

10장 개체 인증(Entity Authentication)

정보보호이론

Spring 2015

공지 Quiz 2

■ Quiz 2

- × 5/19일 오후 2시~2시 20분
- × 중간고사 이후~오늘 수업 분

■ Mid-Term II(재시험)

- × 선택사항 (중간고사 I과 비교하여 좋은 점수 반영)
- × 5/23일 오후 2시~5시
- × 오늘 수업 분까지

■ Final Exam

- × 6/16일 오후 2시~5시
- × 이번 학기 수업 분

10.1 개요

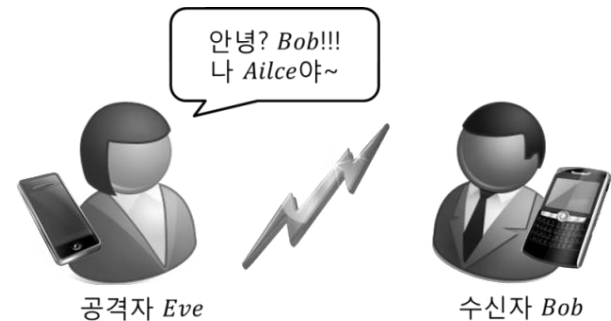
■ 개체인증

✕ 개체의 신원을 증명하기 위한 일련의 과정

▶ 개체 : 사람이나 기기

✕ 인증 정보

- ▶ What you are (voice, fingerprint, Iris)
- ▶ What you know (password)
- ▶ What you have (smart card, token card)



10.2 패스워드 방식

■ 미 AOL 취약 비밀번호 25가지

올해 해커들에게 쉽게 노출된 최악의 비밀번호 25

1 password	2 123456	3 12345678	4 qwerty	5 abc123
6 monkey	7 1234567	8 letmein	9 trustno1	10 dragon
11 baseball	12 111111	13 iloveyou	14 master	15 sunshine
16 ashley	17 bailey	18 passwOrd	19 shadow	20 123123
21 654321	22 superman	23 qazwsx	24 michael	25 football

해킹 취약 비밀번호는?

10.2 패스워드 방식

■ 패스워드 : low entropy

✗ 64-비트 패스워드를 발견하기 위해서는 $< 2^{64}$ 필요.

■ 안전한 패스워드

✗ 국내: 방통위, KISA : 패스워드 선택 및 이용 안내서

▶ 세가지 종류 이상의 문자구성으로 8자리 이상의 길이로 구성된 문자열 (2.148×10^{14}) OR 두 가지 종류 이상의 문자구성으로 10자리 이상의 길이로 구성된 문자열 (3.555×10^{15}) (문자종류는 알파벳 대문자와 소문자, 특수문자, 숫자 4가지)

▶ 안전한 패스워드는 △제3자가 쉽게 추측할 수 없는 패스워드 △패스워드 전송·저장 시 암호화 기준을 충족해야 한다

✗ 해외: NIST 800-63 : Electronic Authentication Guideline (2006)

■ Refer [Strong passwords: How to create and use them.](#)

■ Example : I am 28 years old !

✗ How strong is yours? : [Password Checker](#)

10.2 패스워드 방식

■ 국내 현황

- ✕ 2011년 11월 30일 한국암호포럼이 개최한 ‘암호의 역할 워크숍’
- ✕ 회원가입이 가능한 공공기관 홈페이지 127개 가운데 111개 (87.4%)가 패스워드 안전성이 미흡
- ✕ 전체 조사 대상 기관 중 85%(108개)는 패스워드 구성이나 길이가 기준에 미치지 못했으며, 41.7%(53개)는 패스워드가 암호화되지 않고 전송돼 패스워드가 노출되는 것으로 나타났다. 그 밖에도 안전한 패스워드 기준은 명시하고 있지만 구현상 오류가 있는 경우가 3.7%(4개), 입력 가능한 패스워드 기준 자체를 명시하지 않는 경우도 7.4%(8개)

10.2 패스워드 방식

■ 고정된 패스워드



- ✗ 도청 등의 위협
- ✗ 패스워드 테이블의 유출 시 위험

10.2 패스워드 방식

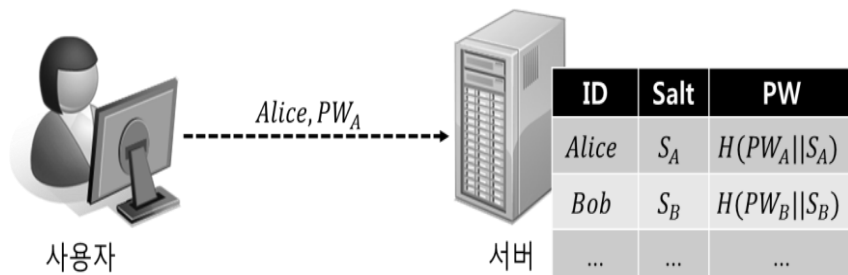
■ 해쉬된 패스워드



- ✗ 패스워드 테이블의 유출 시 해쉬 함수의 역상저항성으로 인하여 안전
 - ▶ 특정인 Alice의 패스워드를 알기 위해서는 $O(2^n)$ 번의 해쉬 평가 (n : 해쉬 함수의 출력 길이)
 - ▶ 임의의 사용자의 패스워드를 알기 위해서는 offline 사전공격 (dictionary attack)이 효과적
- 추측된 패스워드 PW의 해쉬값 $H(PW)$ 와 패스워드 테이블의 모든 해쉬값과 비교

10.2 패스워드 방식

■ 솔트(Salt) 사용



- ✗ 임의의 사용자의 패스워드를 알기 위해서는 offline 사전공격 (dictionary attack)을 방어
 - ▶ 추측된 패스워드 PW의 해쉬값 H(PW)와 패스워드 테이블의 해쉬값과 직접 비교 불가능
 - ▶ 모든 사용자 ID에 대하여 H(PW||S_{ID})와 테이블의 해쉬값과 비교해야 함
 - ID의 개수가 t 인 경우, 추측된 PW에 대하여 t 번씩 증가
 - ▶ 솔트가 공개된 경우, 특정인 Alice의 PW를 알기 위한 계산은 변동없음
 - 여전히 $O(2^n)$ 번의 해쉬 평가 (n : 해쉬 함수의 출력 길이)

10.2 패스워드 방식

■ OTP(One-Time Password)

✕ 매번 다른 난수 사용

- ▶ 사전 공격(Dictionary Attack)이나 재전송 공격(Replay Attack) 등으로 부터 안전

✕ Example : RSA SecureID

✕ Two factor authentication

✕ It has been hacked

✕ 美RSA가 연이은 해킹으로 파문이 일자 자사 일회용비밀번호(OTP)제품인 '시큐어ID' 4천만대를 전면 리콜 조치

- ▶ 최근 '시큐어ID'는 연이은 해킹악재에 시달렸다. 지난 3월 OTP 소스코드 유출을 시작으로 최근에는 美군부의 심장이라 불리는 세계 최대 방위산업체인 록히드마틴 전산망 해킹에도 '시큐어ID'가 침입에 활용된 것으로 알려져 사용자들의 신뢰를 잃었다.
- ▶ '시큐어ID'는 현재 국내 금융권 및 주요기업을 중심으로 100만명 이상이 이용중

10.2 패스워드 방식

- 동기화 방식의 일회용 패스워드(Synchronized OTP)
 - ✗ 사용자와 서버는 시드(Seed)를 공유 후, 동일한 패스워드 생성
 - ✗ 시간 동기화 방식
 - ▶ $sk = h(seed, T)$: current time T
 - ▶ 적절한 오차 허용 → 시간 구간 설정
 - ✗ 이벤트 동기화 방식
 - ▶ $sk = h(seed, C)$: counter C
 - ▶ 전송 오류 시 C 동기화 필요
 - ✗ Hybrid 동기화 방식
 - ▶ 한 구간 내에 여러 번의 패스워드 생성, 카운터 값 증가
 - ▶ 각 구간마다 카운터 값 초기화

10.2 패스워드 방식

■ 동기화 방식의 일회용 패스워드(Synchronized OTP)

✕ 패스워드 업데이트 방식

1. 사용자와 서버는 초기 패스워드 P_1 사전 공유
2. 사용자 \rightarrow 서버 : $E_{P_1}(P_2)$
3. 서버는 $E_{P_1}(P_2)$ 을 복호화한 후 P_2 를 획득. 두 번째 접속 시 P_2 를 패스워드로 사용
4. -----
5. $E_{P_k}(P_{k+1})$

10.2 패스워드 방식

■ 비동기화 방식의 일회용 패스워드(Non-Synchronized OTP)

✖ 질의-응답(Challenge-Response) 방식 (10.3절)

▶ 동기화 불필요, 통신량 증가

✖ Lamport 방식 : 해쉬 체인(Hash chain) 사용

1. 사용자는 비밀값 x 를 생성, 서버의 접근 횟수 제한을 k
2. $x, h(x) = x_1, h(h(x)) = x_2, \dots, h^k(x) = x_k$
3. 사용자와 서버는 초기 값 x_k 공유
4. 사용자 \rightarrow 서버 : x_{k-1}
5. 서버는 $h(x_{k-i}) = x_k$ 검증 후 x_{k-1} 저장
6. -----
7. 사용자 \rightarrow 서버 : x_{k-i}

10.2 패스워드 방식

■ 스마트 OTP

✕ 등록 과정

- ▶ 금융회사 창구에서 대면해 IC카드 OTP 생성 키를 받고, NFC 기능이 있는 스마트폰에 해당 앱을 다운받아 이용

✕ 동작 과정



① 카드를 스마트폰에 접촉



② OTP 생성



③ OTP 자동입력
(뱅킹 이체 거래 시)

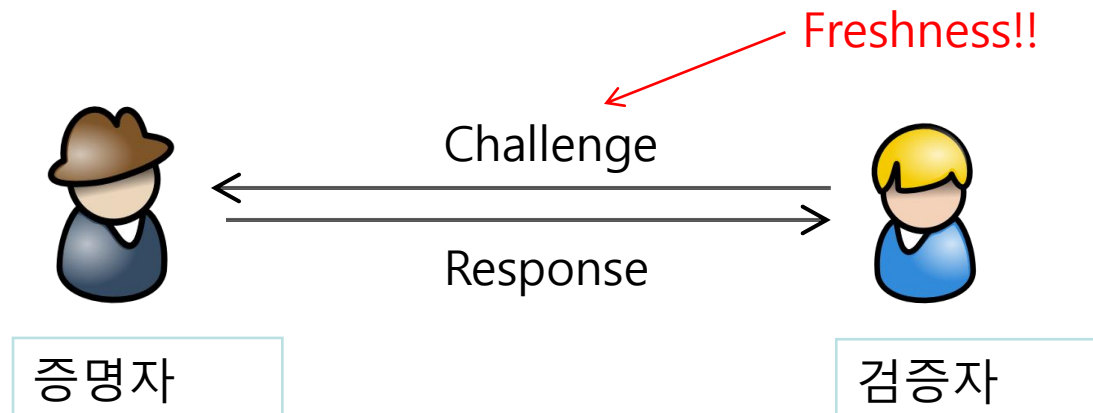
10.2 패스워드 방식

SMS OTP: Bank of America



10.3 질의-응답(Challenge-Response) 인증

- 검증자가 생성한 질의에 대하여 증명자가 응답



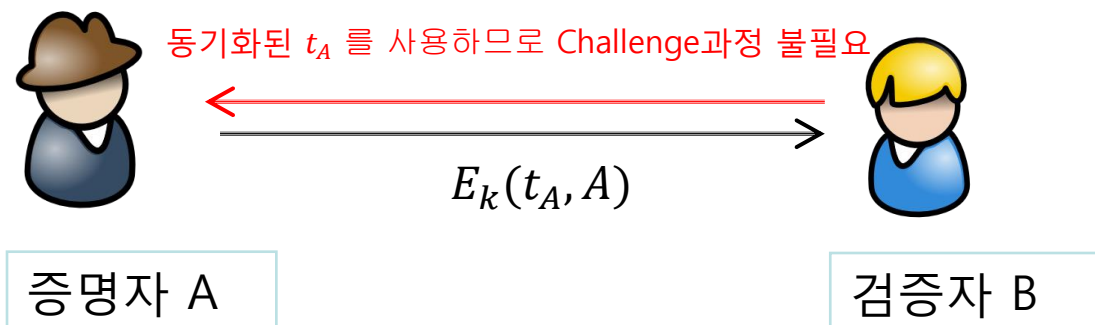
✗ 대칭키를 이용한 방식 & 공개키를 이용한 방식

10.3 질의-응답(Challenge-Response) 인증

■ 대칭키를 이용한 질의-응답 인증

✕ 타임스탬프를 이용한 단방향 인증

1. A: 타임스탬프 t_A 생성
2. $A \rightarrow B: E_k(t_A, A)$ { E_k 는 A와 B가 사전 공유된 k 로 암호}
3. B: $D_k(E_k(t_A, A))$ 후, t_A 가 현재시간 구간에 들어오는지 확인

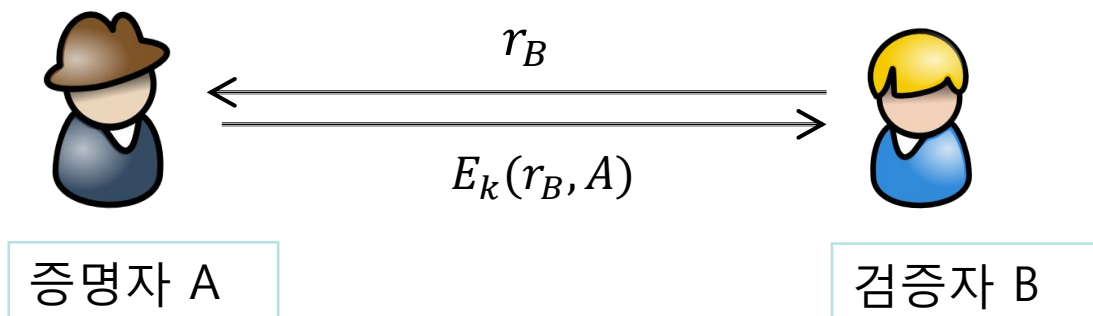


10.3 질의-응답(Challenge-Response) 인증

■ 대칭키를 이용한 질의-응답 인증

✕ 난수(Nonce)를 이용한 단방향 인증

1. $B \rightarrow A : r_B \{\text{challenge}\}$
2. $A \rightarrow B : E_k(r_B, A) \{E_k \text{는 A와 B가 사전 공유된 } k \text{로 암호}\}$
3. 검증자 B: $D_k(E_k(r_B, B))$ 후, r_B 확인

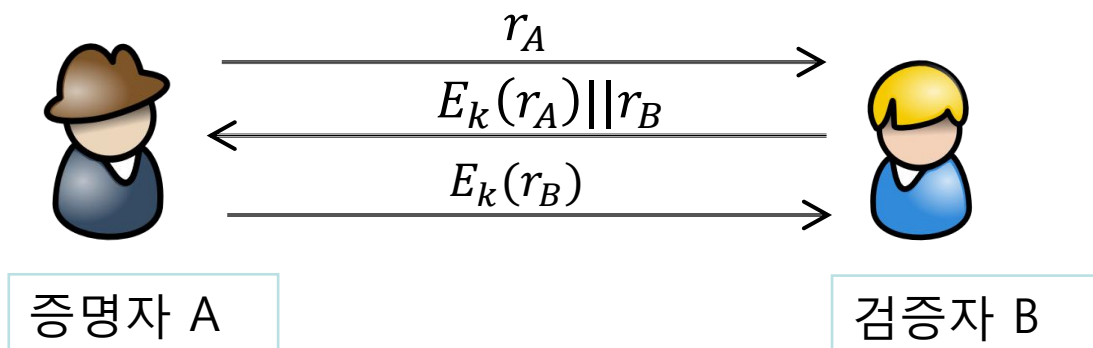


10.3 질의-응답(Challenge-Response) 인증

■ 대칭키를 이용한 질의-응답 인증

✕ 난수(Nonce)를 이용한 양방향 인증

1. $A \rightarrow B : r_A$ {A의 challenge}
2. $B \rightarrow A : E_k(r_A) || r_B$ {B의 challenge}
3. $A : D_k(E_k(r_A))$ 후, r_A 확인
4. $A \rightarrow B : E_k(r_B)$
5. $B : D_k(E_k(r_B))$ 후, r_B 확인

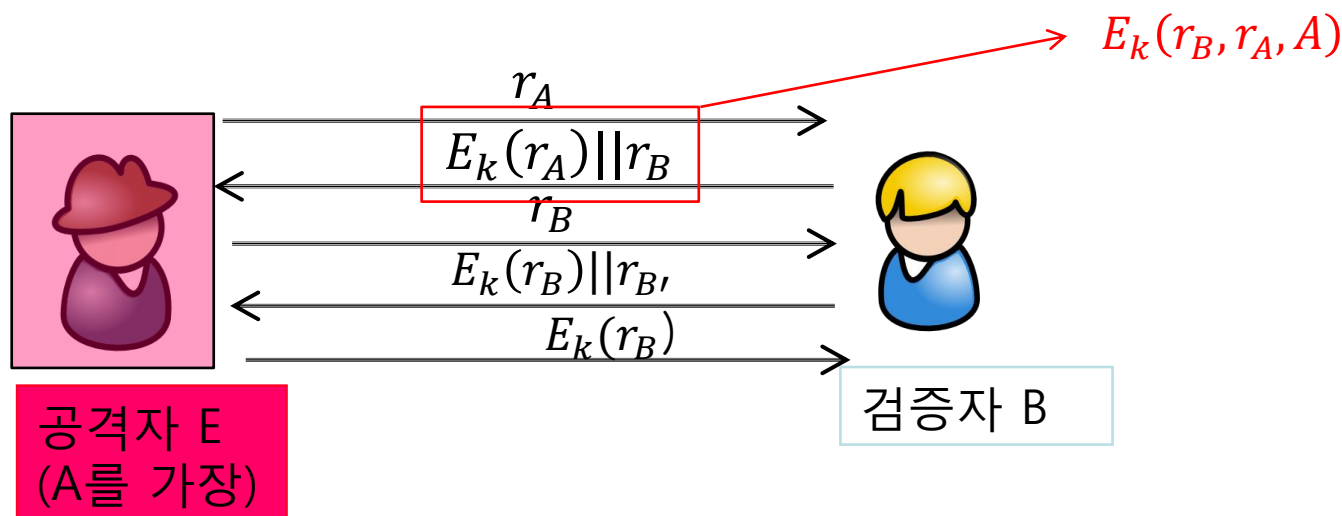


10.3 질의-응답(Challenge-Response) 인증

■ 대칭키를 이용한 질의-응답 인증

✖ 반사공격 (Reflection Attack)

1. $A(E) \rightarrow B : r_A$ {A(E)의 challenge}
2. $B \rightarrow A(E) : E_k(r_A) || r_B$ {응답 & B의 challenge}
3. $A(E) \rightarrow B : r_B$ {E의 challenge, 두 번째 세션 open}
4. $B \rightarrow A(E) : E_k(r_B) || r_{B'}$ {응답 & B의 challenge}
5. $A(E) \rightarrow B$: 첫 번째 세션의 정당한 응답 $E_k(r_B)$ 전송

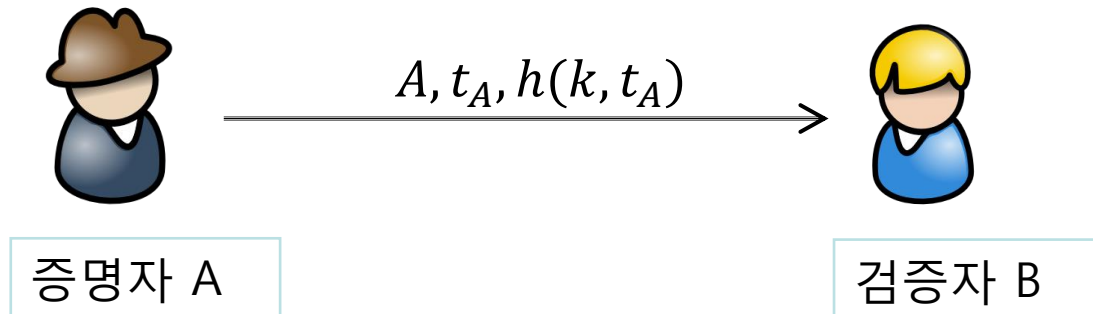


10.3 질의-응답(Challenge-Response) 인증

■ 해시 함수를 이용한 질의-응답 인증

✕ 해시 함수와 타임스탬프를 이용한 단방향 인증

1. $A \rightarrow B : A, t_A, h(k, t_A) \{A\text{의 response}\}$

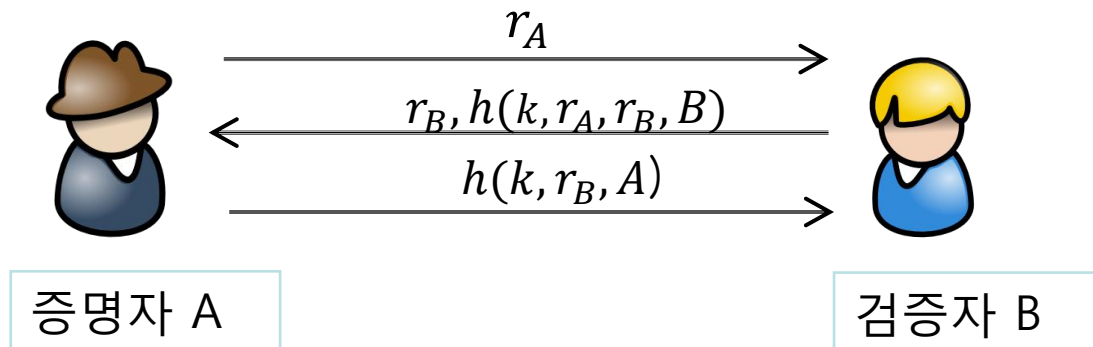


10.3 질의-응답(Challenge-Response) 인증

■ 해시 함수를 이용한 양방향 인증

✖ 난수 nonce)를 이용한 인증

1. $A \rightarrow B : r_A$ {A의 challenge}
2. $B \rightarrow A : r_B, h(r_A, r_B, B)$ {B의 응답 & challenge}
3. $A \rightarrow B : h(r_B, A)$

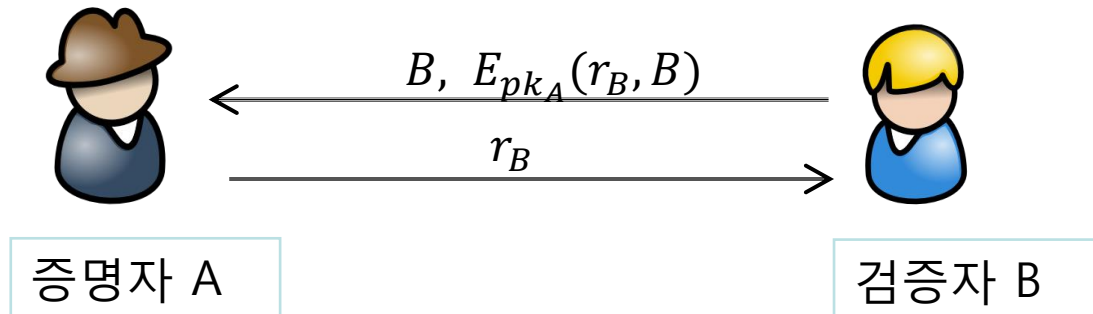


10.3 질의-응답(Challenge-Response) 인증

■ 공개키 암호를 이용한 단방향 인증

✕ 난수를 이용한 인증

1. $B \rightarrow A : B, E_{pk_A}(r_B, B)$ {B의 challenge}
2. $A \rightarrow B : r_B$

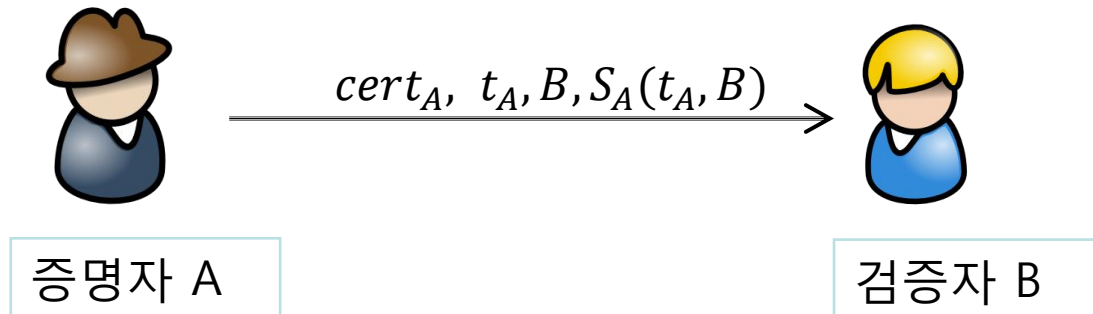


10.3 질의-응답(Challenge-Response) 인증

■ 전자 서명을 이용한 단방향 인증

✕ 타임스탬프를 이용한 단방향 인증

1. $A \rightarrow B : cert_A, t_A, B, S_A(t_A, B)$

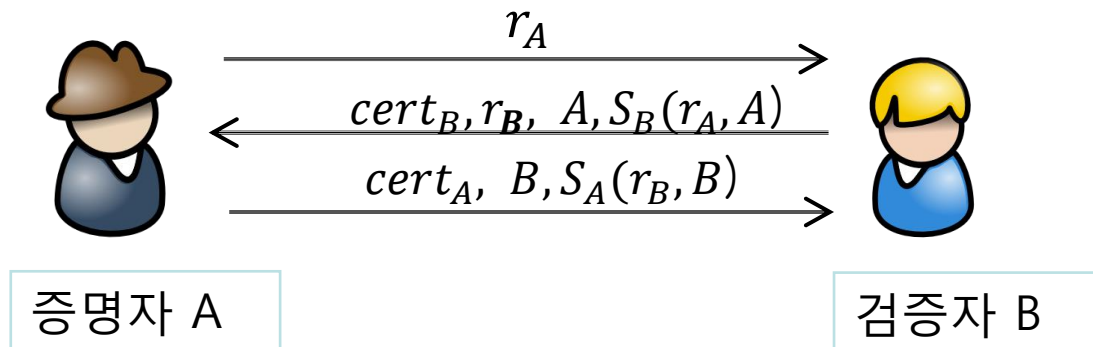


10.3 질의-응답(Challenge-Response) 인증

■ 전자 서명을 이용한 양방향 인증

✕ 난수를 이용한 양방향 인증

1. $A \rightarrow B : r_A \{A\text{의 challenge}\}$
2. $B \rightarrow A : cert_B, r_B, A, S_B(r_A, A) \{B\text{의 응답 \& challenge}\}$
3. $A \rightarrow B : cert_A, B, S_A(r_B, B) \{A\text{의 응답 \& challenge}\}$

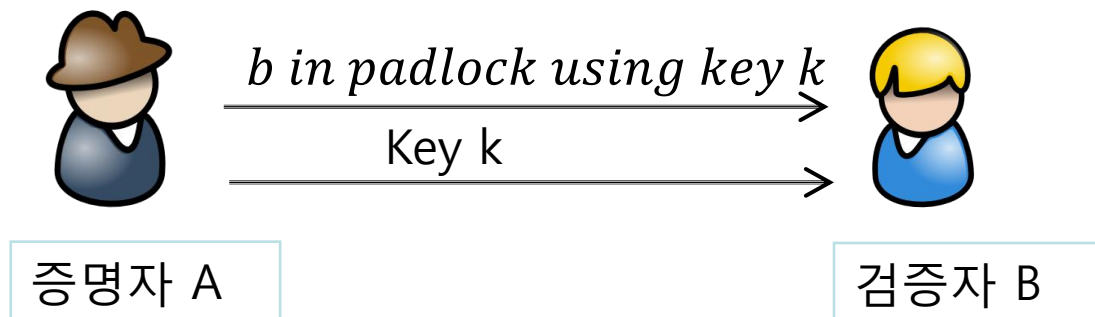


10.4 비트 약속(Bit Commitment)

- 비트 (0또는 1) 값에 대하여 약속(Commitment)을 하고 이후에 약속 값에 대하여 확인하는 방법

생성 $A \rightarrow B$: one bit b in padlock {commitment b 의 생성}

확인 $A \rightarrow B$: Key to open the padlock { b 의 확인}



✕ 비트 약속의 성질

1. **하이딩(Hiding)** : 검증자는 감춰진 비트의 내용에 대해서 알지 못해야 한다.
2. **바인딩(Binding)** : 증명자는 자신이 선택한 비트를 감춘 이후에 변경할 수 없다.

10.4 비트 약속(Bit Commitment)

■ 대칭키 암호를 이용한 비트 약속 생성

1. $B \rightarrow A : R$
2. $A \rightarrow B : E_k(R, b)$

확인

1. $A \rightarrow B : k$
2. B는 $E_k(R, b)$ 를 복호화하여 R, b 확인

✗ What if $E_k(b)$ is used instead of $E_k(R, b)$?

✗ 하이딩?

- ▶ k 없이 $E_k(R, b)$ 에서 commitment를 알 수 없다.

✗ 바인딩?

- ▶ $E_k(R, b) = E_{k'}(R, b')$ 을 만족하는 k' 을 찾는 것이 어려움

10.4 비트 약속(Bit Commitment)

■ 공개키 암호를 이용한 비트 약속

✕ 생성

1. A: 난수 x 의 최하위 비트를 약속 값 b 로 설정
2. $A \rightarrow B : E_{pk}(x)$

✕ 확인

1. $A \rightarrow B : sk$
2. B는 $E_{pk}(x)$ 를 복호화하여 x 의 최하위 비트를 확인

10.5 동전 던지기(Fair Coin Flipping)

■ 해쉬 함수를 이용한 동전 던지기

1. A는 선택한 x 에 대한 해쉬 값 $y = h(x)$ 를 계산
2. $A \rightarrow B : y$
3. B는 x 에 대한 최하위 비트를 추측
4. $A \rightarrow B : x$
5. B는 $y = h(x)$ 를 계산한 후 1단계에서 받은 값과 비교

✖ 하이딩 :

- ▶ 해쉬 함수의 역상 저항성(Preimage Resistance)으로 y 를 보고 x 를 추측할 수 없음

✖ 바이딩

- ▶ 충돌 저항성(Collision Resistance)으로 $\mathbf{lsb}(x) \neq \mathbf{lsb}(x') \wedge h(x) = h(x')$ 를 만족하는 x' 을 발견하기 어려움

10.5 동전 던지기(Fair Coin Flipping)

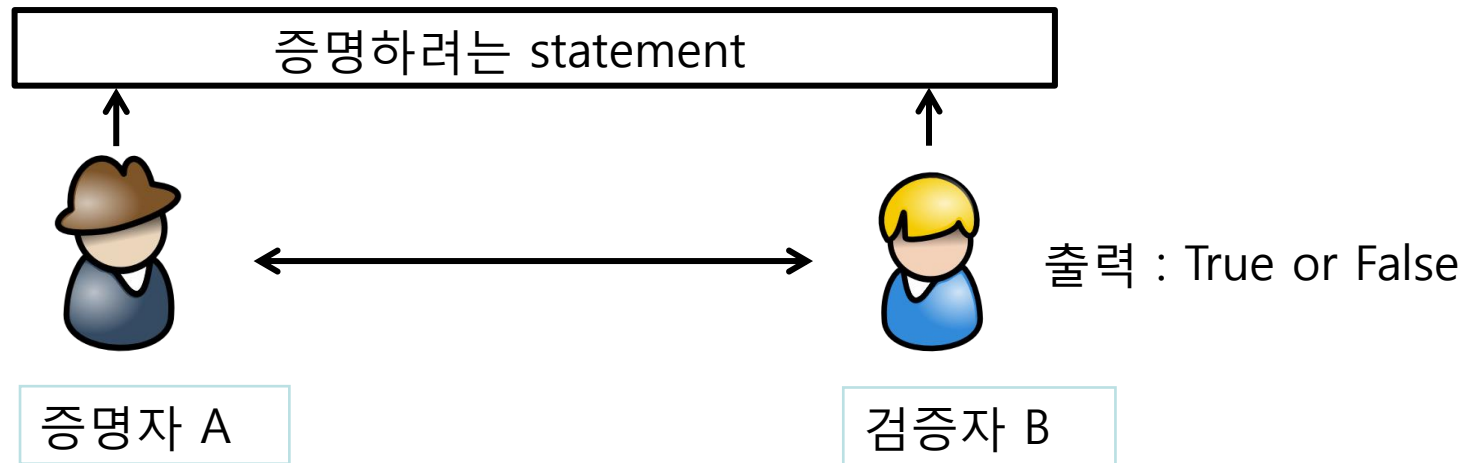
■ 가환성을 가진 공개키 암호를 이용한 동전 던지기

$$\times D_{k_2}(E_{k_1}(E_{k_2}(m))) = E_{k_1}(m)$$

1. $A \rightarrow B : E_A(m_0), E_A(m_1)$
2. B는 $E_A(m_0), E_A(m_1)$ 에서 하나를 선택하여 자신의 공개키로 암호 $E_B(E_A(m_x))$
3. $B \rightarrow A : E_B(E_A(m_x))$
4. $A \rightarrow B : E_B(m_x)$
5. B는 $E_B(m_x)$ 를 복호화한 후 m_x 확인
6. $A \rightarrow B : A$ 의 (공개키, 개인키) {자기 법 집행(Self-Enforcing)}

10.6 영지식 인증(Zero-Knowledge Authentication)

■ 상호 증명 시스템(Interactive Proof System)



✕ 상호 증명 시스템의 특성

- ▶ **완전성(Completeness)**: 문장이 True라는 것을 정직한 증명자가 알고 있다면 검증자는 True를 출력
- ▶ **건전성(Soundness)**: 검증자가 True를 출력한 경우, 문장은 True임

10.6 영지식 인증(Zero-Knowledge Authentication)

■ 영지식 증명 시스템(Zero-Knowledge Interactive Proof System)

✕ 영지식 상호 증명 시스템의 특성

- ▶ **완전성(Completeness)**: 문장이 True라는 것을 정직한 증명자가 알고 있다면 검증자는 True를 출력
- ▶ **건전성(Soundness)**: 검증자가 True를 출력한 경우, 문장은 True임
- ▶ **영지식성(Zero-Knowledgeness)**: 증명자가 문장이 True라는 사실 이외에 어떠한 정보를 노출시킴 없이 문장을 검증자에게 확신시키는 성질
 - 검증자가 True인 문장을 입력 받아 증명자와 교신하는 과정에서 얻은 모든 정보는 영지식 증명 이전에 증명자의 도움 없이 생성할 수 것

10.6 영지식 인증(Zero-Knowledge Authentication)

■ 도전-응답은 영지식 증명 시스템?

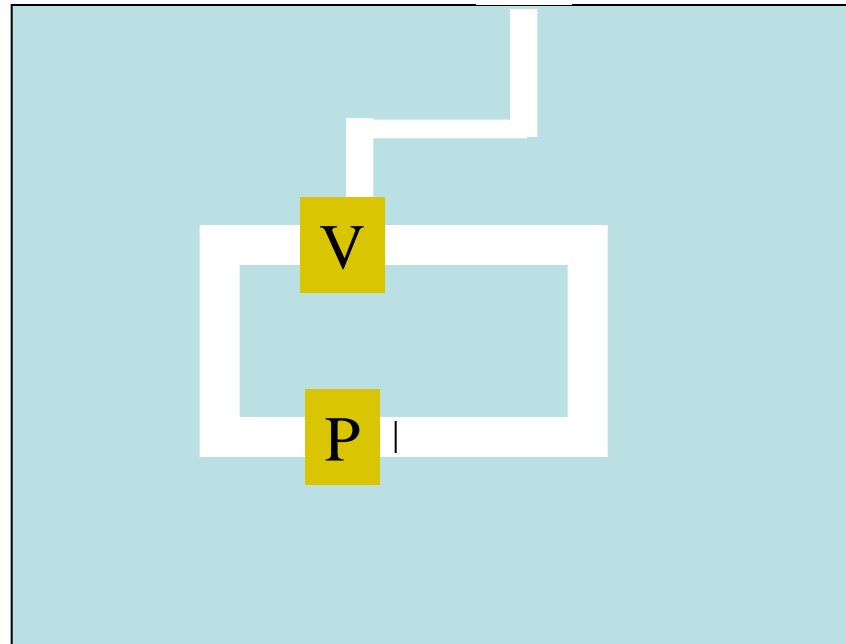
1. $B \rightarrow A : C = E_A(M)$
2. $A \rightarrow B : M$

✗ What if an attacker uses the verifier?

1. $B \rightarrow A : C = E_A(M) \{ \text{B의 challenge가 아니라 } M \text{을 알기 위함} \}$
2. $A \rightarrow B : M$

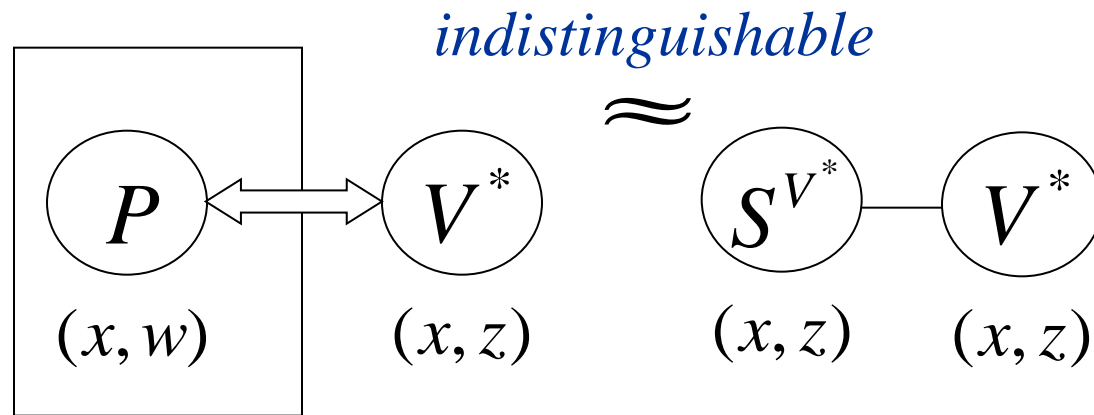
10.6 영지식 인증(Zero-Knowledge Authentication)

■ Basic ZKIP



10.6 영지식 인증(Zero-Knowledge Authentication)

■ Proof of Zero-Knowledgeness of ZKIP



< Real System >

10.6 영지식 인증(Zero-Knowledge Authentication)

■ Fiat-Shamir 프로토콜

✕ History--secrecy order by Patent Office (1986)

✕ 초기 설정 과정

1. 큰 소수 p 와 q 를 선택한 후 $n = p \times q$ 을 계산
2. $\gcd(s, n) = 1$ 와 $1 \leq s \leq n - 1$ 인 비밀키 s 를 선택
3. $v \equiv s^2 \pmod n$ 을 계산. (v, n) 를 공개키로 사용

✕ 인증 과정

1. A는 $1 \leq r \leq n - 1$ 인 r 을 선택
2. $A \rightarrow B : x \equiv r^2 \pmod n$
3. $B \rightarrow A : e \in \{0, 1\}$
4. $A \rightarrow B : y \equiv r \cdot s^e \pmod n$
5. 검증자는 $y^2 \equiv x \cdot v^e \pmod n$ 확인

10.6 영지식 인증(Zero-Knowledge Authentication)

■ Fiat-Shamir 프로토콜

✘ 공격자가 e 를 추측하는 경우

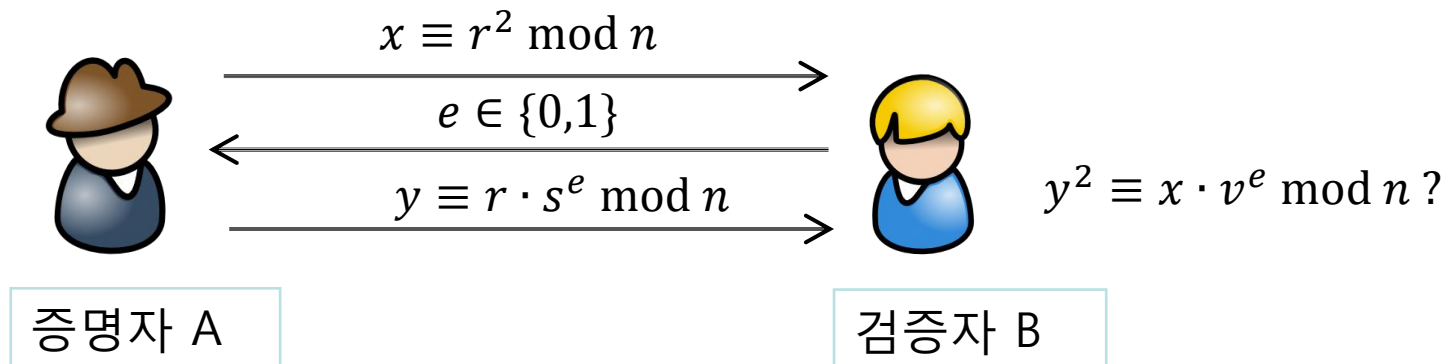
Case 1: $e = 0$ 인 경우 $x \equiv r^2 \pmod n$ 와 $y = r$ 전송

검증식 통과: $x \cdot v^0 = r^2 = y^2$

Case 2: $e = 1$ 인 경우 $x = r^2/v \pmod n$ 와 $y = r$ 전송

검증식 통과: $x \cdot v^1 = (r^2/v) \cdot v^1 = r^2 = y^2$

→ 50%, 반복!!!



10.6 영지식 인증(Zero-Knowledge Authentication)

■ Fiat-Shamir 프로토콜

✕ Security

- ▶ Hardness to find $\text{sqrt}(v)$ which is equivalent to factoring n .

✕ 완전성 : $y^2 = r^2 \cdot s^{2c} = x \cdot v^c$

✕ 건전성 : 정직한 증명자는 검증식에 통과되는 y_1 과 y_2 를 알고 있음

- ▶ $y_1 = r \cdot s^0$ $y_2 = r \cdot s$

- $y_1/y_2 = s$

- ▶ 따라서 증명자는 비밀 “ s ”를 알고 있음

- ▶ In formal proof, need to show that the prob. that y is cheated by dishonest P is negligible → running m iterations guarantees that the prob. is 2^{-m} , that is negligible.

10.6 영지식 인증(Zero-Knowledge Authentication)

■ Zero-Knowledge :

- (1) Choose $c = 0$ or 1 at random (Guess the challenge)
- (2) Choose r at random. If $c = 0$, then $x = r^2$ and output (x, c, r) .
If $c = 1$, then $x = r^2 / v$
- (3) Choose $c' = 0$ or 1 at random. If $c' = c$ then output (x, c, r) , else Goto Step (1)

Such (x, c, r) 's have a probability distribution which is indistinguishable from those generated by interacting with honest prover.

■ Impersonating Prover

Verifier cannot impersonate the prover since he cannot correctly guess “ c ”.

- ### ■ Running this “accreditation” t time results in the odd of fooling V in 2^t .

10.6 영지식 인증(Zero-Knowledge Authentication)

■ Feige-Fiat-Shamir 프로토콜

✖ 초기 설정 과정

1. 큰 소수 p 와 q 를 선택한 후 $n = p \times q$ 을 계산
2. $\gcd(s_i, n) = 1$ 와 $1 \leq s_i \leq n - 1$ 을 만족하는 비밀키 벡터 $s = \{s_1, s_2, \dots, s_k\}$ 를 선택
3. $v_i \equiv (s_i^2)^{-1} \pmod n$, ($v = \{v_1, v_2, \dots, v_k\}, n$) 를 공개키로 사용

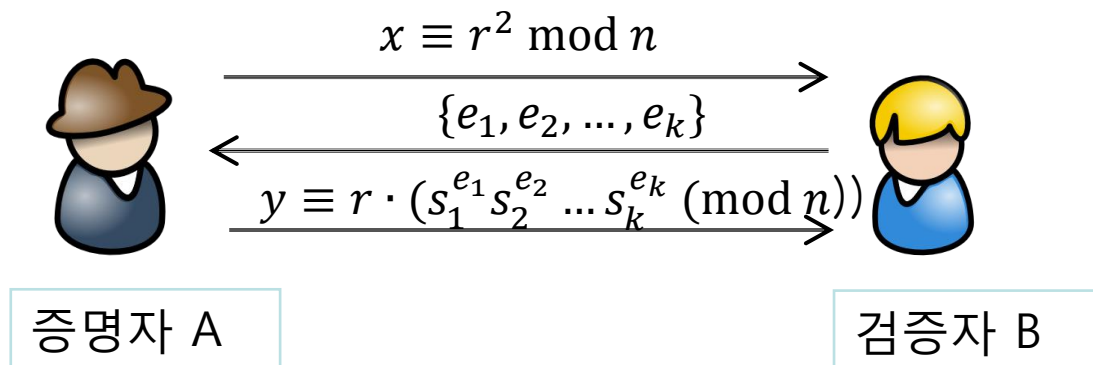
✖ 인증 과정

1. A는 $1 \leq r \leq n - 1$ 인 r 을 선택
2. $A \rightarrow B : x \equiv r^2 \pmod n$
3. $B \rightarrow A : e_i \in \{0, 1\}$ 인 $e = \{e_1, e_2, \dots, e_k\}$
4. $A \rightarrow B : y \equiv r \cdot (s_1^{e_1} s_2^{e_2} \dots s_k^{e_k} \pmod n)$
5. 검증자는 $y^2 v_1^{e_1} v_2^{e_2} \dots v_k^{e_k} \equiv x \pmod n$ 확인

10.6 영지식 인증(Zero-Knowledge Authentication)

■ Feige-Fiat-Shamir 프로토콜

- ✗ Fiat-Shamir 인증 기법을 순차적으로 k 번 수행한 것을 단 한번으로 평행하게(Parallel) 수행



$$y^2 v_1^{e_1} v_2^{e_2} \dots v_k^{e_k} \equiv x \pmod{n}?$$

10.6 영지식 인증(Zero-Knowledge Authentication)

■ Schnorr 프로토콜

✕ 초기 설정 과정

1. 큰 소수 q , $q|p-1$ 인 소수 p 선택
2. $a^q \equiv 1 \pmod{p}$ 를 만족하는 $a (\neq 1)$ 를 선택
3. $v \equiv a^{-s} \pmod{p}$, (v, a, p) 는 공개키, s 는 비밀

✕ 인증 과정

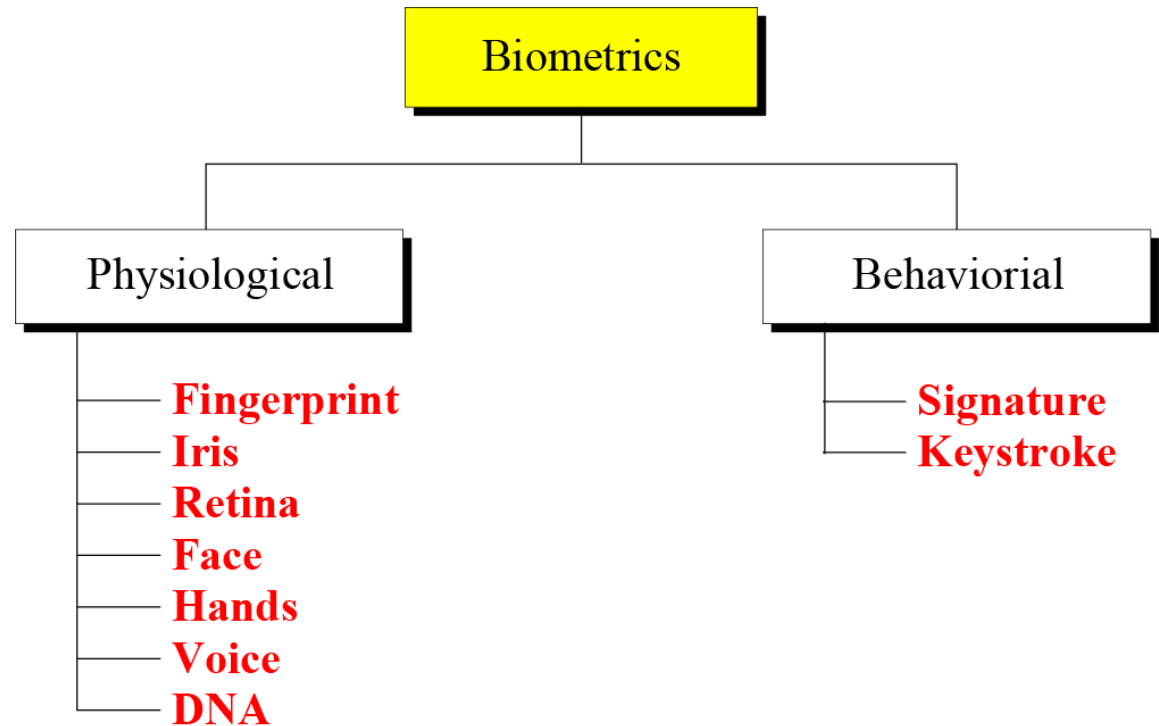
1. A는 $1 \leq r < q$ 인 r 을 선택
2. $A \rightarrow B: x \equiv a^r \pmod{p}$
3. $B \rightarrow A: 1 \leq e < 2^t$ 의 임의의 원소 e
4. $A \rightarrow B: y \equiv (r + e \cdot s) \pmod{q}$
5. 검증자는 $a^y \equiv x \cdot v^e \pmod{p}$ 확인

10.7 차세대 개체 인증

■ BIOMETRICS

✂ Accuracy of biometry techniques

- ▶ False Rejection Rate (FRR)
- ▶ False Acceptance Rate (FAR)

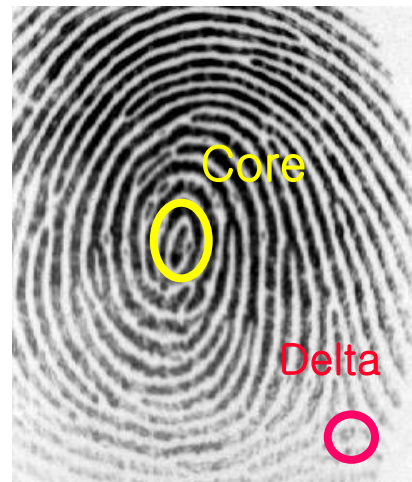
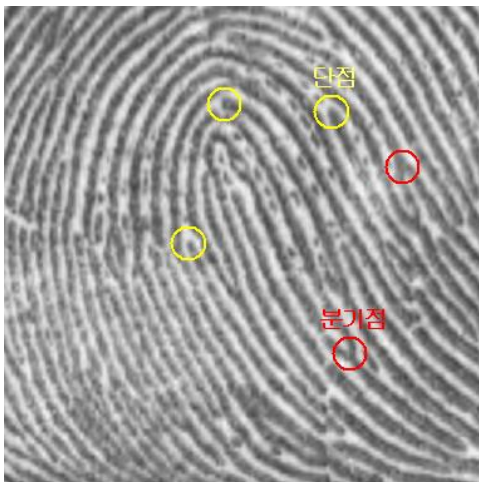


10.7 차세대 개체 인증

■ BIOMETRICS

✕ 지문

- ▶ 다른 두 손가락의 지문은 상이
- ▶ 지문의 모양은 평생 바뀌지 않음
- ▶ 특징점 추출 : 단점, 분기점, Core, Delta
- ▶ 단점 : 마모(화가), 장애인? 여성, 어린이, 노인, 땀이 있는 경우?
→ 다른 BIOMETRICS
- ▶ 사례 : 병기 및 탄약 관리 지문인식 잠금장치, “심플 패스”
Facebook 로그인



출처 : ETRI

10.7 차세대 개체 인증

■ BIOMETRICS

✕ FaceRecog

- ▶ 얼굴의 대칭적 구도, 생김새, 머리카락, 눈의 색상, 얼굴 근육의 움직임 등을 분석하여 얼굴의 특징 이용
- ▶ 사례 : G20서울 정상회의 기간에 얼굴인식 시스템

✕ Iris

- ▶ 사례 : 인도 12억 인구의 생체정보를 등록하는 전자주민등록 사업 진행 중
 - 아다르(Aadhaar)로 지문과 홍채를 기록해 신원을 확인할 수 있는 12자리의 고유 숫자를 부여하고, 전국 어디에서나 이동통신 기기를 통해 8초 안에 개인을 식별



✕ Vein

- ▶ 사례 : 일본 18개 은행을 비롯하여 일본우정공사 등에서 정맥인증을 이용한 ATM

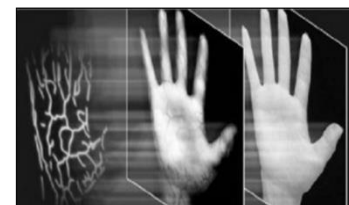


그림 4 지정맥 인증 장치의 ATM에 탑재 사례

10.7 차세대 개체 인증

■ BIOMETRICS

✕ Voice

- ▶ 미리 기록해 둔 음성 패턴과 비교해 개인 인증
- ▶ 사례 : 법무부 보호관찰소, 음성인식 본인확인 시스템 구축



✕ 손 모양

- ▶ 기기상에 올려놓은 손 모양에 대하여 상대적인 거리와 각도 등을 측정 후 저장해 놓은 자신의 바이오 정보와 비교하는 기술 → 높은 신뢰성 제

✕ 서명

- ▶ 이미 작성된 서명을 인식하는 정적인 방법
- ▶ 서명하는 과정을 동적으로 파악하는 방법
- ▶ 서명시간, 속도, 종이로부터 펜이 떨어진 횟수 등

출처 : “차세대 바이오인증
- ICT Standardization
Strategy”, TTA, 2011

✕ 걸음걸이

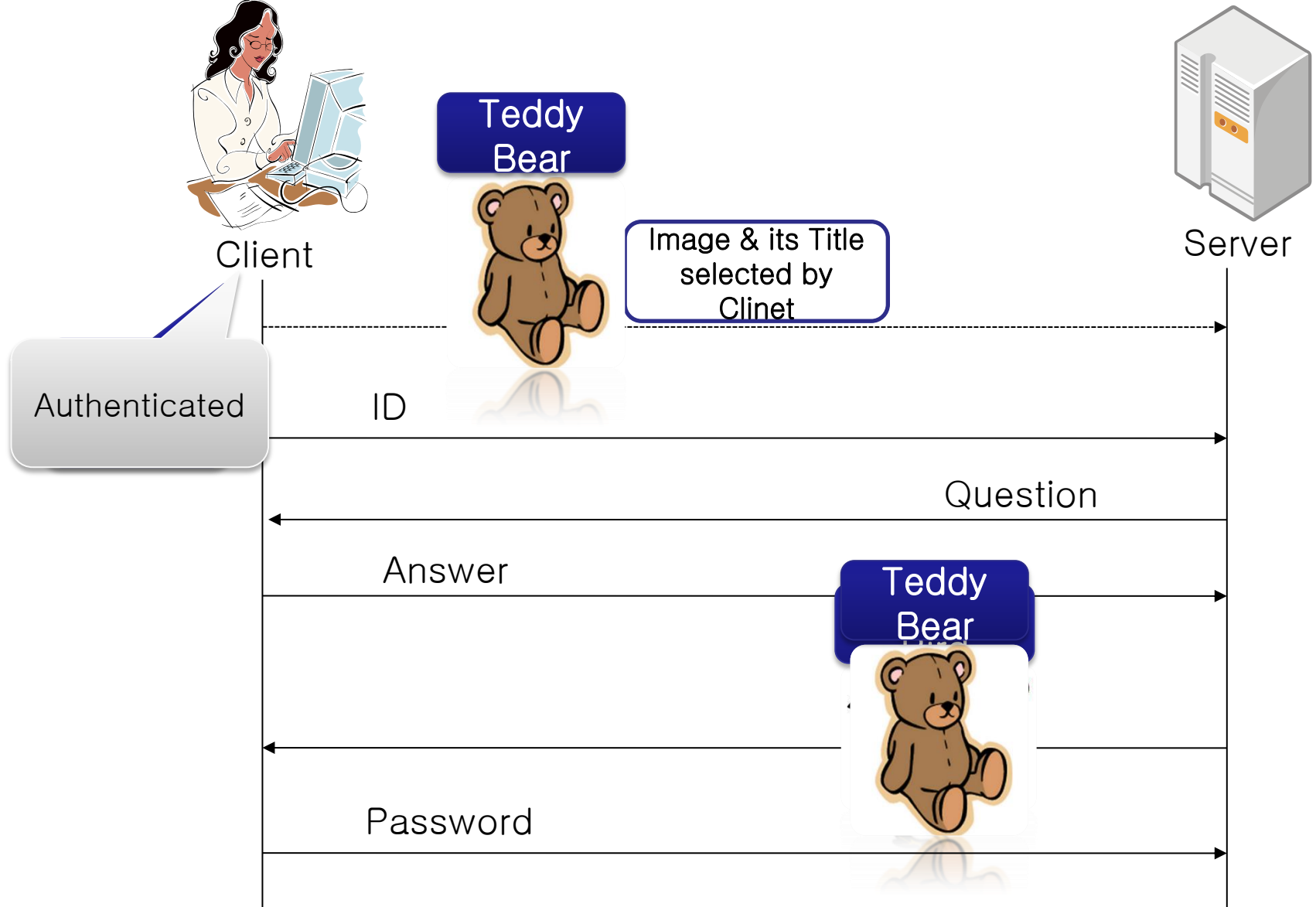
- ▶ 걷는 사람의 실루엣을 정적 혹은 동적으로 획득하여 인식
- ▶ 원거리에서 개인을 인식: 출입통제시스템



✕ DNA

- ▶ DNA 인식은 다른 제공자로부터 획득한 DNA를 포함한 세포 조각들 중에서 핵산의 구성 성분인 뉴클레오티드 비교하는 기술
- ▶ 범죄자 확인, 약물복용확인, 친자확인(부계, 모계 확인)등 다양한 요소에서 활용

10.7 차세대 개체 인증- Cognitive Authentication



10.7 차세대 개체 인증- Cognitive Authentication

Online Banking

Easy. Secure. Free.

Enroll [View demo | Learn more](#)

Enter Online ID:

☐ Save this Online ID

Where do I enter my Passcode?

Sign In

Forgot or need help with your ID?
[Reset Passcode](#)
Sign in for less than 1 minute

Online ID: kthcjstk [Sign in using a different Online ID](#)

In what city were you born? (Enter full name of city only)

Answer:

(Not case sensitive)


[Forgot the answer to your SiteKey Challenge Question?](#)

Do you want us to remember this computer, so you can avoid answering your challenge questions next time you sign in? [Learn more](#)

☒ Yes

☐ No

Your SiteKey:
studyhard



If you don't recognize your personalized SiteKey, don't enter your Passcode.

* Passcode:

(8 - 20 Characters, case sensitive)

Sign In

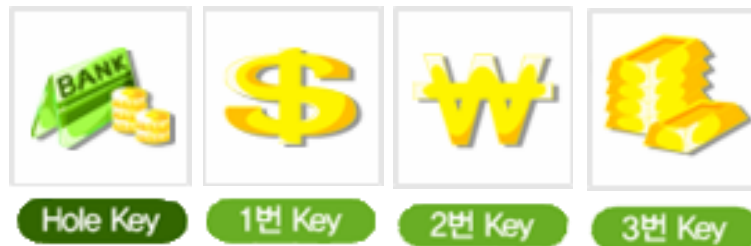
2150 10

User ID

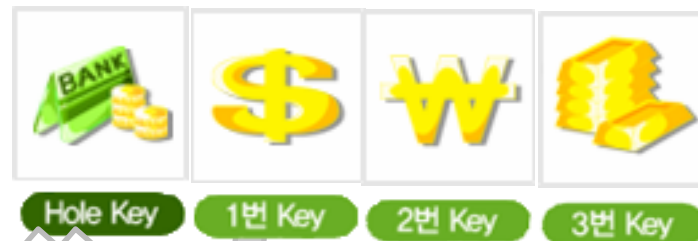
Pre-set Challenge/Response

Comparison PASSWORD Login

10.7 차세대 개체 인증- Cognitive Authentication



Select 4 images



Drag 1,2,3
Keys to
HoleKey

10.7 차세대 개체 인증- Cognitive Authentication

- ✘ Setting: two shared secrets
 - ▶ Secret Question : “Is there a person?”
 - ▶ Secret Sequence : 5 meaningful bits
- ✘ Challenge: 10pics → Answer:

0098030502
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
??NN?N?Y?Y



1



2



3



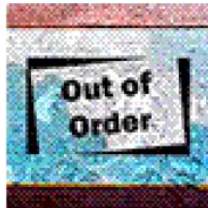
4



5



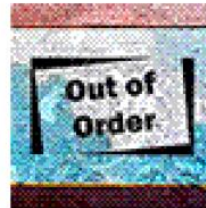
6



7



8



9



?

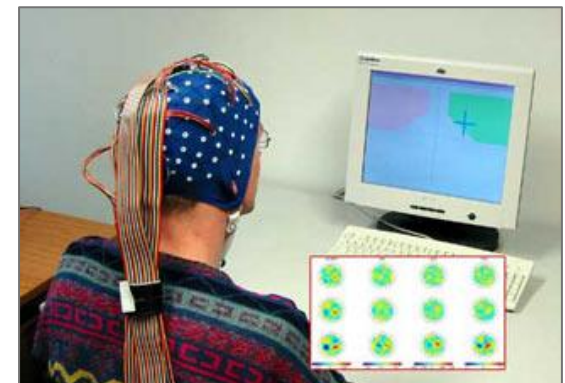
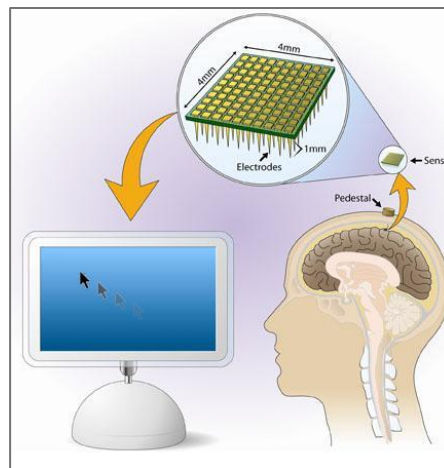
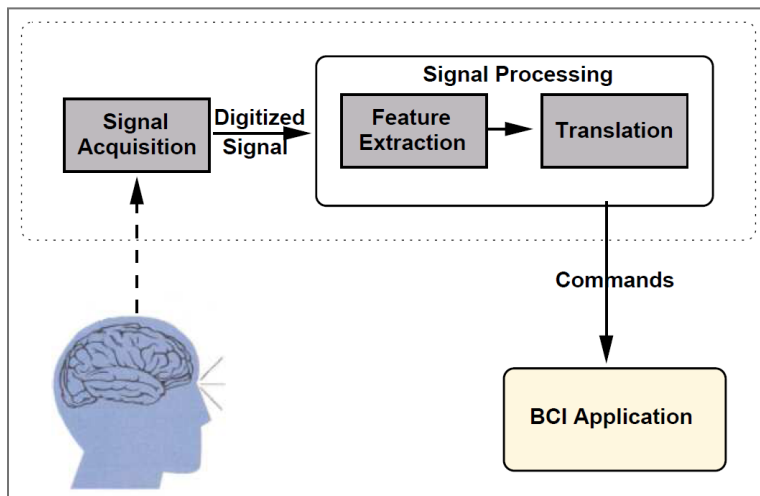
10.7 차세대 개체 인증- Cognitive Authentication

- ✕ Original Answer: ??NN?N?Y?Y
- ✕ User's random at the position of “?”
- ✕ If N, assign “0”.
- ✕ If Y, assign “1”.
- ✕ Response to the previous Example : 1000101101

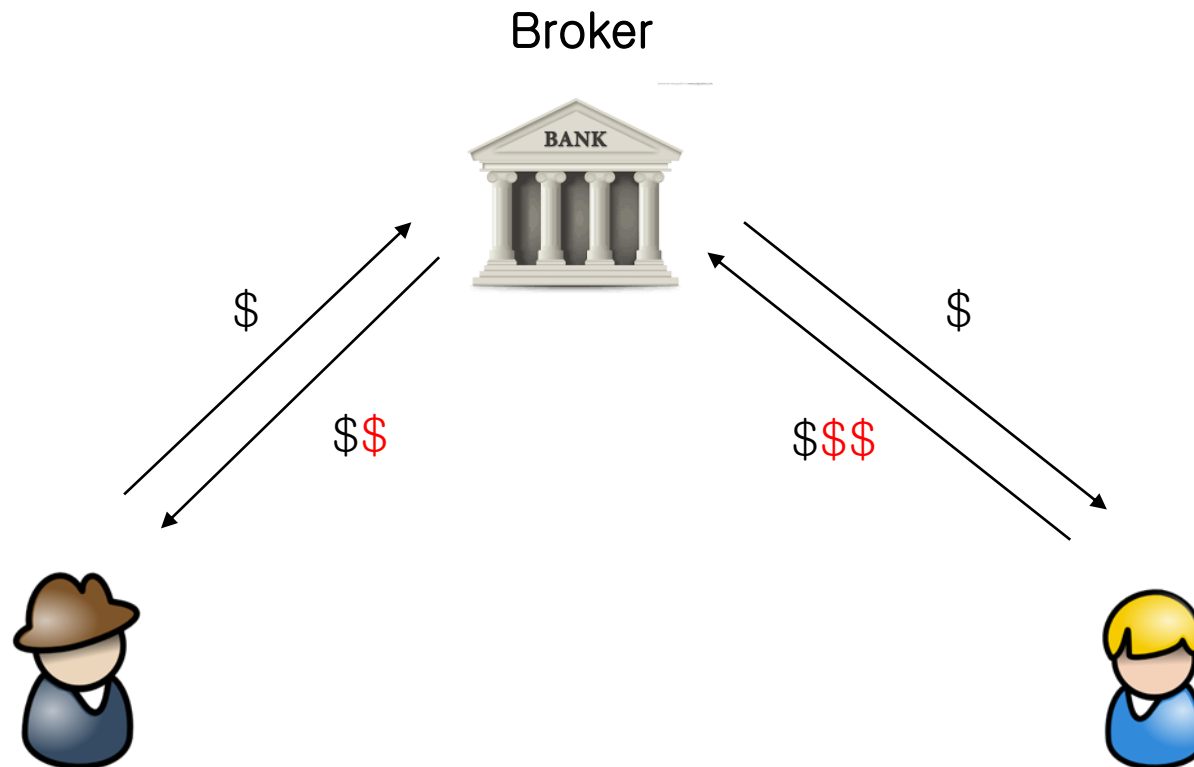
10.7 차세대 개체 인증- Cognitive Authentication

Authentication by *thinking* a password

- ✗ We can avoid the shoulder surfing by simply transmitting some chosen thought.
- ✗ Since every individual brain has a unique characteristic, all users will have different signals even if they are thinking the same word.



10.8 결제방식과 FinTech (Finance + Tech.)



그런데...

I am the only player

I can do

I can do

I can do what you are doing



I can do

I can do



- ✗ What if other parties could provide financial services, especially evaluate credit?
- ✗ Why?
 - ▶ Save Money in Financial Services without using Bank
 - ▶ Remind that Paypal and Alipay began for the purpose of **non-credit card** based payment

성공사례

✂ TransferWise



(사진=트랜스퍼와이즈)

“자신들은 전혀 모르는 사람에게 500달러를 보낸셈이지만,
자신들의 목적인 500달러는 어찌 됐든 전달된거죠”

“그리고 이 과정에서 비싼 해외 송금료가 아닌
국내 송금료만 내면 되구요. ㅎㅎ”

“예전에는 이렇게 딱 매칭이 되는 사례를
도저히 찾을 수 없었겠지만, 인터넷과 SNS로 열린
초연결 시대에 얼마든지 이런 중계모델이 가능한 거죠”

성공사례

✂ LeadingClub



▲미국 렌딩클럽(사진=렌딩클럽)

“은행창구에 앉은 ‘전문가’가 아니라도,
이런 정보를 안전하게 다룰 수 있는 ‘기술’을 가진
회사가 충분히 개인의 ‘대출’ 여력을 알 수 있게
되면서”

“굳이 허가된 은행이나 금융기관에
비싼 이자를 내며 돈을 빌릴 게 아니라
나의 신용정보를 공개하고, 이를 보고 돈을 빌려줄
사람을 찾을 수 있는 시대가 됐습니다”

성공사례

- BitCoin vs AmazonCoin
- ApplePay vs Current C



- What if a person can evaluate credit risk?
 - ✗ KreditTech : using SNS info. and tracking smartphone

대한민국은?

■ FinTech 주무부처 : 금감위?

Powered by CLOUD
스마일서브에서 안정적으로 서비스를 받고 있습니다.

B 블로터
BLOTER.NET



뉴스

아카데미

컨퍼런스

북스

광장

Hjundal Capital 현대캐피탈 다이렉트론

지갑 속 카드처럼 바로 쓰는 신용대출 직장 정보 입력 없이 10분 내 입금

2015.02.05

핀테크 발목 잡는 5대 족쇄

f 70 21



안상욱

박근혜 대통령이 핀테크 육성에 힘쓰라고 말한 뒤 정부 기관이 박차를 가하는 모습이다. 금융위원회(금융위)는 지난 1월27일 IT·금융 융합 지원방안을 내놓았다. 그동안 핀테크 업계에서 문제라고 지적한 점을 거의 모두 손보겠다고 발표했다. 적용 시기까지 6개월에서 1년 뒤로 늦췄다.

핀테크 업계는 금융위 발표를 '중합선물세트'라 부르며 반겼다. 그동안 핀테크 산업에 족줄을 죄던 규제기관이 앞장 서 전방위적으로 핀테크 산업을 육성하겠다는 의지를 밝혔기 때문이다.

하지만 아쉬움도 남았다. 61쪽짜리 보고서에는 큰 틀에서 방향만 제시돼 있을 뿐이다. 구체적으로 어떤 규제를 어떻게 손볼 계획인지는 알 수 없다. 금융위를 일선에 내세운 정부가 진짜로 핀테크 산업을 육성하려면 무엇보다 손봐야 할까. 한국핀테크포럼의 도움을 받아 핀테크 업계가 걸림돌이라고 생각하는 법이나 규제가 무엇인지 들어봤다. IT전문 법무법인 테크앤로에서 법률 자문을 받았다.

〈표 1〉 완화할 필요가 있는 규제들

규제	관련 법률규정
금융기관의 공인인증서 사용 의무	내부 규정
금융실명제법상 대면 확인 의무	금융실명제법, 내부 규정
핀테크 기업들의 금융정보 공유 제한	개인정보보호법
금융기관들의 핀테크 자회사/ 합작회사 설립 제한	금융지주회사법

(자료-LG경제연구원)

간편결제 시스템

■ 분야별 전자지급결제 서비스 동향

구분	정의	관련사	추진동향
Pg사	온라인 결제수단제공, 결제중계 및 정산을 주 업무로 하는 전문지급 결제대행사	PayPal, AliPay	- 글로벌 서비스화 - 결제 신기술 개발
		이니시스, LGU+, 한국사 이버결제	- 간편결제, 원클릭 결제
카드사	카드 결제처리 관련 권한, 인프라 제어권 등 전통적 결제주도권을 보유하고 있는 회사	VISA, Master Card, 신한, 국민, BC 등	- 지급결제서비스 직접제공 - 결제종단간 토탈 솔루션 제공
통신사	이동통신 인프라와 단말에 대한 영향력을 기반으로 모바일 단말 결제(NFC) 혹은 모바일 인프라 기반 결제 수행	ISIS/Vodafone	- NFC 단말 모바일 결제 - 개도국 진출
		SK, KT, LGU+	- 단말, 이통사 인프라 기반 결제 인증 서비스 개발

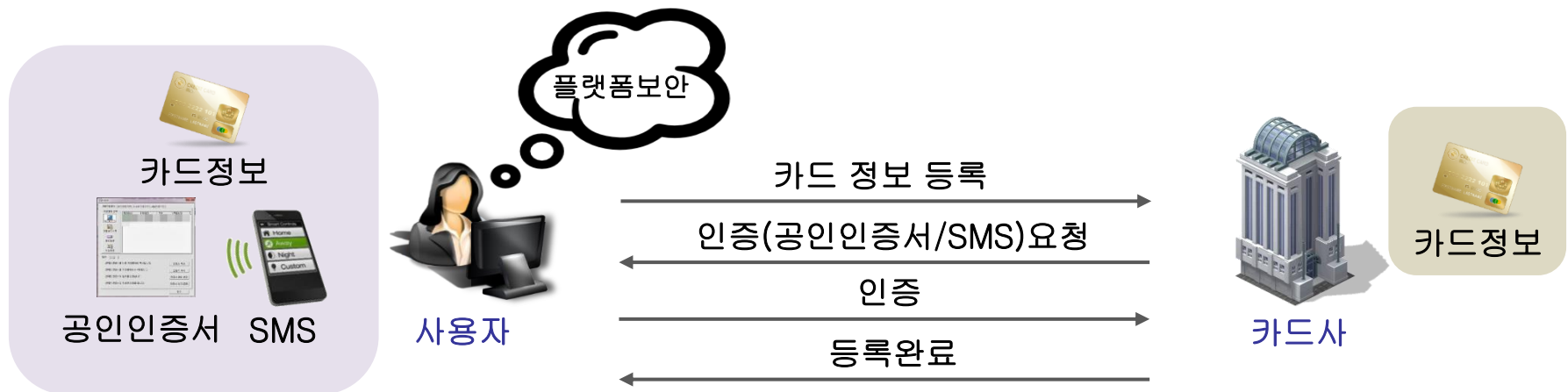
간편결제 시스템

■ 신용카드 지급결제와 인증

	지급결제확인	도용 문제	대응 노력
오프라인	물리적 카드 소유	<ul style="list-style-type: none">• 카드 복제/위조• 카드 절도	<ul style="list-style-type: none">• IC카드(복제방지)• 서명 확인• PIN• FDS
온라인	카드번호 제출	<ul style="list-style-type: none">• 스니핑• 피싱/파밍• 가맹점 도용• 카드정보 유출 (POS, PG, 카드사)	<ul style="list-style-type: none">• 암호통신(SSL)• CVC, 유효기간• 안심결제(1회용 정보 + PW)• 추가인증(고액결제 시 공인인증서)• FDS

국내 결제 서비스

■ 안심클릭 ✕ 등록과정

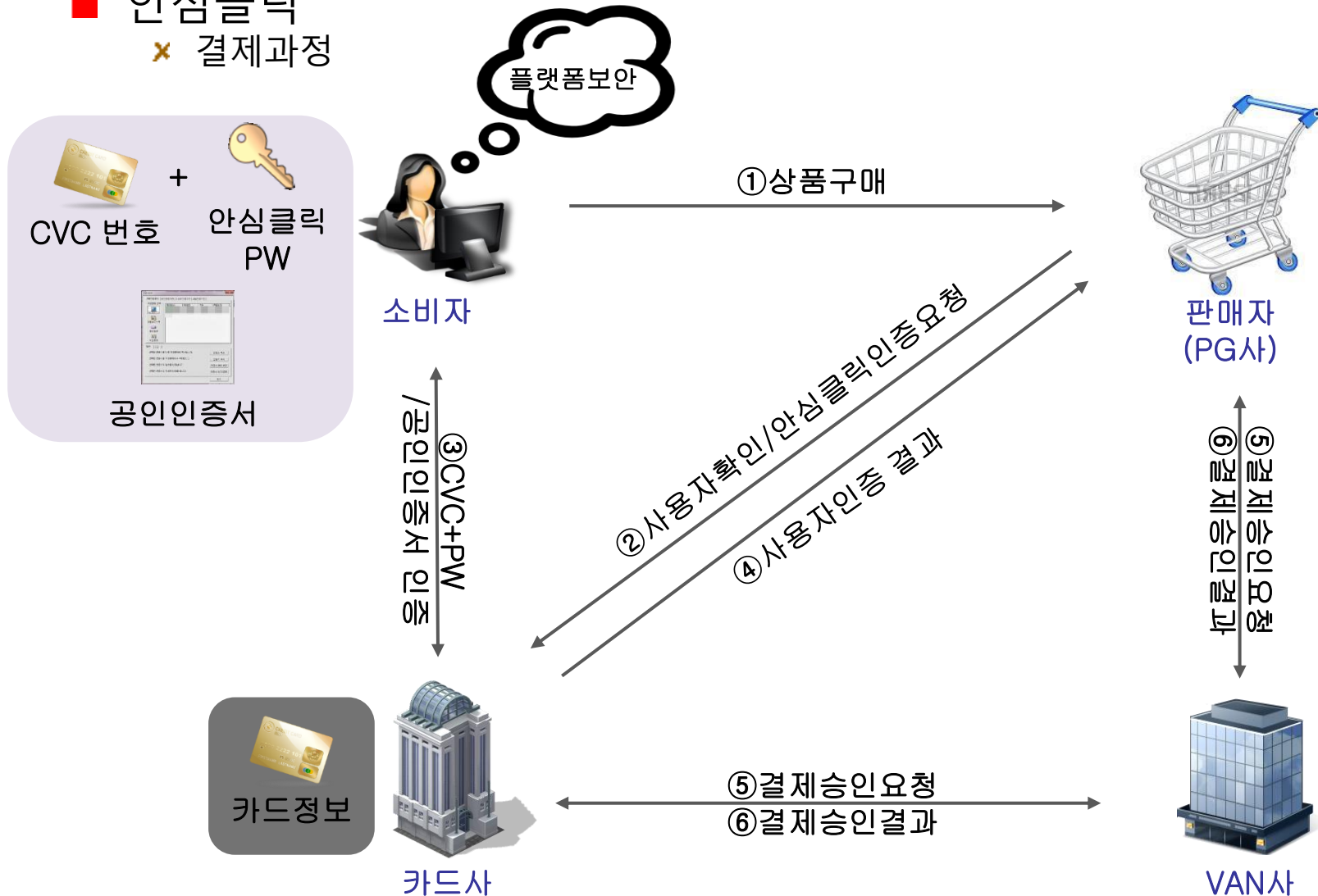


✕ 인증방법(등록과정)

- ▶ 카드번호, 카드 비밀번호, CVC, 유효기간, 휴대폰 SMS 인증번호
- ▶ 카드번호, 카드 비밀번호, CVC, 유효기간, 공인인증서

국내 결제 서비스

■ 안심클릭 ✕ 결제과정



국내 결제 서비스

■ 안심클릭

✕ 인증방법(결제과정)

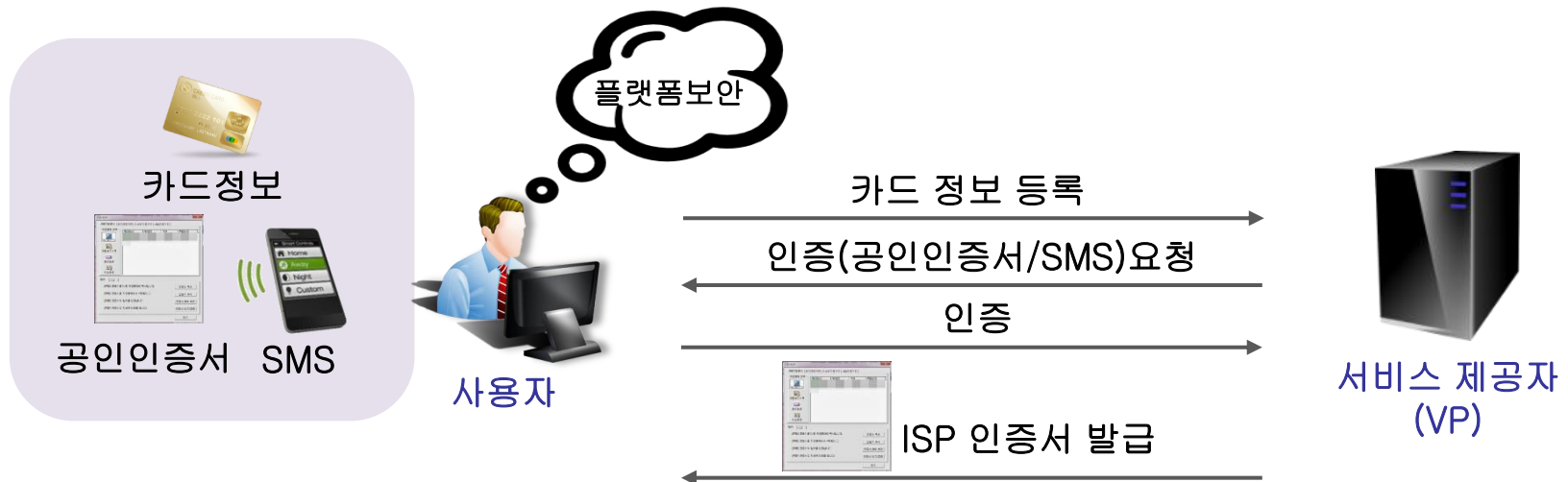
- ▶ 안심클릭 PW + CVC (30만원 미만 결제 시)
- ▶ 공인인증서 (30만원 이상 결제 시)
- ▶ 단, 온라인 게임 등에서는 기준이 10만원으로 변경

✕ 서비스 특징

- ▶ VISA의 3D-Secure(VISA 안심클릭) 모델을 한국형으로 개발하여 적용
- ▶ 안심클릭 서비스 사용 시, 이용자가 등록한 개인확인 메시지 확인가능
- ▶ 서비스 등록 후, 모든 PC에서 사용 가능
- ▶ 사용자 카드의 정보는 사용자와 카드사만 저장하고 있음

국내 결제 서비스

■ ISP 결제 ✕ 등록과정

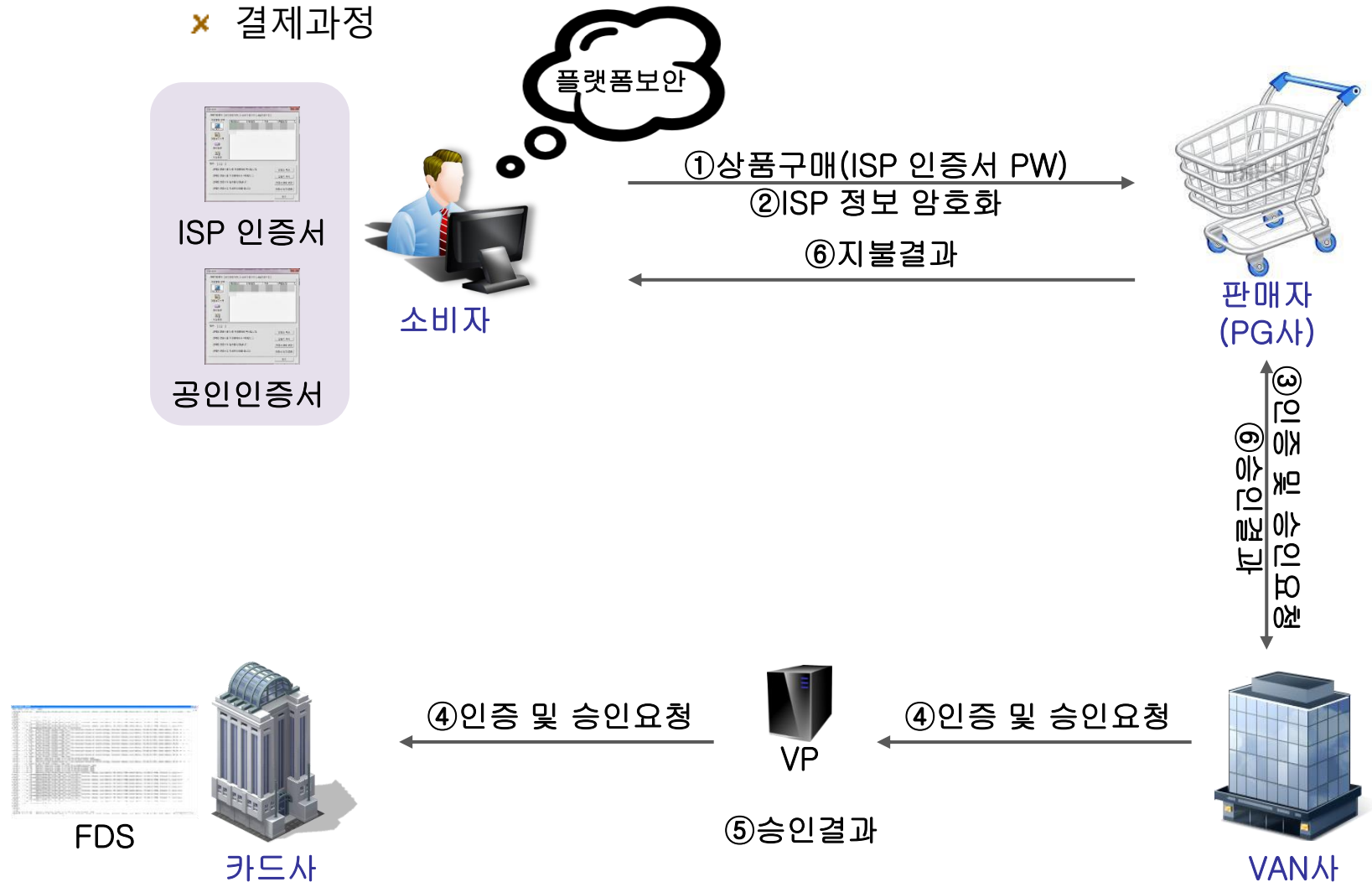


✕ 인증방법(등록과정)

- ▶ 카드번호, 카드 비밀번호, CVC, 유효기간, 휴대폰 SMS 인증번호
- ▶ 카드번호, 카드 비밀번호, CVC, 유효기간, 공인인증서

국내 결제 서비스

■ ISP 결제 ✕ 결제과정



국내 결제 서비스

■ ISP 결제

✕ 인증방법(결제과정)

- ▶ ISP 인증서 PW 입력 (30만원 미만 결제 시)
- ▶ 추가적으로 공인인증서 인증 필요 (30만원 이상 결제 시)
- ▶ 단, 온라인 게임 등에서는 기준이 10만원으로 변경

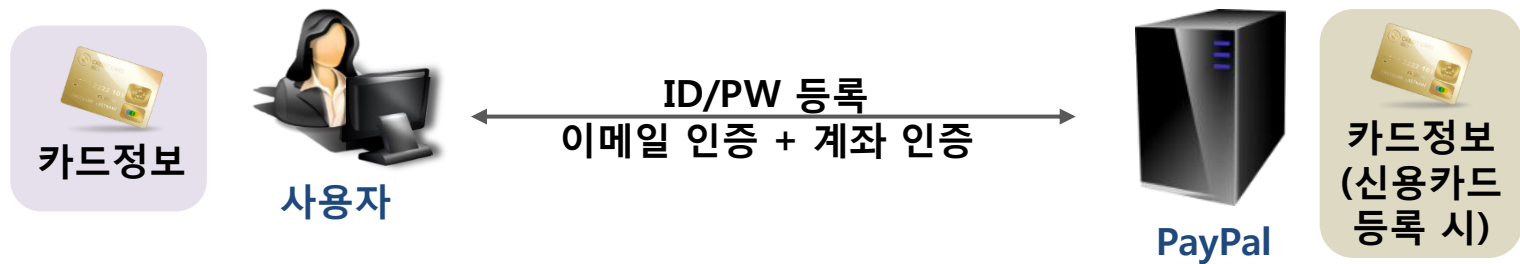
✕ 서비스 특징

- ▶ 인증서 기반 결제 방법으로, 카드정보 입력 없이 ISP 인증서 PW 입력으로 결제 가능
- ▶ 인증서가 저장된 PC에서만 사용 가능
- ▶ 카드사의 FDS 모니터링

해외 결제 서비스

■ 페이팔(Paypal)

✕ 등록과정



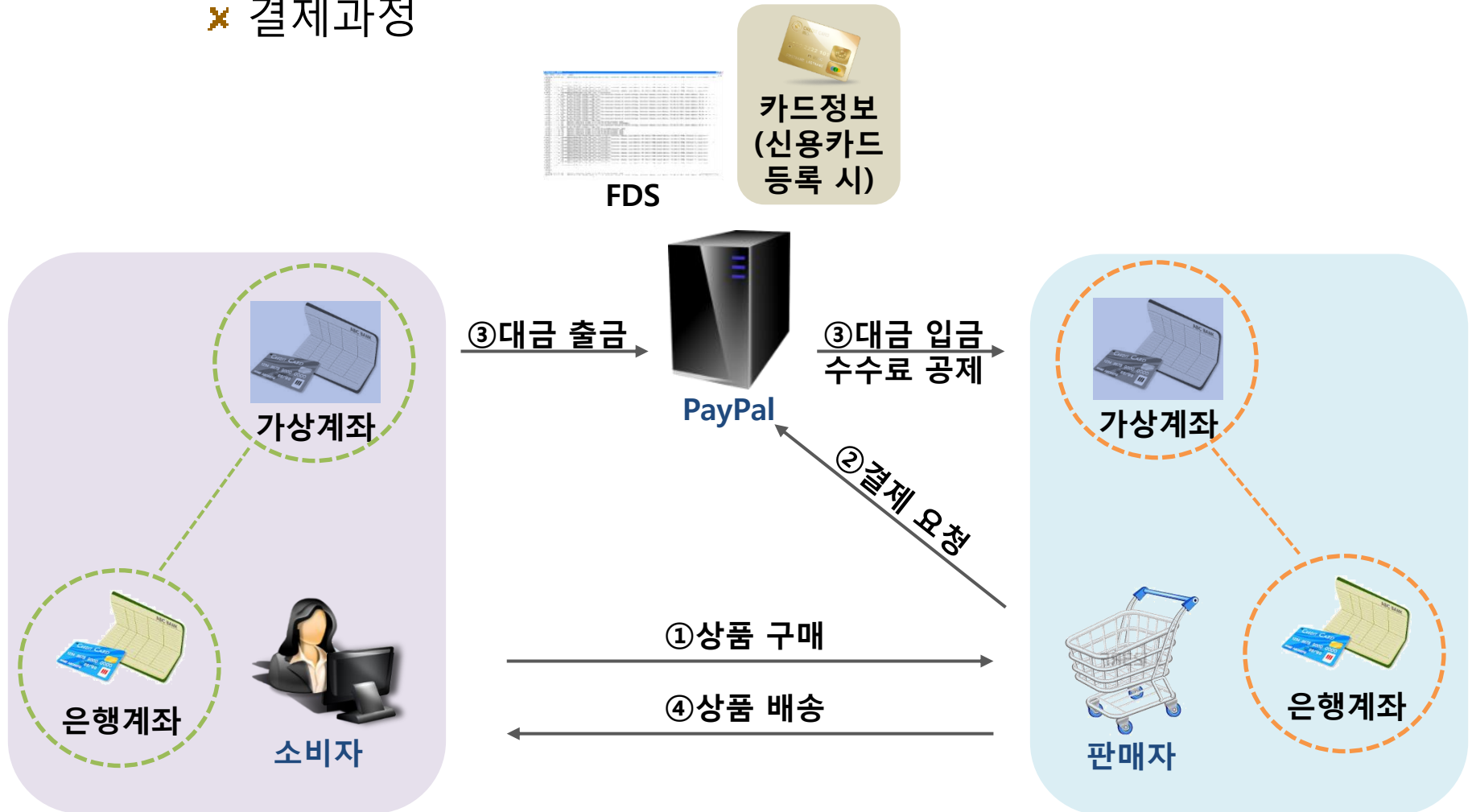
✕ 인증방법(등록과정)

- ▶ 이메일 인증(메일 발송 링크 클릭) + 계좌 인증(은행계좌, 신용카드에 대해 입금 및 결제 처리 테스트)
- ▶ 계좌연동 : PayPal 가입을 통해 가상계좌 생성 후 실 계좌 연동

해외 결제 서비스

■ 페이팔(Paypal)

✕ 결제과정



해외 결제 서비스

■ 페이팔(Paypal)

✕ 인증방법(결제과정)

- ▶ ID/PW 로그인 (SSL 암호화 통신)
- ▶ 추가적으로 SMS인증 또는 OTP카드 사용 가능

✕ 서비스 특징

- ▶ ID/PW 만으로 결제 가능 (추가적인 S/W 설치 없음)
- ▶ 가상계좌 간 거래, 네트워크 상 금융정보(신용카드 정보 등) 미 전송
- ▶ PayPal사가 사용자의 카드정보를 저장 및 관리(신용카드 등록 시)

✕ 보안정책

- ▶ 보안 수준 : PCI-DSS(美 신용카드 보안 규격) 획득
- ▶ 웹 표준 (SSL) 사용
- ▶ FDS 24시간 모니터링
- ▶ 분쟁 조정 : 상품 미 배송, 불일치, 부정결제, 지불거절 등 거래분쟁 직접 조정
- ▶ 버그 바운티 제도 : 보안 취약성 발견자에게 상금 지급

해외 결제 서비스

■ 애플 페이(Apple pay)

✘ 등록과정

- ▶ 아이튠즈 : 아이튠즈에 등록된 신용카드 이용 가능
- ▶ Passbook : 신용카드 사진촬영 / 번호입력



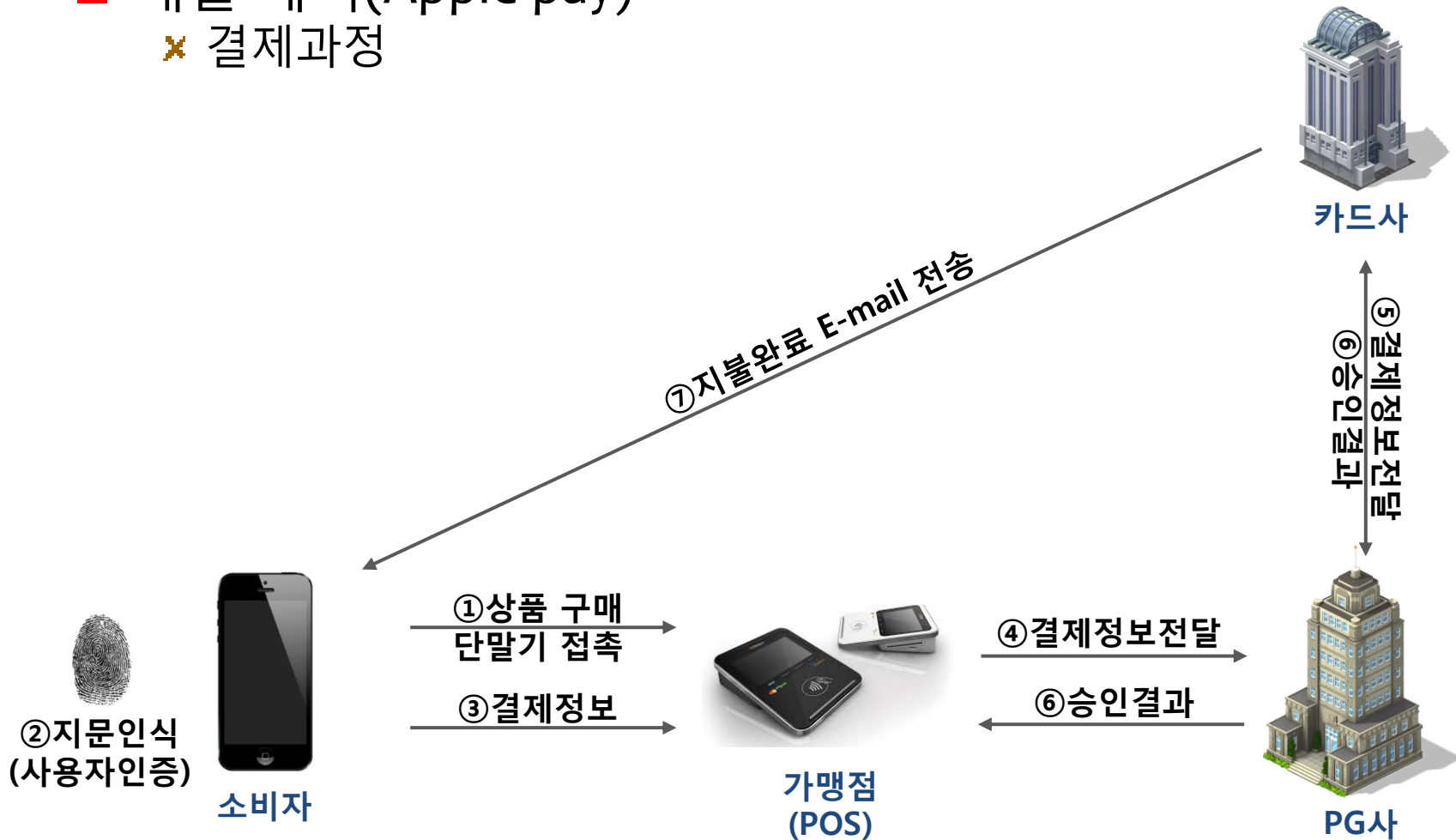
✘ 애플 페이 결제 저장 방식 및 활용 기술

- ▶ 저장방식 : 신용카드 및 사용자의 결제 정보는 암호화를 통해 보안칩(Secure Enclave)에 저장
- ▶ 지문인식 : 사전 등록된 신용카드 정보를 이용해 홈 버튼에 지문을 인식하여 결제하는 방식
- ▶ NFC : NFC 기술을 이용하여 지급결제 서비스 제공
- ▶ 토큰화(Tokenization) : 결제 간 데이터 유출 위험 최소화

해외 결제 서비스

■ 애플 페이(Apple pay)

✕ 결제과정



해외 결제 서비스

■ 애플 페이(Apple pay)

✕ 인증방법(결제과정)

- ▶ Touch ID(지문인증)을 이용한 사용자 인증

✕ 서비스 특징

- ▶ 스마트폰을 이용한 NFC 기반 전자지급결제 서비스
- ▶ Passbook 앱을 이용하여 신용카드 정보 저장
- ▶ 지문인증을 통해 사용자 인증
- ▶ 지문인식, 보안영역활용(SE), 토큰화를 통한 보안성 강화

✕ 보안정책

- ▶ 지문인식 : 반도체식 센서로 홈 버튼에 장착되어 Touch ID로 추출된 특징점을 보안영역(SE)에 저장
- ▶ 위조지문 문제 발생 가능성
- ▶ 보안영역 : 지문정보는 보안영역(Secure Data Repository)에 저장되며, 비교(Matching)를 위해 보안 프로세서(Secure Enclave Processor)와 별도의 채널 사용
- ▶ 토큰화 : 결제정보 암호화에 사용되는 기술, 사용자와 카드사만이 카드정보를 가짐

그 밖의 인증기술 FIDO(1/2)

■ FIDO(Fast Identity Online)

✕ ID/PW 입력 방식보다 더 높은 보안성을 제공하며, 활용도도 높은 인증 서비스

✕ UAF & U2F

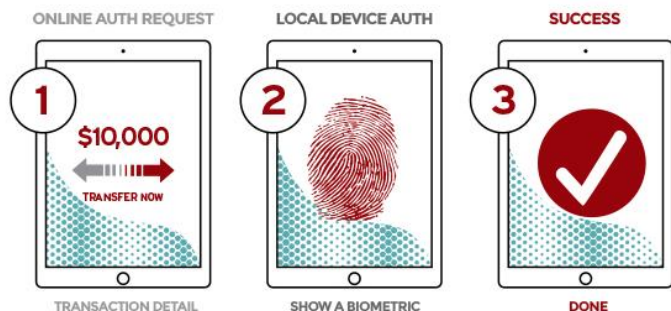
▶ UAF(Universal Authentication Framework) Protocol

: 디바이스에서 제공하는 인증방법을 온라인 서비스와 연동, 사용자를 인증하는 프로토콜

▶ U2F(Universal 2nd Factor) Protocol

: 기존 ID/PW를 사용하는 온라인 서비스에서 두 번째 인증요소를 추가하는 프로토콜

PASSWORDLESS EXPERIENCE (UAF standards)



SECOND FACTOR EXPERIENCE (U2F standards)

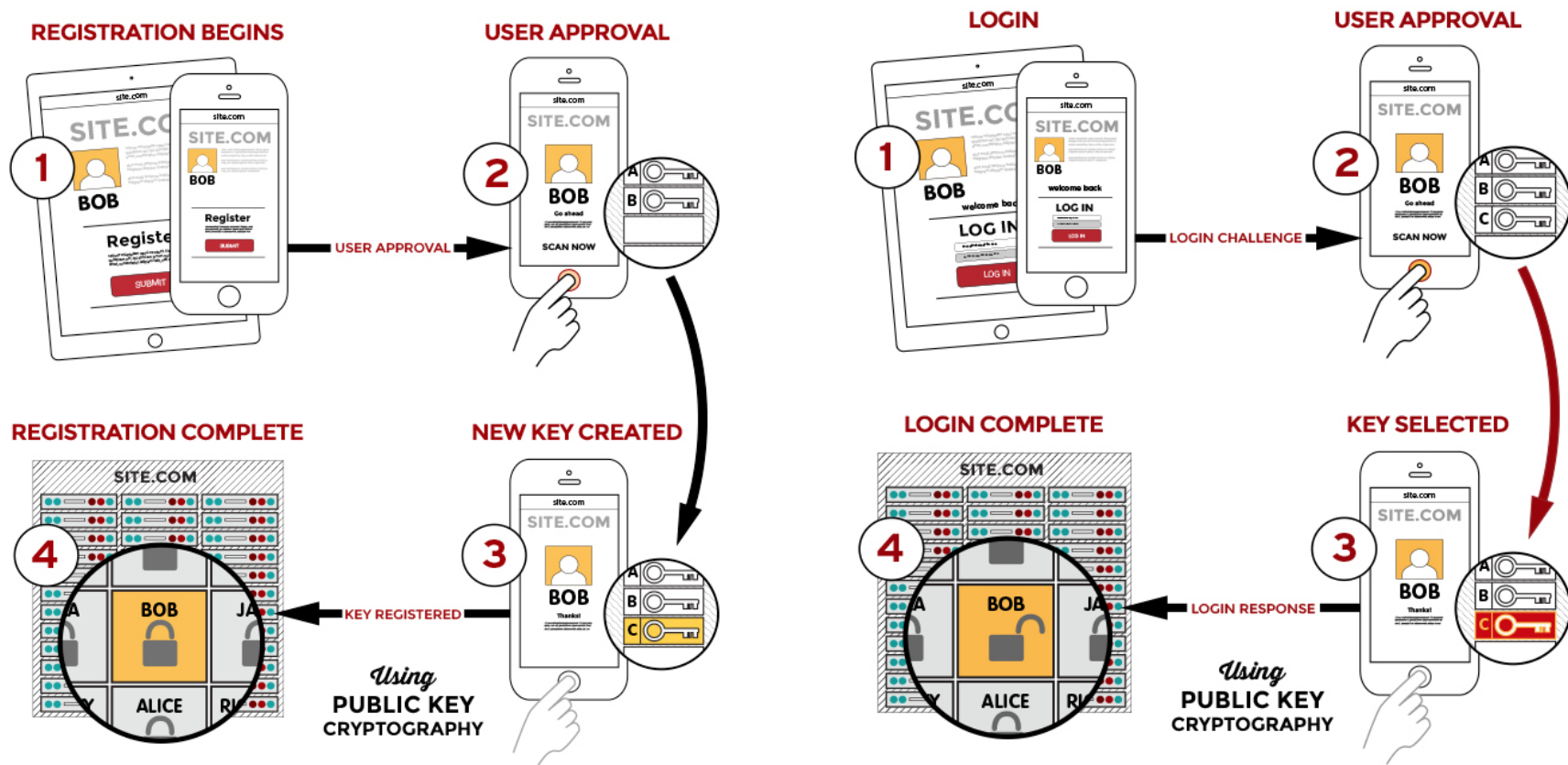


그 밖의 인증기술 FIDO(2/2)

■ UAF(Universal Authentication Framework): 비(非) 비밀번호 인증

✕ 등록 과정

✕ 인증 과정



비대면 인증

■ 해외 주요 비대면 인증 방식

✕ 프랑스 – BNP Paribas ‘Hello Bank!’

- ▶ 고객정보 확인 후 계좌개설에 필요한 임시 비밀번호를 체크카드와 함께 등기우편으로 송부

✕ 일본 – Sony Bank

- ▶ 우체국 직원이 수신인 신분증으로 실명확인

✕ 미국 – Ally Financial

- ▶ 신청고객에게 서명카드 송부, 고객이 카드에 서명하여 보내면 확인함
- ▶ 다른 은행에서 사용중인 계좌를 통해 실명/이체계좌의 보유여부 확인 가능

■ 국내 도입 시 유의사항

✕ 비대면 인증의 복잡도

- ▶ 비대면 인증 절차가 너무 복잡한 경우, 이용자 부족으로 인해 제도개선 효과 반감됨

✕ 추가적인 인증 제도 필요

- ▶ 이동통신사 DB를 활용한 본인명의 휴대폰 SMS 인증
- ▶ 화상통화, 홍채·지문인식 등을 통한 인증
- ▶ 다른 은행의 계좌를 이용한 실명확인 등