Chap 3.
User Authentication

연세대학교

User authentication process

- 사용자 인증
  - 시스템 자원에 접근하려고 하는 시스템 entity가 제출한 ID를 확인하는 과정 (RFC 4949)
- 사용자 인증 절차
  - 컴퓨터 보안의 기본적인 구성 요소 및 기본 방어선
  - 접근 제어 및 user accountability를 위한 기본 과정
    - user accountability – 해당 user의 동작이 추적될 수 있게 보장하는 속성으로 동작에 대해서 책임을 질 수 있게 함.
- 사용자 인증 절차의 두 단계
  - **식별(identification) 단계** – 보안 시스템에 식별자(ID) 제시
  - **확인(verification) 단계** – entity와 ID간의 바인딩을 확증하는 인증 정보를 제시하거나 생성함

연세대학교

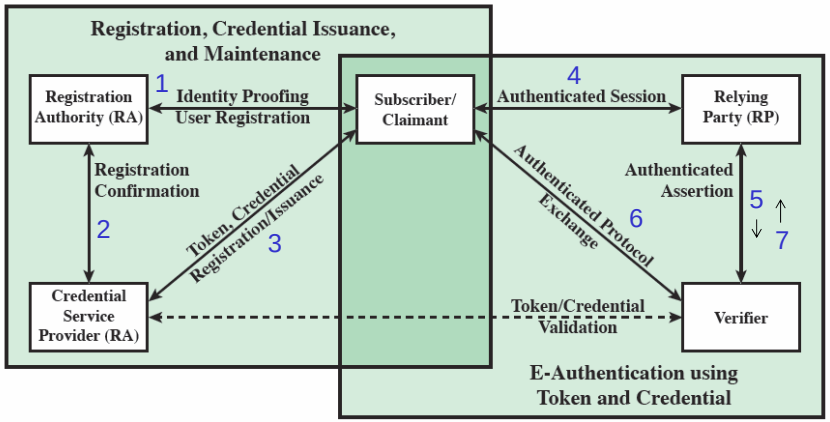
User Authentication

- 사용자를 인증하는 4가지 수단

- 개인이 알고 있는 것
  - 비밀번호, 미리 알려진 질문에 대한 대답
- 개인이 소유하고 있는 것 (토큰)
  - 스마트 카드, 전자키 카드, 물리적 키
- 개인의 어떤 것 (정적 생체정보)
  - 지문, 홍체, 안면
- 개인이 하는 어떤 것 (동적 생체인식)
  - 음성패턴, 필기체, 타이핑 리듬

연세대학교

E-Authentication Architectural Model



RA : 등록기관
CSP : 자격증명(인증서) 서비스 공급자
RP: 의뢰인
subscriber : 가입자, claimant: 청구자
verifier : 검증자

연세대학교

Registration and Authentication Sequence

- 신청자는 **자격증명 서비스 제공자**(CSP)의 가입자가 되기 위해서 **등록기관**(RA)에 신청
- RA는 CSP에게 신청자의 신원을 설정하고 보증함
- CSP는 자격증명 발급을 위해 가입자와 정보를 교환에 참여함
  - 자격증명, 인증서 (Credential Service)
    - ID와 가입자가 소유한 토큰의 속성에 부여된 자료 구조
- CSP는 전체 인증 시스템의 세부사항에 따라 여러 전자 인증서를 가입자에게 발급
- 사용자가 가입자로 등록되면 실제 인증 과정이 가입자와 하나 이상의 시스템 사이에서 발생
  - 청구자(가입자)가 의뢰자(relying party: RP) 요청
  - RP가 검증자(verifier)에서 인증확인 요청
  - 검증자는 청구자와의 인증프로토콜을 통하여 청구자가 인증서 가입자임을 확인하고 RP에게 확인증을 전달
  - RP는 청구자의 세션 사용을 허가

Password Authentication

- 널리 사용되는 침입에 대한 방어방법
  - 사용자는 **로그인 이름**과 **암호(password)**를 제공
  - 시스템은 로그인 이름에 대해 저장된 암호와 입력 암호를 비교
- 사용자 ID :
  - 사용자의 **시스템 접근 허가** 여부를 결정
  - 사용자의 **사용 권한**을 결정
  - 접근 제어에 사용됨

Password Vulnerabilities/Countermeasures

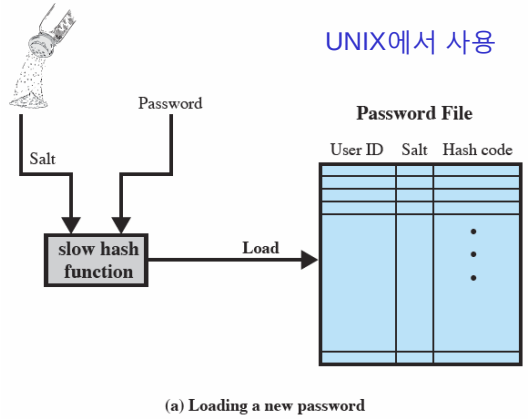
- 오프라인 사전(password file) 공격
  - 비밀번호 파일을 얻은 후 비밀번호 파일의 해시값과 흔히 사용되는 비밀번호 해시값과 비교하여 비밀번호를 알아내려고 함
    - ☞ 패스워드 파일에 대한 강한 접근 통제 - 비인가 접근 방지
    - ☞ 유출된 비밀번호의 신속한 재발급
- 특정계정 공격
  - 특정 계정을 목표로 비밀번호 입력 시도
    - ☞ 일정 횟수(대개 5회)이상 시도가 실패하면 계정 잠금
- 단일 사용자에 대한 비밀번호 추측
  - 사용자의 정보로부터 추측
    - ☞ 흔한 비밀번호 사용 금지, 비밀번호 최소길이 및 문자집합 지정, 사용기간 제한 등.
- 알려진 비밀번호 공격
  - 여러 사용자의 ID에 대해 흔히 사용하는 비밀번호 사용을 시도
    - ☞ 흔한 비밀번호 사용 금지, 인증요청 IP와 사용자 쿠키 스캐닝

- 단말기 강탈(hijacking)
  - ☞ 자동 로그아웃, 침입탐지 기법을 사용한 행동변화 탐지
- 사용자 실수 이용
  - 비밀번호를 기록한 것을 유출 – 실수, 사회공학 전술 등
    - ☞ 사용자 훈련, 침입탐지, 다른 인증방법과 결합
- 다중 비밀번호 사용 악용
  - 여러 네트워크 장치가 같거나 비슷한 비밀번호 사용
    - ☞ 같거나 비슷한 비밀번호 사용 제한
- 컴퓨터 모니터링(감시)
  - 비밀번호가 네트워크에서 원격 교환되면 도청에 취약
    - ☞ 비밀번호의 간단한 암호화는 해결책이 아님 – 재사용 공격

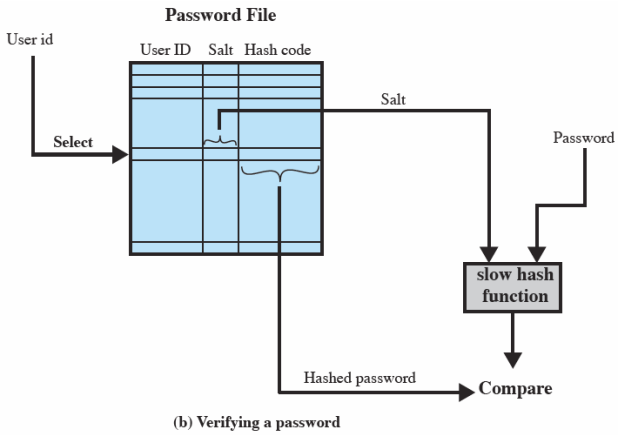
Use of Hashed Passwords

- 널리 사용되는 비밀번호 기술 - **해시된 비밀번호**와 **솔트(salt)** 사용
  - password file에 중복된 비밀번호가 나타나는 것 방지
  - 오프라인 사전 공격 방지 – 경우의 수가 salt에 의해서 증가됨
  - 2개 이상의 시스템에 같은 비밀번호를 사용해도 찾기 어려움

UNIX에서 사용



(a) Loading a new password



(b) Verifying a password

UNIX Implementation

- original scheme
  - 비밀번호로 8자 까지의 printable characters 사용 → 56비트 키
  - 12-bit salt 값 사용. DES 암호화 기반 단방향 해시함수
  - 64비트 zero block에 대해서 DES 암호화를 25번 반복
  - 64비트 출력이 11개 문자로 변환됨
- 현재는 더 이상 적절한 방법이 아님
  - 그렇지만, 기존 계정 관리 소프트웨어 또는 멀티 벤더 환경과의 호환성 유지를 위해 여전히 요구됨

Improved Implementations

- 현재 Unix에 더 강력하진 hash/salt 기법이 존재함
- MD5에 기반한 hash 함수가 추천됨
  - 최대 48-bit 길이의 salt
  - 비밀번호 길이에 제한 없음
  - 128-bit 길이의 hash값 생성
  - 속도 저하를 위해 1000번 반복을 포함한 내부 반복문 사용
- Bcrypt - OpenBSD에서 가장 안전한 hash/salt 기법을 개발함
  - **Blowfish** block symmetric cipher 알고리즘 사용
    - DES 대안으로 1993년에 설계된 알고리즘
  - 55자리까지의 비밀번호, 128-bit salt를 사용, 192-bit hash값 생성

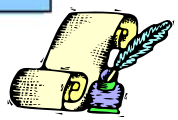
Password Cracking

- dictionary attacks
  - 가능한 패스워드에 대한 사전을 개발하고
    각 암호 파일에 대해 사전의 패스워드들을 시도함
  - 각 암호는 salt값을 사용하여 해시하여 저장된 해시값과 비교
- rainbow table attacks
  - 모든 salt에 대해서 미리 계산한 해시 값 테이블 준비하여 공격을 시도 - 해시 값 계산이 불필요하여 빠른 시간에 공격 가능
  - 충분히 큰 salt 값과 충분히 큰 길이의 해시를 사용하여 이러한 공격에 대처함
- Password crackers는 사람들이 쉽게 추측할 수 있는 암호를 선택한다는 사실을 이용함
  - 짧은 패스워드는 crack하기가 쉬움
- John the Ripper
  - 1996년에 처음 개발된 Open-source password cracker
  - 사전 공격 사용

Observed Password Lengths

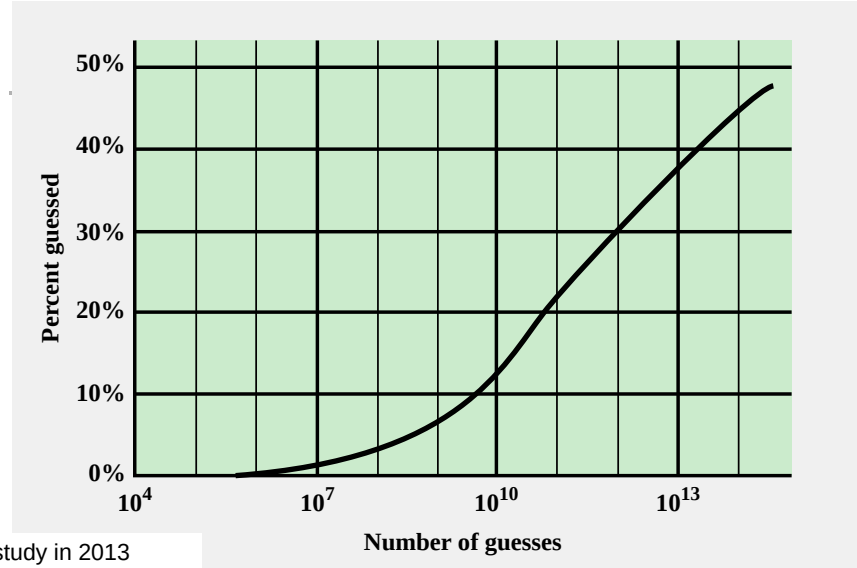
Length	Number	Fraction of Total
1	55	.004
2	87	.006
3	212	.02
4	449	.03
5	1260	.09
6	3035	.22
7	2917	.21
8	5772	.42
Total	13787	1.0

Purdue University study on
54 systems and 7000 users



Modern Approaches

- 복잡한 password 정책
  - 사용자가 더 강력한 암호를 선택하도록 요구함
- 그렇지만 password-cracking 기술도 향상됨
  - password cracking에 사용할 수 있는 처리 능력이 급속히 증가됨.
    - GPU 사용 등
  - potential password를 생성하는 정교한 알고리즘 사용
    - 실제 password의 사례와 구조에 대한 연구를 바탕으로 함
    - 많은 유출된 password를 이용



study in 2013

25,000 students at university with complex password policy

The Percentage of Passwords Guessed After a Given Number of Guesses

Password File Access Control

- 암호화된 password에 대한 접근 통제
  - 특권 사용자에게만 접근 허용
  - shadow file을 사용 – 사용자 ID 파일과 분리된 파일에 password 저장
- 취약성
  - 운영체제 취약성이 파일 접근을 허용할 수 있음
    - 시스템 정지에 취약함. 접근 제어 시스템을 우회함
  - 읽기 허가권을 제공하는 보안 사고 가능성
  - 여러 시스템에 같은 password 사용
  - Back 장치에서 접근
  - Sniffing - 네트워크 트래픽에서 password 수집 접근하여 수집

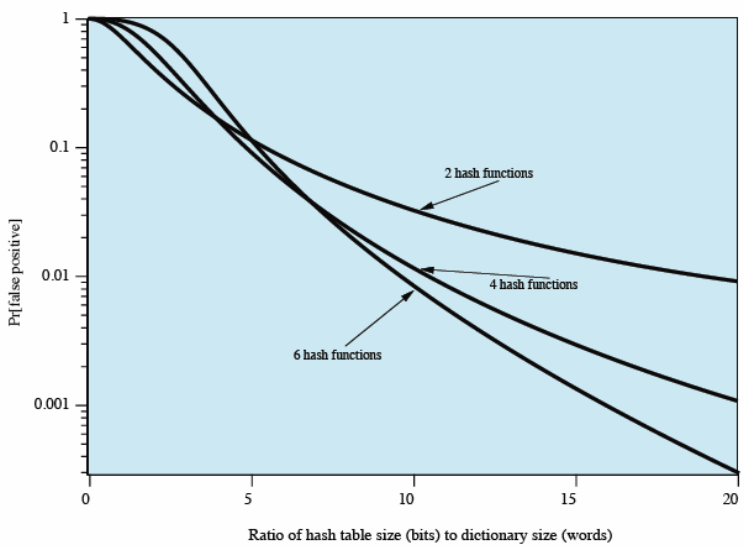
Password Selection Techniques

- 사용자 교육
- 컴퓨터에 의한 password 생성
  - 기억하기가 어려움
- 반응적(reactive) password 확인
  - 시스템이 주기적으로 password cracker를 수행하여 추측 가능한 password를 찾아냄
- 복잡한(proactive) password 정책
  - 사용자가 password 선택하지만 허용여부를 시스템이 확인
  - 사용자가 추측 가능하지 않지만 기억할 수 있는 password를 선택하도록 함

Proactive Password Checking

- 규칙 집행
  - Password 규칙 지정하여 집행
  - 사용하지 않아야 하는 비밀번호 사전 준비하여 검사
- Bloom filter 사용
  - 여러 개의 해시 함수를 사용하여 사전 테이블 구성
    - $H_1(x), H_2(x), \ldots, H_k(x)$
  - 해시테이블에서 Password의 해시 값에 해당하는 bit를 1로 설정
    - $H_1(aaa) = 25$ → H1 테이블의 25번째 비트 = 1
    - $H_2(aaa) = 67$ → H2 테이블의 67번째 비트 = 1
  - 새로운 Password에 대해서 해시 테이블의 해당 비트가 모두 1이면 해당 password는 허용되지 않음
  - false positive 가능 – 사전에 포함되지 않았지만 허용되지 않음
    - k가 클수록 가능성은 감소함, 해시함수에도 영향 받음

Performance of Bloom Filter



Token-based Authentication

- Token
  - 사용자 인증을 위해 사용자가 소유한 객체
- Token으로 사용되는 카드 유형

Card Type	Defining Feature	Example
Embossed	Raised characters only, on front	Old credit card
Magnetic stripe	Magnetic bar on back, characters on front	Bank card
Memory	Electronic memory inside	Prepaid phone card
Smart	Electronic memory and processor inside	Biometric ID card
Contact	Electrical contacts exposed on surface	
Contactless	Radio antenna embedded inside	

Memory Cards

- 데이터 저장 가능, 처리 불가능
- 예
  - 자기 카드 – magnetic stripe를 가짐
  - 전자 메모리 포함
- 비밀번호(또는 PIN)와 결합하여 강력한 보안 제공 가능

- 단점
  - 특수 판독기(reader) 필요
  - 토큰(메모리카드) 손실 가능성
  - 사용자의 불만 - 불편함

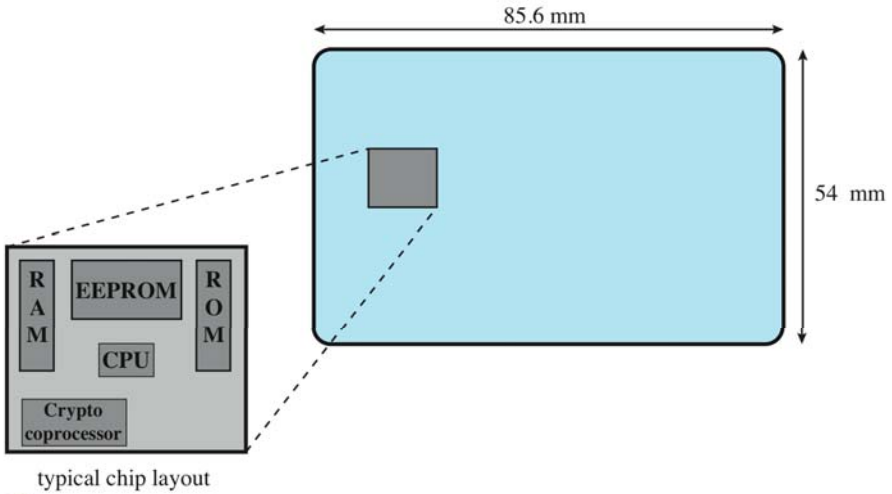
Smart Tokens

- 물리적 특징
  - embedded microprocessor 포함
  - 형태:
    - 스마트 카드 - 은행카드와 같은 형태
    - 계산기, 키, 작은 휴대용 물체 등
- 인터페이스
  - 사용자 인터페이스 – 키패드와 디스플레이 포함
  - 전자 인터페이스 – 호환 판독기/기록기와의 통신
- 인증 프로토콜
  - static - 자신을 토큰에 인증, 토큰이 사용자를 컴퓨터에 인증
  - dynamic password generator
  - challenge-response :
    - 컴퓨터가 랜덤 문장 생성(challenge), 스마트토큰이 응답

Smart Cards

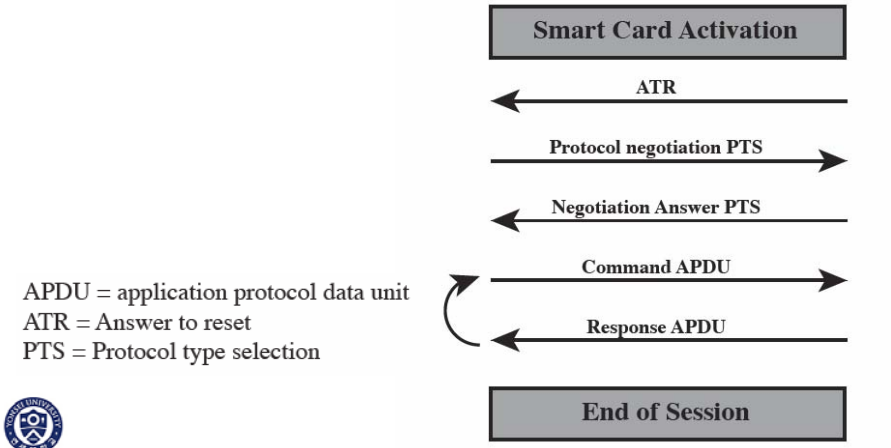
- 가장 중요한 형태의 스마트 토큰
  - 신용카드 모양으로, 전자 인터페이스 보유
  - 스마트 토큰 프로토콜 사용
  - 마이크로프로세서 시스템 포함
    - Processor, Memory, I/O ports
- 세 종류의 메모리 포함
  - Read-only memory (ROM)
    - 변경되지 않는 정보 저장 – 카드번호, 이름 등
  - Electrically erasable programmable ROM (EEPROM)
    - 프로그램과 데이터 저장
  - Random access memory (RAM)
    - 프로그램 실행 동안 일시적인 데이터 저장

Smart Card Dimensions



typical chip layout

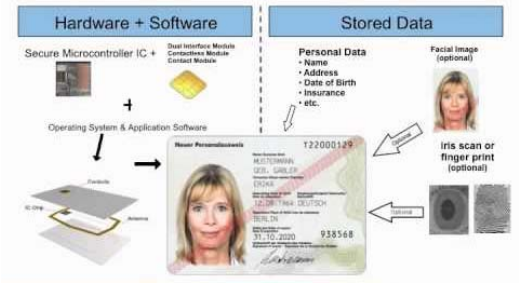
Smart Card/Reader Exchange



APDU = application protocol data unit
ATR = Answer to reset
PTS = Protocol type selection

Electronic Identity Cards (eID)

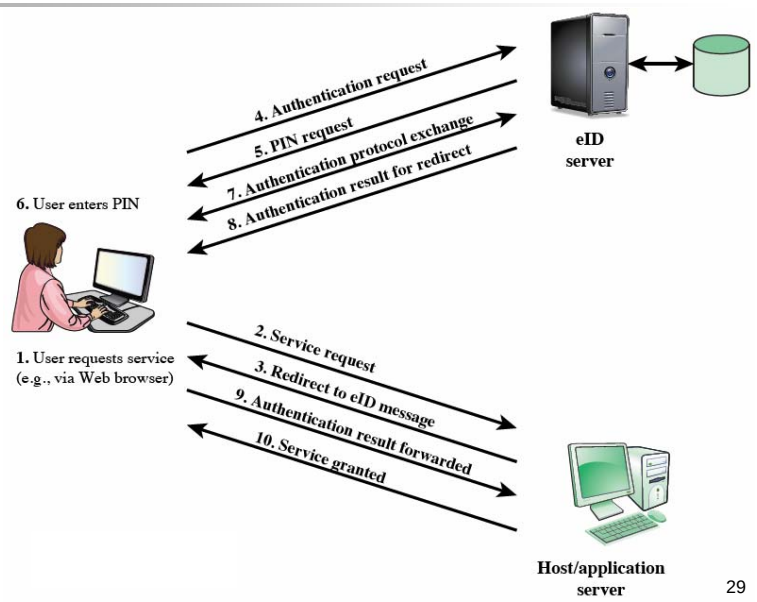
- 스마트 카드를 주민 식별카드로 사용
  - 정부와 상업적 용도로 사용 – 운전면허증, 주민등록증 등 ..
  - 강력한 신원 인증용으로 다양한 응용에 사용 가능
- (예) 독일의 eID (2010년부터 도입됨)의 포함 정보
  - 개인 데이터, 문서(카드)번호,
  - 카드 접근 번호(Card access number: CAN)
  - 기계 판독 영역(Machine readable zone: MRZ)



Electronic Functions and Data for eID Cards

Function	Purpose	PACE Password	Data	Uses
ePass (mandatory)	Authorized offline inspection systems read the data	CAN or MRZ	Face image; two fingerprint images (optional), MRZ data	Offline biometric identity verification reserved for government access
eID (activation optional	Online applications read the data or acess functions as authorized	eID PIN	Family and given names; artistic name and doctoral degree: date and place of birth; address and community ID; expiration date	Identification; age verification; community ID verification; restricted identification (pseudonym); revocation query
	Offline inspection systems read the data and update the address and community ID	CAN or MRZ		
eSign (certificate optional	A certification authority installs the signature certificate online	eID PIN	Signature key; X.509 certificate	Electronic signature creation
	Citizens make electronic signature with eSign PIN	CAN		

User Authentication with eID



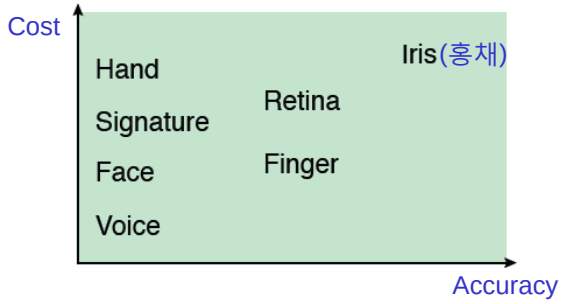
Password Authenticated Connection Establishment (PACE)

- 명시적 접근 제어 없이는 eID 카드의 비접촉식 RF 칩을 읽을 수 없도록 보장
- 온라인 응용 프로그램의 경우
  - 사용자가 6 자리 PIN (eID PIN, 카드 소지자에게만 알려야 함)을 입력하여 접근이 이루어짐
- 오프라인 응용 프로그램의 경우
  - 카드 뒷면에 인쇄 된 MRZ 또는 앞면에 인쇄된 6자리 카드 접근 번호 (CAN)를 사용.

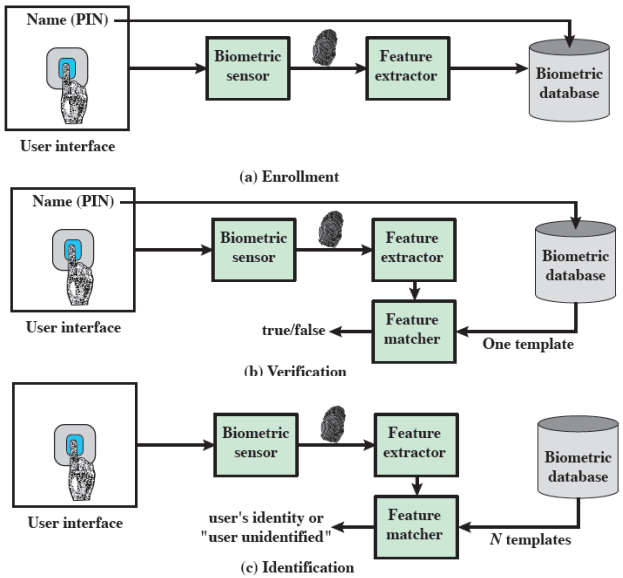
Biometric Authentication

- 신체의 특징을 사용하여 개인을 인증
  - 패턴 인식을 기반으로 함
- 기술적으로 복잡하고 비용이 많이 소요됨
  - 비밀번호와 토큰 방법과 비교할 때에
- 물리적 특성
  - 정적 특성 –얼굴, 지문, 손 모양, 망막/홍채 패턴
  - 동적 특성 – 서명, 음성

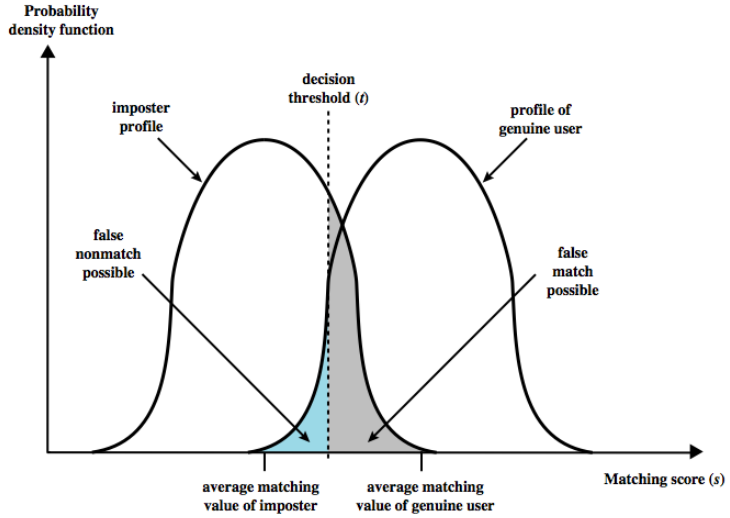
Cost Versus Accuracy



Operation of a Biometric System



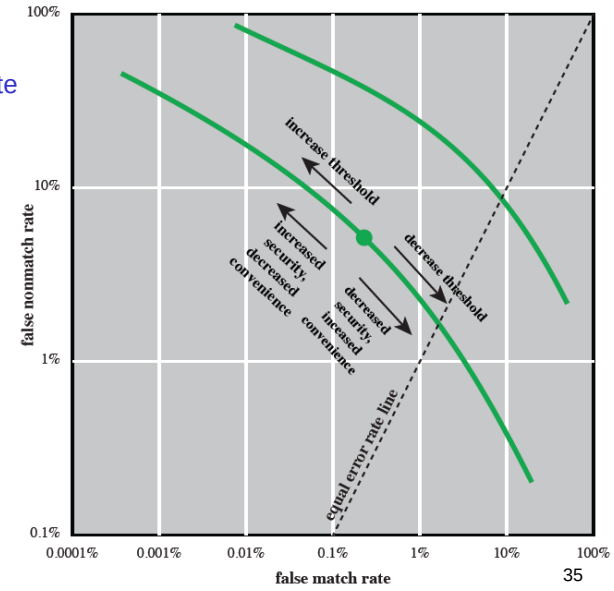
Biometric Accuracy



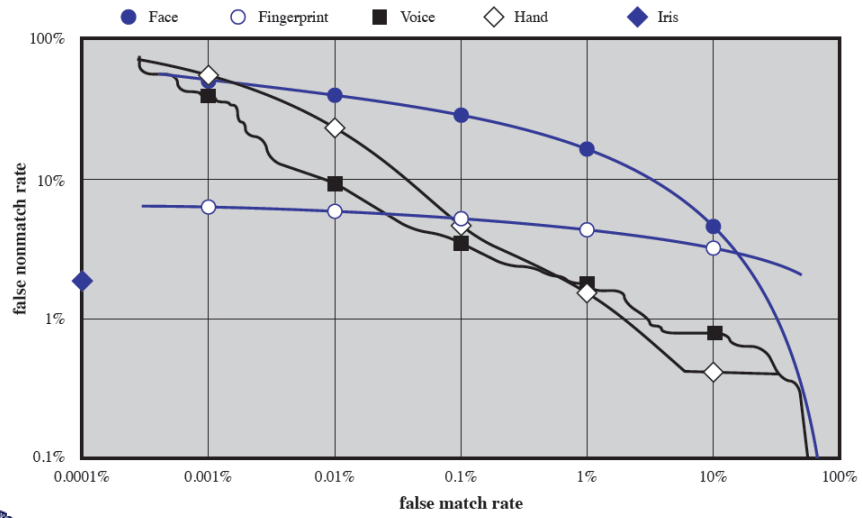
Idealized Biometric Measurement Operating Characteristic Curves

false non-match rate
- security

- 편의성
false match rate



Actual Biometric Measurement Operating Characteristic Curves



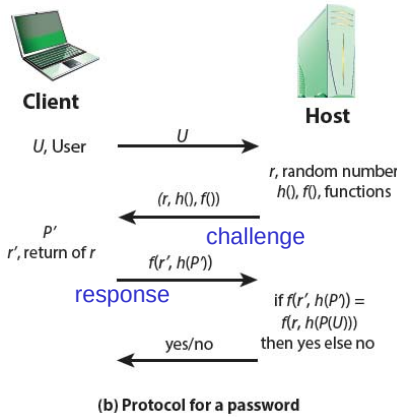
홍채는200만 이상의 비교에서 false match가 없음

Remote User Authentication

- 네트워크, 인터넷, 통신링크를 통한 사용자 인증은 더 복잡함
  - 도청, 암호 캡쳐, 관찰된 인증 시퀀스 재전송(replay)
- challenge-response 프로토콜
  - 이러한 위협에 대처하기 위하여 사용되는 방법

  - **비밀번호** 기반 프로토콜
  - **토큰** 기반 프로토콜
  - **정적 생체정보** 프로토콜
  - **동적 생체정보** 프로토콜

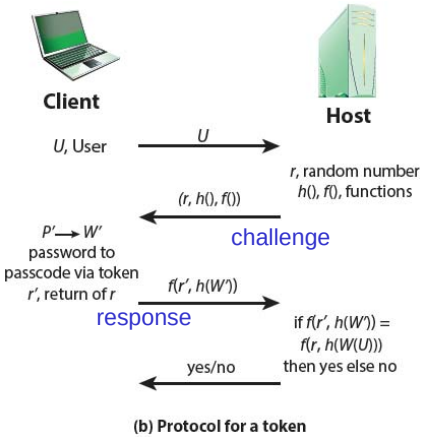
Password Protocol



(b) Protocol for a password

- U : 사용자 신원 정보(id)
- r : 일회용 난수 (**nonce**라고 부름)
  - 재전송 공격 방지
- 호스트는 사용자 암호 P의 해시 코드를 h(P) 저장함
- **challenge**: 호스트는 r과 함께 해시 함수 h와 함수 f 정보를 전송
  - 함수 f는 h(P)를 인수에 포함
- **response**: 사용자는 f(r', h(P')) 전송
  - 암호/암호의 해시코드 보호
  - r'=r, P'=P(U)임
- 호스트는 f(r, h(P(U))와 사용자 응답 f(r',h(P'))를 비교하여 사용자 인증

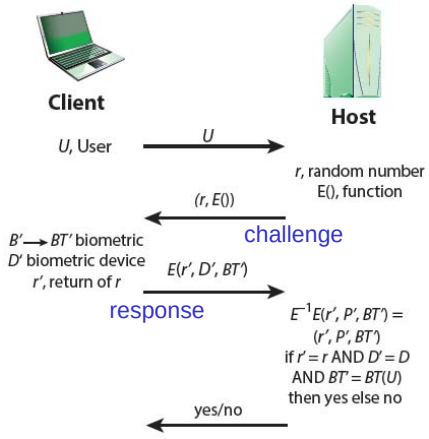
Token Protocol



(b) Protocol for a token

- passcode W' – 토큰이 생성한 값
  - 저장된 정적 패스코드 제공
  - 또는 일회용 난수 생성 (일회용 난수는 호스트와 동기화 되어야 함)
- 사용자가 비밀번호를 입력하여 passcode를 활성화
  - 비밀번호는 사용자와 token 간에 공유하여 호스트와 무관.
- 호스트는 f(r,h(W(U))와 사용자 응답 f(r', h(W'))을 비교하여 사용자 인증

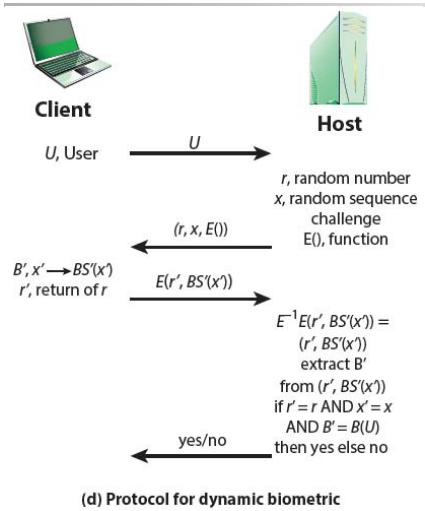
Static Biometric Protocol



(c) Protocol for static biometric

- E() – 암호화 식별자
- B – 사용자 생체 인식
  BT' – 생체인식 정보
- D' – 생체인식장비 식별자
- **challenge –** 호스트는 r과 E() 제공
- **response** – 사용자는 r', D', BT' 정보를 암호화 전송
- 호스트는 복호화후 저장된 정보 r, D, BT(U)와 비교하여 사용자 인증

Dynamic Biometric Protocol



(d) Protocol for dynamic biometric

- **challenge** – 호스트는 r, E() 이외에 랜덤시퀀스 challenge x를 전송
  - 숫자, 문자, 단어 시퀀스
- **response** – 주어진 시퀀스 x'에 따라서 생체신호 시퀀스 BS(x')를 생성하여 r'과 함께 암호화 전송
  - 음성, 타이핑, 필체 정보
- 호스트는 복호화 후 BS'(x)에서 B'를 추출하고, r과 B(U)와 비교하여 사용자 인증

Authentication Security Issues

- Client attack
  - 호스트나 통신 경로에 대한 접근 없이 합법적인 사용자를 가장하여 사용자 인증을 시도
- Host attack
  - 호스트에 있는 암호, 토큰 passcode, 생체인식 template에 접속
- Eavesdropping
  - 사용자 관찰 등의 방법으로 비밀번호를 탐지하려고 시도
- Replay attack
  - 이전에 수집한 사용자 응답을 반복하여 사용
- Trojan horse
  - 합법적 사용자로 가장한 응용프로그램, 하드웨어 장치를 이용하여 사용자 입력 정보를 캡쳐함.
- Denial of Service
  - 다량의 인증 시도를 하여 사용자 인증 서비스를 불가능하게 함

Attacks	Authenticators	Examples	Typical defenses
Client attack	Password	Guessing, exhaustive search	Large entropy; limited attempts
	Token	Exhaustive search	Large entropy; limited attempts, theft of object requires presence
	Biometric	False match	Large entropy; limited attempts
Host attack	Password	Plaintext theft, dictionary/exhaustive search	Hashing; large entropy; protection of password database
	Token	Passcode theft	Same as password; 1-time passcode
	Biometric	Template theft	Capture device authentication; challenge response

Eavesdropping, theft, and copying	Password	"Shoulder surfing"	User diligence to keep secret; administrator diligence to quickly revoke compromised passwords; multifactor authentication
	Token	Theft, counterfeiting hardware	Multifactor authentication; tamper resistant/evident token
	Biometric	Copying (spoofing) biometric	Copy detection at capture device and capture device authentication
Replay	Password	Replay stolen password response	Challenge-response protocol
	Token	Replay stolen passcode response	Challenge-response protocol; 1-time passcode
	Biometric	Replay stolen biometric template response	Copy detection at capture device and capture device authentication via challenge-response protocol
Trojan horse	Password, token, biometric	Installation of rogue client or capture device	Authentication of client or capture device within trusted security perimeter
Denial of service	Password, token, biometric	Lockout by multiple failed authentications	Multifactor with token