

# 암호학과 네트워크 보안

- 2, 3부 암호 수학, 고전 대칭-키 암호화-

이 하 늘([haneul@pel.sejong.ac.kr](mailto:haneul@pel.sejong.ac.kr))

세종대학교 프로토콜공학연구실

# 목 차

---

- 암호 수학

- 정수 연산
- 모듈로 연산
- 행렬

- 고전 대칭-키 암호

- 개요
- 대칭 암호
- 전치 암호
- 스트림 암호와 블록 암호

# 암호 수학

---

- 정수 연산

- 정의

- 양의 정수, 음의 정수, 0으로 이루어진 수의 집합
- $Z = \{..., -2, -1, 0, 1, 2, ...\}$

- 2진 연산

- 두 개의 입력 값에 대하여 하나의 결과 값을 산출

e.g., 덧셈, 뺄셈, 곱셈

- 나눗셈은 하나가 아닌 두 개의 값을 산출하기 때문에 속하지 않음

# 암호 수학

---

- 정수 연산
- 정수의 나눗셈(나눗셈 관계식)
  - $a = q \times n + r$
  - $a$ 는 피제수,  $q$ 는 몫,  $n$ 은 제수,  $r$ 은 나머지
  - 제한 사항
    - $n$ 은 양의 정수( $n > 0$ )
    - 나머지는 양의 정수( $r \geq 0$ )

# 암호 수학

---

- 정수 연산

- 가분성(divisibility)

- $a = q \times n$
- 'n이 a를 나눈다', n은 a의 약수,  $n|a$
- 나머지가 0이 아니면,  $n \nmid a$
- 성질
  - 성질 1: 만약  $a|1$  이면,  $a = \pm 1$
  - 성질 2: 만약  $a|b$  이고,  $b|a$  이면  $a = \pm b$
  - 성질 3: 만약  $a|b$  이고,  $b|c$  이면  $a|c$
  - 성질 4: 만약  $a|b$  이고,  $a|c$  이면  $a|(m \times b + n \times c)$ , 여기서 m과 n은 임의의 정수

# 암호 수학

- 정수 연산

- 가분성(divisibility)

  - 종류

  - 약수

    - 어떤 수나 식을 나누어 나머지가 없이 나누어 떨어지는 수

  - 사실

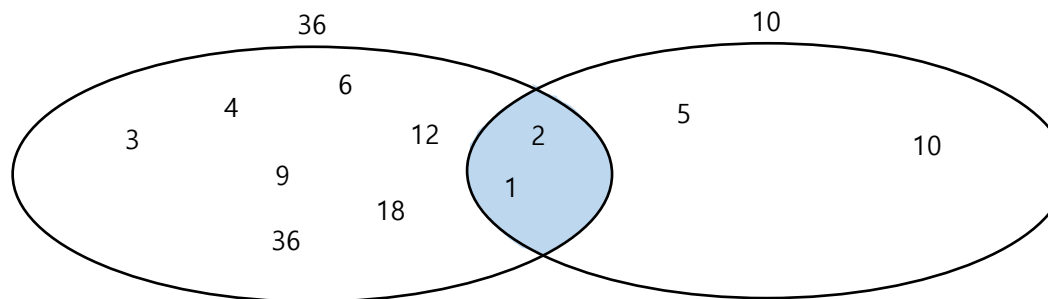
    - 사실 1: 정수 1은 하나의 약수, 1만을 갖는다.

    - 사실 2: 모든 양의 정수는 최소 2개의 약수(1과 자신)를 가진다.

  - 최대 공약수

    - 두 자연수의 공통된 약수 중 가장 큰 정수

      - e.g., 36과 10의 최대공약수 2



# 암호 수학

- 정수 연산

- 가분성(divisibility)

- 최대 공약수

- 유클리드 알고리즘(Euclidean Algorithm)

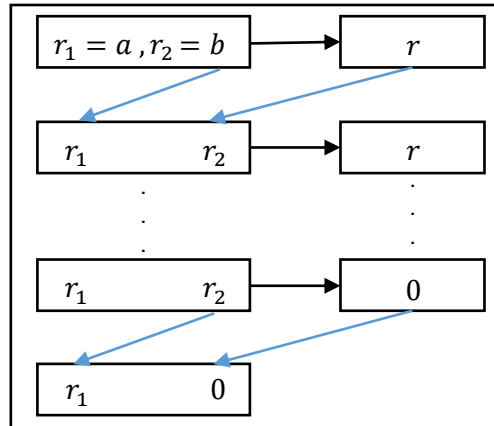
- 두 양의 정수의 최대 공약수를 찾아내는 알고리즘

- 사실

- 사실 1:  $\gcd(a, 0) = a$

- 사실 2:  $\gcd(a, b) = \gcd(b, r)$ , 이때  $r$ 은  $a$ 를  $b$ 로 나눈 나머지

- $\gcd(36, 10) = \gcd(10, 6) = \gcd(6, 4) = \gcd(4, 2) = \gcd(2, 0) = 2$



- $\gcd(a, b) = 1$ 이면,  $a$ 와  $b$ 는 서로소(relatively prime)

# 암호 수학

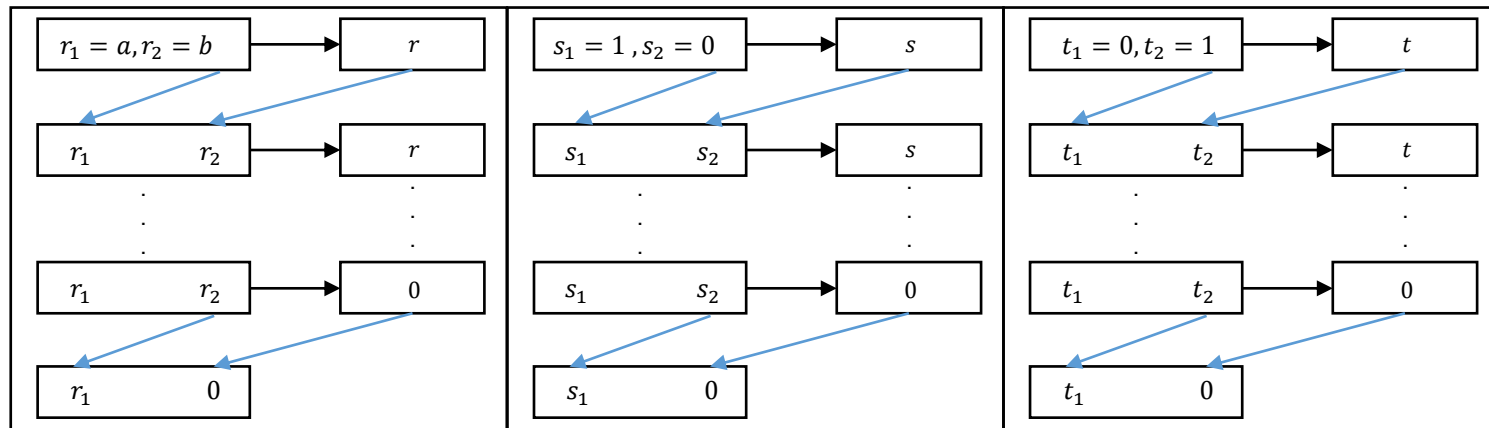
- 정수 연산

- 가분성(divisibility)

  - 최대 공약수

  - 확장 유클리드 알고리즘(Extended Euclidean Algorithm)

    - 최대 공약수를 구하기 위해 이루고 있는 방정식의 해를 구하기 위함
    - $s \times a + t \times b = \gcd(a, b)$
    - $\gcd(a, b)$ 를 계산하고 동시에  $s$ 와  $t$  값 계산 가능





# 암호 수학

- 정수 연산

- 확장 유클리드 알고리즘의 활용

- 선형 디오판투스 방정식( $ax+by=c$ )

- 정의

- 정수를 계수로 갖는 부정 다항 방정식

- 특수 해( $d|c$ 이면)

- 1. 유클리드 호제법으로 최대 공약수를 구함
    - 2. 유클리드 호제법을 역순 전개하여 디오판투스 방정식의 특수 해를 구함

$$x_0 = \left(\frac{c}{d}\right)s, y_0 = \left(\frac{c}{d}\right)t$$

- 일반 해

- $x = x_0 + k\left(\frac{b}{d}\right), y = y_0 - k(b/d)$  이때,  $k$ 는 정수

# 암호 수학

---

- 정수 연산

- 선형 디오판투스 방정식

- 예시( $36x+10y=2$ 의 일반해 구하기)

- 유클리드 호제법으로 최대 공약수를 구함

- $36=10\times 3+6$

- $10=6\times 1+4$

- $6=4\times 1+2$

- $4=2\times 2+0$

- 유클리드 호제법을 역순 전개하여 디오판투스 방정식의 특수 해 구함

- $2=6-4$

- $=6-(10-6)$

- $=2\times 6-10$

- $=2\times(36-10\times 3)-10$

- $=2\times 36-7\times 10$

- $x=2, y=-7$

# 암호 수학

---

- 정수 연산

- 선형 디오판투스 방정식

- 예시( $36x+10y=2$ 의 일반해 구하기)

- 유클리드 호제법을 역순 전개하여 디오판투스 방정식의 특수 해 구함
    - 디오판투스 방정식 일반 해 구하는 공식에 대입
      - $x = 2 + 5k$
      - $y = -7 - 18k$

# 암호 수학

---

- 모듈로 연산

- $a \bmod n = r$
- $n$ 은 모듈로,  $r$ 은 나머지  
e.g.,  $27 \bmod 5$ ,  $36 \bmod 12$

- 잉여류  $Z_n$

- 모듈로  $n$ 의 결과 값의 최소 잉여 집합
  - $a \bmod n$ 의 결과값은 항상  $n$ 보다 작은 음이 아닌 정수
  - $Z_n = \{0, 1, 2, \dots, (n - 1)\}$

# 암호 수학

---

- 모듈로 연산

- 합동

- 어떤 정수로 나눈 나머지가 서로 같은 두 정수 사이의 관계  
e.g.,  $2 \bmod 10 = 2$ ,  $12 \bmod 10 = 2$ ,  $22 \bmod 10 = 2$

- 합동 연산자( $\equiv$ ) 사용

e.g.,  $2 \equiv 12 \pmod{10}$ ,  $13 \equiv 23 \pmod{10}$

- 요점

- 등식 연산자와의 다른 점

- 등식 연산자는  $Z$ 의 원소를  $Z$ 의 원소로 대응하지만, 합동 연산자는  $Z$ 의 원소에서  $Z_n$ 의 원소로 대응
- 등식 연산자는 일대일이지만 합동 연산자는 다대일

- $(\bmod n)$ 은  $Z_n$ 을 나타냄

# 암호 수학

---

- 모듈로 연산

- 합동

- 잉여류(residue classes)

- 잉여류  $[a]$ ,  $[a]n$ 은 모듈로  $n$ 으로 합동인 정수의 집합

e.g.,  $n=5$ 라면

$$[0] = \{\dots, -15, -10, -5, 0, 5, 10, 15, \dots\}$$

$$[1] = \{\dots, -14, -9, -4, 1, 6, 11, 16, \dots\}$$

$$[2] = \{\dots, -13, -8, -3, 2, 7, 12, 17, \dots\}$$

$$[3] = \{\dots, -12, -7, -2, 3, 8, 13, 18, \dots\}$$

$$[4] = \{\dots, -11, -6, -1, 4, 9, 14, 19, \dots\}$$

- 최소 잉여

- $Z_n$ 은 모듈로  $n$ 의 모든 최소 잉여의 집합

e.g.,  $Z_5 = \{0, 1, 2, 3, 4\}$

# 암호 수학

---

- 모듈로 연산

- $Z_n$ 에서의 연산

- 성질

- 성질 1:  $(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$
    - 성질 2:  $(a - b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$
    - 성질 3:  $(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$

# 암호 수학

---

- 모듈로 연산

- 역원

- 덧셈에 대한 역원

- $a+b \equiv 0 \pmod n$
    - $Z_n$ 에서  $a$ 의 덧셈에 대한 역원은  $b = n - a$
    - 어떤 정수와 그 정수의 덧셈에 대한 역원의 합은 모듈로  $n$ 에 대하여 0과 합동

- 곱셈에 대한 역원

- $a \times b \equiv 1 \pmod n$   
e.g.,  $n=10$ 이면 3의 곱셈에 대한 역원은 7
    - 곱셈에 대한 역원은 있을 수도 있고 없을 수도 있음



# 암호 수학

---

- 모듈로 연산
- 덧셈과 곱셈에 대한 다른 집합
  - $Z_n^*$ 은  $Z_n$ 의 부분 집합
  - $Z_n^*$ 은  $Z_n$ 에 속한 정수 중 곱셈에 대한 역원을 가진 원소의 집합
  - 덧셈에 대한 역원이 필요할 때는  $Z_n$
  - 곱셈에 대한 역원이 필요할 때는  $Z_n^*$

# 암호 수학

- 행렬

- 정의

- 행렬은  $l \times m$ 개의 원소를 갖는 직사각형 배열
- $l$ 은 행의 개수,  $m$ 은 열의 개수
- 행 행렬
  - 행렬이 하나의 행만 갖는 경우
- 열 행렬
  - 행렬이 하나의 열만 갖는 경우

$$\text{Matrix } A: \begin{matrix} & \text{M columns} \\ \begin{matrix} \text{L rows} \\ \left[ \begin{array}{ccc} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{l1} & \cdots & a_{lm} \end{array} \right] \end{matrix} \end{matrix}$$

# 암호 수학

- 행렬

- 연산과 관계식

- 등식

- 두 행렬의 행과 열의 개수와 대응되는 원소가 동일하면 두 행렬은 동일
    - 모든  $i$ 와  $j$ 에 대해  $a_{ij} = b_{ij}, A = B$

- 덧셈과 뺄셈

- 두 행렬의 행과 열의 개수가 같으면 덧셈과 뺄셈 가능
    - $\begin{bmatrix} 12 & 4 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 3 & 2 \end{bmatrix} + \begin{bmatrix} 7 & 2 \\ 8 & 10 \end{bmatrix}$
    - $\begin{bmatrix} -2 & 0 \\ -5 & -8 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 3 & 2 \end{bmatrix} - \begin{bmatrix} 7 & 2 \\ 8 & 10 \end{bmatrix}$

# 암호 수학

---

- 행렬

- 연산과 관계식

- 곱셈

- 첫 번째 행렬의 열의 개수가 두 번째 행렬의 행의 개수와 같을 경우

- $[53] = [5 \ 2 \ 1] \times \begin{bmatrix} 7 \\ 8 \\ 2 \end{bmatrix}$

- $53 = 5 \times 7 + 2 \times 8 + 1 \times 2$

- 스칼라 곱

- $\begin{bmatrix} 15 & 6 \\ 9 & 6 \end{bmatrix} = 3 \times \begin{bmatrix} 5 & 2 \\ 3 & 2 \end{bmatrix}$

# 암호 수학

---

- 행렬

- 행렬식

- $M=1$  이면  $\det(A)=a_{11}$

- $M>1$  이면  $\det(A)=\sum_{i=1}^m (-1)^{i+j} \times a_{ij} \times \det(A_{ij})$

e.g.,  $\det \begin{bmatrix} 5 & 2 \\ 3 & 4 \end{bmatrix} = (-1)^{1+1} \times 5 \times \det[4] + (-1)^{1+2} \times 2 \times \det[3] = 14$

# 목 차

---

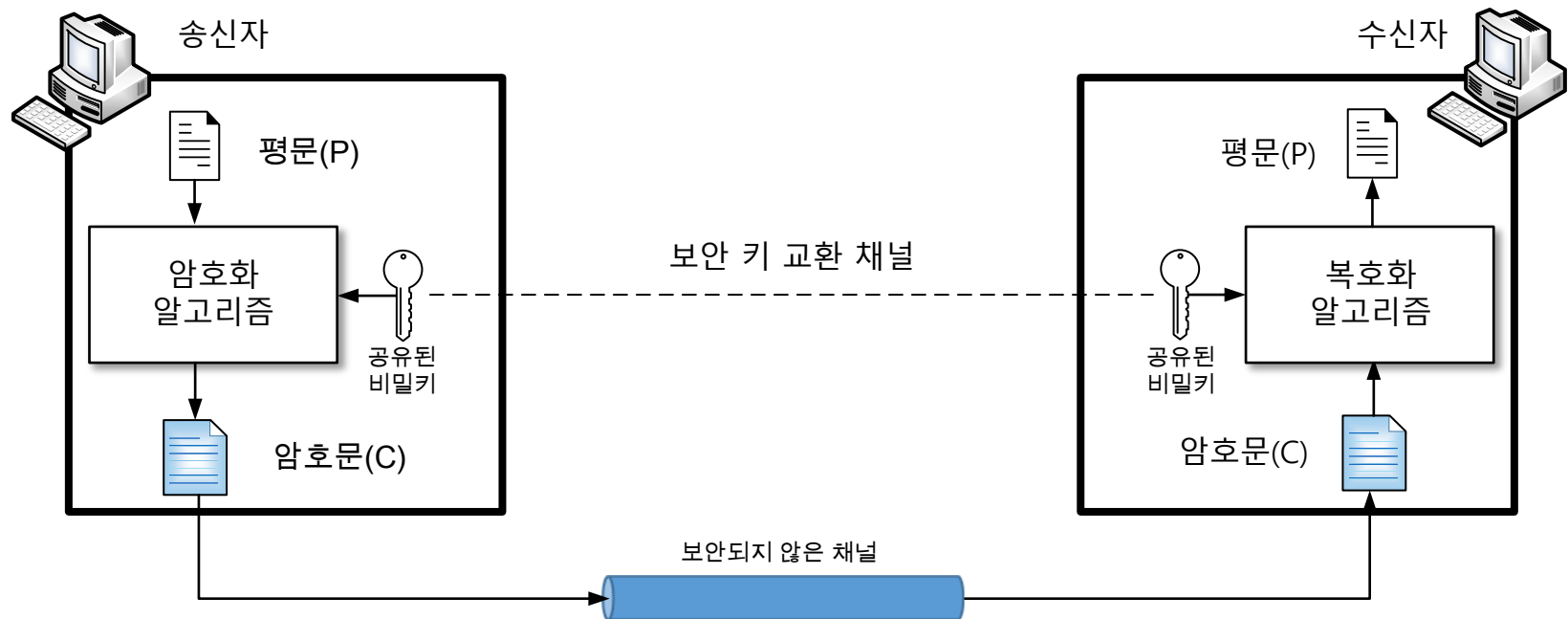
- 암호 수학
  - 정수 연산
  - 모듈로 연산
  - 행렬
- 고전 대칭-키 암호
  - 개요
  - 대치 암호
  - 전치 암호
  - 스트림 암호와 블록 암호

# 고전 대칭-키 암호

- 개요

- 대칭 키 암호화

- 암호/복호화에 동일한 한 개의 키를 사용하는 알고리즘



# 고전 대칭-키 암호

- 개요

- 대칭 키 암호화

- 암호화:  $C = E_k(P)$
- 복호화:  $P = D_k(C)$
- $D_k(E_k(P)) = E_k(D_k(P)) = P$



- Kerckhoff의 원리

- 암호의 안전성은 암호/복호 알고리즘의 비밀을 지키는데 기반을 두어서는 안되고, 키의 비밀성에 기반을 두어야함



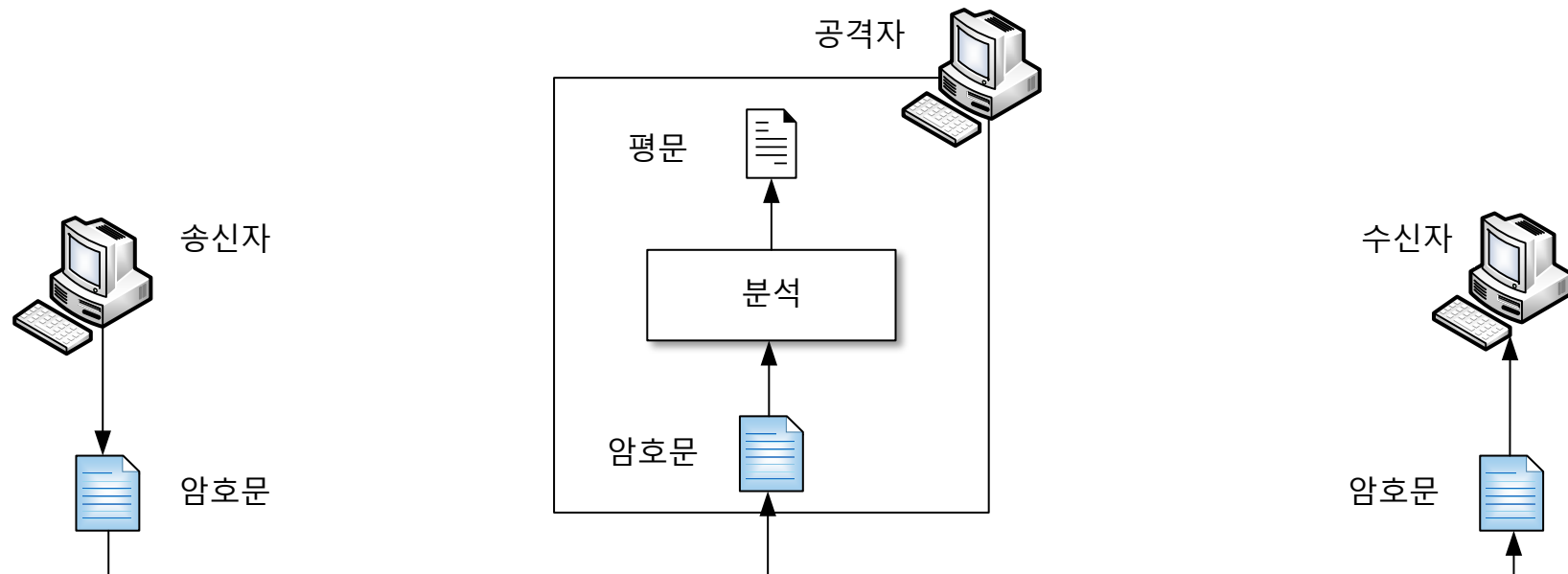
# 고전 대칭-키 암호

- 개요

- 암호 해독

- 암호문 단독 공격

- 오로지 암호문만을 얻어 공격하는 기술



# 고전 대칭-키 암호

---

- 개요
- 암호 해독
  - 암호문 단독 공격
    - 전수조사 공격
      - 모든 가능한 키를 사용하는 방법
      - 의미 있는 평문을 얻을 때까지 반복
      - 이를 막기 위해서 가능한 키의 수가 매우 커야함
    - 통계적인 공격
      - 평문 언어의 고유한 특징으로부터 정보를 얻어 통계를 통한 공격 수행
      - 이를 막기 위해서는 암호문이 평문 언어의 특징을 드러내지 않아야 함
    - 패턴 공격
      - 평문 언어의 특징을 드러내지 않지만 암호문에 특정 패턴이 존재할 경우
      - 이를 막기 위해선 암호문을 랜덤하게 보이도록 만드는 암호를 사용

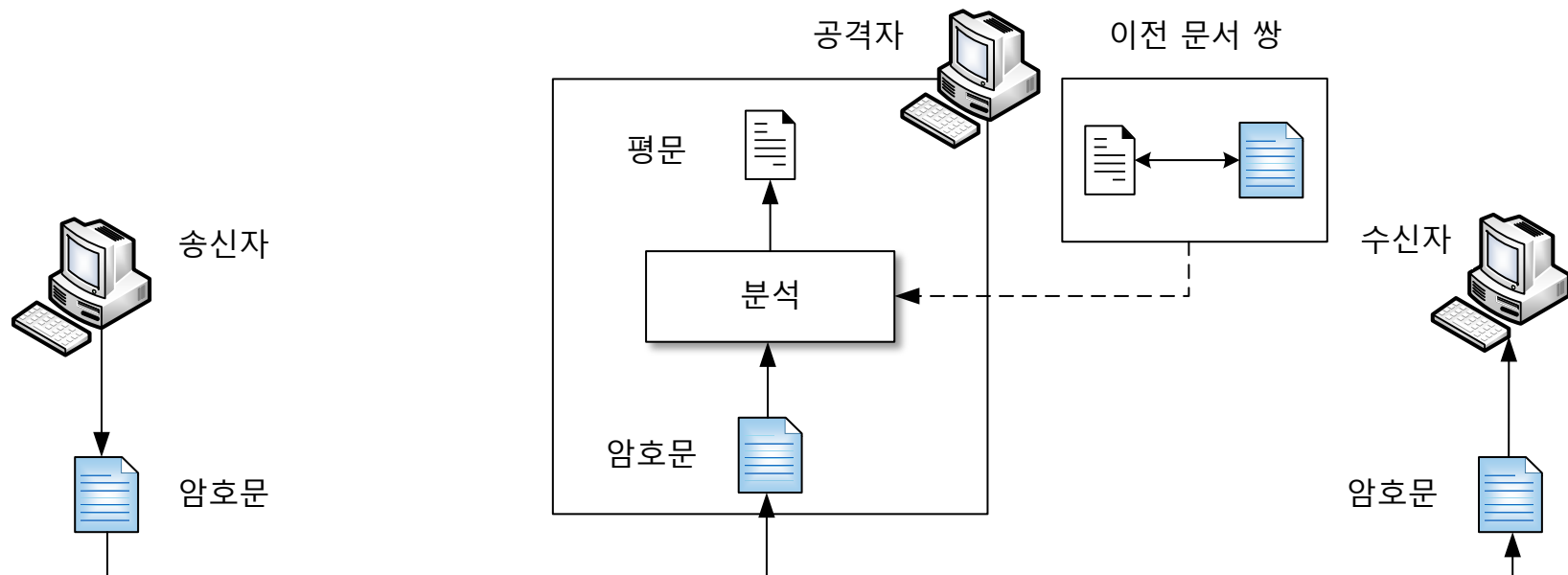
# 고전 대칭-키 암호

- 개요

- 암호 해독

- 알려진 평문 공격

- 주어진 평문/암호문 쌍의 연관성을 이용하여 다음 암호문 해독
    - 키를 변경하거나, 과거에 보낸 메시지를 노출하지 않을 수도 있기 때문에 적용 가능한 상황이 암호문 단독 공격보다 드뭄



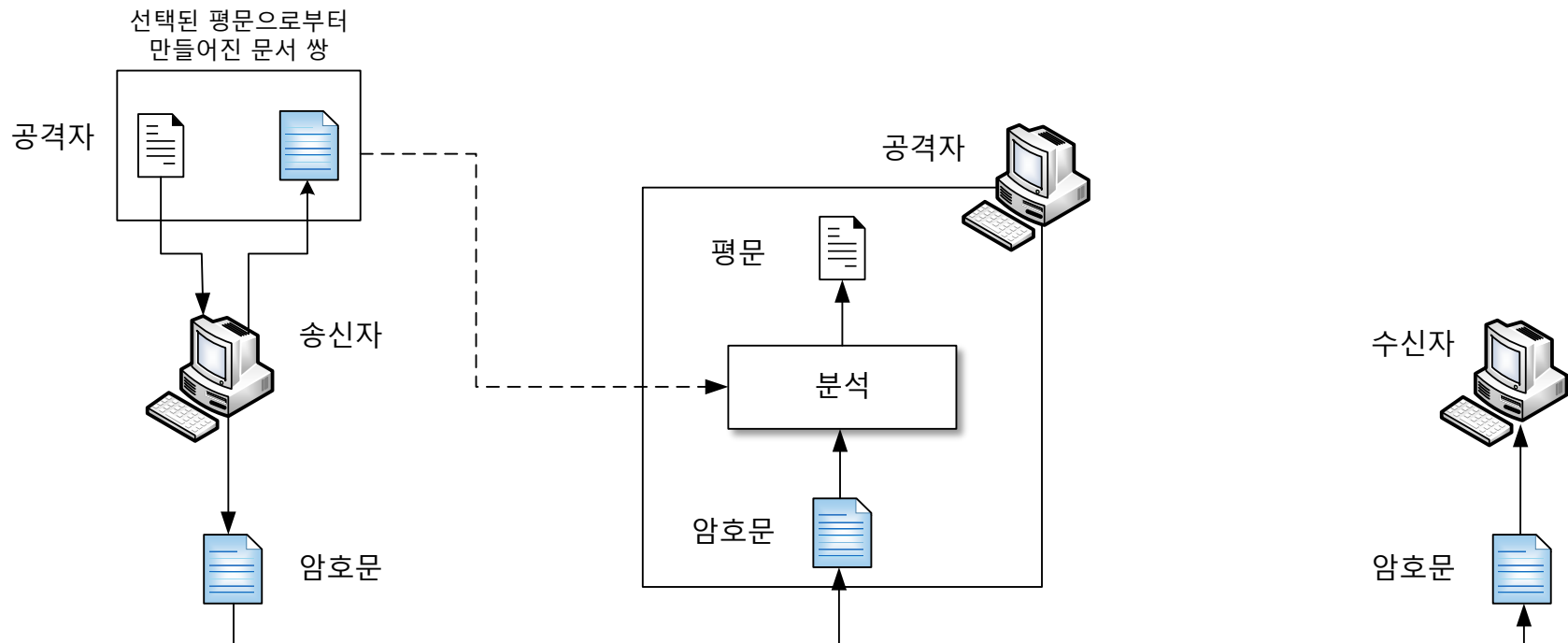
# 고전 대칭-키 암호

- 개요

- 암호 해독

- 선택 평문 공격

- 공격자에게 주어지는 평문/암호문 쌍은 공격자가 선택한 값
    - 송신자의 컴퓨터에 집적 접속
    - 해독하기 쉽지만 적용 가능한 상황은 드뭄



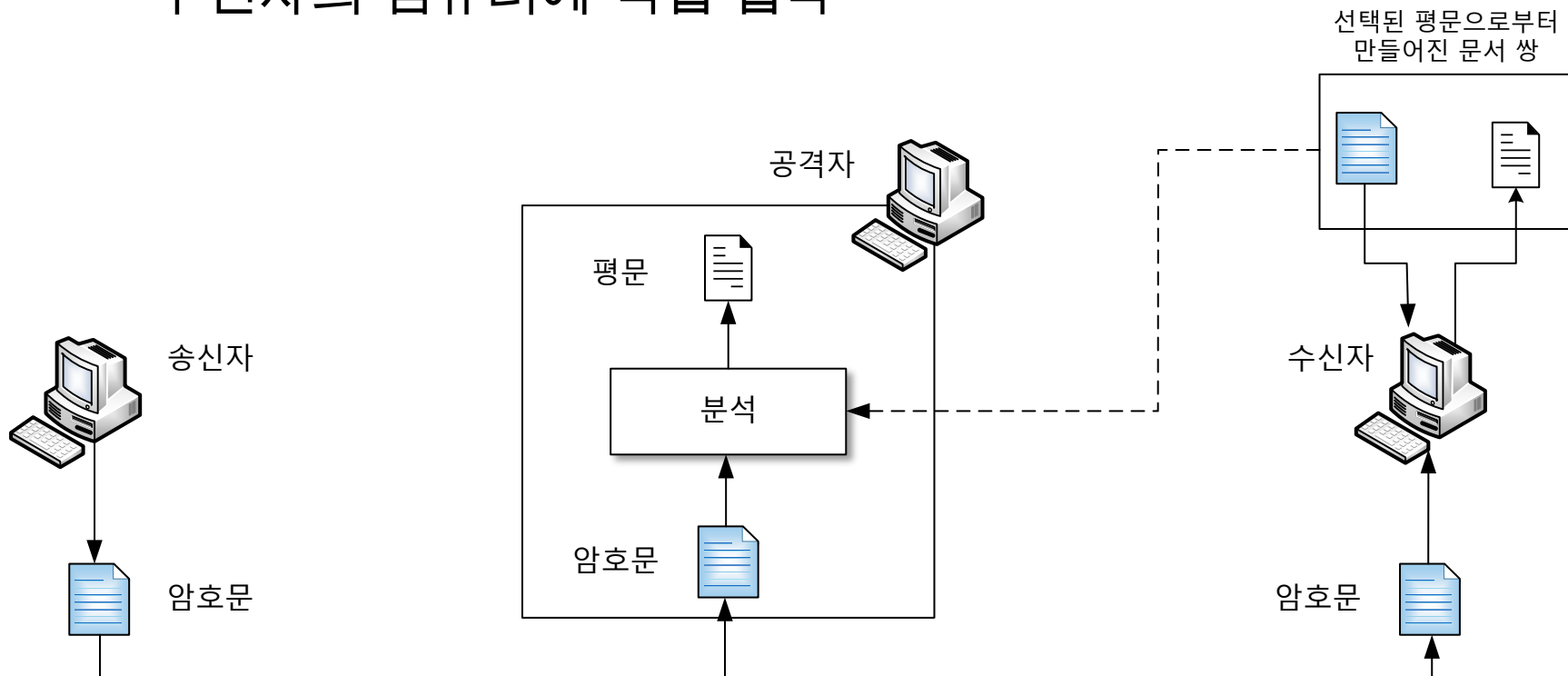
# 고전 대칭-키 암호

- 개요

- 암호 해독

- 선택 암호문 공격

- 공격자가 어떤 암호문을 선택하고 그에 대응되는 평문을 얻은 공격
    - 수신자의 컴퓨터에 직접 접속



# 고전 대칭-키 암호

---

- 대치 암호

- 하나의 기호를 다른 기호로 대체하는 암호

- 단일문자 암호

- 평문에서 하나의 문자 혹은 기호가 위치와 상관없이 암호문에서 항상 같은 문자 혹은 기호로 대체 (일대일 대응)

- e.g., 덧셈 암호, 곱셈 암호, 아핀 암호

- e.g., hello -> KHOOR

# 고전 대칭-키 암호

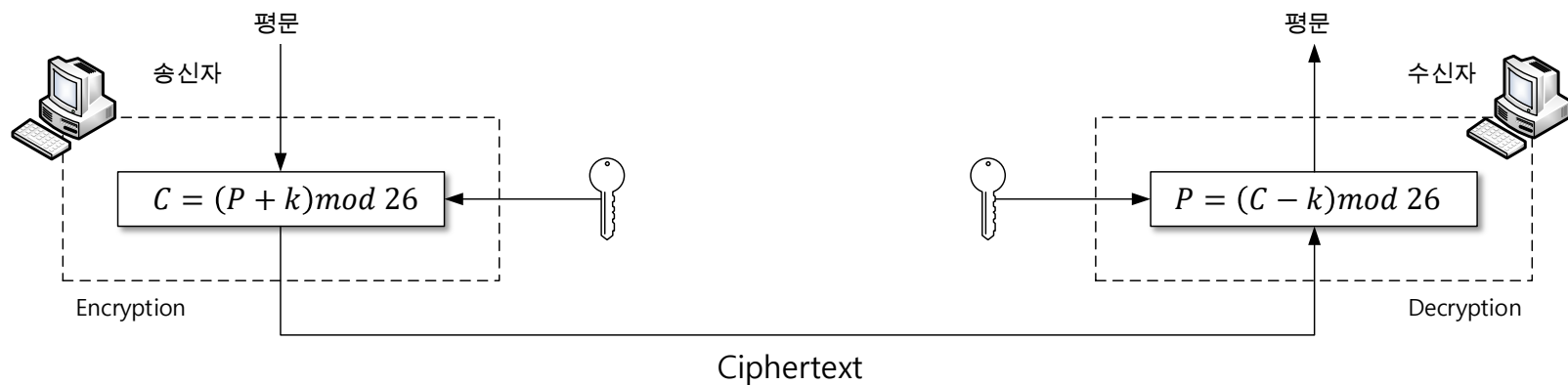
- 대치 암호

- 단일문자 암호

- 덧셈 암호

- 이동 암호 혹은 시저 암호

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	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
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

- 단일문자 암호

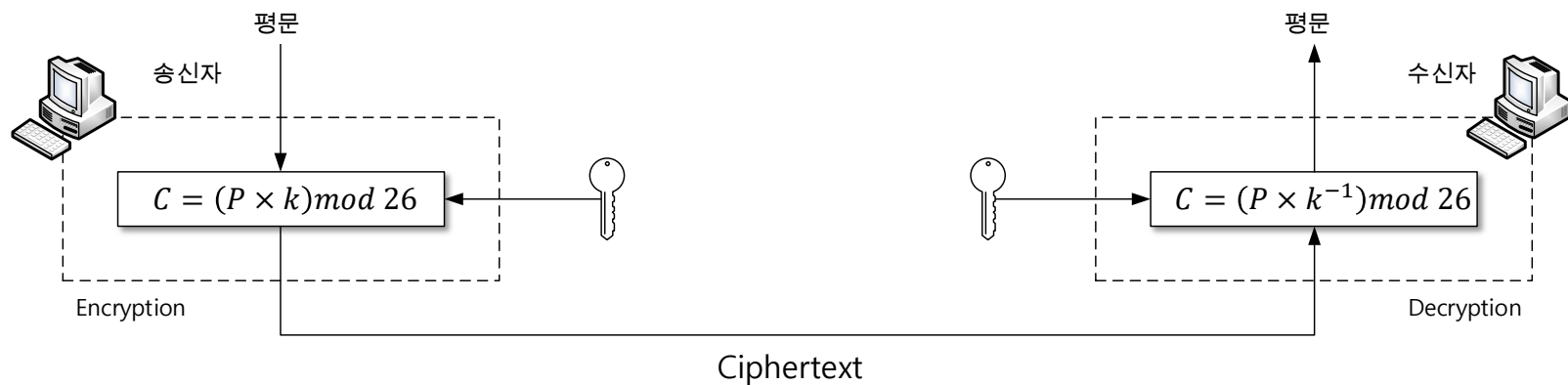
  - 덧셈 암호

    - 암호 해독

      - 전수조사 공격을 이용한 암호문 단독 공격에 취약

- 곱셈 암호

  - 복호화는 곱셈의 역원을 이용





# 고전 대칭-키 암호

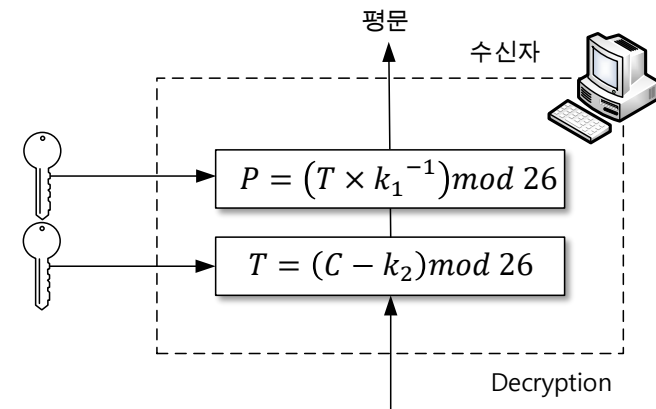
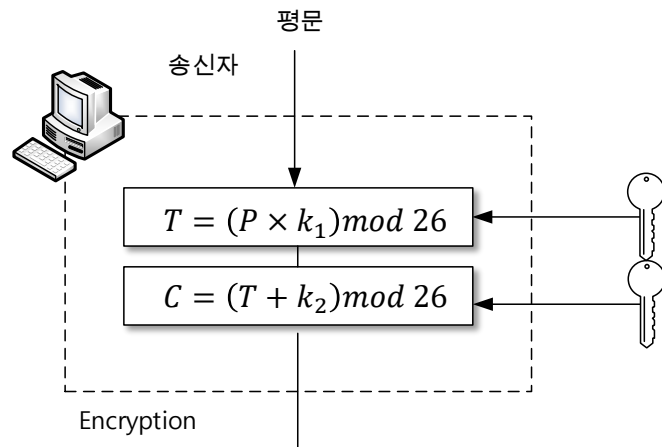
- 대치 암호

- 단일문자 암호

- 아핀 암호

- 덧셈 암호와 곱셈 암호를 결합

- 첫 번째 키는 곱셈 암호에 사용, 두 번째 키는 덧셈 암호에 사용



# 고전 대칭-키 암호

---

- 대치 암호

- 다중문자 암호

- 각 문자가 여러 다른 문자로 대치되는 암호(일대다 대응)  
e.g., 플레이페어 암호, Vigenere 암호, Hill 암호, Rotor 암호
- 언어의 문자 빈도를 감춤
- 플레이페어 암호
  - $5 \times 5$  행렬로 배열된 25개의 알파벳 문자인 비밀 키
  - 행렬에서 문자의 배열을 다르게 함으로써 서로 다른 비밀 키를 많이 생성 가능
  - 암호 해독
    - 빈도 테스트에 기반한 암호문 단독 공격 가능

# 고전 대칭-키 암호

- 대치 암호

- 다중문자 암호

- 플레이페어 암호

- 규칙

- 한 쌍으로 된 두 문자가 비밀 키의 같은 행에 위치하면, 각각의 문자에 대응되는 암호 문자는 같은 행의 오른쪽에 인접하는 문자
      - 한 쌍으로 된 두 문자가 비밀 키의 같은 열에 위치하면, 각 문자의 대응되는 암호 문자는 같은 열에서 그 아래에 위치한 문자
      - 한 쌍으로 된 두 문자가 비밀 키의 같은 행이나 열에 위치하지 않으면, 각 문자에 대응되는 암호 문자는 그 자신의 행에 있지만 다른 문자와 같은 열에 위치한 문자

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

HELLO

He->EC

lx->QZ

lo->BX

# 고전 대칭-키 암호

---

- 대칭 암호

- 다중문자 암호

- Vigenere 암호

- 비즈네르 표를 바탕으로 암호문을 만드는 기법

- 비즈네르표는 26개의 알파벳을 서로 더한 문자를 나타냄

- $P = P_1P_2P_3 \dots$        $C = C_1C_2C_3 \dots$        $K = (K_1, K_2, \dots, K_m),$   
 $(K_1, K_2, \dots, K_m), \dots$

- 암호화:  $C_i = (P_i + K_i)$     복호화:  $P_i = (C_i - K_i)$

# 고전 대칭-키 암호

- 대치 암호

- 다중문자 암호

- Vigenere 암호

- 암호 해독

- 키의 길이를 찾아내는 방법

- Kasiski 테스트

- 최소 3문자로 반복된 원문을 찾아냄

- 찾아낸 원문들 사이의 거리의 최대 공약수를 찾아내고 그 약수를 이용해 추측

e.g.,

A	B	C	D	E	F	G	H	I	A	B	C
+ KEYKEYKEYKEY											
<hr/>											
K	F	A	N	I	D	Q	L	G	K	F	A

"키 k의 길이는 9의 약수일 것 "

# 고전 대칭-키 암호

---

- 대칭 암호

- 다중문자 암호

- Vigenere 암호

- 암호 해독

- 키 자체를 찾아내는 방법

- 빈도 공격을 포함한 덧셈 암호를 해독하기 위해 사용하는 해독 방법 적용  
e.g., 전수조사, 통계적분석 방법

# 고전 대칭-키 암호

- 대칭 암호

- 다중문자 암호

- Hill 암호

- 암호화시 m개의 문자를 한번에 치환하는 암호방식
    - 행렬을 키로 이용하며, 크기는  $m \times m$ 의 정사각형

- $k = \begin{bmatrix} k_{11} & \cdots & k_{1m} \\ \vdots & \ddots & \vdots \\ k_{m1} & \cdots & k_{mm} \end{bmatrix}$

- $\begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix} \begin{bmatrix} 15 \\ 0 \\ 4 \end{bmatrix}$

- 키 행렬은 곱셈에 대한 역원을 가져야 복호화 가능
    - 암호 해독

- m값과 최소 m블록에 대한 평/암호문 쌍을 알고 있다면 알려진 평문 공격 가능

# 고전 대칭-키 암호

- 대칭 암호

- 다중문자 암호

- 에니그마 기계

- Rotor 암호의 원리 기반

- 주요 요소

- 키보드

- 평문을 입력하고 복호화할 때 암호문을 입력하는데 사용

- 램프 보드

- 암호문 문자를 보여주고, 복호화할 때 평문 문자를 보여주는데 사용

- 플러그보드

- 13개의 전선으로 수동 연결된 26개의 플러그를 가진 보드

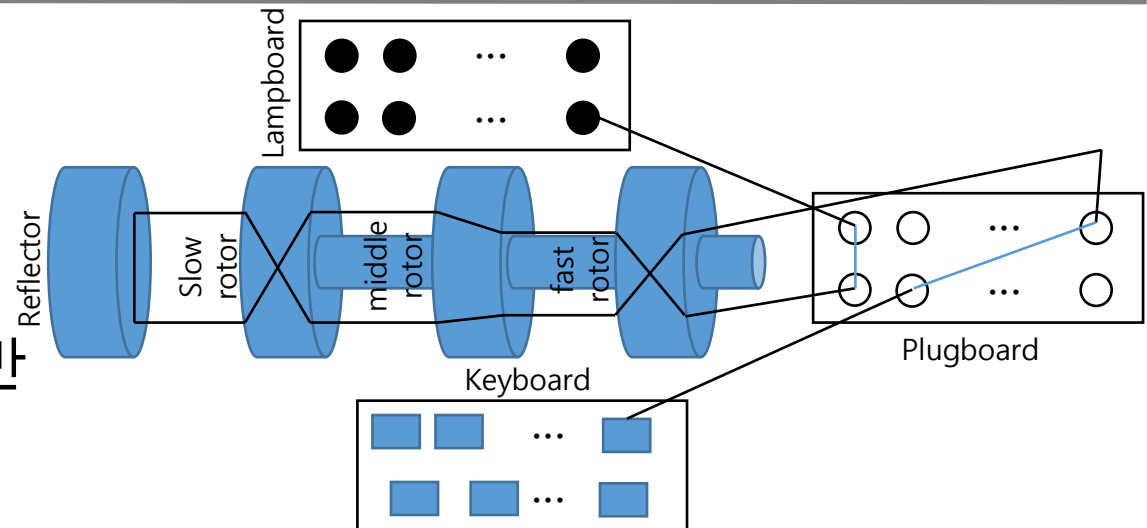
- 구성은 매일 변함

- Rotor

- 세 개의 전선으로 연결되어 26개의 문자를 회전할 수 있는 기계

- 5개의 rotor중에 매일 3개의 rotor 선택

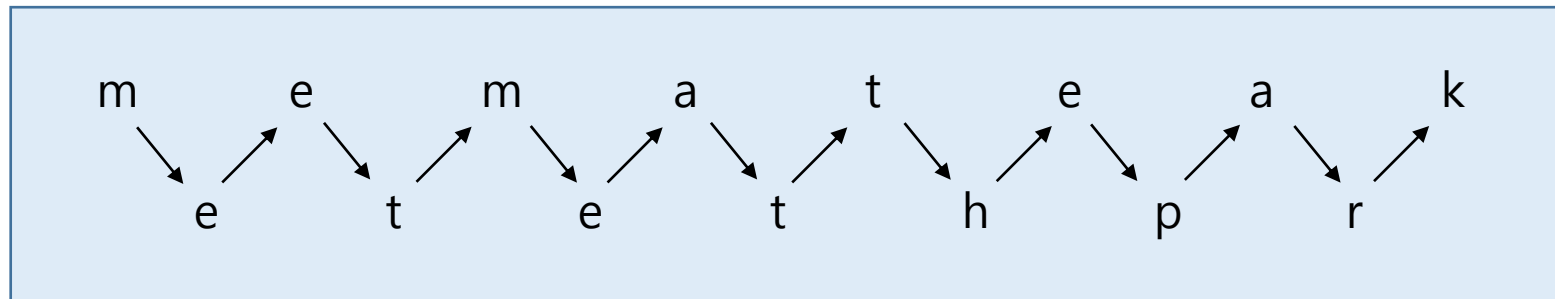
- 반사경





# 고전 대칭-키 암호

- 전치 암호
  - 기호의 위치를 바꾸는 기법
  - 키가 없는 전치 암호
    - Rail fence 암호
      - Meet me at the park->memateaketethpr



# 고전 대칭-키 암호

---

- 전치 암호

- 키가 없는 전치 암호

- 행 순서로 표에 기록된 뒤 열 순서로 전송
  - Meet me at the park->mmtaeehreaekttp



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

# 고전 대칭-키 암호

- 전치 암호

- 키가 있는 전치 암호

- 평문을 블록을 나눈 뒤 각각에 키 적용
  - “enemy attacks tonightz”

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	↑ 복호화
	1	2	3	4	5	

E E M Y N

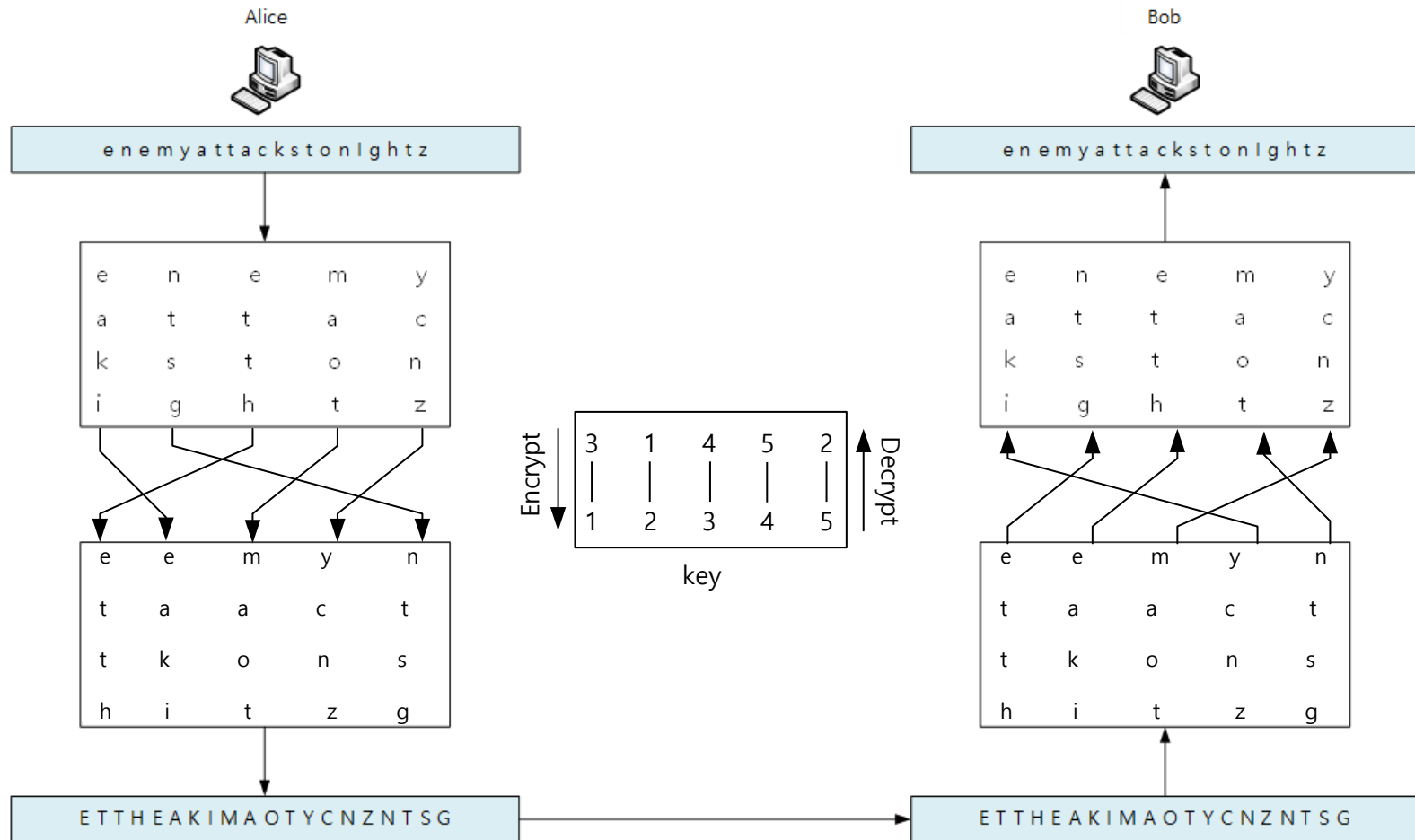
T A A C T

T K O N S

H I T Z G

# 고전 대칭-키 암호

- 전치 암호
- 동작 과정



# 고전 대칭-키 암호

---

- 전치 암호

- 암호 해독

- 통계적인 공격

- 단일 문자 빈도 분석 가능

- 전치 암호는 암호를 재정렬 할 뿐 문자 빈도를 변화 시키지 않음

- 인수조사 공격

- 암호화 할 때 일어날 수 있는 모든 수를 대입하는 공격

- 암호문의 길이의 인수들을 토대로 열의 개수 추측 가능

- e.g.,  $20 = 1 \times 2 \times 2 \times 5$  (1,2,4,5,10,20)

# 고전 대칭-키 암호

- 전치 암호

- 암호 해독

- 패턴 공격

- 키가 있는 전치 암호로 생성된 암호문은 패턴을 가짐

**03 08 13 18 01 06 11 16 04 09 14 19 05 10 15 20 02 07 12 17**

- $(3,8,13,18), (1,6,11,16), (4,9,14,19), (5,10,15,20), (2,7,12,17)$

- 이중 전치 암호

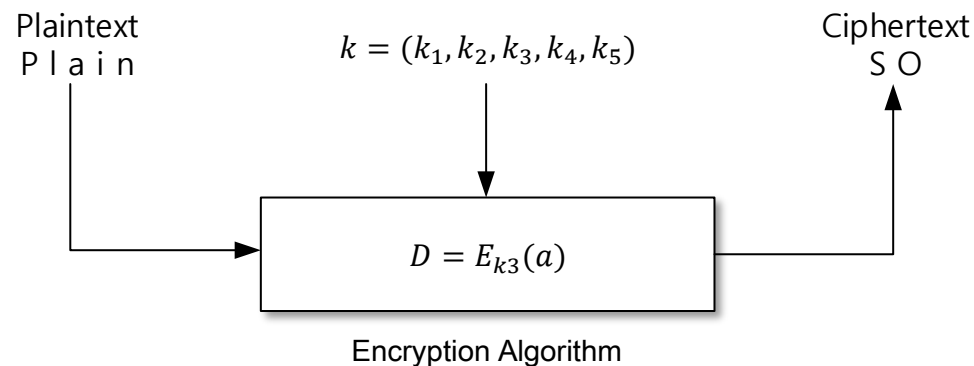
- 암호화, 복호화에 사용되었던 알고리즘을 두 번 반복

# 고전 대칭-키 암호

- 스트림 암호와 블록 암호

- 스트림 암호

- 유사 난수를 연속적으로 생성하여 암호화
- 암호화와 복호화는 한 번에 한 개의 기호에 적용
- 평문 수열  $P$ , 암호문 수열  $C$ , 키 수열  $K$
- 키 수열은 사전에 정의된 수열이거나 알고리즘을 사용하여 한 번에 하나씩 결정되는 값이 될 수 있음

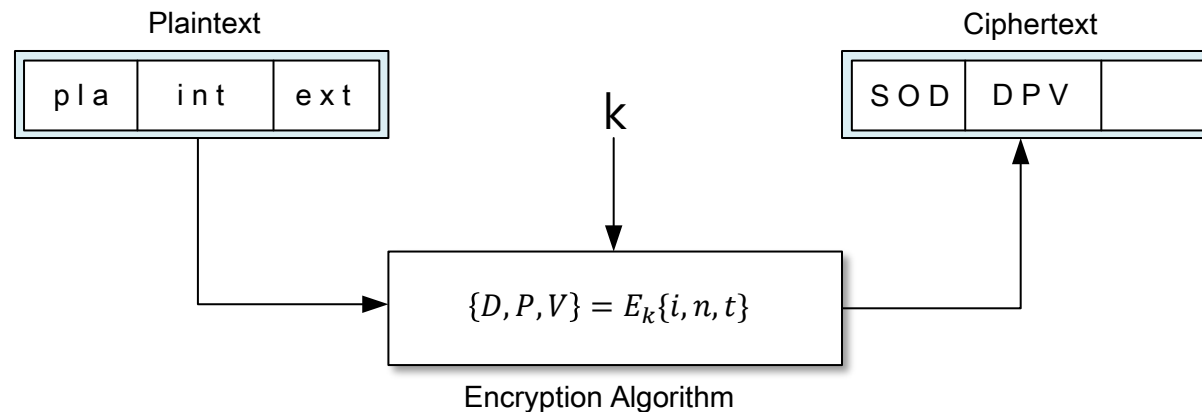


# 고전 대칭-키 암호

- 스트림 암호와 블록 암호

- 블록 암호

- 크기가  $m(m>1)$ 인 평문 기호의 그룹을 만들어 암호화하고, 같은 크기의 암호문 그룹을 생성





---

# Thanks!

이 하 늘 ([haneul@pel.sejong.ac.kr](mailto:haneul@pel.sejong.ac.kr))

# 부록#1

- 대치 암호

- 다중문자 암호
  - Vigenere 암호
  - 비즈네르 표

원문	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
1	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
2	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
3	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
4	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
5	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
6	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
7	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
8	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
9	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
10	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
11	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
12	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
13	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
14	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
15	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
16	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
17	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
18	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
19	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
20	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
21	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
22	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
23	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
24	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
25	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
26	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

출처

[https://ko.wikipedia.org/wiki/%EB%B9%84%EC%A6%88%EB%84%A4%EB%A5%B4\\_%EC%95%94%ED%98%B8](https://ko.wikipedia.org/wiki/%EB%B9%84%EC%A6%88%EB%84%A4%EB%A5%B4_%EC%95%94%ED%98%B8)