2025/07/11, 2025 보안 기초 세미나

암호학과 네트워크 보안

-1장 보안개요, 2장 암호수학-

민 윤 홍(muh4316@gmail.com)

세종대학교 프로토콜공학연구실

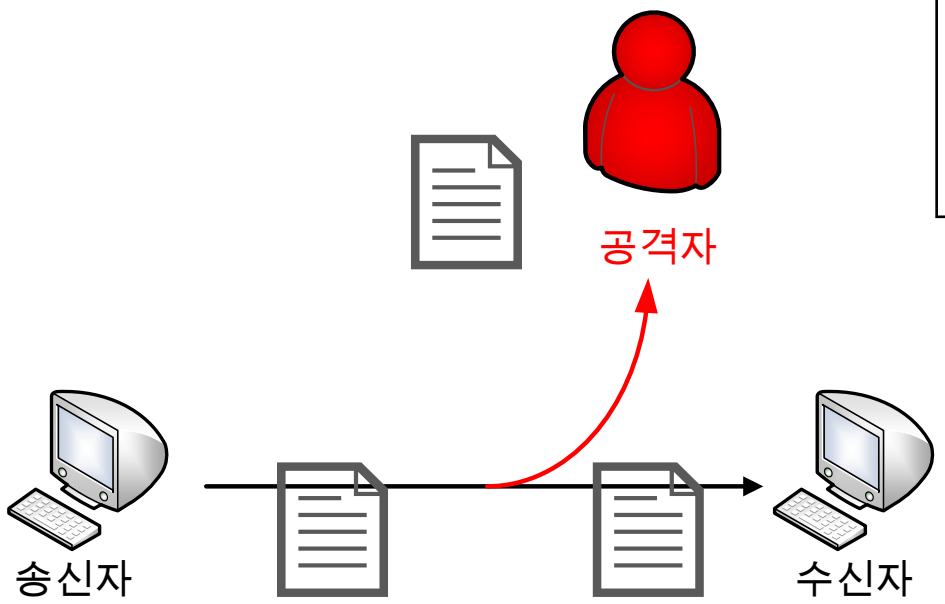
목 차

보 충

- 공격(Attack)
  - 정보 및 자원을 위협하는 행위
  - 보안 목표와 관련된 공격의 종류
    - 기밀성을 위협하는 공격
      - e.g., 스누핑(Snoofing), 트래픽 분석(Traffic Analysis)
    - 무결성을 위협하는 공격
      - e.g., 변조(Modification), 가장(Masquerading), 재전송(Replay) 부인(Repudiation)
    - 가용성을 위협하는 공격
      - e.g., 서비스 거부 공격(DoS, Denial of Service)

보 충

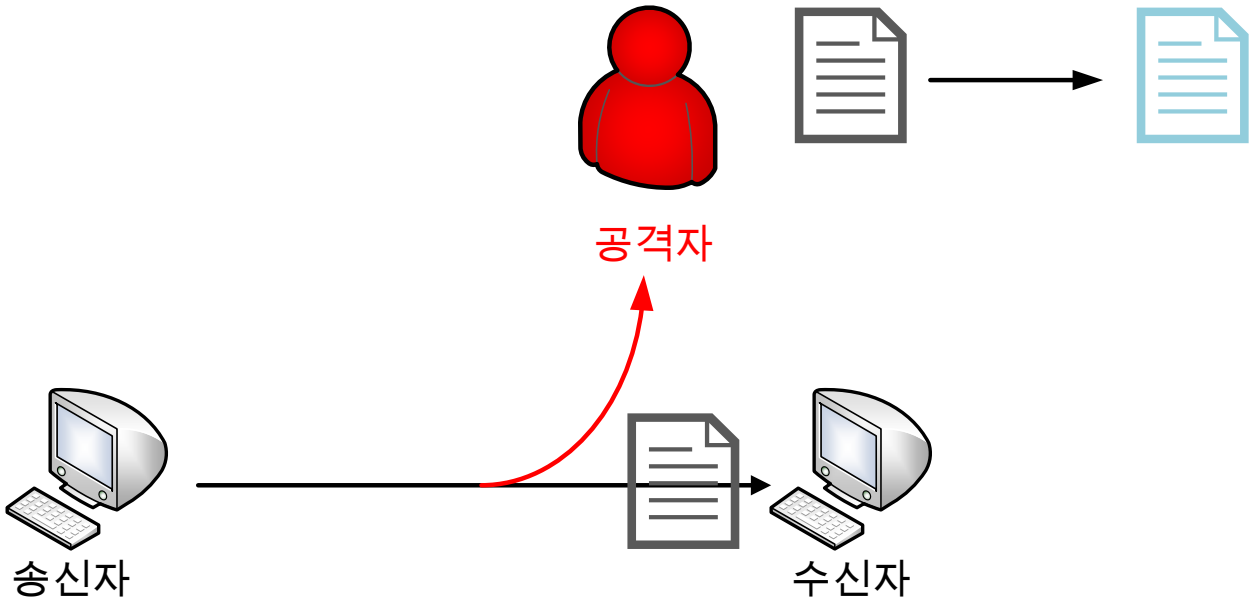
- 공격(Attack)
  - 기밀성을 위협하는 공격(1/2)
    - 스누핑(Snooping)
      - 공격자가 전송되는 정보를 몰래 가로채고 획득하는 행위
        - e.g., 공격자가 수신자의 은행 파일을 캡처하여 로그인하는 행위



*스니핑(Sniffing)
데이터를 몰래 훔쳐보는 행위
*스푸핑(Spoofing)
공격자가 다른 IP주소, 웹페이지 등으로 위변조를 통해 대상 시스템을 속이는 행위

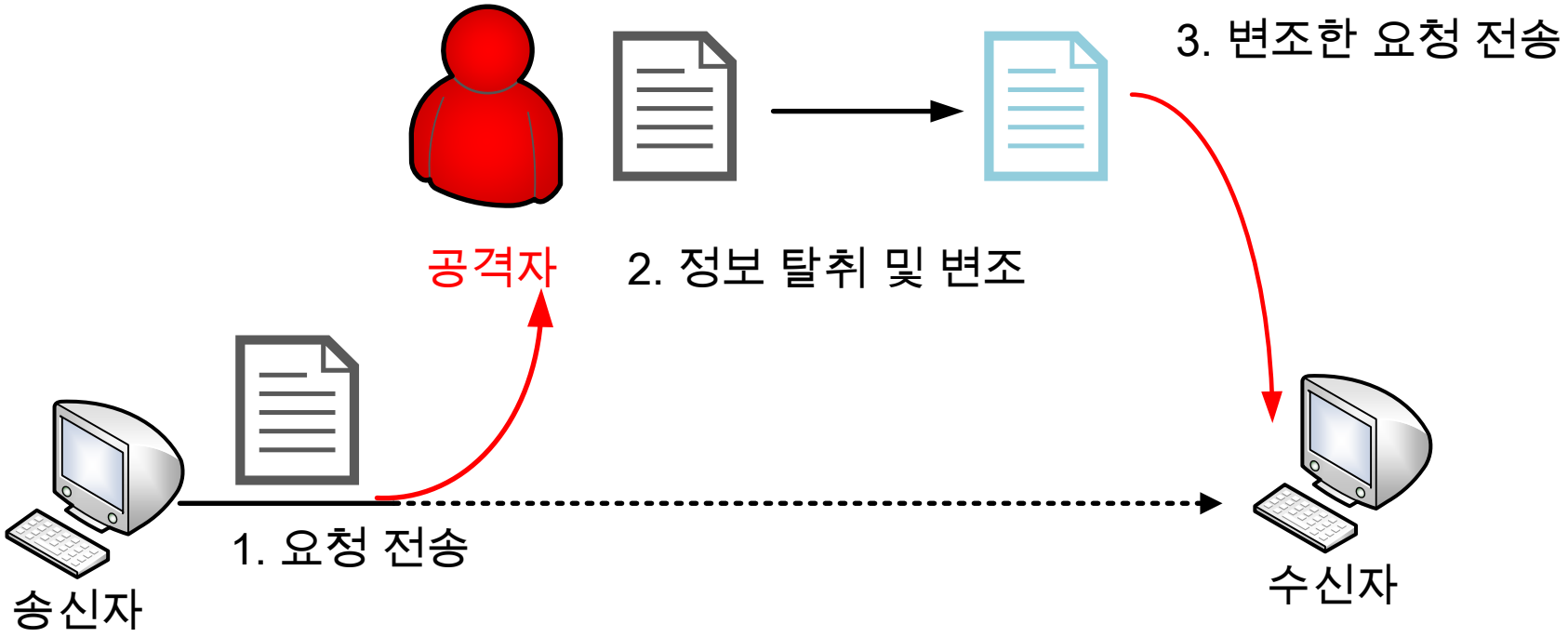
보 충

- 공격(Attack)
  - 기밀성을 위협하는 공격(2/2)
    - 트래픽 분석(Traffic Analysis)
      - 공격자가 암호화된 정보를 이해할 수 없더라도 트래픽을 분석함으로써 다른 형태의 정보를 얻는 행위
        - e.g., 적국이 특정 시간에 반복해서 데이터 패킷 수가 급증하는 것을 보고 통신이 시작되는 시간을 알아낼 수 있는 행위



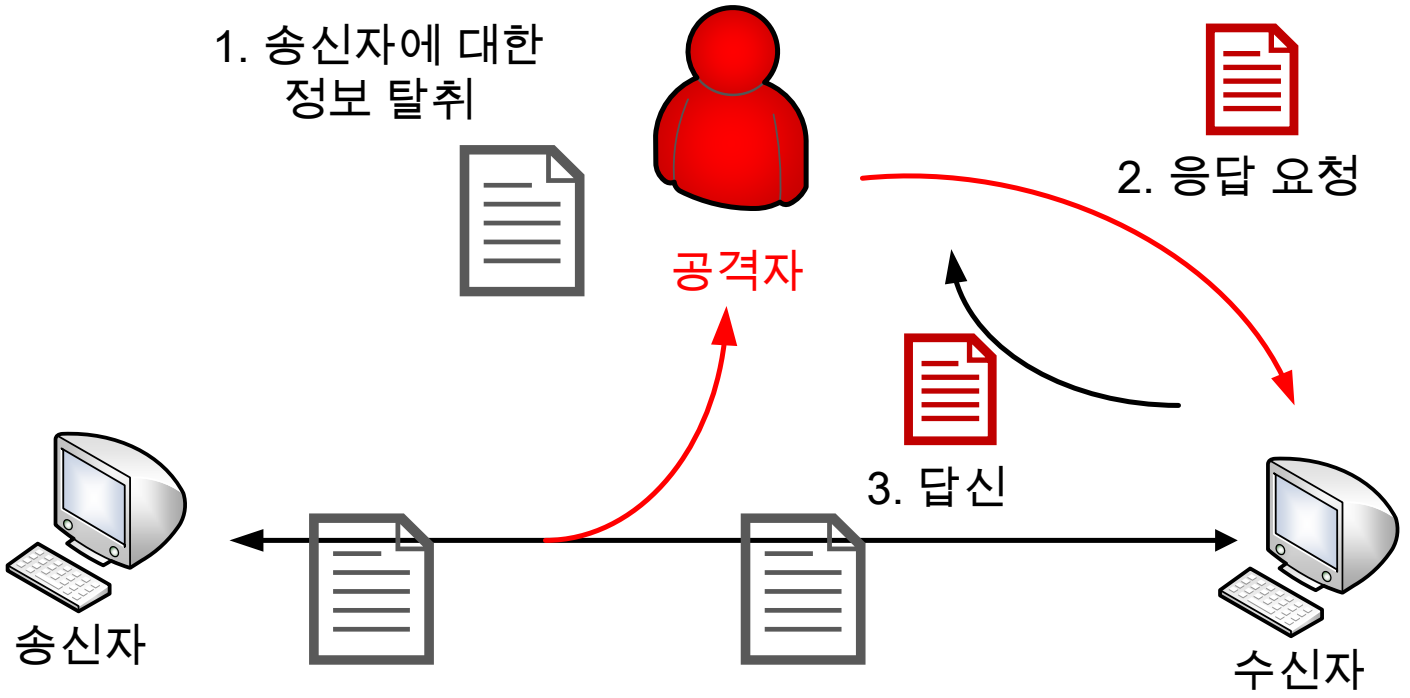
보 충

- 공격(Attack)
  - 무결성을 위협하는 공격(1/4)
    - 변조(Modification)
      - 공격자가 정보를 가로채거나 획득한 후 조작하는 행위
        - e.g., 공격자가 송금 요청을 가로채서 송금 계좌를 자신의 계좌로 변조하는 행위



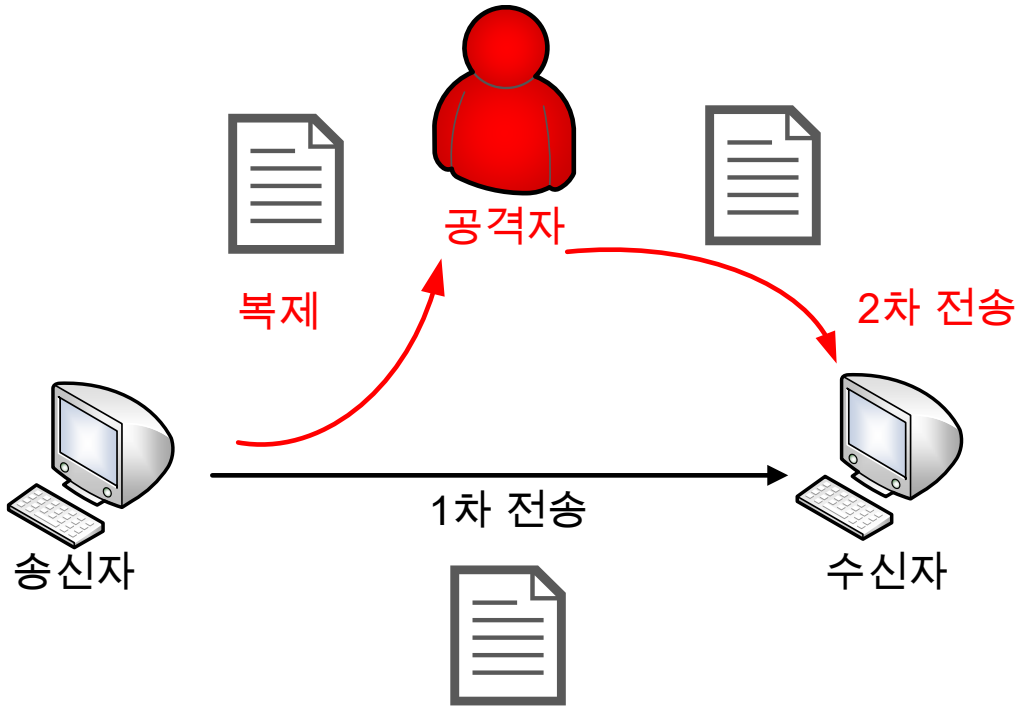
보 충

- 공격(Attack)
  - 무결성을 위협하는 공격(2/4)
    - 가장(Masquerading)
      - 공격자가 사용자의 정보를 탈취한 후, 사용자로 위장하는 행위
        - e.g., 공격자가 송신자로 위장하여 수신자에게 송금 요청을 보내는 행위



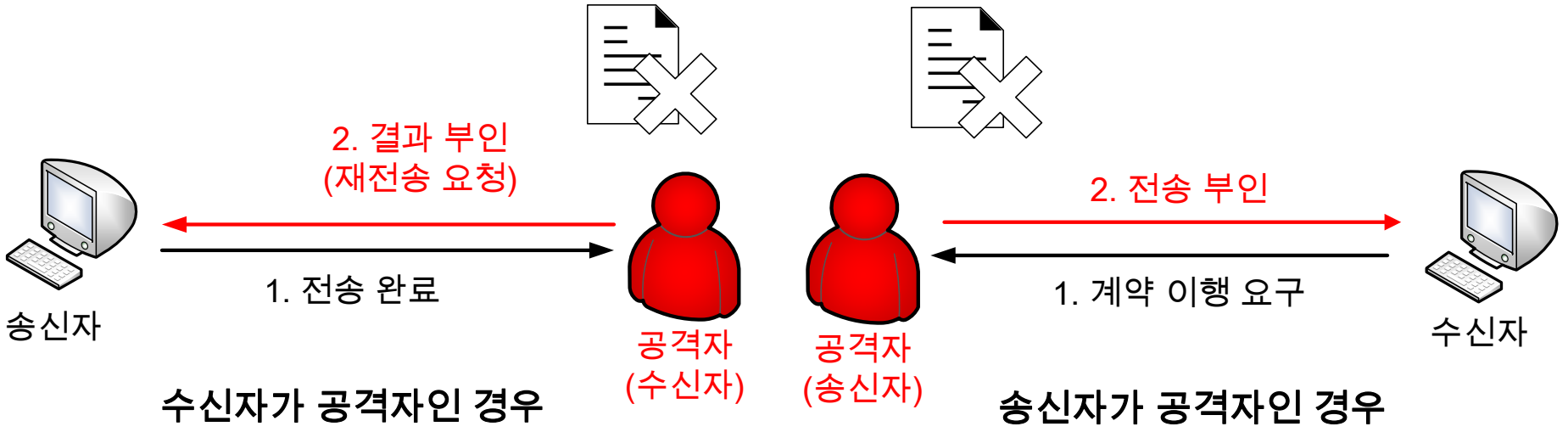
보 충

- 공격(Attack)
  - 무결성을 위협하는 공격(3/4)
    - 재전송(Replay)
      - 공격자가 정보의 복제본을 획득하고 다시 전송하는 행위
        - e.g.,공격자가 무선 자동차 키의 정보를 복제하여 차를 탈취하는 행위



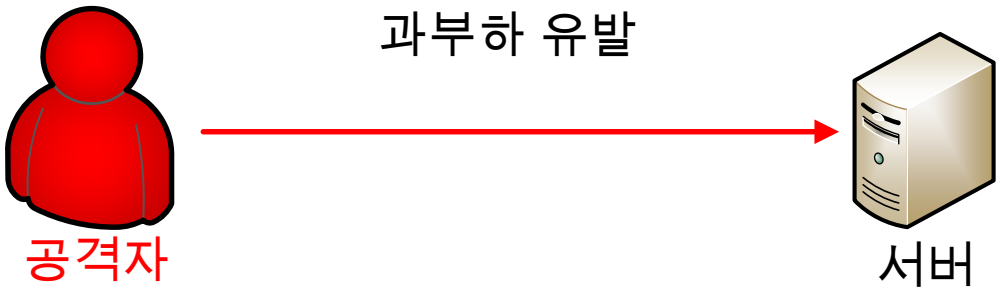
보 충

- 공격(Attack)
  - 무결성을 위협하는 공격(4/4)
    - 부인(Repudiation)
      - 공격자가 체결된 요청을 부인하는 행위
        - e.g., 입금이 완료된 거래에서 데이터를 위조하고 재입금을 요구하는 행위
        - e.g., 계약 체결 후 계약 동의서를 보낸 적 없다고 부인하는 행위



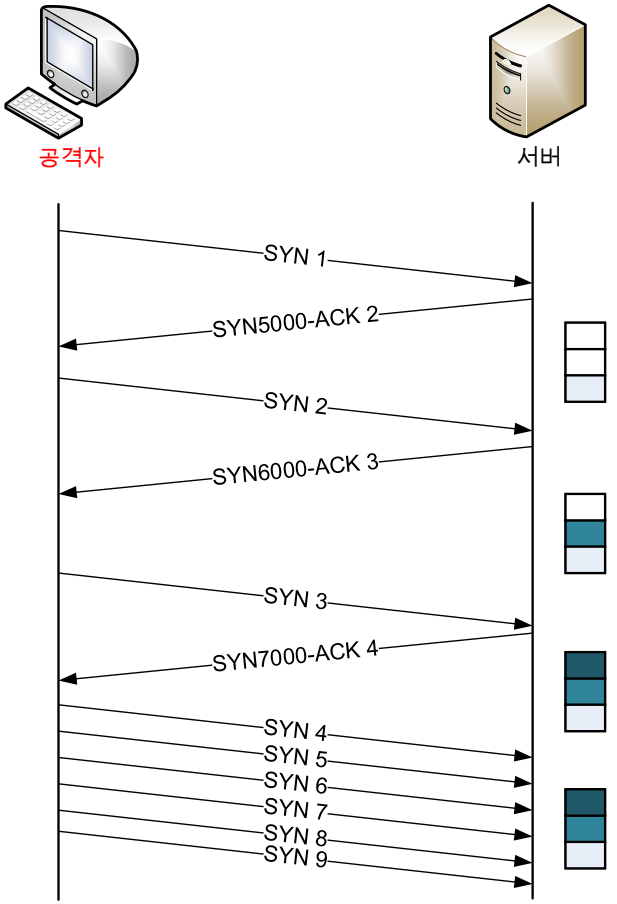
보 충

- 공격(Attack)
  - 가용성을 위협하는 공격
    - 서비스 거부(DoS, Denial of Service)
      - 시스템의 서비스를 마비시키거나 장애를 일으키는 공격
      - 단일 공격자가 공격
      - DDoS (Distributed Denial of Service)
        - 여러 대의 컴퓨터를 이용한 분산 공격



보 충

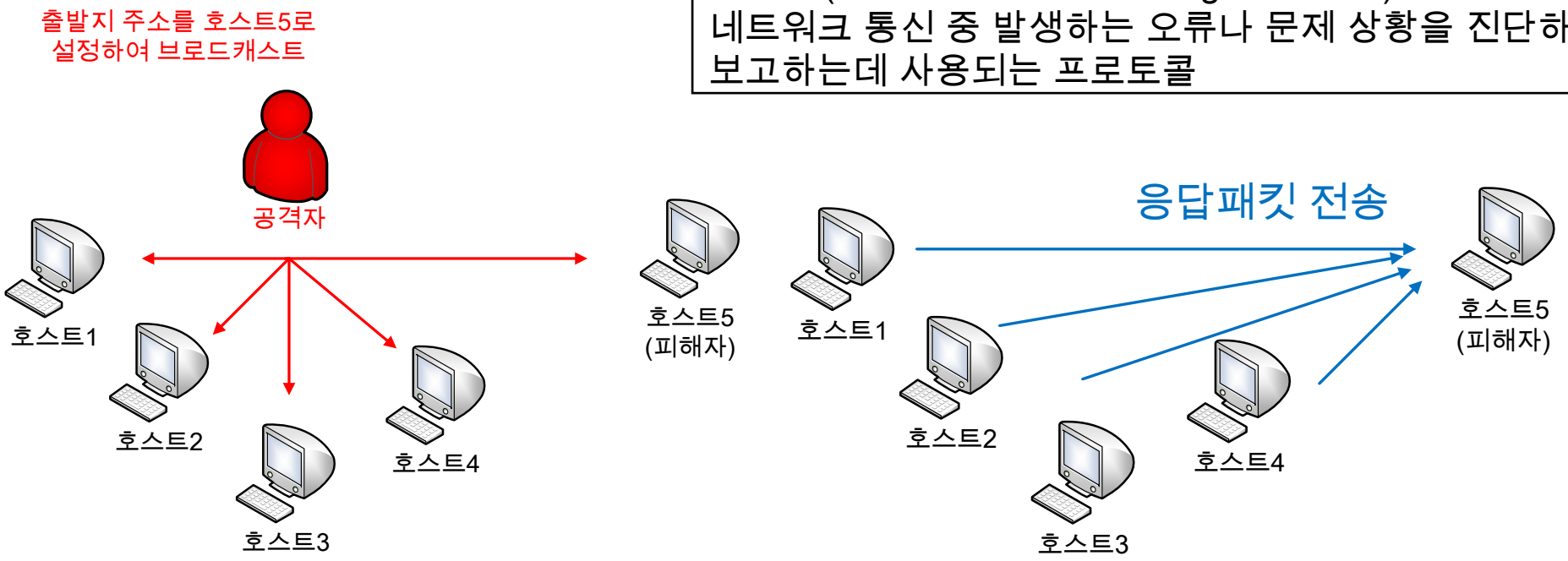
- 공격(Attack)
  - 서비스 거부(DoS, Denial of Service)종류(1/3)
    - SYN (SYNchronize) Flooding
      - 공격자가 서버로 대량의 요청을 보낸 후 연결 응답을 받지 않아 정상적인 사용자가 응답을 받지 못하게 마비시키는 공격



보 충

- 공격(Attack)
  - 서비스 거부(DoS, Denial of Service)종류(2/3)
    - Smurf Attack
      - 공격자가 데이터 패킷의 출발지를 피해자의 IP로 설정하여, 다수의 호스트들에게 *ICMP 메세지를 전송하고 피해자에게 다량의 응답을 전송시켜 시스템을 마비시키는 공격

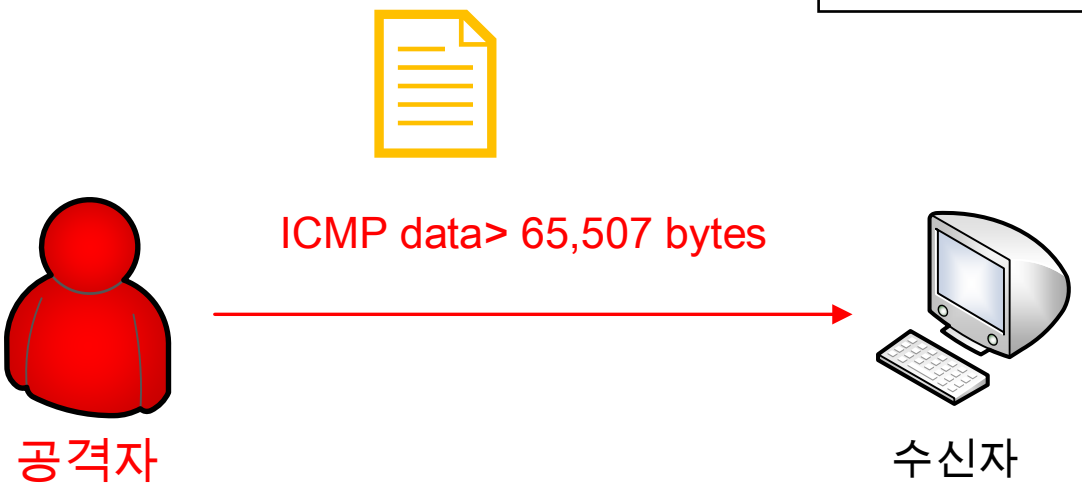
*ICMP (Internet Control Message Protocol)
네트워크 통신 중 발생하는 오류나 문제 상황을 진단하고 보고하는데 사용되는 프로토콜



보 충

- 공격(Attack)
  - 서비스 거부(DoS, Denial of Service)종류(3/3)
    - Ping of Death
      - 전송 가능한 최대 *패킷 크기를 초과한 ICMP 패킷을 전송해 대상 시스템을 지연, 마비시키는 공격
        - 전송 가능한 최대 패킷 크기는 65,507 bytes

*패킷
패킷은 데이터를 작게 나눈 조각으로
네트워크를 통해 전송되는 데이터의 단위



보 충

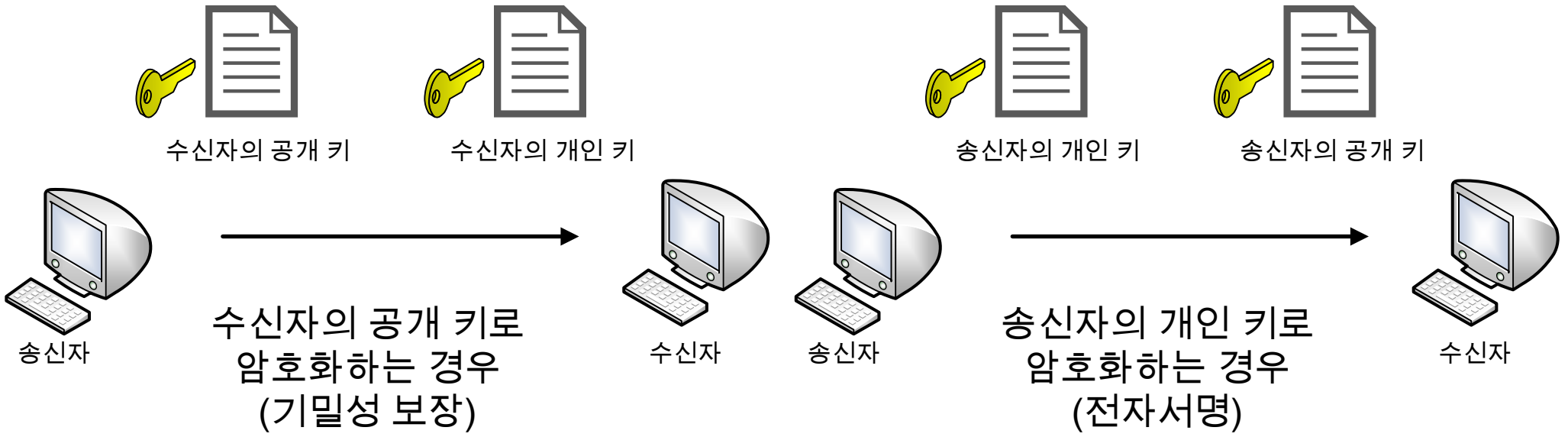
- 보안 매커니즘(2/2)

보안 매커니즘	방법	예시
트래픽 패딩 (Traffic Padding)	네트워크 상의 데이터를 분석하는 것을 방지하기 위해 데이터 스트림에 임의의 비트를 삽입하여 분석을 방해하는 과정	TLS(Transport Layer Security) Record Padding
라우팅 제어 (Routing Control)	데이터가 전달되는 과정에서 도청되는 것을 막기 위해 경로를 설정하는 과정	트래픽이 특정 국가를 경유하지 않도록 하는 것
공증 (Notarization)	공증인이 특정 사항을 증명하여 주는 과정	사용자 인증 프로토콜
접근 제어 (Access Control)	패스워드나 개인식별번호를 이용해 접근권을 확인하는 과정	사이트의 비밀번호

보 충

- 암호(Cryptography)
  - 비대칭-키 암호화(Asymmetric-key Encipherment)
    - 공개 키와 개인 키를 활용하는 암호화 방식

      e.g., RSA (Rivest-Shamir-Adleman) 알고리즘



보 충

- 행렬(Matrices)
  - 잉여 행렬(Residue Matrices)
    - 합동
      - 두 행렬이 열과 행의 개수가 같고 모든 대응되는 원소가 모듈로 n에 대해서 합동인 경우
      - $A \equiv B(mod\ n)$로 표기

      - e.g., $A = \begin{bmatrix} 7 & 11 \\ 3 & 4 \end{bmatrix},\ B = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$ 이면 각 행렬에 모듈로 5를 하면

      $7 \equiv 2(mod\ 5), 11 \equiv 1(mod\ 5),\ 3 \equiv 3(mod\ 5),\ 4 \equiv 4(mod\ 5)$
      ∴ 두 행렬은 모듈로 5에서 합동

보 충

- 선형 합동(Linear Congruences)(1/5)
  - 일변수 선형 방정식(Single-variable linear equations)(1/2)
    - $ax \equiv b(mod\ n)$형태의 방정식의 해를 구함
    - $\gcd(a, n) = d$라고 가정할 때
      - $d \nmid b$ 이면 해가 존재하지 않음
      - $d \mid b$ 이면 d개의 해가 존재
        - 해를 구하는 방법

1. 모듈로를 포함하여 방정식의 양변을 d로 나눔
2. 특수해 x_0를 구하기 위해 약분한 방정식의 양변에 a의 곱셈에 대한 역원을 곱함
3. 일반해는 $k = 0, 1, \cdots, (d-1)$에 대하여 $x = x_0 + k(n/d)$

보 충

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

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

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

        $x_1 = x_0 + 1 \times (18/2) = 15$

보 충

- 선형 합동(Linear Congruences)(3/5)
  - 일차 연립 방정식(1/3)
    - 동일한 모듈로에 대한 일차 방정식들의 집합에서 변수의 계수로 구성된 행렬의 역행렬이 존재하면 주어진 방정식들의 공통 해를 구할 수 있음

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

a. Equations

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

b. Interpretation

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

c. Solution

보 충

- 선형 합동(Linear Congruences)(4/5)
  - 일차 연립 방정식(2/3)
    - 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

1. 연립 방정식을 행렬 A로 바꿈 $\mathrm{A} = \begin{bmatrix} 3 & 5 & 7 \\ 1 & 4 & 13 \\ 2 & 7 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 4 \end{bmatrix}$
2. 행렬 A의 역행렬 구하기

행렬 $\mathrm{A}^{-1} = \frac{1}{129} \begin{bmatrix} 79 & -34 & -37 \\ -23 & 5 & 32 \\ 1 & 11 & -7 \end{bmatrix}$

보 충

- 선형 합동(Linear Congruences)(5/5)
  - 일차 연립 방정식(3/3)

3. 양변에 각각 행렬 A의 곱셈에 대한 역행렬 A^{-1}을 곱함

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \frac{1}{129}\begin{bmatrix}79 & -34 & -37\\ -23 & 5 & 32\\ 1 & 11 & -7\end{bmatrix}\begin{bmatrix}3\\ 5\\ 4\end{bmatrix}$$

4. 모듈로 연산

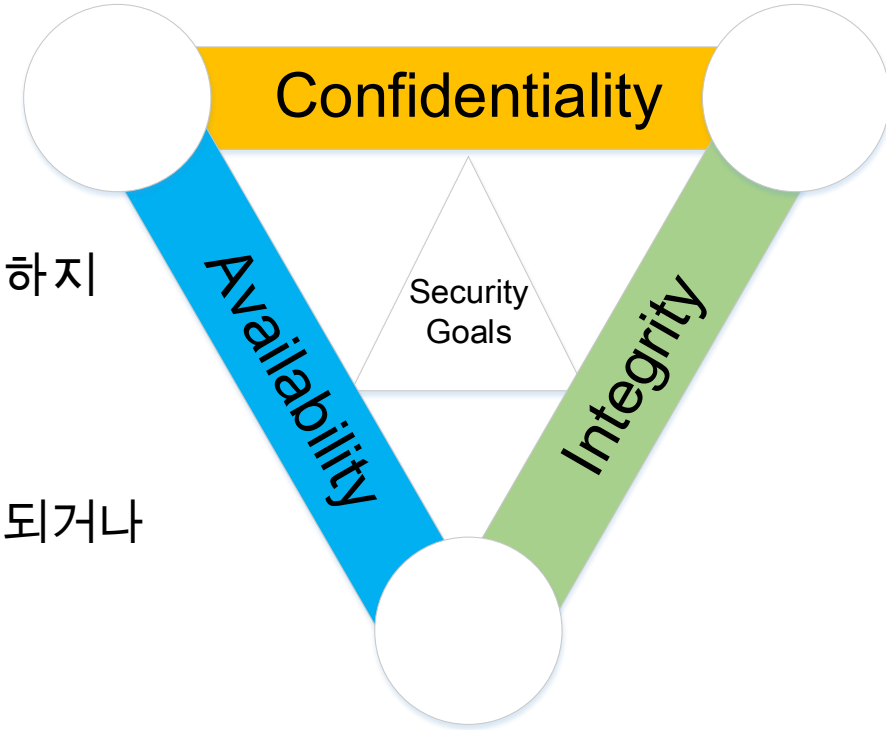
$$\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix}15(mod\ 16)\\ 4(mod\ 16)\\ 14(mod\ 16)\end{bmatrix}$$

목 차

- 보충
- 보안 개요
- 암호 수학

보안 개요

- 보안
  - 보안 목표
    - 정보를 보호하기 위해 지켜야 할 목표
    - CIA Triad
      - 기밀성(Confidentiality)
        - 인가되지 않은 사용자가 정보에 접근하지 못하도록 보호하는 것
          - e.g., 스누핑(Snoofing)
      - 무결성(Integrity)
        - 정보가 인가되지 않은 방식으로 변경되거나 손상되지 않도록 보호하는 것
          - e.g., 변조(Modification)
      - 가용성(Availability)
        - 인가된 사용자가 필요할 때 정보와 자원을 사용할 수 있도록 보장하는 것
          - e.g., 서비스 거부 공격(DoS, Denial of Service)

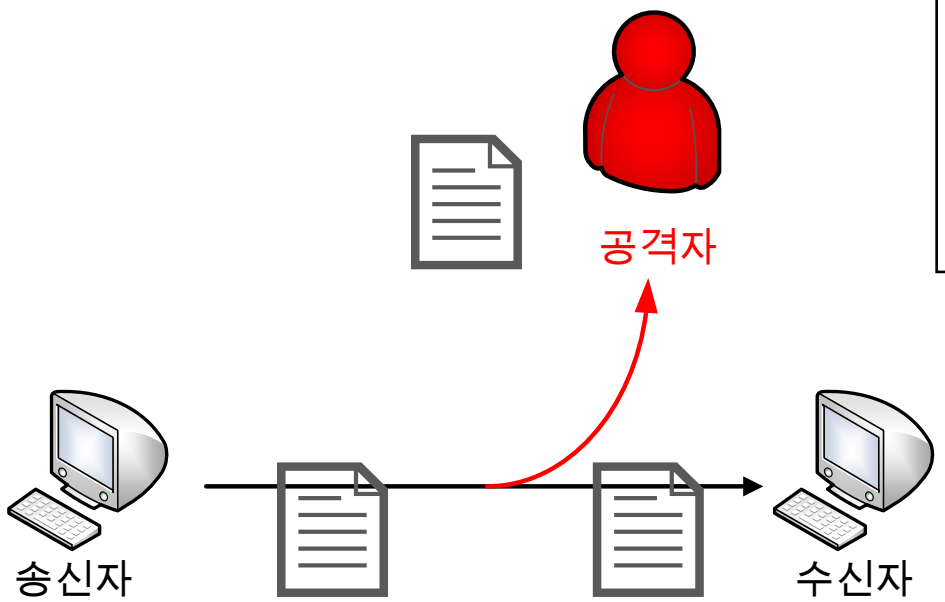


보안 개요

- 공격(Attack)
  - 정보 및 자원을 위협하는 행위
  - 보안 목표와 관련된 공격의 종류
    - 기밀성을 위협하는 공격
      - e.g., 스누핑(Snoofing), 트래픽 분석(Traffic Analysis)
    - 무결성을 위협하는 공격
      - e.g., 변조(Modification), 가장(Masquerading), 재전송(Replay) 부인(Repudiation)
    - 가용성을 위협하는 공격
      - e.g., 서비스 거부 공격(DoS, Denial of Service)

보안 개요

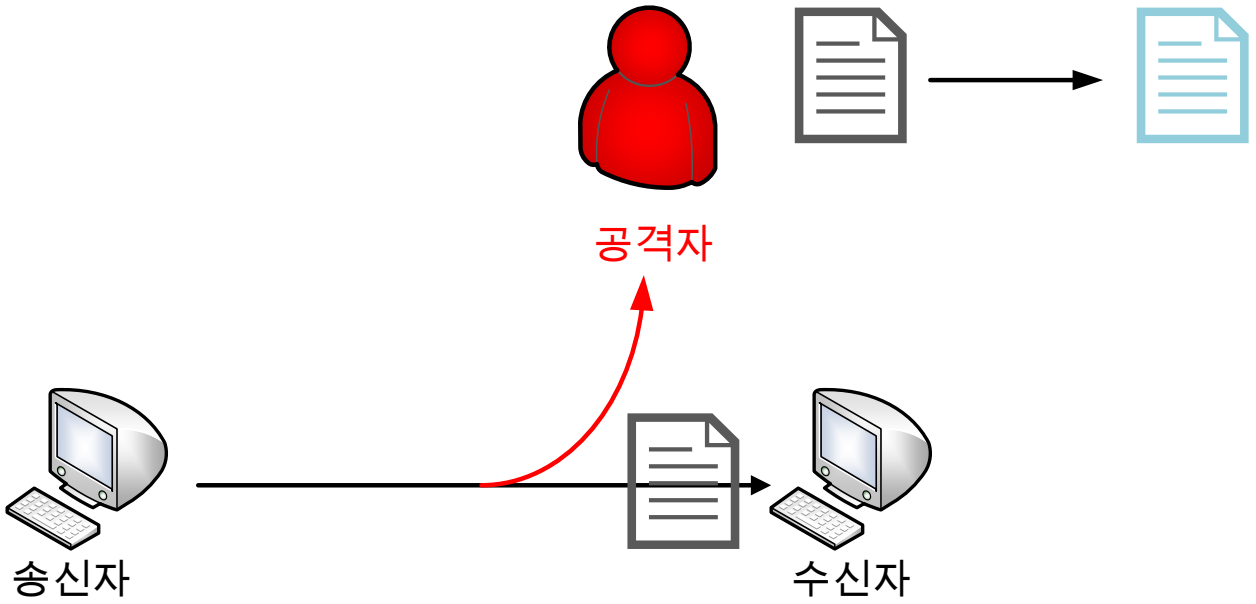
- 공격(Attack)
  - 기밀성을 위협하는 공격(1/2)
    - 스누핑(Snooping)
      - 공격자가 전송되는 정보를 몰래 가로채고 획득하는 행위
        - e.g., 공격자가 수신자의 은행 파일을 캡처하여 로그인하는 행위



*스니핑(Sniffing)
데이터를 몰래 훔쳐보는 행위
*스푸핑(Spoofing)
공격자가 네트워크, 웹사이트 등의 데이터 위변조를 통해 정상 시스템인 것처럼 위장하여 일반 사용자를 속이는 행위

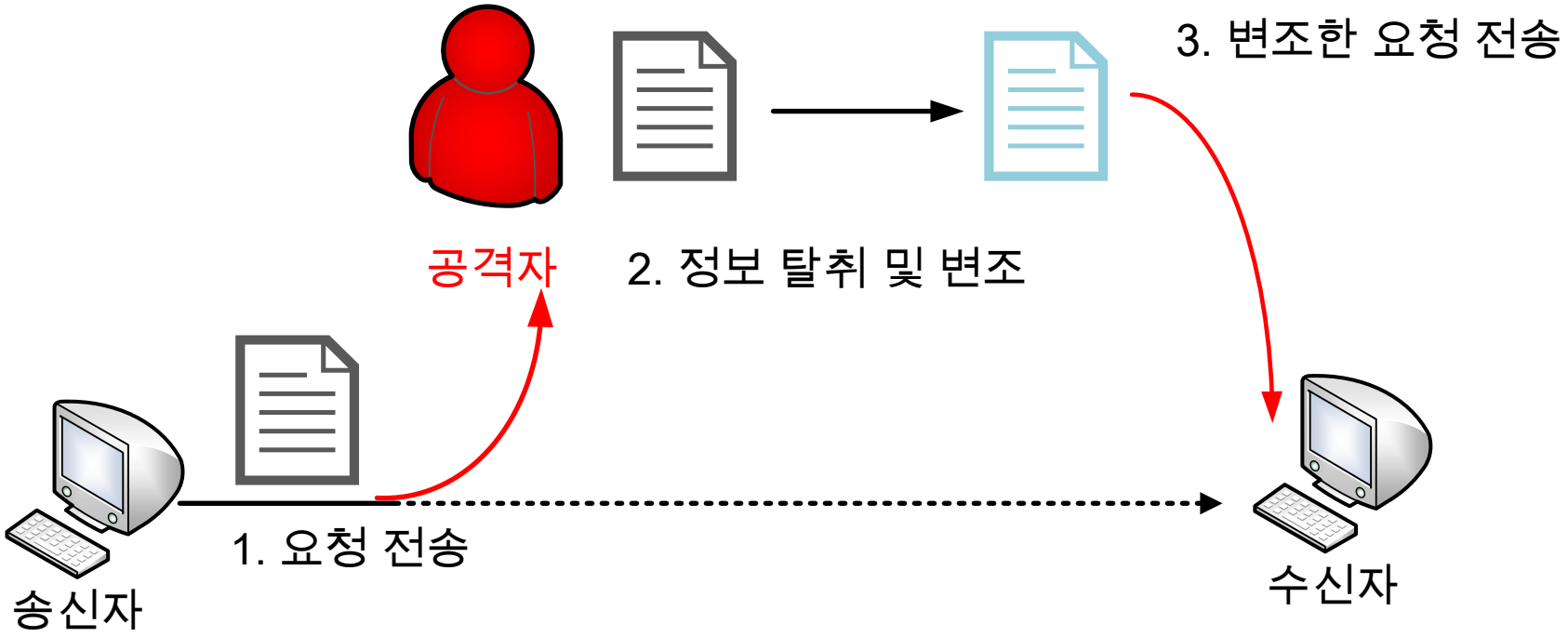
보안 개요

- 공격(Attack)
  - 기밀성을 위협하는 공격(2/2)
    - 트래픽 분석(Traffic Analysis)
      - 공격자가 암호화된 정보를 이해할 수 없더라도 트래픽을 분석함으로써 다른 형태의 정보를 얻는 행위
        - e.g., 적국이 특정 시간에 반복해서 데이터 패킷 수가 급증하는 것을 보고 보안이 강화되는 시간을 알아낼 수 있는 행위



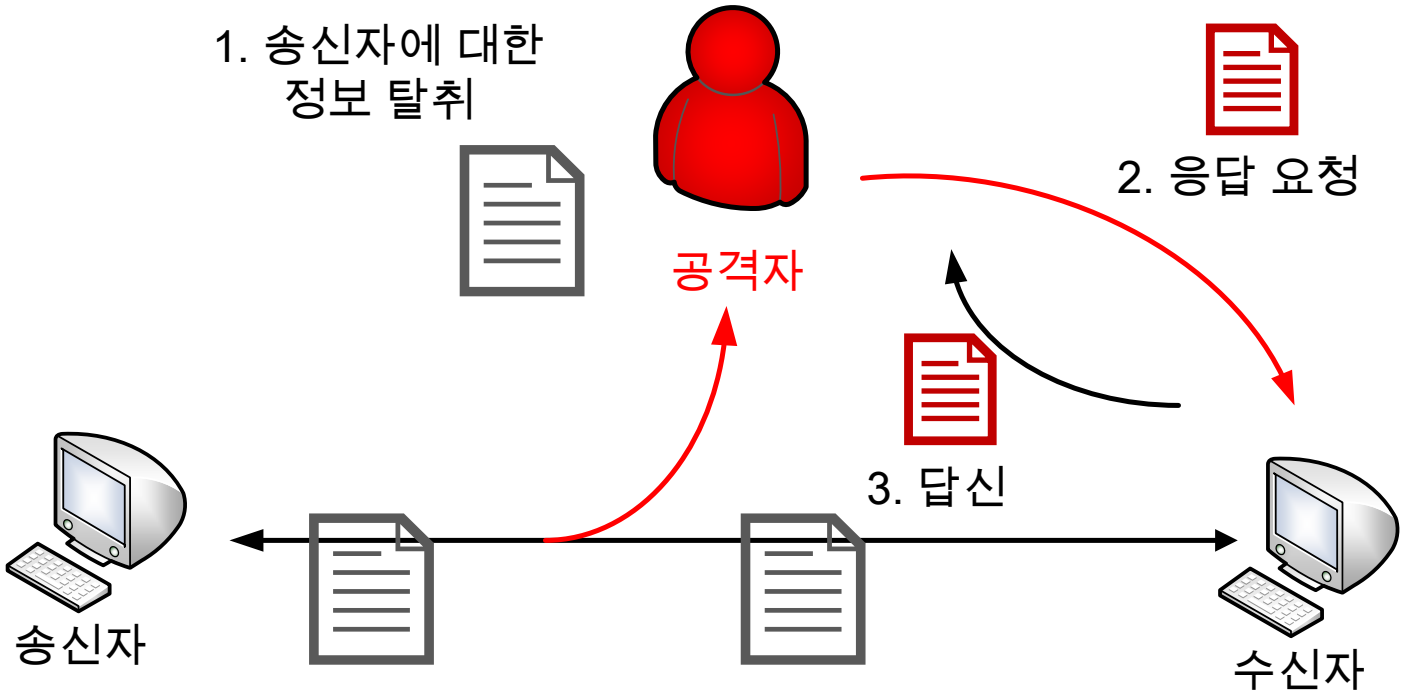
보안 개요

- 공격(Attack)
  - 무결성을 위협하는 공격(1/4)
    - 변조(Modification)
      - 공격자가 정보를 가로채거나 획득한 후 조작하는 행위
        - e.g., 공격자가 송금 요청을 가로채서 송금 계좌를 자신의 계좌로 변조하는 행위



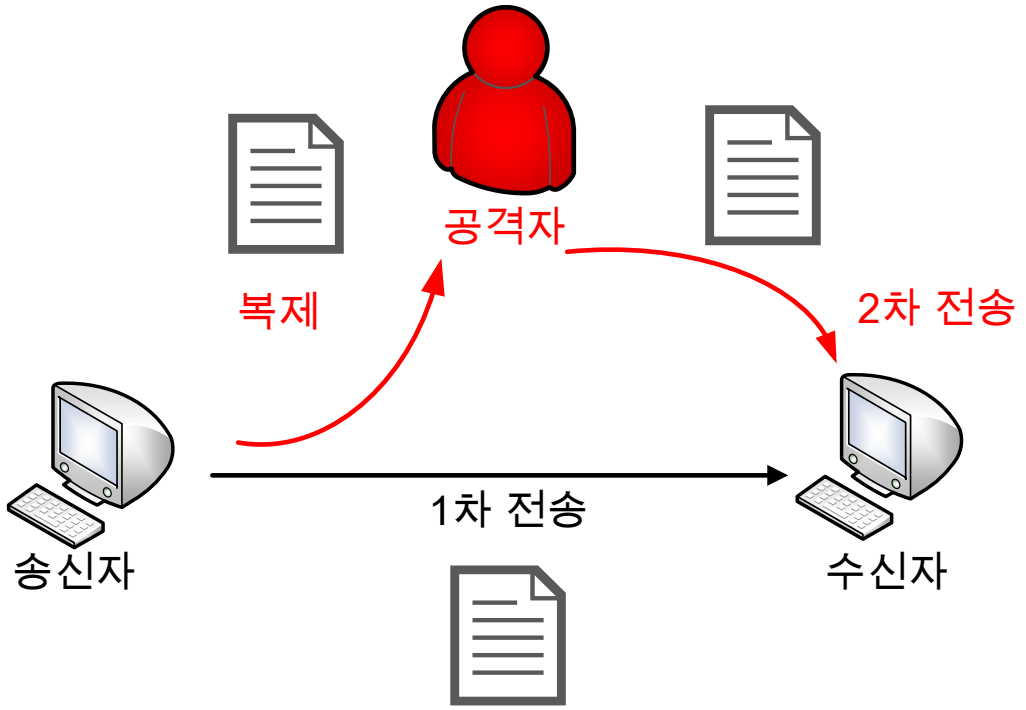
보안 개요

- 공격(Attack)
  - 무결성을 위협하는 공격(2/4)
    - 가장(Masquerading)
      - 공격자가 사용자의 정보를 탈취한 후, 사용자로 위장하는 행위
        - e.g., 공격자가 송신자로 위장하여 수신자에게 송금 요청을 보내는 행위



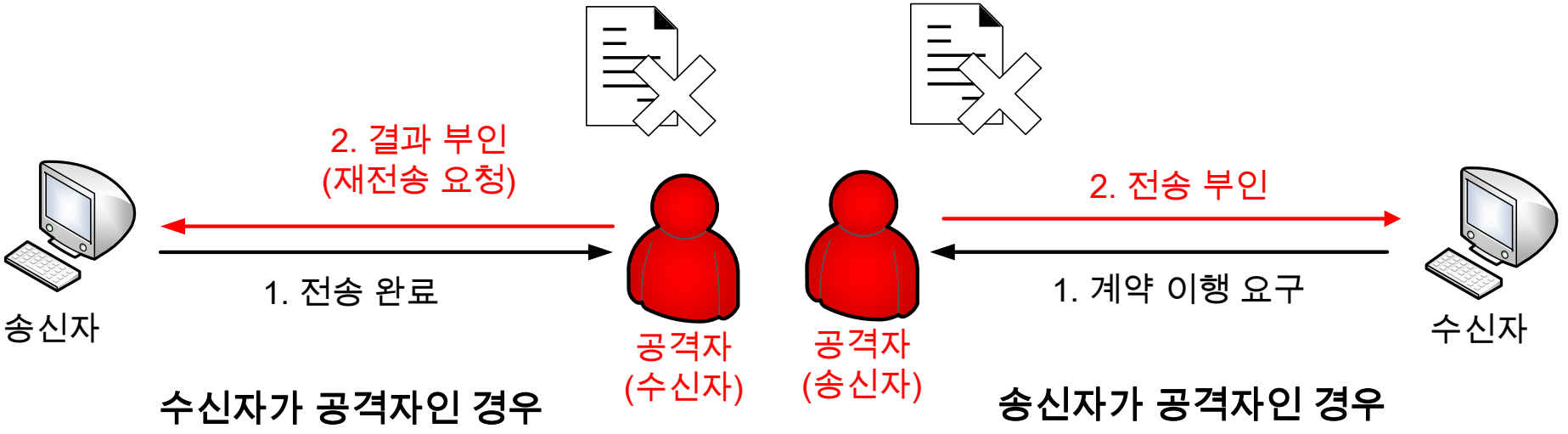
보안 개요

- 공격(Attack)
  - 무결성을 위협하는 공격(3/4)
    - 재전송(Replay)
      - 공격자가 정보의 복제본을 획득하고 다시 전송하는 행위
        - e.g., 공격자가 무선 자동차 키의 정보를 복제하여 차를 탈취하는 행위



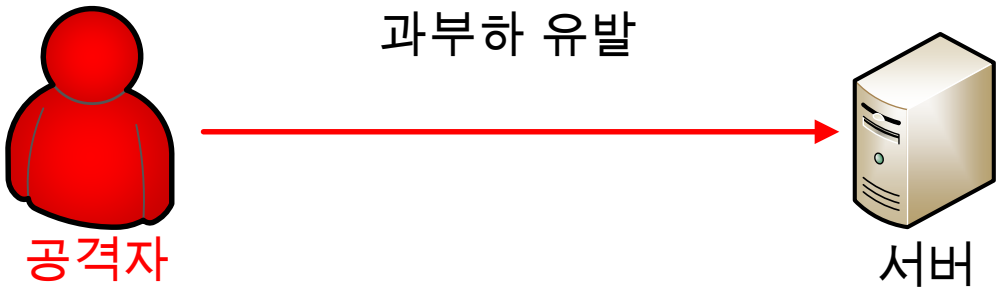
보안 개요

- 공격(Attack)
  - 무결성을 위협하는 공격(4/4)
    - 부인(Repudiation)
      - 공격자가 체결된 요청을 부인하는 행위
        - e.g., 입금이 완료된 거래에서 데이터를 위조하고 재입금을 요구하는 행위
        - e.g., 계약 체결 후 계약 동의서를 보낸 적 없다고 부인하는 행위



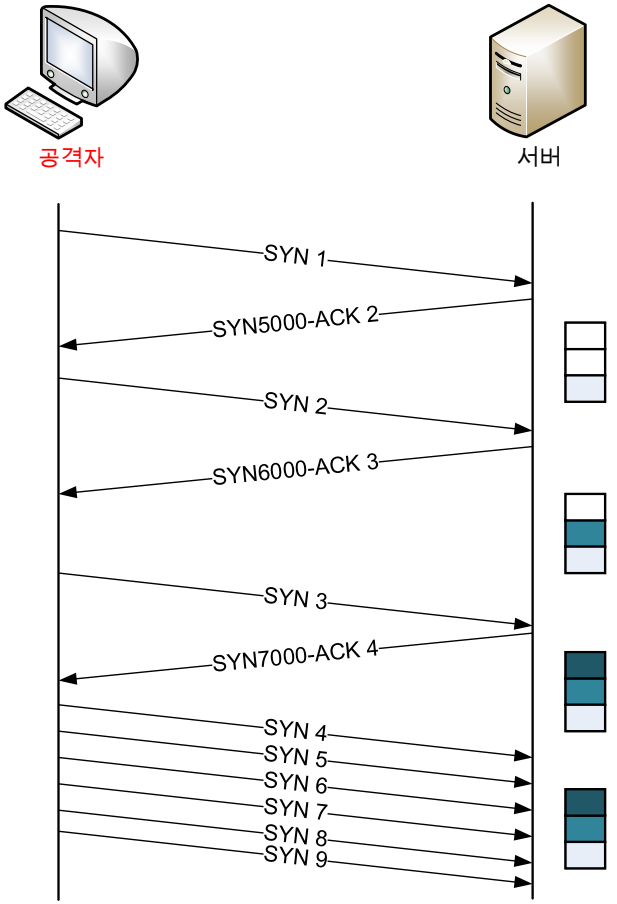
보안 개요

- 공격(Attack)
  - 가용성을 위협하는 공격
    - 서비스 거부(DoS, Denial of Service)
      - 시스템의 서비스를 마비시키거나 장애를 일으키는 공격
      - 단일 공격자가 공격
      - DDoS (Distributed Denial of Service)
        - 여러 대의 컴퓨터를 이용한 분산 공격



보안 개요

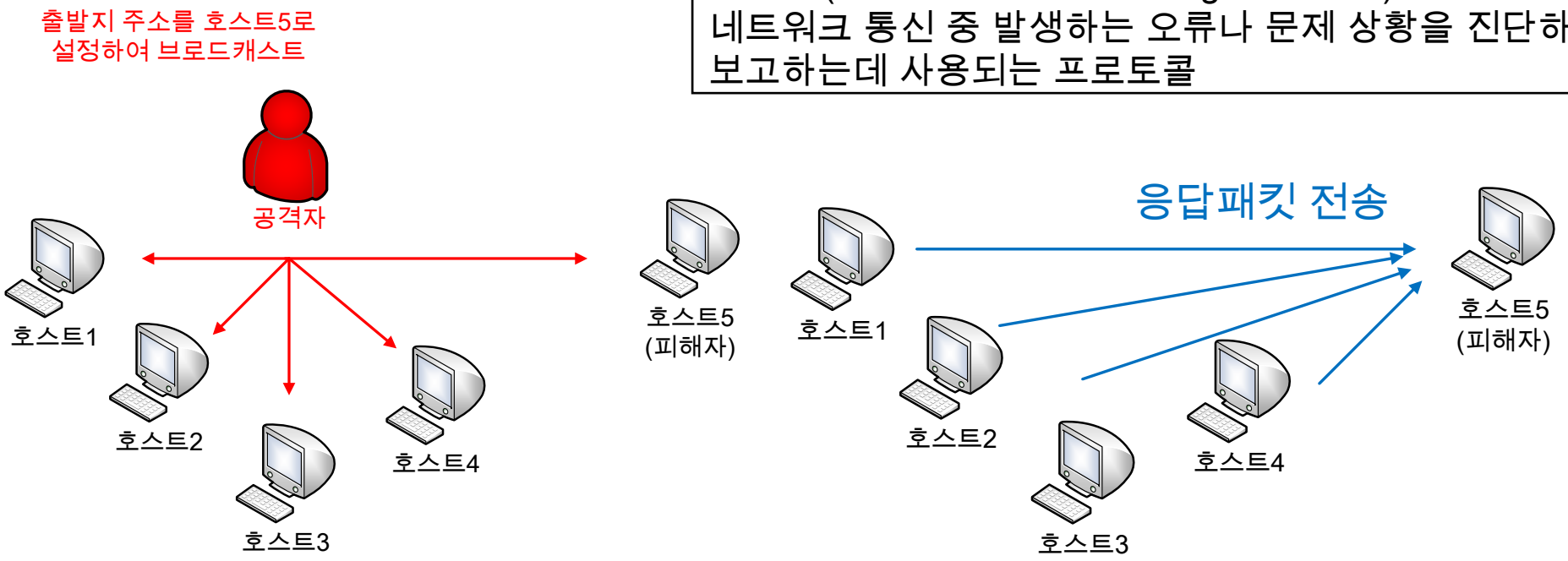
- 공격(Attack)
  - 서비스 거부(DoS, Denial of Service)종류(1/3)
    - SYN (SYNchronize) Flooding
      - 공격자가 서버로 대량의 요청을 보낸 후 연결 응답을 받지 않아 정상적인 사용자가 응답을 받지 못하게 마비시키는 공격



보안 개요

- 공격(Attack)
  - 서비스 거부(DoS, Denial of Service)종류(2/3)
    - Smurf Attack
      - 공격자가 데이터 패킷의 출발지를 피해자의 IP로 설정하여, 다수의 호스트들에게 *ICMP 메세지를 전송하고 피해자에게 다량의 응답을 전송시켜 시스템을 마비시키는 공격

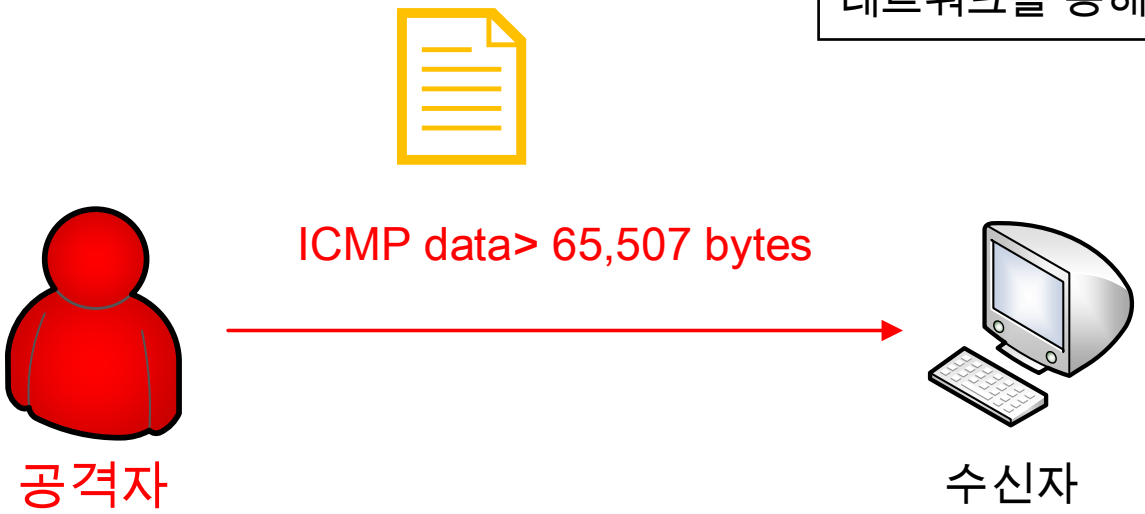
*ICMP (Internet Control Message Protocol)
네트워크 통신 중 발생하는 오류나 문제 상황을 진단하고 보고하는데 사용되는 프로토콜



보안 개요

- 공격(Attack)
  - 서비스 거부(DoS, Denial of Service)종류(3/3)
    - Ping of Death
      - 전송 가능한 최대 *패킷 크기를 초과한 ICMP 패킷을 전송해 대상 시스템을 지연, 마비시키는 공격
        - 전송 가능한 최대 패킷 크기는 65,507 bytes

*패킷
패킷은 데이터를 작게 나눈 조각으로
네트워크를 통해 전송되는 데이터의 단위



보안 개요

- 공격(Attack)
  - 소극적 공격과 적극적 공격
    - 공격의 특성에 따라 공격을 두 그룹으로 나눌 수 있음
      - 소극적 공격(Passive Attack)
        - 기밀성을 위협하는 공격
        - 정보를 획득하는 것을 목표로 하는 공격
        - 데이터의 암호화로 예방 가능
      - 적극적 공격(Active Attack)
        - 무결성과 가용성을 위협하는 공격
        - 데이터를 바꾸거나 시스템에 해를 입히는 공격
        - 방어보다는 탐지하는 것이 더 쉬움

보안 개요

- 보안 서비스
  - 보안 공격에 대응하기 위한 서비스

보안 서비스	설명
데이터 기밀성 (Data Confidentiality)	노출 공격으로부터 데이터를 보호
데이터 무결성 (Data Integrity)	데이터의 변경, 삽입, 삭제, 재전송 등으로 부터 보호
인증 (Authentication)	사용자의 신원 혹은 데이터의 출처를 확인
부인방지 (Nonrepudiation)	송신자나 수신자가 행위를 부인하는 것을 방지
접근 제어 (Access Control)	비 인가자로부터의 접근을 통제

보안 개요

- 보안 매커니즘(1/2)
  - 보안 서비스를 구현, 지원하기 위해 사용되는 기능과 절차

보안 매커니즘	방법	예시
암호화 (Encipherment)	데이터를 숨기거나 보호	AES(Advanced Encryption Standard)
데이터 무결성 (Data Integrity)	데이터가 변경되거나 손상되지 않았음을 검증	해시함수
디지털 서명 (Digital Signature)	송신자가 개인키로 디지털 서명을 생성, 수신자가 송신자의 공개키로 서명을 검증	소프트웨어 설치 시 제공되는 공급자의 전자서명
인증 교환 (Authentication Exchange)	신원을 증명하기 위해 특정 메시지를 주고 받음	OTP(One-Time Password)

보안 개요

- 보안 매커니즘(2/2)

보안 매커니즘	방법	예시
트래픽 패딩 (Traffic Padding)	네트워크 상의 데이터를 분석하는 것을 방지하기 위해 데이터 스트림에 임의의 비트를 삽입하여 분석을 방해하는 과정	TLS(Transport Layer Security) Record Padding
라우팅 제어 (Routing Control)	데이터가 전달되는 과정에서 도청되는 것을 막기 위해 경로를 설정하는 과정	트래픽이 특정 국가를 경유하지 않도록 하는 것
공증 (Notarization)	공증인이 특정 사항을 증명하여 주는 과정	사용자 인증 프로토콜
접근 제어 (Access Control)	패스워드나 개인식별번호를 이용해 접근권을 확인하는 과정	사이트의 비밀번호

보안 개요

- 보안 서비스와 보안 매커니즘의 관계

보안 서비스	보안 매커니즘
데이터 기밀성 (Data Confidentiality)	암호화, 라우팅 제어
데이터 무결성 (Data Integrity)	암호화, 디지털 서명, 데이터 무결성
인증(Authentication)	암호화, 디지털 서명, 인증 교환
부인방지 (Nonrepudiation)	디지털 서명, 데이터 무결성, 공증
접근 제어 (Access Control)	접근 제어

보안 개요

- 암호(Cryptography)(1/4)
  - 정보를 제3자가 볼 수 없도록 비식별화 하는 행위
  - 암호의 종류
    - 대칭-키 암호화(Symmetric-key Encipherment)
    - 비대칭-키 암호화(Asymmetric-key Encipherment)
    - 해싱(Hashing)

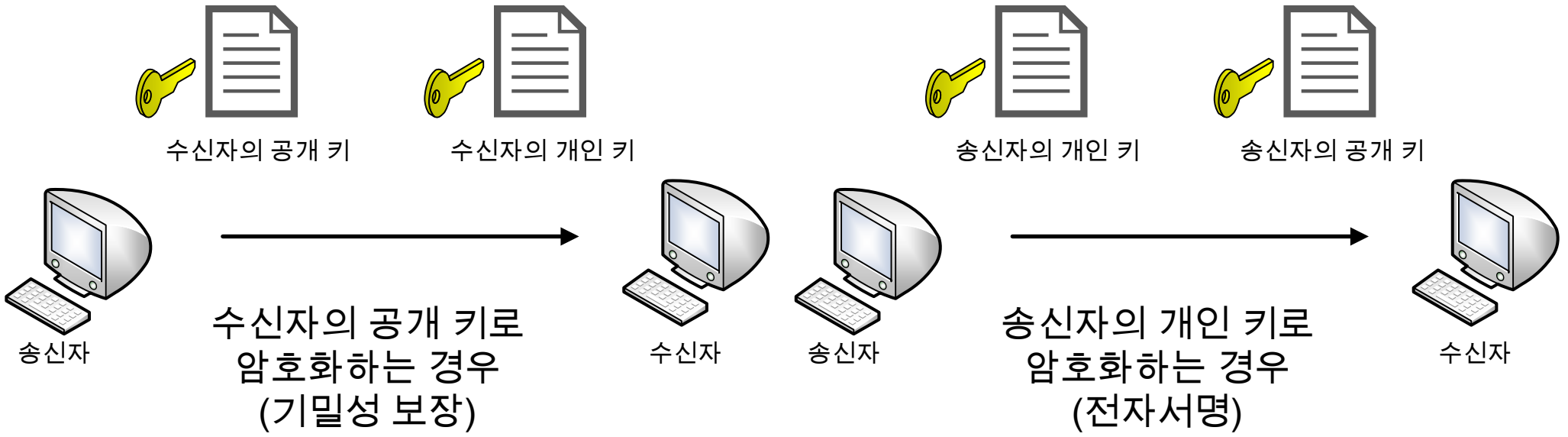
보안 개요

- 암호(Cryptography)(2/4)
  - 대칭-키 암호화(Symmetric-key Encipherment)
    - 암호화와 복호화에 하나의 비밀 키를 활용하는 암호화 방식
      - e.g., AES (Advanced Encryption Standard)알고리즘

보안 개요

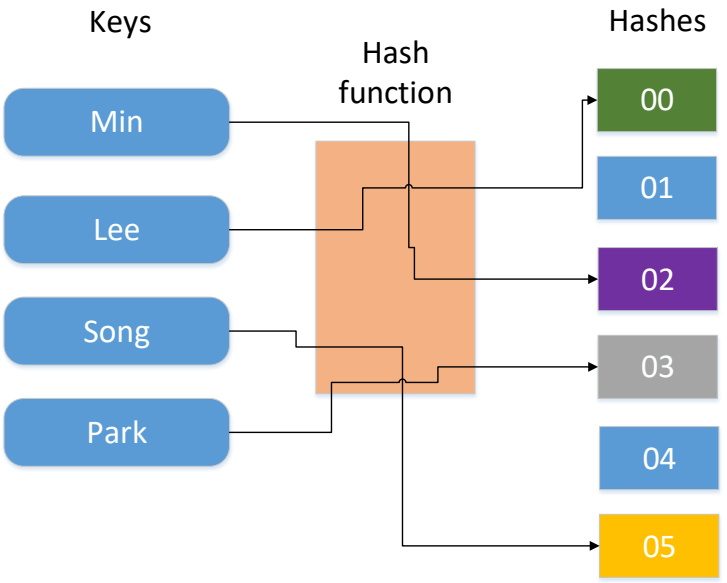
- 암호(Cryptography)(3/4)
  - 비대칭-키 암호화(Asymmetric-key Encipherment)
    - 공개 키와 개인 키를 활용하는 암호화 방식

      e.g., RSA (Rivest-Shamir-Adleman) 알고리즘



보안 개요

- 암호(Cryptography)(4/4)
  - 해싱(Hashing)
    - 입력 데이터를 고정된 비트의 값으로 전환하는 암호화 방식
    - 특징
      - 해시 함수 적용시 원래 데이터를 알 수 없음
      - 해시 충돌이 일어날 확률은 극히 드뭄
      - 데이터 변조의 유무를 증명하는 데 사용함
      - e.g., SHA (Secure Hash Algorithm)

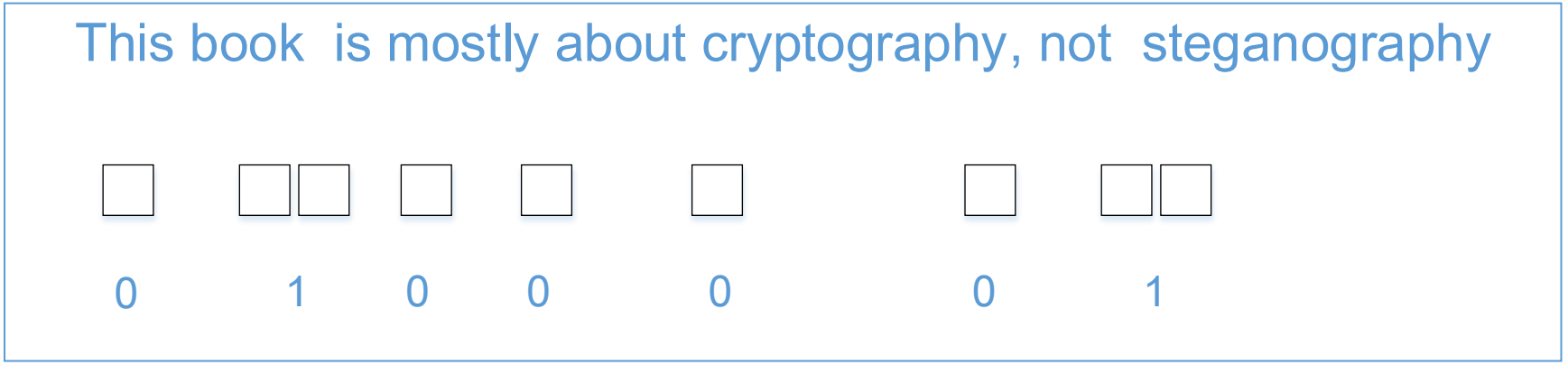


보안 개요

- 스테가노그래피(Steganography)(1/3)
  - 메시지를 다른 것으로 덮어서 감추는 것
    - e.g.,나무 조각에 메시지를 새기고 밀랍으로 봉인, 단순 메시지 안에 비밀 메시지를 숨김
  - 스테가노그래피의 종류
    - 텍스트 커버
      - 텍스트 안에 이진 데이터를 삽입하는 방법
    - 이미지 커버
      - 컬러 이미지 안에 데이터를 감추는 방법

보안 개요

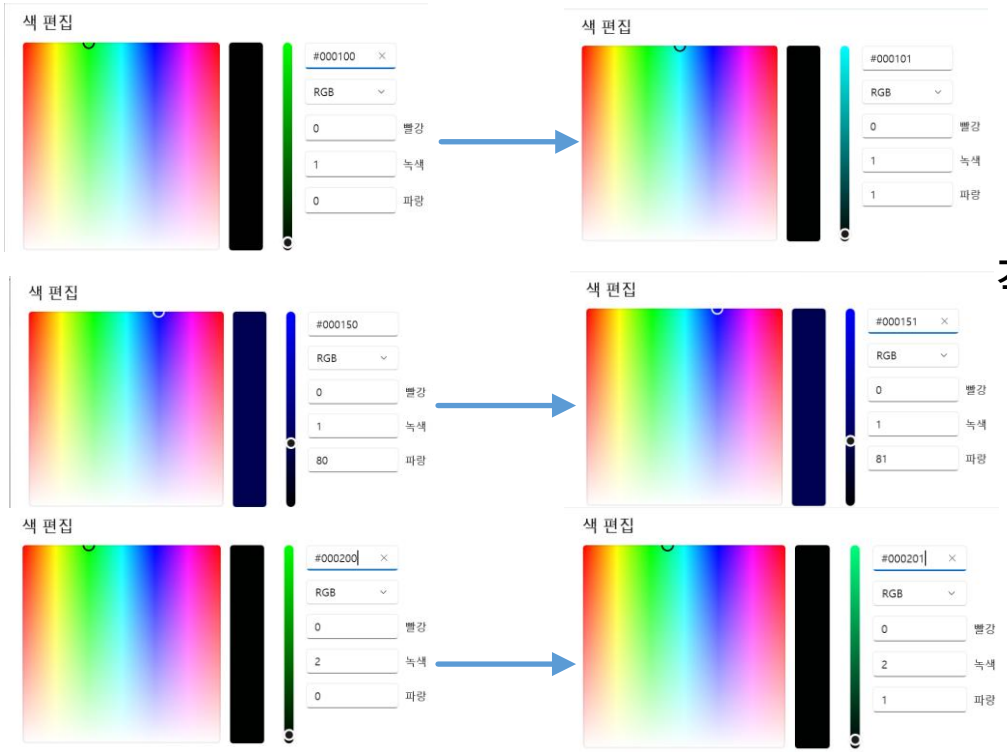
- 스테가노그래피(Steganography)(2/3)
  - 텍스트 커버
    - 문장 속 띄어쓰기 간격을 이용하여 이진 데이터 삽입
      - e.g., 한 칸 띄어쓰기는 0, 두 칸 띄어쓰기는 1이라고 가정할 때



      - 사진 속 이진수는 0100001
      - 이를 아스키코드에 대입 시 문자 !를 나타냄

보안 개요

- 스테가노그래피(Steganography)(3/3)
  - 이미지 커버 예시
    - e.g., RGB 값이(100, 150, 200) 라면, 이진수는 01100100, 10010110, 11001000
    - 각 최하위 비트를 조정하여 원하는 이진 데이터를 은닉



100, 150, 200를 이진수로 바꾸면
01100100, 10010110, 11001000
각 최하위 비트가 0이면 1로, 1이면 0으로 전환
숨겨진 메시지는 111
아스키코드로 전환시 .

목 차

- 보충
- 보안 개요
- 암호 수학

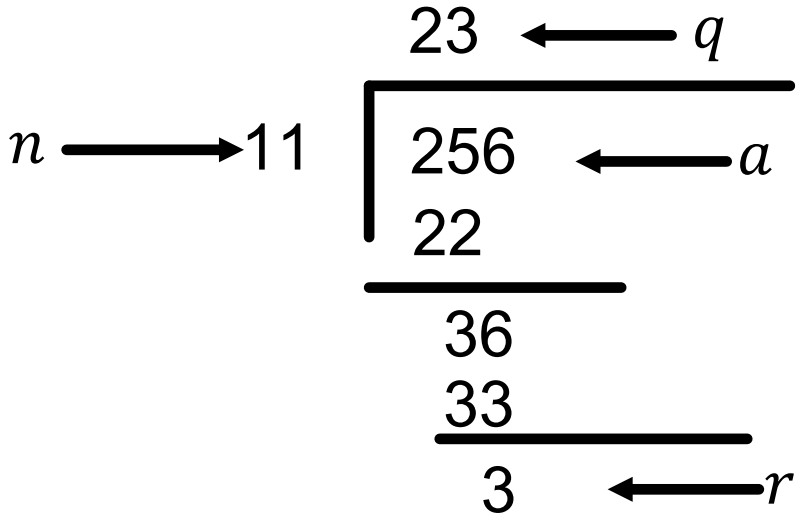
암호 수학

- 정수 연산(1/12)
  - *정수 집합(Z)에서의 산술 연산
  - 이진 연산
    - 두 개의 입력 값을 연산하여 하나의 결과값을 산출하는 것
    - 덧셈, 뺄셈, 곱셈으로 이루어져 있음
      - 나눗셈은 두 개의 결과값(몫, 나머지)을 산출하므로 속하지 않음

*정수집합(Set of Integers) Z는 음의 무한대에서부터 양의 무한대까지의 모든 정수

암호 수학

- 정수 연산(2/12)
  - 정수의 나눗셈
    - 정수 연산에서 a를 n으로 나누면 q와 r을 얻음
    - $a = q \times n + r$
      - 이 관계식에서 a는 피제수, q는 몫, n은 제수, r은 나머지라고 함
      - $n > 0$
      - $r \geq 0$



암호 수학

- 정수 연산(3/12)
  - 가분성(Divisibility)(1/7)
    - 나눌 수 있는 성질
    - $a = q \times n + r$ 에서 $r = 0$이면 $n \mid a$
      - n이 a를 나눈다 라고 함
    - $a = q \times n + r$ 에서 $r \neq 0$이면 $n \nmid a$
      - n이 a를 나누지 않는다고 함

암호 수학

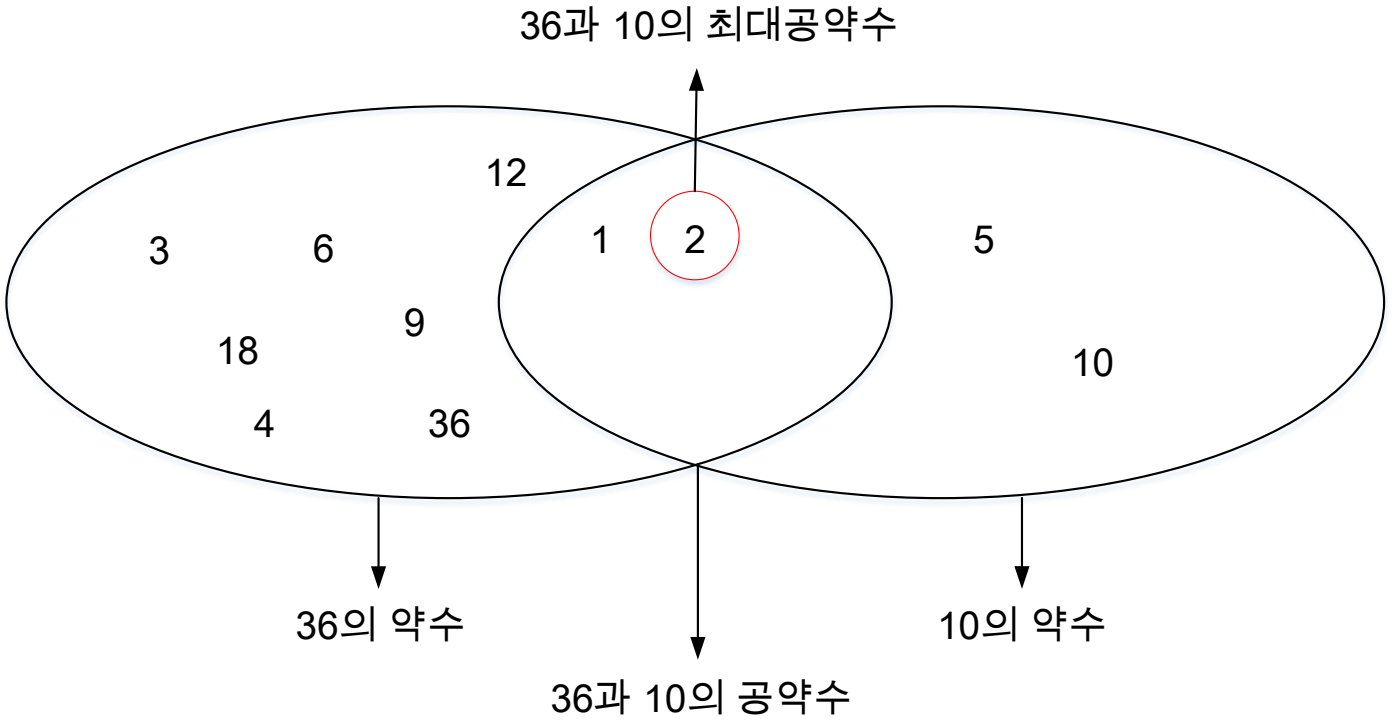
- 정수 연산(4/12)
  - 가분성(Divisibility)(2/7)
    - 성질
      - 만약 $a \mid 1$ 이면 $a = \pm 1$
      - 만약 $a \mid b$ 이고 $b \mid a$ 이면 $a = \pm b$
      - 만약 $a \mid b$ 이고 $b \mid c$ 이면 $a \mid c$
      - 만약 $a \mid b$ 이고 $a \mid c$ 이면 $a \mid (m \times b + n \times c)$, 여기서 m과 n은 임의의 정수

암호 수학

- 정수 연산(5/12)
  - 가분성(Divisibility)(3/7)
    - 약수(Divisor)
      - 사실1. 어떤 수를 나누어 떨어지게 하는 수
      - 사실2. 정수 1은 하나의 약수, 1만을 가짐
      - 사실3. 1을 제외한 모든 양의 정수는 최소 2개의 약수(1과 그 자신)를 가짐

암호 수학

- 정수 연산(6/12)
  - 가분성(Divisibility)(4/7)
    - 최대공약수(GCD, Greatest Common Divisor)
      - 두 개 이상의 정수들의 공통된 약수들 중 가장 큰 수

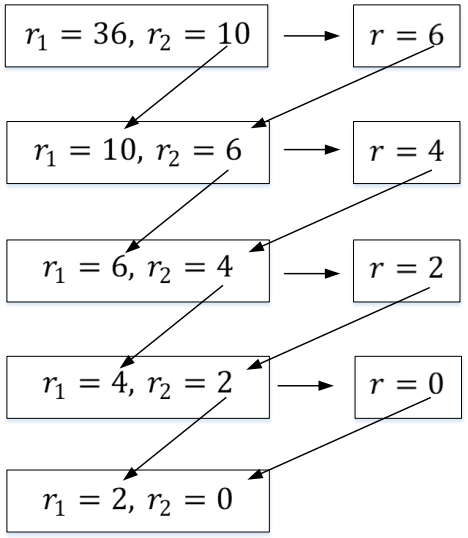


암호 수학

- 정수 연산(7/12)

*코드는 부록#1 참고

  - 가분성(Divisibility)(5/7)
    - 유클리드 알고리즘(Euclidean Algorithm)
      - 두 정수의 최대공약수를 찾아내는 알고리즘
      - 사실1. $gcd\ (a, 0) = a$
      - 사실2. $gcd\ (a, b) = gcd\ (b, r)$, 이때 r은 a를 b로 나눈 나머지
        - e.g., $gc\,d(36, 10) = gc\,d(10, 6) = gc\,d(6, 4) = gc\,d(4, 2) = gc\,d(2, 0) = 2$



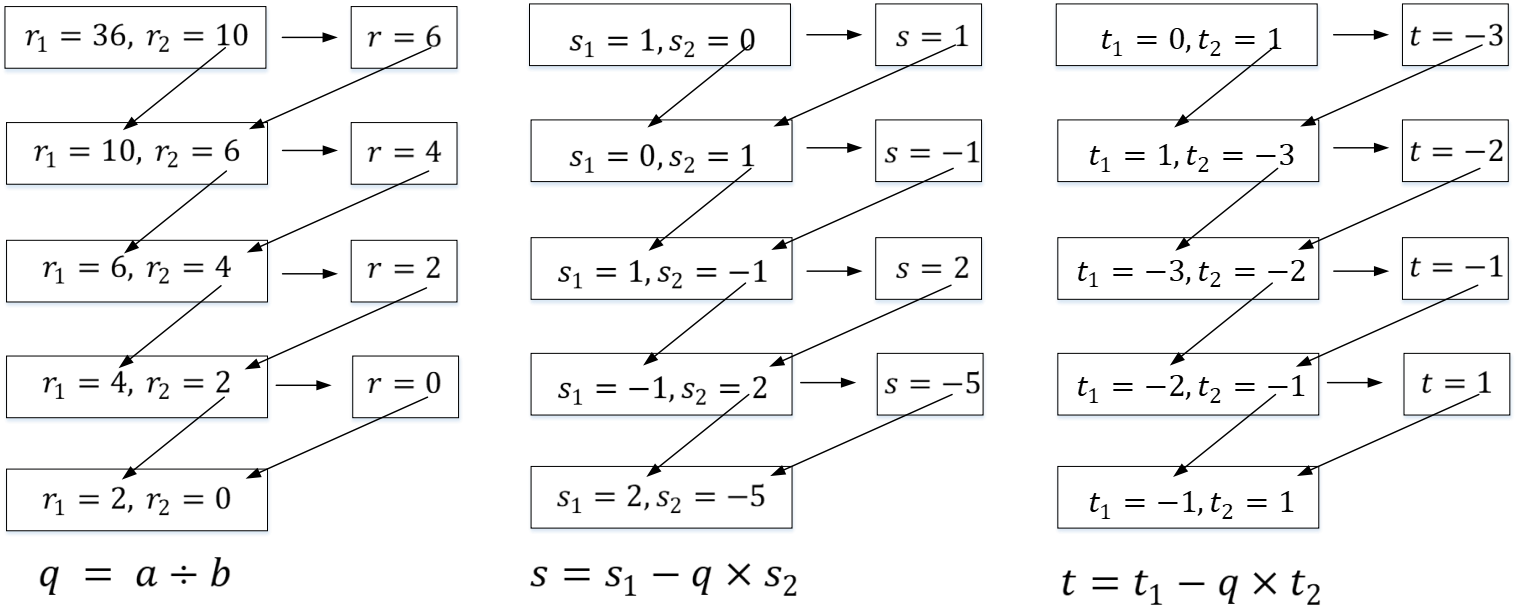
q	r_1	r_2	r
3	36	10	6
1	10	6	4
1	6	4	2
2	4	2	0
	2	0	

암호 수학

- 정수 연산(8/12)

*코드는 부록#2 참고

  - 가분성(Divisibility)(6/7)
    - 확장 유클리드 알고리즘(Extended Euclidean Algorithm)(1/2)
      - $s \times a + t \times b = gcd(a, b)$일때 만족하는 s, t를 $gcd(a, b)$와 동시에 계산하는 방법
      - 한 쌍 대신 세 쌍의 계산과 교환을 사용하는 것이 차이점
      - 변수s_1 과 s_2 는 각각 1과 0으로, t_1 과 t_2는 각각 0과 1로 초기화됨



- 정수 연산(9/12)
  - 가분성(Divisibility)(7/7)
    - 확장 유클리드 알고리즘(Extended Euclidean Algorithm)

$a = 161, b = 28$ 일 때 $gcd(a, b)$와 s, t를 계산해보자

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	

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

암호 수학

- 정수 연산(10/12)
  - 선형 디오판투스 방정식(Linear Diophantine Equations)(1/3)
    - $ax + by = c$의 꼴을 가짐(a, b, c는 정수)
    - x, y 의 정수해를 구하고자 하는 방정식
    - $d = gcd(a, b)$일 때
      - $d \mid c$인 경우, 방정식의 해가 무수히 많음
      - $d \nmid c$인 경우, 방정식의 해가 존재하지 않음

암호 수학

- 정수 연산(11/12)
  - 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 $a_1s + b_1t = 1$를 만족하는 s와 t값을 계산
      - 특수 해는 다음과 같이 계산됨
      - $x_0 = \left(\frac{c}{d}\right)s$
      - $y_0 = (\frac{c}{d})t$
    - 일반 해
      - 특수 해를 찾아낸 후 계산
      - $x = x_0 + k(\frac{b}{d}), (k$는 임의의 정수$)$
      - $y = y_0 - k\left(\frac{a}{d}\right), (k$는 임의의 정수$)$

- 정수 연산(12/12)
  - 선형 디오판투스 방정식(Linear Diophantine Equations)(3/3)

방정식$18x + 12y = 30$의 특수 해와 일반 해를 구해보자

- $d = gcd(18, 12) = 6$
- $6|30$이므로 방정식은 무한한 해를 가짐
- d로 양변 나누기 $3x + 2y = 5$
- 확장 유클리드 알고리즘을 활용한 s, t 계산
- $3s + 2t = 1$를 만족하는 $s = 1, t = -1$
- 특수 해 계산
  - $x_{0=}(\frac{30}{6}) \times 1 = 5,\ y_0 = (\frac{30}{6}) \times (-1) = -5$
- 일반 해 계산
  - $x = 5 + k \times (\frac{12}{6}),\ y = -5 - k \times (\frac{18}{6})$

암호 수학

- 모듈로 연산(Modular Arithmetic)(1/8)
  - 나눗셈 관계식 $a = q \times n + r$에서 나머지 r을 구하는 연산
    - $a \; mod \; n = r$
      - e.g., $27 \; mod \; 5 \; = \; 2$
    - $r < 0$라면 모듈로를 더하여 음이 아닌 수로 만들어야 함
      - e.g., $-7 \; mod \; 10 \; = \; -7$
        $r \; = \; -7$지만 모듈로 10을 더하여 $r \; = \; 3$이 되도록 함
      - $\therefore -7 \; mod \; 10 = 3$

암호 수학

- 모듈로 연산(Modular Arithmetic)(2/8)
  - 잉여류 Z_n(Residue Classes)
    - 정수 n에 대한 $\mathrm{Z}\ mod\ n$을 한 나머지의 집합
    - $Z_n = \{0, 1, 2, 3, \ldots, (n-1)\}$
      - e.g., $Z_5 = \{0, 1, 2, 3, 4\}$
        $Z_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$

암호 수학

- 모듈로 연산(Modular Arithmetic)(3/8)
  - 합동(Congruence)
    - 정수 n으로 나눈 나머지가 서로 같은 두 Z_n
    - 합동연산자 $(\equiv)$ 기호 사용
      - e.g., $2 \equiv 12(mod\ 10)$
  - 등식 연산자와의 차이점
    - 등식 연산자는 Z의 원소에서 Z의 원소로 대응하지만 합동 연산자는 Z의 원소에서 Z_n의 원소로 대응
    - 등식 연산자는 일대일대응이지만, 합동 연산자는 다대일
      - e.g., $2 \equiv 12(mod\ 10)$ 에서 $mod\ 10$은 공역 Z_{10}을 나타냄

암호 수학

- 모듈로 연산(Modular Arithmetic)(4/8)
  - Z_n에서의 연산
    - 3개의 2진 연산$(+, -, \times)$이 정의됨
  - 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)(5/8)
  - 역원(Inverse)(1/2)
    - 덧셈에 대한 역원(Additive Inverse)
      - $(a + b) \equiv 0 \; mod \; n$
        - b를 a의 $mod \; n$에서의 덧셈에 대한 역원이라고 함
      - Z_n에서 a의 덧셈에 대한 역원은 $b = n - a$로 계산됨
      - 모듈로 연산에서 각각의 수는 유일한 한 개의 덧셈에 대한 역원을 가짐
      - 역원은 자기 자신일 수도 있음
        - e.g., 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)(6/8)
  - 역원(Inverse)(2/2)
    - 곱셈에 대한 역원(Multiplicative Inverse)
      - $a \times b \equiv 1 \, mod \, n$
        - b를 a의 $mod \, n$에서의 곱셈 역원이라고 함
        - e.g.,모듈러 연산에서 곱셈에 대한 역원이 없을 가능성 존재
        - e.g.,만약 있다면, 그 정수와 해당하는 곱셈에 대한 역원의 곱은 $mod \, n$ 에서 1과 합동
      - a가 Z_n에서 곱셈에 대한 역원을 갖기 위한 필요충분조건은 $gcd(n, a) = 1$
        - 이 경우 a와n은 서로소
      - e.g., Z_{10}에서 곱셈에 대한 역원을 구하면 (1,1), (3,7), (9,9)가 있음

암호 수학

- 모듈로 연산(Modular Arithmetic)(7/8)
  - 덧셈과 곱셈에 대한 다른 집합
    - $Z_n{}^*$은 Z_n의 부분집합으로 Z_n에 속한 정수 중 곱셈에 대한 역원을 가진 원소의 집합
    - Z_n의 각 원소는 덧셈에 대한 역원을 갖지만 모든 원소가 곱셈에 대한 역원을 가지진 않음
    - $Z_n{}^*$의 각 원소는 곱셈에 대한 역원을 갖지만 모든 원소가 덧셈에 대한 역원을 가지진 않음
      - e.g., $Z_6 = \{0, 1, 2, 3, 4, 5\}$ $\quad Z_6{}^* = \{1, 5\}$
        $Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$ $\quad Z_7{}^* = \{1, 2, 3, 4, 5, 6\}$
        $Z_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ $\quad Z_{10}{}^* = \{1, 3, 7, 9\}$

암호 수학

- 모듈로 연산(Modular Arithmetic)(8/8)
  - 모듈로가 소수인 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
    - ${Z_p}^*$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)
  - $l \times m$개의 원소를 갖는 직사각형 배열
  - l은 행의 개수, m은 열의 개수
  - a_{ij}는 i 번째 행과 j번째 열에 위치

$$Matrix\ A: \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{1l} & \cdots & a_{lm} \end{bmatrix}$$

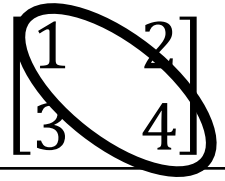
암호 수학

- 행렬(Matrices)(2/10)
  - 행 행렬(Row Matrix)
    - 행렬이 하나의 행 $(l = 1)$만 갖는 경우
    - e.g., $[1 \ 2 \ 3 \ 4]$
  - 열 행렬(Column Matrix)
    - 행렬이 하나의 열 $(m = 1)$만 갖는 경우
    - e.g., $\begin{bmatrix} 1 \\ 5 \\ 6 \end{bmatrix}$

암호 수학

- 행렬(Matrices)(3/10)
  - 정방행렬(Square Matrix)
    - 행의 개수와 열의 개수가 같은 행렬$(l = m)$
    - e.g., $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

*주대각선
정방행렬에서 $a_{11}, a_{22}, a_{33}, \cdots, a_{mm}$ 이 이루는 것



  - 항등 행렬(Identity Matrix)
    - $*$ 주대각선이 1이고 나머지는 0인 정방행렬
    - e.g., $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

암호 수학

- 행렬(Matrices)(4/10)
  - 행렬의 연산과 관계식(1/3)
    - 등식
      - 모든 i와 j에 대해 $a_{ij} = b_{ij}$ 이면 $A = B$
    - 덧셈과 뺄셈
      - 두 행렬의 행과 열이 개수가 같으면 연산 가능
      - 결과를 C행렬이라고 하면 각 원소는 A, B에 대응되는 원소의 합과 차
      - e.g., $c_{ij} = a_{ij} + b_{ij}$
      - $\begin{bmatrix} 2 & 3 \\ 4 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 5 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 2 & 2 \end{bmatrix}$ $\quad$ $\begin{bmatrix} 5 & 2 \\ -5 & 9 \end{bmatrix} = \begin{bmatrix} 7 & 9 \\ 0 & 12 \end{bmatrix} - \begin{bmatrix} 2 & 7 \\ 5 & 3 \end{bmatrix}$

암호 수학

- 행렬(Matrices)(5/10)
  - 행렬의 연산과 관계식(2/3)
    - 곱셈
      - 첫 번째 행렬의 열의 개수가 두 번째 행렬의 행의 개수와 같다면 다른 크기의 두 행렬을 곱할 수 있음
      - $l \times m$ 행렬과 $m \times p$ 행렬의 곱의 크기는 $l \times p$
      - 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., $\begin{bmatrix} 1 & 3 \\ 4 & 6 \end{bmatrix} \times \begin{bmatrix} 0 & 6 & 1 \\ -3 & 5 & -2 \end{bmatrix}$

$$= \begin{bmatrix} 1 \times 0 + 3 \times (-3) & 1 \times 6 + 3 \times 5 & 1 \times 1 + 3 \times (-2) \\ 4 \times 0 + 6 \times (-3) & 4 \times 6 + 6 \times 5 & 4 \times 1 + 6 \times (-2) \end{bmatrix} = \begin{bmatrix} -9 & 21 & -5 \\ -18 & 54 & -8 \end{bmatrix}$$

암호 수학

- 행렬(Matrices)(6/10)
  - 행렬의 연산과 관계식(3/3)
    - 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., $\begin{bmatrix} 15 & 6 \\ 9 & 3 \end{bmatrix} = 3 \times \begin{bmatrix} 5 & 2 \\ 3 & 1 \end{bmatrix}$

암호 수학

- 행렬(Matrices)(7/10)
  - 행렬식(Determinant)
    - 크기가 $m \times m$인 정방 행렬 A의 행렬식은 $\det(A)$로 표기
    - 행렬식은 정방 행렬에서만 정의
    - $m = 1$이면 $\det(A) = a_{11}$
    - $m > 1$이면 $\det(A) = \sum_{j=1,\ldots,m}(-1)^{i+j} a_{ij} \times det(A_{ij})$

    - A_{ij}는 A에서 i열과 j행을 제거한 뒤 얻어진 행렬
    - e.g., $det\begin{bmatrix}5 & 2\\ 3 & 4\end{bmatrix} = (-1)^{1+1} \times 5 \times det[\,4\,] + (-1)^{1+2} \times 2 \times det[\,3\,]$

      $\longrightarrow 5 \times 4 - 2 \times 3 = 14$

암호 수학

- 행렬(Matrices)(8/10)
  - 역행렬(Inverse Matrices)
    - 덧셈에 대한 역행렬(Additive Inverse Matrices)
      - 특정 행렬 A에 대하여 $A + B = 0$ 을 만족하는 행렬 B
      - 모든 i 와 j에 대해 $b_{ij} = -a_{ij}$
      - 행렬 A의 덧셈에 대한 역행렬은 $-A$로 정의
      - e.g., $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} -1 & -2 \\ -3 & -4 \end{bmatrix}$

$$\therefore B = -A$$

암호 수학

- 행렬(Matrices)(8/10)
  - 역행렬(Inverse Matrices)
    - 곱셈에 대한 역행렬(Multiplicated Inverse Matrices)
      - 특정 정방행렬 A에 대해, 행렬 A와 곱했을 때 항등행렬이 되는 행렬
      - A^{-1}로 표기
      - $\mathrm{Det}(A) \neq 0$ 일 경우에만 역행렬을 가짐
      - 곱셈에 대한 역행렬은 정방 행렬에서만 정의
      - e.g., $\begin{bmatrix} 2 & -1 & 0 \\ 1 & 0 & -1 \\ 1 & 0 & 1 \end{bmatrix}$ 의 역행렬은 $-\frac{1}{2}\begin{bmatrix} 0 & 1 & 1 \\ -2 & 2 & 2 \\ 0 & -1 & 1 \end{bmatrix}$

암호 수학

- 행렬(Matrices)(9/10)
  - 잉여 행렬(Residue Matrices)(1/2)
    - 모든 행렬의 원소가 Z_n에 속하는 행렬
    - 모듈로 연산으로 행해지는 것을 제외하고 행렬에서와 동일하게 연산됨
    - 행렬식이 Z_n에서 곱셈에 대한 역원을 가질 경우 곱셈에 대한 역행렬을 가짐
      - $gcd(det(A), n) = 1$ 이면 곱셈에 대한 역행렬을 가짐
      - 잉여 행렬과 곱셈에 대한 역행렬

e.g., $A = \begin{bmatrix} 3 & 5 & 7 & 2 \\ 1 & 4 & 7 & 2 \\ 6 & 3 & 9 & 17 \\ 13 & 5 & 4 & 16 \end{bmatrix}$ $A^{-1} = \begin{bmatrix} 15 & 21 & 0 & 15 \\ 23 & 9 & 0 & 22 \\ 15 & 16 & 18 & 22 \\ 24 & 7 & 15 & 3 \end{bmatrix}$

$det(A) = 21$ $det(A^{-1}) = 5$

암호 수학

- 행렬(Matrices)(10/10)
  - 잉여 행렬(Residue Matrices)(2/2)
    - 합동
      - 두 행렬이 열과 행의 개수가 같고 모든 대응되는 원소가 모듈로 n에 대해서 합동인 경우
      - $A \equiv B(mod\ n)$로 표기
      - e.g., $A = \begin{bmatrix} 7 & 11 \\ 3 & 4 \end{bmatrix},\ B = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$ 이면 각 행렬에 모듈로 5를 하면

        $7 \equiv 2(mod\ 5), 11 \equiv 1(mod\ 5),\ 3 \equiv 3(mod\ 5),\ 4 \equiv 4(mod\ 5)$

        ∴ 두 행렬은 모듈로 5에서 합

암호 수학

- 선형 합동(Linear Congruences)(1/5)
  - 일변수 선형 방정식(Single-variable linear equations)(1/2)
    - $ax \equiv b(mod\ n)$형태의 방정식의 해를 구함
    - $\gcd(a, n) = d$라고 가정할 때
      - $d \nmid b$ 이면 해가 존재하지 않음
      - $d \mid b$ 이면 d개의 해가 존재
        - 해를 구하는 방법

1. 모듈로를 포함하여 방정식의 양변을 d로 나눔
2. 특수해 x_0를 구하기 위해 약분한 방정식의 양변에 a의 곱셈에 대한 역원을 곱함
3. 일반해는 $k = 0, 1, \cdots, (d-1)$에 대하여 $x = x_0 + k(n/d)$

암호 수학

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

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

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

        $x_1 = x_0 + 1 \times (18/2) = 15$

암호 수학

- 선형 합동(Linear Congruences)(3/5)
  - 일차 연립 방정식(1/3)
    - 동일한 모듈로에 대한 일차 방정식들의 집합에서 변수의 계수로 구성된 행렬의 역행렬이 존재하면 주어진 방정식들의 공통 해를 구할 수 있음

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

a. Equations

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

b. Interpretation

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

c. Solution

암호 수학

- 선형 합동(Linear Congruences)(4/5)
  - 일차 연립 방정식(2/3)
    - 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

1. 연립 방정식을 행렬 A로 바꿈 $\mathrm{A} = \begin{bmatrix} 3 & 5 & 7 \\ 1 & 4 & 13 \\ 2 & 7 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 4 \end{bmatrix}$
2. 행렬 A의 역행렬 구하기

행렬 $\mathrm{A}^{-1} = \frac{1}{129} \begin{bmatrix} 79 & -34 & -37 \\ -23 & 5 & 32 \\ 1 & 11 & -7 \end{bmatrix}$

- 선형 합동(Linear Congruences)(5/5)
  - 일차 연립 방정식(3/3)

3. 양변에 각각 행렬 A의 곱셈에 대한 역행렬 A^{-1}을 곱함

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \frac{1}{129}\begin{bmatrix}79 & -34 & -37\\ -23 & 5 & 32\\ 1 & 11 & -7\end{bmatrix}\begin{bmatrix}3\\ 5\\ 4\end{bmatrix}$$

4. 모듈로 연산

$$\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix}15(mod\ 16)\\ 4(mod\ 16)\\ 14(mod\ 16)\end{bmatrix}$$

Thanks!

민 윤 홍(muh4316@gmail.com)

부록#1

- Python 코드
  - 유클리드 알고리즘

```
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

a, b = map(int, input().split())

print(gcd(a, b))
```

부록#2

- Python 코드
  - 확장 유클리드 알고리즘

```
def ex_gcd(a, b):
    if b == 0:
        return a, 1, 0  # gcd, x, y
    else:
        gcd, x1, y1 = ex_gcd(b, a % b)
        x = y1
        y = x1 - (a // b) * y1
        return gcd, x, y

a, b = map(int, input().split())

gcd, x, y = ex_gcd(a, b)
print(f"gcd({a}, {b}) = {gcd}")
print(f"x = {x}, y = {y}")
```