4.16

암호화.

해쉬 함수.

- 단방향 암호화 방식
- 데이터 무결성 체크를 위해 탄생.

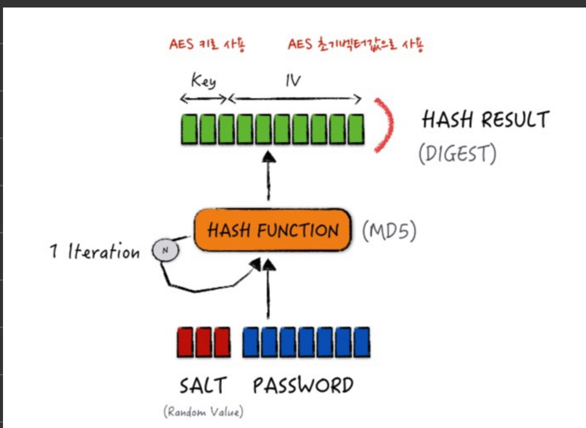
→ 하나의 평문에는 언제나 같은 해쉬 함수 만들어짐

장점: 복호화가 어려움 / 비용↓ (vs RSA)

단점: 모든 평문의 해쉬 함수 기록 → 레인보우 테이블 ← 공격: 레인보우 테이블 어택

↓

Salting (음식은 만들기 전 소금을 뿌리자!)



해쉬함수 만들기 전에 평문 값에 SALT라는 랜덤값을 추가로 넣어서 해쉬함수를 만들겠다.

방법: ① PBKDF2

Salting 후 해쉬함수 생성 → 이걸 또다시 해쉬함수화

⇒ 해쉬함수가 몇번 사용?, Salting된 함수로 알수 X

② BCrypt

웹스토리지

서버가 아닌 클라이언트에 데이터를 저장할 수 있도록 지원하는 HTML5의 새로운 기능.

- 키-밸류 스토리지의 형태

종류: Local Storage vs Session Storage

영구성 브라우저 닫으면 지움.

자동로그인 일회성 로그인 정보

└ 패스워드 X

두 스토리지는 Window 객체 안에 들어있다.

① window.localStorage; '키' : value

value: 문자열 / 불린 / 숫자 / null / undefined 저장 → return : key: 문자열 / value : 문자열

localStorage.setItem(key, 'value') : 로컬스토리지에 저장. → [object 생성자] 형식으로 저장

localStorage.setItem(key) : 키 값으로 조회.

localStorage.removeItem(key): 키값으로 지우기

localStorage.clear() : 스토리지 전체 지우기

↓

JSON.stringify({a, 'b'}) : 객체 한번에 저장

받을때는 JSON.parse 사용.

② window.sessionStorage

브라우저 닫는 동안 → 영구적 보관 X

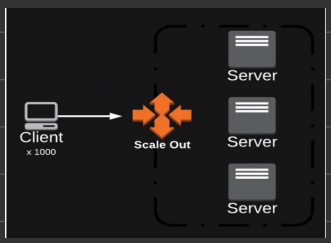
각 세션마다 데이터가 개별적으로 저장

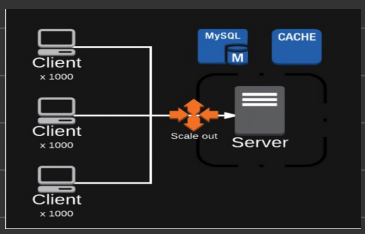
-단점: 서버확장시 세션정보의 동기화 문제

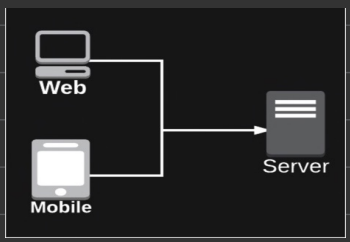
서버/세션 저장소의 부하

웹/앱존의 상이한 쿠키-세션 로직

→ 세션의 한계점.







IndexedDB

JS가 이해하는 어떠한 값이라도 모두저장 가능

(Storage는 String만 가능)

ID 값이 포함된 JSON 객체가 모여 있는 트랜잭션별 크롬 데이터베이스를 위해 W3C 권고 표준 인터페이스

key-value 조합 / B-tree 데이터구조.

Transaction 내 원자 보장 → 모든 변경사항 commit, 이전 상태로 롤아웃

① DB

Version + Object Store

브라우저당 여러개의 DB 가질수있다.

indexedDB.open 로 열 수 있다.

② Object Store

데이터 담는곳

Object Store 이름은 고유해야한다.

IDBRequest.createObjectStore 로 만들 수있다.

key path 설정하면 내부키 사용 / Default는 외부키

③ Transaction

IDBRequest.transaction 으로만듦

트랜잭션은 readwrite, readonly, versionchange 의 상태를 가질 수 있다.

IndexedDB의 API 작업은 트랜잭션 context에서 발생

실패 → 모든 변경사항 이전 상태로 Back

④ Cursor

DB의 Iterator

ObjectStore.openCursor에 key 혹은 keyrange(IDBKeyRange) 쿼리를 전달하고

IDBCursorWithValue를 전달받아 continue로 반복 조회

⑤ Index

Index는 Object를 참조하는 ObjectStore 이다.

IDBObjectStore.createIndex 로 생성

자동 업데이트가 된다

JWT

Stateless

토큰자체로 정보를 가지고 있음. → 별도의 인증서버가 필요X

요청받은 서버 자체에서 인증프로세스 수행 가능.

JSON format.

구조

① Header

typ : 토큰의 타입 / JWT만 쓰여

alg : 해싱 알고리즘 / 토큰 검증시 사용.

② Payload

실제 토큰으로 사용하려는 데이터가 담기는 부분

각 데이터를 Claim 이라고 하며 3개가 있음

1) Reserved Claims

이미 예약된 Claim, 필수X 사용권장 key: 3자리 String

iss(토큰발행자정보) / exp(Number) 만료일 / sub(String) : 제목 / aud (String) audience ...

2) Public Claims

사용자 정의 Claim / 공개정의 / URI 포맷으로 작성

3) Private Claims

사용자가 임의로 정한 정보.

③ Signature

Header와 Payload의 데이터 무결성과 변조 방지를 위한 서명

Header + payload 를 합친후 Secret+ key와 함께 Header의 해싱 알고리즘으로 인코딩

⇒ 이3개 구조를 base 64 인코딩 후 합친다.