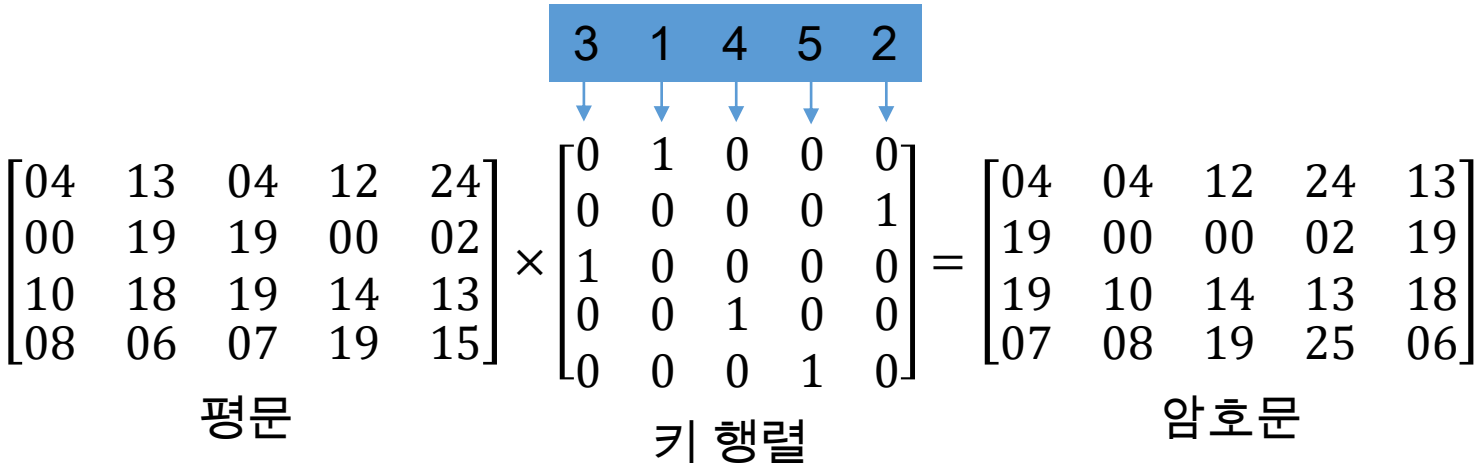
- 암호화/복호화 (1/2)
  - 키 사용
    - 각각의 열에 대해 하나의 입력 값을 가지는 표에 저장됨





전치 암호

- 암호화/복호화 (2/2)
  - 행렬 사용
    - 평문과 암호문은 문자의 수치를 표현하는 $l \times m$ 행렬이고 키는 $m \times m$ 정방 행렬임
    - 암호화는 평문 행렬에 키 행렬을 곱함으로써 암호문 행렬을 얻고 복호화는 암호문에 키 행렬의 역행렬을 곱합으로써 평문 행렬을 얻음



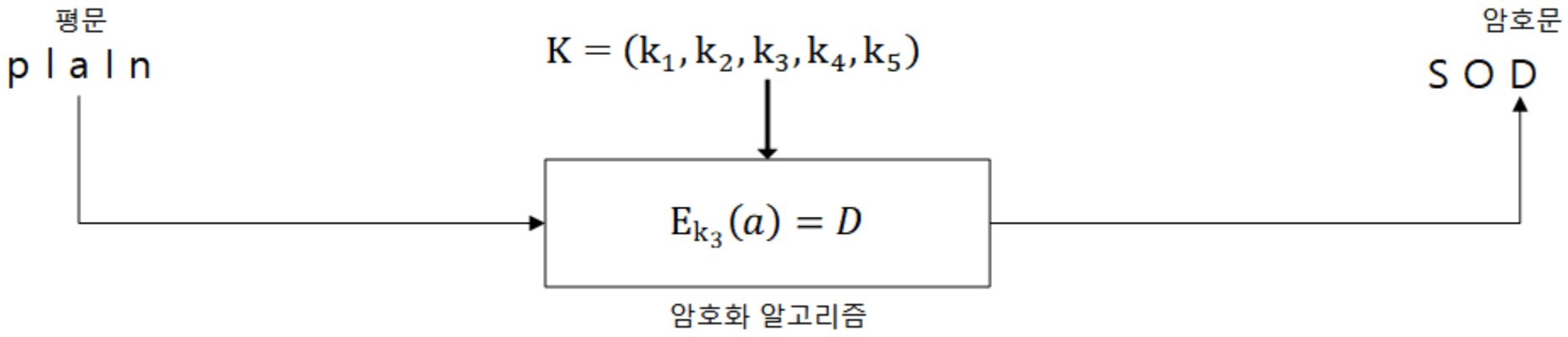
$$\begin{bmatrix} 04 & 13 & 04 & 12 & 24 \\ 00 & 19 & 19 & 00 & 02 \\ 10 & 18 & 19 & 14 & 13 \\ 08 & 06 & 07 & 19 & 15 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 04 & 04 & 12 & 24 & 13 \\ 19 & 00 & 00 & 02 & 19 \\ 19 & 10 & 14 & 13 & 18 \\ 07 & 08 & 19 & 25 & 06 \end{bmatrix}$$

전치 암호

- 암호 해독
  - 통계적인 공격
    - 전치 암호는 암호문의 문자 빈도를 변화시키지 않고 문자를 재정렬하므로 암호문의 길이가 충분히 길 경우 통계적인 공격에 취약함
  - 전수조사 공격
    - 열의 개수를 추측하는 방법으로 접근해야 하며 열의 개수는 문자 길이의 인수들의 조합이 될 수 있음
  - 패턴 공격
    - 키가 있는 전치 암호로 생성된 암호문은 어떤 반복되는 패턴을 가짐

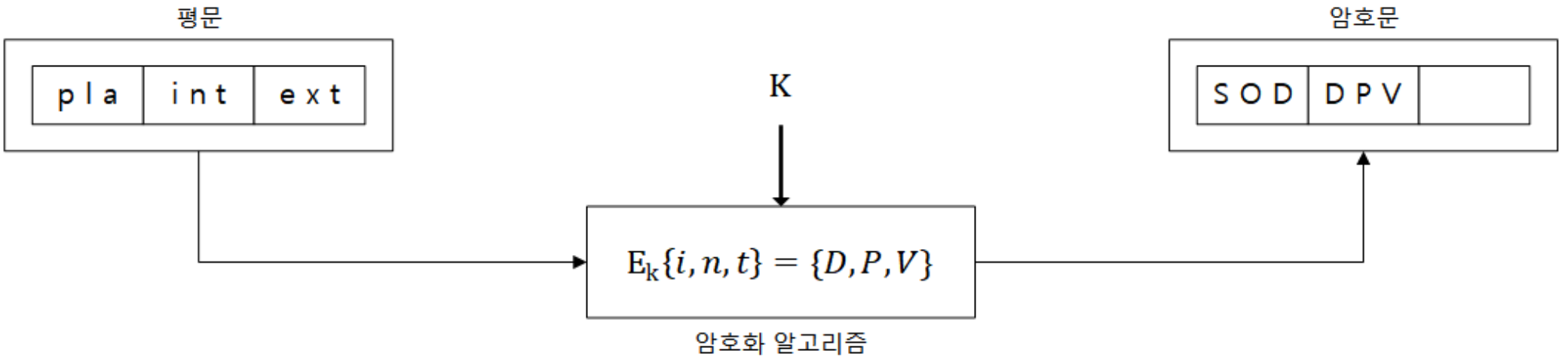
스트림 암호와 블록 암호

- 스트림 암호(Stream Cipher)
  - 정의
    - 평문을 한 비트 또는 한 바이트 단위로 암호화하는 방식
  - 특징
    - 평문 문자는 한 번에 하나씩 암호화되어 하나의 암호문 문자를 생성함



스트림 암호와 블록 암호

- 블록 암호(Block Cipher)
  - 정의
    - 평문을 고정된 크기의 블록으로 나눠 암호화하는 방식
  - 특징
    - 블록으로 묶인 문자들이 함께 암호화됨
    - 블록의 크기를 맞추기 위해 패딩이 사용될 수 있음

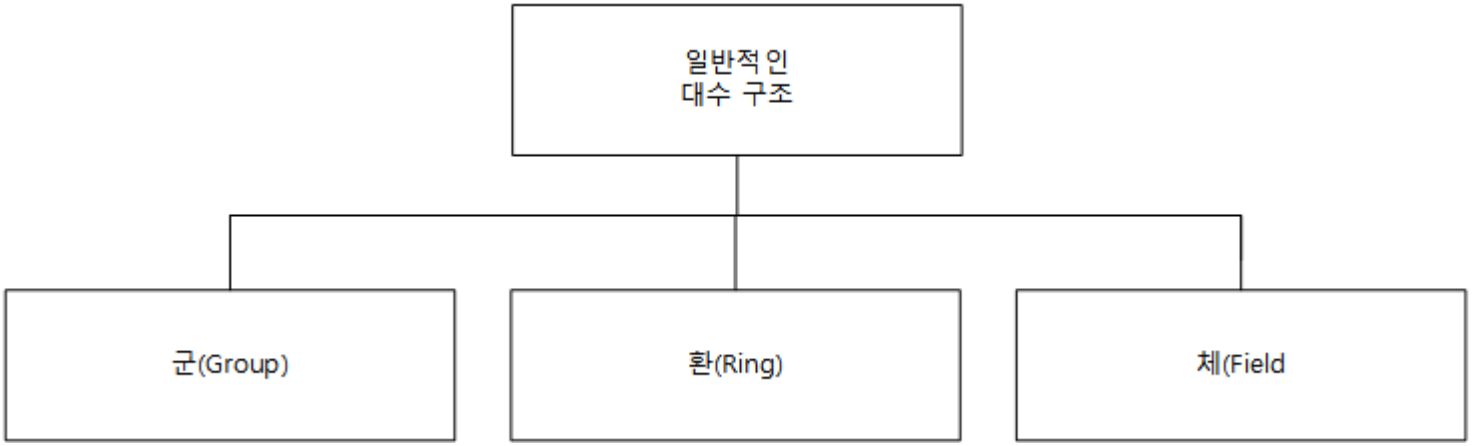


목 차

- 고전 대칭-키 암호
  - 개요
  - 대치 암호(Substitution Ciphers)
  - 전치 암호(Transposition Cipher)
  - 스트림 암호(Stream Cipher)와 블록 암호(Block Cipher)
- 암호 수학
  - 대수 구조
  - $GF(2^n)$ 체

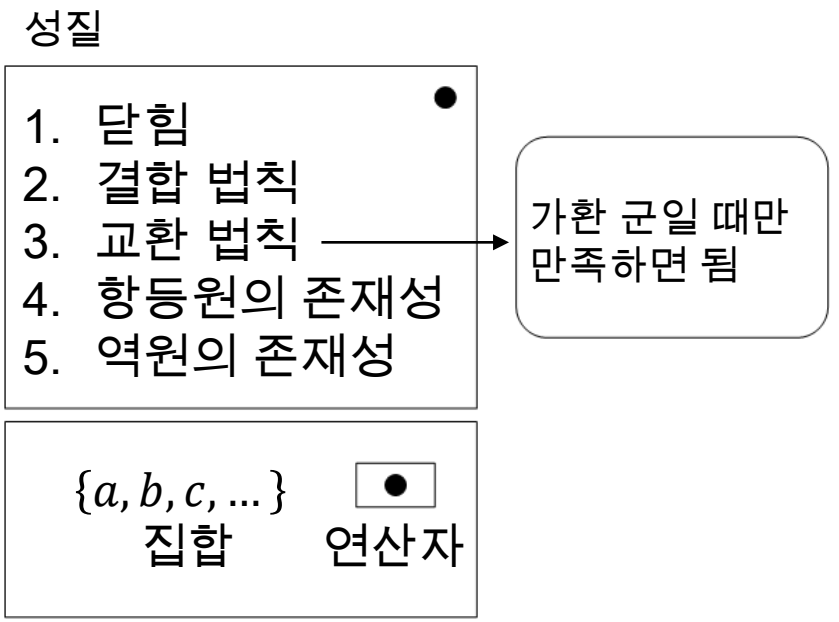
대수 구조(Algebraic Structure)

- 정의
  - 집합과 그 집합에 정의된 연산들로 이루어진 구조
- 종류
  - 군 (G, Group)
  - 환 (Ring)
  - 체 (Field)



대수 구조

- 군(G, Groups) (1/8)
  - 정의
    - 집합 위에 정의된 연산이 군의 성질을 만족하는 대수 구조
  - 성질
    - 닫힘
      - $\forall a, b \in G \rightarrow a \bullet b \in G$
    - 결합 법칙
      - $\forall a, b, c \in G \rightarrow (a \bullet b) \bullet c = a \bullet (b \bullet c)$
    - 교환 법칙
      - $\forall a, b \in G \rightarrow a \bullet b = b \bullet a$
    - 항등원의 존재성
      - $\forall a \in G \rightarrow e \bullet a = a \bullet e = a$
    - 역원의 존재성
      - $\forall a \in G \rightarrow a \bullet a' = a' \bullet a = e$



대수 구조

- 군(G, Groups) (2/8)
  - 특징
    - 하나의 군에는 하나의 연산만 정의하지만 정의된 연산과 역함수 관계인 연산의 사용은 허용됨
      - e.g., $G =< Z_{26}, + >$에서 정의된 연산은 덧셈이지만 뺄셈도 허용됨
    - 군의 성질만 만족한다면 반드시 원소가 숫자이거나 연산이 사칙 연산일 필요는 없음
      - e.g., 군의 성질을 만족하는 $G =< \{a, b, c, d\}, \bullet >$를 정의할 수 있음

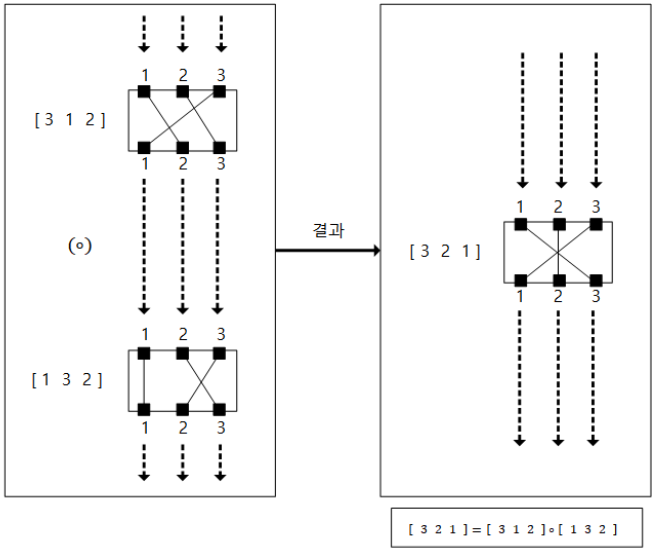
$\cdot$	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

대수 구조

- 군(G, Groups) (3/8)

*합성
하나의 치환을 수행 후 다음 치환을 수행하는 연산

  - 치환 군(Permutation Group)
    - 정의
      - 치환으로 이루어진 집합과 두 치환을 *합성하는 연산으로 이루어진 군
    - 특징
      - 교환법칙을 만족하지 않으므로 가환 군이 아님
      - 치환을 연속하는 것으로는 암호의 안전성을 강화할 수 없음



∘	[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

- 군(G, Groups) (4/8)
  - 유한 군(Finite Group)
    - 유한 개의 원소를 가지고 있는 군
      - e.g., 군 G =< Z_n,+>, 군 G =< Z_n^*,×> 등
  - 무한 군(Infinite Group)
    - 무한 개의 원소를 가지고 있는 군
      - e.g., 정수 덧셈 군 G =< Z,+>, 실수 곱셈 군 G =< R,×> 등
  - 위수(Order)
    - 군의 위수 $|G|$는 군에 있는 원소의 개수
      - e.g., G =< Z_{26},+>에서 G의 위수 $|G| = 26$

대수 구조

- 군(G, Groups) (5/8)
  - 부분군(Subgroup)
    - 정의
      - 군 G의 부분집합 중 상위군(Supergroup)의 연산에 대해 군이 되는 군 H
    - 성질
      - $\forall a, b \in H, G \rightarrow a \bullet b \in H, G$
      - $e_H = e_G$
      - $\forall a \in H, G \rightarrow a^{-1} \in H, G$
      - $H =< \{e\}, \bullet > \rightarrow H \leq G$
      - $G \leq G$
    - 특징
      - $H \leq G$로 표기함
      - 상위군과 같은 연산이 정의되어야 함
        - e.g., $G_1 =< Z_{10}, + >$과 $G_2 =< Z_{12}, + >$는 같은 덧셈 연산처럼 보이지만 G_1은 $mod\ 10$에서의 덧셈 연산이고 G_2는 $mod\ 12$에서의 덧셈 연산임

대수 구조

- 군(G, Groups) (6/8)

*멱승(Power)
원소에 군의 연산을 반복적으로 적용한 것

  - 순환 부분군(Cyclic Subgroup)
    - 정의
      - 군의 부분군 중 한 원소의 *멱승(Power)을 사용하여 생성된 부분군
    - 특징
      - $a^n \rightarrow a \cdot a \cdot \cdots \cdot a(n$번$)$
      - $a^0 = e$
      - 집합 내 중복 원소는 하나의 원소로 취급함

대수 구조

- 군(G, Groups) (7/8)
  - 순환 군(Cyclic Group)
    - 정의
      - 그 자신이 하나의 순환 부분군인 군
    - 특징
      - *생성원 g에 대해 유한 순환 군의 원소 집합 $\{e, g^1, g^2, \ldots, g^{n-1}\}$, $g^n = e$으로 표현됨
      - 하나의 순환 군은 많은 생성원을 가질 수 있음

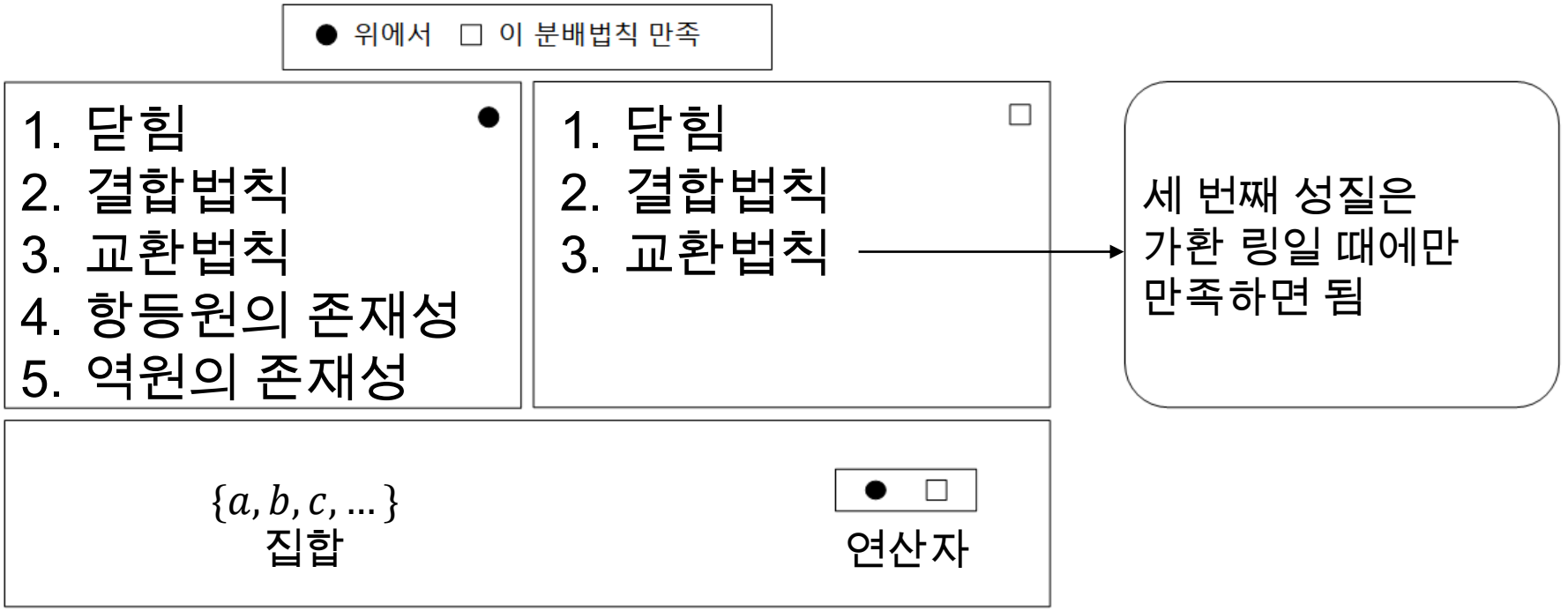
*생성원(Generator)
군 자신을 생성하는 원소

대수 구조

- 군(G, Groups) (8/8)
  - 라그랑지 정리(Lagrange's Theorem)
    - 군의 위수는 부분군의 위수로 나누어 떨어지는 원칙
      - e.g., $G =< Z_{17},+>$의 위수는 1,17이므로 1과 17을 위수로 가지는 부분군이 생성될 수 있음
  - 원소의 위수
    - 군에서 $a^n = e$를 만족하는 가장 작은 정수 n
    - $n = ord(a)$
      - e.g., $G =< Z_{10}{}^*,\times>$에서 $ord(1) = 1, ord(3) = 4, ord(7) = 4, ord(9) = 2$

대수 구조

- 환(Ring) (1/2)
  - 정의
    - 집합과 그 집합 위에 두 가지 연산이 정의되어 일정한 성질을 만족하는 대수 구조



대수 구조

- 환(Ring) (2/2)
  - 특징
    - 첫 번째 연산은 아벨 군에 대해 요구되는 다섯 가지 성질을 반드시 만족해야 함
    - 두 번째 연산은 첫 번째 연산 위에 분배 법칙(Distributivity)을 만족해야 함
    - 환에서 두 번째 연산은 항등원과 역원이 없을 수 있음
      - 예제
  - 가환 환(Commutative Ring)
    - 두 번째 연산에 대해 교환 법칙을 추가로 만족하는 환

대수 구조

- 체(Field) (1/3)

*0원소
첫 번째 연산의 항등원

  - 정의
    - 두 번째 연산에서 *0원소를 제외한 원소들이 역원을 가지는 가환 환

● 위에서 □ 이 분배법칙 만족

1. 닫힘
2. 결합법칙
3. 교환법칙
4. 항등원의 존재성
5. 역원의 존재성

●

1. 닫힘
2. 결합법칙
3. 교환법칙
4. 항등원의 존재성
5. 역원의 존재성

□

첫 번째 연산의 항등원은 두 번째 연산에서 역원을 가지지 않음

$\{a, b, c, \ldots\}$
집합

● □
연산자

대수 구조

- 체(Field) (2/3)
  - 유한 체(Finite Field)
    - 정의
      - 유한 개의 원소를 갖는 체
    - 특징
      - 갈로아 체(Galois Field)라고도 불리며 소수 p의 n 거듭제곱꼴의 원소 개수를 가짐
    - 갈로아 체(Galois Field)
      - 표기
        - $GF(p^n)$
      - 정의
        - p^n개의 원소를 갖는 유한 체

대수 구조

- 체(Field) (3/3)
  - $GF(p)$ 체
    - $n = 1$일 때의 갈로아 체
      - $GF(2)$ 체
        - 특징
          - 이진수 또는 비트로 표현되는 두 개의 원소만 가짐
          - 덧셈 연산은 이진수에서 사용하는 XOR (Exclusive-or) 연산
          - 곱셈 연산은 이진수에서 사용하는 AND 연산
          - 덧셈과 뺄셈 연산은 같음
          - 곱셈과 나눗셈 연산은 같음

$GF(2)$

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

$+$	0	1
0	0	1
1	1	0

덧셈

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

곱셈

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

역원

$GF(2^n)$ 체

- 개요
  - 암호학에서는 사칙연산이 필요한 경우가 많으므로 체를 주로 사용함
  - 컴퓨터로 연산 시 양의 정수는 n-비트 워드로 저장됨

대수 구조	일반적인 연산	일반적인 집합
군(Group)	$(+,-)$ 또는 $(\times,\div)$	Z_n 또는 $Z_n{}^*$
환(Ring)	$(+,-)$와 $(\times)$	Z
체(Field)	$(+,-)$와 $(\times,\div)$	Z_p

$GF(2^n)$ 체

- 컴퓨터에서의 $GF(2^n)$ 체 사용 방법 (1/2)
  - 2^n보다 작은 가장 큰 소수 p를 이용해서 Zp에 정의된 $GF(p)$를 사용함
    - p부터 $2^n - 1$까지의 정수를 사용할 수 없기 때문에 비효율적임
      - e.g., 4-비트일 때 GF(13)을 쓰면 13,14,15을 사용할 수 없음
  - 원소의 개수가 2^n인 $GF(2^n)$체를 사용함
    - 일반적인 4개의 연산들이 적용될 수 없음
      - e.g., 4-비트일 때 GF(2^4)에서 체의 성질을 만족하지 못하는 경우가 생김

$GF(2^n)$ 체

- 컴퓨터에서의 $GF(2^n)$ 체 사용 방법 (2/2)
  - 체의 성질을 만족하는 새로운 연산 정의
    - e.g., $F =< \{00,01,10,11\}, \oplus, \otimes >$로 정의된 체

덧셈

⊕	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

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

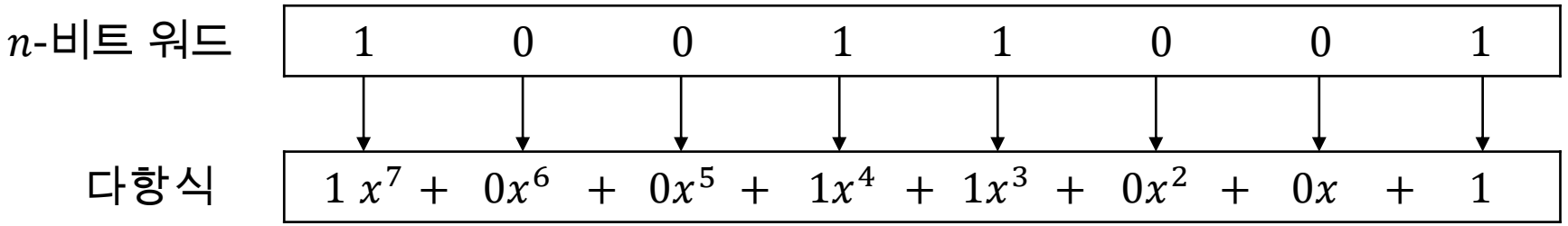
항등원 : 01

$GF(2^n)$ 체

- 다항식(Polynomial)
  - n-비트 워드들을 차수 $n-1$의 다항식 형태로 간단하게 표현
  - $f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x^1 + a_0x^0$로 표현
    - $x_i = i$번째 항(Term)
    - $a_i = i$번째 항의 계수(Coefficient)
  - 규칙
    - x의 지수 승은 n비트 워드에서 비트들의 위치를 정의
      - 가장 왼쪽은 최상위 비트
      - 가장 오른쪽은 최하위 비트
    - 항의 계수는 비트들의 값으로써 정의
      - 0 또는 1의 값

$GF(2^n)$ 체

- 다항식(Polynomial)
  - e.g., 8-비트 워드(10011001)의 다항식 표현



1차 단순화 $1x^7 + 1x^4 + 1x^3 + 1x^0$

2차 단순화 $x^7 + x^4 + x^3 + 1$

$GF(2^n)$ 체

- 다항식(Polynomial)
  - 연산
    - 계수 연산을 위해 $GF(2)$ 체를 사용함
    - 다항식 연산을 위해 $GF(2^n)$체를 사용함
  - 모듈로 (1/2)
    - 곱셈 연산은 $n-1$보다 큰 차수를 갖는 다항식을 생성할 수 있으므로 곱셈한 결과를 모듈로 다항식으로 나누고 나머지를 구해야 함
    - 각 차수에 대해서 기약 다항식이 하나 이상 존재하며 $GF(2^n)$을 정의할 때 모듈로로 사용할 기약 다항식을 반드시 정의해야 함

$GF(2^n)$ 체

- 다항식(Polynomial)
  - 모듈로 (2/2)
    - 기약 다항식
      - 소수 다항식(Prime Polynomial)
      - n보다 작은 차수를 가진 다항식으로 분해될 수 없는 다항식

차수	기약 다항식
1	$(\mathrm{x}+1), (\mathrm{x})$
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

- 다항식(Polynomial)
  - 덧셈
    - $GF(2)$에 대응하는 항의 계수들의 합
    - 차수가 $n-1$인 두 개의 다항식들의 합은 항상 차수 $n-1$의 다항식을 생성
    - 항등원
      - 다항식 자신을 더하는 것은 0의 다항식을 만듦
      - 0의 다항식(모든 계수가 0인 다항식)임
    - 역원
      - $GF(2)$에서 계수를 가지는 다항식의 덧셈에 대한 역원은 자신임
      - 다항식의 덧셈과 뺄셈은 같은 연산임

$GF(2^n)$ 체

- 다항식(Polynomial)
  - 곱셈 (1/2)
    - 첫 번째 다항식의 각 항과 두 번째 다항식의 각 항의 곱셈의 합
    - 곱셈은 $n-1$보다 큰 차수를 가지는 항을 생성 가능
      - 모듈로 다항식을 사용하여 곱셈한 결과 약분 필요
        - e.g., $(x^8+x^4+x^3+x+1)$을 기약다항식으로 가지는 $GF(2^8)$에서 $(x^5+x^2+x)\otimes(x^7+x^4+x^3+x^2+x)$의 계산

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

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

$GF(2^n)$ 체

- 다항식(Polynomial)
  - 곱셈 (2/2)
    - 항등원
      - 항상 1임
    - 역원
      - 확장 유클리드 알고리즘이 반드시 모듈로와 다항식에 적용돼야 함
        - e.g., $GF(2^4)$에서 모듈로 x^4+x+1의 (x^2+1)의 역원 구하기

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
(x^2+1)	(x^4+x+1)	(x^2+1)	(x)	(1)	(0)	(1)	(0)	(1)	(x^2+1)
(x)	(x^2+1)	(x)	(1)	(0)	(1)	(x)	(1)	(x^2+1)	(x^3+x+1)
(x)	(x)	(1)	(0)	(1)	(x)	(x^2)	(x^2+1)	(x^3+x+1)	(0)

$GF(2^n)$ 체

- 다항식(Polynomial)
  - 컴퓨터를 이용하는 곱셈 (1/2)
    - 컴퓨터의 구현에서는 줄여진 다항식에 x를 반복적으로 곱하는 알고리즘을 사용함
      - e.g., $(x^8 + x^4 + x^3 + x + 1)$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

멱승	연산	결과	약분 여부
$x^0 \otimes P_2$		$x^7 + x^4 + x^3 + x^2 + x$	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

- 다항식(Polynomial)
  - 컴퓨터를 이용하는 곱셈 (2/2)
    - 알고리즘 구성
      - 이전 결과의 최상위 비트가 0이라면 이전 결과를 왼쪽으로 한 비트 이동
      - 이전 결과의 최상위 비트가 1이면 왼쪽으로 한 비트 이동하고 최상위 비트를 제외한 후 모듈로 다항식과 XOR 연산
        - e.g., $(x^8+x^4+x^3+x+1)$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

멱승	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
$P_1 \otimes P_2 = (00100111)\oplus(01001110)\oplus(01000110) = 00101111$		

$GF(2^n)$ 체

- 생성원의 사용 (1/5)
  - g가 체의 생성원이라면 $f(g) = 0$
  - $\{0, g^0, g^1, g^2, \ldots, g^N\}, N = 2^n - 2$

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

- 생성된 $GF(2^n)$의 원소 (1/2)

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)

$GF(2^n)$ 체

- 생성원의 사용 (3/5)
  - 생성된 $GF(2^n)$의 원소 (2/2)

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

- 생성원의 사용 (4/5)
  - 역원
    - 덧셈에 대한 역원
      - 원소 자신임
        - e.g., $g + g = 0$
        - e.g., $-g^3 = g^3 = (1000)$
    - 곱셈에 대한 역원
      - 멱승(Power)을 모듈로 $2^n - 1$로 계산함
        - e.g., $(g^a)^{-1} = g^{-a \, mod \, 2^n - 1}$
        - e.g., $(g^3)^{-1} = g^{-3} = g^{12} = g^3 + g^2 + g + 1 = (1111)$

$GF(2^n)$ 체

- 생성원의 사용 (5/5)
  - 연산
    - 덧셈과 뺄셈
      - 동일한 연산임
        - e.g., $g^3 + g^{12} + g^7 = g^3 + (g^3 + g^2 + g + 1) + (g^3 + g + 1) = g^3 + g^2 = (1100)$
        - e.g., $g^3 - g^6 = g^3 + g^6 = g^3 + (g^3 + g^2) = g^2 = (0100)$
    - 곱셈과 나눗셈
      - 곱셈은 멱승의 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!

이 성 현(dltjdgus2166@naver.com)

부록#1

- 알파벳 출현 빈도수(%) 표

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		

- 빈도 높은 두 문자열(Diagrams), 세 문자열(Trigrams)

종류	빈도 높은 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, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH

부록#2

- Vigenere 표

	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