

컴퓨터 보안 출석수업 강의 (2024년)

방송통신대 컴퓨터과학과

출처 : 김진욱교수님 강의자료

1강. 컴퓨터 보안의 개요

목차	1	컴퓨터 보안의 개념
	2	정보보호의 목표
	3	정보화 환경과 역기능
	4	컴퓨터 보안의 역사

1. 컴퓨터 보안의 개념

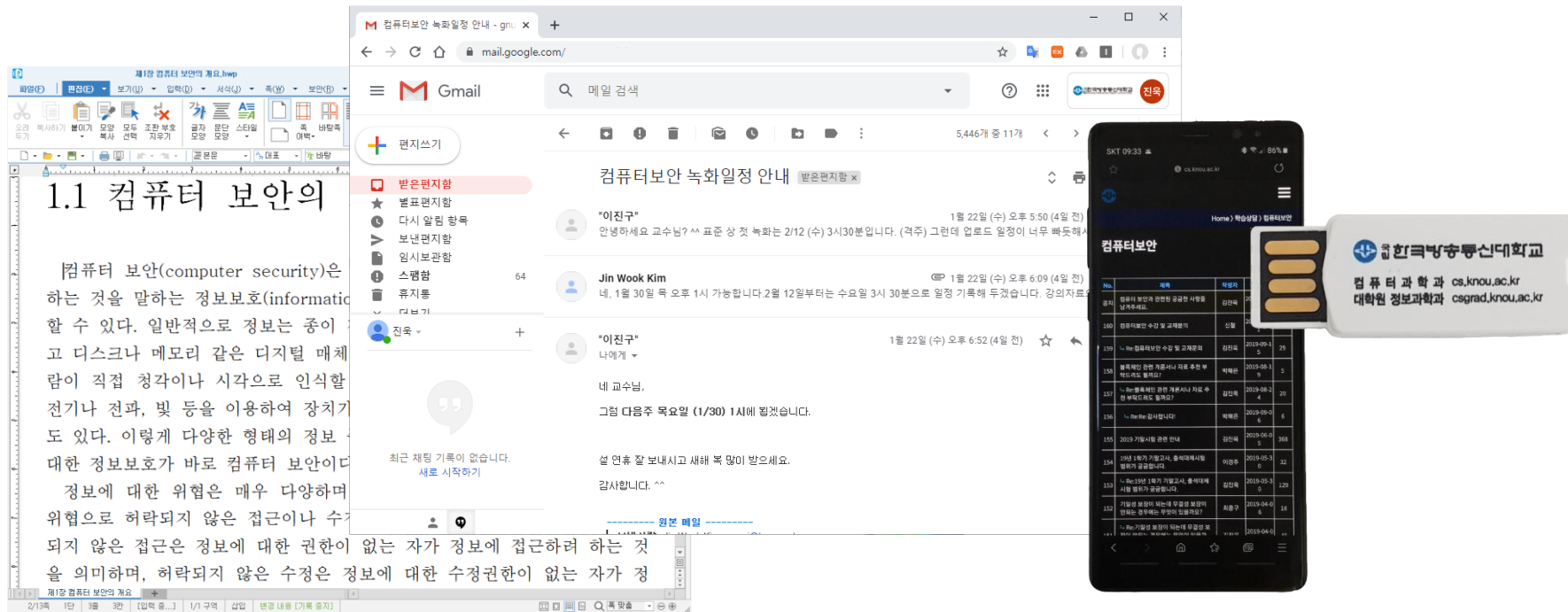
정보보호의 개념



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

컴퓨터 보안의 개념

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



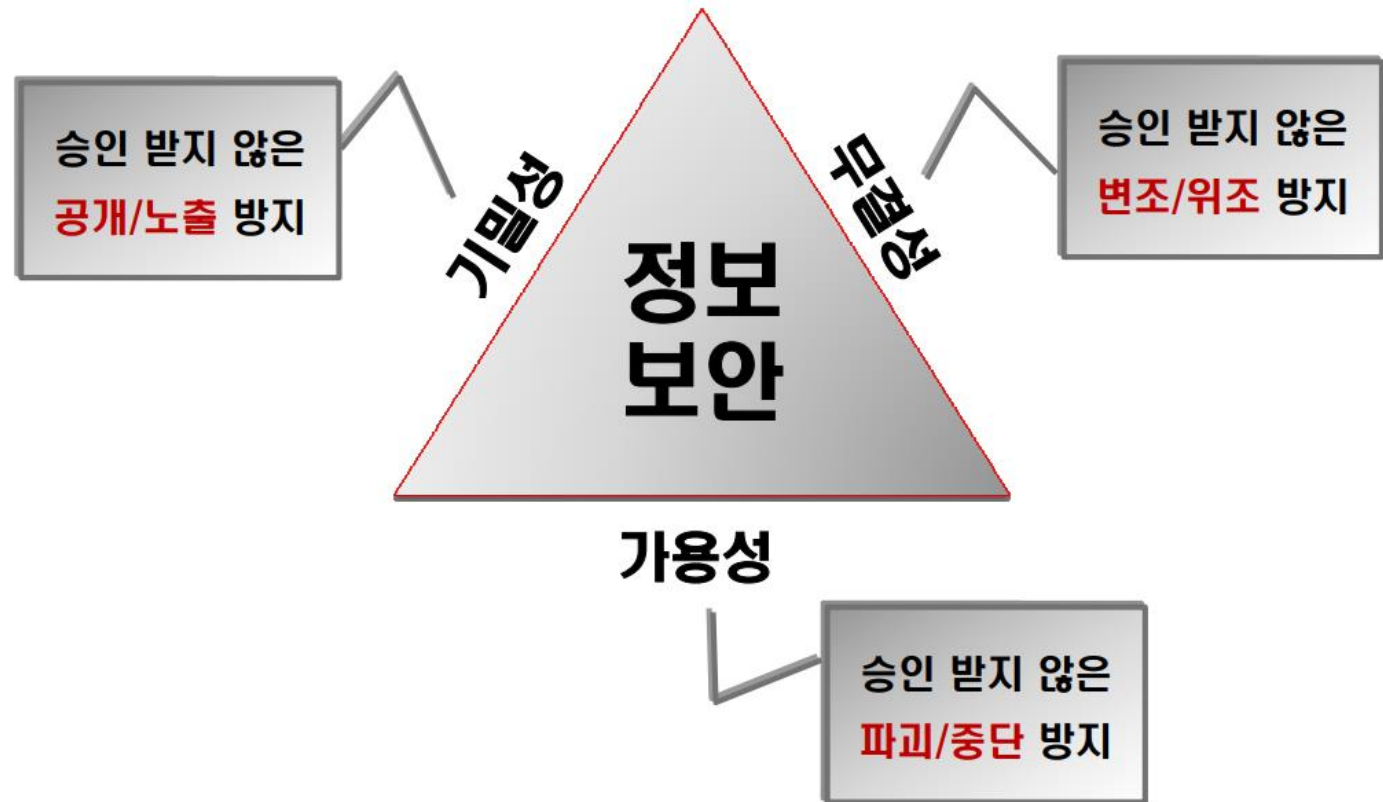
- 컴퓨팅 환경에 저장되거나 처리되는 정보를
다양한 위협으로부터 보호하기 위한 정책 및 기법

2. 정보보호의 목표

정보보호의 목표

■ 정보보호의 핵심목표

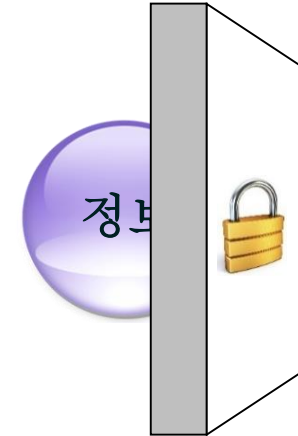
- 기밀성(Confidentiality)
- 무결성(Integrity)
- 가용성(Availability)



정보보호의 핵심목표

■ 기밀성

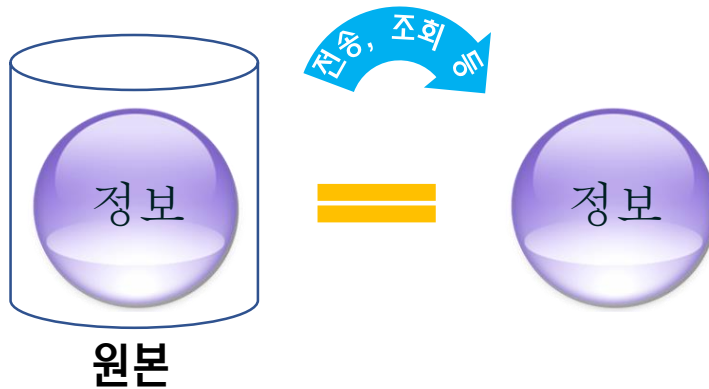
- 허락되지 않은 자가 정보의 내용을 알 수 없도록 하는 것
- 예: 은행에서 고객의 개인정보나 계좌정보 같은 기밀정보가 제3자에게 알려지는 것을 방지하기 위해 이를 보호
- 기밀성을 지키는 방법
 - ➡ 허락되지 않은 자가 정보에 접근을 아예 못하도록 함
 - ➡ 정보에 접근하더라도 무의미한 내용만 보이도록 함
 - ➡ 기밀성 보장을 위해서 개인정보는 암호화 하는 것이 중요하다.



정보보호의 핵심목표

■ 무결성

- 허락되지 않은 자가 정보를 함부로 수정할 수 없도록 하는 것

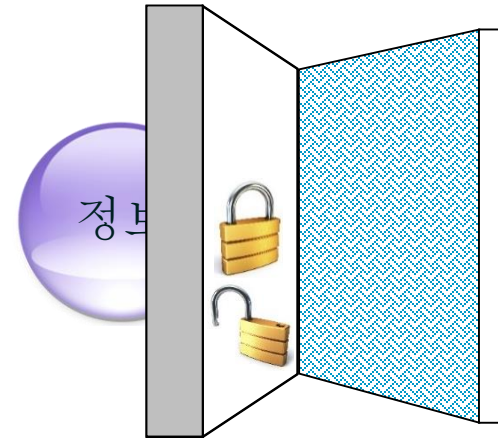


- 예: DB 내 고객의 개인정보가 임의로 수정되지 않도록 보호, 고객 본인이 조회할 때 DB에서 고객까지 전달과정에서 위변조되지 않도록 보호
- 만약 허락되지 않은 자에 의한 수정이 발생했다면 이를 확인할 수 있는 것

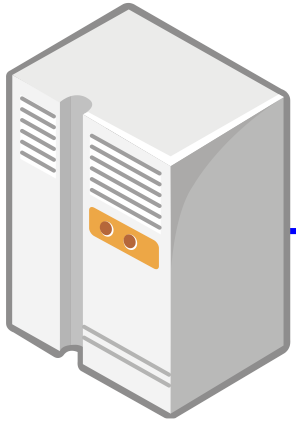
정보보호의 핵심목표

■ 가용성

- 허락된 자가 정보에 접근하고자 할 때 이것이 방해 받지 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)



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

정보보호의 목표

■ 정보보호의 핵심목표

- 기밀성(Confidentiality)
- 무결성(Integrity)
- 가용성(Availability)



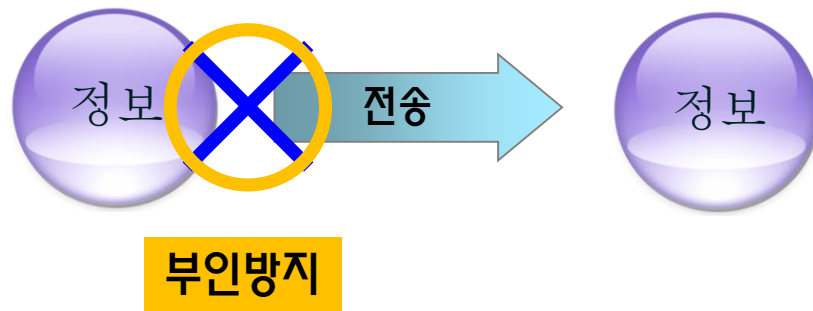
■ 그 외의 목표

- 부인방지(non-repudiation)
- 인증(authentication)
- 접근제어(access control) 등

정보보호의 목표

■ 부인방지

- 정보에 관여한 자가 이를 부인하지 못하도록 하는 것



- 발신 부인방지
 - ➔ 정보를 보낸 사람이 나중에 정보를 보냈다는 것을 부인하지 못하도록 함
- 수신 부인방지
 - ➔ 정보를 받은 사람이 나중에 이를 부인하지 못하도록 하는 것

정보보호의 목표

■ 인증

- 어떤 실체가 정말 주장하는 실체가 맞는지 확인할 수 있고 신뢰할 수 있는 것



- 실체: 정보 자체, 정보를 이용하는 사용자 등

정보보호의 목표

■ 접근제어

- 정보에 대한 허락된 접근만 허용하고 그 외의 접근은 허용하지 않는 것
- 즉, 접근권한이 있는 자와 없는 자를 구분하여 제어
- 접근권한은 정보에 따라, 사용자에 따라 다양하게 부여될 수 있음

3. 정보화 환경과 역기능

정보화 환경과 역기능

■ 정보화 사회의 선진화

- 과거에는 정보의 전파가 굉장히 느렸으나 통신회선 설치 이후 빨라짐
- 인터넷을 통한 지구 반대편에서 일어나는 일도 실시간으로 알 수 있음

■ 정보화 사회의 역기능도 증가

- 악성 댓글, 스팸 메일, 개인정보 유출, 금전적인 목적을 대상으로 하는 피싱이나 파밍, 스미싱에 따른 개인적인 피해 증가
- 불건전 정보유통, 개인 사생활 침해 등과 같은 부작용
- 심각한 사회문제로 대두

정보화 환경과 역기능

■ 새로운 수법의 지속적인 등장

- 과거 이메일을 이용하던 피싱은 보이스 피싱과 스미싱 등으로 다양화
- 랜섬웨어처럼 안전한 암호 알고리즘을 악용하여 금전을 요구하는 수법도 등장

■ 주요 기반시설 위협에 따른 국가안보적인 측면의 위협

- 주요 기반시설이 점차 정보통신 네트워크에 의해 관리 및 통제됨
- 사이버 공격의 주요 목표가 됨

4. 컴퓨터 보안의 역사

컴퓨터 보안의 역사 – 컴퓨터의 등장

■ 앨런 튜링(1912년 ~ 1954년)

- 컴퓨터의 이론적인 개념을 정립
- 제2차 세계대전(1940~1945) 당시 암호분석가로 활동
- 독일군이 사용했던 에니그마의 암호문을 해독할 수 있는 콜로서스(Colossus)를 만들어 제2차 세계대전을 2년 빨리 끝냄



< 에니그마 >



< 콜로서스 >

컴퓨터 보안의 역사 – 컴퓨터의 등장

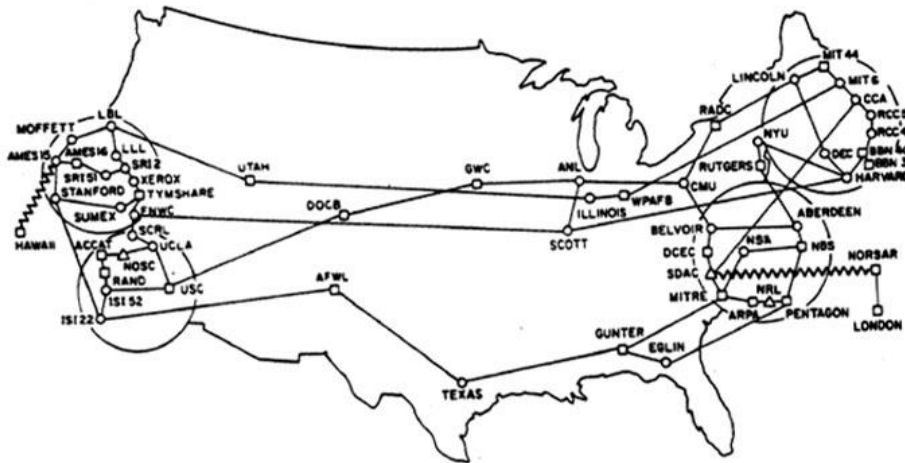
■ 1950년대와 1960년대

- 메인프레임 형태의 컴퓨터들이 개발되어 기관이나 학교 등에서 이용
- MIT의 TMRC(Tech Model Railroad Club) 학생들이 해커(hacker)라는 용어를 처음 사용
 - ➡ 해커: 컴퓨터의 세세한 부분까지 적극적으로 탐구하고 활용성을 확대하기 위해 연구하는 것을 즐기는 사람
- ★ 크래커(cracker): 컴퓨터 보안에 위협을 가하는 악의적인 사람

컴퓨터 보안의 역사 – 인터넷의 모체 등장

■ ARPANET(Advanced Research Project Agency Network)

- 1960년대 후반, 미국 국방부에서는 각 지역에 위치한 기관들의 컴퓨터를 연결하는 네트워크를 개발
- 1970년대를 거치며 상당히 복잡한 형태로 발전



- 이를 바탕으로 1980년대 인터넷이 등장

컴퓨터 보안의 역사 – 개인용 컴퓨터의 등장

■ 1970년대

- 애플컴퓨터에서 개인용 컴퓨터(PC) 판매 시작
- 연구자 뿐만 아니라 일반인들도 컴퓨터를 접할 수 있게 됨

■ 1980년대

- IBM에서 저가의 PC 판매를 시작하며 컴퓨터가 본격적으로 대중화
- 인터넷 표준 통신 프로토콜인 TCP/IP가 개발되며
누구나 PC와 인터넷으로 다양한 정보를 접할 수 있는 환경이 만들어짐
- 악의적인 사람들도 역시 컴퓨터와 인터넷을 사용하게 되어
컴퓨터 보안의 필요성 증가

컴퓨터 보안의 역사 – 다양한 위협 발생

■ 1980년대

- 모리스 웜(Morris worm): 인터넷으로 연결된 수천 대의 UNIX 컴퓨터를 감염
- 이를 계기로 침해대응센터인 CERT 만들어짐

■ 1990년대

- 시티뱅크 시스템에 침입하여 자금 탈취
- 여러 회사 시스템에 침입하여 각종 정보 탈취
- 정부 시스템에 침입하여 걸프전 정보 탈취

컴퓨터 보안의 역사 – 다양한 위협 발생

■ 2000년대

- 2000년 초, 야후, CNN, 아마존 등 가장 소통량이 많은 몇 개의 사이트에 분산 서비스 거부(DDoS, Distributed Denial of Service) 공격 발생
- 네트워크 상의 취약한 서버를 찾아 미리 감염시킨 후 이 서버들이 정해진 시간에 목표 사이트에 수많은 패킷을 전송하도록 함

■ 2010년대

- 랜섬웨어가 새롭게 유행
- 컴퓨터에 저장되어 있는 문서나 그림 등을 암호화하여 사용자가 사용할 수 없게 만듦
- 암호를 풀기 위해서는 비트코인 등으로 송금하도록 유도

2강. 암호의 개념

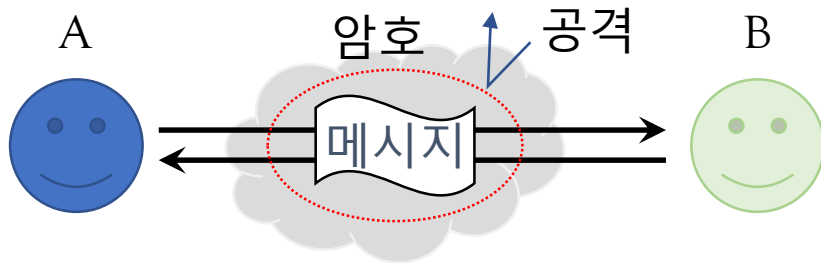
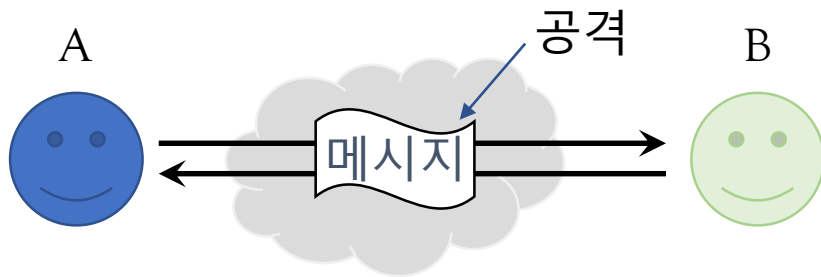
목차

- 1 암호의 정의
- 2 암호의 역사
- 3 대칭키 암호의 개념
- 4 공개키 암호의 개념

1. 암호의 정의

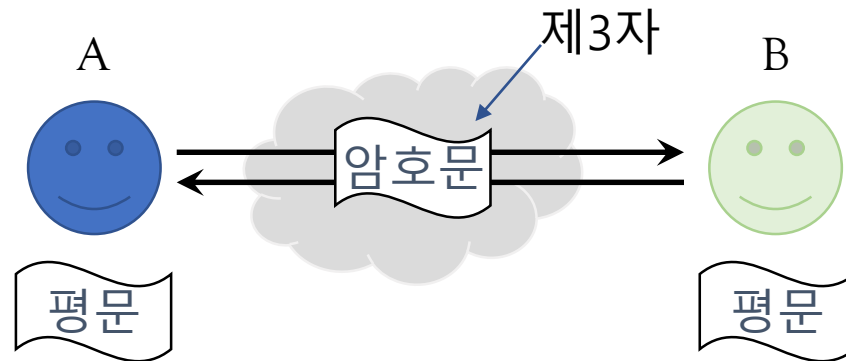
암호의 정의

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



암호의 정의

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



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

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

2. 암호의 역사

암호의 역사

■ 개요

- 처음에는 군사와 정치적인 목적으로 주로 사용
- 컴퓨터와 통신이 결합됨에 따라 불법 사용자의 봉쇄 또는 데이터의 위조 및 변조를 막는 수단으로 이용
- 최근 인터넷 뱅킹에 사용되는 인증서, 보안 키패드, 소프트웨어의 시리얼 키뿐만 아니라 전자투표에도 점차 널리 이용

암호의 역사 – 고대 암호

■ 스테가노그래피

- 실제로 전달하고자 하는 정보 자체를 숨기는 것
- 최초의 암호는 BC 480년, 스파르타에서 추방되어 페르시아에 살던 데마라토스가 페르시아의 침략계획 소식을 나무판에 조각 후 밀랍을 발라 스파르타에 보낸 것
- 하지만 엄밀히 구분하면 최초의 암호로 보기 힘들
 - ➔ 일반적인 암호의 요건: 제3자에게 암호 알고리즘을 알려주더라도 제3자가 키를 모르면 암호를 풀 수 없는 것을 가정

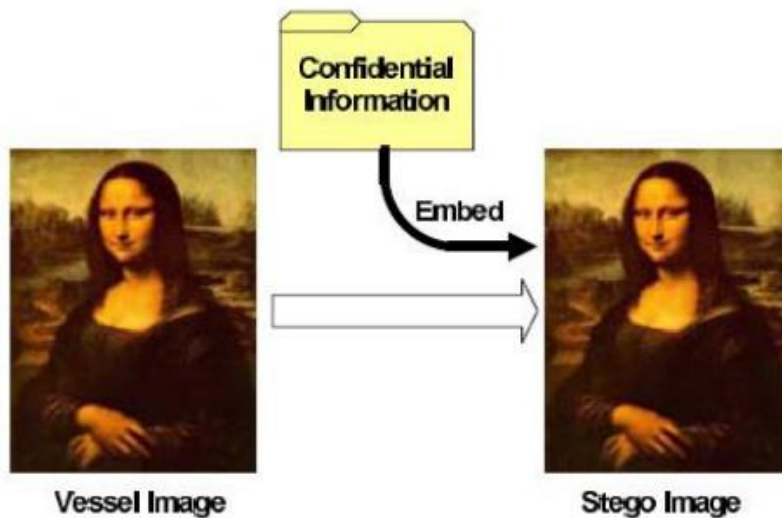


■ 두 가지 암호 방식: 전치법, 치환법

암호의 역사 – 현재의 스테가노그래피

스테가노그래피

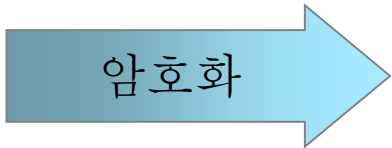
- 스테가노그래피
[Steganography]란
메시지를 이미지에 숨기는
기술이다
- 복잡한 이미지에서 사소한
차이는 식별해 내지 못하는
인간의 한계를 이용한다.
- 저작권보호에 사용되는
디지털 워터마크가
대표적이다.



© Copyright IBM Corporation 2011

암호의 역사 – 고대 암호

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

암호 알고리즘  암호화 호암고알즘리

- 가장 단순한 방식: 두 문자씩 앞뒤로 섞는 방법

암호화

암	호	알	고	리	즘
↘	↘	↘			
호	암	고	알	즘	리

복호화

호	암	고	알	즘	리
↘	↘	↘			
암	호	알	고	리	즘

암호의 역사 – 고대 암호

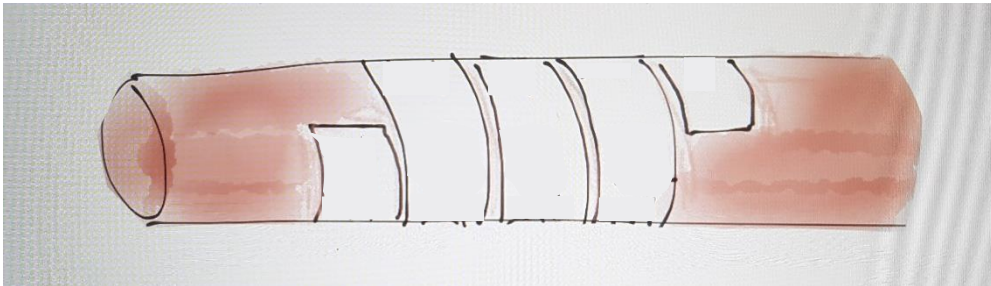
■ 전치법(permutation cipher 혹은 transposition cipher)

- 평문에 있는 문자들의 순서를 바꿈으로써 암호화하는 기법

- 스파르타의 봉 암호

키: 봉의 굵기

컴퓨터보안재미있는과목입니다~!



컴안는니퓨재과다터미목~보있입!

암호의 역사 – 고대 암호

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

39. 스파르타의 봉 암호로 평문 '12345678'을 암호화할 때 만들어

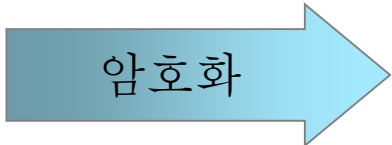
질 수 없는 암호문은? (4점)

- ① 13572468
- ② 12348765
- ③ 14725836
- ④ 15263748

암호의 역사 – 고대 암호

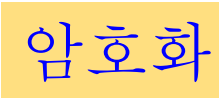
■ 치환법(substitution cipher)


- 평문의 문자들을 다른 문자로 치환함으로써 암호화하는 기법

암호 알고리즘  암호화 C + - 가 3 Z

- 치환 규칙에 따라 암호화 및 복호화

평문 문자	고	리	알	암	즘	호
암호문 문자	가	3	-	C	Z	+

 암호 알고리즘
↓ ↓ ↓ ↓ ↓ ↓
C + - 가 3 Z

 복호화
C + - 가 3 Z
↓ ↓ ↓ ↓ ↓ ↓
암호 알고리즘

암호의 역사 – 고대 암호

■ 치환법(substitution cipher)

- 평문의 문자들을 다른 문자로 치환함으로써 암호화하는 기법
- 시저 암호: 각 문자를 알파벳 순서로 세 번째 뒤 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

CAESAR
↓ ↓ ↓ ↓ ↓ ↓
FDHVDU

복호화

FDHVDU
↓ ↓ ↓ ↓ ↓ ↓
CAESAR

암호의 역사 – 고대 암호

■ 치환법(substitution cipher)

- 평문의 문자들을 다른 문자로 치환함으로써 암호화하는 기법
- 시프트 암호: 각 문자를 알파벳 순서로 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	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E

$k = 5$

암호화

CAESAR

↓ ↓ ↓ ↓ ↓ ↓

HFJ XFW

복호화

HFJ XFW

↓ ↓ ↓ ↓ ↓ ↓

CAESAR

암호의 역사 – 현대 암호

■ 표준 암호 알고리즘의 등장

- 컴퓨터가 점차 발전하면서 데이터 보호에 대한 필요성도 증가
- 1977년 미국 NBS(현재의 NIST)에서 표준 암호 알고리즘인 DES 공표
→ DES(Data Encryption Standard)는 대표적인 대칭키 암호 알고리즘
- 2001년 새로운 표준 암호 알고리즘인 AES가 공표될 때까지 널리 이용됨

암호의 역사 – 현대 암호

■ 공개키 암호 알고리즘의 등장

- 1976년 디피(Diffie)와 헬만(Hellman)이 공개키 암호의 개념을 제시
 - ➔ 공개키 암호: 암호화와 복호화에 서로 다른 키를 사용



디피

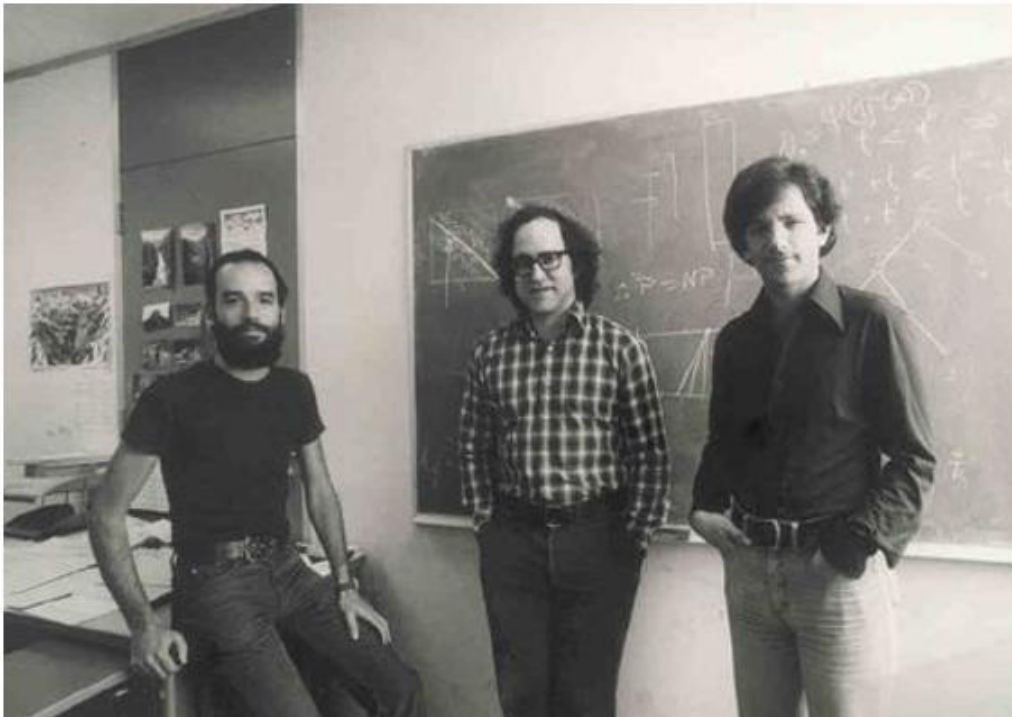


헬만

암호의 역사 – 현대 암호

■ 공개키 암호 알고리즘의 등장

- 1978년 리베스트(Rivest), 샤미르(Shamir), 애들먼(Adleman)이 RSA 공개키 암호 알고리즘 개발
 - RSA: 소인수분해 문제에 기반을 둔 대표적인 공개키 암호 알고리즘

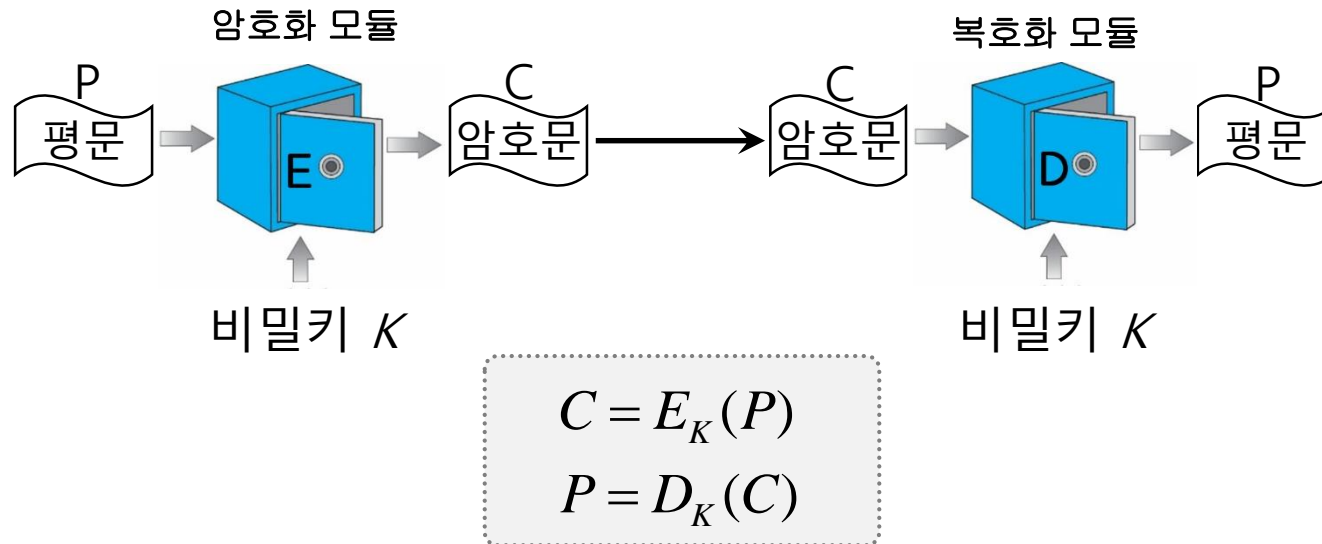


왼쪽부터 리베스트(R), 샤미르(S), 애들먼(A)

3. 대칭키 암호의 개념

대칭키 암호

- 암호화와 복호화에 하나의 같은 비밀키를 사용하는 암호 방식

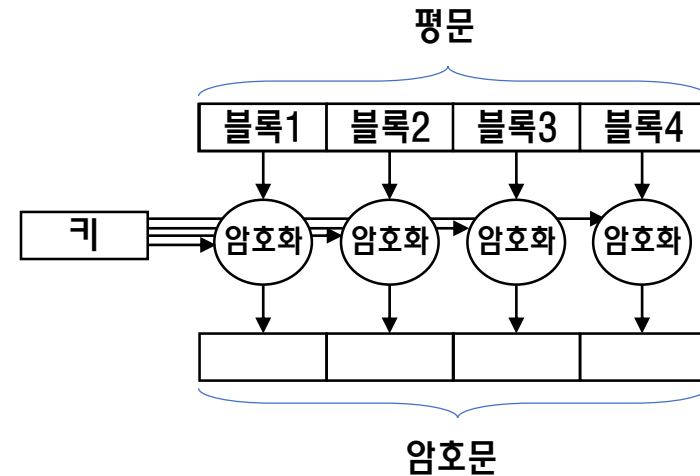


- 장점: 암호화와 복호화 속도가 빠름
- 대표적인 알고리즘: DES, AES, IDEA 등
- 다양한 이름 : 비밀키 암호, 단일키 암호, 관용 암호

대칭키 암호의 분류

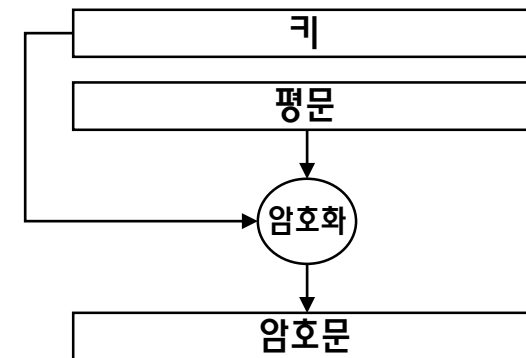
■ 블록 암호

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



■ 스트림 암호

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



대칭키 암호의 분류

■ XOR 활용 간단한 예시 (암호화)

- 메시지(평문) 1 0 1 1 0 1 0 1
 - 암호화키 1 1 0 0 1 1 0 0
-
- XOR 0 1 1 1 1 0 0 1 (암호문)

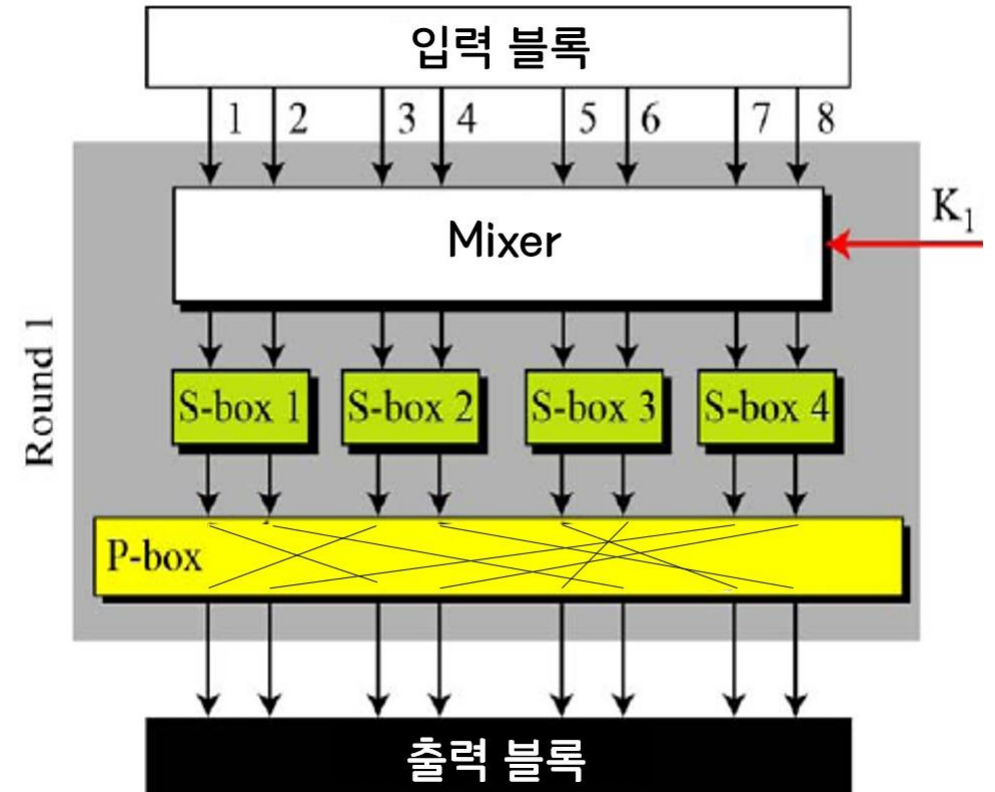
■ XOR 활용 간단한 예시 (복호화)

- 암호문 0 1 1 1 1 0 0 1
 - 복호화키 1 1 0 0 1 1 0 0
-
- XOR 1 0 1 1 0 1 0 1 (평문)

블록 암호 알고리즘의 구조

■ SPN(Substitution Permutation Network) 구조

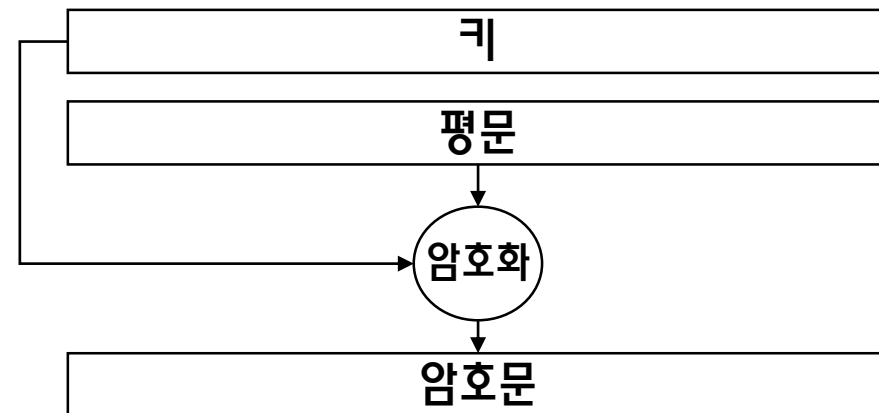
- 하나의 입력 블록을 여러 개의 소블록으로 나눈 후 라운드를 진행
- 각 라운드의 출력 블록이 다음 라운드의 입력 블록이 됨
- 더 많은 병렬성 제공
- AES, ARIA 등 최근의 블록 암호에 사용됨



스트림 암호의 개념

■ 스트림 암호

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



■ 키 스트림

- 임의의 길이의 평문에 대해 항상 생성 가능
- 규칙성이 없어 예측이 불가능한 랜덤 수열이 가장 안전
- 의사 랜덤(pseudorandom) 수열
 - ➔ 자동화된 생성이 가능하면서도 예측이 어려운 수열

대칭키 암호 알고리즘 - DES

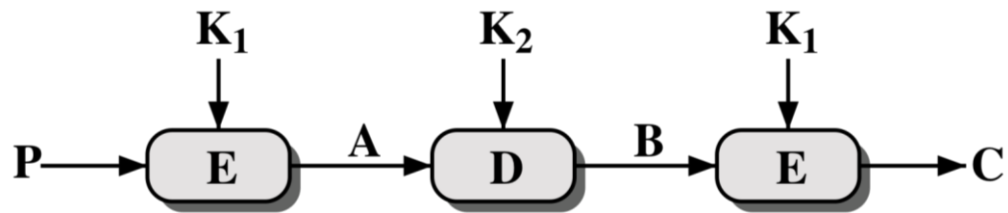
■ DES (Data Encryption Standard)

- 1977년 미국 NBS(현재의 NIST)에서 공표
- 2001년 새로운 표준 암호 알고리즘인 AES가 공표될 때까지 널리 이용됨
- 블록 암호 알고리즘
 - ➔ 평문 블록: 64 bits
 - ➔ 키: 56 bits
 - ➔ 암호 블록: 64 bits
 - ➔ 라운드 수: 16
 - ➔ 라운드 키: 48 bits

대칭키 암호 알고리즘 - TDEA

■ TDEA (Triple Data Encryption Algorithm)

- 3DES(triple DES): DES를 3회 반복하여 사용



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

대칭키 암호 알고리즘 - AES

■ AES (Advanced Encryption Standard)

- 2001년 미국 NIST에서 공표
- DES를 대신하는 새로운 표준 암호 알고리즘
- 공모를 통해 Rijndael을 AES로 선정
- 블록 암호 알고리즘
 - ➔ 평문 블록: 128 bits
 - ➔ 키: 128 bits, 192 bits, 256 bits 중 택일
 - ➔ 암호 블록: 128 bits

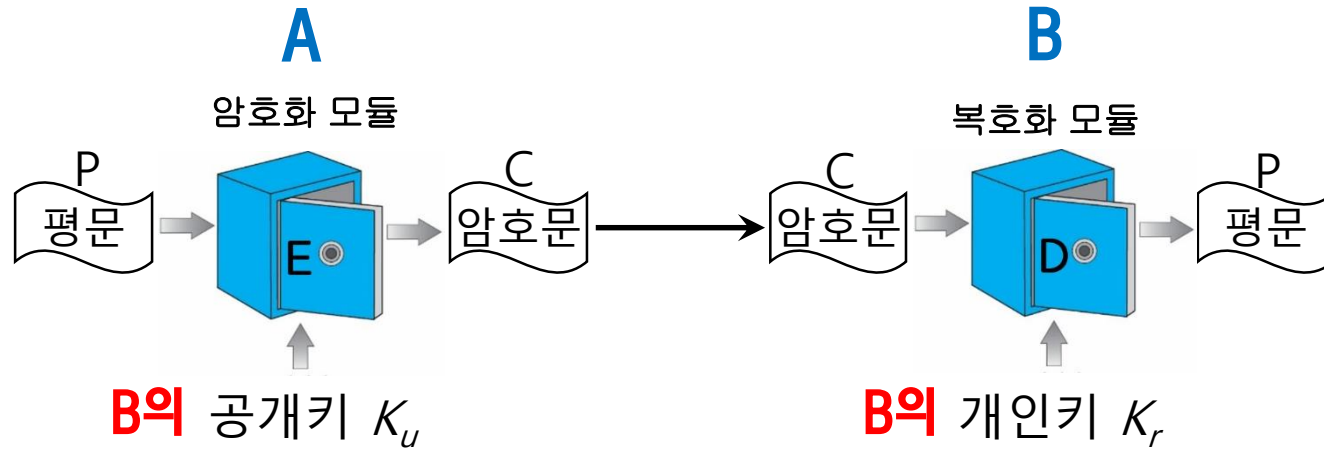
대칭키 암호 알고리즘 - AES

- NIST는 1997년 암호화 알고리즘을 공모하였으며 조건은 앞으로 30년 정도 사용할 수 있는 안정성, 128비트 암호화 블록, 다양한 키의 길이 요구
- 1997년 9월부터 1998년 4월까지 알고리즘 공모를 받았으며 12개국에서 총 15개의 알고리즘이 제안됨
- 1998년 8월까지 1차 예선 평가가 이루어져 구현상의 문제점을 검증하였고, 1999년 3월까지 효율성 평가를 거쳐 미국의 MARS, RC6, Twofish, 벨기에의 Rijndael, 영국/이스라엘/덴마크의 합작인 Serpent가 결선에 오름
- 결선에서는 공개적으로 암호학적 안전성 분석을 하였는데 리즈멘(Rijmen)과 대먼(Daemen)의 Rijndael 알고리즘이 2000년 10월 최종 AES(Advanced Encryption Standard)로 선정 (공표는 2001년)

4. 공개키 암호의 개념

공개키 암호

- 암호화와 복호화에 두 개의 서로 다른 키를 사용하는 암호 방식



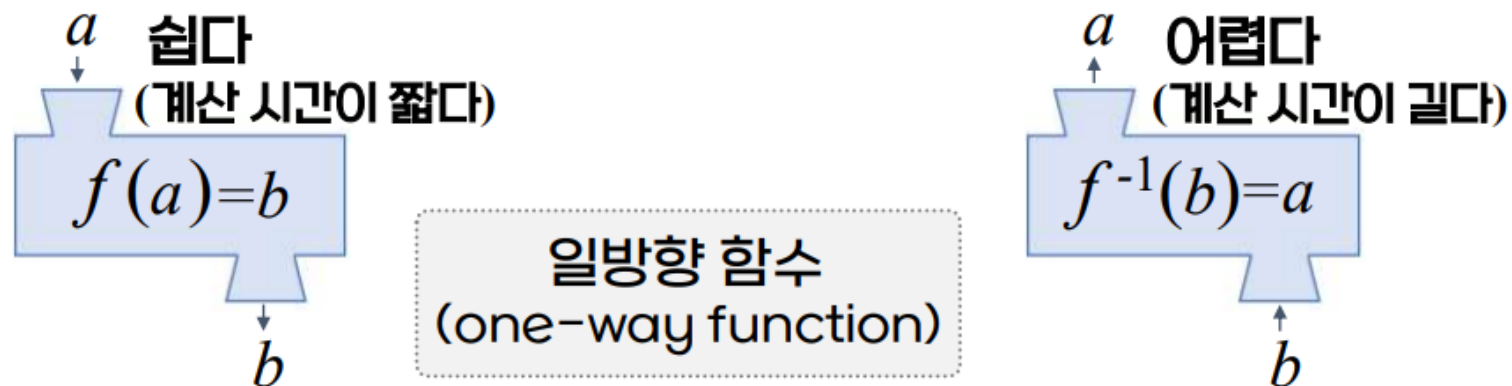
$$C = E_{K_u}(P)$$

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

- 공개키와 개인키는 언제나 한 쌍으로 존재
 - 공개키: 누구나 공개키를 이용할 수 있도록 공개함
 - 개인키: 오직 자신만 이용하도록 아무에게도 공개하지 않음
- 대칭키 암호에 비해 속도가 느림
- 대표적인 알고리즘: RSA, ECC(타원곡선 암호 알고리즘), ElGamal 등

기반 문제

- 공개키 암호 알고리즘은 수학적으로 어려운 문제들에 기반



- 다양한 일방향 함수

- 소인수분해 문제, 이산대수 문제, 타원곡선 이산대수 문제 등

공개키 암호 알고리즘 - RSA 알고리즘

■ RSA 알고리즘

- 1978년 Rivest, Shamir, Adleman이 개발
- 가장 널리 사용되는 공개키 암호
- 소인수분해 문제에 기반
 - 자릿수가 비슷하지만 두 수의 차가 큰 서로 다른 두 소수 p , q 이용

소인수분해 문제

- 어떤 두 정수의 곱은 빠른 시간에 구할 수 있지만, 임의의 양의 정수의 소인수분해는 매우 어려움

- $2 \times 5 =$

- $9539 \times 8887 =$

- $\underbrace{\hspace{1cm}}_{256 \text{ bits}} \times \underbrace{\hspace{1cm}}_{256 \text{ bits}} =$

- $10 =$

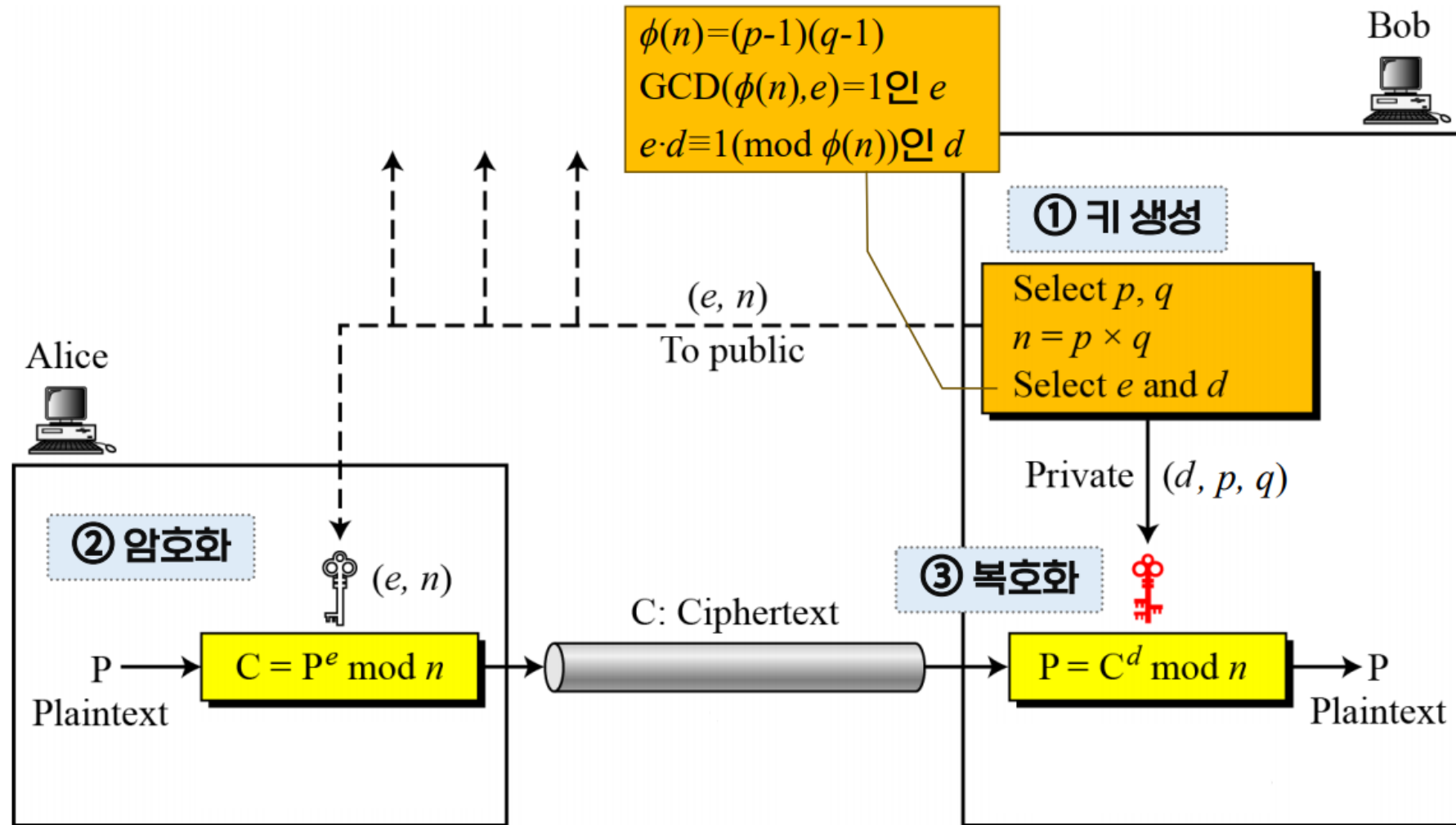
- $8477\ 3093 =$

- $\underbrace{\hspace{2cm}}_{512 \text{ bits}} =$

8,000
MIPS-Year

- 소인수분해 기반 알고리즘: RSA
- 안전한 암호로 사용하기 위한 키의 크기: 2,048 bits

RSA 암호 알고리즘



공개키 암호 알고리즘 - ElGamal 알고리즘

■ ElGamal 알고리즘

- 1985년 ElGamal이 개발
- 유한체상에서의 이산대수 문제에 기반
 - 수신자의 공개키 y 를 가지고 개인키 x 를 계산하는 것은 이산대수 문제

이산대수 문제

- 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

- $n = 11, a = 2, x = 6$

$$2^6 \pmod{11} =$$

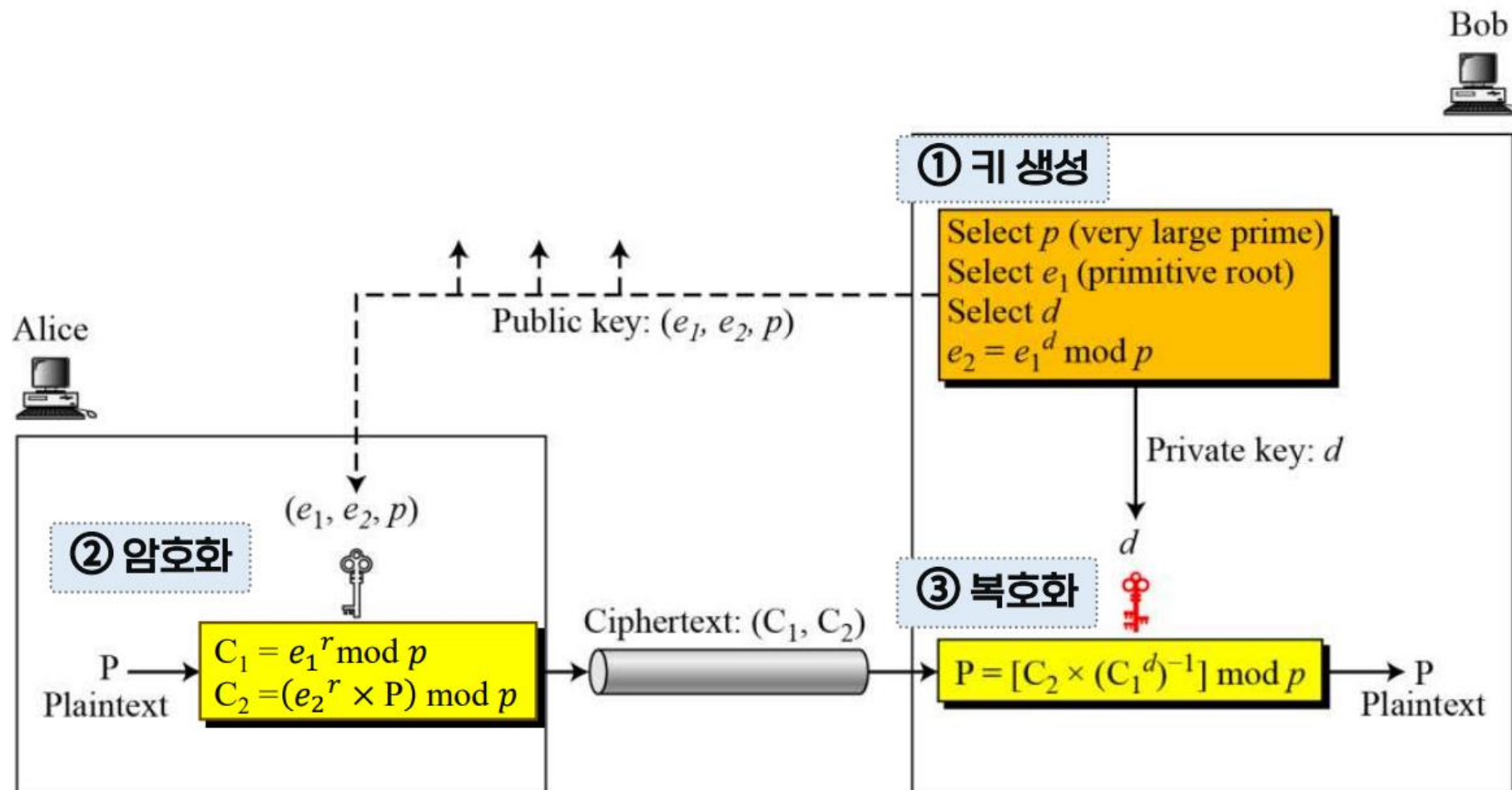
- $n = 11, a = 2, y = 9$

$$9 = 2^x \pmod{11}$$

$$x =$$

- 이산대수 문제 기반 알고리즘: ElGamal, DSA, KCDSA, DHKE 프로토콜
- 안전한 암호로 사용하기 위한 키의 크기: 2,048 bits

ElGamal 암호 알고리즘



■ ECC (Elliptic Curve Cryptosystem)

- 1985년 Miller와 Koblitz가 독립적으로 개발
- 유한체상에서 정의된 타원곡선 군에서의 이산대수 문제에 기반
- RSA, ElGamal 등과 동일한 수준의 보안성을 제공하면서 키의 길이는 짧음
 - 타원곡선 암호는 사양 변화 없이 높은 보안성 제공
- 이산대수 문제에 기반을 둔 ElGamal 등의 알고리즘을 변환하여 타원곡선 암호 알고리즘으로 적용

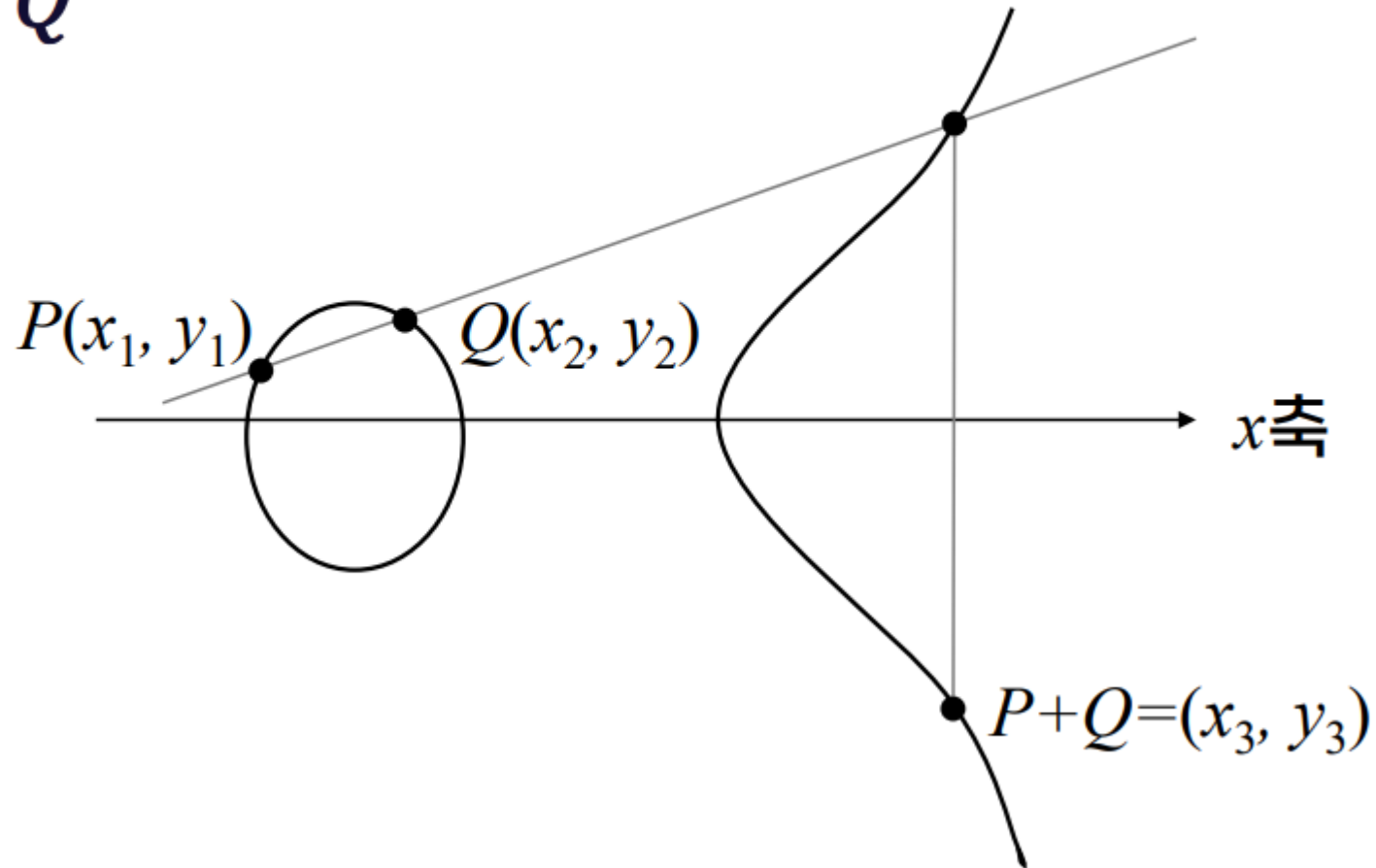
타원곡선 이산대수 문제

- 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

타원곡선상에서의 덧셈 연산

▪ 타원곡선 $y^2 = x^3 + ax + b$

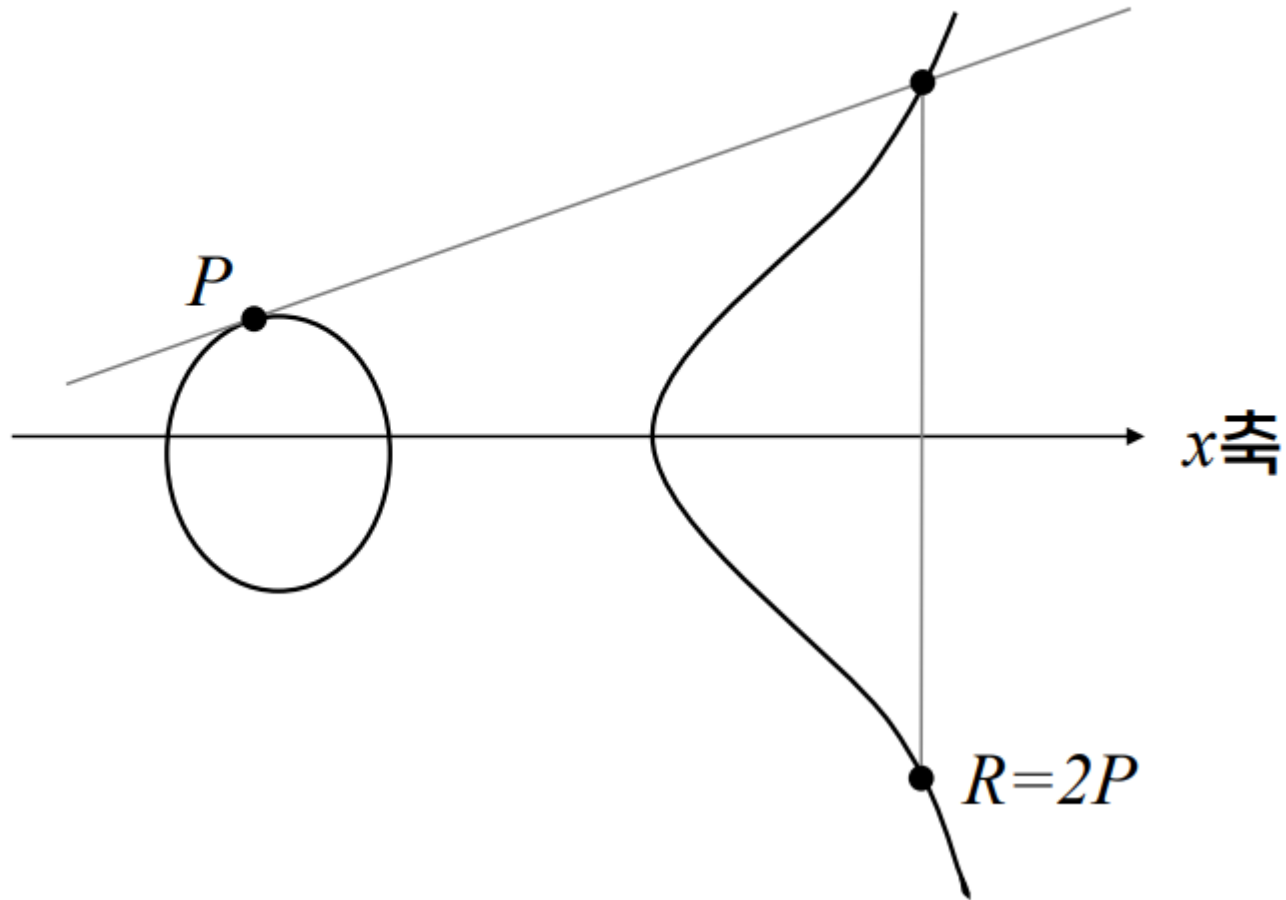
• $P + Q$



타원곡선상에서의 덧셈 연산

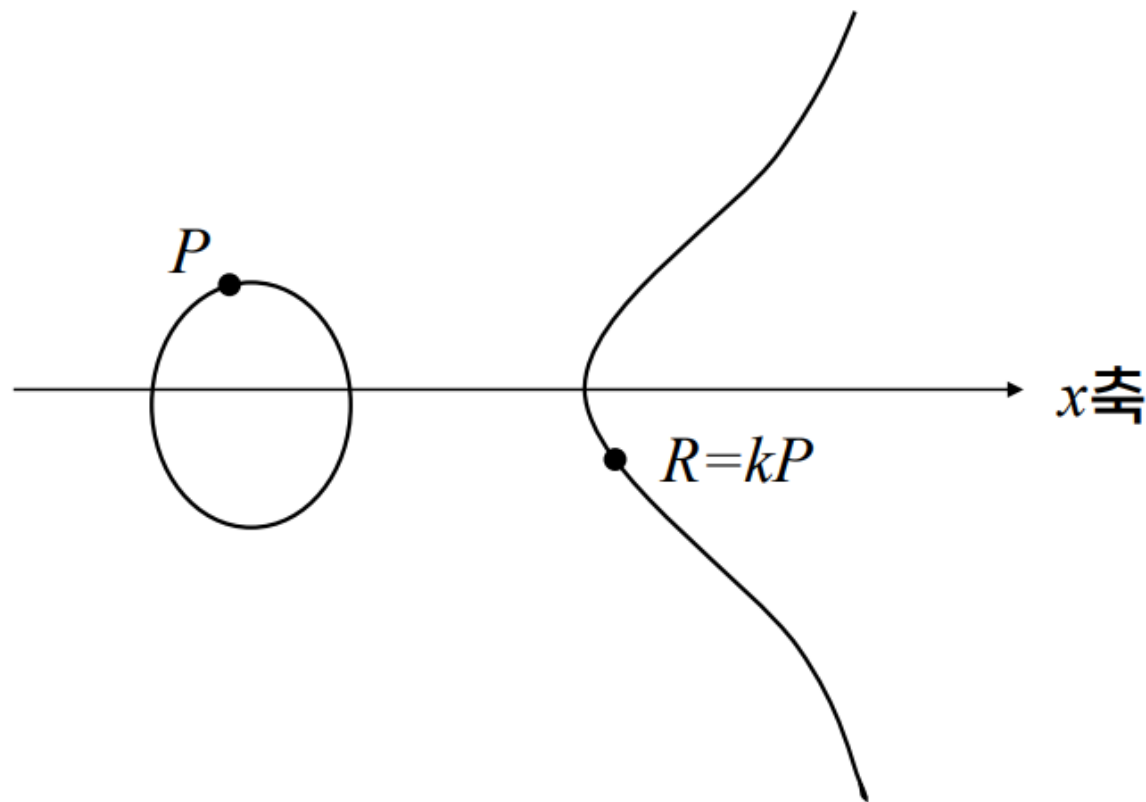
■ 타원곡선 $y^2 = x^3 + ax + b$

• $P + P$



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

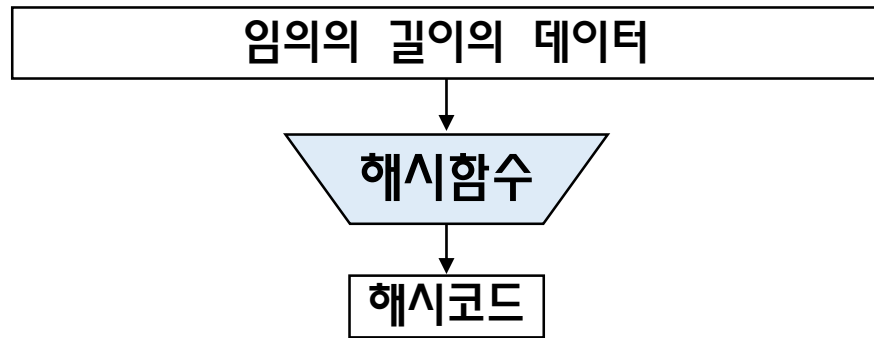


3강. 해시함수 및 전자서명

1. 해시함수

해시(hash)함수의 개념

- 임의의 길이의 입력 데이터를 고정된 길이의 해시코드(항상 일정한 길이)로 대응시키는 함수



- 송신자가 보낸 해시 값과 수신자가 받은 해시 값을 비교하여 일치하면 무결성 보장
- 일방향 함수
 - 임의의 데이터에 대한 해시코드는 쉽게 생성
 - 해시코드에 대응되는 데이터를 거꾸로 찾는 것은 어려움

해시 알고리즘 - 전용 해시 알고리즘

■ MD4 (Message Digest 4)

- 1990년 발표
- 대부분의 전용 해시 알고리즘의 기본 모델
- 해시코드 크기: 128비트

■ MD5 (Message Digest 5)

- MD4보다 보수적으로 설계
- 해시코드 크기: 128비트
- 널리 이용되었으나, 충돌 저항성과 일방향성에 문제가 있음이 알려져 사용하지 말 것을 권고

해시 알고리즘 - 전용 해시 알고리즘

■ SHA-1 (Secure Hash Algorithm-1)

- 1995년 발표된 표준 해시 알고리즘
- 2^{64} 비트보다 작은 크기의 입력 데이터에 적용 가능
- 해시코드 크기: 160비트
- 조금씩 취약성이 드러나고 있음

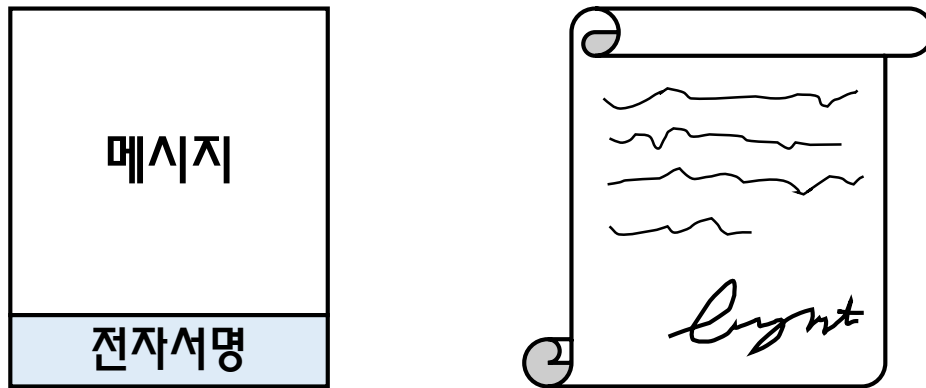
■ SHA-2, SHA-3

- 표준 해시 알고리즘으로 추가
- 해시코드의 크기에 따라 SHA-224, SHA-256, SHA-384, SHA-512 존재

2. 전자서명

전자서명(digital signature)의 개념

- 메시지를 보낸 사람의 신원이 진짜임을 증명
- 또한 전달된 메시지의 원래 내용이 변조되지 않았음을 보증



- 수기서명과 동일한 효력을 지님
- 메시지의 암호화 여부와 상관없이 사용됨

전자서명의 동작원리

■ 서명

- 개인키 이용 – 오직 자신만 서명할 수 있음
- 해시함수 이용 – 메시지에 의존하지만
작은 크기의 서명을 생성

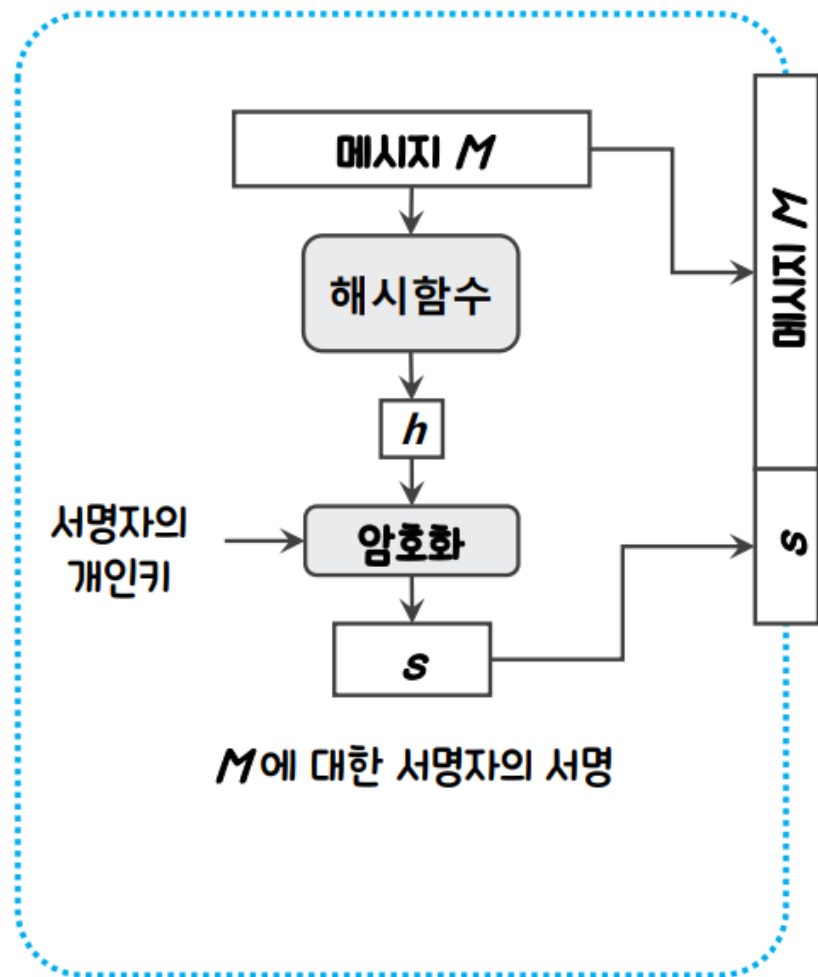


■ 검증

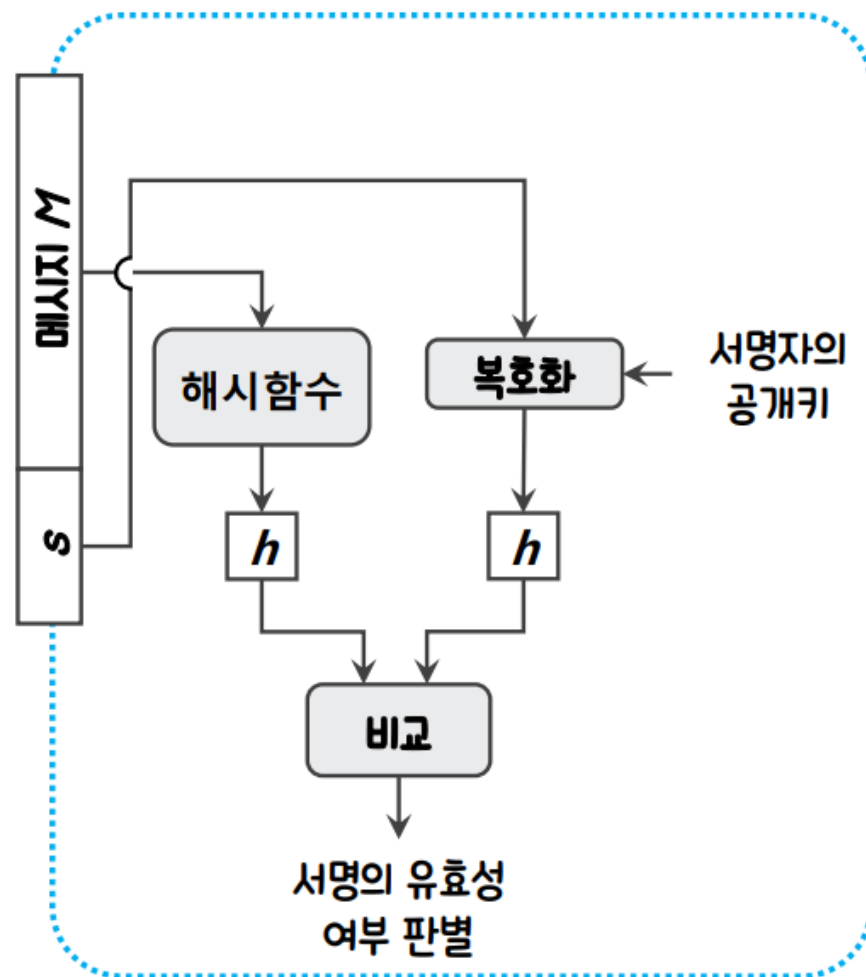
- 공개키 이용 – 누구나 서명을 검증할 수 있음

전자서명의 동작원리

■ 서명(송신자)



■ 검증(수신자)



4강. 사용자 인증

사용자 인증의 개념

- 시스템에 접근하려는 사용자가 정말 그 사용자가 맞는지 확인하는 것

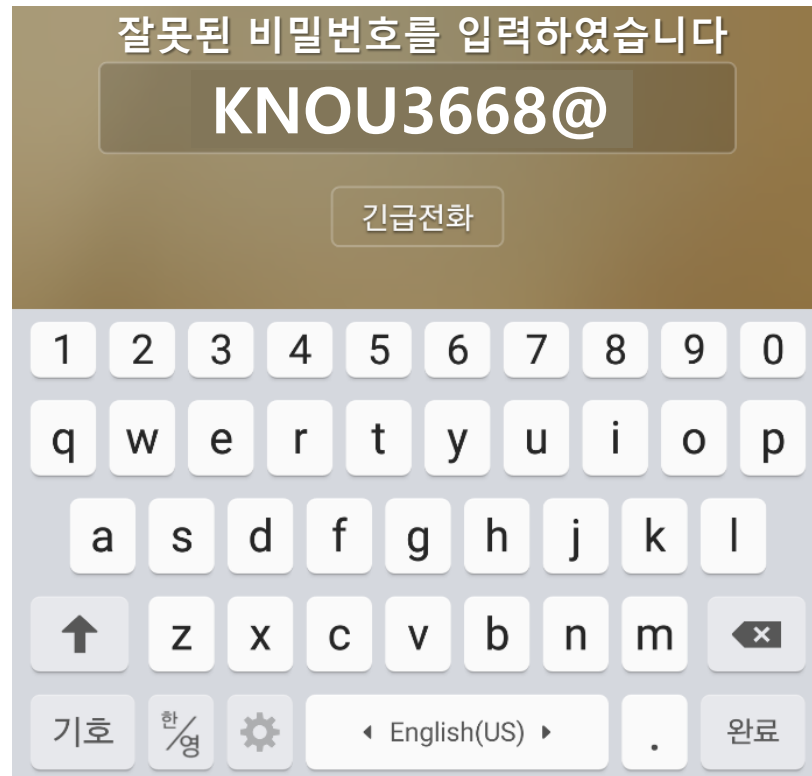
➤ 시스템: 서버, PC, 스마트폰, 홈페이지, 건물의 전자식 잠금장치 등



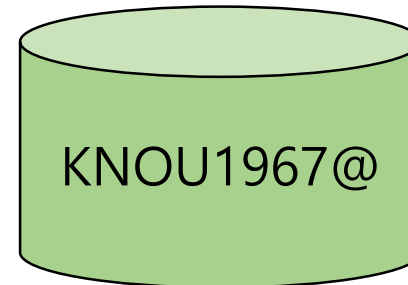
- 비밀번호 방식, 생체인식 방식, 토큰 방식, 2단계 인증 등

비밀번호 방식

- 가장 일반적인 방식의 사용자 인증
- 사용자가 입력하는 비밀번호가 시스템에 저장된 정보와 일치하는지 여부로 인증



시스템에
저장된 정보



비밀번호 방식

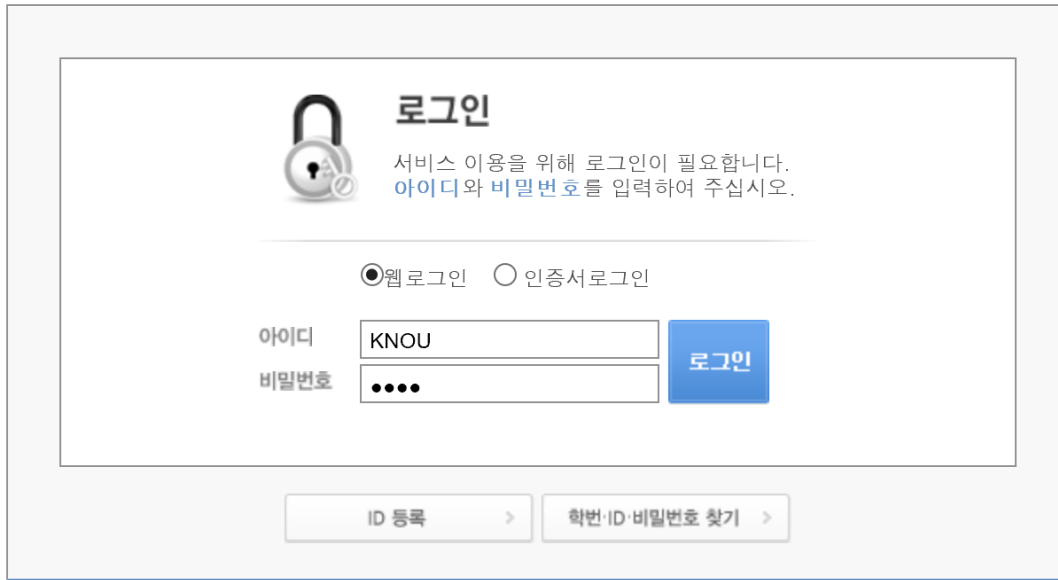
내부사용자의 비밀번호에 대한 접근통제, 암호화 저장, 무작위 시도 차단을 위한 대책을 적용해야 한다. **이용자들이 강도가 높은 비밀번호를** 사용하게 하고, **유추/도난을 방지**하기 위한 관리방안을 적용해야 한다.



By Scott Adams
- UFS Inc.

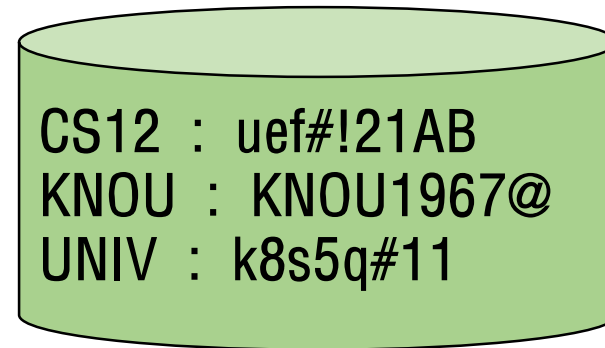
비밀번호 방식

■ 사용자 ID와 비밀번호를 함께 이용하는 경우

A screenshot of the KNOU login page. It features a lock icon and the title '로그인'. Below the title, it says '서비스 이용을 위해 로그인이 필요합니다. 아이디와 비밀번호를 입력하여 주십시오.' There are two radio buttons for '웹로그인' (selected) and '인증서로그인'. Below these are input fields for '아이디' (containing 'KNOU') and '비밀번호' (masked with dots). A blue '로그인' button is to the right of the password field. At the bottom, there are links for 'ID 등록' and '학번·ID·비밀번호 찾기'.

- 비밀번호로 인증하게 되는 사용자: 사용자 ID
- 사용자 ID와 비밀번호의 쌍을 시스템은 알고 있어야 함

시스템에 저장된 정보

A green cylinder representing a database. It contains the following text:

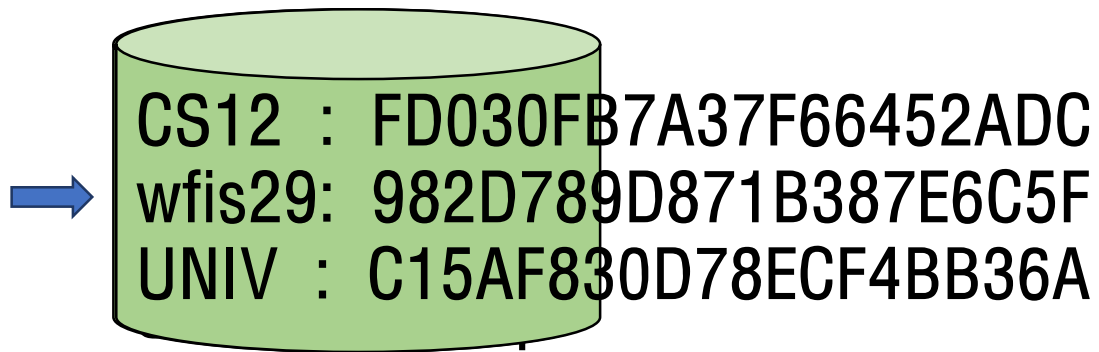
CS12 : uef#!21AB
KNOU : KNOU1967@
UNIV : k8s5q#11

비밀번호 방식

■ 비밀번호 저장 방법

- 시스템에 저장된 비밀번호가 유출되는 것에 대비하기 위하여 비밀번호를 저장할 때 해시코드로 저장

시스템에 저장된 정보



아이디

비밀번호

해시함수

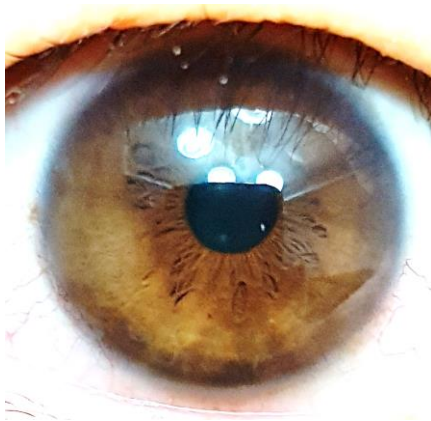
982D789D871B387E6C5F

=

생체인식 방식

- **개개인의 고유한 정보인 특정 생체정보를 이용하는 사용자 인증**

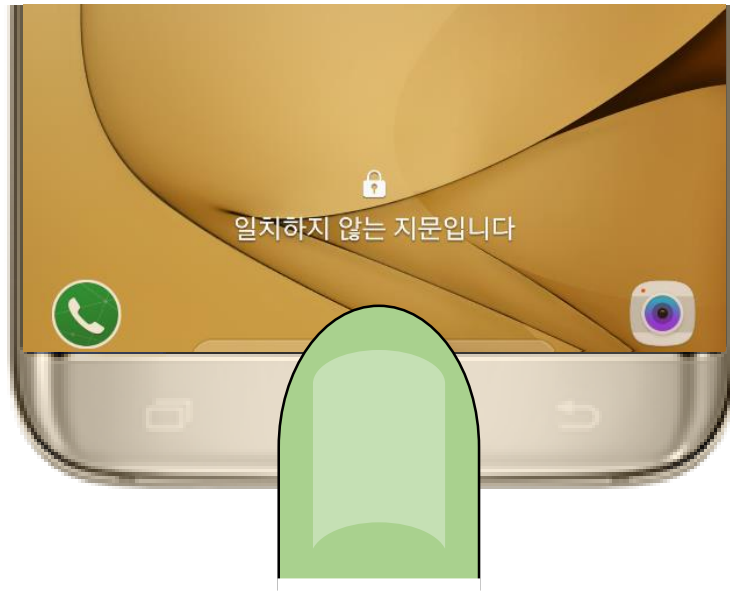
➤ 생체정보: 지문, 홍채, 음성, 손등의 혈관 등



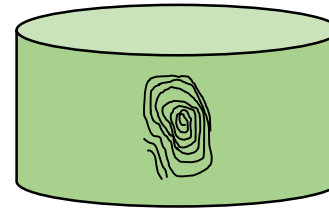
- 스마트폰, 노트북 등에 생체인식 모듈이 포함되어 장치에서의 사용자 인증 뿐만 아니라 홈페이지 접근이나 결제 시에도 활용되고 있음

생체인식 방식

- 사용자의 생체정보를 미리 시스템에 저장해 두어야 함



시스템에
저장된 정보

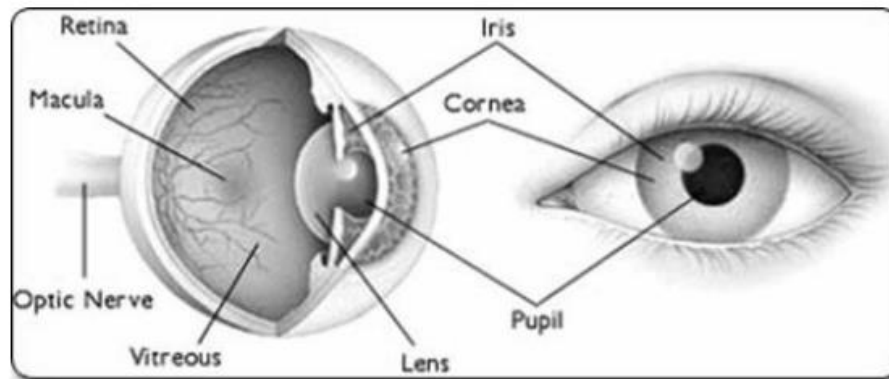
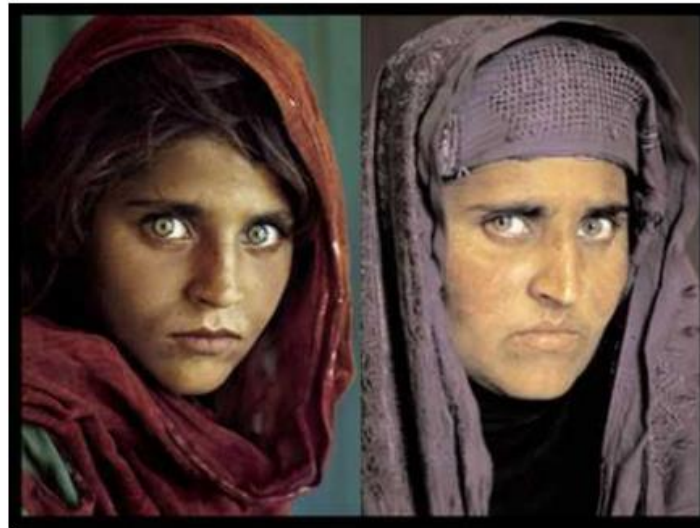


생체인식 방식

■ 홍채 인증

홍채 인식

1985년 내셔널지오그래픽 잡지 표지에 실려 스티브 맥커리를 세계에 알린 사진 '아프가니스탄 소녀' [왼쪽 · 당시 12세]와 17년 후의 그녀. 사진작가 맥커리는 파키스탄 페샤와르 난민촌을 수소문한 끝에 2001년 샤르밧 굴라와 극적으로 재회했다. 그녀는 세 아이를 둔 서른 살의 여인이 됐지만 강렬한 눈빛만은 여전했다. 이때 이 소녀의 신원을 확인할 수 있었던 것은 바로 **홍채 인식 기술**의 덕분이었다.



© Copyright IBM Corporation 2011

5강. 사이버 공격

■ 트로이 목마(Trojan Horse)

- 정상적인 기능을 하는 프로그램으로 가장하여 프로그램 내에 숨어서 의도하지 않은 기능을 수행하는 악성코드
- 그리스군이 트로이목마에 숨어 트로인군을 함락한 것에서 유래
- 사용자들로 하여금 거부감 없이 설치를 유도
- 표면적으로 드러나는 기능과 함께 비인가된 기능 수행
 - 개인정보유출, 감염 대상 원격 조정 등



<출처: 영화 'Troy' >

■ 백도어(Backdoor)

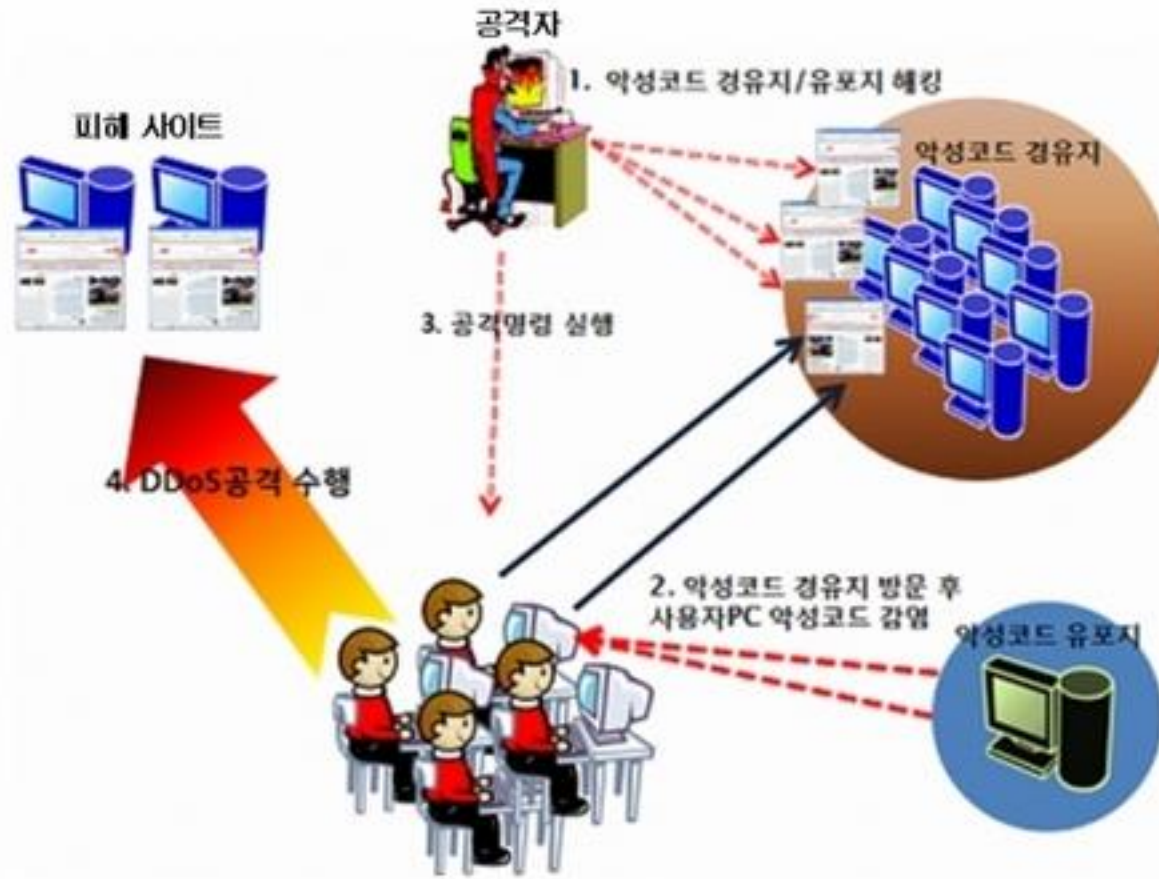
- 공격자가 시스템에 침입한 후, 이후에도 손쉽게 피해 시스템에 대한 접근 권한을 획득하기 위한 용도로 설치한 악성코드
- 시스템 설계자나 관리자에 의해 고의로 남겨진 시스템의 보안 구멍으로 응용 프로그램이나 운영체제에 삽입된 프로그램 코드
 - 사용자 인증 등 정상적인 절차를 걸치지 않고 응용 프로그램 또는 시스템에 쉽게 접근할 수 있도록 함

■ 랜섬웨어(Ransomware)

- 사용자의 중요한 정보를 인질로 삼아 금전을 요구하는 악성코드
 - 사용자의 문서 파일이나 그림 파일 등을 암호화
 - 암호를 풀기 위해서는 비트코인 등으로 송금하도록 유도
 - . 2009년 사토시 나카모토에 의해 비트코인이 만들어지면서 추적이 어려운 비트코인으로 송금하도록 함
- 2010년대 중반부터 새롭게 유행
- 키를 모르고서는 쉽게 풀 수 없는 안전한 암호 알고리즘을 이용
- 랜섬웨어에 감염되면 암호화된 문서를 풀기 위해 복호화 키 값을 알아야 하는데 감염시킨 주체만 키 값을 알고 있으므로 키 값을 알아내기란 매우 어려움

네트워크 공격

- 분산 서비스 거부(DDoS, Distributed Denial of Service) 공격
 - 여러 대의 공격자를 분산적으로 배치해 동시에 서비스 거부 공격을 하는 방법



피싱(Phishing)

- 유명한 금융기관이나 공신력 있는 업체의 이름을 사칭하여 메일을 보내 수신자들이 믿도록 하고, 수신자들로부터 개인정보나 금융정보를 얻어내 범죄수단으로 악용하는 행위
- 피싱 사이트 활용
 - 금융기관이나 공공기관 사이트처럼 만든 피싱 사이트로 수신자가 들어오게 함
 - 수신자는 자신이 접속한 사이트가 실제 사이트인지 제대로 확인하지 않은 상태에서 민감한 개인정보를 공격자에게 전달할 수 있음

피싱(Phishing)

- 대처 기법: 피싱이 이메일로 전달되는 점에 착안하여 이메일 내부에 존재하는 하이퍼링크들을 불허
- SMS를 이용하는 스미싱(smshing) 등 피싱 공격도 진화
- 파밍(pharming)
 - 사용자 PC의 도메인 정보를 조작
 - 사용자가 아무리 URL 주소를 주의 깊게 살펴봐도 속게 됨

6강. 보안 시스템

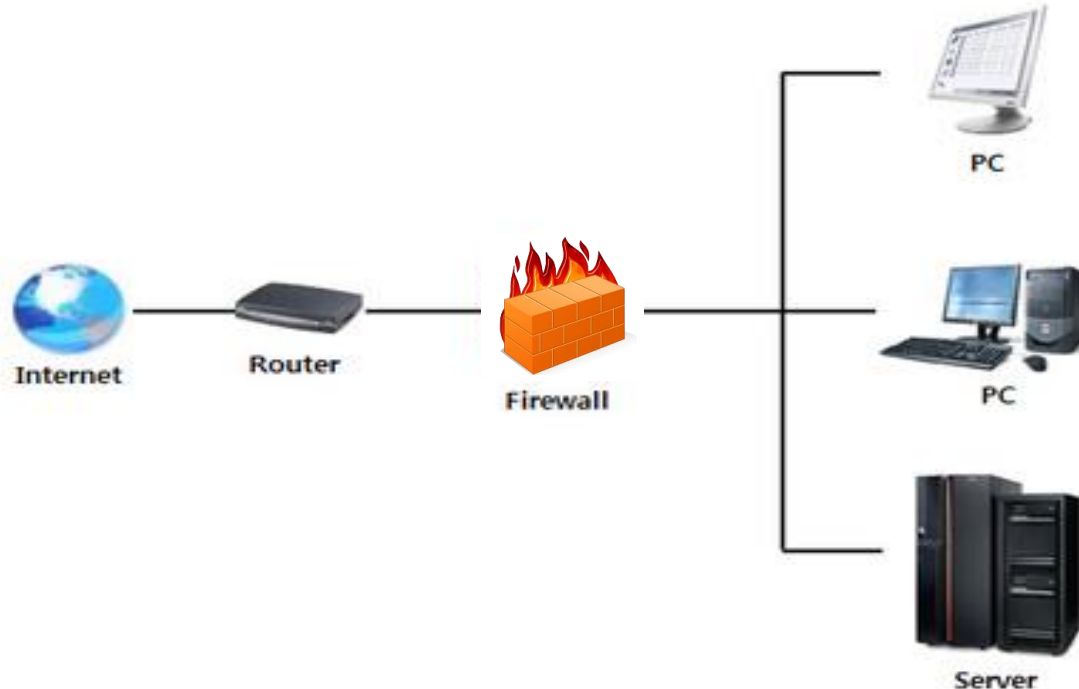
침입차단 시스템(방화벽, Firewall)

- 내부 네트워크의 컴퓨터들을 외부로부터 보호하기 위한 보안정책과 이를 수행하는 하드웨어 및 소프트웨어 등 침입차단을 위한 모든 것
- 외부의 공격자나 인증되지 않은 사용자, 혹은 유해한 정보가 내부 네트워크로 진입하지 못하도록 차단해 주는 보호정책과 보호장치



침입차단 시스템(방화벽)의 기능

- 기업, 공공기관, 사무실의 내부 네트워크와 외부 네트워크 중간지점에 위치시켜 두 네트워크 사이를 오가는 트래픽의 종류와 양을 제어
- 정상적인 사용자에게만 접근을 가능하게 해주고 불법적이고 인증되지 않은 사용자의 접근을 차단

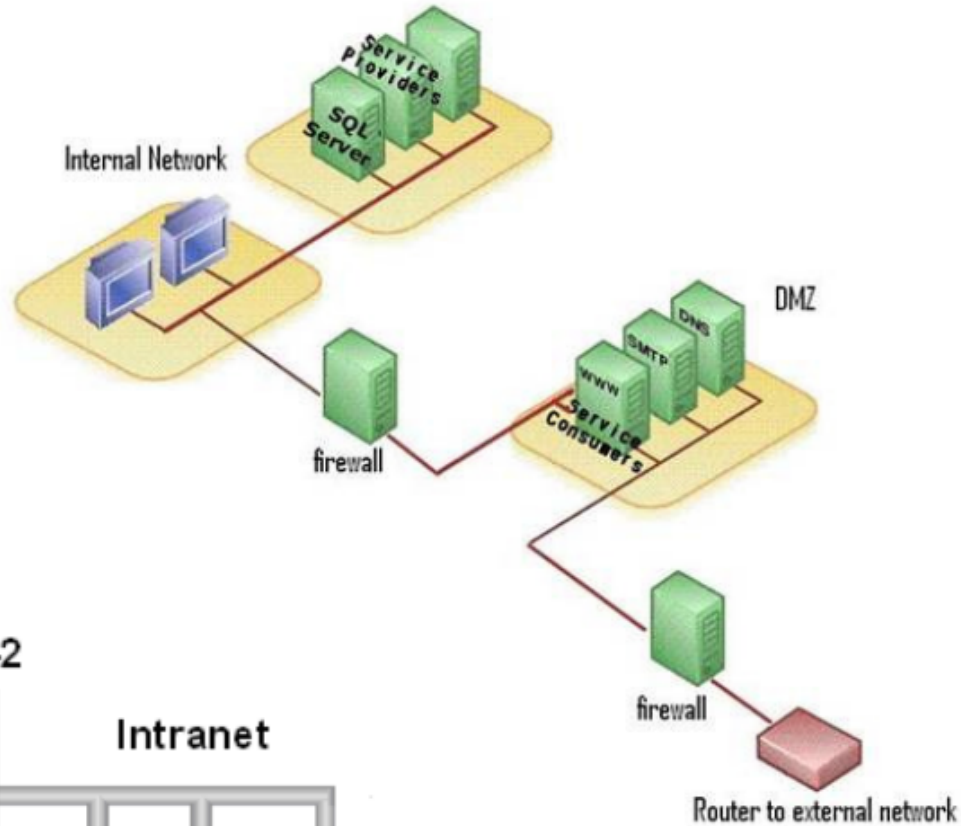
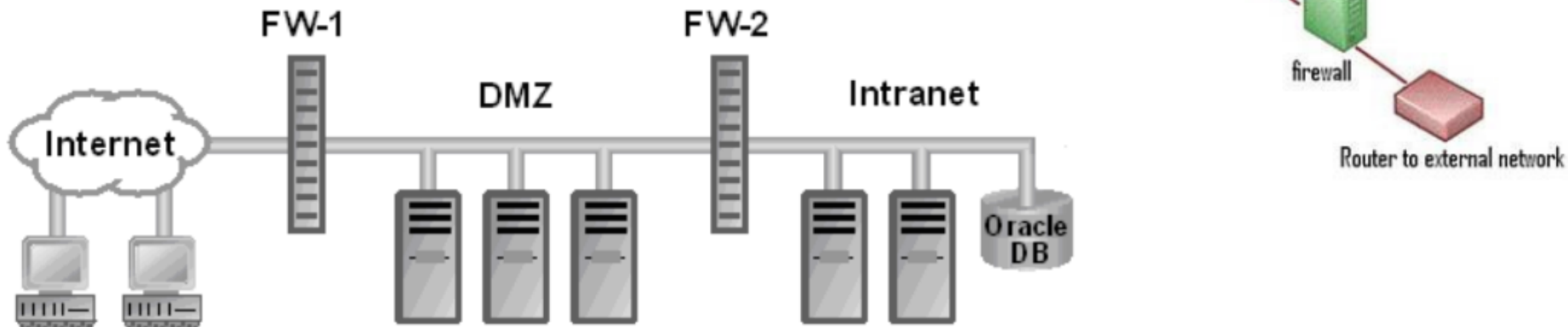


침입차단 시스템(방화벽)의 기능

DMZ

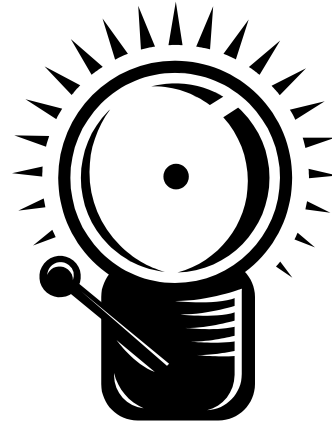
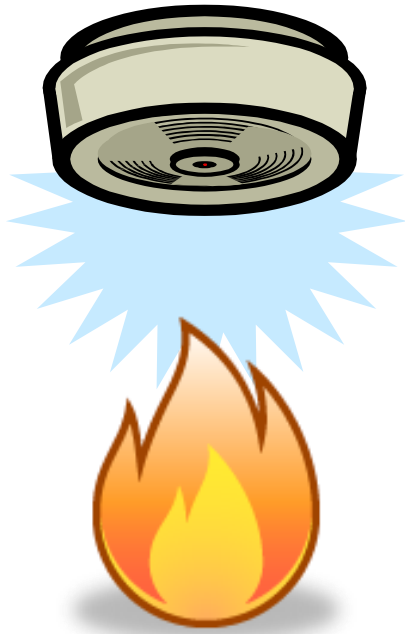
DMZ (De-Militarized Zone)의 약자로서, 스크린드 서브네트워크 (screened sub-network)라고도 부른다.

외부에 공개되어 있는 서버들이 위치하며, 민감한 정보들은 포함시키지 않는다.



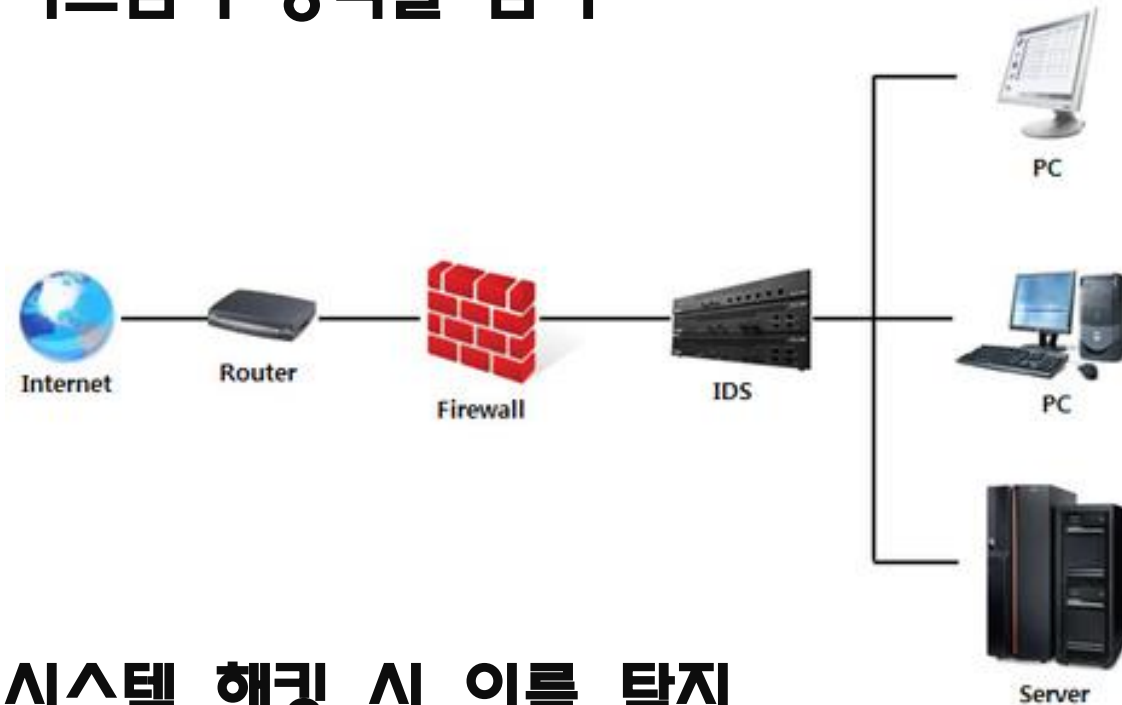
침입탐지 시스템(IDS, Intrusion Detection System)

- 컴퓨터가 사용하는 자원의 기밀성, 무결성, 가용성을 저해하는 행위를 실시간으로 탐지하는 시스템
- 허가 받지 않은 접근이나 해킹 시도를 감지하여 시스템 또는 망 관리자에게 통보해주는 시스템



침입탐지 시스템(IDS, Intrusion Detection System)

- 침입차단 시스템(방화벽)이 막을 수 없거나 해킹된 경우에도 침입탐지 시스템이 공격을 탐지



- 서버넷의 시스템 해킹 시 이를 탐지
- 해킹의 구체적인 내용을 관리자에게 알려주어 그에 따른 대응을 할 수 있도록 하는 솔루션

침입탐지 시스템(IDS)의 분석 방법

■ 시그니처(Signature) 분석

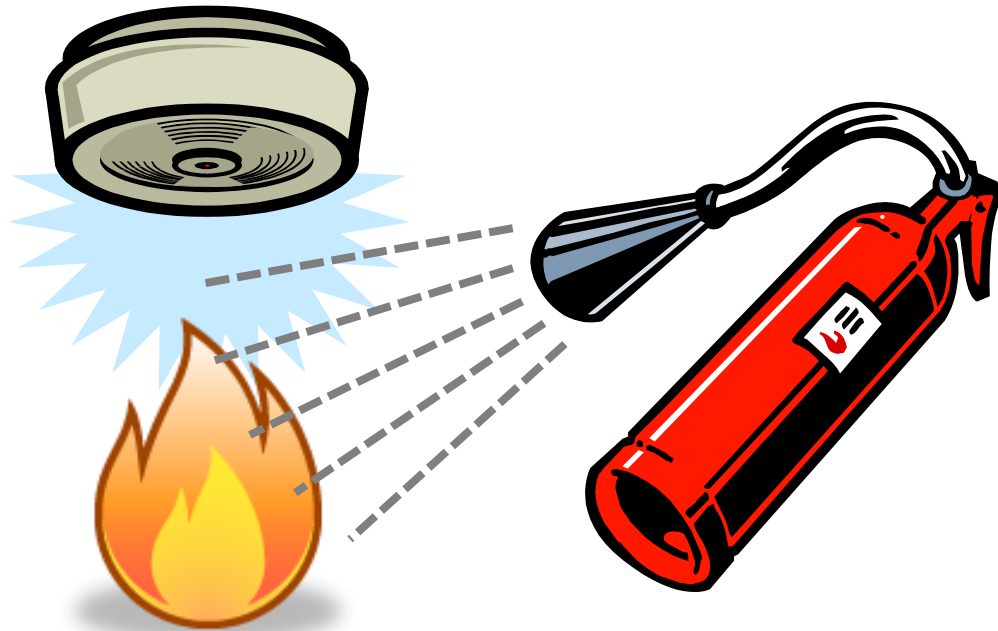
- 알려진 공격이나 시스템 오용과 관련된 것을 규정한 패턴과의 일치 여부 확인
 - 벤더가 공급하는 알려진 데이터베이스 공격에 대한 시그니처 분석 수행
 - 고객에 의해 명시된 시그니처가 추가될 수 있음
 - 벤더는 주기적으로 시그니처 데이터베이스의 업데이트 제공

■ 통계적(Statistical) 분석

- 정상적 행위 패턴으로부터 그 편차를 찾아내는 것
 - 프로파일은 해당 시스템의 정상적인 사용에 대한 다양한 속성을 판정하여 생성
 - 관찰된 값이 정상적인 범위에 속하지 않을 경우 침입 가능성 신호 발생

침입방지 시스템(IPS, Intrusion Prevention System)

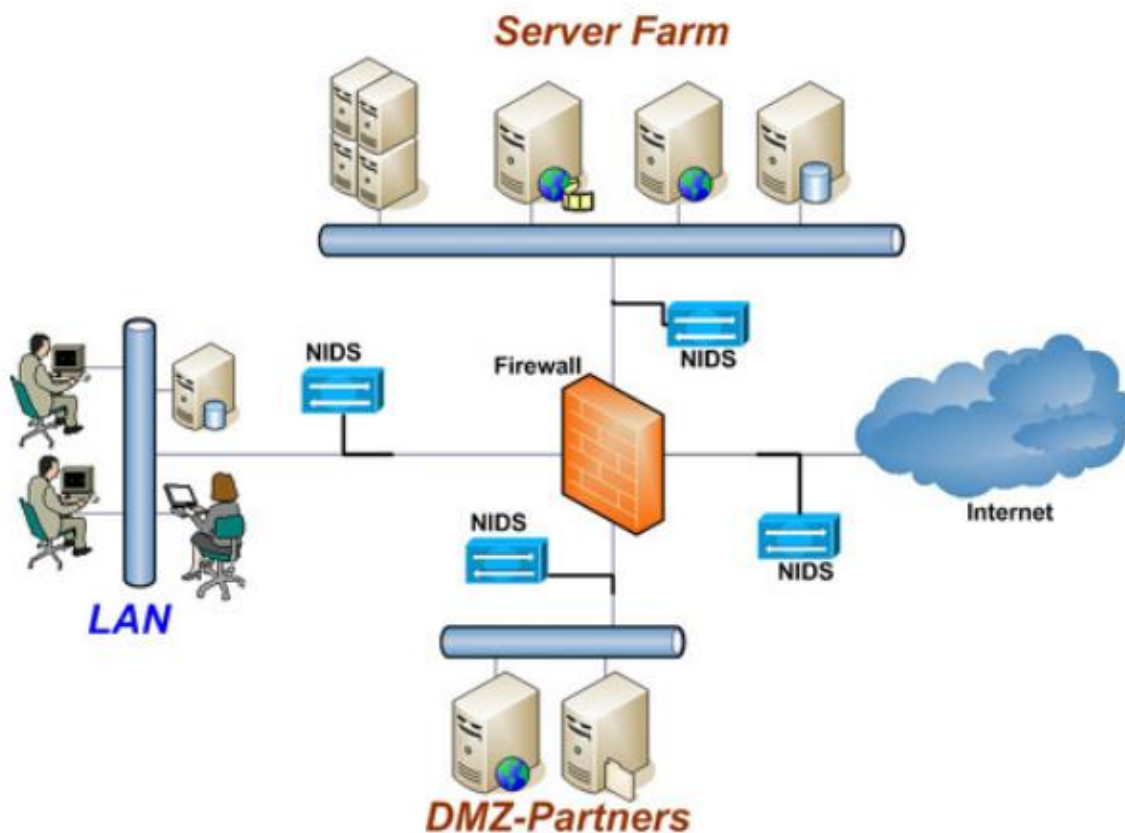
- 컴퓨터가 사용하는 자원의 기밀성, 무결성, 가용성을 저해하는 행위를 실시간으로 탐지하고 자동으로 대응 작업을 수행하여 행위를 중지시키는 보안 솔루션
- 수동적인 침입탐지 시스템과는 대비되어 능동적으로 동작



침입탐지 시스템(IDS) & 침입방지 시스템(IPS)

IPS & IDS

- **IPS (Intrusion Prevention System, 침입차단 시스템)**이란 공격자의 공격행위를 중단시키는 시스템이다.
- **IDS (Intrusion Detection System, 침입탐지 시스템)**이란 공격자에 의해 공격이 진행되고 있음을 경보하는 시스템이다.

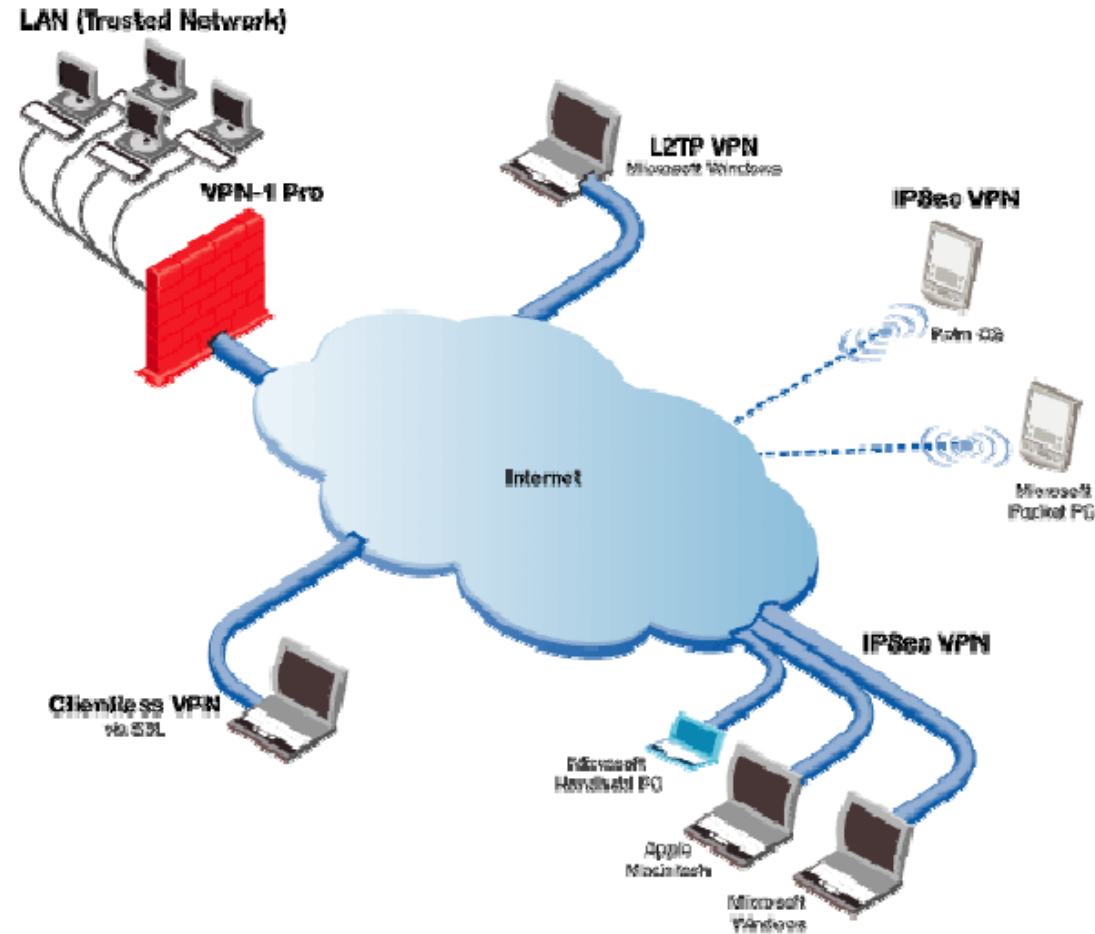


© Copyright IBM Corporation 2011

가상사설망(VPN, Virtual Private Networks)

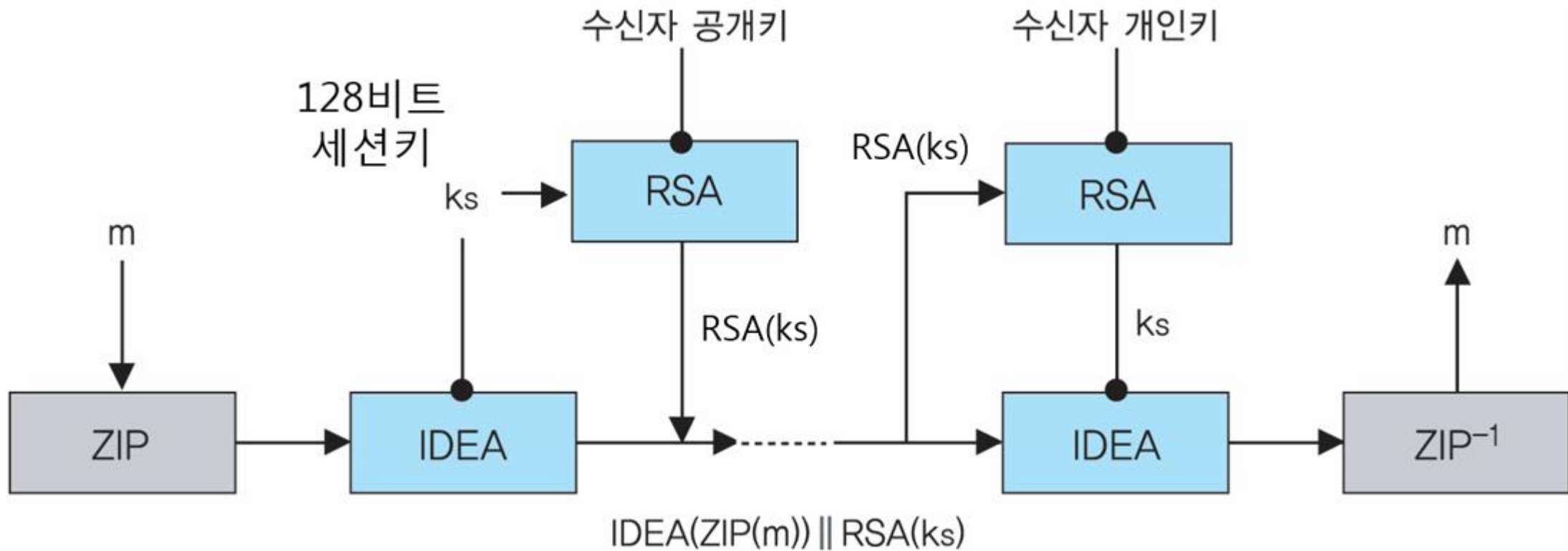
VPN

- VPN (Virtual Private Network, 가상사설망)이란 인터넷과 같이 공개된 네트워크 상에서 통신 주체들 간에 암호기술을 이용하여 통신함으로써, (1) 서로의 신원을 검증하고 (2) 메시지 무결성을 검증하며 (3) 통신 내용이 도청되지 않도록 보호해주는 통신기술이다.



© Copyright IBM Corporation 2011

이메일의 기밀성 보장을 위한 전송 방법

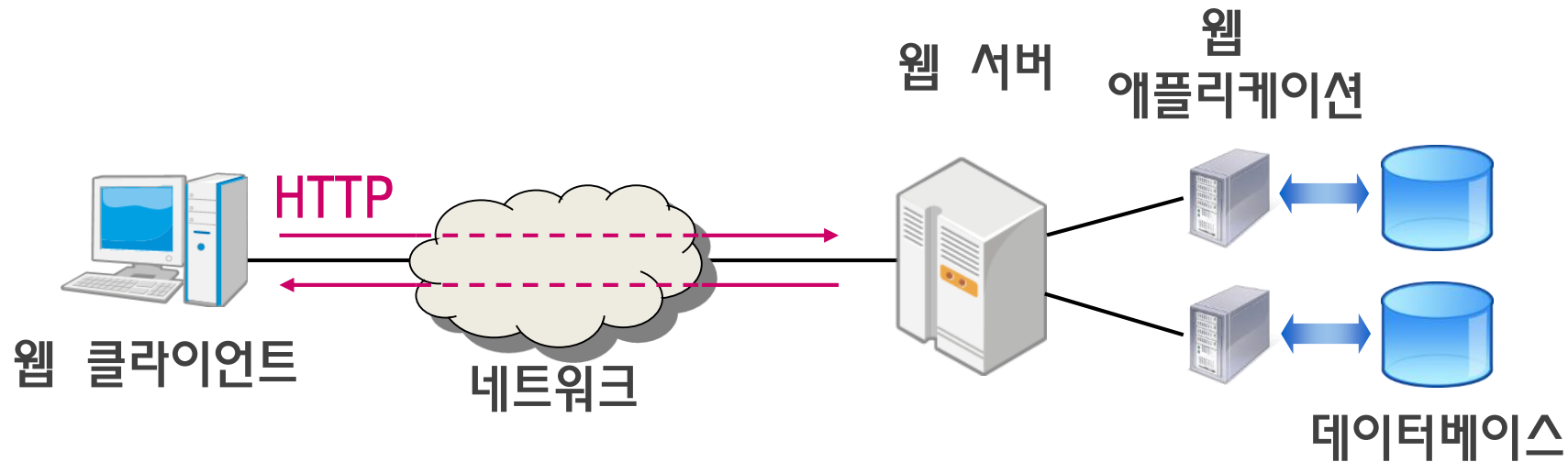


- ① 공간절약 및 보안강도를 높이기 위해 메시지를 압축한다.
- ② 일회용 세션키(k_s)로 압축된 메시지를 대칭키 암호 알고리즘인 IDEA를 이용하여 암호화 한다.
- ③ 세션키(k_s)는 공개키 암호 알고리즘인 RSA를 이용하여 수신자의 공개키로 암호화 한다.
- ④ 암호화된 세션키(k_s)와 메시지를 수신자 측에 보낸다.
- ⑤ 수신자 측은 수신자의 개인키를 이용하여 RSA 암호 알고리즘으로 세션키(k_s)를 복호화 한다.
- ⑥ 복호화된 세션키(k_s)를 이용하여 IDEA 암호 알고리즘으로 암호화된 메시지를 복호화 한다.
- ⑦ 복호화된 ZIP파일의 압축을 해제하고 메시지를 확인한다.

7강. 웹보안 및 모바일보안

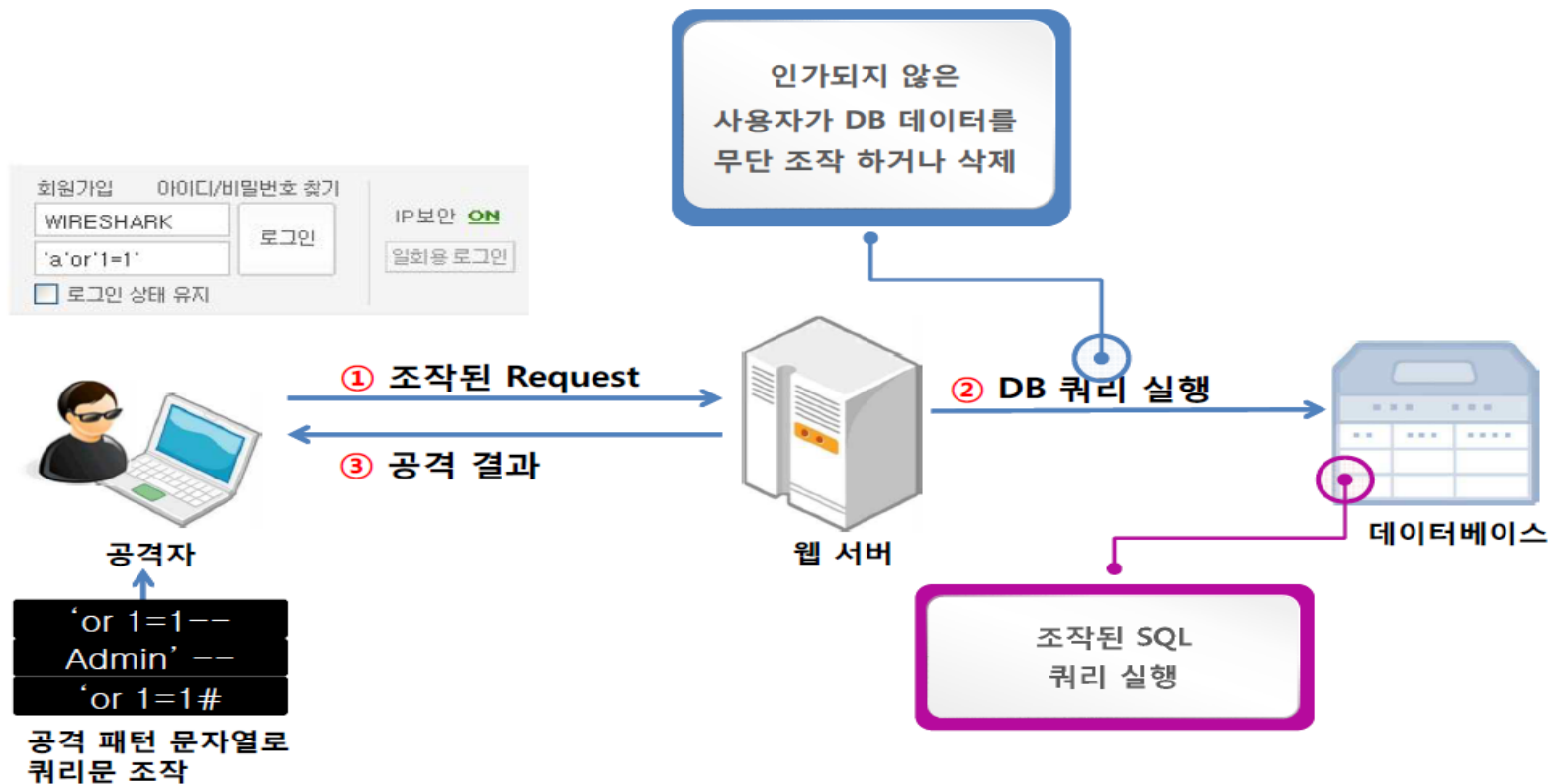
웹 서비스

- 웹 서버와 웹 클라이언트 사이에서 HTTP를 통해 제공되는 서비스
- 예: 인터넷 검색, 학교 홈페이지에서의 수강신청 등



SQL injection

- 데이터베이스 쿼리인 SQL문에 추가적인 SQL을 삽입(injection)함으로써 악의적인 행위를 가능하게 하는 공격
ex) 로그인 페이지에서 ID와 비밀번호 부분에 SQL문을 넣어 인증 우회



SQL injection

- 로그인 페이지에서 ID와 비밀번호 부분에 SQL문을 넣어 인증 우회



```
SELECT ... FROM ... WHERE id= 'ABC' AND  
pw= 'q1w2e3r4' ;
```

SQL injection

- 로그인 페이지에서 ID와 비밀번호 부분에 SQL문을 넣어 인증 우회



SELECT ... FROM ... WHERE id=" or '1'='1' AND pw=" or '1'='1';

- 따옴표를 이용하는 방법, 더블 하이픈을 이용하는 방법을 사용하지 못하도록 체크해야 함

SQL injection에 대비한 **시큐어 코딩** 사례

JDBC API 사용하는 경우 SQL 삽입 원인코드와 안전한 코드에

```
String userid=request.getParameter("userid");
String password=request.getParameter("password");
String query = "SELECT * FROM users WHERE userid =" + userid + " " +
               "AND password=" + password + "";
Statement stmt = connection.createStatement();
ResultSet rs = stmt.executeQuery(query);
```

안전하지 않은 코드

```
String userid=request.getParameter("userid");
String password=request.getParameter("password");
String query = "SELECT * FROM users WHERE userid=? AND password=?" ;
PreparedStatement stmt = connection.prepareStatement(query);
stmt.setString(1, userid);
stmt.setString(2, password);
ResultSet rs = stmt.executeQuery();
```

안전한 코드

Bluetooth

- 블루투스라는 명칭은 10세기 덴마크와 노르웨이를 통일한 바이킹 헤럴드 블루투스 (910 ~ 985)의 이름에서 따왔다. 블루투스가 스칸디나비아 반도를 통일한 것처럼 PC와 휴대폰 및 각종 디지털기기 등을 하나의 무선통신 규격으로 통일한다는 상징적 의미가 담겨 있다. 처음에는 프로젝트명으로 사용했으나 브랜드 이름으로 발전했다.



■ 사례 : 회사 출입 후 카메라 차단

- 직원이 안면인식 장치를 통해 **출입** 게이트가 열리는 순간 본인 휴대폰의 카메라 차단
- 사무실 내에서는 카메라가 차단되므로 내부 기밀자료를 사진촬영 할 수 없음
- 직원이 안면인식 장치를 통해 **퇴실** 게이트가 열리는 순간 본인 휴대폰의 카메라 정상 작동
- 보안을 우회하는 문제점이 무엇일까요? 해결방안은?



모바일 장치



액세스 포인트