2026/04/14, 2026 보안 기초 세미나

암호학과 네트워크 보안

- 1장 보안개요, 2장 암호수학 -

김 민 식 (ms6127@naver.com)

세종대학교 프로토콜공학연구실

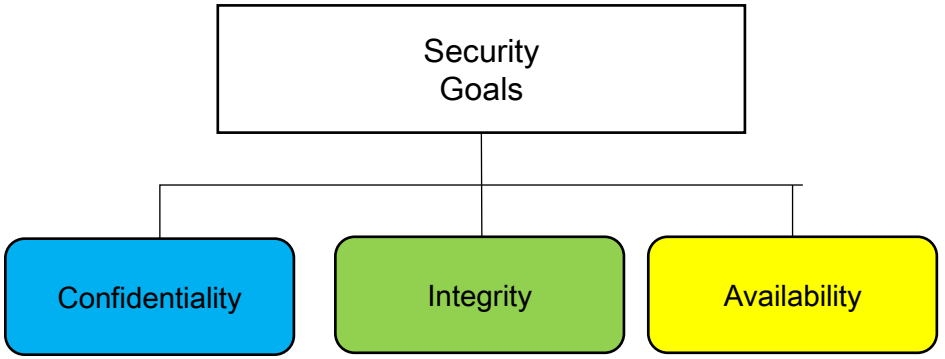
목 차

- 암호 개요
- 암호 수학

목 차

보안 개요

- 정의
  - 정보와 시스템을 무단 접근, 변조, 파괴로부터 보호하여 기밀성, 무결성, 가용성을 보장
    - 보안 목표
      - 기밀성(Confidentiality)
        - 허가된 사용자만 정보에 접근할 수 있도록 보장
      - 무결성(Integrity)
        - 정보가 허가 없이 변경되지 않고 정확하게 유지되도록 보장
      - 가용성(Availability)
        - 필요할 때 언제든지 시스템과 정보를 사용할 수 있도록 보장



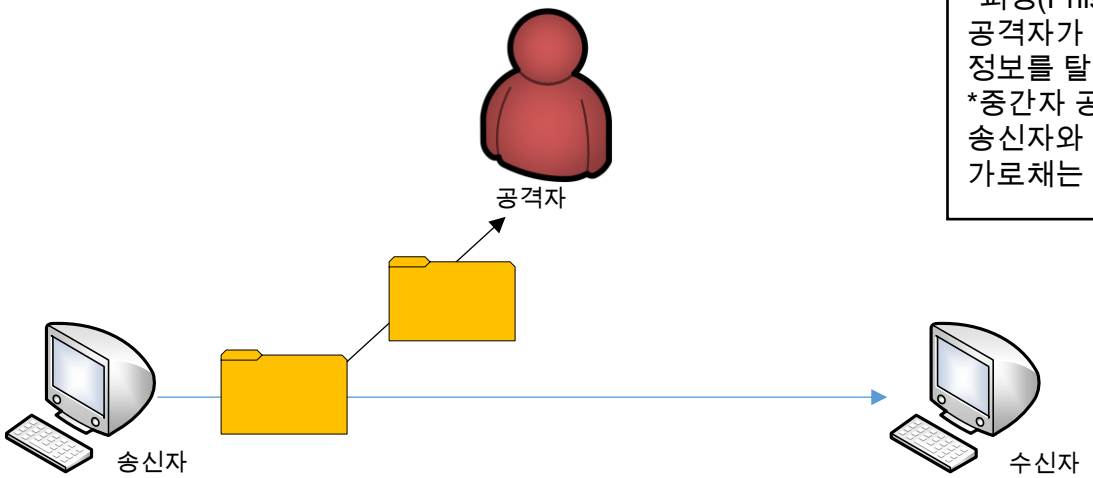
보안 개요

- 공격(Attack)
  - 시스템이나 정보를 무단으로 접근, 변경, 파괴하려는 행위
  - 보안 목표와 관련하여 세 가지 유형으로 구분됨
    - 기밀성을 위협하는 공격
      - e.g., 스누핑(Snooping), 트래픽 분석(Traffic Analysis)
    - 무결성을 위협하는 공격
      - e.g., 변경(Modification), 가장(Masquerading), 재전송(Replay), 부인(Repudiation)
    - 가용성을 위협하는 공격
      - e.g., 서비스 거부 공격(DoS, Denial of Service)

보안 개요

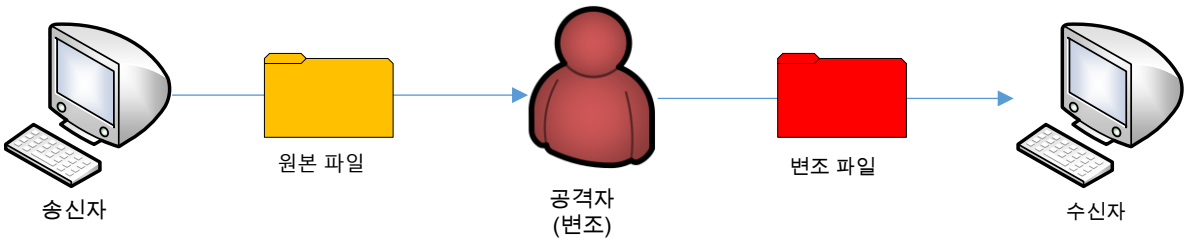
- 기밀성을 위협하는 공격
  - 스누핑(Snooping)
    - 네트워크나 시스템에서 데이터를 비인가적으로 열람하거나 감청하는 행위
  - 트래픽 분석(Traffic Analysis)
    - 통신 내용이 아닌 트래픽 패턴(빈도, 시간, 크기)을 분석하여 정보를 추론하는 공격

*피싱(Phishing)
공격자가 신뢰할 수 있는 기관으로 위장해 사용자의 개인정보나 인증 정보를 탈취하는 공격
*중간자 공격(Man-in-the-Middle, MITM)
송신자와 수신자 사이에 공격자가 끼어들어 통신 내용을 도청하거나 가로채는 공격



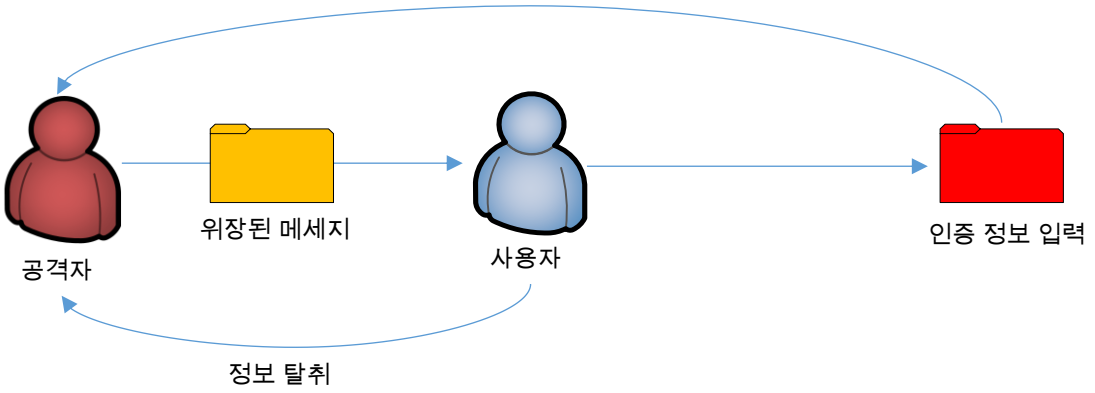
보안 개요

- 무결성을 위협하는 공격 (1/4)
  - 변경(Modification)
    - 전송되거나 저장된 데이터를 허가 없이 바꾸는 공격
      - 무결성을 침해하는 공격
      - 공격자가 중간에서 데이터 조작
      - 수신자 측에서는 정상 데이터로 인식되도록 위장
        1. 송신자가 데이터 전송
        2. 공격자가 중간에서 가로챔
        3. 데이터를 수정
        4. 수정된 데이터를 수신자에게 전달



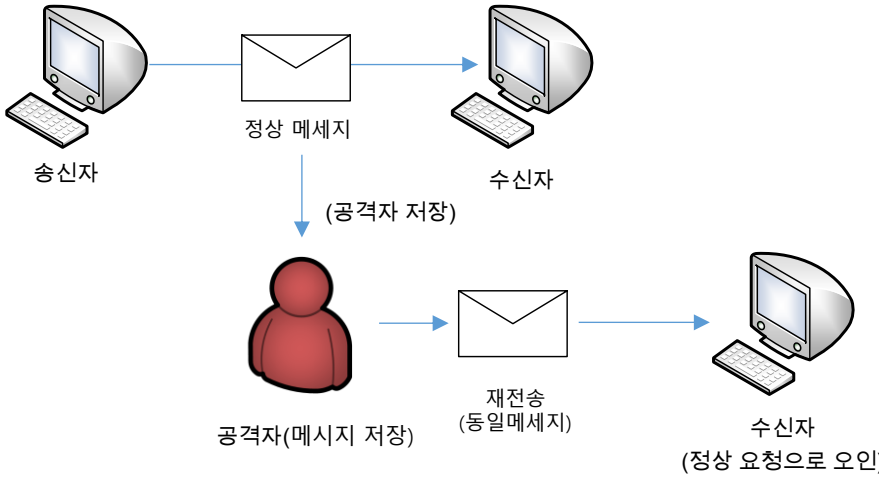
보안 개요

- 무결성을 위협하는 공격 (2/4)
  - 가장(Masquerading)
    - 공격자가 인가된 사용자나 시스템인 것처럼 속이는 공격
      1. 공격자가 신원을 위장
      2. 정상적인 것으로 위장된 메세지를 피해자에게 전달
      3. 피해자가 공격자에 의해 위장된 메시지를 신뢰하고 응답
      4. 인증 정보 획득
         - 가장 공격은 기술적 취약점보다 사용자의 신뢰를 악용하는 유형의 공격



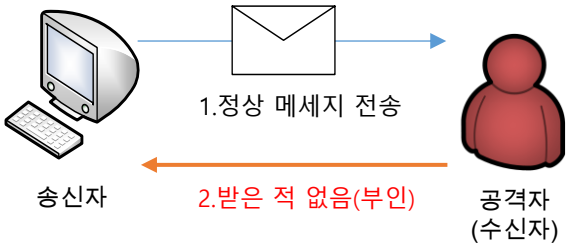
보안 개요

- 무결성을 위협하는 공격 (3/4)
  - 재전송(Replay)
    - 공격자가 정상적인 데이터를 가로채어 이후에 다시 전송하는 공격
      1. 송신자가 정상 데이터를 전송
      2. 공격자가 데이터를 가로채어 저장
      3. 이후 동일한 데이터를 수신자에게 재전송
      4. 수신자는 정상 요청으로 인식하여 처리
         - 재전송 공격은 이전의 정상 데이터를 악용하는 공격

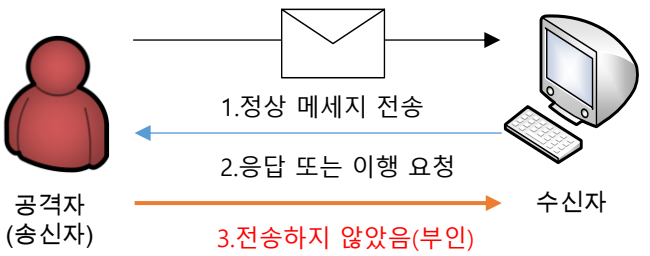


보안 개요

- 무결성을 위협하는 공격 (4/4)
  - 부인(Repudiation)
    - 사용자가 자신이 수행한 행위를 부인하는 공격
      1. 사용자가 요청 수행
      2. 시스템에 기록 또는 전송
      3. 이후 사용자가 행위를 부인
      4. 책임 추적 불가 또는 어려움 발생
         - 시스템의 신뢰성 저하



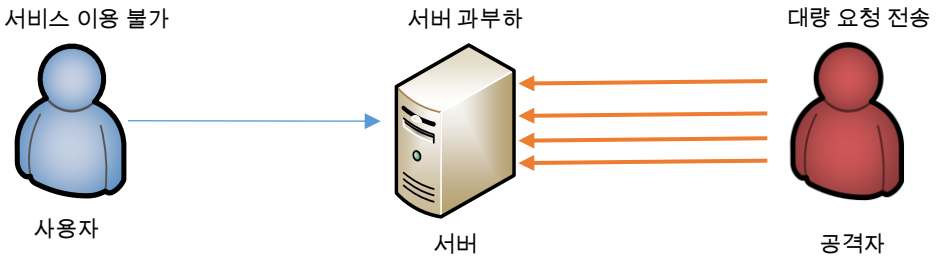
수신자 부인:메시지를 수신했음에도 불구하고 수신 사실을 부인



송신자 부인:메시지를 보냈음에도 불구하고 전송 사실을 부인

보안 개요

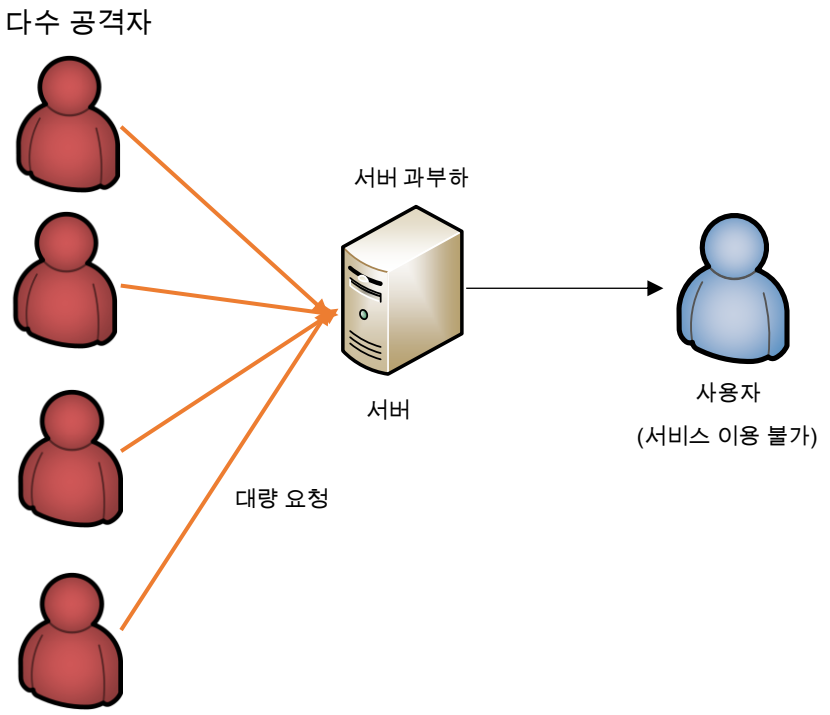
- 가용성을 위협하는 공격 (1/7)
  - 서비스 거부(DoS, Denial of Service)
    - 정상 사용자가 서비스 이용을 불가하게 하는 공격
      - 시스템 자원을 과도하게 소모
      - 정상 요청 처리 불가
      - 서비스 지연 또는 중단 발생
      - 가용성을 저해
        1. 공격자가 대량의 요청을 전송
        2. 서버의 자원 과부하 상태
        3. 정상 사용자의 요청 처리 불가
        4. 서비스 이용 불가



서비스 거부 공격은 정상 사용자의 접근을 차단하는 것을 주된 목적으로 함

보안 개요

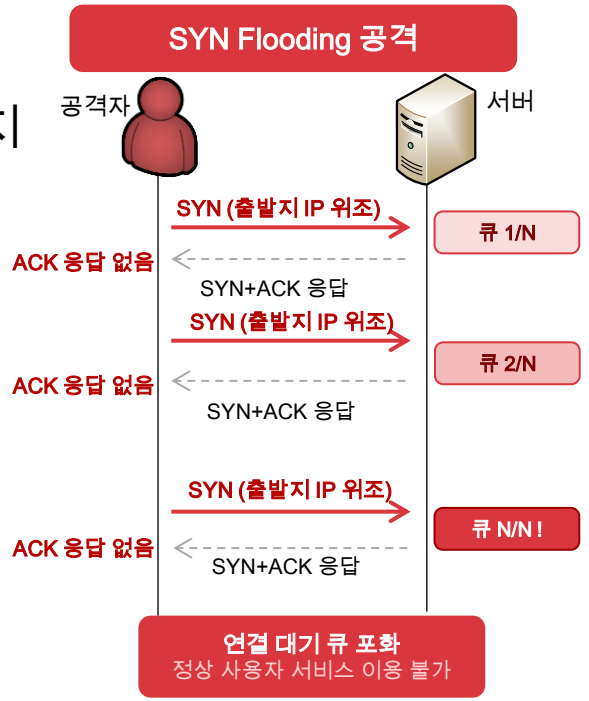
- 가용성을 위협하는 공격 (2/7)
  - 분산 서비스 거부(DDoS, Distributed Denial of Service)
    - 여러 공격자가 동시에 대량의 요청을 보내 서비스 이용을 방해하는 공격
      - 다수 공격 장치의 동시 요청 전송
      - 서버 자원 급격히 소모
      - 정상 요청 처리 불가
      - 서비스 지연 또는 중단
      - DoS 대비 대규모 공격
        - 서버 자원 과부하
        - 정상 요청이 처리 불가
        - 서비스 이용 불가



보안 개요

- 가용성을 위협하는 공격 (3/7)
  - 서비스 거부(DoS, Denial of service) 종류 (1/5)
    - SYN Flooding (트래픽 기반 공격)
      - SYN Flooding은 공격자가 ACK를 보내지 않아 서버의 연결 대기 큐를 고갈시키는 공격
        - TCP 3-way handshake 취약점 기반 공격
        - SYN 패킷 대량 전송
        - ACK 미전송으로 연결이 완료되지 않은 상태 유지
        - 연결 대기 큐 포화 → 서비스 이용 불가

*ACK
수신 확인을 의미하는 TCP 제어 신호



SYN → 서버 → SYN+ACK 응답 → (ACK 응답 없음)

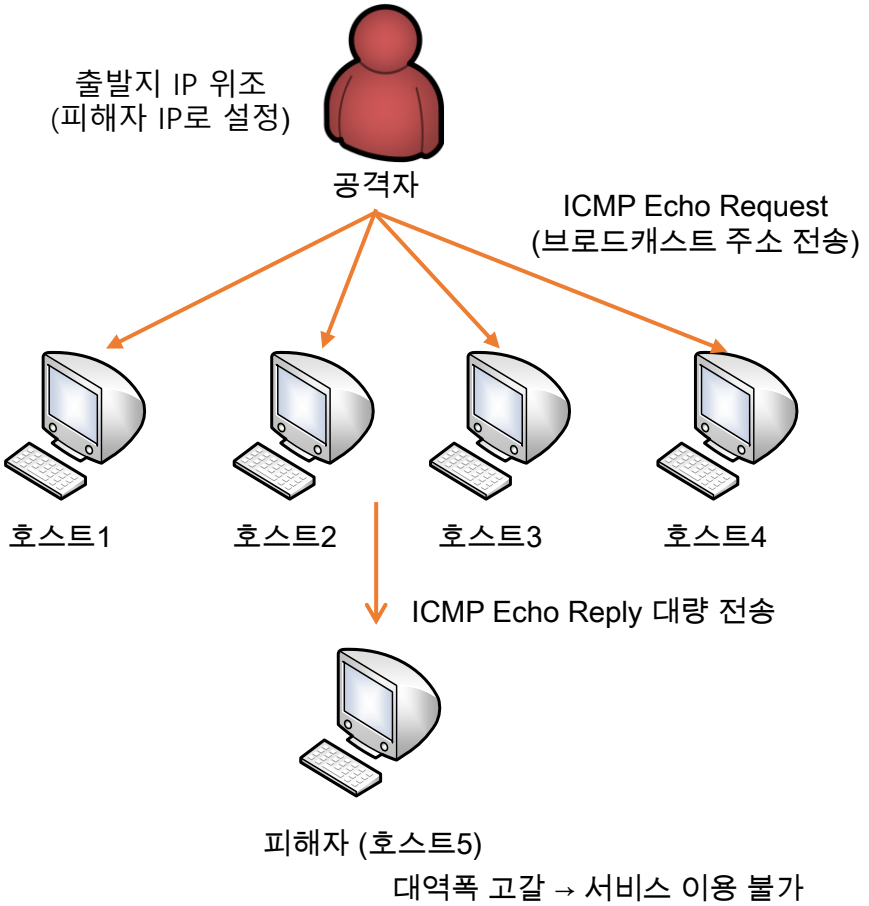
보안 개요

- 가용성을 위협하는 공격 (4/7)
  - 서비스 거부(DoS, Denial of service)종류 (2/5)
    - Smurf Attack (프로토콜 취약점 공격) (1/2)
      - ICMP Echo Request를 이용한 반사 및 증폭 공격
        1. 출발지 IP를 피해자 IP로 위조하여 요청 전송
        2. 브로드캐스트 주소로 ICMP 요청 전송
        3. 다수 호스트가 피해자에게 ICMP Echo Reply 전송
        4. 응답 트래픽 급증 → 네트워크 및 시스템 과부하 유발
      - 특징
        - ICMP 프로토콜 기반 공격
        - 출발지 IP 위조 사용
        - 브로드캐스트 트래픽 이용
        - 트래픽 증폭 효과 발생
        - 대역폭 고갈 및 서비스 이용 불가 초래

*ICMP
네트워크 상태 확인 및 오류 메시지 전달을 위한 제어 프로토콜
*브로드캐스트
네트워크 내 모든 장치에게 동시에 전송

보안 개요

- 가용성을 위협하는 공격 (5/7)
  - 서비스 거부(DoS, Denial of service)종류 (3/5)
    - Smurf Attack (프로토콜 취약점 공격) (2/2)

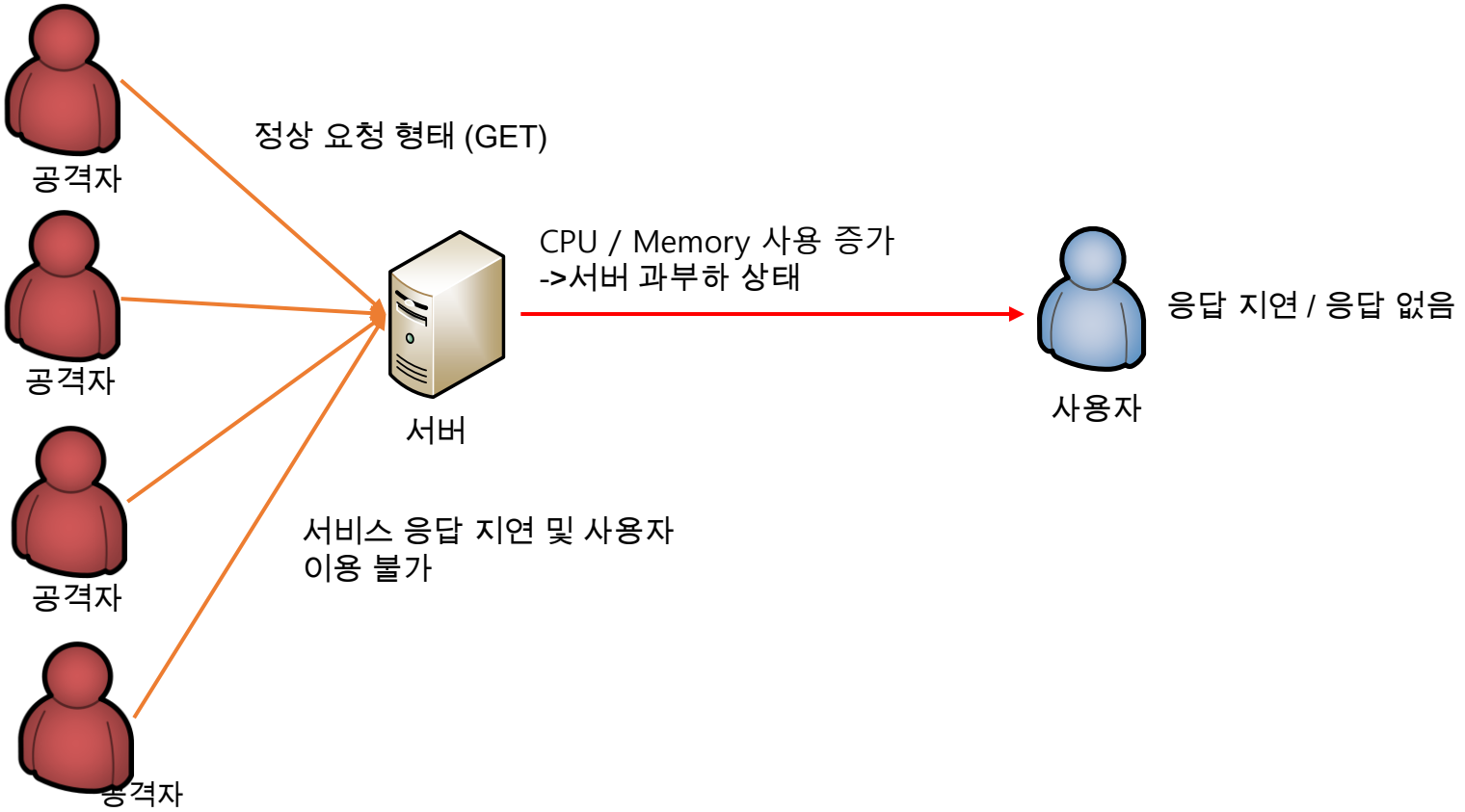


보안 개요

- 가용성을 위협하는 공격 (6/7)
  - 서비스 거부(DoS, Denial of service)종류 (4/5)
    - HTTP Flood (애플리케이션 계층 공격) (1/2)
      - 정상적인 HTTP 요청을 대량으로 전송하여 서버 자원을 고갈시키는 애플리케이션 계층 DoS 공격
        1. 공격자가 웹 서버에 HTTP 요청(GET/POST) 전송
        2. 요청은 정상 사용자 요청처럼 보임
        3. 서버는 요청 처리를 위해 CPU, 메모리, 애플리케이션 자원을 사용
        4. 대량 요청으로 인해 서버 자원 고갈 → 응답 지연 및 서비스 불가
      - 특징
        - HTTP 요청 기반 애플리케이션 계층 공격
        - 정상 요청 위장 트래픽 사용
        - 서버 자원 과다 소모 유발
        - 서비스 응답 지연 및 장애 발생

보안 개요

- 가용성을 위협하는 공격 (7/7)
  - 서비스 거부(DoS, Denial of service)종류 (5/5)
    - HTTP Flood (애플리케이션 계층 공격) (2/2)



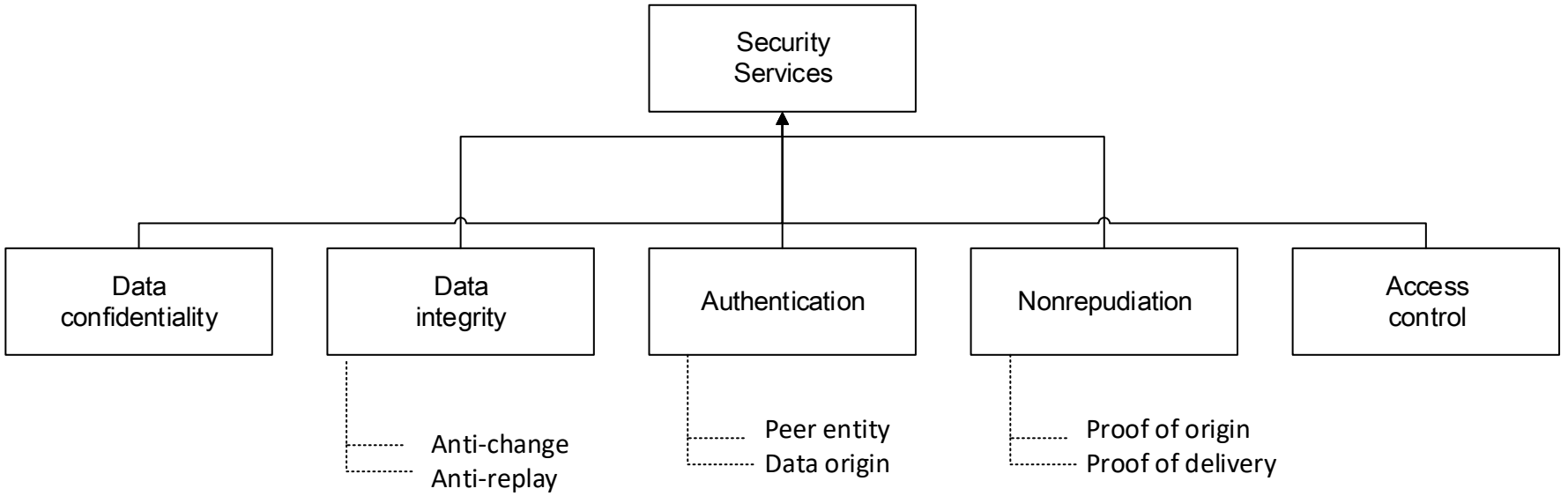
보안 개요

- 공격의 분류
  - 소극적 공격(Passive Attack)
    - 시스템이나 데이터에 영향 없음
    - 정보 비인가 수집
    - 데이터 변경 없음
    - 탐지 어려움
    - 정보 유출 목적
  - 적극적 공격 (Active Attack)
    - 시스템 또는 데이터 변경
    - 서비스에 영향 발생
    - 데이터 변경 발생
    - 탐지 가능
    - 시스템에 직접적인 영향

구분	소극적 공격	적극적 공격
목적	정보 수집	시스템 영향
데이터 변화	없음	있음
탐지 여부	어려움	비교적 쉬움
예시	스누핑, 트래픽 분석	변경, 가장, DoS

보안 개요

- 보안 서비스(Security Service) (1/2)
  - 보안 서비스는 다양한 공격으로부터 시스템과 데이터를 보호하기 위한 기능
    - 정보 시스템을 보호하기 위해 제공되는 기능
    - 다양한 공격으로부터 데이터와 시스템을 보호
    - 기밀성, 무결성, 가용성 등을 보장



보안 서비스

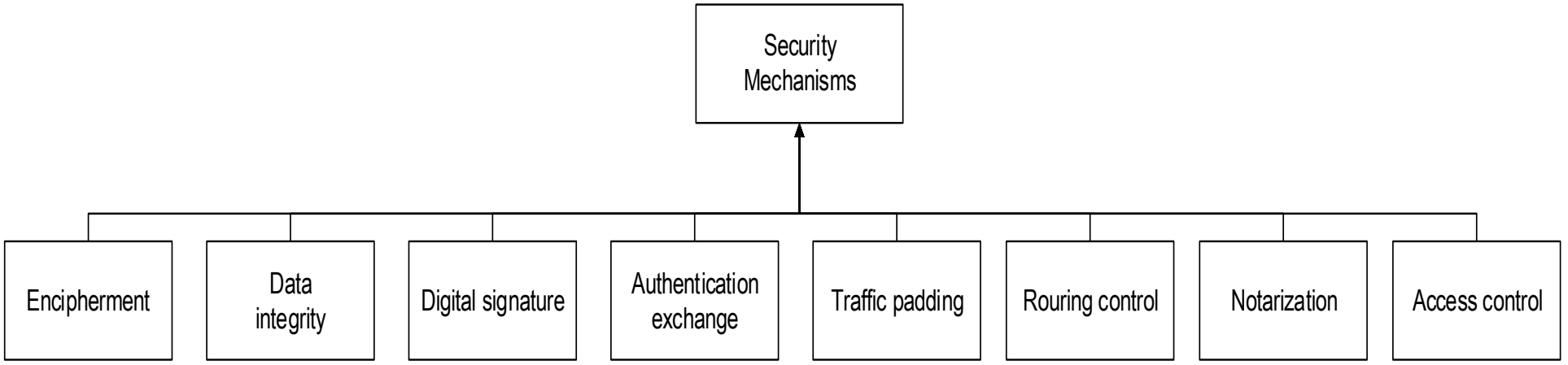
보안 개요

- 보안 서비스(Security Service) (2/2)

보안 서비스	설명
데이터 기밀성 (Data Confidentiality)	허가된 사용자만 정보에 접근할 수 있도록 보호 데이터 유출 방지
데이터 무결성 (Data Integrity)	데이터가 변경되지 않았음을 보장 데이터 위변조 방지
인증 (Authentication)	사용자의 신원을 확인하는 과정 신뢰할 수 있는 사용자만 접근 허용
부인봉쇄 (Non-repudiation)	수행한 행위를 부인하지 못하도록 보장 행위에 대한 책임 추적 가능
접근 제어 (Access Control)	비인가된 접근으로부터 데이터를 보호

보안 개요

- 보안 메커니즘 (Security Mechanisms) (1/2)
  - 보안 메커니즘은 보안 서비스를 구현하기 위한 기술적 방법
    - 시스템과 데이터를 보호하기 위한 구체적인 수단
    - 다양한 공격을 방어하기 위해 사용



보안 메커니즘

보안 개요

- 보안 메커니즘 (Security Mechanisms) (2/2)

보안 메커니즘	방법	예시
암호화 (Encipherment)	데이터를 암호화하여 내용을 보호	HTTPS 통신
데이터 무결성 (Data Integrity)	데이터가 변경되지 않았음을 보장	파일 다운로드 시 SHA-256 해시 확인
디지털 서명 (Digital Signature)	데이터의 출처와 무결성 보장 부인 방지 기능 제공	공인인증서
인증 교환 (Authentication Exchange)	통신 당사자의 신원을 확인 신뢰할 수 있는 사용자 검증	OTP
트래픽 패딩 (Traffic Padding)	의미 없는 데이터를 추가하여 트래픽 분석 방지	VPN에서 일정한 트래픽 유지
라우팅 제어 (Routing Control)	안전한 경로로 데이터 전송 공격 위험 경로 회피	군/금융망에서 특정 전용망 사용
공증 (Notarization)	제3자를 통해 통신 사실을 증명 분쟁 시 증거 제공	전자계약 타임스탬프 서비스
접근 제어 (Access Control)	허가된 사용자만 접근 가능하도록 제한 비인가 접근 차단	관리자 페이지 로그인 제한

보안 개요

- 보안 서비스와 보안 메커니즘 관계
  - 보안 서비스는 무엇을 보호할 것인지 정의
  - 보안 메커니즘은 이를 어떻게 보호할 것인지 구현하는 방법

보안 서비스	보안 메커니즘
데이터 기밀성(Data Confidentiality)	암호화,라우팅 제어
데이터 무결성(Data Integrity)	암호화, 디지털 서명, 데이터 무결성
인증(Authentication)	암호화, 디지털 서명, 인증 교환
부인봉쇄(Non-repudiation)	디지털 서명, 데이터 무결성, 공증
접근 제어(Access Control)	접근 제어

보안 개요

- 암호(Cryptography) (1/4)
  - 정의
    - 정보 보호를 위한 데이터 변환 기술
      - 기밀성, 무결성, 인증 서비스 제공
      - 데이터를 안전하게 전달하기 위한 핵심 기술
  - 메커니즘
    - 대칭-키 암호화(Symmetric-key Encipherment)
    - 비대칭-키 암호화(Asymmetric-key Encipherment)
    - 해싱(Hashing)

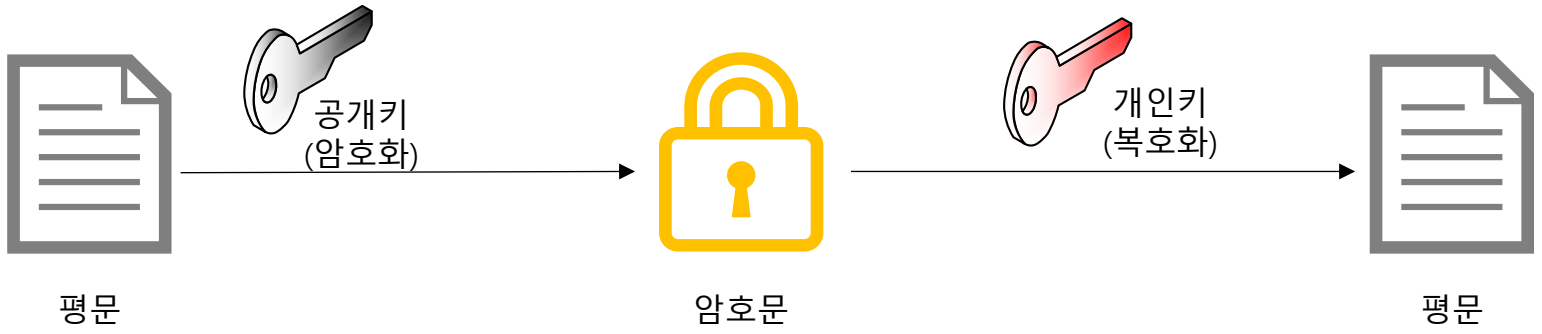
보안 개요

- 암호(Cryptography) (2/4)
  - 대칭-키 암호화(Symmetric-key Encipherment)
    - 동일한 키를 사용하여 데이터를 암호화하고 복호화하는 방식
      - 송신자와 수신자가 같은 비밀키를 공유
      - 암호화와 복호화에 동일한 키 사용
      - 연산 속도가 빠르고 효율적

암호화 키와 복호화 키가 서로 같음

보안 개요

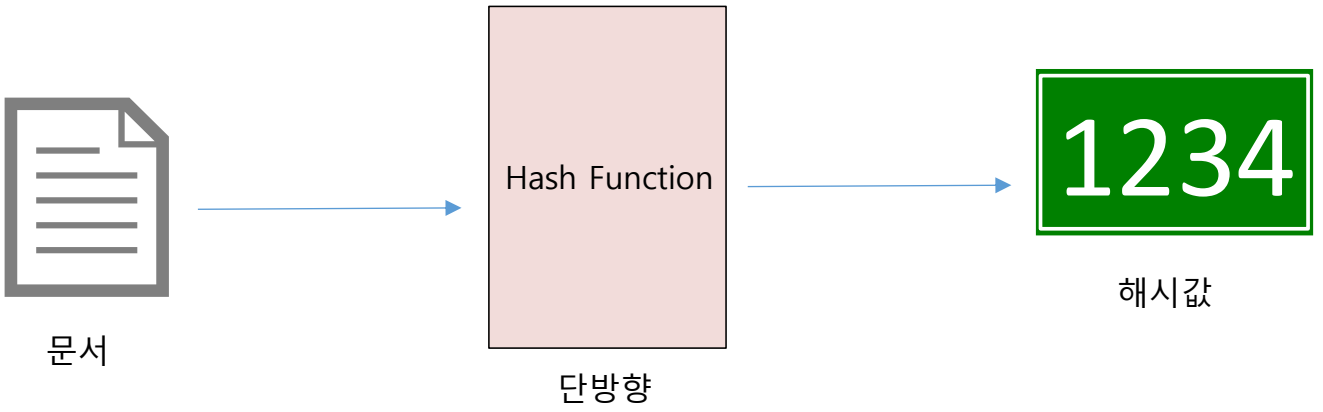
- 암호(Cryptography) (3/4)
  - 비대칭-키 암호화(Asymmetric-key Encipherment)
    - 서로 다른 두 개의 키(공개키, 개인키)를 사용하여 암호화와 복호화를 수행하는 방식
      - 공개키 기반 암호화
      - 개인키 기반 복호화
      - 키 사전 공유 불필요



암호화 키와 복호화 키가 서로 다름

보안 개요

- 암호(Cryptography) (4/4)
  - 해싱(Hashing)
    - 데이터를 고정된 길이의 값으로 변환하는 함수
      - 임의 길이의 데이터를 일정한 길이의 해시값으로 변환
      - 동일한 입력에 대해 항상 동일한 출력
      - 입력 변화에 대한 출력 급격 변화



해싱은 복호화가 불가한 단방향 변환 함수

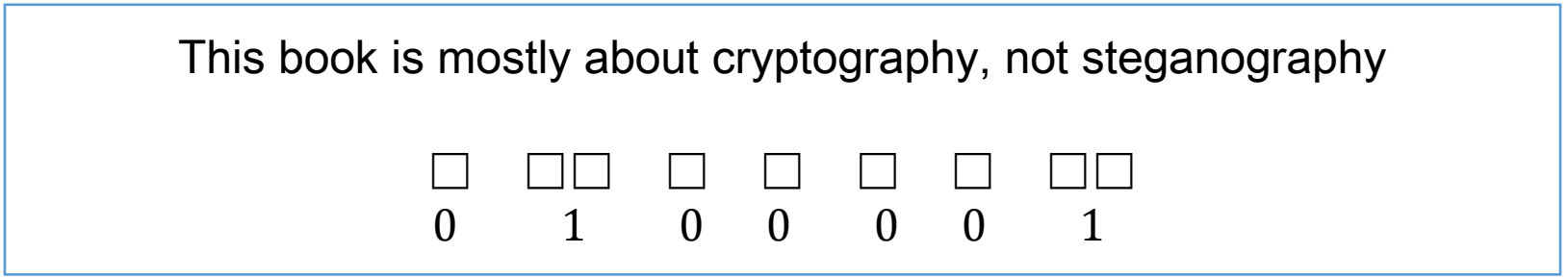
보안 개요

- 스테가노그래피(Steganography) (1/3)
  - 정보를 다른 데이터 안에 숨겨서 전달하는 기술
    - 메세지의 존재 자체를 숨김
    - 이미지, 오디오, 영상 등에 데이터를 삽입
    - 외부에서는 일반 데이터처럼 보임
  - 종류
    - 텍스트 커버(Text Cover)
    - 이미지 커버(Image Cover)
    - 다른 커버(Other Cover)

보안 개요

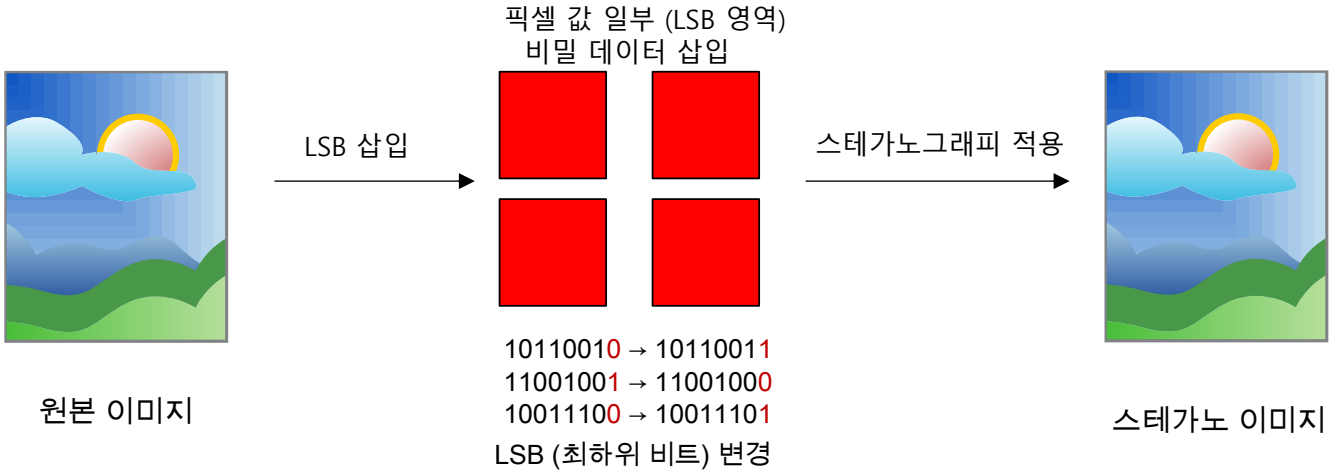
- 스테가노그래피(Steganography) (2/3)
  - 텍스트 커버(Text Cover)
    - 텍스트 문서 내에 비밀 메시지를 숨기는 방식
      - 공백, 글자 위치, 문장 구조 등을 이용해 정보 삽입
      - 겉으로 보기에는 일반 텍스트와 동일
      - 육안으로 식별하기 어려움
    - 텍스트 커버 특징
      - 구현이 용이함
      - 데이터 삽입 용량이 제한적
      - 미세한 변화에도 손상될 수 있음

문장 내 공백의 개수를 이용하여
0과 1의 비트 정보를 숨긴 예시



보안 개요

- 스테가노그래피(Steganography) (3/3)
  - 이미지 커버(Image Cover)
    - 정의
      - 이미지의 픽셀 값 기반 비밀 데이터를 숨기는 방식
        - 각 픽셀의 비트 값을 조작하여 정보 삽입
        - 주로 LSB(Least Significant Bit)를 사용
        - 육안 식별 어려움



LSB는 색상에 거의 영향을 주지 않아 육안으로 식별하기 어려움

목 차

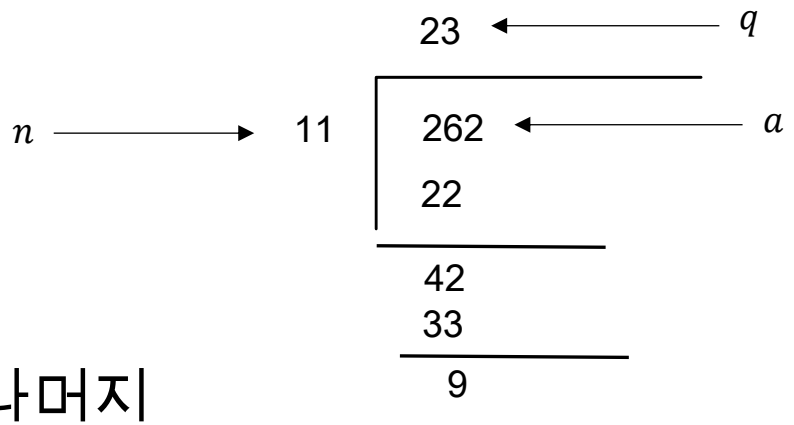
- 암호 개요
- 암호 수학

암호 수학

- 정수 연산(Integer Arithmetic) (1/10)
  - 정수 집합(Set of Integers)
    - 정수는 음의 정수, 0, 양의 정수를 포함하는 수의 집합
    - 보통 Z 로 표현함
  - 이진 연산(Binary Operation)
    - 집합의 두 원소를 입력으로 받아 하나의 결과를 만드는 연산
    - 정수 집합에서 대표적인 이항 연산
      - 덧셈, 뺄셈, 곱셈으로 이루어져 있음

암호 수학

- 정수 연산(Integer Arithmetic) (2/10)
  - 정수의 나눗셈(Division of Integers)
    - 하나의 정수를 다른 정수로 나누는 연산
    - 일반적으로 몫과 나머지로 표현
  - 나눗셈의 표현
    - 임의의 정수 $a, n(n \neq 0)$에 대해
    - $a = qn + r$
      - a는 피제수, q는 몫, n은 제수, r은 나머지
      - $n > 0$
      - $r \geq 0$



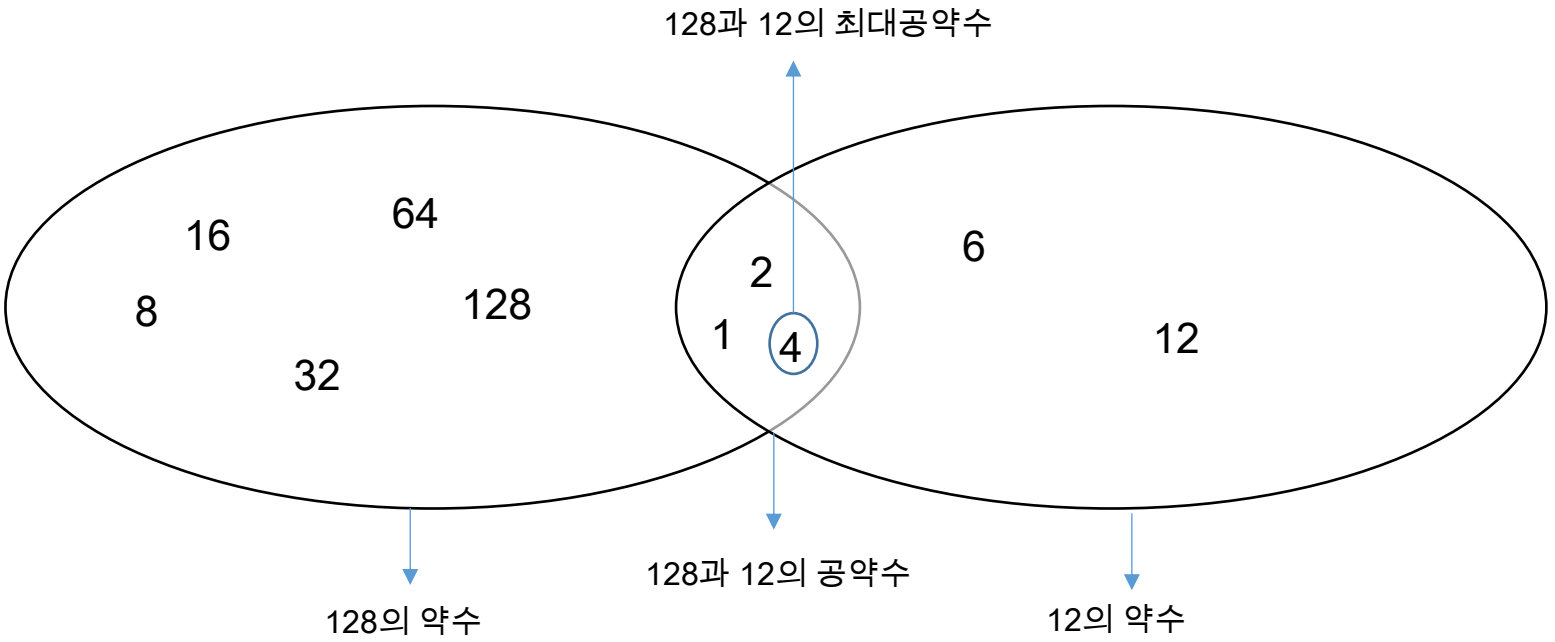
암호 수학

- 정수 연산(Integer Arithmetic) (3/10)
  - 가분성(Divisibility) (1/5)
    - 정의
      - 정수 a와 n에 대해 나머지 $r = 0$인 경우 가분성 성립
      - $a = q \times n$
        - 이때 n은 a의 약수
        - $n \mid a$로 표현
    - 성질
      - 만약 $a \mid 1$ 이면 $a = \pm 1$
      - 만약 $a \mid b$ 이고 $b \mid a$ 이면 $a = \pm b$
      - 만약 $a \mid b$ 이고 $b \mid c$ 이면 $a \mid c$
      - 만약 $a \mid b$ 이고 $a \mid c$ 이면 $a \mid (m \times b + n \times c)$
        - m과 n은 임의의 정수

암호 수학

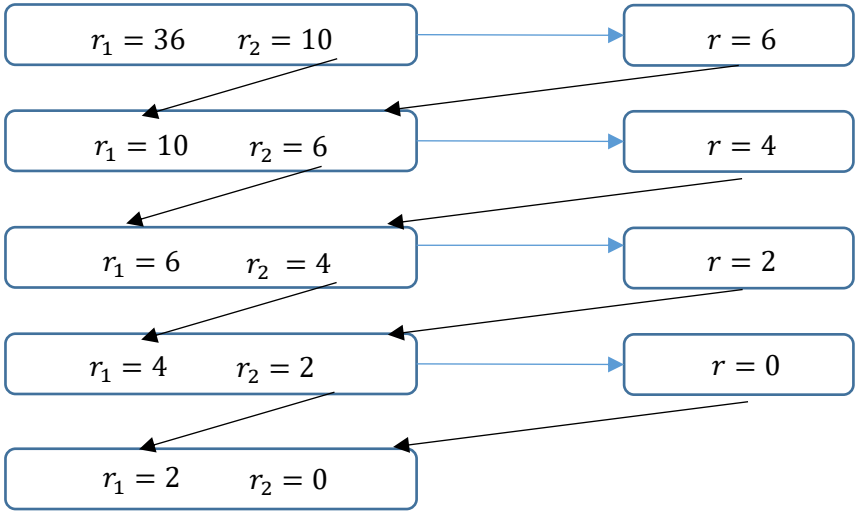
- 정수 연산(Integer Arithmetic) (4/10)
  - 가분성(Divisibility) (2/5)
    - 약수(Divisor)
      - 정수 a를 나누어 떨어지게 하는 정수
      - 정수 1은 하나의 약수, 1만을 가짐
      - 1을 제외한 모든 양의 정수는 최소 2개의 약수(1과 그 자신)를 가짐 (2개 이상도 가능)

- 정수 연산(Integer Arithmetic) (5/10)
  - 가분성(Divisibility) (3/5)
    - 최대 공약수(GCD, Greatest Common Divisor)
      - 두 정수를 동시에 나누는 공약수 중 가장 큰 값
      - 보통 $\gcd(a, b)$로 표현



암호 수학

- 정수 연산(Integer Arithmetic) (6/10)
  - 가분성(Divisibility) (4/5)
    - 유클리드 알고리즘(Euclidean Algorithm)
      - 두 정수의 최대공약수(GCD) 를 효율적으로 구하는 방법
      - 나눗셈의 나머지를 이용하여 반복적으로 계산함
        1. $\gcd(a, 0) = a$
        2. $\gcd(a, b) = \gcd(b, r)$, 이때 r은 a를 b로 나눈 나머지



1. 두 변수 r_1, r_2를 사용 (초기값: a, b)
2. $r = r_1 mod\, r_2$ 계산
3. $r_1 \leftarrow r_2, r_2 \leftarrow r$ 로 갱신
4. $r_2 = 0$이 될 때까지 반복
5. 종료 시 $\gcd(a, b) = r_1$

$\gcd(a, b) = 1$이면, a 와 b는 서로 소 라고 한다

암호 수학

- 정수 연산(Integer Arithmetic) (7/10)
  - 가분성(Divisibility) (5/5)
    - 확장 유클리드 알고리즘(Extended Euclidean Algorithm)
      - 최대공약수와 함께 정수 s, t를 구하는 방법
      - 다음 식을 만족하는 값을 계산
        - $s \times a + t \times b = \gcd(a, b)$

$< a = 161$이고 $b = 28$일 때 $\gcd(a, b)$와 s, t 구하는 방법>

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
3	21 7	0	1 −1	4	−5 6	−23
	7 0		−1 4		6 −23	

1. r, q 계산
2. $r_1 \leftarrow r_2, r_2 \leftarrow r$
3. s, t 함께 갱신
4. $r_2 = 0$까지 반복

$-> \gcd(a, b) = r_1$

$\gcd(161, 28) = 7, s = -1, t = 6$

검증:$(-1) \times 161 + 6 \times 28 = 7$

암호 수학

- 정수 연산(Integer Arithmetic) (8/10)
  - 선형 디오판투스 방정식(Linear Diophantine Equations) (1/3)
    - 정수 해를 갖는 1차 방정식
    - $ax + by = c$ (a, b, c는 정수)
    - 정수 x, y를 구하는 문제
      - $d = gcd(a, b)$일 때
      - $d \mid c$인 경우 -> 방정식의 해가 존재(무수히 많음)
      - $d \nmid c$인 경우 -> 방정식의 해가 존재하지 않음

암호 수학

- 정수 연산(Integer Arithmetic) (9/10)
  - 선형 디오판투스 방정식(Linear Diophantine Equations) (2/3)
    - 특수 해($d \mid c$ 인 경우)
      - d로 방정식의 양변을 나누어 $a_1x + b_1y = c_1$ 식으로 만듦
      - 확장 유클리드 알고리즘을 이용하여 식 $a_1s + b_1t = 1$을 만족하는 s와 t 값을 계산
      - 특수 해 식: $x_0 = (c/d)s, y_0 = (c/d)t$
    - 일반 해
      - 특수 해를 찾아 낸 뒤 계산
      - 일반 해 식: $x = x_0 + k(b/d), y = y_0 - k(b/d)$ (k는 임의의 정수)

암호 수학

- 정수 연산(Integer Arithmetic) (10/10)
  - 선형 디오판투스 방정식(Linear Diophantine Equations) (3/3)

$21x + 14y = 35$의 특수 해와 일반 해를 구하시오.

- $d = gcd(21, 14) = 7$
- $7|35$이므로 무한한 해를 가짐
- 양변을 7로 나누면 $3x + 2y = 5$
- 확장 유클리드 알고리즘을 이용
- $3s + 2t = 1$을 만족하는 s와 t값을 계산, $s = 1, t = -1$
- 특수해식: $x_0 = 5 \times 1 = 5, y_0 = 5 \times (-1) = -5$ ($35/7 = 5$이기때문)
- 일반 해 식: $x = 5 + k \times 2, y = -5 - k \times 3$ (k는 정수)
- 그러므로 해는 $(5, -5), (7, -8), (9, -11)$ 등이 됨

암호 수학

- 모듈로 연산(Modular Arithmetic) (1/9)
  - 정수를 어떤 수 n으로 나눈 나머지를 기준으로 연산하는 방법
  - 나눗셈 관계식
    - 임의의 정수 a와 양의 정수 n에 대해 $a = n \times q + r$
    - $r = a \, mod \, n$ 따라서 $a \equiv r \, (mod \, n)$
      - 나눗셈 관계식은 모듈로 연산의 기반
        - $17 = 12 \times 1 + 5 -> q = 1, r = 5$
          - $17 \equiv 5 \, (mod \, 12)$
  - 모듈로 연산자(Modulo Operator)
    - 2진 연산자를 모듈로 연산자(Modulo Operator)라고 하며 mod로 표기함
    - 입력 값은 모듈로(Modulus)라고 하고, 결과 값 r은 나머지라고함

암호 수학

- 모듈로 연산(Modular Arithmetic) (2/9)
  - 잉여류 Z_n(Residue Class)
    - 잉여류는 같은 나머지를 가지는 정수들의 집합
    - $Z_n = \{[0], [1], [2], \dots, [n-1]\}$
      - $Z_{10} = \{[0], [1], [2], [3], [4], [5], [6], [7], [8], [9]\}$

암호 수학

- 모듈로 연산(Modular Arithmetic) (3/9)
  - 합동(Congruence)
    - $a \equiv b \ (mod\ n)$
      $\rightarrow a$와 b를 n으로 나눈 나머지가 같음
  - 합동의 의미
    - 같은 나머지를 가지는 정수들의 관계
    - 하나의 합동식은 여러 정수를 포함
  - 등식 연산자와의 차이
    - 등식 연산자는 Z의 원소에서 Z의 원소로 대응 (일대일대응)
    - 합동 연산자는 Z의 원소에서 Z_n의 원소로 대응 (다대일대응)

암호 수학

- 모듈로 연산(Modular Arithmetic) (4/9)
  - Z_n에서의 연산
    - 세 개의 2진 연산(덧셈, 뺄셈, 곱셈)이 정의됨
  - 성질
    - 덧셈
      - $(a + b) mod\ n = [(a\ mod\ n) + (b\ mod\ n)] mod\ n$
        - $(2{,}345{,}678 + 3{,}456{,}789) mod\ 13 = (9 + 12) mod\ 13 = 8$
    - 뺄셈
      - $(a - b) mod\ n = [(a\ mod\ n) - (b\ mod\ n)] mod\ n$
        - $(2{,}345{,}678 - 3{,}456{,}789) mod\ 13 = (9 - 12) mod\ 13 = 10$
    - 곱셈
      - $(a \times b) mod\ n = [(a\ mod\ n) \times (b\ mod\ n)] mod\ n$
        - $(2{,}345{,}678 \times 3{,}456{,}789) mod\ 13 = (9 \times 12) mod\ 13 = 4$

암호 수학

- 모듈로 연산(Modular Arithmetic) (5/9)
  - 역원(Inverse)
    - 정수 a에 대해 $a \times x \equiv 1 \ (mod\ n)$을 만족하는 x를 a의 역원이라 함
    - 즉 $a^{-1} \equiv x \ (mod\ n)$
      - $\gcd(a, n) = 1$일 때만 역원이 존재
  - 덧셈에 대한 역원
    - $(a + b) \equiv 0\ mod\ n$
      - a, b는 덧셈에 대한 역원이 된다
    - z_n에서 a의 덧셈에 대한 역원은 $b = n - a$로 계산됨
    - 모듈로 연산에서 각각의 수는 유일한 한 개의 덧셈에 대한 역원을 가짐
      - 덧셈에 대한 역원이 자기 자신일 수도 있음
        - z_{10}에서 모든 덧셈에 대한 역원은 (0, 0), (1, 9), (2, 8), (3, 7), (4, 6), (5, 5)

암호 수학

- 모듈로 연산(Modular Arithmetic) (6/9)
  - 역원(Inverse)
    - 곱셈에 대한 역원
      - $a \times b \equiv 1 \; mod$ n
        - 모듈로가 10이면 3의 곱셈에 대한 역원은 7, $(3 \times 7) mod \; 10 = 1$
      - 모듈로 연산에서 정수는 곱셈에 대한 역원이 있을 수 도 있고 없을 수 도 있음
      - 정수 a가 Z_n에서 곱셈 역원을 가지기 위한 필요충분조건

$$\gcd(a, n) = 1$$

        - a와 n이 서로소일 때만 a의 역원 a^{-1}이 존재
        - $\gcd(a, n) \neq 1$이면 역원은 존재하지 않음
        - 역원이 존재한다면, 그 정수와 해당하는 곱셈에 대한 역원의 곱은 모듈로 n에서 1과 합동
          - z_{10}에서 곱셈에 대한 역원은 $(1,1), (3,7), (9,9)$가 있음

암호 수학

- 모듈로 연산(Modular Arithmetic) (7/9)
  - 덧셈표와 곱셈표(Addition and Multiplication Tables)
    - Z_n에서의 연산 결과를 표 형태로 나타낸 것
    - 덧셈표
      - 각 원소는 덧셈에 대한 역원을 가짐
      - 항상 결과가 Z_n 안에 존재
    - 곱셈표
      - 일부 원소만 곱셈 역원을 가짐
      - 곱셈 결과가 1이 되는 경우 → 역원 존재
    - 교환법칙 성립
      - $a + b = b + a$
      - $a \times b = b \times a$

암호 수학

- 모듈로 연산(Modular Arithmetic) (8/9)
  - 덧셈과 곱셈에 대한 다른 집합
    - Z_n^* 은 Z_n의 부분집합으로 Z_n에 속한 정수 중 곱셈에 대한 역원을 가지는 원소들의 집합
    - Z_n의 각 원소는 덧셈에 대한 역원을 가지므로 덧셈 연산은 Z_n 전체에서 정의됨
    - 모든 원소가 곱셈에 대한 역원을 가지지는 않음
      - 따라서 곱셈 연산을 위해다 Z_n^*을 사용
      - $Z_6 = \{0,1,2,3,4,5\}$ $\quad Z_6^* = \{1,5\}$
        $Z_7 = \{0,1,2,3,4,5,6\}$ $\quad Z_7^* = \{1,2,3,4,5,6\}$

암호 수학

- 모듈로 연산(Modular Arithmetic) (9/9)
  - 모듈로가 소수인 경우의 집합
    - Z_p는 소수 n을 제외한 Z_n과 같음
    - Z_p는 0부터 $p-1$까지의 모든 정수를 포함
    - Z_p의 각 원소는 덧셈에 대한 역원을 가지고, 0을 제외한 각각의 원소는 곱셈에 대한 역원을 가짐
    - Z_p^*는 소수 n을 제외한 Z_n^*와 같음
    - Z_p^*는 1부터 $p-1$까지의 모든 정수를 포함
    - Z_p^*의 각 원소는 덧셈과 곱셈에 대한 역원을 가짐

e.g., $Z_{13} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
$Z_{13}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

암호 수학

- 행렬(Matrices) (1/10)
  - 행렬은 수들을 직사각형 형태로 배열한 것
  - $l \times m$ 행렬
    - l: 행의 개수
    - m: 열의 개수
  - 각 원소 a_{ij}는 i번째 행, j번째 열의 값

e.g., 행렬 A: $\begin{bmatrix} 23 & 14 & 56 \\ 12 & 21 & 18 \\ 10 & 8 & 41 \end{bmatrix}$

암호 수학

- 행렬(Matrices) (2/10)
  - 행 행렬(Row Matrix)
    - 행렬이 하나의 행 $(l = 1)$만 갖는 경우
  - 열 행렬(Column Matrix)
    - 행렬이 하나의 열 $(m = 1)$만 갖는 경우
  - 정방 행렬(Square Matrix)
    - 행의 개수와 열의 개수가 같은 행렬$(l = m)$
  - 항등 행렬(Identity Matrix)
    - 주대각선이 1이고 나머지는 0인 정방행렬

e.g., $\begin{bmatrix} 2 & 1 & 5 \end{bmatrix}$ $\begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix}$ $\begin{bmatrix} 23 & 14 \\ 8 & 10 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

행 행렬 **열 행렬** **정방 행렬** **항등 행렬**

암호 수학

- 행렬(Matrices) (3/10)
  - 행렬의 연산과 관계식
    - 등식
      - 두 행렬의 행과 열의 개수와 대응되는 원소가 동일하면 그 두 행렬은 동일
      - 모든 i와 j에 대해 $a_{ij} = b_{ij}$ 이면 $A = B$
    - 덧셈과 뺄셈
      - 두 행렬의 행과 열의 개수가 같으면 두 행렬을 더할 수 있음
      - 덧셈은 $C = A + B$, 뺄셈은 $D = A - B$

e.g., $\begin{bmatrix} 14 & 5 & 7 \\ 2 & 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 2 & 4 \\ 1 & 3 & 2 \end{bmatrix} + \begin{bmatrix} 8 & 3 & 3 \\ 1 & 0 & 6 \end{bmatrix}$ $\quad C = A + B$

$\begin{bmatrix} -3 & 1 & -5 \\ 5 & 0 & -4 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 3 \\ 9 & 7 & 1 \end{bmatrix} - \begin{bmatrix} 7 & 1 & 8 \\ 4 & 7 & 5 \end{bmatrix}$ $\quad D = A - B$

암호 수학

- 행렬(Matrices) (4/10)
  - 행렬의 연산과 관계식
    - 곱셈
      - 첫 번째 행렬의 열의 개수가 두 번째 행렬의 행의 개수와 같다면 다른 크기의 두 행렬을 곱할 수 있음
      - A가 $l \times m$ 행렬이고 B가 $m \times p$ 행렬의 경우 C 곱의 크기는 $l \times p$가 됨
      - 행렬 A의 원소를a_{ij}, 행렬 B의 원소를 b_{jk}라 했을 때 행렬 C의 각 원소c_{ik}계산법
        - $c_{ik} = \sum a_{ij} \times b_{jk} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + \cdots + a_{im} \times b_{mj}$

e.g., $\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix} \times \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} (1 \times 2 + 3 \times 1) & (1 \times 4 + 3 \times 3) \\ (5 \times 2 + 7 \times 1) & (5 \times 4 + 7 \times 3) \end{bmatrix} = \begin{bmatrix} 5 & 13 \\ 17 & 41 \end{bmatrix}$

암호 수학

- 행렬(Matrices) (5/10)
  - 행렬의 연산과 관계식
    - 스칼라 곱
      - A가 $l \times m$행렬이고, x가 스칼라일 때
      - $C = xA$는 $c_{ij} = x \times a_{ij}$ 인 크기가 $l \times m$인 행렬이 됨

e.g., $\begin{bmatrix} 8 & 6 & 4 \\ 12 & 6 & 10 \end{bmatrix} = 2 \times \begin{bmatrix} 4 & 3 & 2 \\ 6 & 3 & 5 \end{bmatrix}$

암호 수학

- 행렬(Matrices) (6/10)
  - 행렬식
    - 정방행렬에 대해 정의됨
    - 크기가$m \times m$인 정방 행렬 A의 행렬식은 det(A)로 표기
    - m=1이면 $\det(A) = a_{11}$
    - m>1이면 $\det(A) = \sum_{j=1,\ldots.m}(-1)^{i+j}\, a_{ij} \times \det(A_{ij})$
    - A_{ij}는 A에서 i행과 j열을 제거한 부분행렬

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

$= 2 \times 3 - 5 \times 1 = 6 - 5 = 1$

암호 수학

- 행렬(Matrices) (7/10)
  - 역행렬
    - 행렬은 덧셈과 곱셈에 대한 역행렬을 모두 가짐
    - 덧셈에 대한 역행렬(Additive Inverse Matrices)
      - 행렬 A의 덧셈에 역행렬은 $A + B = 0$을 만족하는 행렬 B
      - 모든 i와 j에 대해 $b_{ij} = -a_{ij}$ 이고 A의 덧셈에 대한 역행렬은 $-A$로 정의됨

e.g., $A = \begin{bmatrix} 2 & 5 \\ 4 & 3 \end{bmatrix}, B = \begin{bmatrix} -2 & -5 \\ -4 & -3 \end{bmatrix}$

$$B = -A$$

암호 수학

- 행렬(Matrices) (8/10)
  - 역행렬
    - 곱셈에 대한 역행렬(Multiplicative Inverse Matrix)
    - 행렬 A의 덧셈에 역행렬은 $A + B = 0$을 만족하는 행렬
    - 모든 i와 j에 대해 $b_{ij} = -a_{ij}$ 이고 A의 덧셈에 대한 역행렬은 $-A$로 정의됨

e.g., $\mathrm{A} = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 0 & -1 \\ 0 & 2 & 1 \end{bmatrix}$의 역행렬 $\mathrm{A}^{-1} = \frac{1}{7}\begin{bmatrix} 2 & 3 & 1 \\ -1 & 2 & 3 \\ 2 & -4 & 1 \end{bmatrix}$

암호 수학

- 행렬(Matrices) (9/10)
  - 잉여행렬 (Residue Matrices)
    - 모든 원소가 Z_n에 속하는 행렬을 잉여 행렬이라 함
    - 행렬의 덧셈과 곱셈은 $mod\ n$ 연산으로 수행됨
    - 일반 행렬 연산과 동일하지만,모든 결과는 $mod\ n$으로 계산됨
    - 곱셈에 대한 역행렬 존재 조건
      - $\gcd(\det(A), n) = 1$
        - $\det(A)$가 Z_n 에서 역원을 가져야 함
    - 위 조건을 만족하면 A는 역행렬 A^{-1} 을 가짐

$$A = \begin{bmatrix} 3 & 5 \\ 2 & 7 \end{bmatrix} \qquad \det(A) = (3 \times 7 - 5 \times 2) = 21 - 10 = 11$$

$$\gcd(11, 26) = 1 \rightarrow \text{역행렬 존재}$$

$$11^{-1} = 19\ (mod 26) \rightarrow A^{-1} = 19 \times \begin{bmatrix} 7 & -5 \\ -2 & 3 \end{bmatrix} (mod\ 26) = \begin{bmatrix} 3 & 9 \\ 14 & 5 \end{bmatrix}$$

암호 수학

- 행렬(Matrices) (10/10)
  - 잉여행렬 (Residue Matrices)
    - 합동
      - 두 행렬 A, B가 같은 크기를 가지고, 모든 원소가 $mod\ n$에서 합동이면 두 행렬 A, B가 $mod\ n$에서 합동이라 함
      - $a \equiv b\ (mod\ n)\ -> a$와 b는 $mod\ n$에서 같은 값을 가짐
        - e.g., $17\ \equiv\ 5\ (mod\ 12)$, $14 = 2(mod\ 12)$
          - 모듈로 12에서 합동
        - $A \equiv B\ (mod\ n)$
          $\Leftrightarrow$모든 i,j에 대해 $a_{ij} \equiv\ b_{ij}\ (mod\ n)$

암호 수학

- 선형 합동(Linear Congruences) (1/3)
  - 일변수 선형 방정식(Single-variable linear equations)
    - $ax \equiv b \ (mod\ n)$의 형태를 갖는 식
      - 여기서 a, b, n은 정수이고, x는 미지수
    - $\gcd(a,n) = d$라고 가정
      - $d \nmid b$ 이면 해가 존재하지 않음
      - $d \mid b$이면 d개의 해가 존재

<해를 구하는 방법>

1. 모듈로를 포함하여 방정식의 양변을 d로 나눔
2. 특수해 $x0$를 구하기 위해 약분한 방정식의 양변에 a의 곱셈에 대한 역원을 곱함
3. 일반해는 $k = 0,1,\cdots,(d-1)$에 대하여 $x = x0 + k(\ n/d)$

암호 수학

- 선형 합동(Linear Congruences) (2/3)
  - 일변수 선형 방정식(Single-variable linear equations)
    - 방정식 $10x \equiv 2(mod\ 15)$의 해를 구하시오
      - $\gcd(10,15) = 5$이고 5가 2를 나누지 않기 때문에 해가 존재하지 않음
    - 방정식 $14x \equiv 12(mod\ 18)$의 해를 구하시오
      - $\gcd(14,18) = 2$이고 $2 \mid 12$ 이므로 두 개의 근이 존재함

$$14x \equiv 12(mod\ 18) \rightarrow 7x \equiv 6(mod\ 9) \rightarrow x \equiv 6(7^{-1})(mod\ 9)$$

$$x_0 = (6 \times 7^{-1})\ mod\ 9 = (6 \times 4)\ (mod\ 9) = 6$$

$$x1 = x0 + 1 \times (18/2) = 15$$

암호 수학

- 선형 합동(Linear Congruences) (3/3)
  - 일차 연립 방정식
    - 동일한 모듈로 n에 대해 여러 개의 선형 합동식으로 이루어진 방정식
    - 계수 행렬 A의 역행렬이 존재하면 해를 구할 수 있음

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\equiv b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\equiv b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &\equiv b_n \end{aligned}$$

A. Equations

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

B. Interpretation

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

C. Solution

Thanks!

김 민 식(ms6127@naver.com)