6-1 : 과제 피드백 및 5주차 내용 토론

들어가기 전에

이번 시간에는…

과제 피드백 & QnA

JWT
개인정보 암호화
OAuth2
스프링 시큐리티

과제 피드백

인증, 인가 구현(JWT, OAuth2 활용)

토론 키워드

JWT
AccessToken, Refresh Token

JWT

JWT 인증 과정

1.클라이언트 로그인 요청

2. 로그인 성공 시, 유저 정보 및 유효기간 등을 Payload에 담고, 비밀키를 사용해 Access Token(JWT)을 발급

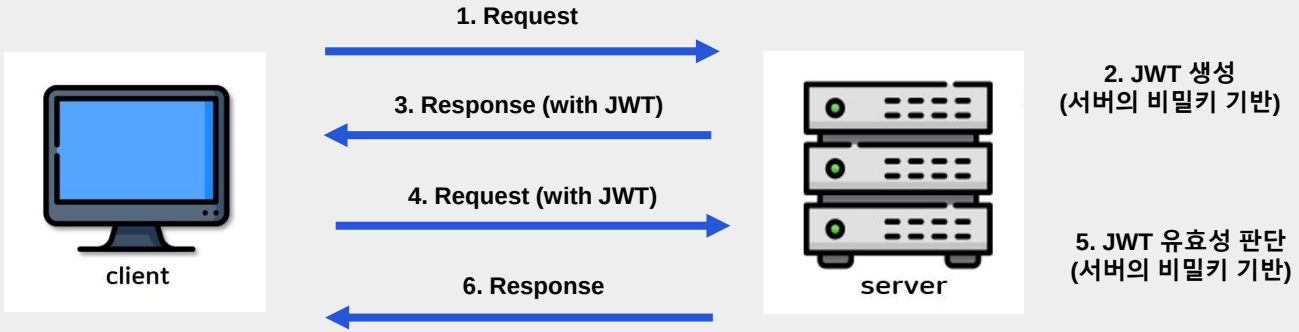
3. 서버는 생성한 JWT 토큰을 클라이언트에게 전달

4. 클라이언트는 전달받은 토큰을 저장해두고, 요청할 때 마다 토큰을 요청 헤더 Authorization에 포함시켜 함께 전달

5. 서버는 받은 JWT의 헤더, 페이로드에 비밀키를 더해 암호화

5. 암호화 값이 JWT의 Signature와 일치하는지 판단 후, 유효 기간 등을 확인하여 유효한 토큰인지 확인

6. 유효한 토큰이라면 요청에 응답



JWT

JWT 장단점

장점

- Header와 Payload를 가지고 Signature를 만들기 때문에 데이터 위변조를 막을 수 있음
- 인증 정보에 대한 별도의 저장소가 필요 없음
- 토큰 기반으로 다른 로그인 시스템에 접근 및 권한 공유가 가능
- 모바일 환경에서도 정상적으로 동작

(쿠키, 세션은 브라우저 내에서만 사용 가능. 모바일은 별도 구현 필요)

단점

- 쿠키, 세션에 비해 데이터의 길이가 길어 네트워크 부하 가중
- Payload 자체는 암호화가 되지 않기 때문에 중요한 정보를 담을 수 없음
- 현재 로그인 중인지 여부를 알 수 없음
- 토큰을 탈취당하면 대처하기 어려움

(서버 측에서 대응할 방법이 없음)

JWT

JWT 단점 보완

Refresh Token을 추가로 두어 JWT 단점을 보완

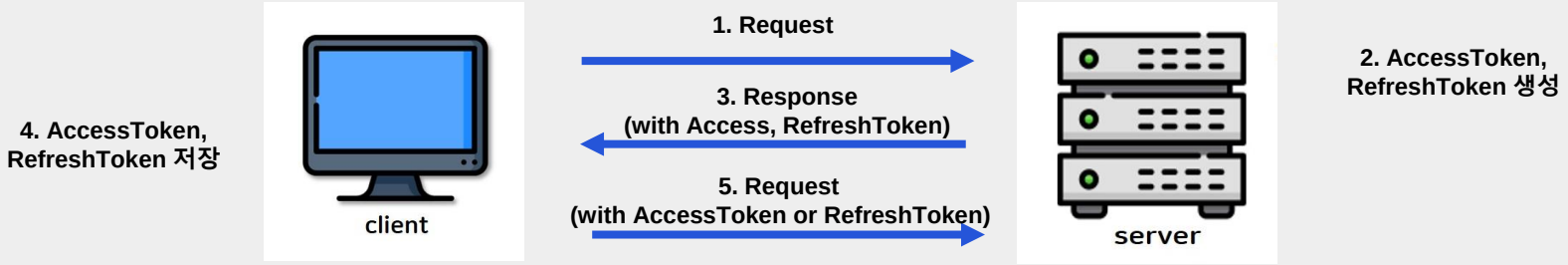
1.클라이언트 로그인 요청

2. 로그인 성공 시, AccessToken(짧은 유효 기간)과 RefreshToken(긴 유효 기간)을 발급

3. 서버는 생성한 AccessToken, RefreshToken을 클라이언트에게 전달

4. 클라이언트는 AccessToken을 private 변수, RefreshToken을 쿠키(httpOnly, secure 보안 강화 옵션 추가)에 저장

5. 클라이언트는 AccessToken으로 인가를 진행, 토큰 만료 시 RefreshToken으로 재발급 요청



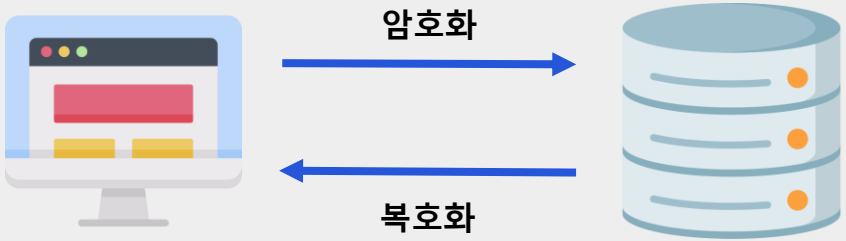
토론 키워드

개인정보 암호화

암호화 알고리즘

암호화 알고리즘

우리는 외부에 노출되면 안되는 정보들을 DB에 저장할 때
암호화 알고리즘을 사용하고 있다.



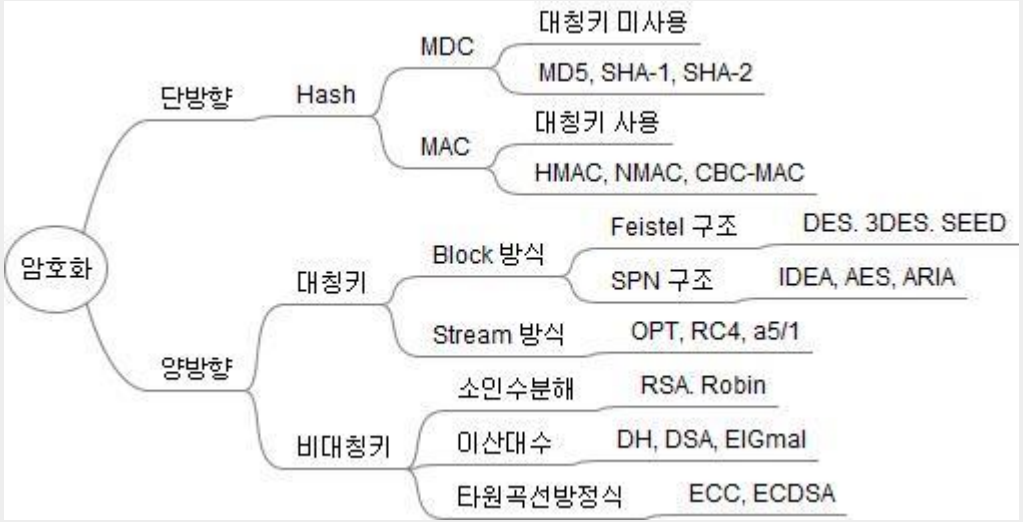
암호화 알고리즘

암호화 알고리즘

암호화에는 크게

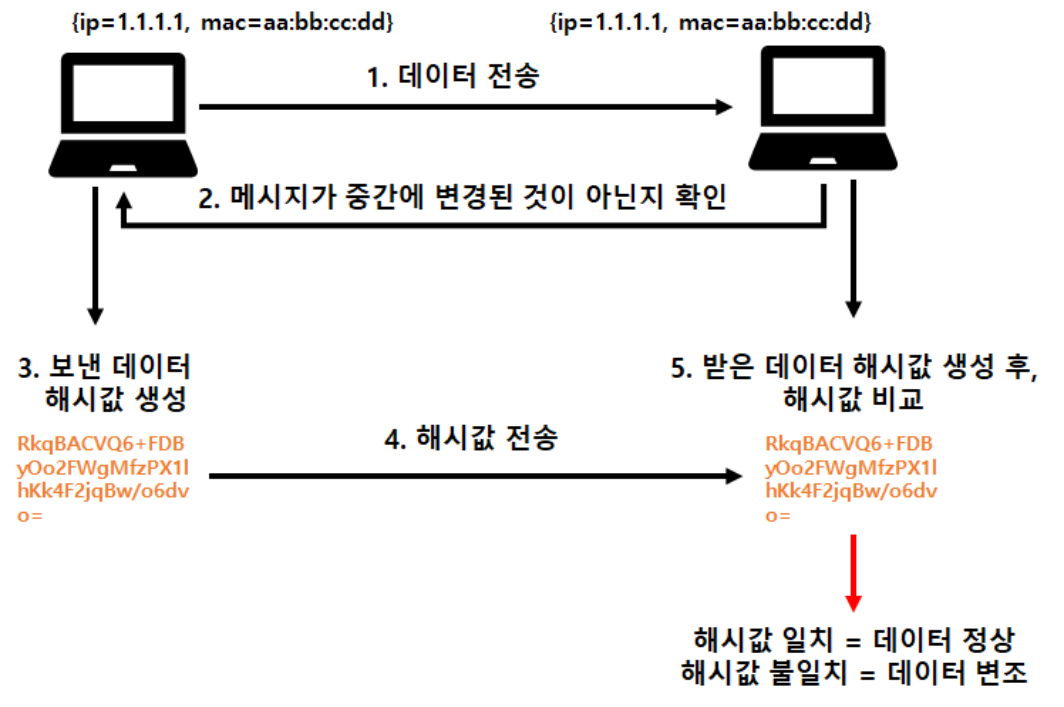
복호화가 불가능한 단방향 암호화

복호화가 가능한 양방향 암호화가 있다.



암호화 알고리즘

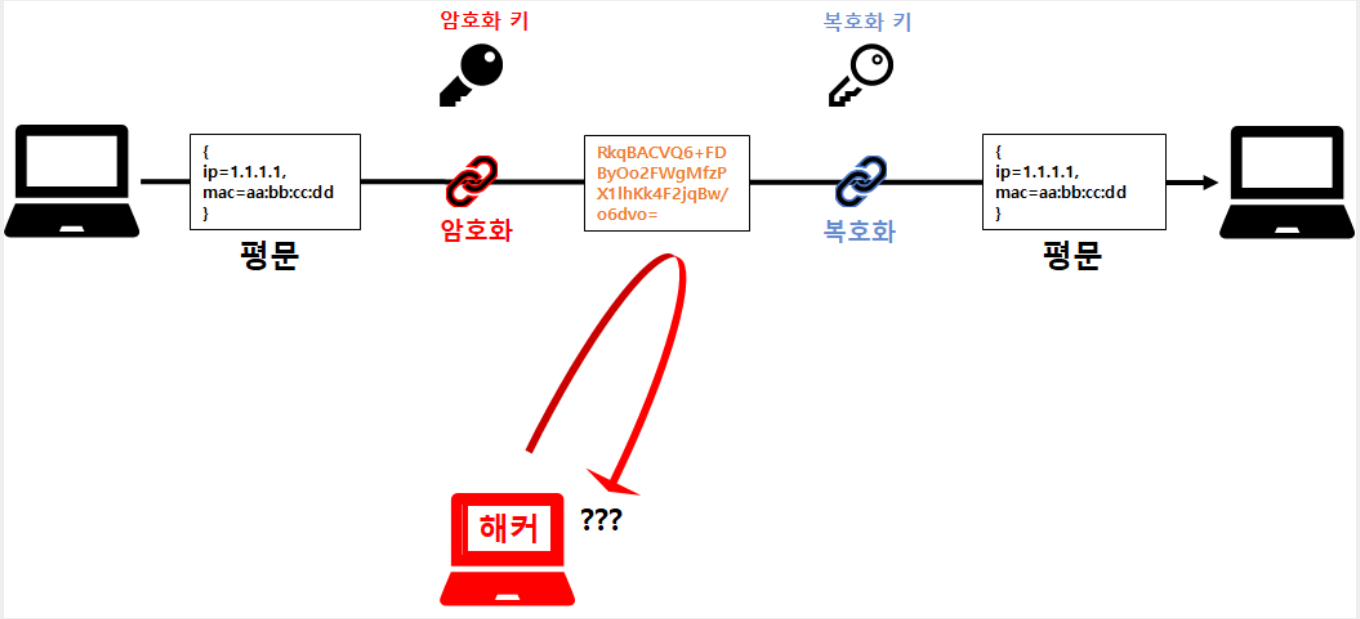
단방향 암호화 과정



암호화 알고리즘

양방향 암호화 과정

키가 같으면 대칭키, 다르면 비대칭키

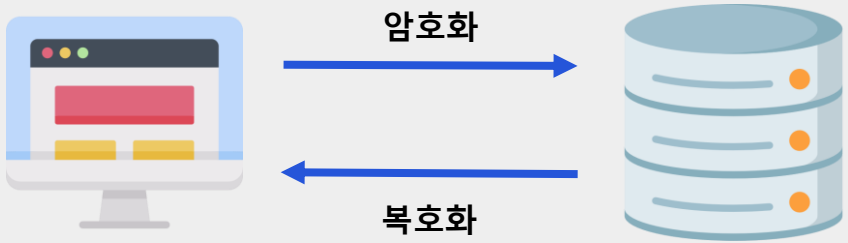


암호화 알고리즘

암호화 알고리즘

민감한 정보, 데이터 조회가 필요한 경우: 양방향 알고리즘 ex) 카드번호

민감한 정보, 데이터 조회가 불필요한 경우: 단방향 알고리즘 ex) 패스워드



토론 키워드

JWT 서명
대칭키 vs 비대칭키 방식

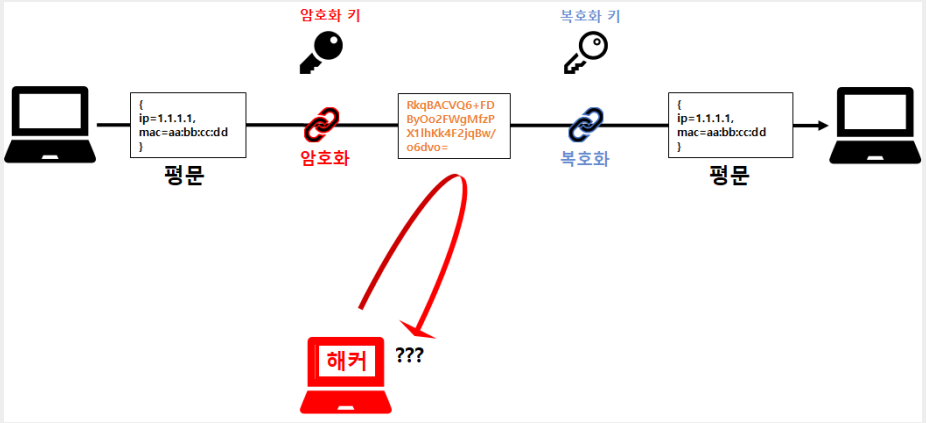
JWT

HS256 vs RS256

JWT 서명 방식에는 크게 대칭키 방식과 비대칭키 방식이 있다.

HS256은 단방향 암호화 방식 중에서도 대칭키 사용 방식

RS256은 양방향 암호화 방식 중에서도 비대칭키 사용 방식



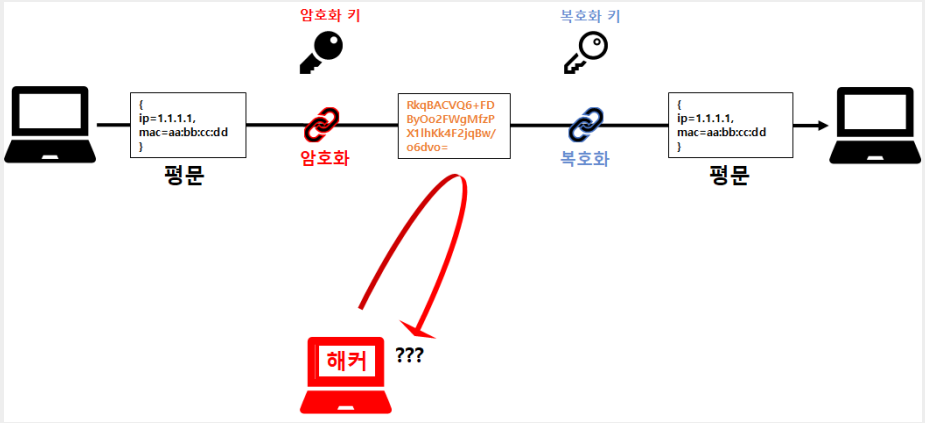
JWT

JWT HS256 방식 특징

하나의 Key 값을 통해 암호화와 복호화

장점: 속도가 비대칭키 방식에 비해 빠르다. 비교적 구현이 쉽다.

단점: Key가 하나이기 때문에 상대적으로 보안에 취약하고, 관리가 어렵다.



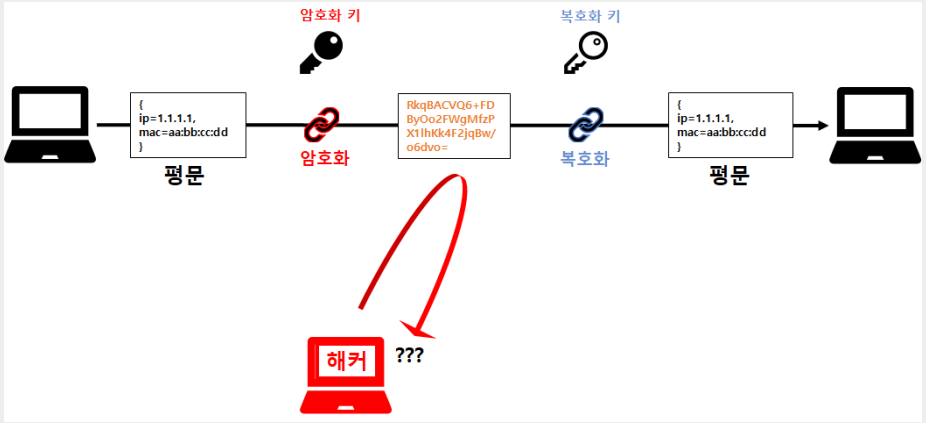
JWT

JWT RS256 방식 특징

공개키, 개인키 2개의 Key 값을 통해 암호화와 복호화

장점: 공개키를 통해 여러 대상이 토큰 생성 가능. 상대적으로 보안에 강함.

단점: 속도가 대칭키에 비해 느리고, 구현이 비교적 어렵다



토론 키워드

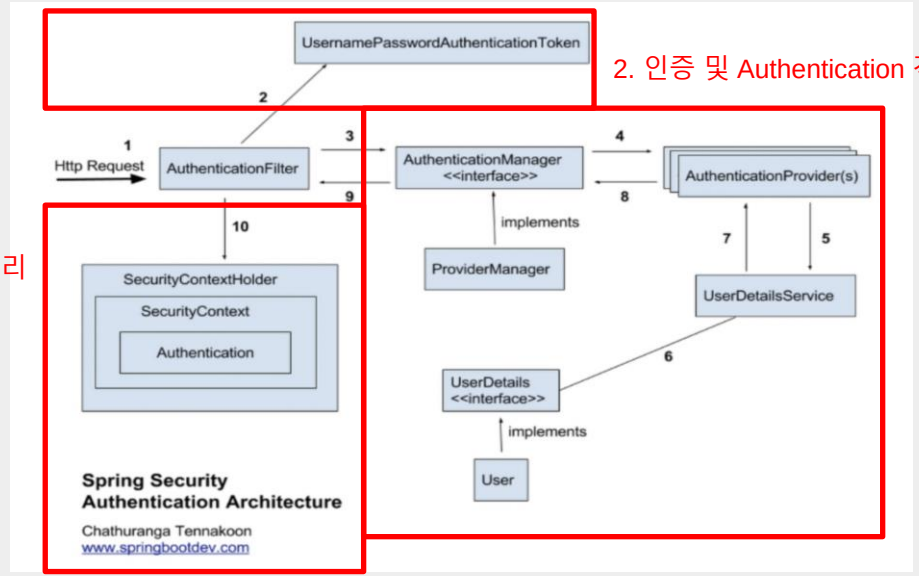
스프링 시큐리티

스프링 시큐리티

1. Authentication 객체 생성

2. 인증 및 Authentication 객체 채우기

3. Authentication 객체 저장, 관리



1. 사용자 인증 요청
2. AuthenticationFilter가 Authentication 객체를 생성하고, 요청 정보를 AuthenticationManger에게 전달
3. AuthenticationManager는 AuthenticationProvider들을 조회하며 인증을 요구(커스텀 필터 등 조회)
4. AuthenticationProvider는 UserDetailsService를 통해 요청 정보를 기반으로 사용자 정보를 DB에서 조회 후 Authentication 반환
5. AuthenticationManager는 반환된 Authentication 객체를 AuthenticationFilter로 전달
6. AuthenticationFilter는 Authentication 객체를 SecurityContextHolder에 저장

Hello World!