< 대칭키 & 공개키 >

1. 대칭키(Symmetric Key)

암호화와 복호화에 같은 암호키(대칭키)를 사용하는 알고리즘

동일한 키를 주고받기 때문에, **매우 빠르다는 장점**이 있음

대칭키 전달과정에서 해킹 위험에 노출 될 수 있다는 단점이 있음

2. 공개키(Public Key)/비대칭키(Asymmetric Key)

암호화와 복호화에 사용하는 암호키를 분리한 알고리즘

대칭키의 키 분배 문제를 해결하기 위해 고안됨.

대칭키일 때는 송수신자 간만 키를 알아야하기 때문에 분배가 복잡하고 어렵지만 공개키와 비밀키로 분리할 경우, 남들이 알아도 되는 공개키만 공개하면 되므로 키 분배 문제가 해결됨.

자신이 가지고 있는 고유한 암호키(비밀키)로만 복호화할 수 있는 암호키(공개키)를 대중에 공개함

암호화하는 키가 복호화하는 키가 서로 다르기 때문에 대칭키에 비해 암호화 복호화가 매우 복잡함

2.1) 공개키 암호화 방식 진행 과정

1. A가 웹 상에 공개된 'B의 공개키'를 이용해 평문을 암호화하여 B에게 보냄

2. B는 자신의 비밀키로 복호화한 평문을 확인, A의 공개키로 응답을 암호화하여 A에개 보냄

3. A는 자신의 비밀키로 암호화된 응답문을 복호화함

하지만 이 방식은 Confidentiallity(기밀성)만 보장해줄 뿐, Integrity(무결성)나 Authenticity는 보장해주지 못함

기밀성: 알 필요성에 근거하여 정당한 권한이 주어진 사용자, 프로세스, 시스템만 접근 가능해야 한다.

무결성: 네트워크를 통해 송수신되는 정보의 내용이 임의로 생성, 변경, 삭제가 일어나면 안된다.

2.2) 대칭키와 공개키 암호화 방식을 적절히 혼합해보면? (하이브리드 방식)

SSL 탄생의 시초가 됨

1. A가 B의 공개키로 암호화 통신에 사용할 대칭키를 암호화하고 B에게 보냄

2. B는 암호문을 받고, 자신의 비밀키로 복호화함

3. B는 A로부터 얻은 대칭키로 A에게 보낼 평문을 암호화하여 A에게 보냄

4. A는 자신의 대칭키로 암호문을 복호화함

5. 앞으로 이 대칭키로 암호화를 통신함

즉, 대칭키를 주고받을 때만 공개키 암호화 방식을 사용하고 이후에는 계속 대칭키 암호화 방식으로 통신하는 것!

Reference

https://github.com/gyoogle/tech-interview-for-developer/blob/master/Computer%20Science/Network/%EB%8C%80%EC%B9%AD%ED%82%A4%20%26%20%EA%B3%B5%EA%B0%9C%ED%82%A4.md