출석 수업

Computer Security
컴퓨터보안

# 컴퓨터 보안

컴퓨터 과학과 김진욱 교수

I

# 컴퓨터 보안의 개요



컴퓨터보안
Computer Security

Chapter 01

# 컴퓨터 보안의 개념

## 정보보호의 개념

Computer Security

- **정보를 여러 가지 위협으로부터 보호하기 위한 정책 및 기법**
  - 정보의 상태: 저장, 전달
  - 위협의 종류: 허락되지 않는 접근, 수정, 훼손, 유출 등

## 컴퓨터 보안의 개념

Computer Security

- 정보보호의 한 영역
- 컴퓨팅 환경이 관여된 모든 상황에 대한 정보보호

- 컴퓨팅 환경에 저장되거나 처리되는 정보를 다양한 위협으로부터 보호하기 위한 정책 및 기법

컴퓨터보안
Computer Security

Chapter 02 정보보호의 목표

## 정보보호의 목표

Computer Security

- **정보보호의 핵심목표**
  - 기밀성(Confidentiality)
  - 무결성(Integrity)
  - 가용성(Availability)


CIA triad

## 정보보호의 핵심목표

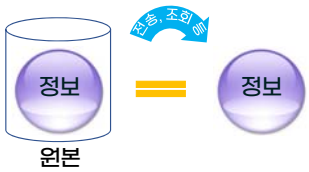
Computer Security

- **기밀성**
  - 허락되지 않은 자가 정보의 내용을 알 수 없도록 하는 것


  - 예: 은행에서 고객의 개인정보나 계좌정보 같은 기밀정보가 제3자에게 알려지는 것을 방지하기 위해 이를 보호
  - 기밀성을 지키는 방법:
    - 허락되지 않은 자가 정보에 접근을 아예 못하도록 함
    - 정보에 접근하더라도 무의미한 내용만 보이도록 함

## 정보보호의 핵심목표

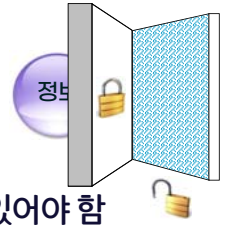
- **무결성**
  - 허락되지 않은 자가 정보를 임의로 수정할 수 없도록 하는 것




  - 예: DB 내 고객의 개인정보가 임의로 수정되지 않도록 보호
    고객 본인이 조회할 때 DB에서 고객까지 전달과정에서 위변조되지 않도록 보호
  - 만약 허락되지 않은 자에 의한 수정이 발생했다면 이를 확인할 수 있는 것

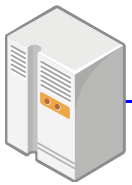
## 정보보호의 핵심목표

- **가용성**
  - 허락된 자가 정보에 접근하고자 할 때 이것이 방해받지 않도록 하는 것
  - 즉, 정보에 대한 접근권한이 있는 자는 필요할 때 언제든지 정보를 사용할 수 있어야 함




  - 예: 고객이 본인의 개인정보를 확인하고자 할 때 즉시 조회가 가능하게 하는 것
  - 정해진 시간 내에 정보를 볼 수 있음을 보장

## [예제] 자동화기기(ATM)



은행 서버


- **기밀성**
  - 비밀번호
- **무결성**
  - 계좌번호, 입출금정보
- **가용성**
  - 서버, 자동화기기, 네트워크

## 정보보호의 목표

- **정보보호의 핵심목표**
  - 기밀성(Confidentiality)
  - 무결성(Integrity)
  - 가용성(Availability)


- **그 외의 목표**
  - 부인방지(non-repudiation)
  - 인증(authentication)
  - 접근제어(access control) 등

II

# 암호의 개념

01 암호의 정의

02 암호의 역사

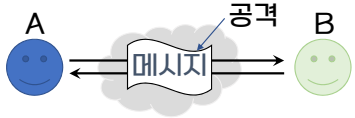
컴퓨터보안
Computer Security

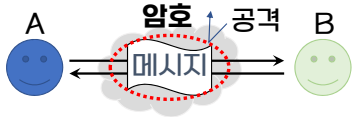
Chapter 01 암호의 정의

## 암호의 정의

Computer Security

- 두 사람이 안전하지 않은 채널을 통해 정보를 주고받더라도 제3자는 이 정보의 내용을 알 수 없도록 하는 것

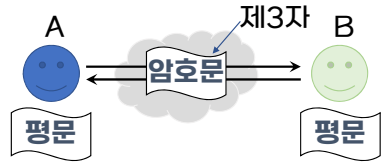



## 암호의 정의

Computer Security

- 두 사람이 안전하지 않은 채널을 통해 정보를 주고받더라도 제3자는 이 정보의 내용을 알 수 없도록 하는 것




- 평문: 원래의 메시지 (plaintext)
- 암호문: 코드화된 메시지 (ciphertext)
- 암호화: 평문 → 암호문 (encryption)
- 복호화: 암호문 → 평문 (decryption)

- 키(key): 암호화와 복호화를 위한 가장 중요한 열쇠
- 암호는 기밀성을 보장하기 위한 필수적인 기술

컴퓨터보안
Computer Security

Chapter 02 암호의 역사

## 고대 암호

Computer Security

- 스테가노그래피(steganography)
  - 실제로 전달하고자 하는 정보 자체를 숨기는 것
  - 최초의 암호는 BC 480년경, 스파르타에서 추방되어 페르시아에 살던 데마라토스가 페르시아의 침략계획 소식을 나무판에 조각 후 밀랍을 발라 스파르타에 보낸 것

## 고대 암호

Computer Security

- 일반적인 암호의 요건(Kerckhoff의 원리)
  - 제3자에게 암호 알고리즘을 알려주더라도 제3자가 키를 모르면 암호를 풀 수 없다는 것을 가정
- 스테가노그래피를 최초의 암호로 보기는 힘듦
- 두 가지 암호 방식
  - 전치법
  - 치환법

## 고대 암호

Computer Security

- 전치법(permutation 혹은 transposition cipher)
  - 평문에 있는 문자들의 순서를 바꿈으로써 암호화하는 기법

암호알고리즘 → (암호화) → 호암고알즘리

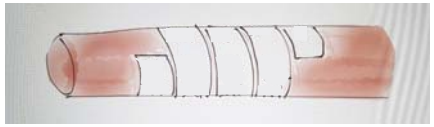
  - 가장 단순한 방식: 두 문자씩 앞뒤로 섞는 방법

암호화: 암호알고리즘 → 호암고알즘리
복호화: 호암고알즘리 → 암호알고리즘

## 고대 암호

- **전치법(permutation 혹은 transposition cipher)**
  - 평문에 있는 문자들의 순서를 바꿈으로써 암호화하는 기법
  - 스파르타의 봉 암호 키:

컴퓨터보안재미있는과목입니다~!



컴안느니퓨재과다터미목 ~ 보잤입!

## 고대 암호

- **치환법(substitution cipher)**
  - 평문의 문자들을 다른 문자로 치환함으로써 암호화하는 기법

암호알고리즘 → 암호화 → S&/7무B

  - 치환 규칙에 따라 암호화 및 복호화

| 평문 문자 | 고 | 리 | 알 | 암 | 즘 | 호 |
|---|---|---|---|---|---|---|
| 암호문 문자 | 7 | 무 | / | S | B | & |

암호화: 암호알고리즘 → S & / 7 무 B

복호화: S & / 7 무 B → 암호알고리즘

## 고대 암호

- **치환법(substitution cipher)**
  - 평문의 문자들을 다른 문자로 치환함으로써 암호화하는 기법
  - 시저 암호: 평문의 각 문자를 알파벳 순서상 세 문자 뒤에 위치하는 문자로 치환

| 평문 문자 | 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 문자 | 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 |

암호화: CRYPTO → FUBSWR

복호화: FUBSWR → CRYPTO

## 고대 암호

- **치환법(substitution cipher)**
  - 평문의 문자들을 다른 문자로 치환함으로써 암호화하는 기법
  - 시프트 암호: 평문의 각 문자를 알파벳 순서상 $k$번째 뒤 문자로 치환( $0 \leq k \leq 25$ )

| 평문 문자 | 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 문자 | 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 |

$k = 4$

암호화: CRYPTO → GVCTXS

복호화: GVCTXS → CRYPTO

## 고대 암호

Computer Security

- **치환법(substitution cipher)**
  - 평문의 문자들을 다른 문자로 치환함으로써 암호화하는 기법
  - 시프트 암호: 평문의 각 문자를 알파벳 순서상 $k$번째 뒤 문자로 치환( $0 \leq k \leq 25$ )

| 평문 문자 | 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 문자 | | | | | | | | | | | | | | | | | | | | | | | | | | |

$k$ =

KNOU

암호화 복호화

## 근대 암호

Computer Security

- **비즈네르 암호(Vigenère cipher)**
  - 시프트 암호를 개선한 새로운 치환법
  - 키: 여러 개의 정숫값
  - 예: $k = 3, 5, 0$

암호화 $k = 3\downarrow 5\downarrow 0\downarrow 3\downarrow 5\downarrow 0\downarrow$

CRYPTO → FWYSYO

복호화 $k = 3\downarrow 5\downarrow 0\downarrow 3\downarrow 5\downarrow 0\downarrow$

FWYSYO → CRYPTO

- **20세기 초반까지 플레이페어 암호, 힐 암호 등 다양한 암호방식 등장**

## 근대 암호

Computer Security

- **20세기 들어 암호에 대한 연구가 활발하게 진행됨**
  - 통신기술의 발전, 기계식 계산기에 대한 연구
  - 두 차례의 세계대전을 통해 암호설계와 해독에 대한 필요성 증가
- **1949년 섀넌(Shannon)**
  - 일회성 암호체계(one-time pad)가 안전함을 증명
  - 암호체계 설계의 두 가지 기본원칙 제시
    - 혼돈(confusion): 평문과 암호문 사이의 상관관계를 숨김
    - 확산(diffusion): 평문의 통계적 성격을 암호문 전반에 확산시켜 숨김

## 현대 암호

Computer Security

- **1970년대 두 가지 큰 변화 발생**
  - 표준 암호 알고리즘의 등장
  - 공개키 암호 알고리즘의 등장

## 현대 암호

Computer Security

- **표준 암호 알고리즘의 등장**
  - 컴퓨터가 점차 발전하면서 데이터 보호에 대한 필요성도 증가
  - 1977년 미국 NBS(현재 NIST)에서 표준 암호 알고리즘 공표
    - DES(Data Encryption Standard): 대칭키 암호 알고리즘
  - 2001년 새로운 표준 암호 알고리즘인 AES가 공표될 때까지 널리 이용됨

## 현대 암호

Computer Security

- **공개키 암호 알고리즘의 등장**
  - 1976년 디피(Diffie)와 헬먼(Hellman)이 공개키 암호의 개념을 제시
    - 공개키 암호: 암호화와 복호화에 서로 다른 키를 사용
  - 1978년 리베스트(Rivest), 샤미르(Shamir), 애들먼(Adleman)이 RSA 공개키 암호 알고리즘 개발
    - RSA: 소인수분해 문제에 기반을 둔 대표적인 공개키 암호 알고리즘

III

# 대칭키 암호



컴퓨터보안
Computer Security

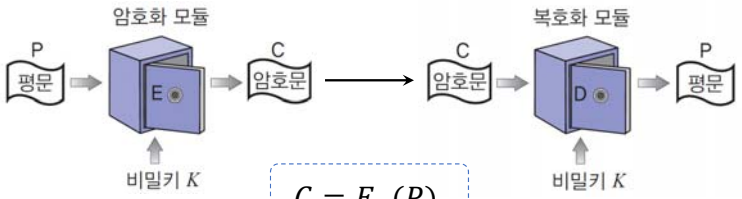
Chapter 01

# 대칭키 암호의 개념

## 대칭키 암호

Computer Security

- 암호화와 복호화에 같은 키 하나를 사용하는 암호방식



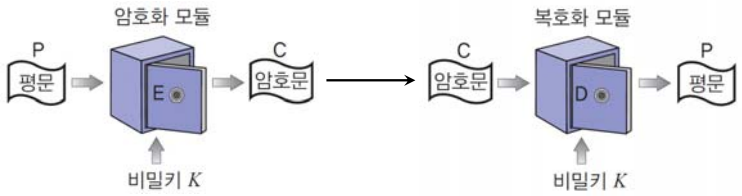

$$C = E_K(P)$$
$$P = D_K(C)$$

- 다양한 이름
  - 대칭키 암호, 비밀키 암호, 단일키 암호, 관용 암호

## 대칭키 암호

Computer Security

- 암호화와 복호화에 같은 키 하나를 사용하는 암호방식

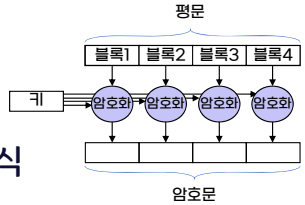



- 장점: 암호화와 복호화 속도가 빠름
- 단점: 키 분배 문제 존재
- 대표적인 알고리즘: DES, AES, IDEA 등

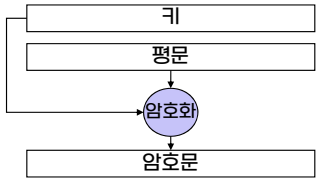
## 대칭키 암호의 분류

Computer Security

- 블록 암호
  - 평문을 고정된 크기의 블록으로 나누어 각 블록마다 암호화 과정을 수행하여 블록 단위로 암호문을 얻는 대칭키 암호 방식

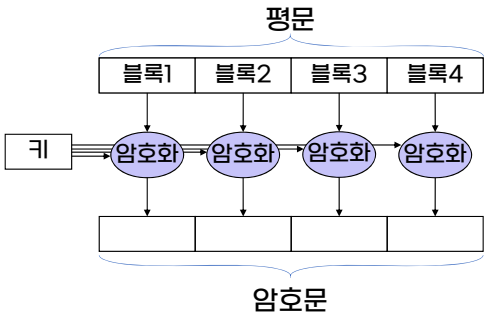



- 스트림 암호
  - 평문과 같은 길이의 키 스트림을 생성하여 평문과 키를 비트 단위로 XOR하여 암호문을 얻는 대칭키 암호 방식



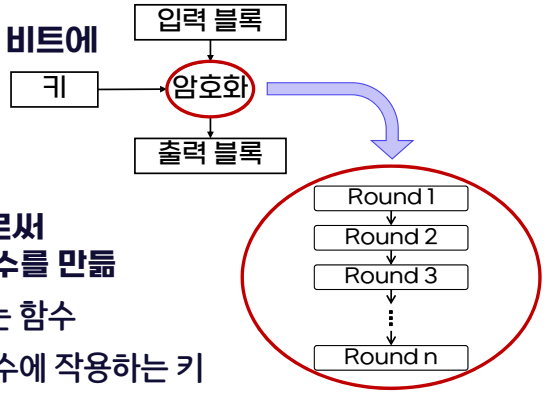

## 블록 암호

- 평문을 고정된 크기의 블록으로 나누어 각 블록마다 암호화 과정을 수행하여 블록 단위로 암호문을 얻는 대칭키 암호 방식




## 블록 암호 알고리즘의 구조

- 출력 블록의 각 비트는 입력 블록과 키의 모든 비트에 영향을 받음




- 주로 단순한 함수를 반복적으로 적용함으로써 암호학적으로 강한 함수를 만듦
  - 라운드 함수: 반복되는 함수
  - 라운드 키: 라운드 함수에 작용하는 키
  - 키 스케줄: 키를 입력하여 라운드 키를 발생시키는 과정

## 블록 암호 알고리즘의 구조

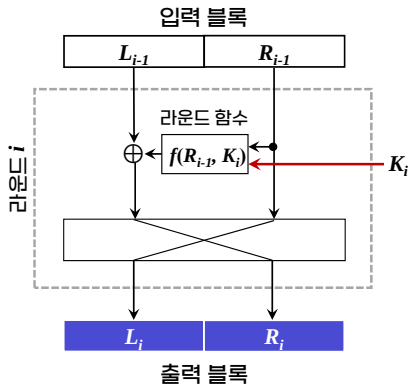
- 파이스텔 구조
- SPN 구조

## 파이스텔(Feistel) 구조

- 하나의 입력 블록을 분할하여 좌우 두 개의 블록으로 구분 후 짝수 번의 라운드를 진행
- 각 라운드의 출력 블록이 다음 라운드의 입력 블록이 됨
- $i$번째 라운드 처리과정

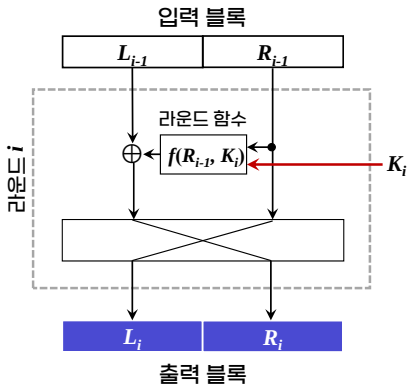
$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

  - $f$: 라운드 함수
  - $K_i$: 라운드 키



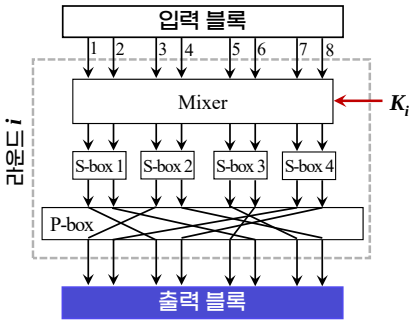
## 파이스텔(Feistel) 구조

- 라운드 함수와 관계없이 역변환 가능
- 두 번의 수행으로 블록 간의 완전한 확산이 이루어짐
- DES, SEED 등 많은 블록 암호에 사용됨



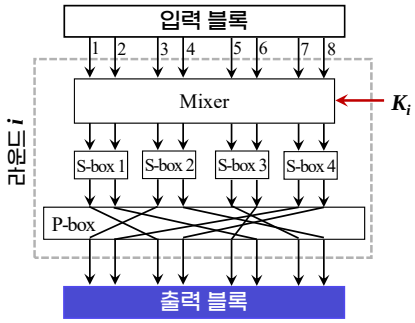

## SPN(Substitution Permutation Network) 구조

- 하나의 입력 블록을 여러 개의 소블록으로 나눈 후 라운드를 진행
- 각 라운드의 출력 블록이 다음 라운드의 입력 블록이 됨
- $i$번째 라운드 처리과정
  - 각 소블록을 S-box로 입력하여 치환
  - S-box 출력을 P-box로 전치




## SPN(Substitution Permutation Network) 구조

- 라운드 함수가 역변환 가능해야 함
- 더 많은 병렬성을 제공
- AES, ARIA 등 최근의 블록 암호에 사용됨

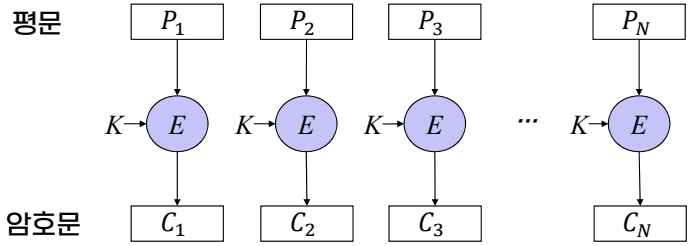



## 블록 암호의 사용 모드

- 각 블록에 블록 암호를 어떤 방식으로 사용할 것이냐에 따라 구분
  - 전자 코드 북(ECB) 모드
  - 암호 블록 연결(CBC) 모드
  - 암호 피드백(CFB) 모드
  - 출력 피드백(OFB) 모드
  - 카운터(CTR) 모드

## 전자 코드 북(ECB) 모드

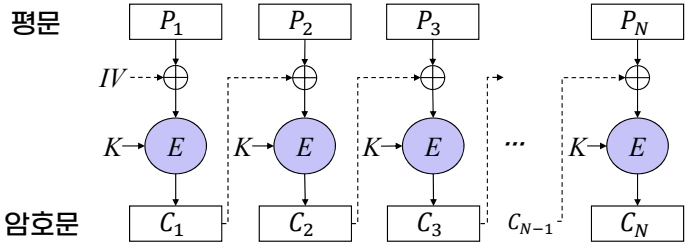
- Electronic Code Book




- 암·복호화 시 병렬처리 가능
- 암호문 블록의 오류가 다른 블록에 영향을 미치지 않음
- 동일한 평문 블록은 동일한 암호문 생성(패턴 분석 가능)

## 암호 블록 연결(CBC) 모드

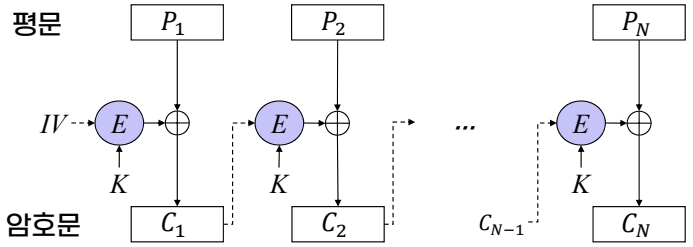
- Cipher Block Chaining




- 암호화 시 병렬처리 불가능
- 암호화 시 평문 블록 오류가 그 다음 모든 암호문에 영향
  → 메시지 인증에 사용

## 암호 피드백(CFB) 모드

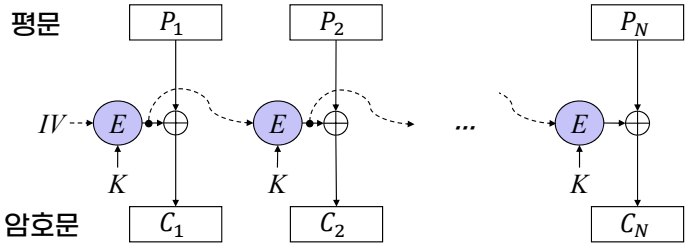
- Cipher FeedBack




- 암호화 시 특정 입력이 이후로 영향을 미침
  → 메시지 인증에 사용
- 복호화 함수 필요 없음

## 출력 피드백(OFB) 모드

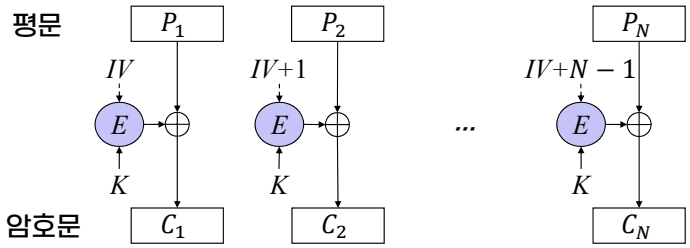
- Output FeedBack




- 암호문 블록의 오류는 한 블록에만 영향을 미침
  → 영상이나 음성 같은 디지털 신호화된 아날로그 신호에 사용
- 복호화 함수 필요 없음

## 카운터(CTR) 모드

- CounTeR



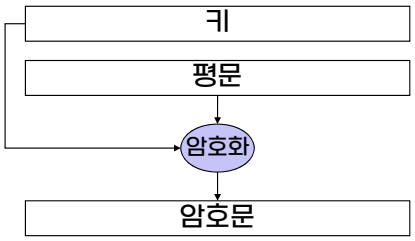

- 암·복호화 시 병렬처리 가능
- 오류의 확산이 일어나지 않음
- 복호화 함수 필요 없음

컴퓨터보안
Computer Security

Chapter 03 스트림 암호

## 스트림 암호

- 평문과 같은 길이의 키 스트림을 생성하여 평문과 키를 비트 단위로 XOR하여 암호문을 얻는 대칭키 암호 방식



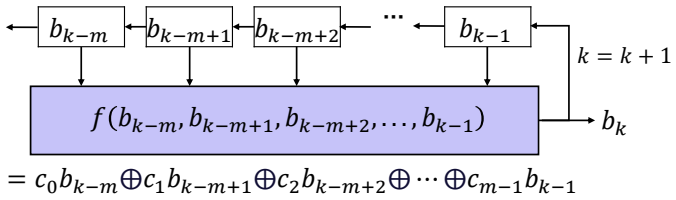

## 키 스트림

- 임의의 길이의 평문이 주어져도 동일한 길이의 키 스트림 생성 가능
- 규칙성이 없어 예측이 불가능한 랜덤 수열이 가장 안전
- 의사 랜덤(pseudorandom) 수열 생성
  - 예측이 어려우면서도 자동화된 생성이 가능
  - 예: LFSR

## 선형 귀환 시프트 레지스터(LFSR)

Computer Security

- Linear Feedback Shift Register
- 직전 $m$개의 비트값을 선형 결합하여 새로운 한 비트값을 생성




$= c_0 b_{k-m} \oplus c_1 b_{k-m+1} \oplus c_2 b_{k-m+2} \oplus \cdots \oplus c_{m-1} b_{k-1}$

- 주기: 최대 $2^m - 1$ 비트 길이

## 선형 귀환 시프트 레지스터(LFSR)

Computer Security

- 예: $b_k = b_{k-1} \oplus b_{k-3}$, 초깃값 110
  - $b_3 = b_2 \oplus b_0 = 0 \oplus 1 = 1$
  - $b_4 = b_3 \oplus b_1 = 1 \oplus 1 = 0$
  - $b_5 = b_4 \oplus b_2 = 0 \oplus 0 = 0$
  - $b_6 = b_5 \oplus b_3 = 0 \oplus 1 = 1$
  - ...
  - 키 스트림: 1101001 1101001 1101001 ⋯
- LFSR 단독 사용은 쉽게 해독됨

컴퓨터보안
Computer Security

Chapter 04

# 대칭키 암호 알고리즘

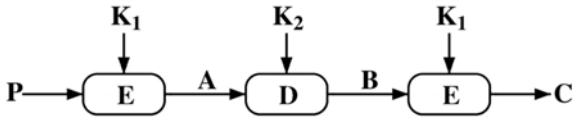
## DES

Computer Security

- Data Encryption Standard
- 1977년 미국에서 데이터 암호 알고리즘의 표준으로 공표
- 블록 암호 알고리즘
  - 블록 크기: 64 bits
  - 키 길이: 56 bits
  - 파이스텔 구조: 16 라운드, 라운드 키 길이 48 bits
- 컴퓨터 속도 개선과 암호해독기술의 발전으로 2001년 AES에 표준 자리를 물려줌

## TDEA

- Triple Data Encryption Algorithm
- DES를 3회 반복(3DES)

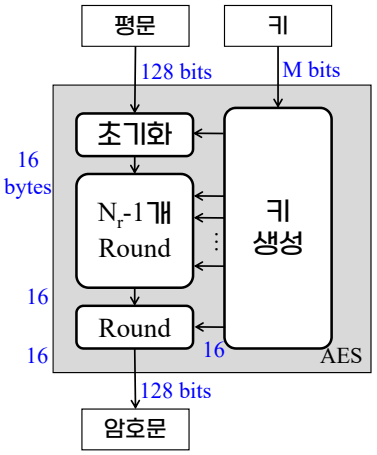



- DES의 짧은 키 길이로 인한 안전성 문제 해결
- DES보다 3배 정도 느림

## AES

- Advanced Encryption Standard
- DES를 대신하는 새로운 표준
- 2001년 미국 NIST에서 공표
  - 공모를 통해 Rijndael을 AES로 선정
- 블록 암호 알고리즘
  - 블록 크기: 128 bits
  - 키 길이: 128 bits, 192 bits, 256 bits 중 택일
  - SPN 구조

## AES 구성




| 키 길이 M bits | 라운드 수 $N_r$ |
|---|---|
| 128 | 10 |
| 192 | 12 |
| 256 | 14 |

## AES 암호화 과정

- 라운드 1 ~ 라운드 $N_r - 1$
  - SubBytes: 비선형성을 갖는 S-Box로 바이트 단위 치환
  - ShiftRows: 행 단위로 순환 시프트
  - MixColumns: 열 단위로 혼합
  - AddRoundKey: 라운드 키를 XOR
- 라운드 $N_r$
  - SubBytes
  - ShiftRows
  - AddRoundKey

# IV 공개키 암호



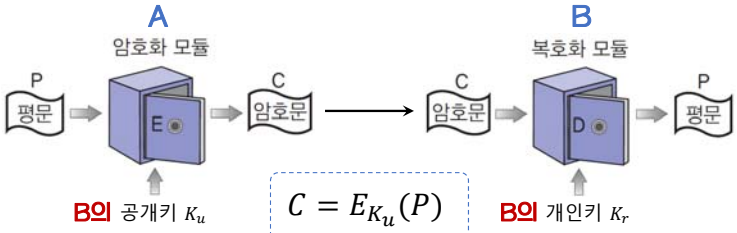
컴퓨터보안
Computer Security

Chapter 01 공개키 암호의 개념

## 공개키 암호

Computer Security

- 암호화와 복호화에 **두 개의 서로 다른 키**를 사용하는 암호방식



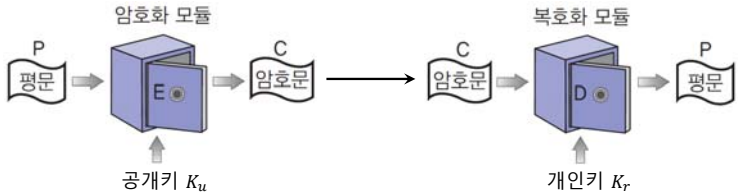

$$C = E_{K_u}(P)$$
$$P = D_{K_r}(C)$$

- **한 쌍의 키**
  - 공개키: 누구나 공개키를 이용하여 암호화 가능
  - 개인키: 오직 자신만 개인키를 이용하여 복호화 가능

## 공개키 암호

Computer Security

- 암호화와 복호화에 **두 개의 서로 다른 키**를 사용하는 암호방식




- **장점:** 키 관리 쉬움, 키 분배 문제 해결
- **단점:** 대칭키 암호에 비해 속도가 느림
- **대표적인 알고리즘:** RSA, ECC, ElGamal 등

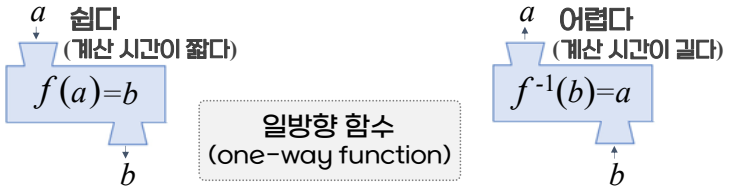
컴퓨터보안
Computer Security

Chapter 02 기반 문제

## 기반 문제

Computer Security

- 공개키 암호 알고리즘은 수학적으로 어려운 문제들에 기반




- **다양한 일방향 함수**
  - 소인수분해 문제, 이산대수 문제, 타원곡선 이산대수 문제 등

## 소인수분해 문제

Computer Security

- 어떤 두 정수의 곱은 빠른 시간에 구할 수 있지만, 임의의 양의 정수의 소인수분해는 매우 어려움
  - 2 x 5 =
  - 9539 x 8887 =
  - [redacted] x [redacted] =
    256 bits 256 bits
  - 10 =
  - 8477 3093 =
  - [redacted] =
    512 bits
    8,000 MIPS-Year
- 소인수분해 기반 알고리즘: RSA
- 안전한 암호로 사용하기 위한 키의 크기: 2,048 bits

## 이산대수 문제

Computer Security

- 양의 정수 $n$, $a$, $x$에 대하여 $a^x \pmod n$은 빠른 시간에 구할 수 있지만, 양의 정수 $n$, $a$, $y$에 대하여 $y = a^x \pmod n$인 $x$를 구하는 것은 매우 어려움
  - $n = 11, a = 2, x = 6$
    $2^6 \pmod{11} =$
  - $n = 11, a = 2, y = 9$
    $9 = 2^x \pmod{11}$
    $x =$
- 이산대수 문제 기반 알고리즘: ElGamal, DSA, KCDSA, DHKE 프로토콜
- 안전한 암호로 사용하기 위한 키의 크기: 2,048 bits

## 타원곡선 이산대수 문제

- **타원곡선상의 점과 타원곡선에서 정의되는 덧셈 연산을 이용하여 정의되는 이산대수 문제**
  - 정수에서의 이산대수 문제처럼 일방향 함수의 성질을 가짐
- **타원곡선 이산대수 문제 기반 알고리즘: ECDSA, EC-KCDSA**
- **안전한 암호로 사용하기 위한 키의 크기: 224 bits**


## RSA 암호 알고리즘

- **1978년 Rivest, Shamir, Adleman이 개발**
- **국제암호표준으로 활용**
- **많은 응용 분야에서 전 세계적으로 활용**
  - 신용카드 결제, 증권거래, 이메일 등
  - 암호키의 안전한 분배 및 관리
- **소인수분해 문제에 기반**
  - 자릿수는 비슷하지만 두 수의 차가 큰 서로 다른 두 소수 $p, q$ 이용

## RSA 암호 알고리즘

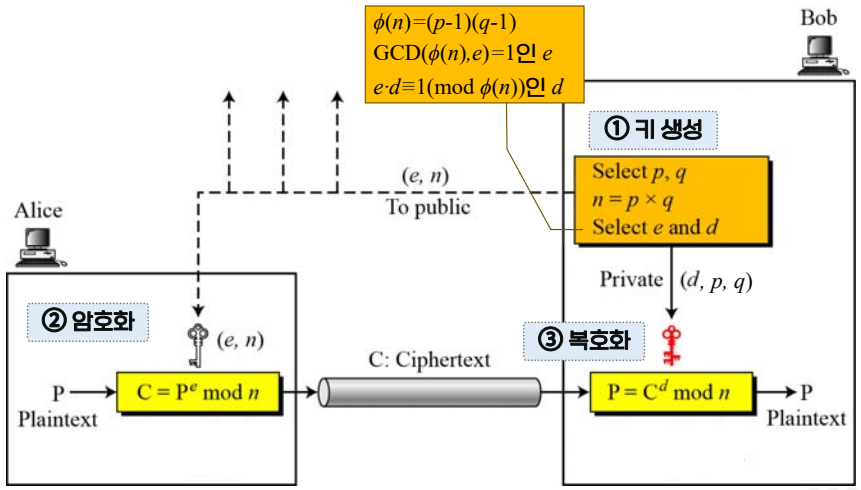
- **키 생성(수신자)**
  - 공개키 $(e, n)$ 공개
  - 개인키 $(d, p, q)$ 준비
- **암호화(송신자)**
  - $n$ 보다 작은 숫자인 평문 $M$을 수신자의 공개키 $(e, n)$ 이용하여 암호문 $C$ 계산

$$C = M^e \pmod{n}$$

- **복호화(수신자)**
  - 암호문 $C$를 수신자의 개인키 $(d, p, q)$ 이용하여 복호화

$$P = C^d \pmod{n}$$

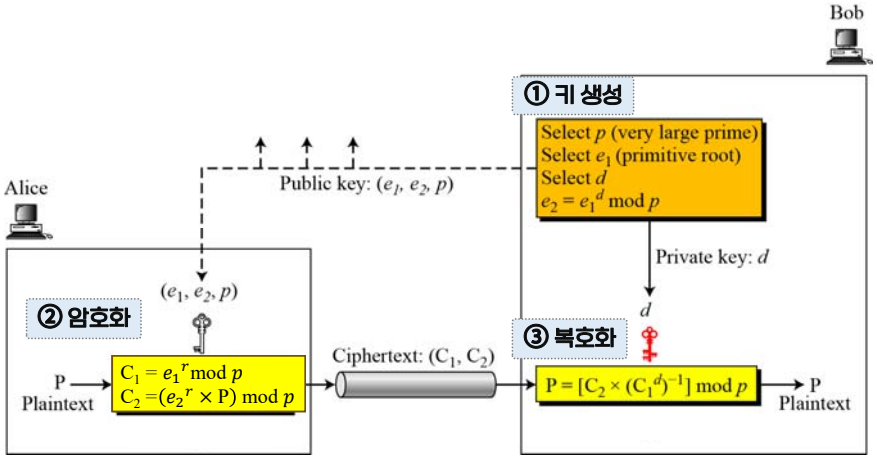
# RSA 암호 알고리즘




# ElGamal 암호 알고리즘

- 1985년 ElGamal이 제안
- 유한체상에서의 이산대수 문제에 기반
  - 수신자의 공개키를 가지고 개인키를 계산하는 것은 이산대수 문제
- Diffie-Hellman 키 교환과 같은 원리를 이용

# ElGamal 암호 알고리즘

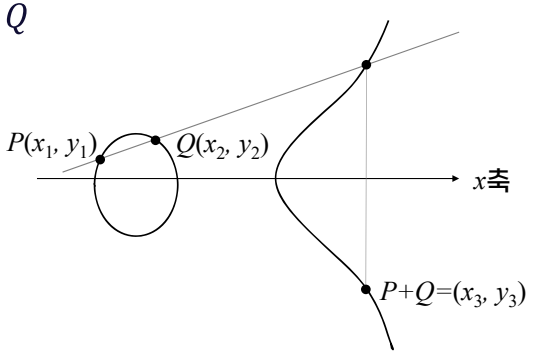



# 타원곡선 암호 알고리즘

- 1985년 Miller와 Koblitz가 독립적으로 제안
- 유한체상에서 정의된 타원곡선 군에서의 이산대수 문제에 기반
- RSA, ElGamal 등과 동일한 수준의 보안성을 제공하면서도 키의 길이는 짧음
- 이산대수 문제에 기반을 둔 ElGamal 암호 알고리즘 등을 변환하여 타원곡선 암호 알고리즘으로 적용

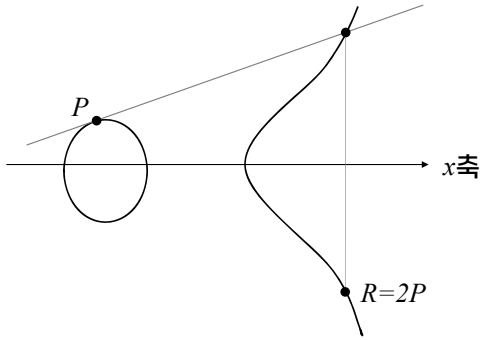
## 타원곡선상에서의 덧셈 연산

- 타원곡선 $y^2 = x^3 + ax + b$
  - $P + Q$



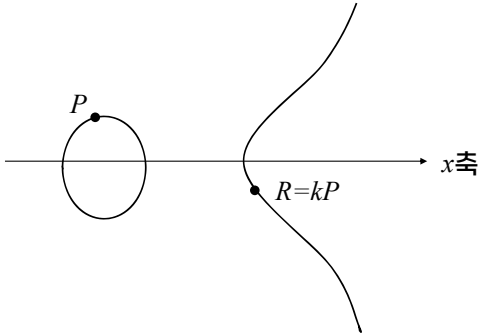

## 타원곡선상에서의 덧셈 연산

- 타원곡선 $y^2 = x^3 + ax + b$
  - $P + P$




## 타원곡선 이산대수 문제

- $kP$는 빠른 시간에 구할 수 있지만, 두 점 $R$과 $P$로부터 $R = kP$인 $k$를 구하는 것은 매우 어려움




출석 수업

## 수고 많으셨습니다.