

암호학과 네트워크 보안

-14장 개체 인증-

민 윤 홍(yunhong@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 개요
- 패스워드 기반 인증(Password-based authentication)
- 시도-응답 인증(Challenge-Response authentication)
- 영지식 인증(Zero-knowledge authentication)
- 생체인식(Biometrics)

목 차

- 개요
- 패스워드 기반 인증(Password-based authentication)
- 시도-응답 인증(Challenge-Response authentication)
- 영지식 인증(Zero-knowledge authentication)
- 생체인식(Biometrics)

개요

- 개체 인증(Entity authentication)
  - 정의
    - 한 개체가 다른 한 개체의 신원을 증명할 수 있도록 설계된 기술
  - 구성 요소
    - 개체(Entitiy)
      - 자신의 신원을 증명할 수단을 보유하고, 인증 프로토콜에 참여할 수 있는 주체
        - 사람, 프로세스, 클라이언트, 서버 등
      - 주장자(Claimant)
        - 자신의 신원을 증명하고자 하는 개체
      - 검증자(Verifier)
        - 주장자의 신원을 증명하기 위해 노력하는 개체



<그림 1> 개체 인증 개요

개요

- 메시지 인증과 개체 인증의 차이
  - 메시지 인증(데이터-출처 인증)
    - 검증자가 주장자를 인증하는 시간에 주장자가 통신상태를 유지할 수도, 아닐 수도 있음
      - 실시간으로 실시될 수도, 아닐 수도 있음
    - 모든 메시지에 대해서 인증을 함
      - e.g., HTTPS - TLS 레코드 레이어의 MAC
  - 개체 인증
    - 검증자에 의해 주장자가 인증 받는 시간에 통신상태를 유지해야 함
      - 실시간으로 실시됨
    - 한 번 인증이 끝나면 한 세션동안 인증됨
      - e.g., 홈페이지 로그인

개요

- 검증 범주
  - 정의
    - 주장자가 자신의 신원을 증명하기 위해 제시하는 증거의 종류
  - 종류
    - 알고 있는 것(Something known)
      - 주장자만 알고 있는 비밀임
      - 검증자에 의해 점검될 수 있음
      - e.g., 패스워드, 비밀 키, 개인 키 등
    - 소유하고 있는 것(Something possessed)
      - 주장자의 신원을 증명할 수 있는 대상임
      - e.g., 여권, 운전면허증, 신분증, 신용카드 등
    - 태생적으로 가지고 있는 것(Something inherent)
      - 주장자의 타고난 특성임
      - e.g., 손금, 목소리, 홍채 패턴 등

목 차

- 개요
- 패스워드 기반 인증(Password-based authentication)
- 시도-응답 인증(Challenge-Response authentication)
- 영지식 인증(Zero-knowledge authentication)
- 생체인식(Biometrics)

패스워드 기반 인증

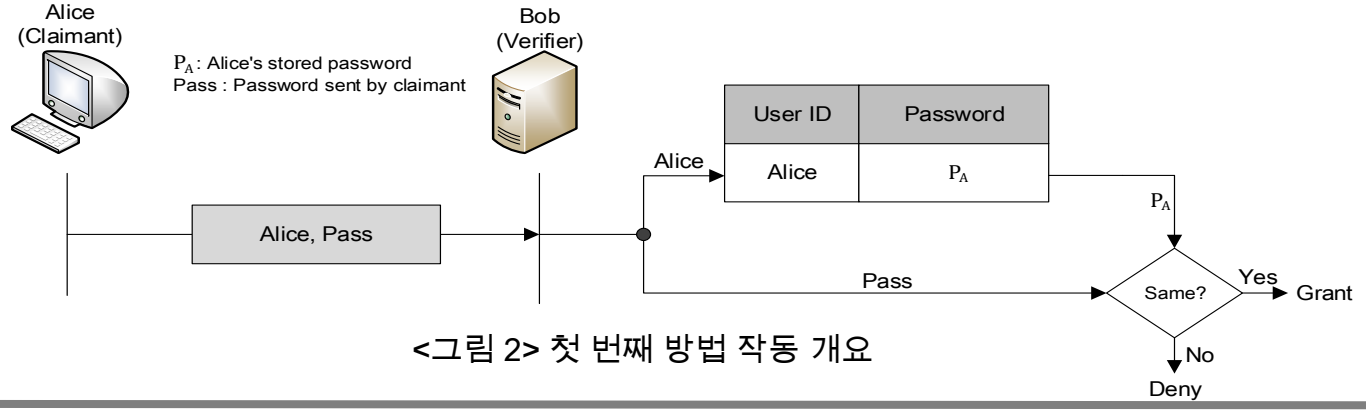
- 패스워드(Password)
  - 정의
    - 주장자가 시스템의 자원을 활용하기 위해 시스템에 접속하고자 할 때 사용하는 문자열
  - 특징
    - 주장자만 아는 비밀 정보를 제시하여 신원을 증명하는 ‘알고 있는 것’에 해당됨
    - 주장자와 검증자가 패스워드를 공유 및 합의하고 있어야 함
  - 종류
    - 고정된 패스워드(Fixed password)
    - 일회용 패스워드(OTP, One-Time Password)

패스워드 기반 인증

- 고정된 패스워드(Fixed password)
  - 정의
    - 주장자가 검증자의 시스템 자원에 접속할 때 동일한 값을 반복해서 사용되는 패스워드
  - 종류
    - 첫 번째 방법
      - 시스템이 평문 패스워드 저장하는 방법
    - 두 번째 방법
      - 시스템이 패스워드의 해시값을 저장하는 방법
    - 세 번째 방법
      - 시스템이 패스워드에 랜덤 문자열을 추가한 후 해시를 적용하여 저장하는 방법
    - 네 번째 방법
      - '알고 있는 것' 과 '소유하고 있는 것' 을 함께 사용하는 방법

패스워드 기반 인증

- 고정된 패스워드 – 첫 번째 방법
  - 시스템이 주장자에 의해 정렬된 표나 파일을 평문으로 저장하는 방법
  - 작동 방식
    1. 시스템 자원에 접속하기 위해서 주장자는 자신의 아이디와 패스워드를 평문 형태로 시스템에 제공함
    2. 검증자의 시스템은 표에서 패스워드를 찾기 위해 아이디를 사용함
    3. 주장자가 보낸 패스워드가 표나 파일에 나타난 패스워드와 일치한다면 접속을 허락하고 그렇지 않으면 접속이 거부됨



<그림 2> 첫 번째 방법 작동 개요

패스워드 기반 인증

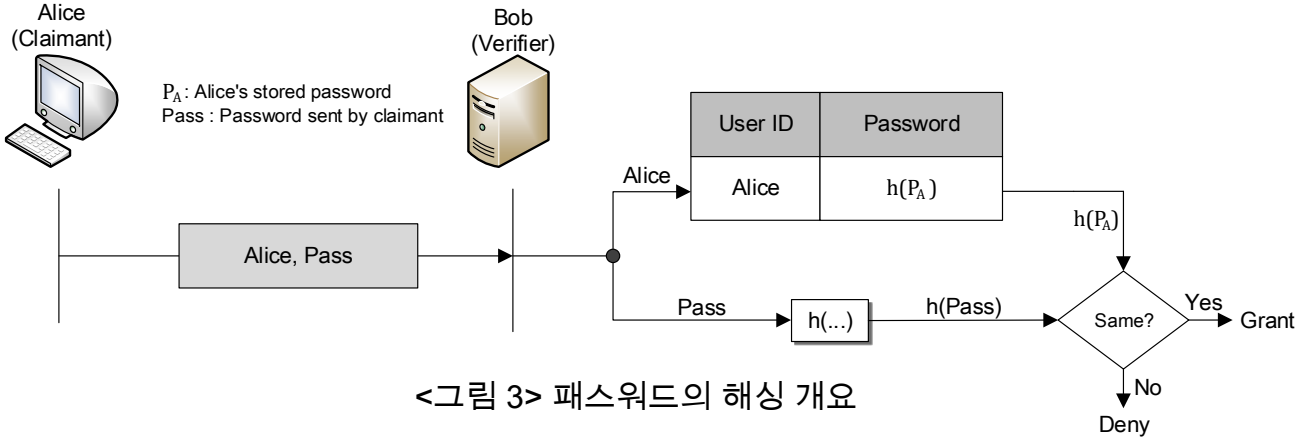
- 고정된 패스워드 – 첫 번째 방법
  - 취약점 (1/2)
    - 패스워드 도청
      - 공격자는 통신선을 통해 도청을 하여 패스워드 메시지를 얻을 수 있음
      - 패스워드로 매번 다른 값을 사용하여 대응할 수 있음
    - 패스워드 파일 접근
      - 공격자가 시스템을 해킹하여 아이디, 패스워드 파일에 대한 접근을 시도함
      - 공격자가 접근을 해도 패스워드 파일을 알아보지 못 하도록 패스워드 파일을 변형하여 대응 가능함

패스워드 기반 인증

- 고정된 패스워드 – 첫 번째 방법
  - 취약점 (2/2)
    - 패스워드 도난
      - 공격자가 주장자의 패스워드를 물리적으로 훔쳐내려고 시도함
      - 주장자는 패스워드를 메모 등에 작성하지 말고 기억해야 함
    - 패스워드 추측
      - 주장자는 패스워드를 기억하기 쉬운 조합으로 생성하는 경우, 공격자는 이를 악용하여 패스워드를 추측할 수 있음
      - 추측이 쉽거나 매우 짧은 패스워드를 사용하는 것은 지양해야 함

패스워드 기반 인증

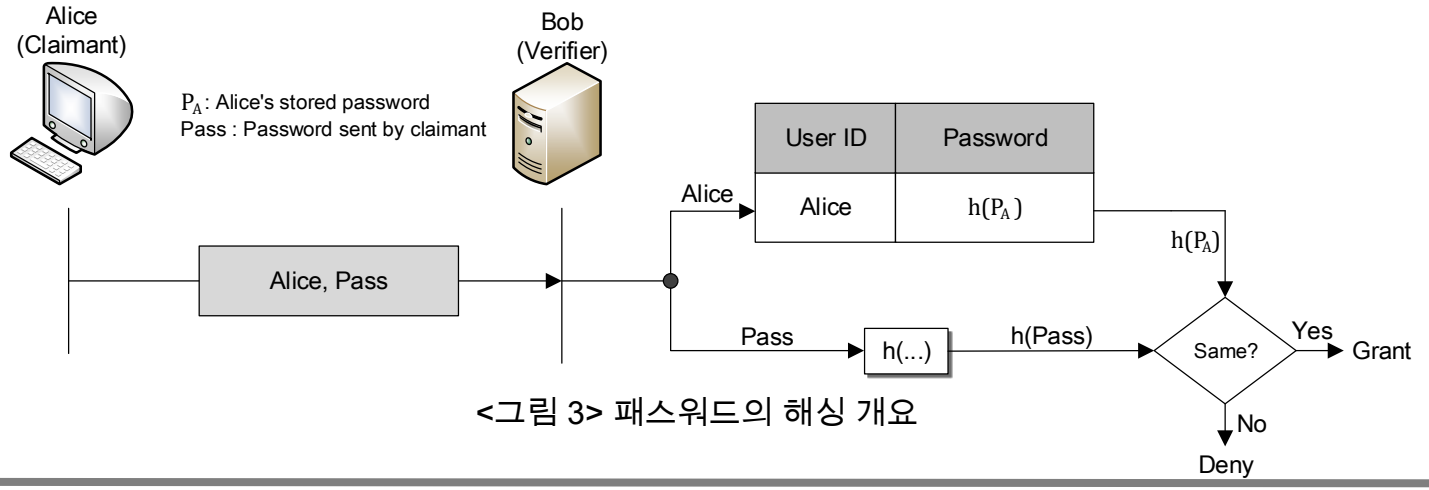
- 고정된 패스워드 – 두 번째 방법
  - 시스템이 패스워드 파일에 평문 대신 패스워드의 해시값을 저장하는 방법
  - 작동 방식
    1. 주장자가 패스워드를 생성하면 시스템은 패스워드를 해시함수로 처리한 다음 해시 값을 패스워드 파일에 저장함
    2. 주장자가 아이디와 패스워드를 보내면 시스템은 주장자의 패스워드를 사용해 해시를 생성하고 파일에 저장된 해시와 동일한지 아닌지 비교함



<그림 3> 패스워드의 해싱 개요

패스워드 기반 인증

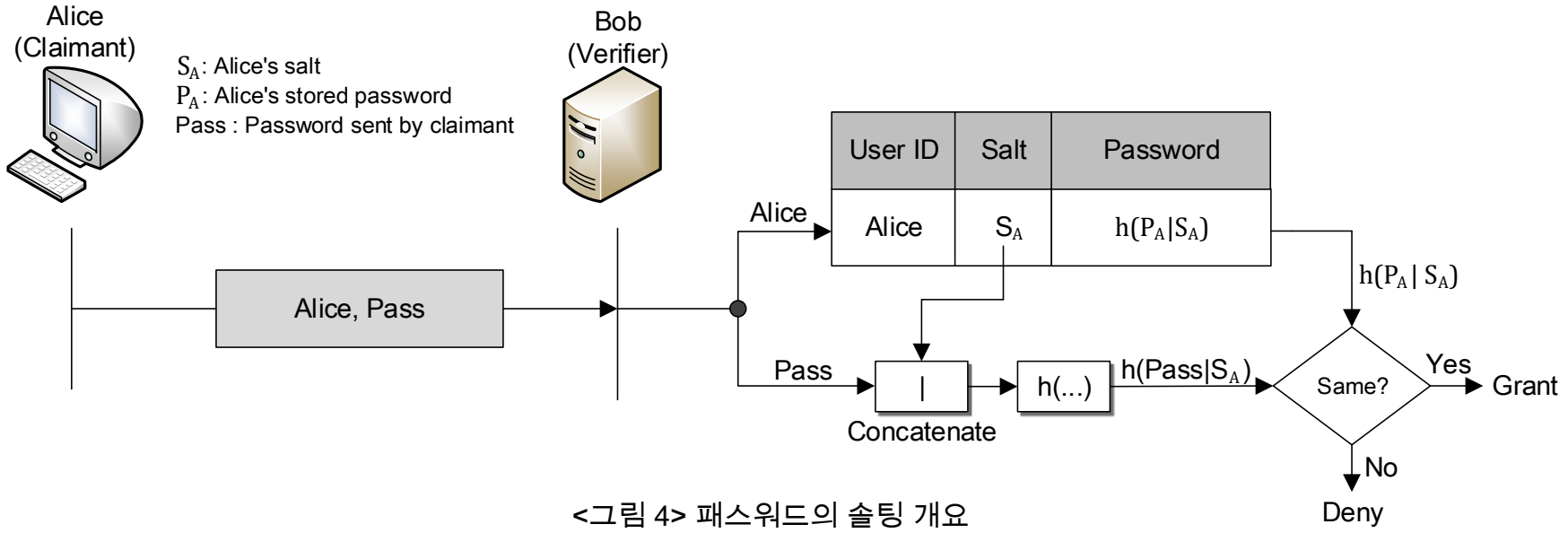
- 고정된 패스워드 – 두 번째 방법
  - 취약점
    - 사전 공격(Dictionary attacks)
      - 공격자는 주장자들이 자주 사용하는 패스워드에 해시 함수를 적용시켜, 패스워드 – 해시 목록을 파악할 수 있음
      - 공격자는 시스템의 패스워드 파일을 탈취해 만들어 둔 패스워드 – 해시 목록과 비교하여 일치하는 것을 찾을 수 있음
      - 해시 함수를 적용하기 이전에 패스워드에 랜덤 문자열을 추가하는 방식으로 대응할 수 있음



<그림 3> 패스워드의 해싱 개요

패스워드 기반 인증

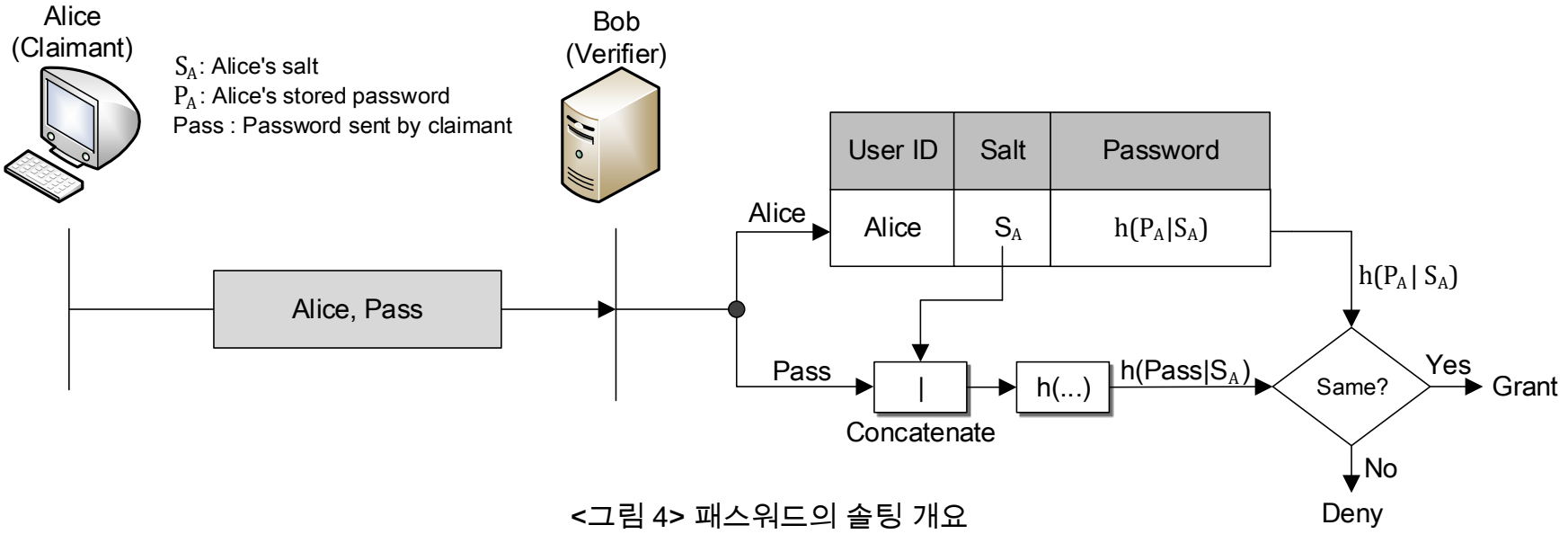
- 고정된 패스워드 – 세 번째 방법(Salting)
  - 패스워드에 랜덤 문자열인 솔트(Salt)를 추가한 후 해시함수를 적용하여 저장하는 방법
  - 작동 방식 (1/2)
    1. 시스템은 주장자가 패스워드를 생성하면 패스워드에 솔트를 붙인 후 해시 함수를 적용한 값과 아이디, 솔트를 파일에 저장함



<그림 4> 패스워드의 솔팅 개요

패스워드 기반 인증

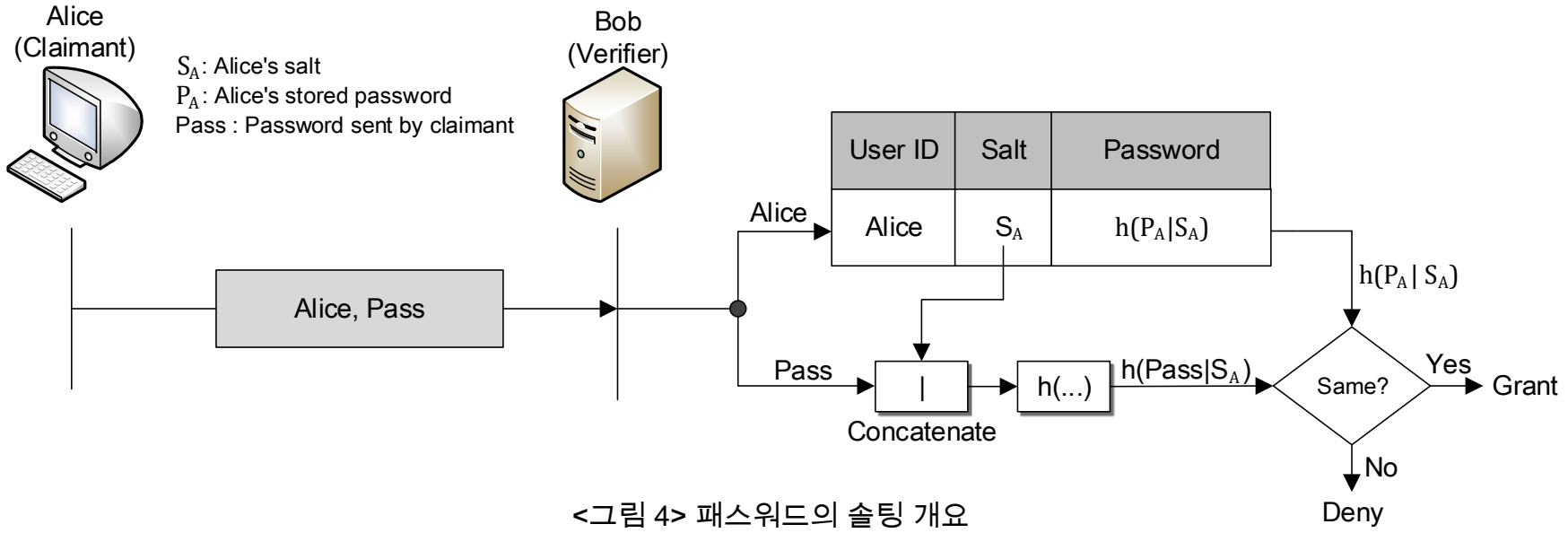
- 고정된 패스워드 – 세 번째 방법(Salting)
  - 작동 방식 (2/2)
    2. 주장자가 접근을 요청하면, 시스템은 패스워드 파일에서 솔트를 추출해내고 수신된 패스워드에 이어 붙인 후 해시를 계산함
    3. 계산한 해시 값과 파일에 저장해 놓았던 해시 값과 비교함



<그림 4> 패스워드의 솔팅 개요

패스워드 기반 인증

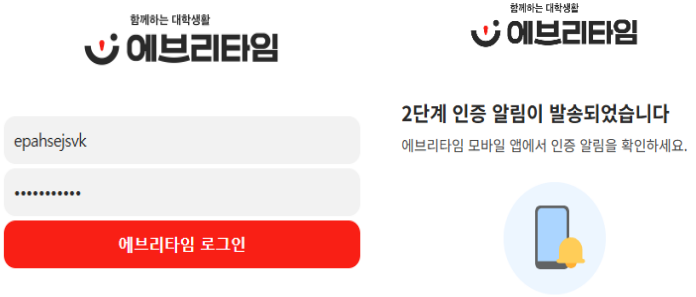
- 고정된 패스워드 – 세 번째 방법(Salting)
  - 취약점
    - 대상을 특정한 공격에는 취약함
      - 미리 계산된 해시를 재사용하는 공격에는 강건하나, 해시값이 유출되었을 때 특정 주장자의 후보 비밀번호에 대입 및 비교하는 공격에는 취약함
      - 의도적으로 메모리 연산을 많이 하도록 하는 해시 함수를 사용하여 대응할 수 있음



<그림 4> 패스워드의 솔팅 개요

패스워드 기반 인증

- 고정된 패스워드 – 네 번째 방법
  - ‘알고 있는 것’인 패스워드와 ‘소유하고 있는 것’을 함께 사용하는 방법
  - 활용 사례
    - PIN (Personal Identification Number, 개인 식별 번호)사용
      - ‘소유하고 있는’ 카드를 사용할 때 ‘알고 있는’ PIN을 사용함
      - PIN의 특성 상 짧기 때문에 추측 공격에 취약함
    - SMS 인증 사용
      - 고정된 패스워드인 비밀번호에 추가로 SMS 인증을 더하는 방법
      - SMS 탈취 시 무력화 될 수 있음



<그림 5> 에브리타임 웹의 SMS 추가 인증

패스워드 기반 인증

- 고정된 패스워드
  - 정리

[표 1] 고정된 패스워드 정리

구분	첫 번째 방법	두 번째 방법	세 번째 방법	네 번째 방법
저장 형태	패스워드 평문 P	해시값 $h(P)$	솔트 + 해시값$h(P\|S)$	해시 저장 및 소유 요소 추가
작동 방식	표에 아이디와 패스워드를 그대로 보관	패스워드를 알아볼 수 없게 해시로 변환	주장자마다 다른 랜덤값을 더한 후 해시처리	'알고 있는 것'에 '소유하고 있는 것' 결합
직전 세대 대비 보완점	-	비밀번호를 평문 대신 해시값으로 저장하도록 보완, 파일이 유출되어도 평문이 드러나지 않음	해시를 미리 계산하여 대조하는 공격을 솔트로 보완, 같은 비밀번호도 서로 다른 해시값이 저장됨	하나의 비밀번호에 의존하던 방식을 보완, 소유물이 없으면 인증 불가
취약점	도청, 도난, 파일 접근, 추측에 취약	사전 공격에 취약	대상을 특정한 공격에 취약	소유물 도난, SMS 탈취와 같이 소유하고 있는 것의 분실에 취약

패스워드 기반 인증

- 일회용 패스워드(OTP, One-Time Password)
  - 정의
    - 인증 시에 오직 한 번만 사용되는 패스워드
  - 종류
    - 첫 번째 방법
      - 주장자와 시스템이 사전에 사용할 패스워드를 정하는 방법
    - 두 번째 방법
      - 주장자와 시스템이 패스워드를 순차적으로 업데이트하는 방법
    - 세 번째 방법
      - 주장자와 시스템이 해시함수를 이용하여 순차적으로 패스워드를 생성하는 방법

패스워드 기반 인증

- 일회용 패스워드 – 첫 번째 방법
  - 주장자와 시스템이 사전에 사용할 패스워드 목록을 정해놓는 방식
  - 작동 방식
    - 주장자와 시스템이 패스워드 목록에 합의를 함
      - e.g., 첫 번째 비밀번호는 1111, 두 번째는 2222
    - 목록상의 각 패스워드는 오직 한 번만 사용하기로 함
  - 취약점
    - 합의과정에서 공격자에게 목록이 탈취된다면 모든 패스워드가 유출됨
    - 주장자가 패스워드를 순서대로 사용하지 않는다면 시스템은 일치하는 패스워드를 찾는데 시간을 많이 소비함

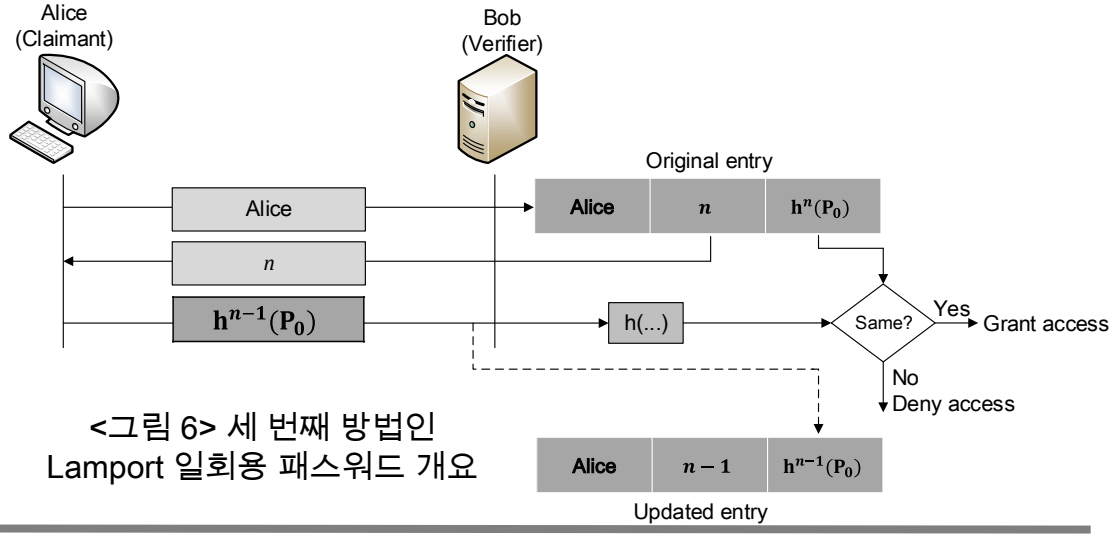
패스워드 기반 인증

- 일회용 패스워드 – 두 번째 방법
  - 주장자와 시스템이 패스워드를 순차적으로 업데이트하는 방법
  - 작동 방식
    1. 주장자와 시스템은 패스워드 P_1에 동의함
    2. 첫 번째 접속 시도에서 주장자는 새로운 패스워드 P_2를 생성하고 P_1를 키로 사용해서 암호화함
    3. P_2는 두 번째 접속시에 패스워드로 사용됨
    4. 두 번째 접속 시도에서 주장자는 새로운 패스워드 P_3를 생성하고 P_2를 키로 사용해서 암호화함
    5. P_3는 세 번째 접속시에 패스워드로 사용됨
    6. 이를 반복하여 P_i를 이용하여 P_{i+1}을 생성함
  - 취약점
    - 만약 공격자가 첫 번째 패스워드인 P_1을 알아낸다면 이어지는 모든 패스워드를 알 수 있음

패스워드 기반 인증

- 일회용 패스워드 – 세 번째 방법
  - 주장자와 시스템이 해시함수를 이용해서 순차적으로 업데이트된 패스워드를 생성하는 방법
  - 작동 방식 (1/3)
    1. 주장자와 시스템은 원래의 패스워드 P_0와 카운터 n을 정함
    2. 시스템은 $h^n(P_0)$를 계산함
    3. 시스템은 주장자의 아이디, n, $h^n(P_0)$를 저장함
       - h^n은 해시함수를 n번 적용한 것, 식으로 정리하면 다음과 같음

$$
\begin{aligned}
& h^n(x) \\
&= h\left(h^{n-1}(x)\right), h^{n-1}(x) \\
&= h\left(h^{n-2}(x)\right), \ldots h^2(x) \\
&= h\left(h(x)\right), \\
& h^1(x) = h(x)
\end{aligned}
$$

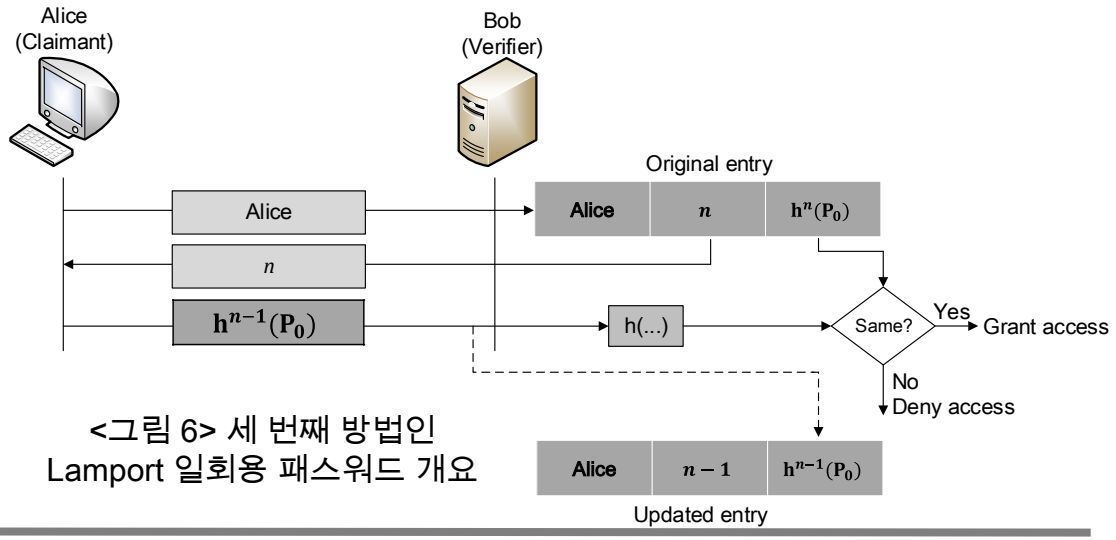


<그림 6> 세 번째 방법인
Lamport 일회용 패스워드 개요

패스워드 기반 인증

- 일회용 패스워드 – 세 번째 방법
  - 작동 방식 (2/3)
    4. 시스템이 주장자의 응답을 받게 되면, 수신된 값에 해시를 적용하여 엔트리에 저장된 값과 일치하는지 확인함
    5. 시스템은 엔트리의 값 n을 줄여 $n-1$로 바꾸고 패스워드 $h^n(P_0)$값을 새로운 값 $h^{n-1}(P_0)$로 변경함
    6. 주장자가 두 번째 접속을 시도한다면 시스템이 수신하는 카운터의 값은 $n-1$이 되고, 주장자가 보낸 세 번째 메시지는 $h^{n-2}(P_0)$임

$$
\begin{aligned}
& h^n(x) \\
&= h\left(h^{n-1}(x)\right), h^{n-1}(x) \\
&= h\left(h^{n-2}(x)\right), \ldots h^2(x) \\
&= h\left(h(x)\right), \\
& h^1(x) = h(x)
\end{aligned}
$$

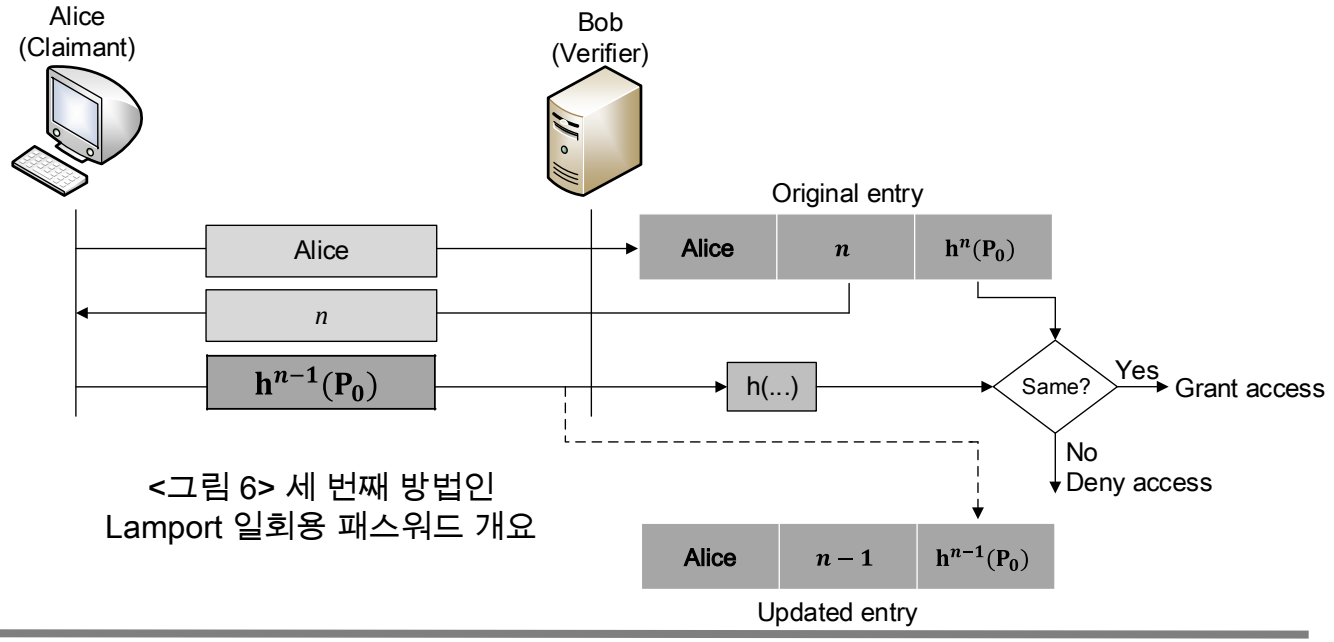


<그림 6> 세 번째 방법인
Lamport 일회용 패스워드 개요

패스워드 기반 인증

- 일회용 패스워드 – 세 번째 방법
  - 작동 방식 (3/3)
    7. 시스템이 해당 메시지를 받게 되면 해시함수를 적용하여 $h^{n-1}(P_0)$를 구하고 업데이트된 엔트리와 비교함
    8. 엔트리에 저장된 n값은 매번 접속이 있을 때마다 1씩 줄임
       - n값이 0이 되면 주장자는 더 이상 시스템에 접속을 할 수 없게 되고 처음부터 다시 설정해야 해서, 보통 1000이상의 큰 수로 정함

$$
\begin{aligned}
&h^n(x)\\
&= h\left(h^{n-1}(x)\right), h^{n-1}(x)\\
&= h\left(h^{n-2}(x)\right), \ldots h^2(x)\\
&= h\left(h(x)\right),\\
&h^1(x) = h(x)
\end{aligned}
$$



<그림 6> 세 번째 방법인
Lamport 일회용 패스워드 개요

패스워드 기반 인증

- 일회용 패스워드
  - 정리

[표 2] 일회용 패스워드 정리

구분	첫 번째 방법	두 번째 방법	세 번째 방법
작동 방식	미리 합의한 목록의 패스워드를 하나씩 한 번만 사용함	접속 시마다 새 패스워드를 만들어 직전 패스워드를 키로 암호화함	$h^n(P_0)$에서 인증마다 해시 적용 횟수를 하나씩 낮춰 사용함
생성 및 진행 방향	고정된 목록을 차례로 소비함	$P_i \rightarrow P_{i+1}$로 순방향 생성	$h^n \rightarrow h^{n-1}$ 로 역방향 생성
직전 세대 대비 보완점	-	긴 목록에 대한 저장 부담 완화	값이 유출돼도 다음 값을 역산하지 못함
취약점	합의한 목록에 대한 저장 부담	P_1이 노출되면 이후 전체가 노출될 수 있음	n값이 0이 되면 재설정 필요

목 차

- 개요
- 패스워드 기반 인증(Password-based authentication)
- 시도-응답 인증(Challenge-Response authentication)
- 영지식 인증(Zero-knowledge authentication)
- 생체인식(Biometrics)

시도 – 응답 인증

- 정의
  - 검증자가 매번 새로운 시도(문제, Challenge)를 주장자에게 던지고 그에 대한 응답(Response)을 요구하는 인증 방식
    - e.g., TLS 핸드셰이크
  - 시도
    - 검증자가 보내는 예측이 불가능한 값
    - e.g., 난수, 타임스탬프 등
  - 응답
    - 시도에 사전에 서로 합의된 알고리즘을 적용하여 얻는 값
      - e.g., 해시 값, 타임스탬프 등
- 특징
  - 주장자는 자신의 비밀을 노출하지 않으면서도 자신이 알고 있는 것을 증명할 수 있음

시도 – 응답 인증

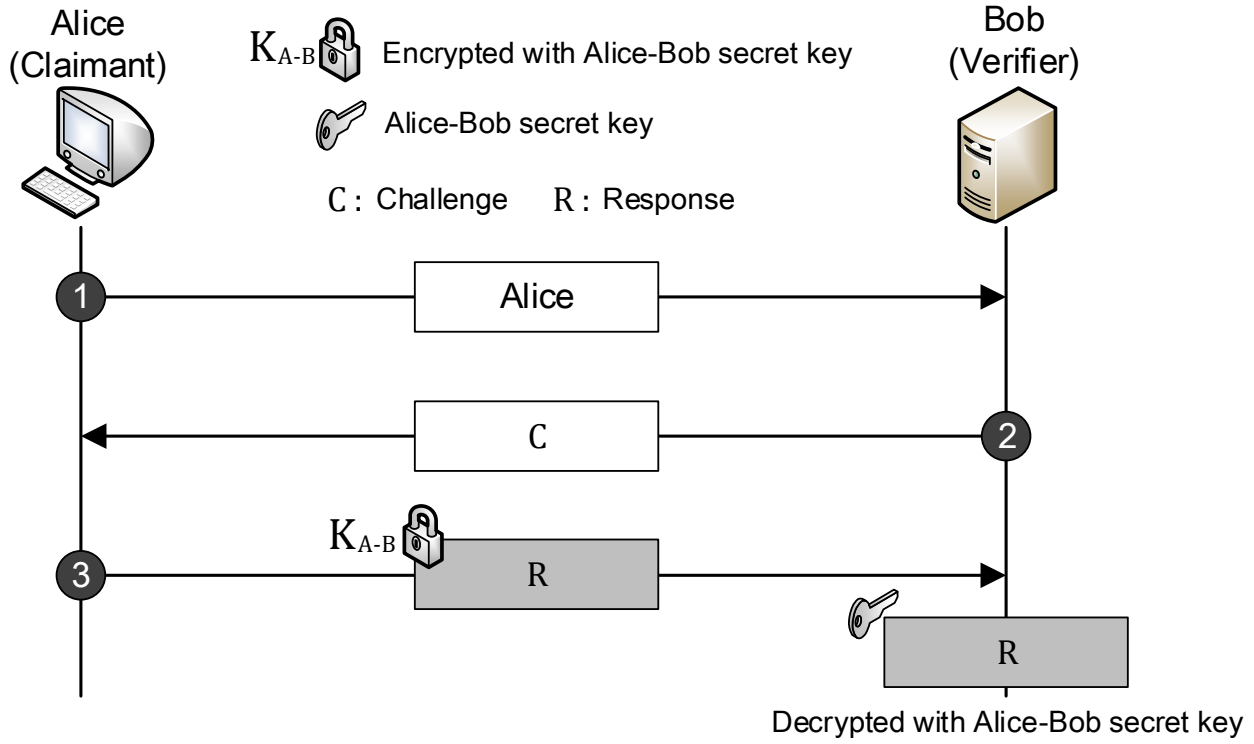
- 종류
  - 대칭-키 암호를 이용한 인증
  - 키-해시함수를 이용한 인증
  - 비대칭-키 암호를 이용한 인증
  - 디지털 서명을 이용한 인증

시도 – 응답 인증

- 대칭-키 암호를 이용한 인증
  - 정의
    - 시도-응답 인증에 대칭-키 암호화를 사용하는 인증 방식
  - 특징
    - 주장자와 검증자가 비밀 키를 같이 공유하고 있음
    - 함수는 시도에 적용되는 암호화 알고리즘임
  - 종류
    - 비표(Nonce) 시도를 이용한 인증
    - 타임스탬프를 이용한 인증
    - 양방향 인증을 이용한 인증

시도 – 응답 인증

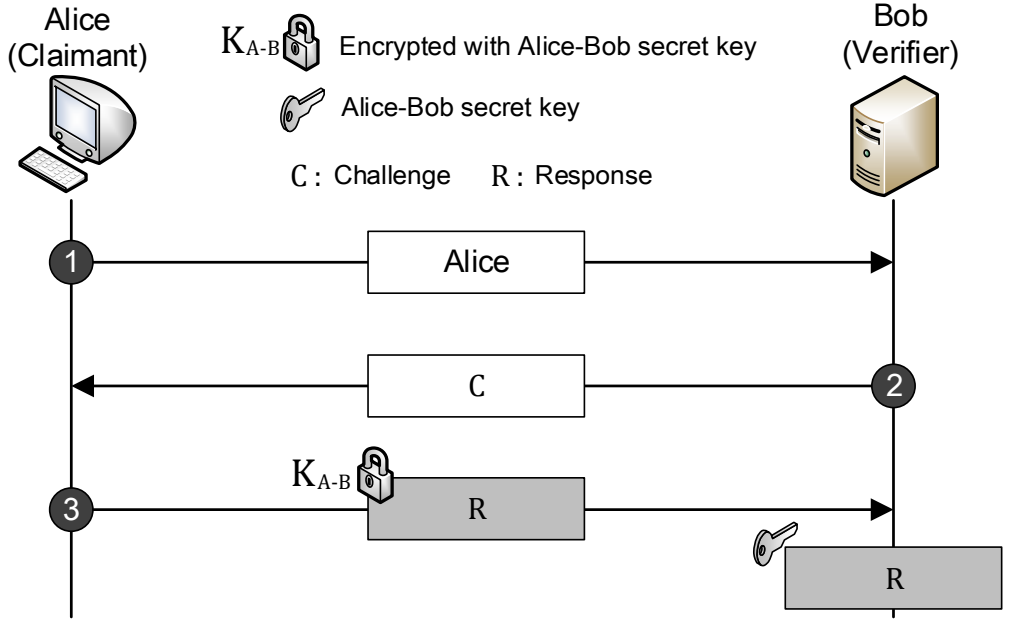
- 대칭-키 암호를 이용한 인증
  - 비표(Nonce) 시도를 이용한 인증
    - 한 번만 사용하게 될 난수를 시도로 사용하는 인증 방식
      - 비표는 공격자의 재전송 공격을 막기 위해, 시간에 따라 달라지는 값이어야만 함



<그림 7> 비표를 시도로 사용한 인증 방식

시도 – 응답 인증

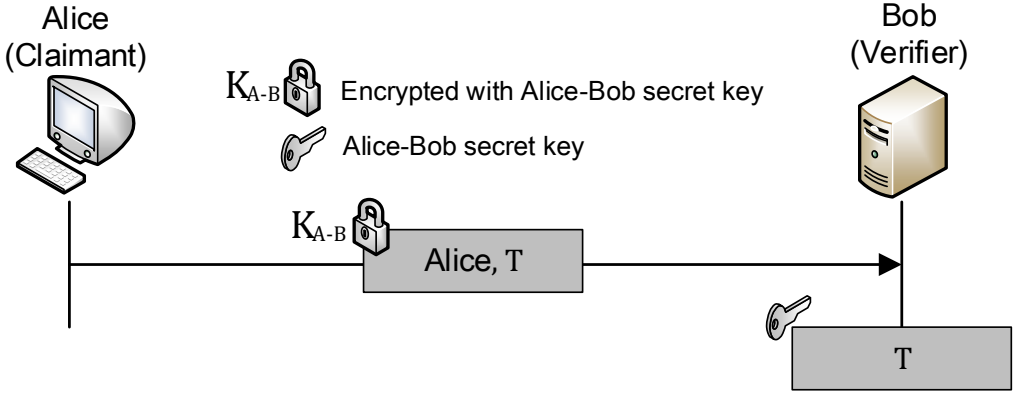
- 대칭-키 암호를 이용한 인증
  - 비표(Nonce) 시도를 이용한 인증
    - 작동 방식
      1. 주장자가 검증자에게 자신의 식별자를 보내 인증 시작을 요청함
      2. 검증자는 시도인 비표를 주장자에게 보냄
      3. 주장자는 검증자와 공유하고 있는 비밀 키로 비표를 암호화하여 보내면 검증자에게 보내면 검증자는 수신한 암호화된 비표를 복호화함



<그림 7> 비표를 시도로 사용한 인증 방식

시도 – 응답 인증

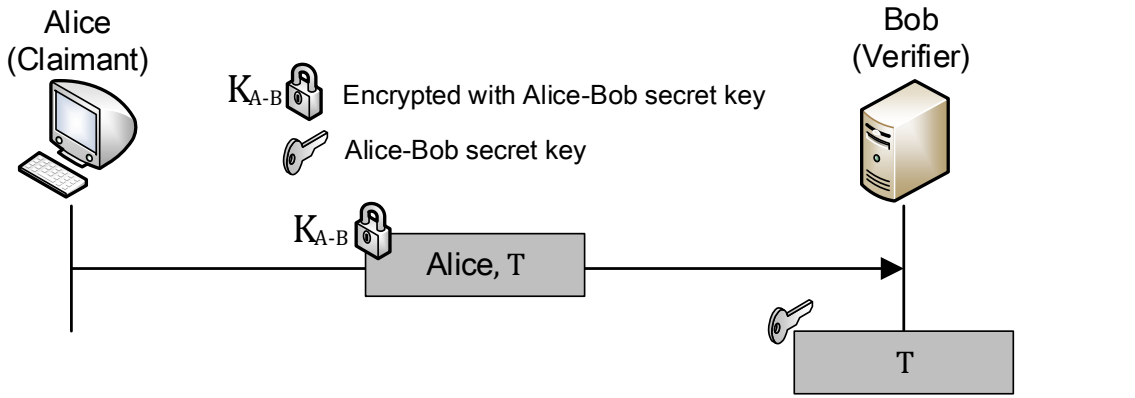
- 대칭-키 암호를 이용한 인증
  - 타임스탬프 시도를 이용한 인증
    - 현재 시간을 반영한 값인 타임스탬프를 사용한 인증 방식
    - 작동 방식
      - 주장자가 현재 시간을 포함한 메시지를 공유하고 있는 비밀 키를 이용해 암호화 한 후, 검증자에게 보냄
    - 특징 (1/2)
      - 주장자의 단일 메시지로 인증을 함
        - 주장자와 검증자의 시계는 동기화 되어 있어야만 함
        - 주장자가 현재 시간을 정확하게 알고 있어야 함



<그림 8> 타임스탬프 시도

시도 – 응답 인증

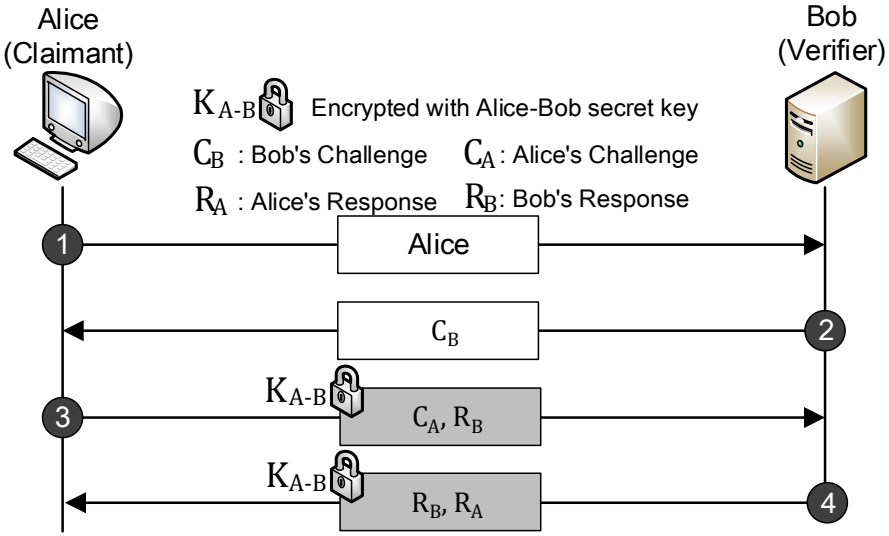
- 대칭-키 암호를 이용한 인증
  - 타임스탬프 시도를 이용한 인증
    - 특징 (2/2)
      - 전 세계 표준 시간인 UTC (Universal Time Coordinated)를 사용함
        - 1970년 1월 1일 00:00:00부터의 경과 시간을 초로 나타낸 시간임
      - 송, 수신 장애 등을 고려해 오차 범위를 설정함
        - 검증자가 정하나, 일반적인 프로토콜 표준이나 시스템 설정에서는 미리 규정함
          - e.g., Kerberos인증 시스템은 5분의 오차범위를 가짐



<그림 8> 타임스탬프 시도

시도 – 응답 인증

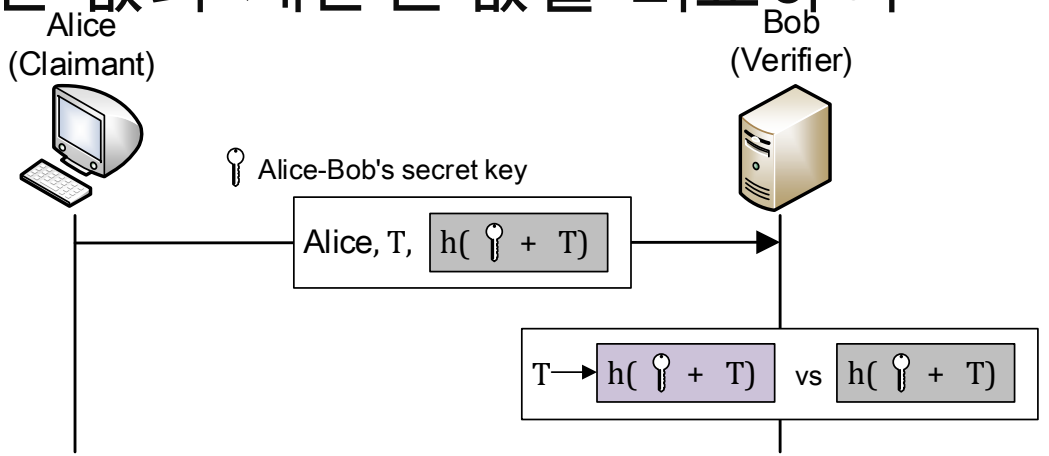
- 대칭-키 암호를 이용한 인증
  - 양방향 시도를 이용한 인증
    - 검증자와 주장자 모두 신원을 인증하는 방식
    - 작동 방식
      1. 주장자가 검증자에게 자신의 식별자를 보내 인증 시작을 요청함
      2. 검증자가 주장자에게 시도를 보냄
      3. 주장자는 검증자의 시도에 응답을 하고 검증자에게 자신의 시도를 보냄
      4. 검증자가 주장자의 시도에 대한 응답을 주장자에게 보냄



<그림 9> 양방향 인증

시도 – 응답 인증

- 키-해시함수를 이용한 인증
  - 인증에 암호화, 복호화를 사용하는 대신에 키-해시함수 기반 MAC(key-Hashed Message Authentication Code)을 사용하는 방식
  - 작동 방식
    1. 타임스탬프는 평문으로도 보내고 키-해시함수를 이용해 변경된 값으로도 보냄
    2. 검증자가 메시지를 수신하면 평문 T에 키-해시함수를 적용한 후, 자신이 수신한 값과 계산된 값을 비교하여 인증을 결정함



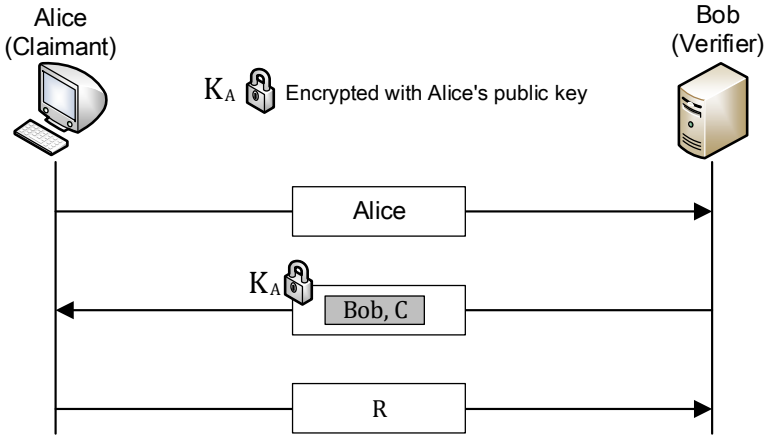
<그림 10> 타임스탬프 방식에 키-해시함수 인증 결합한 인증

시도 – 응답 인증

- 비대칭-키 암호를 이용한 인증
  - 정의
    - 시도-응답 인증에 비대칭-키 암호를 사용하는 인증 방식
  - 특징
    - 주장자는 자신의 공개 키와 쌍을 이루고 있는 개인 키의 소유자임을 보여야만 함
    - 함수는 주장자의 개인 키로 복호화하는 알고리즘임
  - 종류
    - 일방향 비대칭-키 인증
    - 양방향 비대칭-키 인증

시도 – 응답 인증

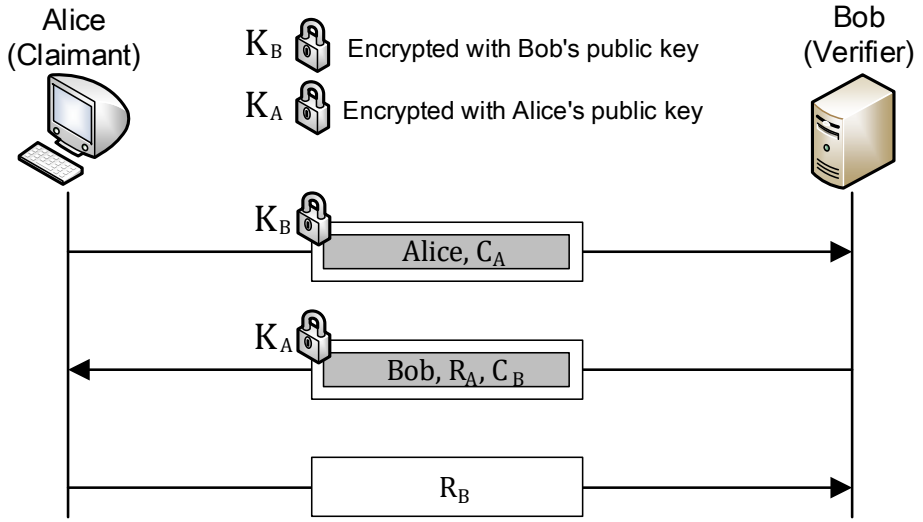
- 비대칭-키 암호를 이용한 인증
  - 일방향 비대칭-키 인증
    - 주장자가 자신의 개인 키 소유를 증명하여 신원을 인증하는 방식
    - 작동 방식
      1. 주장자가 검증자에게 자신의 식별자를 보내 인증 시작을 요청함
      2. 검증자가 시도를 주장자의 공개 키로 암호화하여 만든 후 주장자에게 전송함
      3. 주장자는 해당 시도를 자신의 개인 키로 복호화 한 후 검증자에게 전송함



<그림 11> 일방향 비대칭-키 인증

시도 – 응답 인증

- 비대칭-키 암호를 이용한 인증
  - 양방향 비대칭-키
    - 주장자와 검증자가 서로를 동시에 인증하는 인증 방식
    - 작동 방식
      1. 주장자는 자신의 식별자와 검증자의 공개 키로 암호화한 비표를 전송함
      2. 검증자는 주장자의 공개 키로 암호화된 자신의 비표로 응답함
      3. 주장자는 응답으로 검증자의 복호화된 비표를 전송함



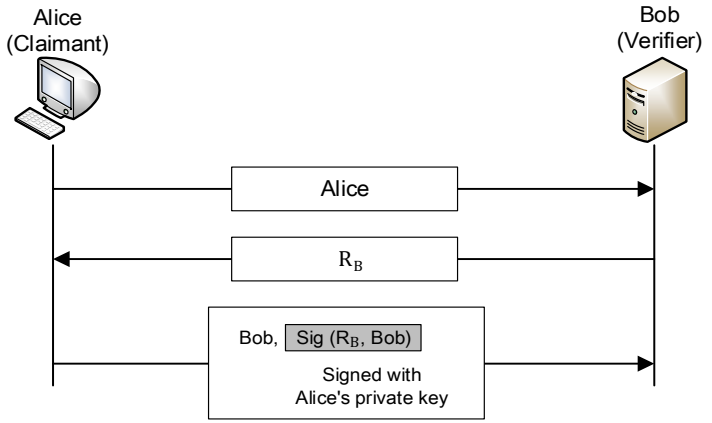
<그림 12> 양방향 비대칭-키 인증

시도 – 응답 인증

- 디지털 서명을 이용한 인증
  - 정의
    - 주장자가 자신의 개인 키를 가지고 서명을 하여 인증을 하는 방식
  - 특징
    - 부인 방지 성질이 있음
      - 개인 키 소유자만 서명을 만들 수 있으므로, 검증 가능한 증거를 만들 수 있음
    - 함수는 서명 및 서명 검증 알고리즘임
  - 종류
    - 일방향 디지털 서명을 이용한 인증
    - 양방향 디지털 서명을 이용한 인증

시도 – 응답 인증

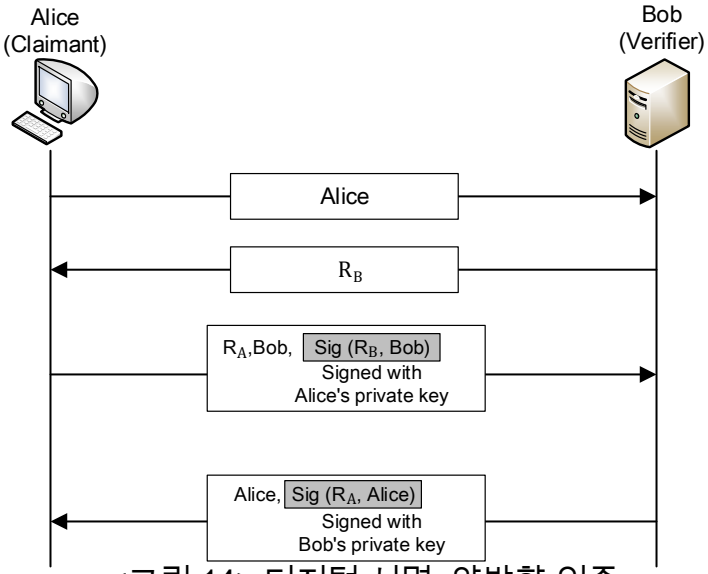
- 디지털 서명을 이용한 인증
  - 일방향 디지털 서명을 이용한 인증
    - 주장자가 시도값에 자신의 개인 키로 서명하여 개인 키 소유를 증명하는 방식
    - 작동 방식
      1. 주장자가 검증자에게 자신의 식별자를 보내 인증 시작을 요청함
      2. 검증자는 주장자에게 평문 시도를 전송함
      3. 주장자는 자신의 개인 키로 서명한 값을 주장자에게 보냄
      4. 검증자는 주장자의 공개 키로 서명을 검증함



<그림 13> 디지털 서명, 일방향 인증

시도 – 응답 인증

- 디지털 서명을 이용한 인증
  - 양방향 디지털 서명을 이용한 인증
    - 주장자와 검증자가 각각 상대의 시도값에 자신의 개인 키로 서명하여 서로를 인증하는 방식
    - 작동 방식
      1. 주장자가 검증자에게 자신의 식별자를 보내 인증 시작을 요청함
      2. 검증자는 주장자에게 평문 시도를 보냄
      3. 주장자가 검증자의 시도에 대한 응답과 자신의 새로운 시도를 검증자에게 보냄
      4. 검증자는 자신의 개인 키로 주장자의 시도에 응답함



<그림 14> 디지털 서명, 양방향 인증

시도 – 응답 인증

- 정리

[표 3] 시도-응답 인증 정리

구분	대칭-키 암호	키-해시함수(HMAC)	비대칭-키 암호	디지털 서명
시도 생성	검증자와 주장자가 공유하고 있는 비밀키로 암호화	평문 시도와 HMAC을 사용한 시도를 사용	검증자가 주장자의 공개 키로 암호화	평문 시도를 사용
사용하는 키	공유 비밀 키 K_{A-B}	공유 비밀 키 K_{A-B}	공개 키/개인 키 쌍	공개 키/개인 키 쌍
응답 생성	시도를 비밀 키로 암호화	시도에 키-해시(HMAC)를 적용	시도를 개인 키로 복호화	시도에 개인 키로 서명
검증 방법	검증자가 같은 키로 복호화해 비교	검증자가 같은 키로 MAC을 계산해 수신값과 비교	검증자가 공개 키로 확인	검증자가 공개 키로 확인

목 차

- 개요
- 패스워드 기반 인증(Password-based authentication)
- 시도-응답 인증(Challenge-Response authentication)
- 영지식 인증(Zero-knowledge authentication)
- 생체인식(Biometrics)

영지식 인증

- 정의
  - 주장자가 자신의 비밀 정보를 공개하지 않고 검증자에게 본인이 비밀 정보를 알고 있음을 증명하는 인증 방식
- 특징
  - 주장자는 검증자가 자신의 비밀을 포함해서 보내준 시도에 함수를 적용함
  - 메시지를 교환한 뒤 검증자는 예, 아니오 둘 중 한 개의 비트정보만 알 수 있음
- 종류
  - Fiat-Shamir 프로토콜
  - Feige-Fiat-Shamir 프로토콜
  - Guillou-Quisquater 프로토콜

영지식 인증

- Fiat-Shamir 프로토콜
  - 합성수에 대해서 제곱근을 구하는 것이 난해함에 기초한 프로토콜
    - $v = s^2 mod\ n$에서 v를 알더라도 s를 찾는 것이 어려움
  - 키 생성
    - 신뢰받는 제 3자는 두 개의 큰 소수 p, q를 선택하여 $n = p \times q$를 계산함
      - n은 공개, p, q는 비공개
    - 주장자는 비밀값 s를 $1 < s < n - 1$에서 선택하여 $v = s^2 mod\ n$을 계산함
    - 주장자는 s를 자신의 개인 키로 소유하고 v를 공개 키로써 제 3자에게 등록함

영지식 인증

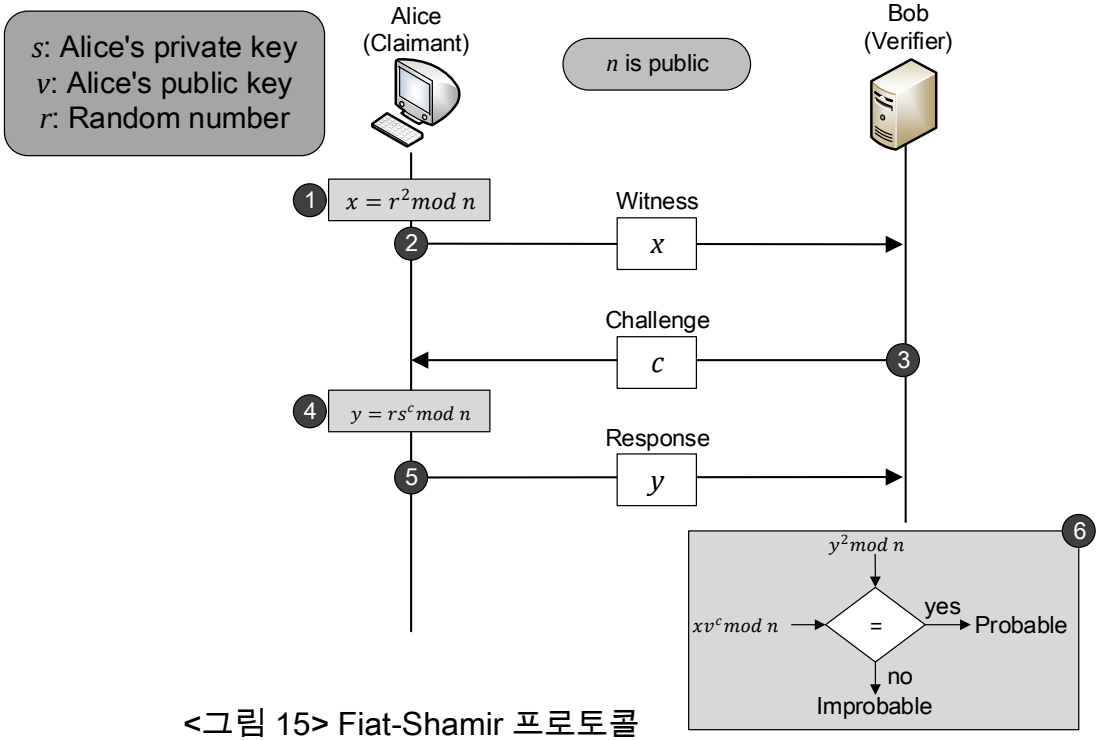
- Fiat-Shamir 프로토콜
  - 키 생성 예시

[표 4] Fiat-Shamir 프로토콜 키 생성 계산 예시

구분	값
p (제 3자)	7
q (제 3자)	11
n (제 3자)	77
s (개인 키)	9
v (공개 키)	$s^2 mod\ n = 81\ mod\ 77 = 4$
공격자가 풀어야 할 문제	$x^2 mod\ 77 = 4$를 만족하는 x

영지식 인증

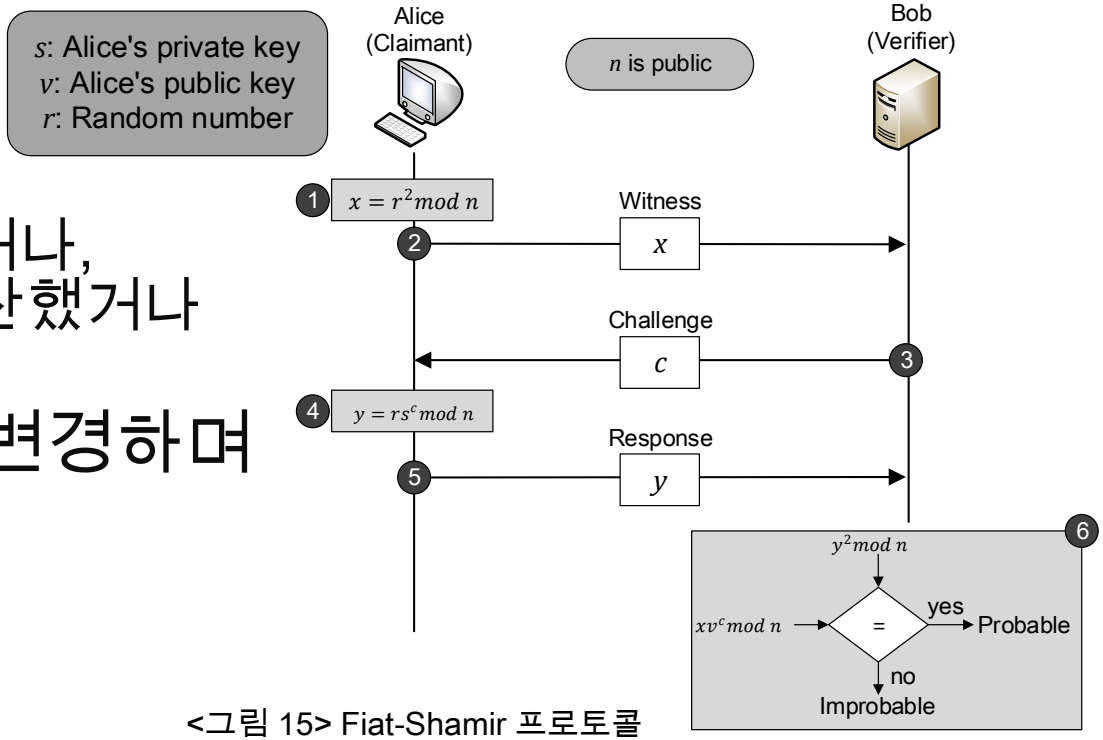
- Fiat-Shamir 프로토콜
  - 작동 방식 (1/2)
    1. 주장자는 0과 $n-1$ 사이의 임의의 수 r을 한 개 선택하여 $x = r^2 mod\ n$을 계산함
       - r은 공약(Commitment)이라고 함
    2. 주장자는 x를 증거로 검증자에게 보냄
    3. 검증자는 시도 c를 주장자에게 보냄
       - c는 0이나 1임



<그림 15> Fiat-Shamir 프로토콜

영지식 인증

- Fiat-Shamir 프로토콜
  - 작동 방식 (2/2)
    4. 주장자는 응답 $y = rs^c mod\ n$을 계산함
    5. 주장자는 응답 y를 검증자에게 보내 자신이 주장자의 개인 키인 s를 알고 있다는 것을 보임
    6. 검증자는 y^2과 xv^c를 계산함
       - 만약 $y^2 \equiv xv^c$이라면 주장자가 s값을 알고 있거나, y값을 다른 방법으로 계산했거나 둘 중 하나임
    7. 1~6 과정에 대해 C를 변경하며 반복함



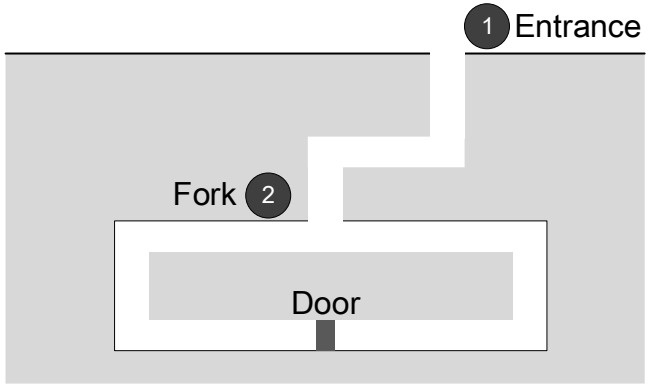
<그림 15> Fiat-Shamir 프로토콜

영지식 인증

- Fiat-Shamir 프로토콜
  - 특징
    - c값을 0과 1중에서 임의로 선택해가며 1~6단계를 반복함
      - 1~6단계를 한 라운드라고 표현함
    - 주장자는 자신의 인증을 위해서 모든 라운드를 통과해야 함
      - 한 번이라도 실패하면 작동은 멈추고 인증에 실패함
    - 공격자는 시도 값을 추측해서 몇 번의 라운드를 통과할 수 있기 때문에, 라운드 횟수를 늘려 공격을 방지함
      - 한 라운드를 추측으로 성공할 확률은 $(\frac{1}{2})$임
      - 20 라운드를 추측으로 모두 성공할 확률은 $\left(\frac{1}{2}\right)^{20}$임

영지식 인증

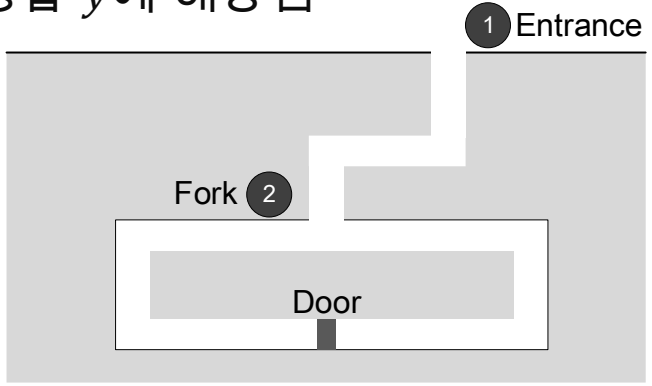
- Fiat-Shamir 프로토콜
  - 동굴 예
    - 개요
      - Fiat-Shamir 프로토콜을 설명하는 예시
    - 작동 방식
      1. 동굴의 끝에 문이 있고 문은 매직워드(개인 키)로만 열린다고 가정함
      2. Alice(주장자)는 자신이 매직워드를 알고 있기 때문에 문을 열 수 있다고 주장함
      3. Bob(검증자)와 Alice는 굴의 입구 지점인 ①에서 대기함
      4. Alice가 굴로 들어가고 갈래길 지점인 ②에 도착함



<그림 16> 동굴 예

영지식 인증

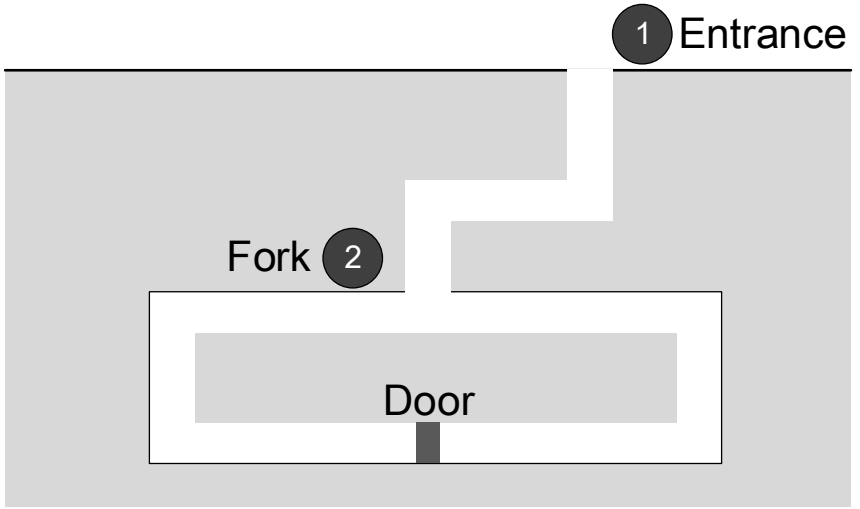
- Fiat-Shamir 프로토콜
  - 동굴 예
    - 작동 방식
      5. Alice는 왼쪽이나 오른쪽을 선택할 수 있음
         - 이것은 증거 x를 보내는 것에 대응됨
      6. Alice가 굴 속으로 들어간 이후, Bob은 갈래 길까지 온 다음 Alice에게 오른쪽이나 왼쪽 중 한 쪽으로 나오라고 요청함
         - 이것은 시도 c를 보내는 것에 대응됨
      7. 만약 Alice가 매직워드를 안다면 Bob이 요청한 쪽으로 나올 수 있음, 혹은 매직워드를 모르더라도 Bob이 요청한 쪽으로 나올 수 있음
         - 이것은 응답 y에 해당됨



<그림 16> 동굴 예

영지식 인증

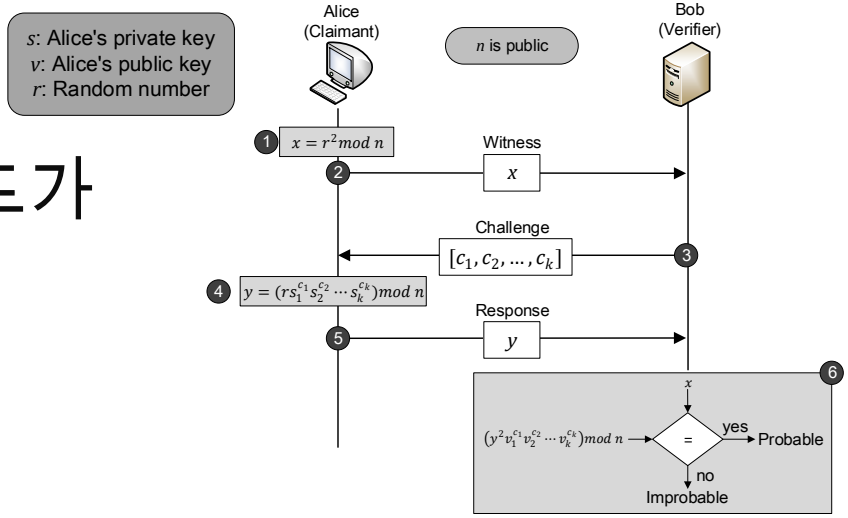
- Fiat-Shamir 프로토콜
  - 동굴 예
    - 작동 방식
      9. 이런 게임을 반복하여 여러 차례(N회) 한다고 가정하고, Alice는 모든 게임에서 성공할 경우에 이긴다고 가정함
      10. N이 늘어날수록 Alice가 매직워드를 모르면서 게임을 이길 확률은 0에 가까워짐



<그림 16> 동굴 예

영지식 인증

- Feige-Fiat-Shamir 프로토콜
  - Fiat-Shamir 프로토콜에서 비밀 값을 s 단일이 아닌 여러 개의 비밀 값을 이용하여 보안을 높인 프로토콜
  - 작동 방식
    - Fiat-Shamir 프로토콜과 동일하게 작동하나 비밀 값을 여러 개 사용함
      - 개인 키로는 $s_1, s_2, \dots, s_k$를 사용함
      - 공개 키로는 $v_1, v_2, \dots, v_k$를 사용함
      - 시도 값들로는 $c_1, c_2, \dots, c_k$를 사용함
    - 3번의 교환을 하면 한 번의 라운드가 끝나고, 주장자가 한 번이라도 실패하면 인증을 받지 못 함



<그림 17> Feige-Fiat-Shamir 프로토콜

영지식 인증

- Guillou-Quisquater 프로토콜
  - Fiat-shamir 프로토콜에서 시도의 범위를 넓혀 라운드 수를 줄인 프로토콜
  - 키 생성
    - 신뢰받는 제 3자는 다음과 같은 수를 선택함
      - $n = p \times q$를 계산할 두 개의 소수 p와 q
      - $\phi = (p-1)(q-1)$, ϕ와 서로소인 e
      - 지수로 사용할 c
      - n과 e는 공개, p와 q는 비밀로 유지함
    - 제 3자는 v와 s를 각 개체에게 부여함
      - v는 공개 값, s는 비밀 값임
        - $s^e \times v = 1 \, mod \, n$임
      - 항등관계 식은 다음과 같음

$$y^e \times v^c = (r \times s^c)^e \times v^c = r^e \times s^{ce} \times v^c = r^e \times (s^e \times v)^c = x \times 1^c = x$$

영지식 인증

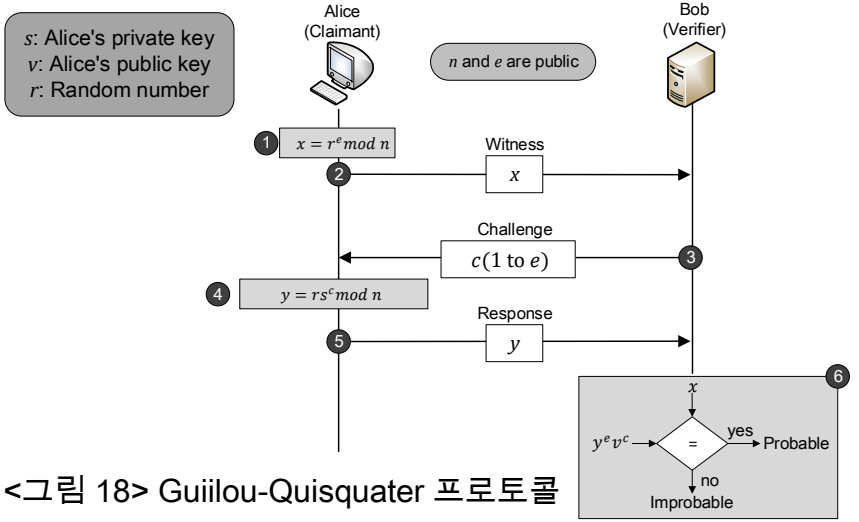
- Guillou-Quisquater 프로토콜
  - 키 생성 예시

[표 5] Guillou-Quisquater 프로토콜 키 생성 계산 예시

구분	값
p (제 3자)	7
q (제 3자)	11
n (제 3자)	77
e (공개 지수)	7
s (비밀 키)	5 $(1 < s < 77)$
v (공개 키)	$s^e mod\ n = 5^7\ mod\ 77 = 47$
공격자가 풀어야 할 문제	$s^7 \equiv 47\ mod\ 77$을 만족하는 s

영지식 인증

- Guillou-Quisquater 프로토콜
  - 작동 방식
    1. 세 번의 교환을 통해 한 번의 라운드가 끝남
    2. 검증은 임의의 수 c(시도)를 1과 e사이에서 추출하여 라운드를 여러 차례 반복함
    3. 주장자는 모든 라운드에 각기 다른 c에 대한 응답인 y를 보내 성공적으로 통과해야만 인증을 받을 수 있음
    4. 만약 한 라운드라도 실패하면 주장자는 인증에 실패함



<그림 18> Guiilou-Quisquater 프로토콜

영지식 인증

- 정리

<표 6> 영지식 인증 정리

구분	Fiat-Shamir	Feige-Fiat-Shamir	Guillou-Quisquater
기반 문제	합성수의 제곱근 구하기 (인수분해)	합성수의 제곱근 구하기	인수분해와 관련된 RSA 문제
개인 키	1개	k개	1개
개선점	-	비밀을 늘려 라운드 수를 절감	비밀을 1개로 유지하여 라운드와 저장 공간을 절감
한계점	라운드를 많이 반복해야 함	비밀을 여러 개 저장해야 함	라운드마다 e의 제곱연산을 해야함
활용 사례	스마트카드 인증	스마트카드 인증	보안칩 인증

목 차

- 개요
- 패스워드 기반 인증(Password-based authentication)
- 시도-응답 인증(Challenge-Response authentication)
- 영지식 인증(Zero-knowledge authentication)
- 생체인식(Biometrics)

생체 인식

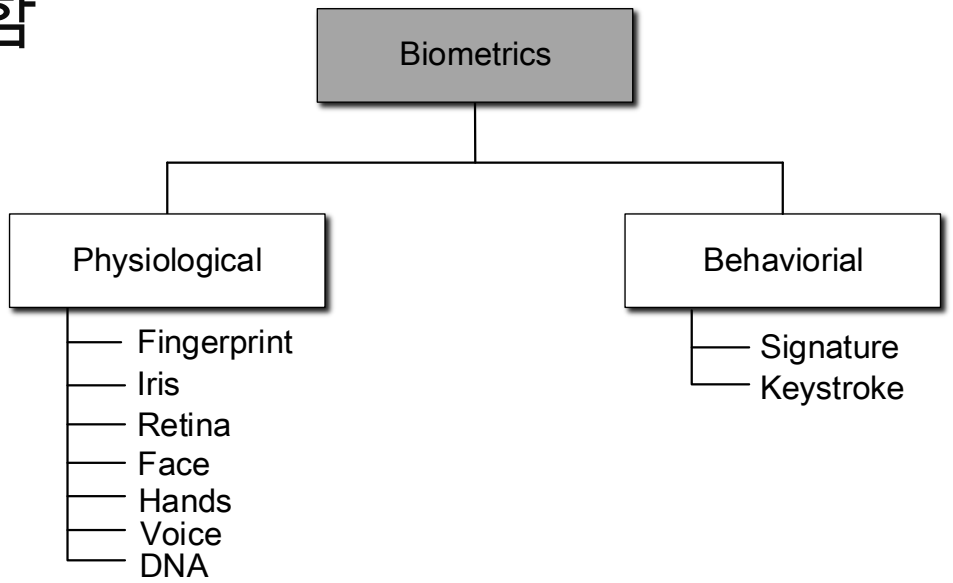
- 정의
  - 생체적 혹은 행동적 특성을 측정하여 신원을 인증하는 방법
- 구성 요소
  - 인식장치
    - 생체정보를 측정하기 위한 장치
    - e.g., 스마트폰 지문 인식기, 얼굴 인식 카메라 모듈 등
  - 프로세서
    - 측정된 특성을 저장할 수 있는 형태의 자료로 변경하는 장치
    - e.g., 슈프리마의 DSP (Digital Signal Processor) 지문 인식 알고리즘 등
  - 저장장치
    - 인증처리 결과를 저장할 때 사용하는 장치
    - e.g., 보안 저장 영역(TEE, Trusted Execution Environment)등

생체 인식

- 등록
  - 생체인식 기술을 사용하게 될 주장자들의 특성을 데이터베이스로 구축하는 절차
- 인증
  - 검증이나 확인을 통해 주장자가 자신의 신원을 인증 받음
  - 검증
    - 주장자가 자신이 올바른 주장자라고 하는 것을 확인하는 것
  - 확인
    - 주장자가 데이터베이스에 저장되어 있는지 확인하는 것

생체 인식

- 기술
  - 생리적 기술 (Physiological)
    - 사람 몸의 신체적인 특성을 측정해서 인증이나 확인을 하는 수단
      - 특성은 사람들을 구별할 수 있도록 사람마다 고유한 성질을 가져야만 함
    - 나이가 들거나 수술, 질병, 건강 등의 조건에 따라 변경이 가능해야 함



<그림 19> 생체 인식 종류

생체 인식

- 기술
  - 생리적 기술
    - 손금 (Fingerprint)
      - 미뉴셔 기반(Minutiae-based) 측정
        - 개별적인 손금이 시작하고 끝나는 곳 혹은 갈라지는 곳에 근거하여 그래프를 그림
      - 이미지 기반 측정
        - 손가락 끝의 이미지를 생성하고 데이터베이스에 저장된 이미지와 유사성을 찾아냄
    - 홍채 (Iris)
      - 레이저 광선(적외선)을 사용하여 사람마다 고유한 홍채 패턴을 측정함
      - 잘 변하지 않지만, 안구 질병에 의해 패턴이 변할 수도 있음
    - 망막 (Retina)
      - 검은 눈동자 부분의 혈관을 관찰함
      - 가격이 비싸 보급이 많이 되지 않음

생체 인식

- 기술
  - 생리적 기술
    - 얼굴 (Face)
      - 얼굴을 구성하는 코, 입, 눈 사이의 거리에 기반하여 얼굴의 형상을 분석함
      - 피부의 특성까지 얼굴과 연계하기도 함
    - 손 (Hands)
      - 손의 모양과 손가락의 길이 같이 손의 각 부분에 대한 길이를 측정함
    - 목소리 (Voice)
      - 목소리의 높이, 억양과 톤을 측정함
      - 지협적으로도 측정 가능하고 원격으로도 측정 가능함
      - 배경 잡음, 질병, 나이 등에 따라 영향을 받음

생체 인식

- 기술
  - 생리적 기술
    - DNA (DeoxyriboNucleic Acid)
      - 인간의 모들 셀과 기타 기관속에 있는 염색체에서 발견되는 화학물질인 DNA를 검증에 사용함
      - 일생동안 변하지 않으며 죽은 다음에도 사용할 수 있음
      - 일란성 쌍둥이는 DNA가 같을 수도 있다는 문제점이 있음

생체 인식

- 기술
  - 행동적 기술 (Behaviorial)
    - 인간행동의 특성을 측정하여 인증이나 확인을 하는 수단
    - 서명 (Signature)
      - 서명 태블릿과 특수 펜을 이용하여 같은 사람이 한 서명이 맞는지를 측정함
      - 최종적으로 나온 서명도 확인하지만 서명에 걸리는 시간 같은 행동적 특성도 측정함
    - 키 스트로크(Typing Rhythm)
      - 자판에서 작업하는 것과 관련된 사람의 행동을 측정함
      - 키를 누르고 있는 시간, 키와 키 사이의 간격, 오타의 빈도, 키를 누르는 압력 등을 측정함
      - 시간이 지나면서 변할 수 있는 특성들이기에 정확도가 떨어짐

생체 인식

- 정확도
  - 부정 거부율(FRR, False Rejection Rate)
    - 사전에 등록을 완료하여 인식되어져야 할 사람이 얼마나 자주 시스템에 의해 인식이 되지 않는지를 나타내는 값
      - 등록된 본인인데 거부당하는 경우
  - 부정 허용율(FAR, False Acceptance Rate)
    - 사전에 등록이 되지 않아 인식되어서는 안 될 사람이 얼마나 자주 시스템에 의해 인식이 되는지를 나타내는 값
      - 등록되지 않은 사람이 통과되는 경우
- 응용
  - 생체인식은 시설물 출입, 정보시스템에 대한 접근, 국경 출입국 통제시스템 등 다양한 방면에서 사용되고 있음

Thanks!

민 윤 홍(yunhong@pel.sejong.ac.kr)