2024/07/04, 2024 보안 기초 세미나

암호학과 네트워크 보안

-2장 암호 수학, 3장 고전 대칭-키 암호-

이 은 진(eunjin@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

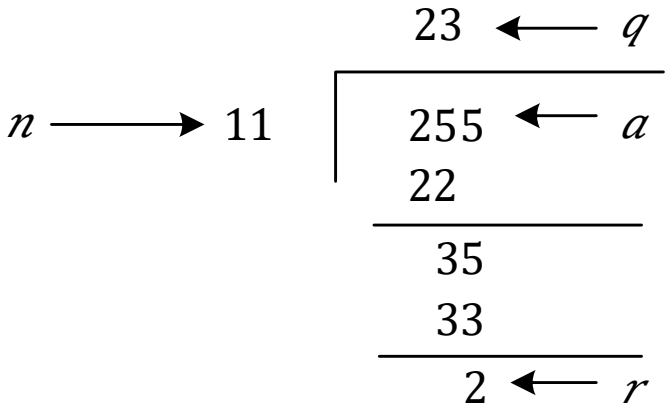
목 차

암호 수학

- 정수 연산
  - 정의
    - 정수 집합에서 정의된 산술 연산
  - 정수 집합
    - 음의 무한대에서부터 양의 무한대까지의 모든 정수를 포함
    - $Z = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
  - 2진 연산
    - 두 개의 입력 값에 대하여 하나의 결과 값을 산출하는 연산
    - 곱셈, 뺄셈, 덧셈
      - 나눗셈은 하나가 아닌 두 개의 결과 값을 산출하므로 이 부류에 속하지 않음

암호 수학

- 정수 연산
  - 정수의 나눗셈
    - 정수 a를 정수 n으로 나눌 때, $a = q \times n + r$으로 표기
      - a = 피제수, q = 몫, n = 제수, r = 나머지
      - e.g., $a = 255, n = 11$인 경우는 $q = 23, r = 2$임
    - a 를 n 으로 나눈 결과가 두 정수 q 와 r 이기 때문에, 연산이 아닌 나눗셈 관계식임(Division Relation)
    - 나눗셈 관계식을 사용할 때, 제한 사항
      - 제수(n)는 양의 정수 ($n > 0$)
      - 나머지(r)는 양의 정수 ($r \geq 0$)



암호 수학

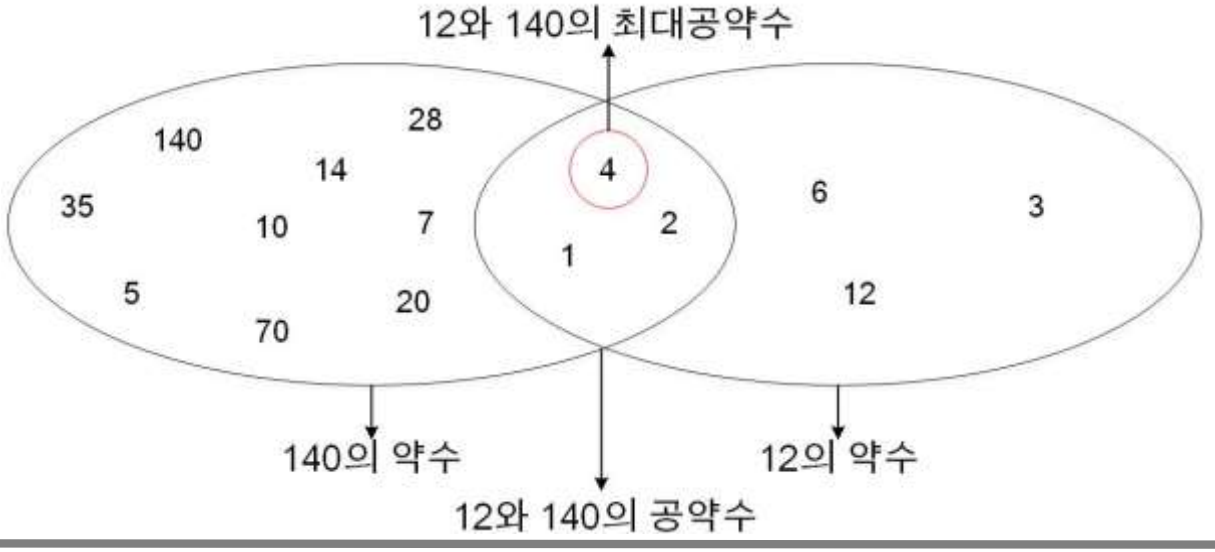
- 정수 연산
  - 가분성(Divisibility)(1/6)
    - 정의
      - 나눌 수 있는 성질
    - $a = q \times n$
      - n이 a를 나누는 경우, n은 a의 약수이고, n | a
    - $a = q \times n + r$
      - n이 a를 나누지 않는 경우, n ∤ a

암호 수학

- 정수 연산
  - 가분성(Divisibility)(2/6)
    - 성질
      - 만약 $a \mid 1$인 경우, $a = \pm 1$
      - 만약 $a \mid b$ 이고 $b \mid a$인 경우, $a = \pm b$
      - 만약 $a \mid b$ 이고 $b \mid c$인 경우, $a \mid c$
        - e.g., $3 \mid 15$이고 $15 \mid 45$이기 때문에 $3 \mid 45$
      - 만약 $a \mid b$ 이고 $a \mid c$인 경우, $a \mid (m \times b + n \times c)$, 여기서 m과 n은 임의의 정수
        - e.g., $3 \mid 15$이고, $3 \mid 9$이기 때문에 $3 \mid (15 \times 2 + 9 \times 4)$이고, 따라서 $3 \mid 66$을 의미

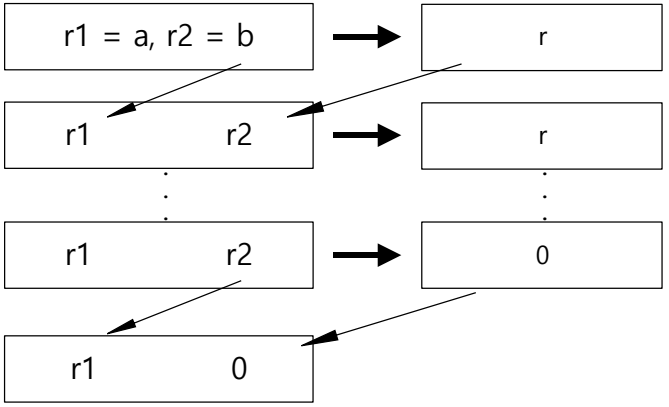
암호 수학

- 정수 연산
  - 가분성(Divisibility)(3/6)
    - 약수(Divisor)
      - 어떤 수를 나누어떨어지게 하는 수
        - 사실 1 : 정수 1은 하나의 약수, 1만을 가짐
        - 사실 2 : 모든 양의 정수는 1과 자신을 포함하는 최소 2개 이상의 약수를 가짐
    - 최대 공약수(GCD, Greatest Common Divisor)
      - 두 정수를 나누는 가장 큰 정수



암호 수학

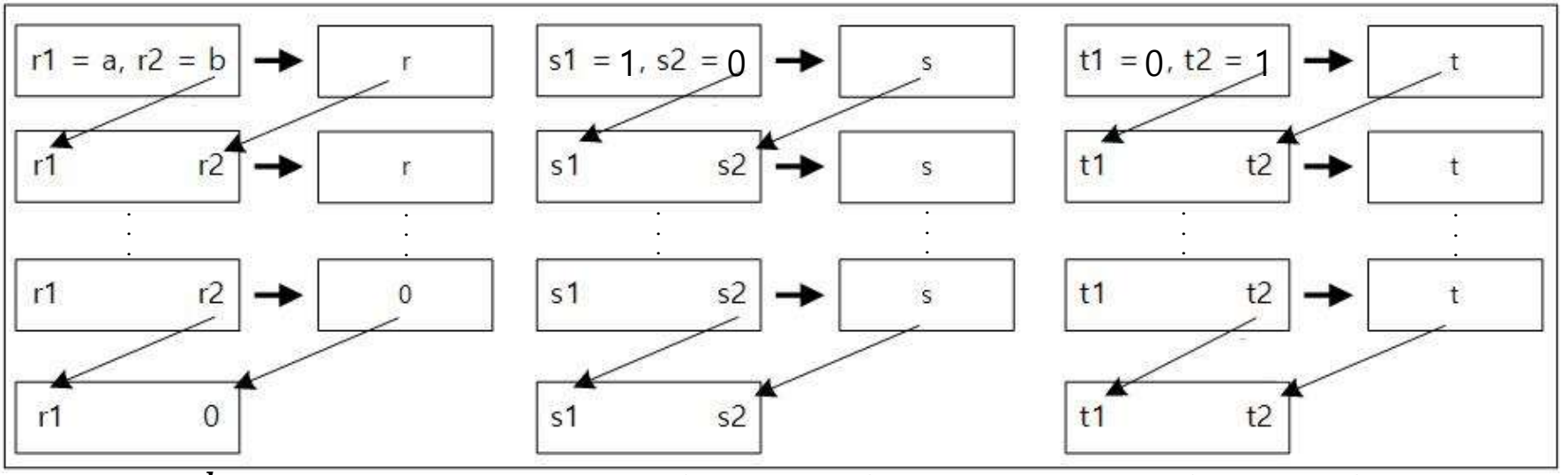
- 정수 연산
  - 가분성(Divisibility)(4/6)
    - 유클리드 알고리즘(Euclidean Algorithm)
      - Eucild가 고안한 두 양의 최대 공약수를 찾아내는 알고리즘
      - 두 가지 사실에 기반함
        - 사실 1 : $\gcd(a, 0) = a$
        - 사실 2 : $\gcd(a, b) = \gcd(b, r)$, 이때 r은 a를 b로 나눈 나머지
          - e.g., $\gcd(25, 60) = \gcd(60, 25) = \gcd(25, 10) = \gcd(10, 5) = \gcd(5,0) = 2$
      - $\gcd(a, b) = 1$이면, a와 b는 서로 소(Relatively Prime)



q	r_1	r_2	r
0	25	60	25
2	60	25	10
2	25	10	5
2	10	5	0
	5	0	

암호 수학

- 정수 연산
  - 가분성(Divisibility)(5/6)
    - 확장 유클리드 알고리즘(Extended Euclidean Algorithm)
      - $s \times a + t \times b = \gcd(a, b)$
      - $\gcd(a, b)$를 계산하고 동시에 s와 t값 계산 가능
      - 유클리드 알고리즘과 달리 각 단계에서 세 쌍의 계산과 교환 사용
      - 변수 s1과 s2는 각각 1과 0으로, 변수 t1과 t2는 각각 0과 1로 초기화



암호 수학

- 정수 연산
  - 가분성(Divisibility)(6/6)
    - 확장 유클리드 알고리즘(Extended Euclidean Algorithm)
      - e.g., $a = 161$이고 $b = 28$일 때 $\gcd(a, b)$의 $s,\ t$를 계산

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
	7	0		-1	4		6	-23	

      - $gcd(161, 28) = 7,\ s = -1,\ t = 6$을 얻을 수 있음
      - $(-1) \times 161 + 6 \times 28 = 7$로 검증 가능

암호 수학

- 정수 연산
  - 선형 디오판투스 방정식(Linear Diophantine Equations)(1/3)
    - 정의
      - 어떤 방정식의 정수 해를 구하는 데 있어, 미지수가 2개 이상이 되는 경우를 디오판투스 방정식이라고 함
    - 확장 유클리드 알고리즘의 응용
    - 2변수 선형 디오판투스 방정식은 $ax + by = c$
    - $d \mid c$ 인 경우, 방정식의 무수히 많은 해가 존재함
      - 여기서 d는 a와 b의 최대공약수임
    - $d \nmid c$인 경우, 방정식의 해가 존재하지 않음

암호 수학

- 정수 연산
  - 선형 디오판투스 방정식(Linear Diophantine Equations)(2/3)
    - 특수 해($d \mid c$ 인 경우)
      1. d로 방정식의 양변을 나누어, $a_1x + b_1y = c_1$ 식으로 만듦
      2. 확장 유클리드 알고리즘을 이용하여 식 $a_1s + b_1t = 1$을 만족하는 s와 t 값을 계산함
      3. 특수 해는 $x_0 = \left(\frac{c}{d}\right)s$, $y_0 = \left(\frac{c}{d}\right)t$로 계산
    - 일반 해
      - 특수 해를 찾아낸 후 계산
      - $x = x_0 + k\left(\frac{b}{d}\right), y = y_0 - k\left(\frac{a}{d}\right)$ (k는 임의의 정수)

암호 수학

- 정수 연산
  - 선형 디오판투스 방정식(Linear Diophantine Equations)(3/3)
    - e.g., 방정식 $21x + 14y = 35$의 특수 해와 일반 해 구하기
      - $d = \gcd(21,14) = 7$
      - $7|35$이므로 방정식은 무한한 해를 가짐
      - d로 양변 나누기
        - $d = \gcd(21,14) = 7, 3x + 2y = 5$
      - 확장 유클리드 알고리즘을 활용한 s, t 계산
        - $3s + 2t = 1$을 만족하는 $s = 1, t = -1$
      - 특수 해 x_0, y_0 계산
        - $x_0 = (\frac{35}{7}) \times 1 = 5,\ y_0 = (\frac{35}{7}) \times (-1) = -5$
      - 일반 해 x,y 계산
        - $x = 5 + k \times (\frac{14}{7}),\ y = -5 - k \times (\frac{14}{7})$ (k는 정수)

암호 수학

- 모듈로 연산(Modular Arithmetic)
  - 나눗셈 관계식($a = q \times n + r$)에서 r 값이 무엇인지 알기 위한 연산
    - 모듈로 연산자(Modular Operator)
      - $a \bmod n = r$
        - n은 모듈로(Modulus), r은 나머지(Residue)
        - E.g., $27 \bmod 5$에서 $r = 0$이고, $27 \bmod 5 \equiv 2$

암호 수학

- 모듈로 연산(Modular Arithmetic)
  - 잉여류 Z_n(Residue Classes)
    - 정수 n에 대한 $mod\ n$을 한 나머지 집합
      - 모든 정수는 모듈로 연산 후 나머지로 $0,1 \ldots, (n-1)$ 사이의 정수 값을 가짐
      - $Z_n = \{0, 1, 2, \ldots, (n-1)\}$
      - e.g., $n = 5$이면

$$[0] = \{\cdots, -15, -10, -5, 0, 5, 10, 15, \cdots\}$$
$$[1] = \{\cdots, -14, -9, -4, 1, 6, 11, 16, \cdots\}$$
$$[2] = \{\cdots, -13, -8, -3, 2, 7, 12, 17, \cdots\}$$
$$[3] = \{\cdots, -12, -7, -2, 3, 8, 13, 18, \cdots\}$$
$$[4] = \{\cdots, -11, -6, -1, 4, 9, 14, 19, \cdots\}$$

암호 수학

- 모듈로 연산(Modular Arithmetic)
  - 합동(Congruence)
    - 어떤 정수로 나눈 나머지가 서로 같은 두 정수 사이의 관계
    - 합동 연산자(≡) 사용
      - e.g., $2 \equiv 12(mod\ 10),\ 13 \equiv 23(mod\ 10),\ 34 \equiv 24(mod\ 10)$

1. 합동 연산자와 등식 연산자와의 차이점
   - 등식 연산자는 Z의 원소를 Z의 원소로 대응하지만, 합동 연산자는 Z의 원소에서 Z_n의 원소로 대응

2. $(mod\ n)$은 Z_n을 나타냄
   - $12mod10$에서 mod는 연산자지만, $2 \equiv 12(mod\ 10)$에서 $mod\ 10$은 공역 Z_{10}을 의미함

암호 수학

- 모듈로 연산(Modular Arithmetic)
  - Z_n에서의 연산
    - 성질
      - $(a + b)\ mod\ n = [(a\ mod\ n) + (b\ mod\ n)]mod\ n$
        - e.g., $(1{,}723{,}345) + (2{,}124{,}945)\ mod\ 11\ =\ (8 + 9)\ mod\ 11\ =\ 6$
      - $(a - b)\ mod\ n = [(a\ mod\ n) - (b\ mod\ n)]mod\ n$
        - e.g., $(1{,}723{,}345) - (2{,}124{,}945)\ mod\ 11\ =\ (8 - 9)\ mod\ 11\ = 10$
      - $(a \times b)\ mod\ n = [(a\ mod\ n) \times (b\ mod\ n)]mod\ n$
        - e.g., $(1{,}723{,}345) \times (2{,}124{,}945)\ mod\ 11\ =\ (8 \times 9)\ mod\ 11\ =\ 6$

암호 수학

- 모듈로 연산(Modular Arithmetic)
  - 역원(Inverse)(1/2)
    - 정의
      - 어떤 수 a와 연산했을 때 항등원이 나오는 값
        - 항등원이란 a, b가 연산자 O에 대해 닫혀 있을 때, 모든 수 a에 대하여 $a\ O\ b\ =\ b\ O\ a\ =\ a$를 만족하는 b
        - 합에 대한 항등원 : 0
        - 곱에 대한 항등원 : 1
    - 덧셈에 대한 역원
      - $a + b \equiv 0\ mod\ n$
      - Z_n에서 a의 덧셈에 대한 역원은 $b = n - a$와 같이 계산
      - 모듈로 연산에서 각각의 정수는 덧셈에 대한 역원을 가짐
        - e.g., $0\ +\ 0 \equiv\ 0\ (mod\ 5),\ 1\ +\ 4 \equiv 0\ (mod\ 5),\ 2\ +\ 3 \equiv 0\ (mod\ 5)$

암호 수학

- 모듈로 연산(Modular Arithmetic)
  - 역원(Inverse)(2/2)
    - 곱셈에 대한 역원
      - $a \times b \equiv 1 \, mod \, n$
      - 모듈로 연산에서 정수는 곱셈에 대한 역원이 있을 수도 있고 없을 수도 있음
        - 역원이 있는 경우, 그 정수에 해당하는 곱셈에 대한 역원의 곱은 모듈로 n에서 1과 합동임
      - a가 Z_n에서 곱셈에 대한 역원을 갖기 위한 필요충분조건이 $\gcd(n, a) = 1$임
        - 이 경우 a와 n은 서로소임
      - e.g., Z_{11}에서 곱셈에 대한 역원 $(1,1), (2,6), (3,4), (5,9), (7,8), (10,10)$이 존재함
        - Z_{11}에서 0을 제외한 모든 a에 대해 $\gcd(11, a) = 1$(서로 소)임
        - Z_n에 속한 정수 a가 곱셈에 대한 역원을 가질 필요충분조건은 $\gcd(n, a) \equiv 1 (mod \, n)$임

암호 수학

- 모듈로 연산(Modular Arithmetic)
  - 덧셈과 곱셈에 대한 다른 집합
    - $Z_n{}^*$은 Z_n의 부분 집합으로 Z_n에 속한 정수 중 곱셈에 대한 역원을 가진 원소의 집합
    - Z_n과 $Z_n{}^*$의 차이점
      - Z_n
        - Z_n의 각 원소는 덧셈에 대한 역원을 갖지만, 단지 몇몇의 원소만이 곱셈에 대한 역원을 가짐
        - 덧셈에 대한 역원이 필요할 때 사용
      - $Z_n{}^*$
        - $Z_n{}^*$의 각 원소는 곱셈에 대한 역원을 갖지만, 단지 몇몇의 원소만이 덧셈에 대한 역원을 가짐
        - 곱셈에 대한 역원이 필요할 때 사용

암호 수학

- 모듈로 연산(Modular Arithmetic)
  - 다른 두 집합
    - Z_p와 Z_p^*
      - 두 집합에서 모듈로는 소수
    - Z_p
      - 0부터 $p-1$까지의 모든 정수 포함
      - 덧셈에 대한 역원을 가지고, 0을 제외한 각각의 원소는 곱셈에 대한 역원을 가짐
    - Z_p^*
      - 1부터 $p-1$까지의 모든 정수 포함
      - 덧셈과 곱셈에 대한 역원을 가짐

암호 수학

- 행렬(Matrices)
  - 정의
    - 행렬은 $l \times m$개의 원소를 갖는 직사각형 배열
      - l은 행의 개수
      - m은 열의 개수
    - a_{ij}는 i번째 행과 j번째 열에 위치

$$Matrix\ A: \begin{bmatrix} a_{11} & \cdots & a_{1j} \\ \vdots & & \vdots \\ a_{i1} & \cdots & a_{ij} \end{bmatrix}$$

암호 수학

- 행렬(Matrices)
  - 행 행렬
    - 행렬이 하나의 행($l = 1$)만 갖는 경우
  - 열 행렬
    - 행렬이 하나의 열($m = 1$)만 갖는 경우
  - 정방 행렬
    - 행의 개수와 열의 개수가 같은($l = m$) 경우
      - 주대각선(정방 행렬에서 a_{ij}중 $i = j$인 원소들을 잇는 선)을 형성
  - 항등 행령
    - 주대각선이 1이고 나머지는 0인 정방 행렬일 경우

암호 수학

- 행렬(Matrices)
  - 연산과 관계식(1/2)
    - 등식
      - 두 행렬의 행과 열의 개수와 대응되는 원소가 동일한 경우, 그 두 행렬은 동일함
      - $a_{ij} = b_{ij}$이면, $A = B$
    - 덧셈과 뺄셈
      - 두 행렬의 행과 열의 개수가 같은 경우, 덧셈과 뺄셈 가능
      - e.g., $\begin{bmatrix} 12 & 4 & 4 \\ 11 & 12 & 30 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 10 \end{bmatrix} + \begin{bmatrix} 7 & 2 & 3 \\ 8 & 10 & 20 \end{bmatrix}$,
        $\begin{bmatrix} -2 & 0 & -2 \\ -5 & -8 & -10 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 10 \end{bmatrix} - \begin{bmatrix} 7 & 2 & 3 \\ 8 & 10 & 20 \end{bmatrix}$

암호 수학

- 행렬(Matrices)
  - 연산과 관계식(2/2)
    - 곱셈
      - A가 $l \times m$ 행렬이고, B가 $m \times p$ 행렬인 경우, 두 행렬의 곱은 $l \times p$ 크기를 갖는 행렬 C가 됨
      - $c_{ik} = \sum a_{ij} \times b_{jk} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + \cdots + a_{im} \times b_{mj}$
        - e.g., $\begin{bmatrix} 1 & 3 \\ 4 & 6 \end{bmatrix} \times \begin{bmatrix} 0 & 6 & 1 \\ -3 & 5 & -2 \end{bmatrix}$
          $= \begin{bmatrix} 1 \times 0 + 3 \times (-3) & 1 \times 6 + 3 \times 5 & 1 \times 1 + 3 \times (-2) \\ 4 \times 0 + 6 \times (-3) & 4 \times 6 + 6 \times 5 & 4 \times 1 + 6 \times (-2) \end{bmatrix} = \begin{bmatrix} -9 & 21 & -5 \\ -18 & 54 & -8 \end{bmatrix}$
    - 스칼라 곱
      - 행렬 A에 실수 k를 곱하는 연산
        - e.g., $\begin{bmatrix} 15 & 6 & 3 \\ 9 & 6 & 12 \end{bmatrix} = 3 \times \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 4 \end{bmatrix}$

암호 수학

- 행렬(Matrices)
  - 행렬식
    - 크기가 $m \times m$인 정방 행렬 A의 행렬식은 $\det(A)$로 표기
    - $m = 1$ 인 경우, $\det(A) = a_{11}$
    - $m > 1$ 인 경우, $\det(A) = \sum_{i=1\cdots m}(-1)^{i+j} \times a_{ij} \times \det(A_{ij})$
      - 여기서 A_{ij}는 A에서 i열과 j행을 제거한 뒤 얻어진 행렬임
      - e.g., $\det\begin{bmatrix}5 & 2 & 1\\ 3 & 0 & -4\\ 2 & 1 & 6\end{bmatrix} = (-1)^{1+1}\times 5 \times \det\begin{bmatrix}0 & -4\\ 1 & 6\end{bmatrix} +$
        $(-1)^{1+2}\times 2 \times \det\begin{bmatrix}3 & -4\\ 2 & 6\end{bmatrix} + (-1)^{1+3}\times 1 \times \det\begin{bmatrix}3 & 0\\ 2 & 1\end{bmatrix}$
        $= (1 \times 5 \times 4) + (-1) \times 2 \times 26 + 1 \times 1 \times 3 = -29$

암호 수학

- 행렬(Matrices)
  - 역행렬(1/2)
    - 덧셈에 대한 역행렬
      - A에 대하여 $A + B = 0$을 만족하는 행렬 B
      - $A + B = 0$
      - 모든 i와 j에 대해 $b_{ij} = -a_{ij}$

암호 수학

- 행렬(Matrices)
  - 역행렬(2/2)
    - 곱셈에 대한 역행렬
      - 정방행렬 A에 대하여 $AB = BA = I$(항등행렬)을 만족하는 행렬 B
      - $A \times B = I$
      - 여인수행렬과 행렬식을 구한 후 여인수행렬에 대해서 수반행렬로 바꿈 그 후 행렬식의 역수를 곱함
        - 여인수행렬: 정사각형 $A = [a_{ij}]$에서 원소 a_{ij}에 관한 계수와 그 계수를 원소로 갖는 행렬
        - 수반행렬: 여인수행렬의 전치행렬
        - 전치행렬: 주대각선을 기준으로 행과 열의 위치를 바꾼 행렬

암호 수학

- 행렬(Matrices)
  - 잉여 행렬
    - 곱셈에 대한 역행렬이 존재하는 행렬
      - $gcd(\det(A), n) = 1$인 경우, 곱셈에 대한 역행렬을 가짐
        - 여기서 n은 Z_n의 n
  - 합동
    - 두 행렬이 열과 행의 개수가 같고 모든 대응되는 원소가 모듈로 n에 대하여 합동이면, 그 두 행렬은 모듈로 n에 대하여 합동임
    - $A \equiv B (mod\ n)$

암호 수학

- 선형 합동
  - 일변수 선형 방정식(1/2)
    - $ax \equiv b\ (mod\ n)$ 형태의 방정식의 해를 구함
    - $gcd(a,n) = d$일 때 $d \mid b$인 경우, 다음과 같이 해를 구함
      1. 모듈로를 포함하여 방정식의 양변을 d로 나눔
      2. 특수 해 x_0를 구하기 위해 약분한 방정식의 양변에 a의 곱셈에 대한 역원을 곱함
      3. 일반 해는 k = 0, 1, $\cdots$, $(d-1)$에 대하여, $x = x_0 + k(\frac{n}{d})$임

암호 수학

- 선형 합동
  - 일변수 선형 방정식(2/2)
    - e.g., 방정식 $14x \equiv 12(mod 18)$의 해 구하기
      - $d = \gcd(14,18) = 2$이고 $2|12$이므로 두 개의 근이 존재함
      - 방정식 약분하기
        - $7x \equiv 6(mod 9) \rightarrow x \equiv 6(7^{-1})(mod 9)$
      - 특수 해 구하기
        - $x_0 = (6 \times 7^{-1}) mod 9 = (6 \times 4)(mod 9) = 6$
        - $x_1 = x_0 + 1 \times \left(\frac{18}{2}\right) = 15$
      - 해 6과 15는 $(14 \times 6) mod 18 = 12, (14 \times 15) mod 18 = 12$이기에 합동 관계식을 만족함

암호 수학

- 선형 합동
  - 일차 연립 방정식(1/2)
    - 동일한 모듈로에 대한 일차 방정식들의 집합에서 변수의 계수로 구성된 행렬의 역행렬이 존재하면 주어진 방정식들의 공통 해를 구할 수 있음

암호 수학

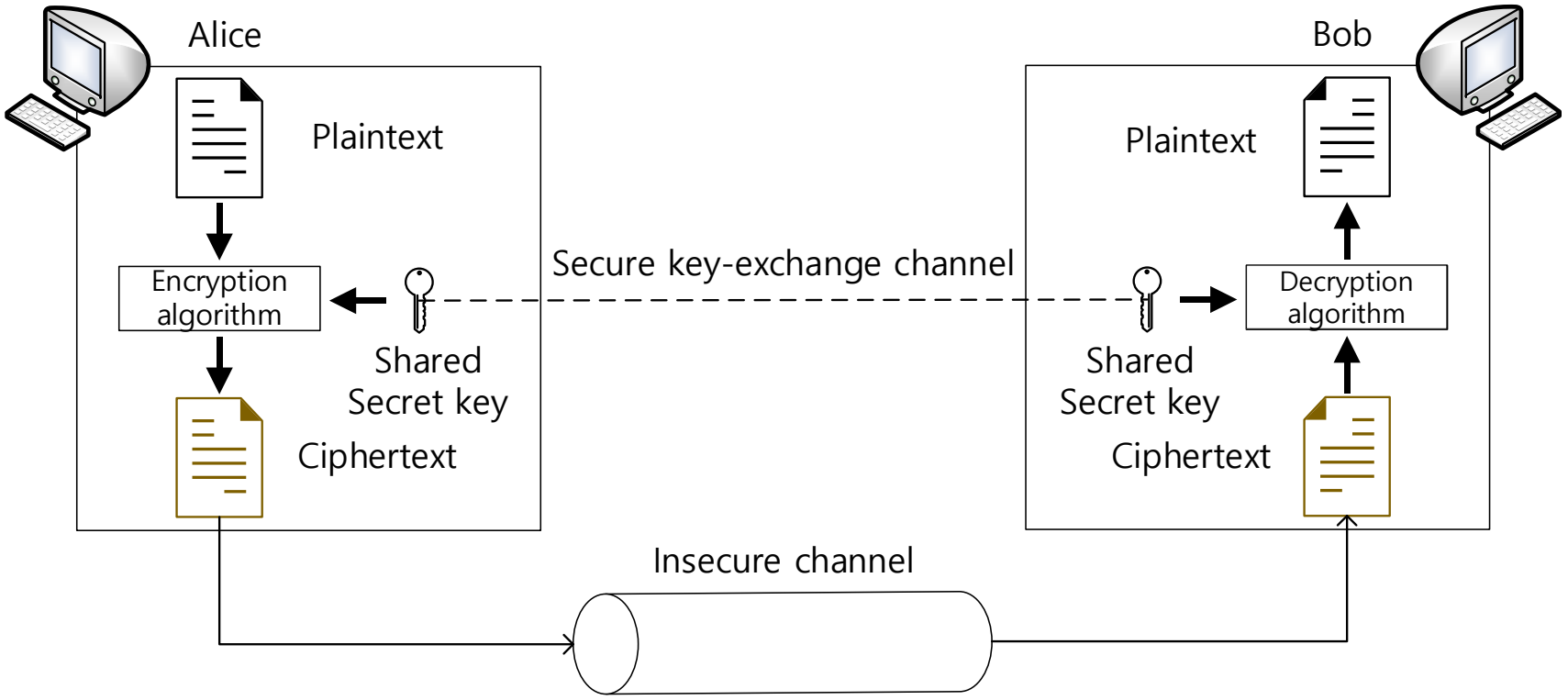
- 선형 합동
  - 일차 연립 방정식(2/2)
    - e.g.,$3x + 5y + 7z \equiv 3(mod\ 16), x + 4y + 13z \equiv 5(mod\ 16), 2x + 7y + 3z \equiv 4(mod\ 16)$의 해 구하기
      - 연립 방정식을 행렬 A로 바꿈
        - 행렬 $A\begin{bmatrix}3 & 5 & 7\\ 1 & 4 & 13\\ 2 & 7 & 3\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix}3\\ 5\\ 4\end{bmatrix}$
      - 양변에 각각 행렬 A의 곱셈에 대한 역행렬 A^{-1}을 곱함
        - $A^{-1} = \frac{1}{129}\begin{bmatrix}79 & -39 & -37\\ 23 & 5 & 32\\ 1 & 11 & -7\end{bmatrix}, \begin{bmatrix}x\\ y\\ z\end{bmatrix} = \frac{1}{129}\begin{bmatrix}79 & -39 & -37\\ 23 & 5 & 32\\ 1 & 11 & -7\end{bmatrix}\begin{bmatrix}3\\ 5\\ 4\end{bmatrix}$
      - 모듈러 연산을 함
        - $\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix}15(mod\ 16)\\ 4(mod\ 16)\\ 14(mod\ 16)\end{bmatrix}$

목 차

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

고전 대칭-키 암호

- 개요
  - 대칭-키 암호 개념(1/2)
    - 암/복호화 과정 모두 동일한 한 개의 키를 사용하는 암호



고전 대칭-키 암호

- 개요
  - 대칭-키 암호 개념(2/2)
    - 암호화 : $C = E_k(P)$
    - 복호화 : $\mathrm{P} = D_k(C)$
    - $D_k(E_k(P)) = E_k\big(D_k(P)\big) = \mathrm{P}$
  - 키의 개수
    - m명의 사용자가 서로 통신을 해야 한다면 $(m \times (m-1))/2$개의 키가 필요함
      - 각각의 사용자는 나머지 $m-1$명의 사용자와 통신하기 위해 $m-1$개의 키를 필요로 하고 사용자 A와 사용자 B는 서로 통신하기 위해 1개의 키만을 사용

고전 대칭-키 암호

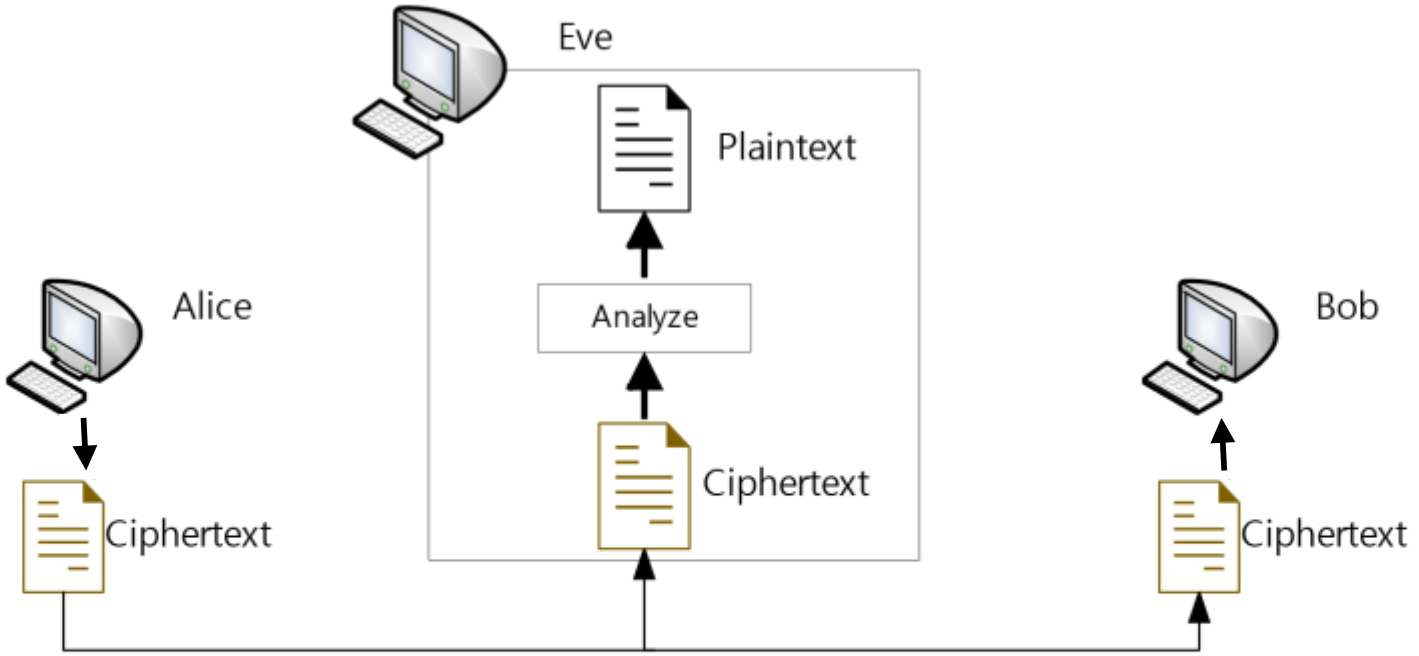
- 개요
  - Kerckhoff의 원리
    - 암호의 안전성은 키의 비밀성에만 기반을 두어야 함
      - 키를 추측하는 것이 매우 어려워서 암호/복호 알고리즘을 비공개로 할 필요가 없어야 함
      - 시스템의 비밀이 적고 단순할 수록, 시스템의 안전성을 유지하기 쉬움

고전 대칭-키 암호

- 개요
  - 암호 해독 기술(1/5)
    - 전수조사 공격(Brute-force Attack)
      - 공격자가 모든 가능한 키를 사용
      - 공격자는 의미 있는 평문을 얻을 때까지 반복
      - 이 공격을 막기 위해서 가능한 키의 수가 매우 커야 함
    - 통계적인 공격(Statistical Attack)
      - 평문 언어의 고유한 특징으로부터 정보를 얻음
      - 이 공격을 막기 위해서 암호문이 평문 언어의 특징이 드러나지 않아야 함
    - 패턴 공격(Pattern Attack)
      - 평문 언어의 특징은 드러나지 않지만, 암호문에 어떤 패턴이 존재할 경우 사용
      - 이 공격을 막기 위해서 암호문을 랜덤하게 보이도록 만드는 암호를 사용함

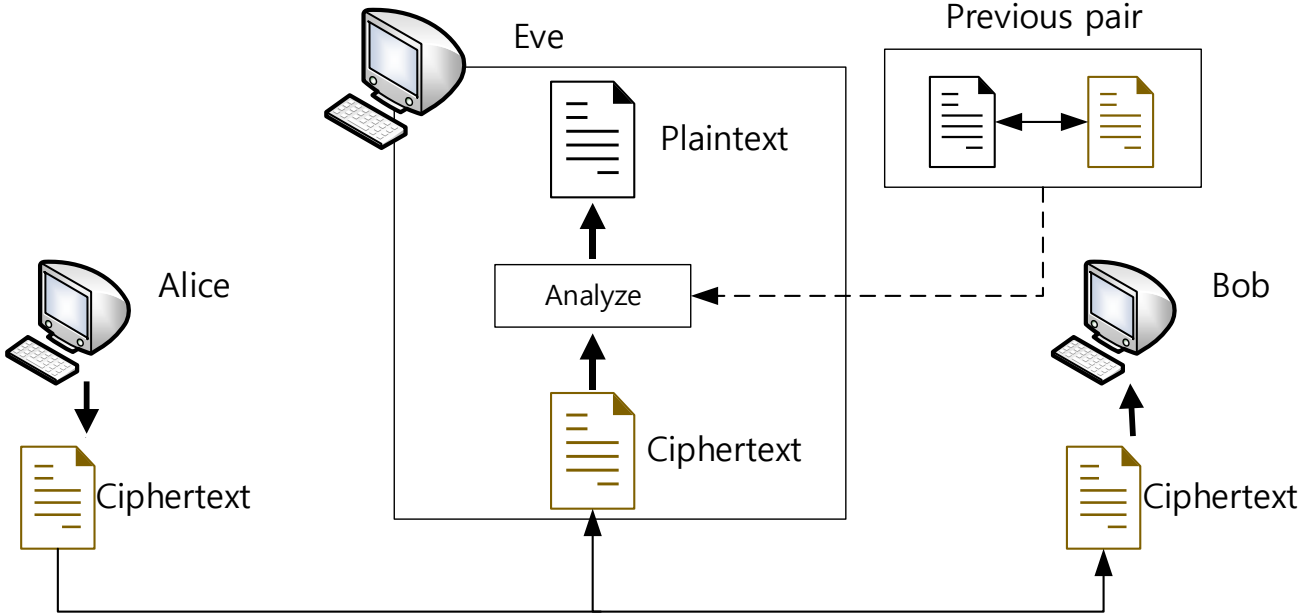
고전 대칭-키 암호

- 개요
  - 암호 해독 기술(2/5)
    - 암호문 단독 공격(Ciphertext-Only Attack)
      - 공격자가 단지 암호문만을 얻어서 대응되는 평문과 키를 찾음



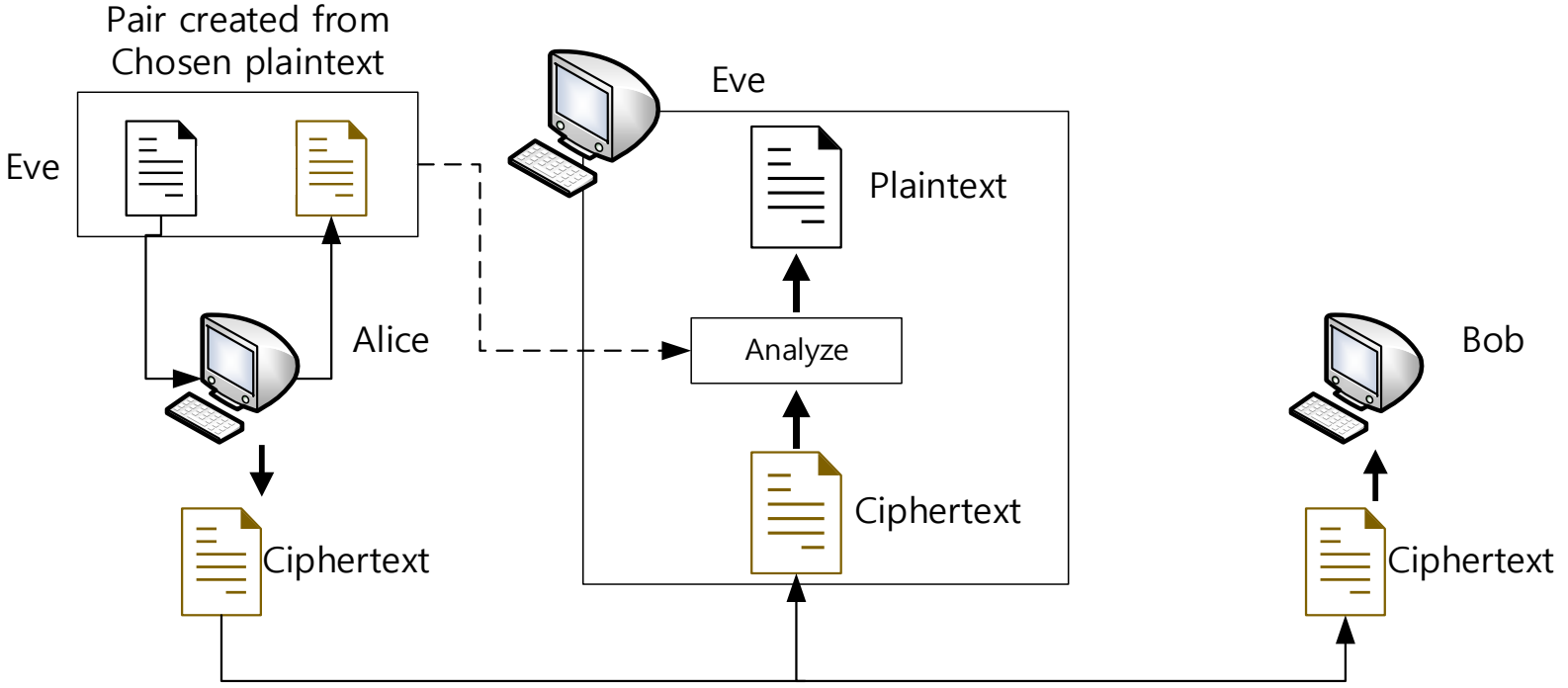
고전 대칭-키 암호

- 개요
  - 암호 해독 기술(3/5)
    - 알려진 평문 공격(Known-Plaintext Attack)
      - 공격자에게 미리 주어진 평문/암호문 쌍의 연관성을 이용하여 암호문을 해독함
      - 키를 변경하거나 과거에 보낸 메시지를 노출하지 않을 수도 있기 때문에 적용 가능한 상황이 암호문 단독 공격보다는 드묾



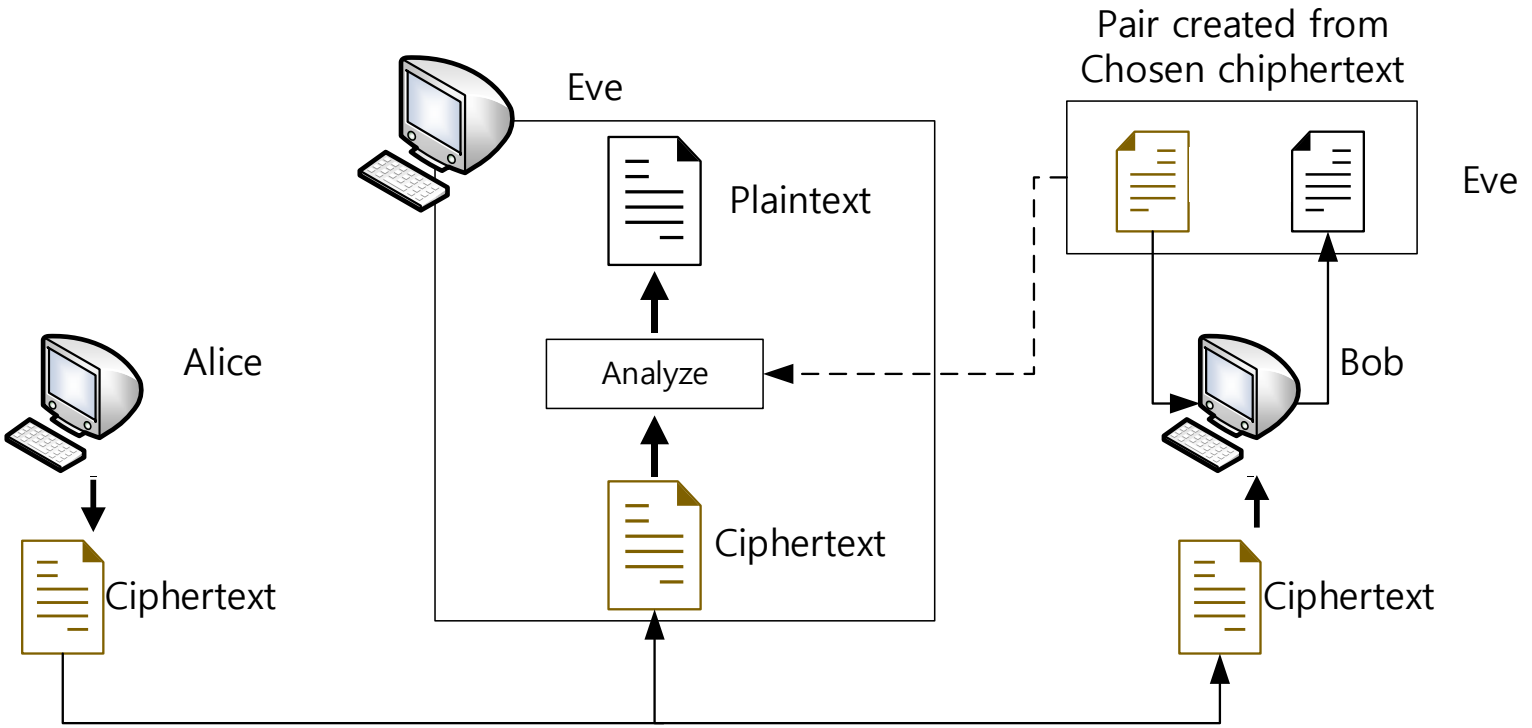
고전 대칭-키 암호

- 개요
  - 암호 해독 기술(4/5)
    - 선택 평문 공격(Chosen-Plaintext Attack)
      - 공격자가 어떤 평문을 선택하고 그에 대응되는 암호문을 얻음
      - 공격자가 송신자의 컴퓨터에 접속할 수 있어야 함
      - 해독하기 쉽지만 적용 가능한 상황은 드묾



고전 대칭-키 암호

- 개요
  - 암호 해독 기술(5/5)
    - 선택 암호문 공격(Chosen-Ciphertext Attack)
      - 공격자가 어떤 암호문을 선택하고 그에 대응되는 평문을 얻음
      - 공격자가 수신자의 컴퓨터에 접속할 수 있어야 함

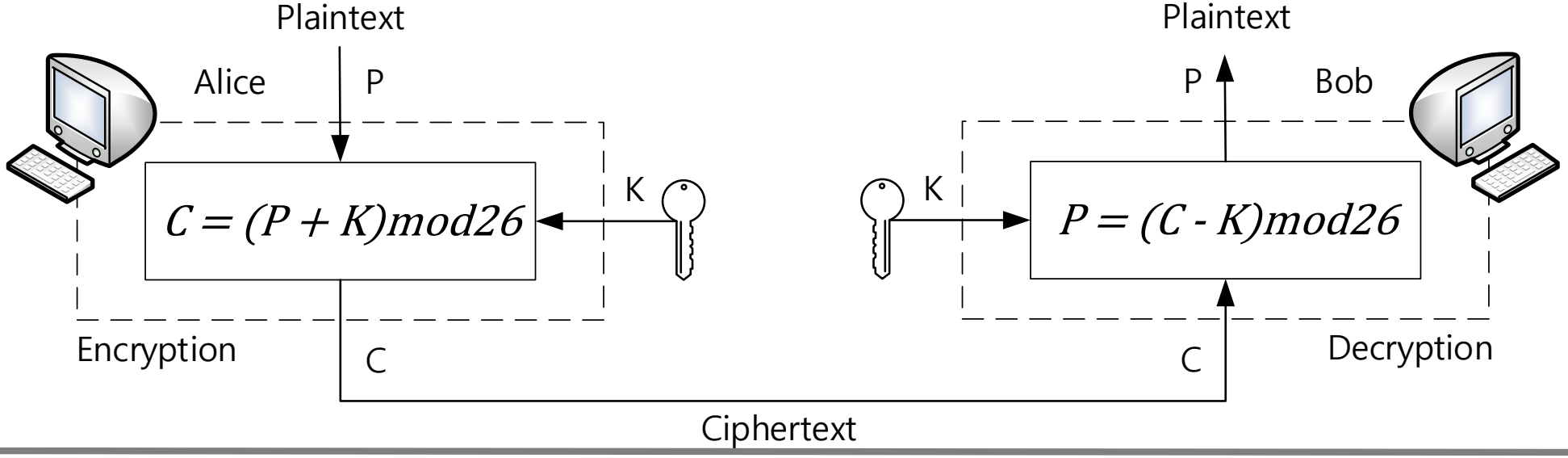


고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 하나의 문자를 다른 문자로 대체하는 암호
  - 단일문자 암호(Monoalphabetic Ciphers)(1/5)
    - 평문에서 하나의 문자가 위치와 상관없이 암호문에서 항상 같은 문자로 대체되는 암호
    - 일대일 대응 관계
    - e.g. 평문: hello -> 암호문: KHOOR

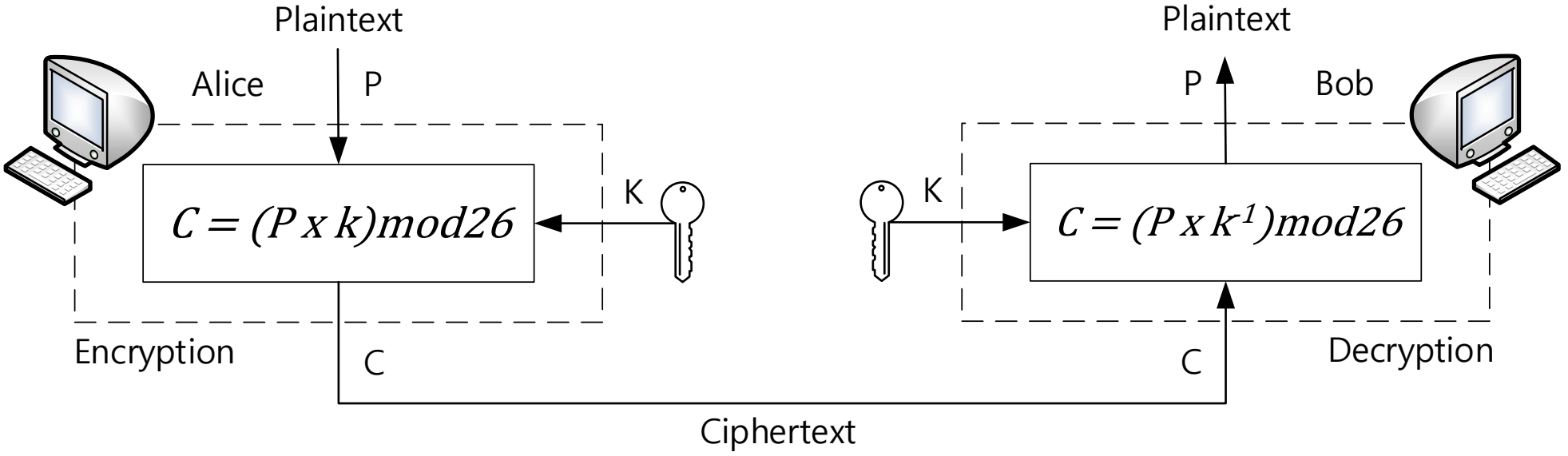
고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 단일문자 암호(2/5)
    - 덧셈 암호(Additive Cipher)
      - 평문에 키를 더하여 암호화하고, 암호문에서 키를 빼서 복호화하는 암호
      - 이동 암호(Shift Cipher) 혹은 시저 암호(Caesar Cipher)
      - 전수조사 공격을 이용한 암호문 단독 공격에 취약함
        - 키 공간이 매우 작으며 단지 26개의 키만 존재하기 때문



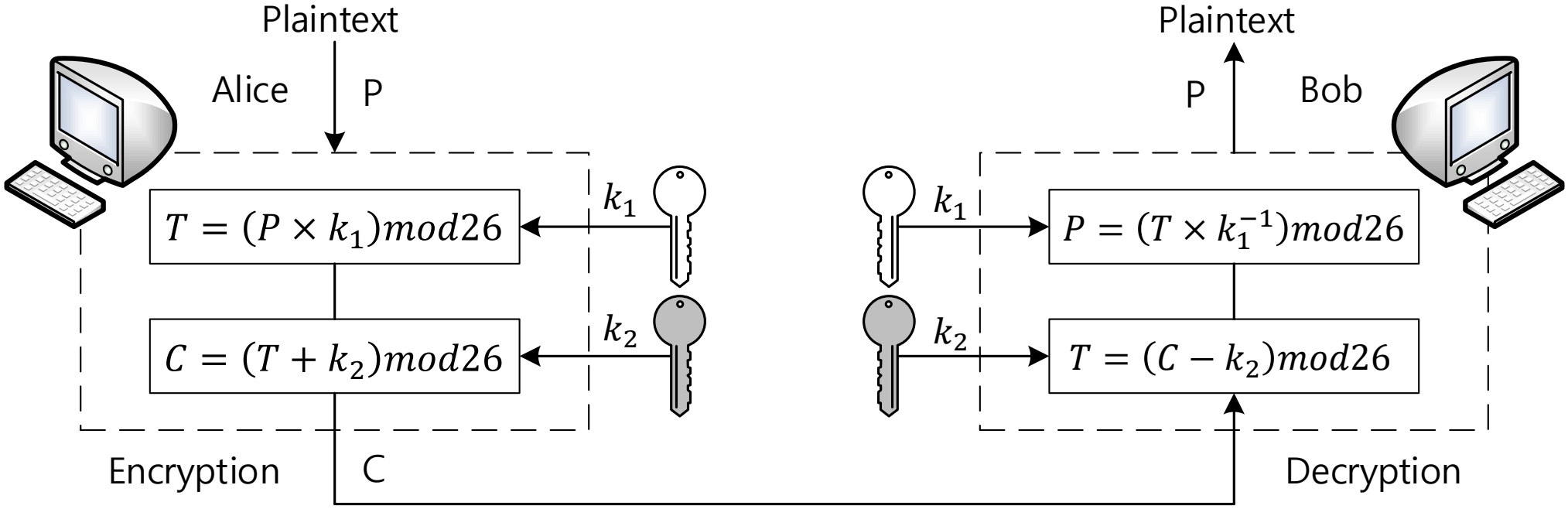
고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 단일문자 암호(3/5)
    - 곱셈 암호(Multiplicative Ciphers)
      - 평문에 키를 곱하여 암호화하고 암호문을 키로 나누어 복호화하는 암호
      - 암호화와 복호화가 서로 역함수 관계에 있음을 보장하기 위해, 키는 집합 Z_{26}^{*}의 원소여야 함



고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 단일문자 암호(4/5)
    - 아핀 암호(Affine Cipher)
      - 두 개의 키를 사용하고, 덧셈 암호와 곱셈 암호를 결합한 암호
      - 첫 번째 키는 곱셈 암호에 사용, 두 번째 키는 덧셈 암호에 사용



고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 단일문자 암호(5/5)
    - 덧셈 암호, 곱셈 암호, 아핀 암호는 작은 키 공간을 가져 전수조사 공격에 취약
    - 평문 문자와 대응되는 암호문 문자 사이의 대응을 구성하면 해결 가능

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	N	O	A	T	R	B	E	C	F	U	X	D	Q	G	Y	L	K	H	V	I	J	M	P	Z	S	W

    - 암호 해독
      - 키 공간의 크기는 26!이라 전수조사 공격이 힘듦
      - 대치 암호는 암호문에서 각 문자의 출현 빈도수를 변경시키지 않기 때문에 통계적인 공격에 취약함

고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(1/13)
    - 각 문자가 다른 대치를 가짐
    - 평문 문자와 암호문 문자와의 관계가 일대다 대응
    - 언어의 문자 빈도를 감추는 장점이 있음
    - 자동키 암호
      - 첫 번째 부분키는 두 사람이 비밀리에 합의한 사전에 정의된 값
      - 두 번째 부분키는 첫 번째 평문의 값
      - 세 번째 부분키는 두 번째 평문의 값
      - 자동키는 암호화 과정 중 부분키가 평문으로부터 자동으로 생성됨

$P = P_1P_2P_3 \ldots$ C= $C_1C_2C_3 \ldots$ k= $(k_1, P_1, P_2, \ldots)$

암호화: $C_i = (P_i + k_i) mod 26$ 복호화: $P_i = (C_i - k_i) mod 26$

고전 대칭-키 암호

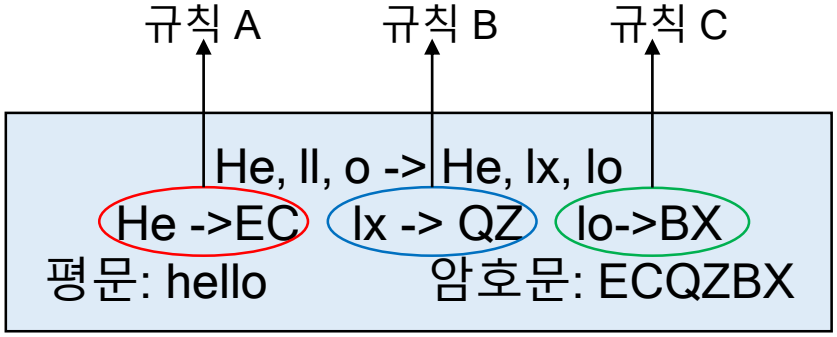
- 대치 암호(Substitution Cipher)
  - 다중문자 암호(2/13)
    - 플레이 페어 암호(Playfair Cipher)
      - 5x5 행렬로 배열된 25개의 알파벳 문자인 비밀 키(I와 J는 암호화될 때 동일한 것으로 간주)
      - 행렬에서 문자의 배열을 다르게 함으로서 서로 다른 많은 비밀 키 생성 가능
      - 플레이 페어 암호 해독
        - 두 문자열 빈도 테스트에 기반한 암호문 단독 공격 사용

고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(3/13)
    - 플레이 페어 암호
      - 규칙
        A. 한 쌍으로 된 문자가 비밀 키의 같은 행에 위치하면, 각각의 문자에 대응되는 암호 문자는 같은 행의 오른쪽에 인접하는 문자임
        B. 한 쌍으로 된 두 문자가 비밀 키의 같은 열에 위치하면, 각 문자의 대응되는 암호 문자는 같은 열에서 그 아래에 위치한 문자임
        C. 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 문자임

Secret Key =

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



고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(4/13)
    - Vigenere 암호(1)
      - 두 사람이 합의한 초기 비밀 키로 암호화
      - 암호화: $C_i = (P_i + k_i)\ mod 26$ 복호화: $P_i = (C_i - k_i)\ mod 26$
    - Vigenere 암호(2)
      - Vigenere 표를 이용하여 암호화

/	a	b	c		z
A	A	B	C		Z
B	B	C	D		A
C	C	D	E	…	B
				⋮	
Z	Z	A	B		Y

'PASCAL' 키로 'she is listening'을 암호화

첫 번째 행에서 s를 첫 번째 열에서 P를 찾으면 그 교차점은 암호 문자 H가 됨
이를 반복하면 'HHWKSWXSLGNPCG'가 완성됨

고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(5/13)
    - Vigenere 암호의 암호 해독(1/2)
      1. 키 길이를 찾아내는 방법(Kasiski 테스트)
         - 암호문에서 최소 3문자로 구성된 반복된 원문 조각을 찾아냄
         - 찾아낸 원문 조각들의 최대 공약수(m)를 찾음
           - 키의 길이가 m의 약수라는 것을 의미
         - 암호문을 m 조각으로 나눠서 각 조각에 따로따로 통계적인 공격을 적용함
         - e.g., 평문 'P = ABCDEFGHIABC' 를 키 'K = KEY' 를 이용하여 암호화하는 경우

           A B C D E F G H I A B C
           + K E Y K E Y K E Y K E Y

           K F A N I D Q L G K F A

           9의 간격을 두고 반복됨
           ➔ 최대 공약수(m) = 9
           ➔ 키 길이는 3, 9 중 하나

고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(6/13)
    - Vigenere 암호의 암호 해독(2/2)

      2. 키 자체를 찾아냄
         - 암호문을 m개의 다른 부분으로 나눈 뒤, 빈도 공격을 포함한 덧셈 암호를 해독하기 위해 사용하는 방법에 적용함
         - 각 암호문 조각은 해독되어 전체 평문을 생성하기 위해 조합됨

고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(7/13)
    - Hill 암호
      - 평문을 같은 크기의 블록으로 나누고 블록이 한 번에 암호화됨
      - Hill 암호화 키 $\mathrm{K} = \begin{bmatrix} k_{11} & \cdots & k_{1m} \\ \vdots & & \vdots \\ k_{m1} & \cdots & k_{mm} \end{bmatrix}$

$$C_1 = P_1k_{11} + P_2k_{21} + \cdots + P_mk_{m1}$$
$$C_2 = P_1k_{12} + P_2k_{22} + \cdots + P_mk_{m2}$$
$$\vdots$$
$$C_m = P_1k_{m1} + P_2k_{2m} + \cdots + P_mk_{mm}$$

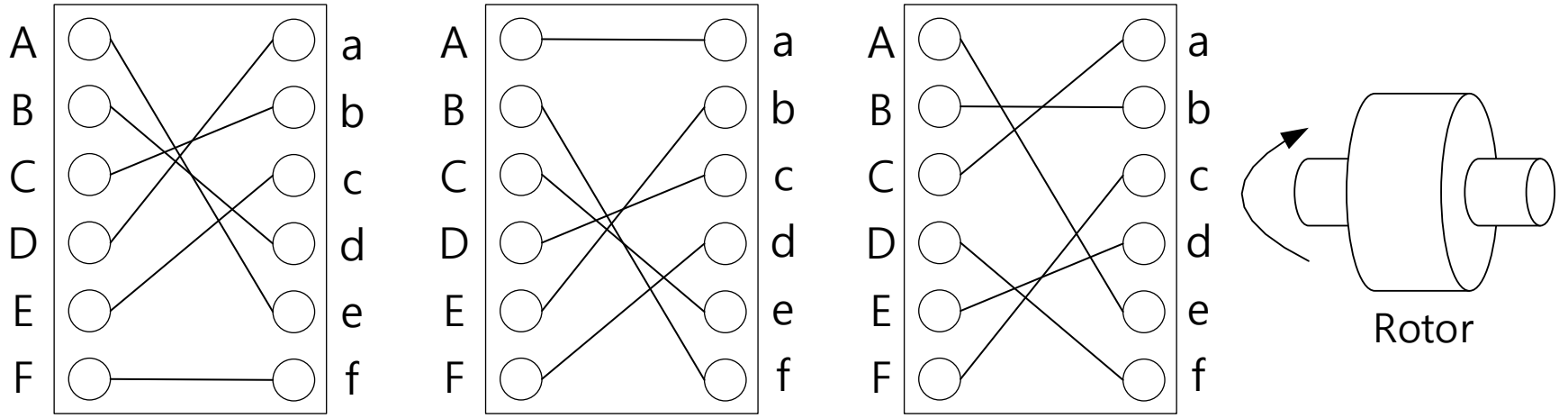
      - C_1과 같은 각 암호문 문자가 블록에 속한 모든 평문 문자에 의존함
        - Hill 암호에서 키 행렬은 곱셈에 대한 역원을 가져야 함

고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(8/13)
    - Hill 암호의 암호 해독
      - 전수조사 공격은 키가 $m \times m$ 행렬이므로 해독이 어려움
      - 빈도 분석은 Hill 암호가 평문의 통계를 보존하지 않으므로 해독이 어려움
      - 공격자가 m값과 최소 m 블록에 대한 평문/암호문 쌍을 알고 있는 경우, 알려진 평문 공격을 적용할 수 있음
        - 공격자는 대응되는 행이 대응되는 알려진 평문/암호문 쌍을 나타내는 두 개의 $m \times m$행렬 P(평문)와 C(암호문)를 생성 가능
        - $C = PK$이므로 공격자는 P의 역행렬이 존재하면 키를 찾기 위해 K=$P^{-1}C$ 사용 가능
        - 역행렬이 존재하지 않을 경우, 다른 m개의 평문/암호문 쌍을 이용해야 함
    - 일회용 패드(One-Time Pad)
      - 키는 평문과 같은 길이를 가지며 완전히 랜덤하게 선택되는 암호
      - 완벽한 암호지만 상업적으로 구현되기는 어려움

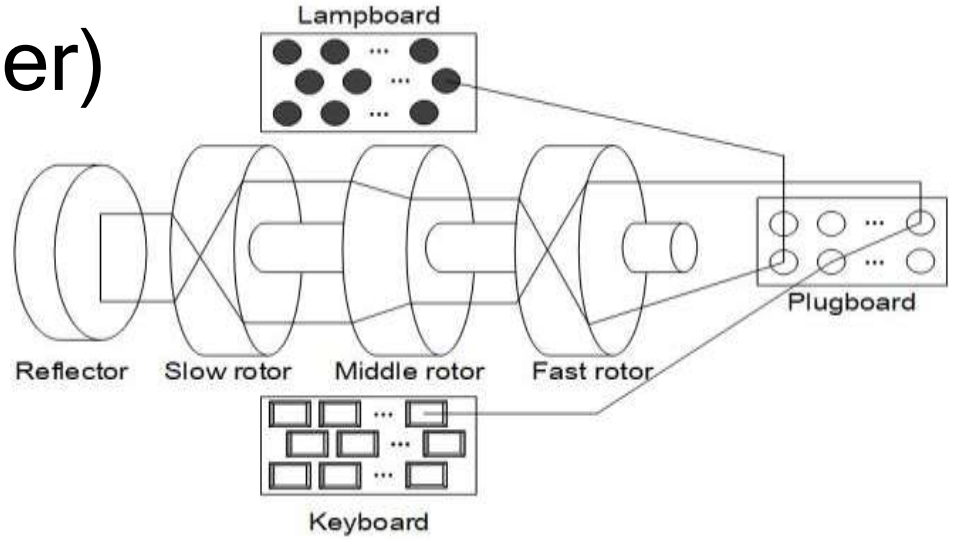
고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(9/13)
    - Rotor 암호
      - 단일 알파벳 치환에 따른 아이디어를 이용하고, 각각의 평문 문자에 대해 평문과 암호문 문자의 대응을 변화시킴
      - Rotor의 초기 설정은 두 사람 사이의 비밀키임
      - 첫 번째 평문 문자가 초기 설정을 사용하여 암호화되고, 두 번째 문자가 첫 번째 회전 이후 암호화되며, 나머지도 이와 같이 암호화됨



고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(10/13)
    - 에니그마(Enigma)
      - Rotor 암호의 원리 기반 기계
      - 주요 요소
        - 키보드
          - 암호화할 때 평문을 입력하고, 복호화할 때 암호문을 입력하는데 사용
        - 램프보드
          - 암호화할 때 암호문 문자를 보여주고, 복호화할 때 평문 문자를 보여주는 데 사용
        - 플러그보드
          - 13개의 전선으로 수동으로 연결된 26개의 플러그를 가짐(다른 전송을 제공하기 위해 구성이 매일 변함)
        - Rotor
          - 세 개의 전선으로 구성
          - 5개의 사용 가능한 rotor로부터 매일 3개의 rotor을 선택
        - 사전 연결되어 고정된 반사경



고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(11/13)
    - 코드북(Code Book)
      - 에니그마를 사용하기 위해 매일 다음을 포함하는 여러 설정을 제공
        - 다섯 개의 사용 가능한 Rotor 중에 선택된 세 개의 Rotor
        - Rotor가 설치되는 순서
        - 플러그보드의 설정
        - 그날의 세 문자 코드

고전 대칭-키 암호

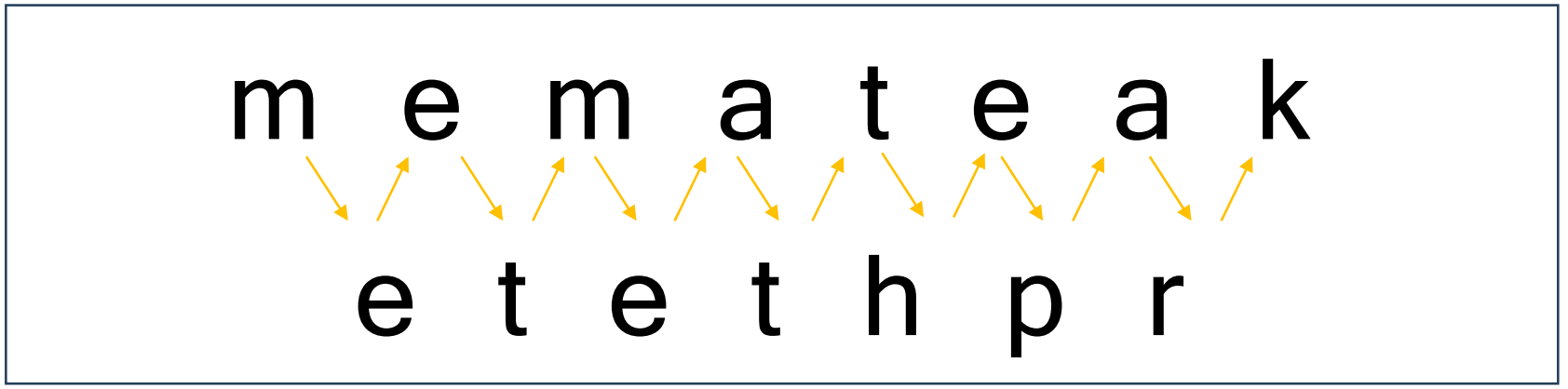
- 대치 암호(Substitution Cipher)
  - 다중문자 암호(12/13)
    - 에니그마 암호화하는 과정
      1. Rotor의 시작점을 그날의 코드로 맞춤
         - e.g., 코드가 “HUA”면 rotor는 각각 “H”, “U”, “A”로 초기화됨
      2. “ACF”와 같은 랜덤한 세 문자 코드를 선택함
         - 단계 1에서 rotor의 초기 설정(반복된 코드)을 이용하여 문장 “ACFACF”를 암호화함
         - e.g., 암호화된 코드가 “OPNABT”라고 가정함
      3. Rotor의 시작점을 OPN(암호화된 코드의 반)으로 놓음
      4. 단계 2에서 얻은 암호화된 6개 문자(“OPNABT”)를 메시지 시작에 추가함
      5. 그 6개의 문자 코드를 포함한 메시지를 암호화하고 암호화된 메시지를 보냄

고전 대칭-키 암호

- 대치 암호(Substitution Cipher)
  - 다중문자 암호(13/13)
    - 에니그마 복호화하는 과정
      1. 메시지를 받고 첫 번째 여섯 글자를 분리함
      2. Rotor의 시작점을 그날의 코드로 설정함
      3. 단계 2의 시작점을 사용해 처음 여섯 개 글자를 복호화함
      4. 복호화된 코드의 첫 번째 반으로 rotor의 위치를 설정함
      5. (첫 번째 여섯 글자 없이) 메시지를 복호화함

고전 대칭-키 암호

- 전치 암호(Transposition Cipher)
  - 문자의 위치를 바꾸는 암호
  - 키가 없는 전치 암호
    - Rail Fence 암호
      - 평문을 지그재그 패턴으로 두 열에 배열하여 암호화하는 암호
        - 암호문은 행 순서의 패턴으로 읽혀져 생성됨
      - Meet me at the park -> MEMATEAKETETHPR



고전 대칭-키 암호

- 전치 암호(Transposition Cipher)
  - 키가 있는 전치 암호
    - 평문을 블록으로 나눈 뒤 각각의 블록에서 문자를 치환하기 위해 따로 키를 사용함
    - e.g., Enemy attacks tonight 암호화
      - 마지막 블록은 다른 블록과 같은 크기를 갖도록 마지막에 가짜 문자 z 삽입

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

암호화 ↓

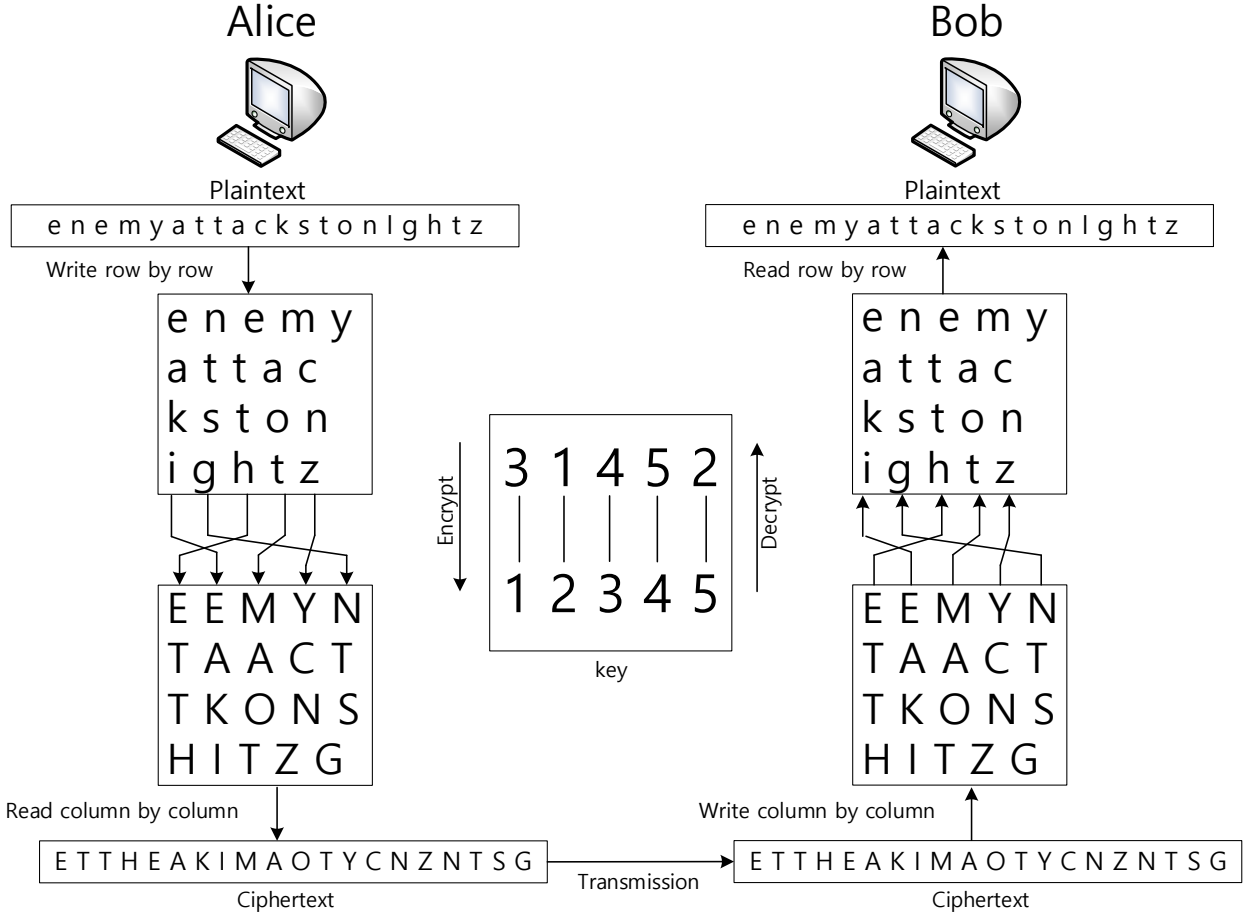
3	1	4	5	2
1	2	3	4	5

↑ 복호화

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

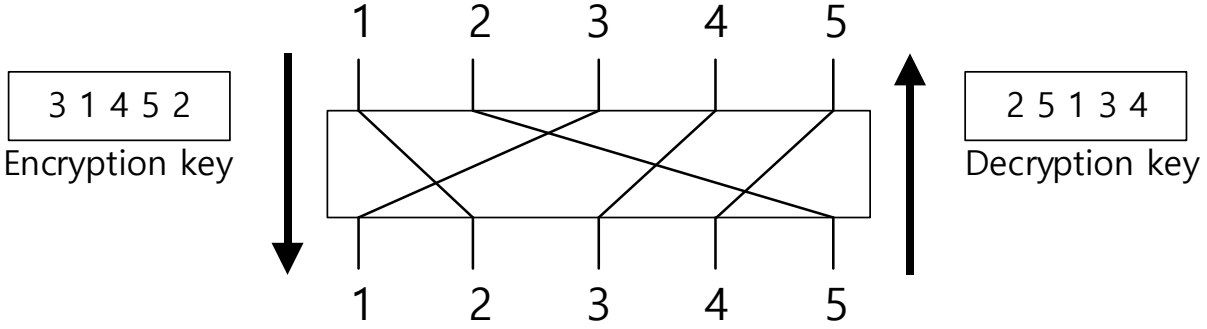
고전 대칭-키 암호

- 전치 암호(Transposition Cipher)
  - 두 가지 접근법의 결합(1/3)
    - 키가 있는 열 전치 암호 또는 열 전치 암호



고전 대칭-키 암호

- 전치 암호(Transposition Cipher)
  - 두 가지 접근법의 결합(2/3)
    - 키 생성
      - 하나는 암호화에 이용되고, 다른 하나는 복호화에 사용되는 키



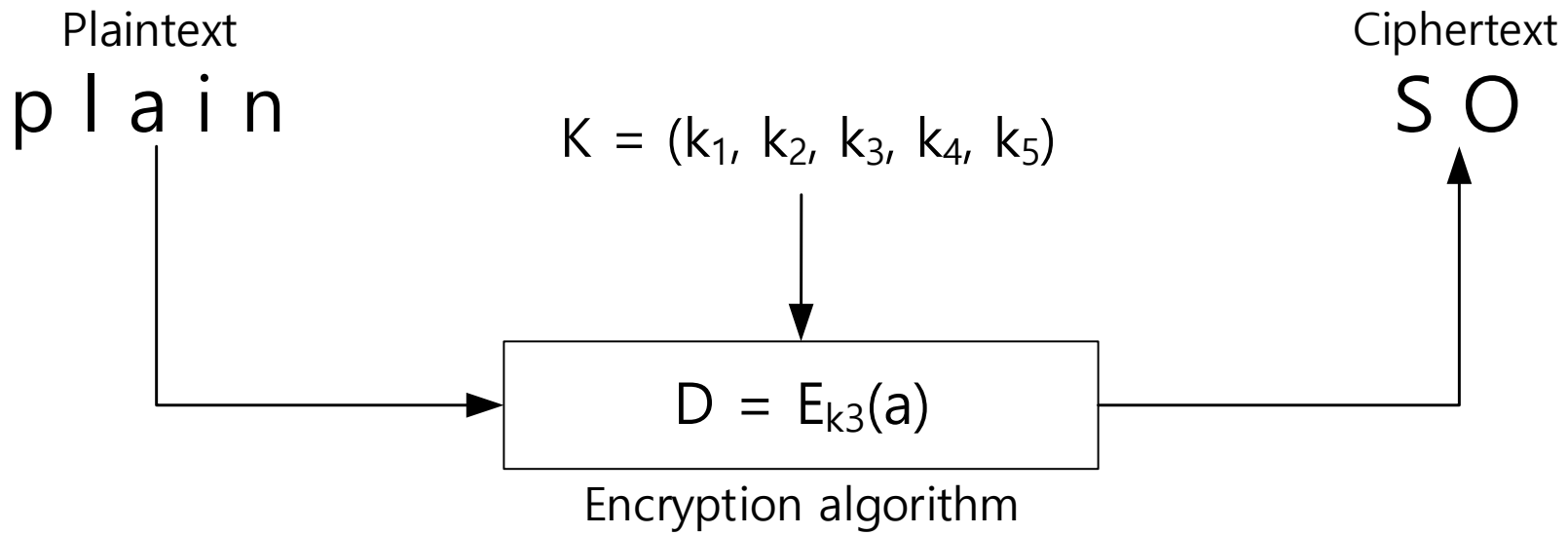
    - 행렬의 사용
      - 암호화는 평문 행렬에 키 행렬을 곱하여 암호문 행렬을 얻음
      - 복호화는 암호문에 키 행렬의 역행렬을 곱하여 평문 행렬을 얻음
      - 복호화 행렬은 암호화 행렬의 역행렬

고전 대칭-키 암호

- 전치 암호(Transposition Cipher)
  - 두 가지 접근법의 결합(3/3)
    - 전치 암호에 적용될 수 있는 공격
      - 여러 종류의 암호문 단독 공격에 취약함
      - 통계적인 공격
        - 전치 암호는 암호문의 문자 빈도를 변화시키지 않고, 단지 문자를 재정렬함
      - 전수 조사 공격
        - 공격자가 모든 가능한 키를 시도할 수는 있지만, 키의 개수는 매우 크기 때문에 열의 개수를 추측함
        - e.g., 암호문의 길이가 20일 때 20 = 1 x 2 x 2 x 5 (키의 길이: 1,2,4,5,10,20)
      - 패턴 공격
        - 키가 있는 전치 암호로 생성된 암호문은 어떤 반복되는 패턴을 가짐
        - e.g., **03 08 13 18** 01 06 11 16 **04 09 14 19** 05 10 15 20 **02 07 12 17**
    - 이중 전치 암호
      - 암호화, 복호화에서 사용되었던 알고리즘을 두 번 반복하는 암호

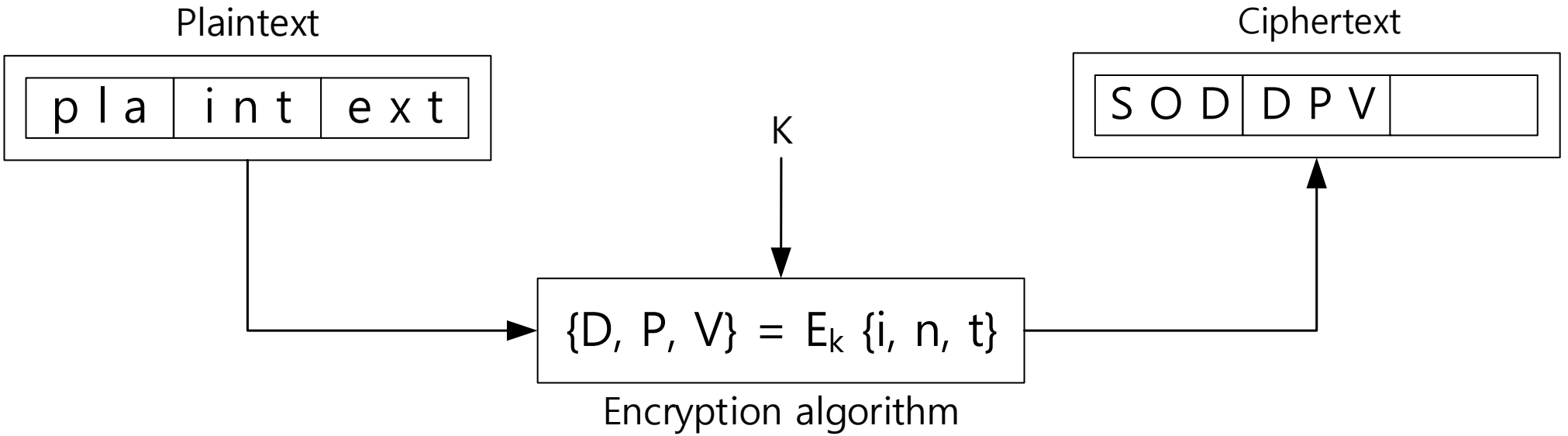
고전 대칭-키 암호

- 스트림 암호와 블록 암호
  - 스트림 암호(Stream Cipher)
    - 암호화와 복호화가 한 번에 한 개의 문자에 적용되는 암호
    - 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(Block Cipher)
    - 평문을 고정된 크기 $m(m > 1)$의 블록으로 나누어 각 블록을 개별적으로 암호화하는 암호



Thanks!

이 은 진(eunjin@pel.sejong.ac.kr)