
1. 서약서 2. 포스터 양식

보안프로그래밍

개체인증 (Entity Authentication)

필요한 양식이 더 있을 경우, 추후 업데이트 하겠습니다.

숭실대학교 소프트웨어학부 조효진

Contents

- 개체인증 개요
- 패스워드 기반 인증
- OTP 기반 인증
- 질의 응답 기반 인증
- 차세대 인증

개체인증 (Entity Authentication) 개요

□ 개체인증 (Entity Authentication or Identification)

- 특정 개체를 다른 개체에 인증하는 것
- 개체는 사람, 프로세스, 디바이스 등이 될 수 있음
 - Access to computer system
 - Entry to building
 - Access to ATM
 - Server login

개체인증 (Entity Authentication) 개요

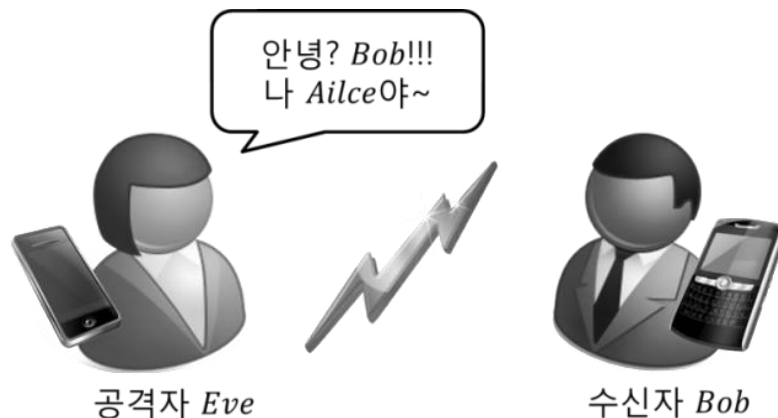
□ Data-Origin (Message) vs. Entity Authentication

- Entity Authentication은 일반적으로 실시간으로 이루어져야 함. 이에 반해, Message authentication은 실시간으로 이뤄지지 않을 수 도 있음
 - E.g., 출입구 관리 시스템 vs. 문서 열람 시스템
- Entity Authentication은 한번 수행 된 후, 일정 시간동안 유지됨. 이에 반해, Message authentication은 각 메시지마다 수행되어야 함
 - E.g., 은행 사이트 로그인 후, 10분 동안 로그인 유지 vs. 이메일 인증

개체인증 (Entity Authentication) 개요

□ 개체인증

- 개체의 신원을 증명하기 위한 일련의 과정
 - 개체: 사람, 기기, 프로세스 등
- 인증 정보
 - What you are (voice, fingerprint, Iris)
 - What you know (password)
 - What you have (smart card, token card)
 - 최근에는 Two-factor 인증 or Multi-factor 인증이 쓰이고 있음



패스워드 기반 인증

□ 미 AOL (America Online) 취약 비밀번호 25가지

- “25 [Most Used Passwords](#)”

password	12345	letmein	michael	2000
123456	dragon	monkey	shadow	jordan
12345678	pussy	696969	master	superman
1234	baseball	abc123	jennifer	harley
qwerty	football	mustang	111111	1234567

해킹 취약 비밀번호는?

패스워드 기반 인증

□ 패스워드: low entropy

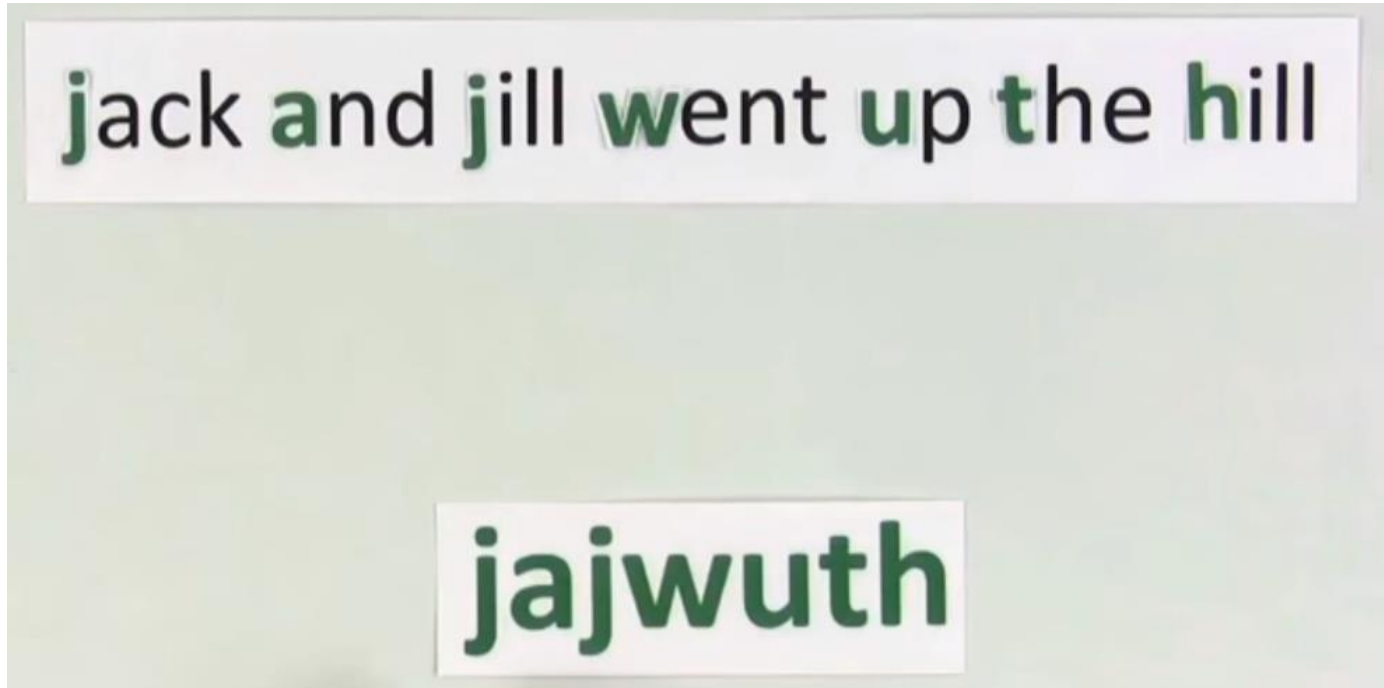
- 64-비트 패스워드를 발견하기 위해서는 최대 2^{64} 번의 공격시도 필요함 (Brute force 공격)

□ 안전한 패스워드

- 국내: 한국인터넷진흥원 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 함
- 해외: NIST 800-63 : Electronic Authentication Guideline (2006)

패스워드 기반 인증

□ 안전한 패스워드 만들기



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

패스워드 기반 인증

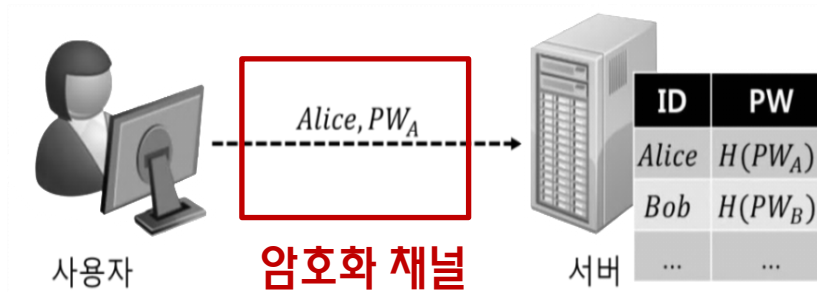
□ 고정된 패스워드

- 패스워드 테이블의 유출 시 위험



패스워드 기반 인증

□ 해쉬된 패스워드



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

Appendix#1 Dictionary attack

DICTIONARY ATTACK!



<https://null-byte.wonderhowto.com/how-to/crack-wpa-wpa2-with-wifite-0161976/>

Appendix#1 Dictionary attack



<https://www.adviservice.com.au/2018/11/the-art-of-the-password/>

패스워드 기반 인증

□ 해쉬된 패스워드 Pre-computation: 2^n hash computations

- 공격자는 해쉬 계산을 미리 계산하고 저장함 (해쉬함수의 output size가 n 비트일 경우)
 - Chose 0 $\rightarrow H(0) = 0xFFDD\cdots AAFF$
 - Chose 1 $\rightarrow H(1) = 0x112F\cdots AA23$
 - ...
 - Chose $k \rightarrow H(k) = 0xABDE\cdots EAFD$
 - ...
 - Chose $2^n - 1 \rightarrow H(2^n - 1) = 0xDDAF\cdots EA44$
- 유출된 패스워드 파일에서 계정 A의 해쉬된 패스워드가 “0xABDE \cdots EAFD” 이라면, Pre-computation table에서 “0xABDE \cdots EAFD” 을 찾음

패스워드 기반 인증

- 또한, 공격자는 자주 사용되는 패스워드 사전 (Dictionary)에 대한 pre-computation table을 활용하여 좀 더 쉽고 빠르게 패스워드를 유추할 수 있음

패스워드	해시 값
Love	551234523452 $\leftarrow \text{hash}(\text{Love})$
Soongsil	123452323242 $\leftarrow \text{hash}(\text{Soongsil})$
apple	523233452333 $\leftarrow \text{hash}(\text{apple})$
...	...

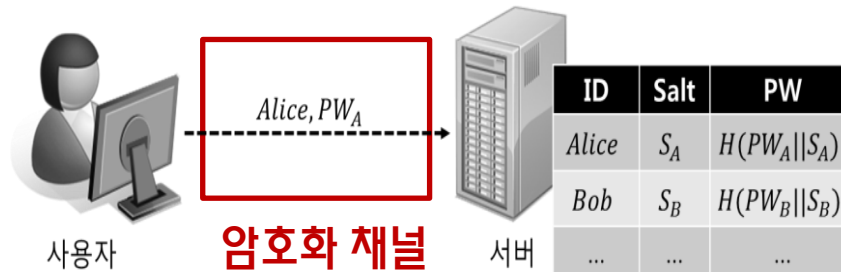
[Pre-computation table]

계정	해시 값
Alice	551234523452
Bob	74823012930
Tomas	532423452343
...	...

[유출된 Password DB]

패스워드 기반 인증

□ Hash+솔트(Salt) 사용



- 임의 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
 - 솔트가 공개된 경우, **특정인 Alice의 PW를 알기 위한 계산은 변동없음**
 - 여전히 $O(2^n)$ 번의 해쉬 평가 (n :해쉬 함수의 출력 길이)
 - 하지만 **pre-computation table**을 사용하기가 **어려움**
 - **Offline dictionary attack**도 적용이 **어려움**

패스워드 기반 인증

□ Hash+솔트(Salt) 사용 Pre-computation: $2^{n+|salt|}$ hash computations

- 공격자는 해쉬 계산을 미리 계산할때 salt의 경우의 수도 고려해서 모든 경우의 수에 대한 해쉬값을 저장함 (해쉬함수의 output size가 n 비트일 경우)
 - If we assume that 2-bit salt is used for explanation
 - Chose 0 and salt 0 $\rightarrow H(0||0) = 0xFFDE\cdots EAF F$
 - Chose 0 and salt 1 $\rightarrow H(0||1) = 0xAADE\cdots FAFA$
 - Chose 0 and salt 2 $\rightarrow H(0||2) = 0xADDD\cdots BBAF$
 - Chose 0 and salt 3 $\rightarrow H(0||3) = 0xFFFF\cdots EAF F$
 - Chose 1 and salt 0 $\rightarrow H(1||0) = 0xABBB\cdots FAAB$
 - Chose 1 and salt 1 $\rightarrow H(1||1) = 0x1234\cdots 3456$
 - Chose 1 and salt 2 $\rightarrow H(1||2) = 0x2222\cdots FFFF$
 - Chose 1 and salt 3 $\rightarrow H(1||3) = 0xABDE\cdots EAFD$
 - ...
- 일반적으로 salt의 크기는 128비트 이상의 값이 사용되므로, 공격자는 미리 계산된 pre-computation파일을 만들기 어려움

패스워드 기반 인증

□ Hash+솔트(Salt) 사용에 대한 Dictionary attack

패스워드	해시 값
Love	551234523452 ← $hash(Love)$
Soongsil	123452323242 ← $hash(Soongsil)$
apple	523233452333 ← $hash(apple)$
...	...

[Pre-computation table]

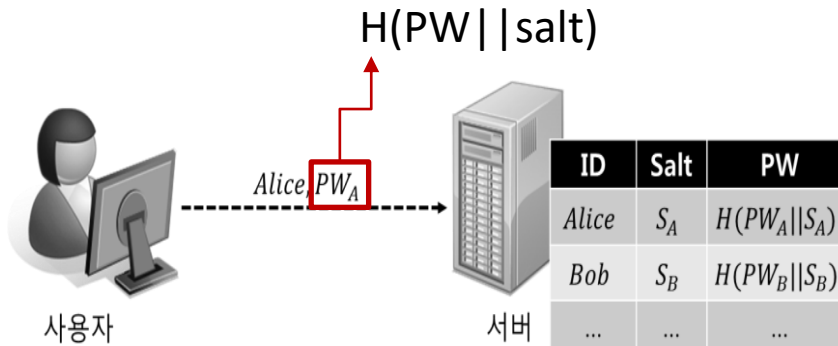
계정	Salt	해시 값
Alice	abcd	551234523452
Bob	1234	74823012930
Tomas	aabds	532423452343
...		...

[유출된 Password DB]

Hash(**alice 패스워드** || abcd) = 551234523452 이므로, Alice의 패스워드는 *Love*가 아님

패스워드 기반 인증

❑ 서버에게 패스워드 노출 문제



■ 안전한 프로토콜 설계 필요

- 서버 → 사용자: Salt 제공
- 사용자 → 서버: PW 대신, $H(PW || salt)$ 전송
- 서버와 사용자 통신에 기밀성과 인증이 제공되어야 함 (e.g., SSL/TLS)

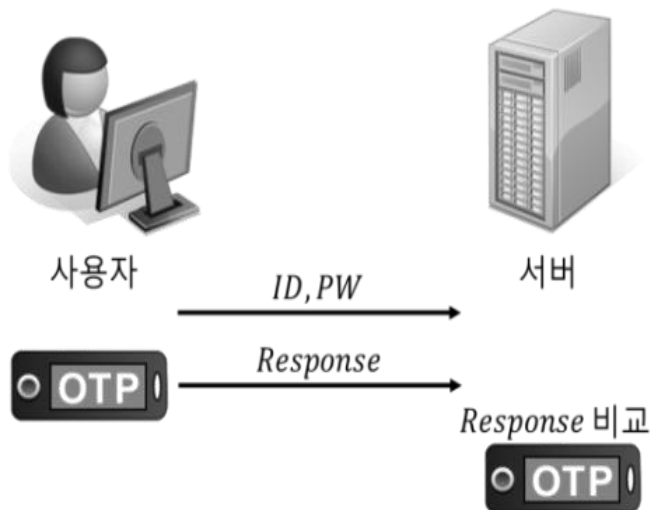
OTP 기반 인증

□ OTP(One-Time Password)

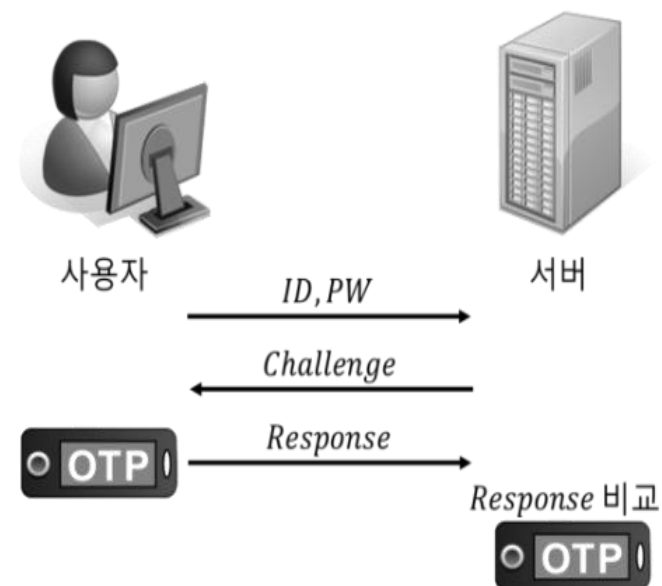
- 매번 다른 난수 사용
- 사전 공격(Dictionary Attack)이나 재전송 공격(Replay Attack) 등으로 부터 안전
- [Two factor authentication](#)

□ OTP에는 동기화 방식과 비동기화 방식이 있음

동기화 방식



비동기화 방식



OTP 기반 인증

□ 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

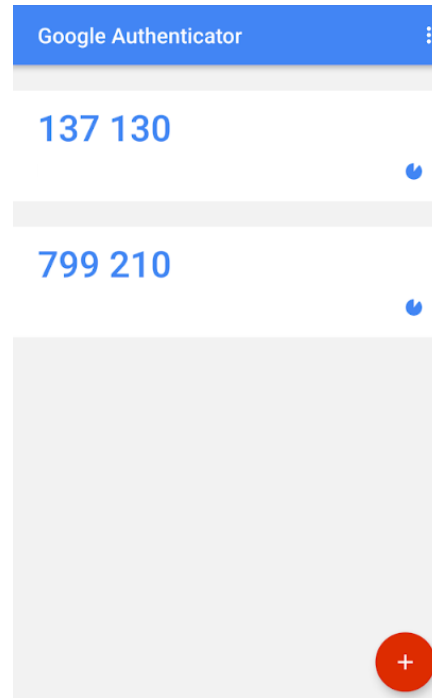


[동기화 방식의 예]

OTP 기반 인증

□ 비동기화 방식의 일회용 패스워드(Non-Synchronized OTP)

- 질의-응답(Challenge-Response) 방식
- 동기화 불필요, 통신량 증가

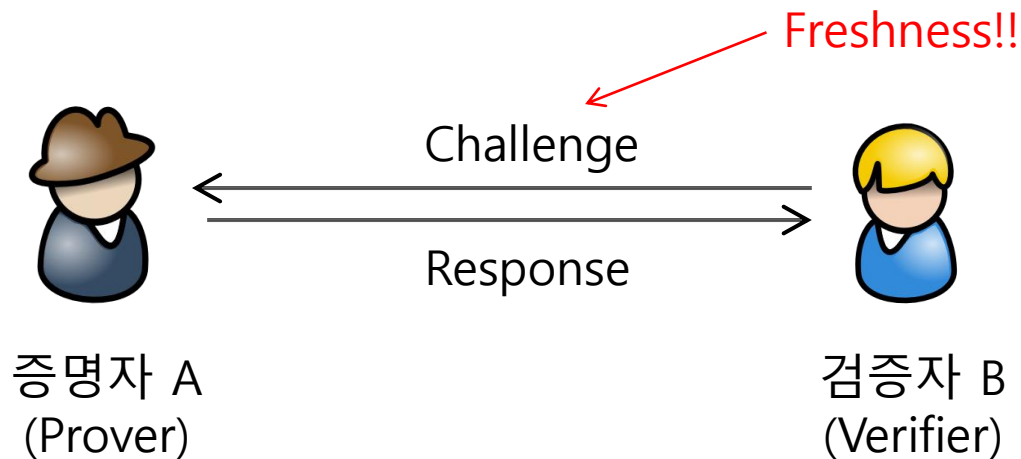


[비동기화 방식의 예]

질의-응답(Challenge-Response) 인증

□ 검증자가 생성한 질의에 대하여 증명자가 응답

- 대칭키를 이용한 방식, 해쉬 함수를 이용한 방식, 공개키를 이용한 방식



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

이 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

동기화된 t_A 를 사용하므로 Challenge과정 불필요



$E_k(t_A, A)$



증명자 A
(Prover)

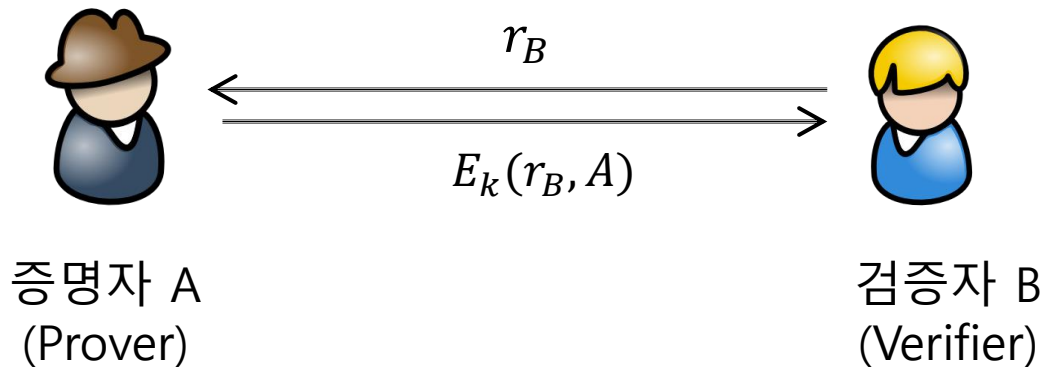
검증자 B
(Verifier)

질의-응답(Challenge-Response) 인증

□ 대칭키를 이용한 질의-응답 인증

▪ 난수(Nonce)를 이용한 단방향 인증

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

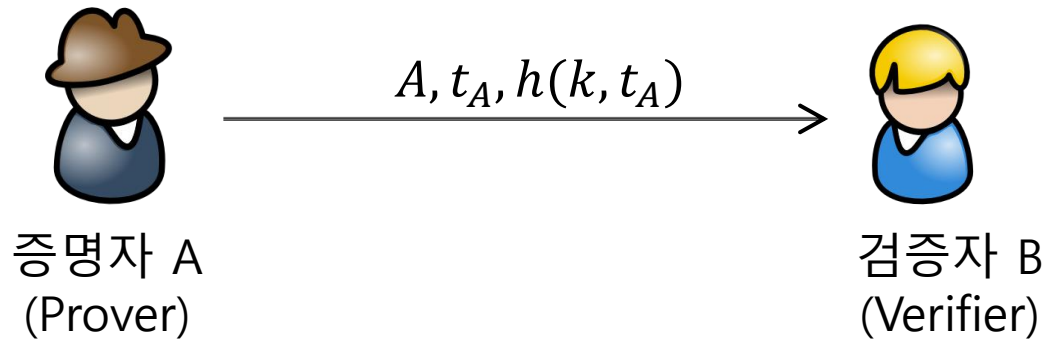


질의-응답(Challenge-Response) 인증

□ 해시 함수를 이용한 질의-응답 인증

- 해시 함수와 타임스탬프를 이용한 단방향 인증

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



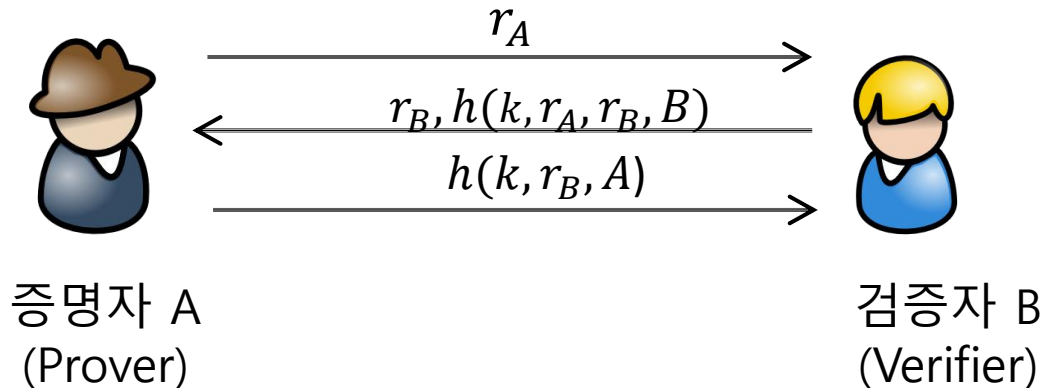
k : Verifier와 Prover사이의 공유한 대칭키

질의-응답(Challenge-Response) 인증

□ 해쉬 함수를 이용한 질의-응답 인증

▪ 난수 nonce)를 이용한 양방향 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. $A \rightarrow B : h(r_B, A)$

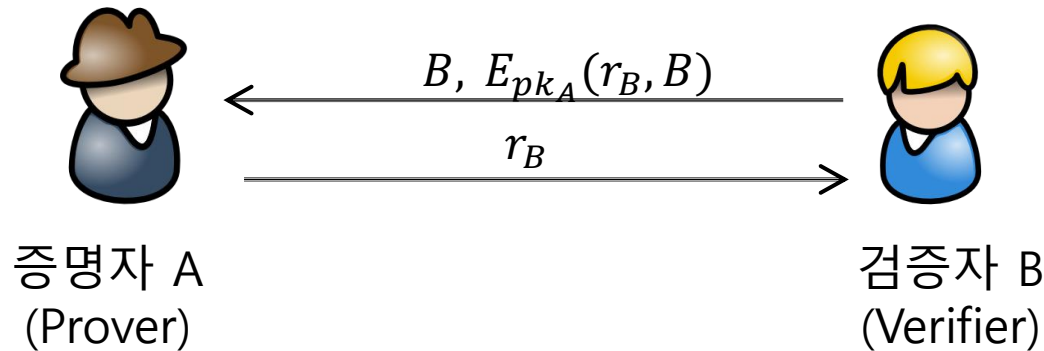


질의-응답(Challenge-Response) 인증

□ 공개키 암호를 이용한 단방향 인증

▪ 난수를 이용한 인증

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



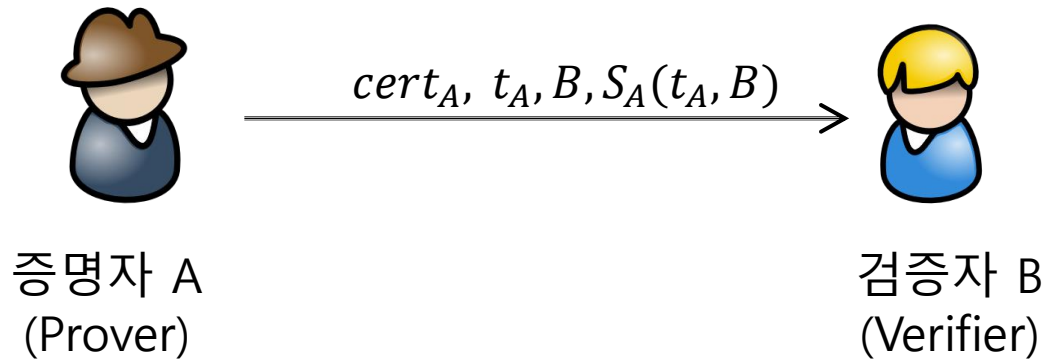
$E_{pk_A}()$: 증명자 (A)의 공개키(pk_A)로 암호화된 값

질의-응답(Challenge-Response) 인증

□ 전자 서명을 이용한 단방향 인증

- 타임스탬프를 이용한 단방향 인증

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



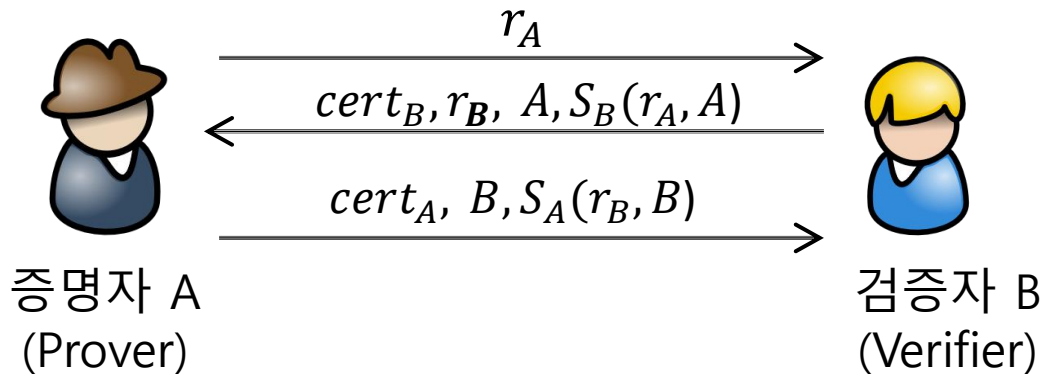
$S_A()$: 증명자 (A)의 개인키로 서명된 값

질의-응답(Challenge-Response) 인증

□ 전자 서명을 이용한 단방향 인증

▪ 난수를 이용한 양방향 인증

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

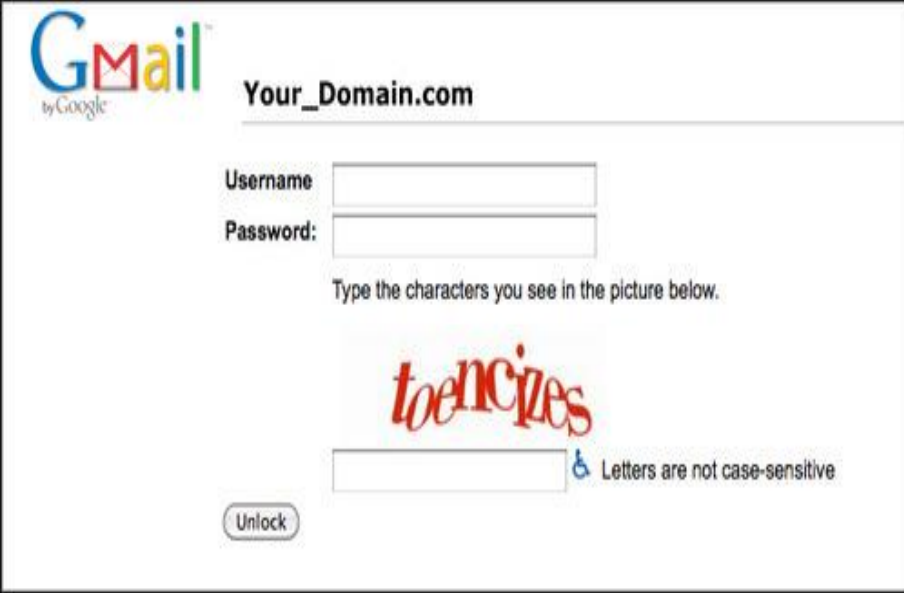


$S_A()$: 증명자 (A)의 개인키로 서명된 값, $S_B()$: 검증자 (B)의 개인키로 서명된 값

차세대 인증: 그래픽 기반 인증

□ CAPTCHA (Completely Automated Public Test to tell Computers and Humans Apart)를 통한 인증

- 악의적인 사용자는 자동화되어 있는 봇을 통하여 대량의 계정 생성을 시도하는 경우가 있음
- 이를 막기 위하여 CAPTCHA를 통한 인증 방식을 사용하고 있음



The image shows a Gmail login interface. At the top left is the Gmail logo with "by Google" underneath. To its right is the text "Your_Domain.com". Below this are two input fields: "Username" and "Password:". Under the password field is a CAPTCHA challenge with the text "Type the characters you see in the picture below." and a distorted image of the word "toencizes" in red. Below the CAPTCHA is another input field and a link that says "Letters are not case-sensitive" with a small icon. At the bottom left is an "Unlock" button.

차세대 인증: 인지 인증

□ 이미지 등록 기반 인증

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 min

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

studyhard



이미지 선택

User ID

Pre-set Challenge/Response

Comparison PASSWORD Login

Your SiteKey:

studyhard



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

* Passcode:

(8 - 20 Characters, case sensitive)

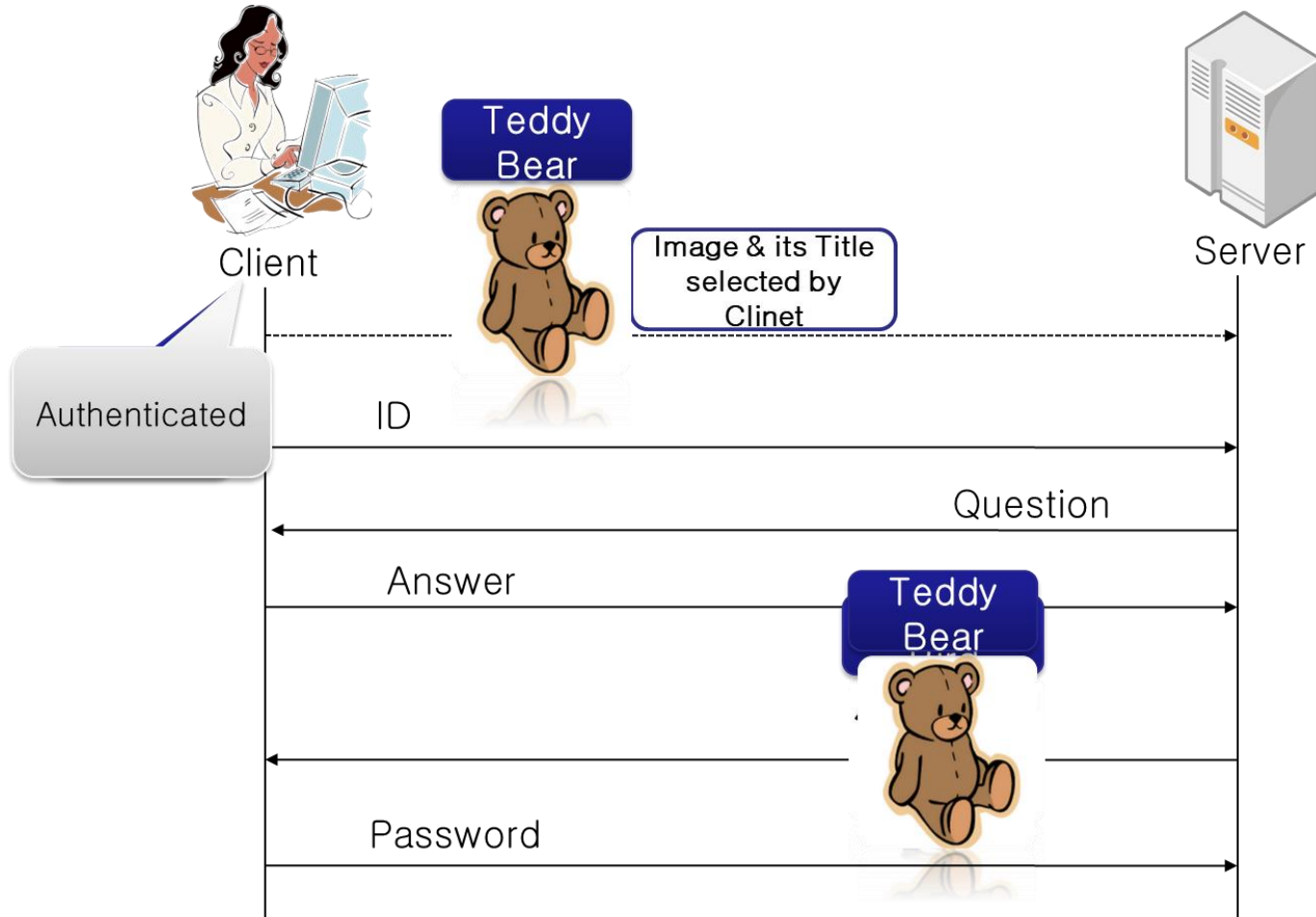
Sign In

2100 10

차세대 인증: 인지 인증

□ 이미지 등록 기반 인증

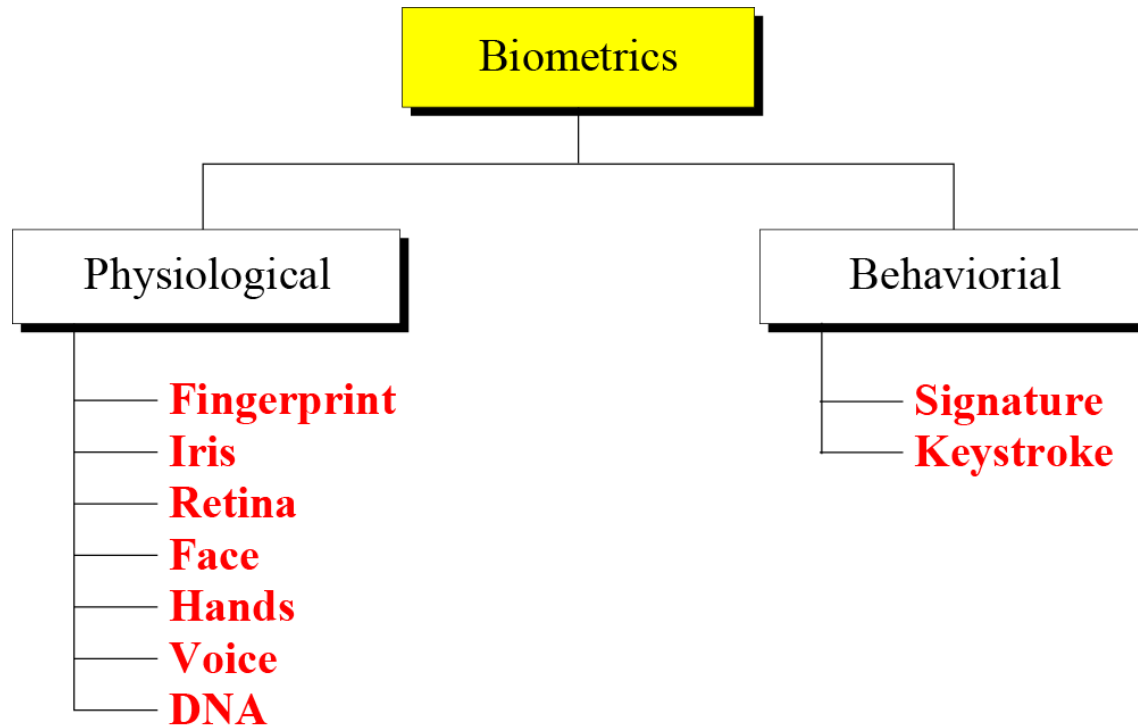
- 사용자를 악의적인 웹 사이트 (Phishing 웹 사이트)부터 보호함



차세대 인증: 생체 인증

□ BIOMETRICS

- Accuracy of biometry techniques
 - False Rejection Rate (FRR)
 - False Acceptance Rate (FAR)

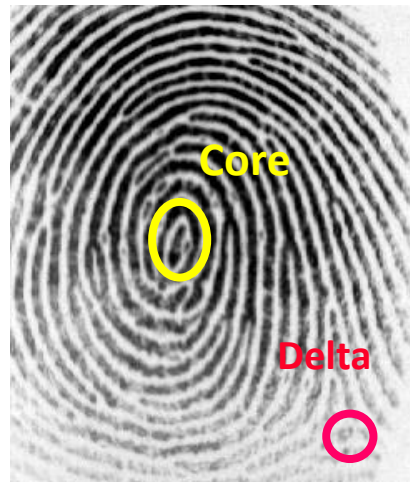
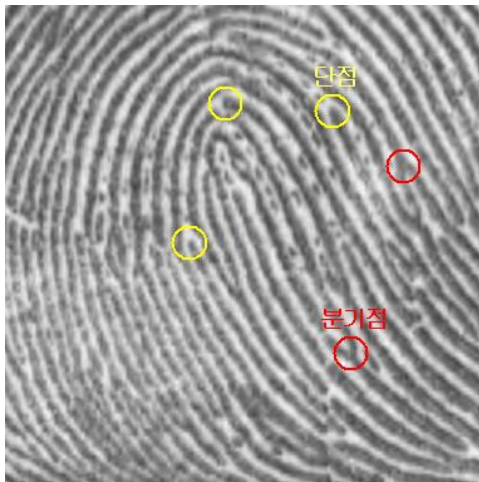


차세대 인증: 생체 인증

□ 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, 스마트폰 로그인



출처 : ETRI

* 일반적으로 여성, 어린이, 노인의 경우는 성인 남성에 비해 지문 패턴이 약하다고 알려져 있음

차세대 인증: 생체 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 제공
- 서명
 - 이미 작성된 서명을 인식하는 정적인 방법
 - 서명하는 과정을 동적으로 파악하는 방법
 - 서명시간, 속도, 종이로부터 펜이 떨어진 횟수 등
- 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 뉴클레오타이드 비교하는 기술
 - 범죄자 확인, 약물복용확인, 친자확인(부계, 모계 확인)등 다양한 요소에서 활용



차세대 인증: 생체 인증

□ 시 기반 얼굴 인식

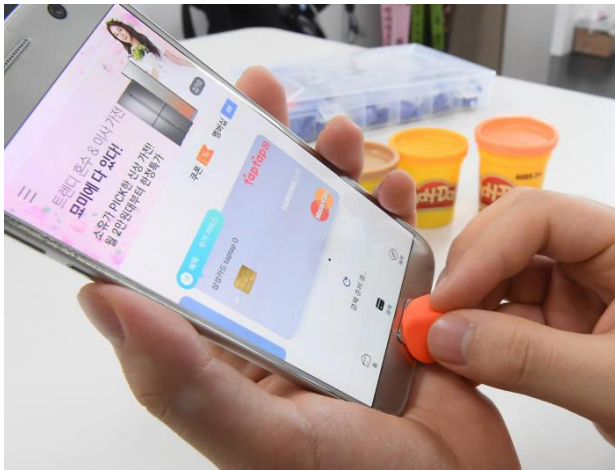


생체 인증의 문제? Liveness check + 정보 유출

차세대 인증: 생체 인증

□ Liveness check 문제

- 지문 위조



스마트폰 지문인식 20분만에 뚫렸다... 中 해킹 성공

노컷뉴스 (보도자료) - 2019. 11. 3.

보안 문제로 구체적인 방식은 공개되지 않았지만, 이 이미지 분석 앱은 3D 프린터를 사용해 지문을 복제해 데이터를 추출하는 방식과 흡사한 것으로 ...

www.digitaltoday.co.kr > news > articleView ▼

갤럭시S8 안면인식 기능 보안 구멍..."사진만으로 잠금해제 ...

2017. 4. 1. - [디지털투데이 홍하나 기자] 최근 삼성이 공개한 갤럭시 S8의 '안면인식 기능'이 사진만 보여줘도 잠금이 해제될 정도로 보안성이 취약한 것으로 ...

□ 정보 유출

- United States Office of Personnel Management (OPM)

www.boannews.com > media > view ▼

23기가에 달하는 개인 식별 정보와 생체 정보 노출 ... - 보안뉴스

2019. 8. 16. - 2015년 미국 OPM 해킹 사건 당시 110만 명의 지문 정보가 유출되기도 했었다. 그러나 슈프리마의 사건은 생체 인증 정보와 더불어 관리자 계정 ...

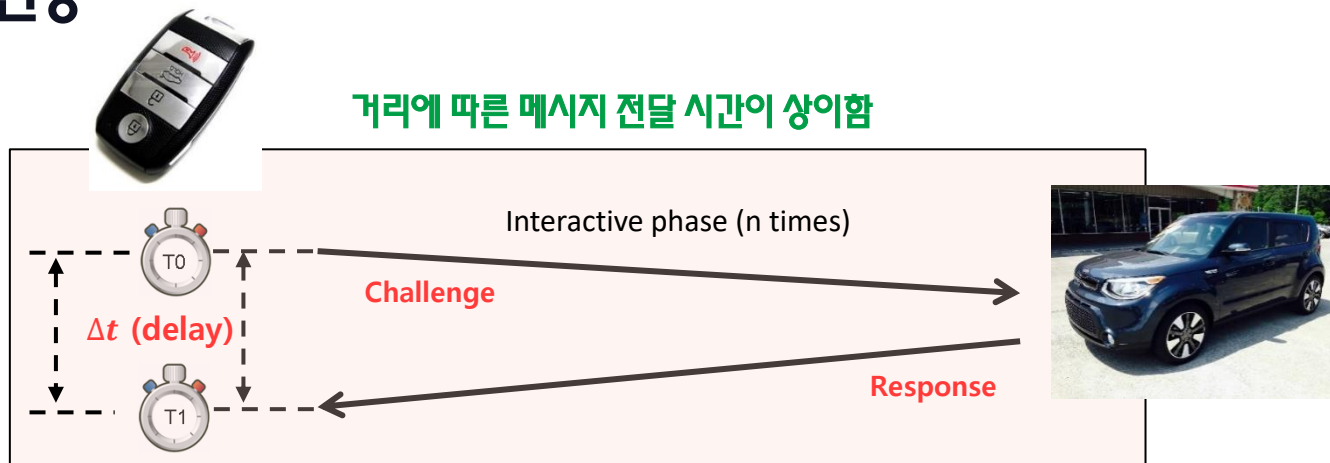
차세대 인증: 거리기반 인증

□ Relay attacks on the passive key less system



<https://www.youtube.com/watch?v=ef8ZrV2xb5g>

□ 거리기반 인증



기타

□ 2 factor 인증의 중요성

- https://www.zdnet.com/article/microsoft-99-9-of-compromised-accounts-did-not-use-multi-factor-authentication/?ftag=COS-05-10aaa0g&taid=5e62abee05296a0001045474&utm_campaign=trueAnthem%3A+Trending+Content&utm_medium=trueAnthem&utm_source=twitter

□ 알고리즘 취약점 (자동차)

- https://www.wired.com/story/hackers-can-clone-millions-of-toyota-hyundai-kia-keys/?utm_brand=wired&utm_social-type=owned&utm_medium=social&mbid=social_twitter&utm_campaign=wired&utm_source=twitter&fbclid=IwAR2HhQt7oVHMFPawKudYuG-R0loLJojrG_Moaxl-TS6N6N3RAQ4vuUJNdLM

Thank you 