[IT Essencial] BlockChain 1

> 블록체인의 기초 개념과 암호화 알고리즘

- 진행: 박종철 컨설턴트
- 날짜: 2020.09.09
- 목차

1. BlockChain

1. BlockChain 이란?

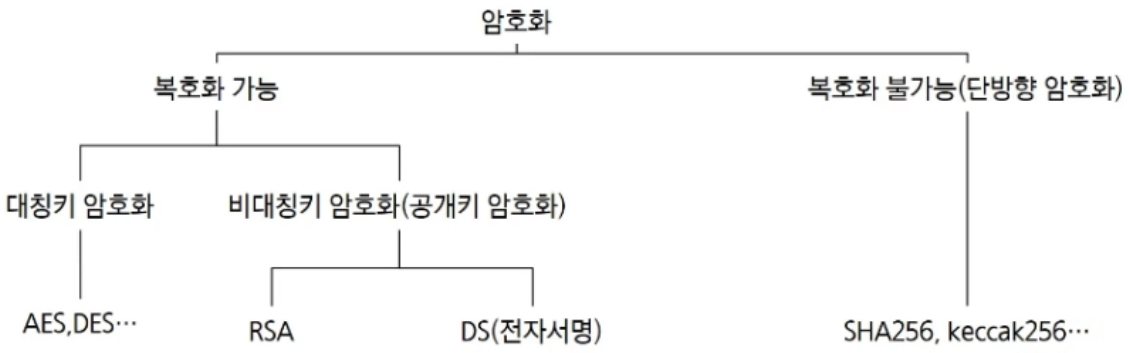
1. 블록과 체인의 합성어

   - Block
     - 관리 대상 데이터, 데이터의 논리적 저장 단위
     - 비트코인 - 거래내역을 저장, 이더리움 - 계약조건을 저장
   - Chain
     - Block 이 n 번 째 생성 될 때, n 번째 block 은 n-1 번째 block 을 보증한다.
     - Linked list 와 같은 형태로 엮였다고 하여 chain 이라 한다.
     - 보증한다의 의미: Chain 을 만들 때 Hash Function 을 사용하기 때문

2. 특징
   1. 블록과 체은을 통해 저장 된 데이터는 위변조가 거의 불가능에 수렴한다.
   2. 위변조 하면 안 되는 중요한 데이터를 투명하게(탈중앙화) 저장할 수 있다.

2. Cryptography

1. 암호화의 종류



2. 단방향 암호화

> 평문을 암호문으로 암호화 하는것은 가능하지만 암호문을 평문으로 복호화하는 것은 불가능한 암호화 방법

1. SHA256

- 어떤 길이의 값을 입력하더라도 256비트(64자리 문자열)의 고정 된 길이의 결과값을 만들어내는 알고리즘
- 결과가 같다면 입력으로 주어진 문장들이 같다는 의미

```
Hello SSafy!
-> BF21D9C9B5EE25E28C7EADA0F9C709FEC09016A8E0AC3F695530C1B1E205E4C0

Hello SSafy!
-> BF21D9C9B5EE25E28C7EADA0F9C709FEC09016A8E0AC3F695530C1B1E205E4C0
```

- 입력값이 조금이라도 달라지면 전혀 다른 결과값이 만들어진다.

```
Hello SSafy!
-> 67FD612D914172387FCCEA7B87013779A732D6ECA794299596E7691C588D4E5F
```

```
Q. 복호화가 불가능한데 암호문을 왜 사용하나요?
A. 검증을 할 때 사용합니다. 복호화가 된다는 의미는 해킹의 위험성이 있음을 말합니다.
```

2. Hash Function

- 임의의 길이의 데이터를 고정 된 길이의 데이터로 매핑하는 함수

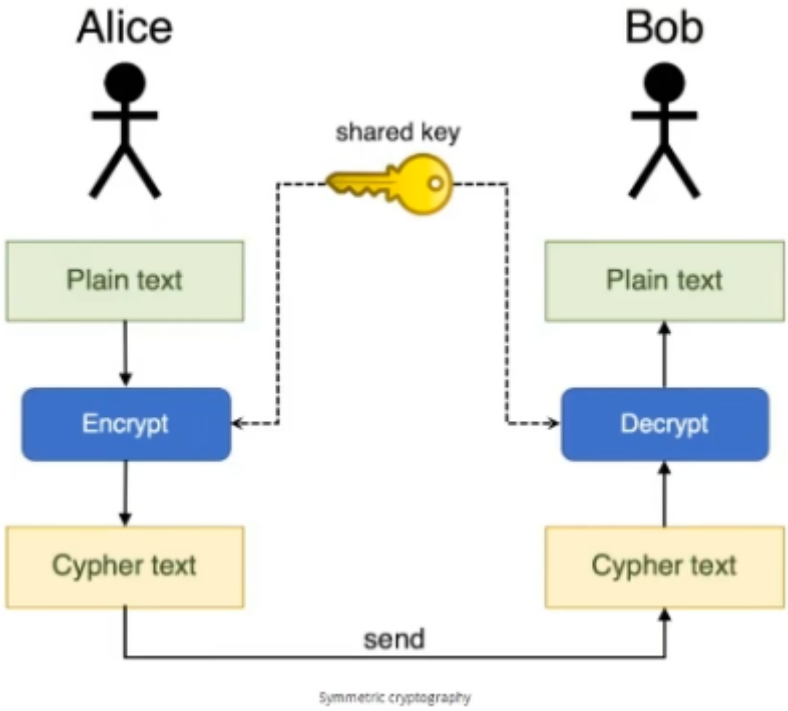
```
y = f(x)

y = 2x + 1 라는 함수가 주어지면 x 값에 따라 정해진 y 값을 구할 수 있다.

x = 1 ---> y = 3
x = 2 ---> y = 5

하지만, y 값을 안다고 해서 f(x) 를 구할 수 는 없다.
y = 7 ---> f(x)??
```

- 해시 값으로부터 원래의 입력값과의 관계를 찾기 어려운 성질을 가지는 경우를 의미한다.
- 해시 함수를 사용하게 되면 값의 위조, 변조가 일어나는 경우 원래의 입력값과 매핑 된 결과값이 달라진다.

4. 양방향 암호화

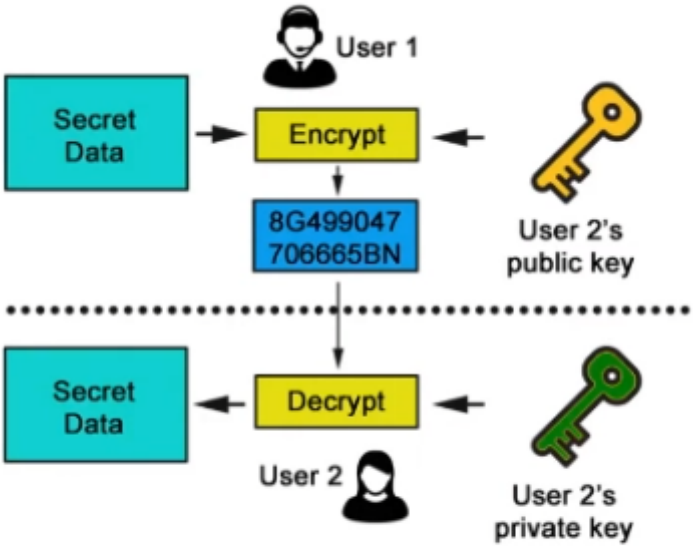
> 암호화 된 암호문을 평문으로 복호화할 수 있는 암호화

1. Symmetric Encryption (대칭키 암호화)



- 암호화 하는 키와 복호화 하는 키가 동일한 알고리즘
  - AES, ARIA, SEED 알고리즘
  - https://seed.kisa.or.kr/kisa/index.do 에 가면 무료로 사용 가능
- 취약점
  - 키 관리가 매우 중요하다.
  - 키 관리 서버를 따로 두거나, 주기적으로 변경해 주어야 한다.

2. Asymmetric Encryption (비대칭 키 암호화, 공개키 암호화)

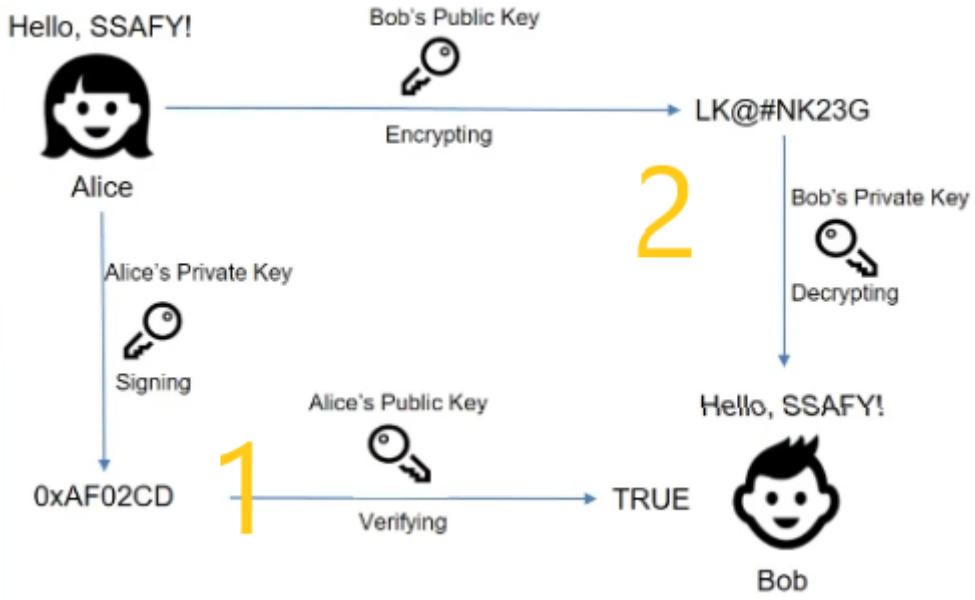


Asymmetric Encryption

- 암호화와 복호화를 위해 각각 별개로 한 쌍의 키를 두는 암호화
  - 암호화 하기 위한 public key (공개 키)
  - 복호화 하기 위한 private key (개인 키, 비밀 키)
  - 개인 키로 공개 키를 만들 수 있으나, 공개 키로 개인 키를 만들 수는 없다.

    즉, 복호화는 수신자 자신만 할 수 있다.
- RSA, ECC 알고리즘
- 취약점
  - Man in the middle attack (중간자 공격)
  - 암호화 된 내용이 누구로부터 온 것인지 알 수 없기 때문에, 타인을 사칭하여 정보를 탈취하는 공격 기법에 취약하다.

3. Digital Signature

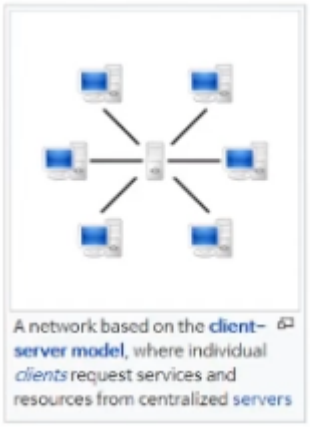


- 과정
  1. 송신자가 메시지를 보낼 때, 원문과 원문의 핵심 값을 자신의 개인키로 암호화 하여 보낸다.
  2. 수신자는 송신자의 서명을 확인하고, 송신자의 공개 된 키 페어를 통해 True/False를 확인한다. True 의 경우 복호화 과정을 진행한다.
  3. HASH 값이 나오면 원문과 HASH 값의 비교를 통해 데이터 검증을 한다. 그림에서 2번으로 표시 된 과정은 RSA 과정과 동일하다.
- 취약점
  - 느리다. 프로세스가 길다.

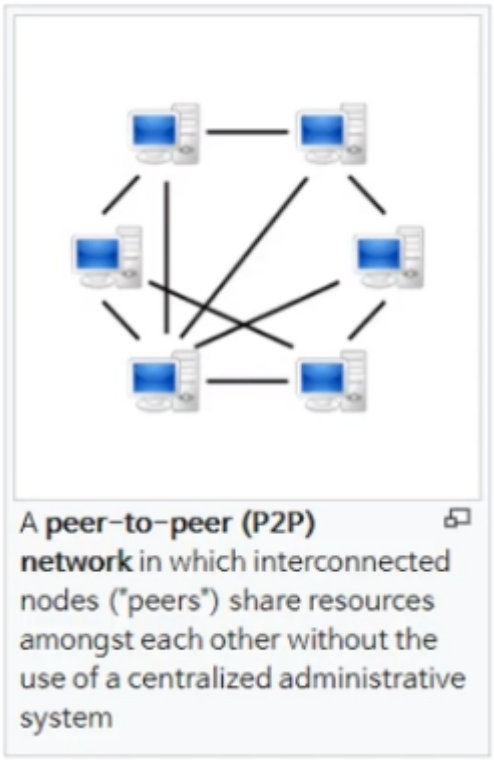
3. BlockChain Network

1. P2P

1. 작동 과정

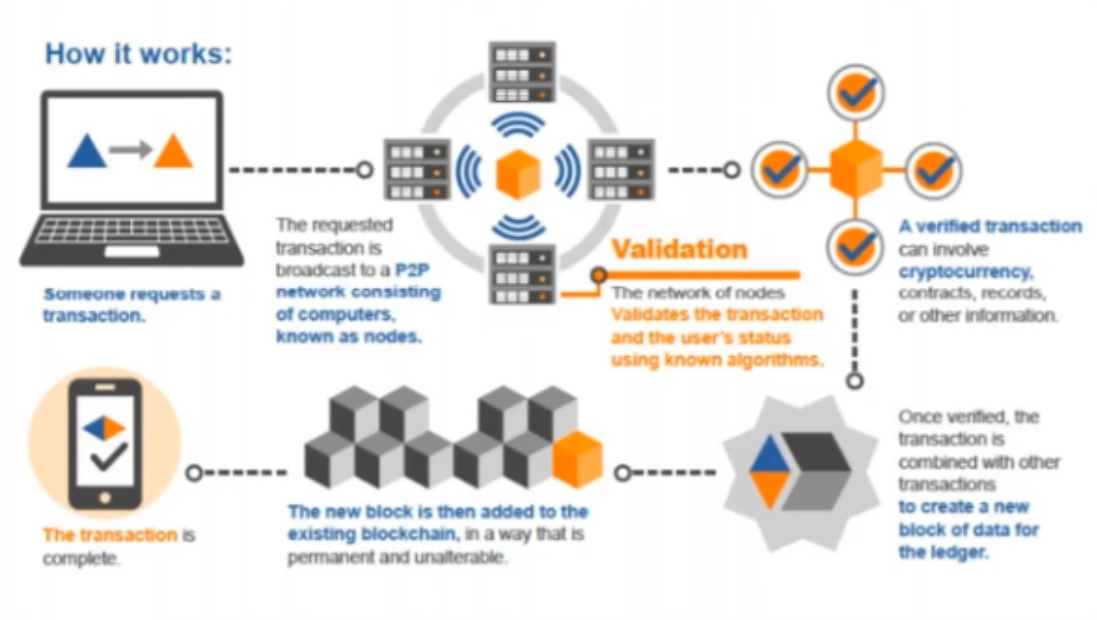


A network based on the **client–server model**, where individual *clients* request services and resources from centralized servers

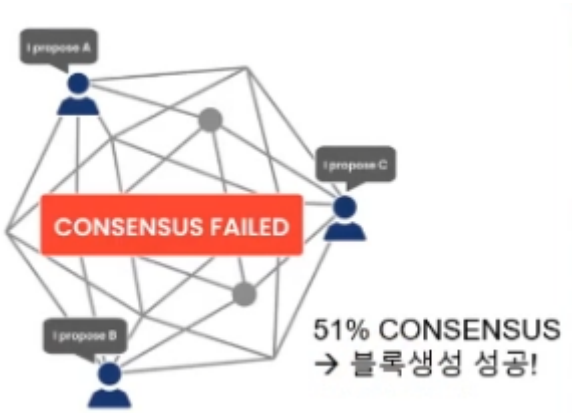


A **peer-to-peer (P2P) network** in which interconnected nodes ("peers") share resources amongst each other without the use of a centralized administrative system

- 블록체인은 기본적으로 P2P 망에서 작동한다.



1. P2P 망에 접속하면 node라는 peer 들 끼리 정보를 공유하게 된다.
2. 참여자(node)들은 모두 같이 정보를 공유할 수 있으며,
3. 누군가 거래를 발생시키면(transaction)
4. 참여자들은 이 거래가 valid 한지 서명을 확인한다.
5. 이 때 노드들 사이에서 51% 이상의 합의(consensus) 가 이루어지면
6. 정식으로 새로운 블록을 추가할 수 있다.

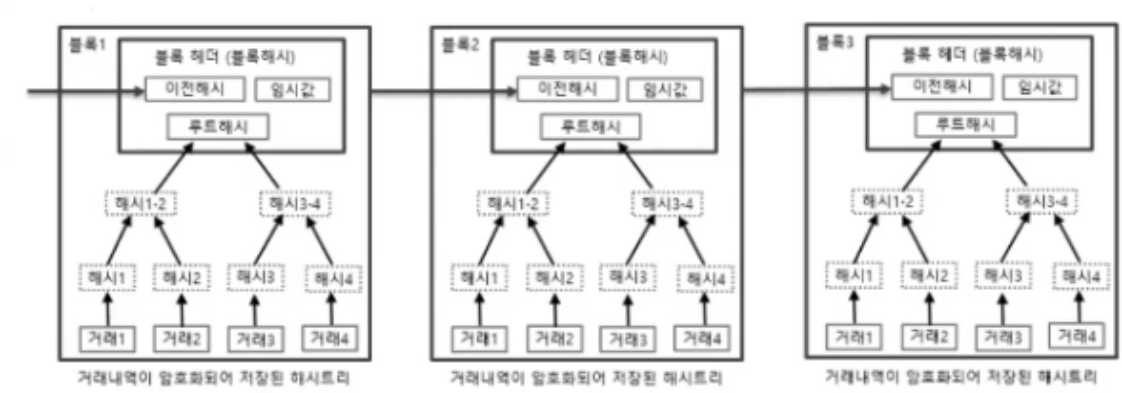


2. 탈중앙화(decentralized)

- 위의 과정처럼, 중앙 서버를 두지 않고서 참여하는 노드끼리 통신하는 구조를 말한다.

3. 특징

- 블록체인은 해킹이 매우 어렵다. 사실상 불가능에 수렴한다고도 이야기 할 수 있다.



- 블록체인은 위의 그림과 같이 거래 내역이 암호화 되어 저장 된 해시 트리들이 각각 체인으로 연결되어 있다.
- 이 때, 예를들어 블록이 100번 까지 있다 했을 때, 80번째 블록의 거래 내역을 조작하고자 한다면 80번째부터 81, 82, ... 100 번째 블록의 내용까지 모두 해킹을 통해 내용을 바꿔야 한다. 이러한 이유로 사실상 해킹이 불가능하다.

4. Applications

> 활용 분야, 블록체인을 활용하여 어떤 것을 만들 수 있는가

1. 결제

- 느려서 요즘엔 많이 사용되지 않는다.
- 비트코인은 블록으로 만드는데 10분이 걸리기 때문에 UX 측면으로 한계가 있다.

2. 데이터 Storage

- 블록체인으로 데이터를 저장하게 되면 위변조가 불가능한 무결성 상태로 유지가 가능하다는 특징을 활용한다.
- 암호화폐, 송금내역, 스마트계약, 문서관리, 의료정보관리, 전자투표와 신원 확인 emd 다양한 활용이 가능 하다. ex) SCM 물류 관리 시스템 (월마트 케이스)

3. World Computing

- P2P 에 참여하는 노드들은 모두 같은 연산을 하고 있다는 전제 하에 smart contract 를 배포하여 각각이 실행하게 된다면 모두가 같은 결과를 얻을 수 있다는 개념

참고

- 앱 개발 유형은 **Fully Decentralized** v.s **Semi-Decentralized** with Centralized Proxy 에 따라서도 구분 할 수 있다. (다음 강의 참고!)

5. 요약 정리

1. 블록체인은 보안성, 투명성, 무결성, 탈중앙화의 특징을 가진다. 이 특징을 통해 여러가지 어플리케이션을 만들 수 있다.
2. 암호화에는 복호화 가능 암호화, 단방향 암호화로 나눠진다.
    1. 단방향 암호화는 검증에 사용되며, SHA256, keccak256 등이 있다.
    2. 복호화 가능 암호화에는 대칭키 암호화, 비대칭키 암호화(공개키 암호화) 알고리즘으로 다시 구분 가능하다. 공개키 암호화의 종류는 RSA 와 전자서명, 대칭키 암호화 알고리즘에는 AES 와 ARIA, SEED 등이 있다.

6. 추가 자료

1. Proof of Work (PoW)

- 합의 알고리즘
- 블록체인에서 어떤 트랜잭션이 발생할 때, 해당 트랜잭션의 유효성에 대한 검증이 이루어진다. 이 때 유효한 트랜잭션인가 합의 하는 방식을 말한다. 보안성이 높아지는 효과가 있다.
- 작업증명 단계를 통과해야만 블록이 생성되기 때문에, 작업 증명 과정이 존재하면 P2P 의 모든 노드가 동시에 블록을 만들 수 없게 된다. 이를 통해 서비스 남용을 방지할 수 있다.
- 작업증명 알고리즘이란?
- PoW 와 PoS 의 정의

2. Hash Function 과 Salt

1. hash 함수를 활용한 사례

1. 비밀번호 암호화
    1. 회원가입시 비밀번호을 암호화 하여 저장

```
const hash = await bcrypt.hash(password, 12);
await User.create({
    password: hash,
    email,
    nick,
});
const token = signJWT(user);
```

    2. 로그인 시 비밀번호를 검증

```
const hash = await bcrypt.hash(password, 12);
await User.create({
    password: hash,
    email,
    nick,
});
const token = signJWT(user);
```

2. 토큰을 통한 인증
    1. JWT의 구성

```
xxxxx.yyyyy.zzzzz
```

- **Header** (`xxxxx`)— JWT인 토큰의 유형이나 HMAC SHA256 또는 RSA와 같이 사용되는 해시 알고리즘이 무엇으로 사용했는지 등 정보가 담긴다. Base64Url로 인코딩되어있다.
- **Payload** (`yyyyy`)— 클라이언트에 대한 정보나, meta Data같은 내용이 들어있고, Base64Url로 인코딩되어있다.
- **Signature** (`zzzzz`)— header에서 지정한 알고리즘과 secret 키, 서명으로 payload와 header를 담는다.

    2. JWT 를 통한 로그인 여부 및 사용자 검증

```
if (req.cookies['myCookie']) {
    const token = jwt.verify(req.cookies['myCookie'],
process.env.JWT_SECRET);
    const exUser = await User.findByPk(token.id);
```

2. Salt

- 소금을 치듯, 단방향 해시 함수를 보안하기 위한 방법
- 암호화에서 솔트는 데이터, 비밀번호 또는 비밀번호 문구를 해시하는 단방향 함수에 대한 추가 입력으로 사용되는 임의의 데이터.

- Salt 라는 특정 값을 통해서 해시 함수를 통해 나온 결과값을 변형하고, 해커들이 결과값으로부터 원래 암호를 유추할 수 없도록 한다.
- 해시(hash)와 암호화(Encryption) 차이점, 사용 용도
- 안전한 패스워드 저장