한 번에 끝내는
블록체인 개발 A to Z

Chapter 1

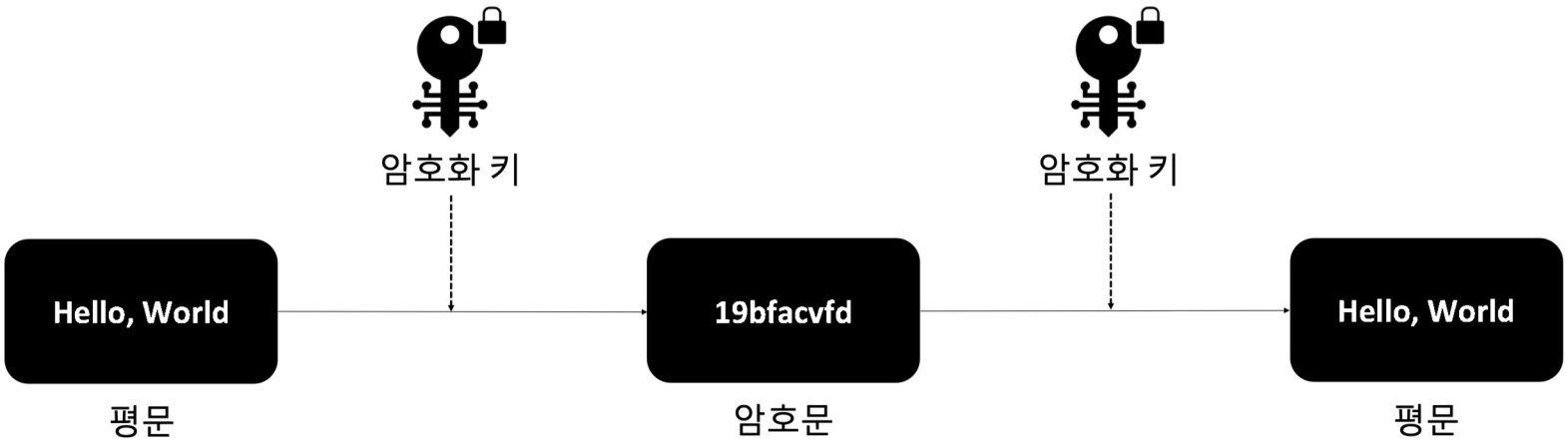
Blockchain 1.0 - Bitcoin

Chapter 1

Blockchain 1.0 - Bitcoin

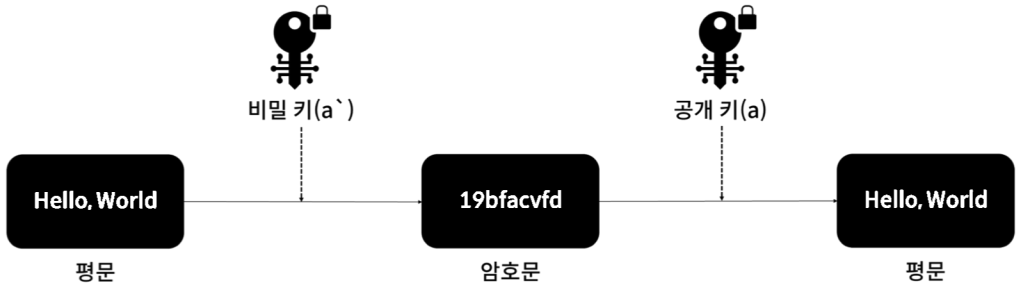
암호화 기술

암호화

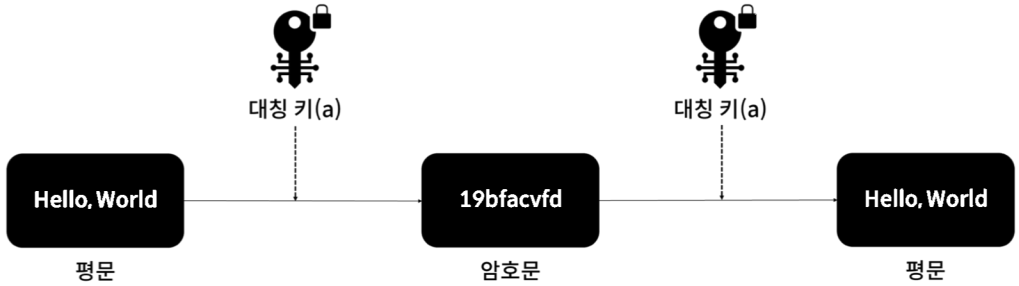


대칭키와 비대칭키

비대칭키



대칭키



RSA

❶ 공개키 암호시스템

RSA는 PKI(공개키 암호시스템)의
대표적인 알고리즘으로 1978년
처음 등장하였다.

❷ 큰 수의 소인수분해

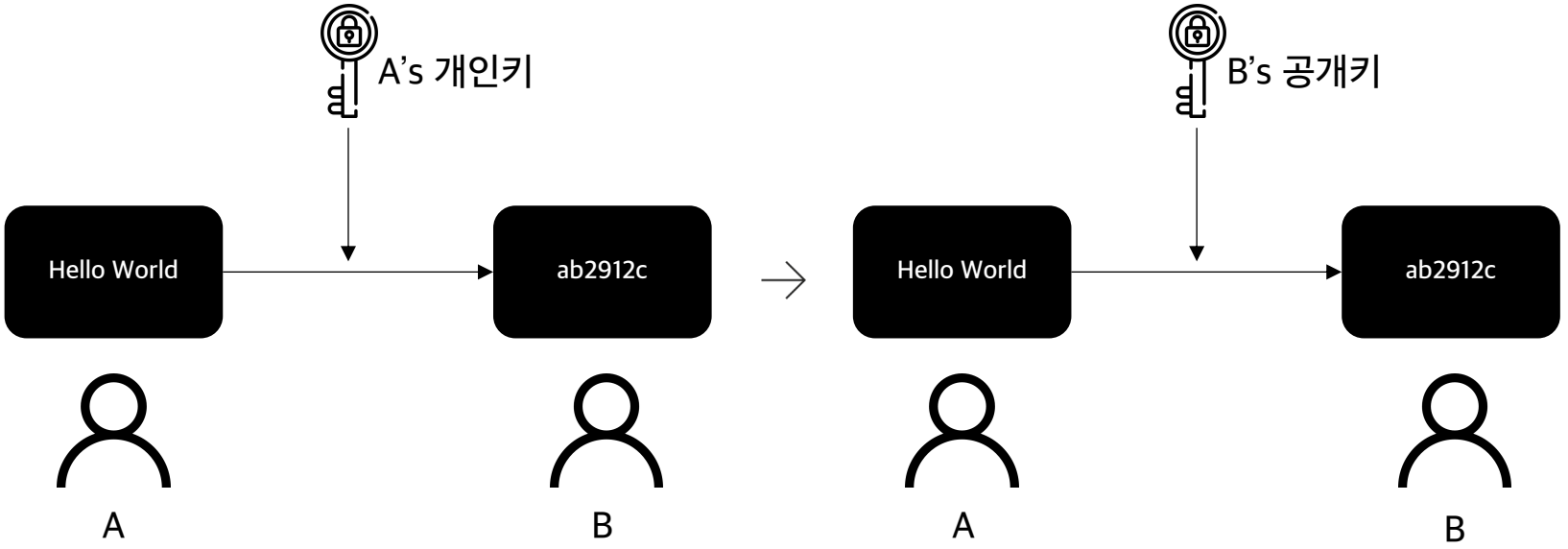
큰 수의 소인수 분해가 어렵다는
수학적 특징을 이용한
알고리즘으로 양자 컴퓨터 등장 시
무용지물 될 가능성이 존재한다.

❸ 비대칭키

공개키와 비밀키 두 개를 사용한
비대칭키 방식으로 어떤 키로
암호화 하느냐에 따라 다른 방식을
제공한다.

❹ RSA-2048

RSA 2048은 2048개의 bit를
사용하는 가장 대중적인
알고리즘으로 인터넷 뱅킹의
대부분이 사용 중 이다.



개인키 암호화

공개키 암호화

이산대수 문제의 어려움(ElGamal 방정식)

p가 소수이고 g가 원시원소일 때 g,x,p 를 이용하여 $y \equiv g^x \ mod \ p$ 를 구하긴 쉽지만, g,y,p 값을 이용하여 지수승 x를 구하기는 어려움을 이산대수 문제의 어려움이라 칭함

예제) g=5, p=23

$mod\ 23$	$mod\ 23$	$mod\ 23$
$5^1 \equiv 5$	$5^8 \equiv 16$	$5^{15} \equiv 19$
$5^2 \equiv 2$	$5^9 \equiv 11$	$5^{16} \equiv 3$
$5^3 \equiv 10$	$5^{10} \equiv 9$	$5^{17} \equiv 15$
$5^4 \equiv 4$	$5^{11} \equiv 22$	$5^{18} \equiv 6$
$5^5 \equiv 20$	$5^{12} \equiv 18$	$5^{19} \equiv 7$
$5^6 \equiv 8$	$5^{13} \equiv 21$	$5^{20} \equiv 12$
$5^7 \equiv 17$	$5^{14} \equiv 13$	$5^{21} \equiv 14$

y = 7 일때, x의 값은 얼마 인가?

이산대수 문제의 어려움(ElGamal 방정식)

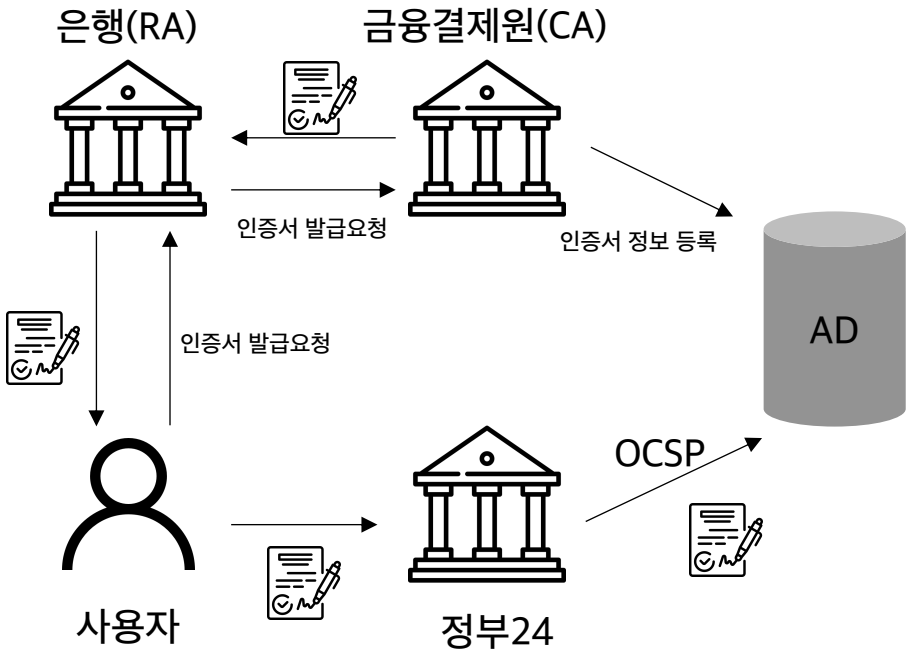
1) 두 소수 p, q 를 생성한다.
2) p-1, q-1과 각각 서로서인 정수 e를 준비한다.
3) ed를 (p-1)(q-1)으로 나눈 나머지가 1이 되도록 하는 d를 찾는다.
4) N = pq를 계산한 후, N과 e를 공개한다.
5) p, q, (p-1)(q-1)를 파기한다.
공개키는 e가 되고, d는 개인키가 된다.

암호화) $x \equiv a^e \ mod \ N$ (평문 a을 공개키로 암호화하여 x 생성)

복호화) $a` \equiv x^d \ mod \ N$ (암호화문 x를 개인키 d로 복호화하여 a 생성)

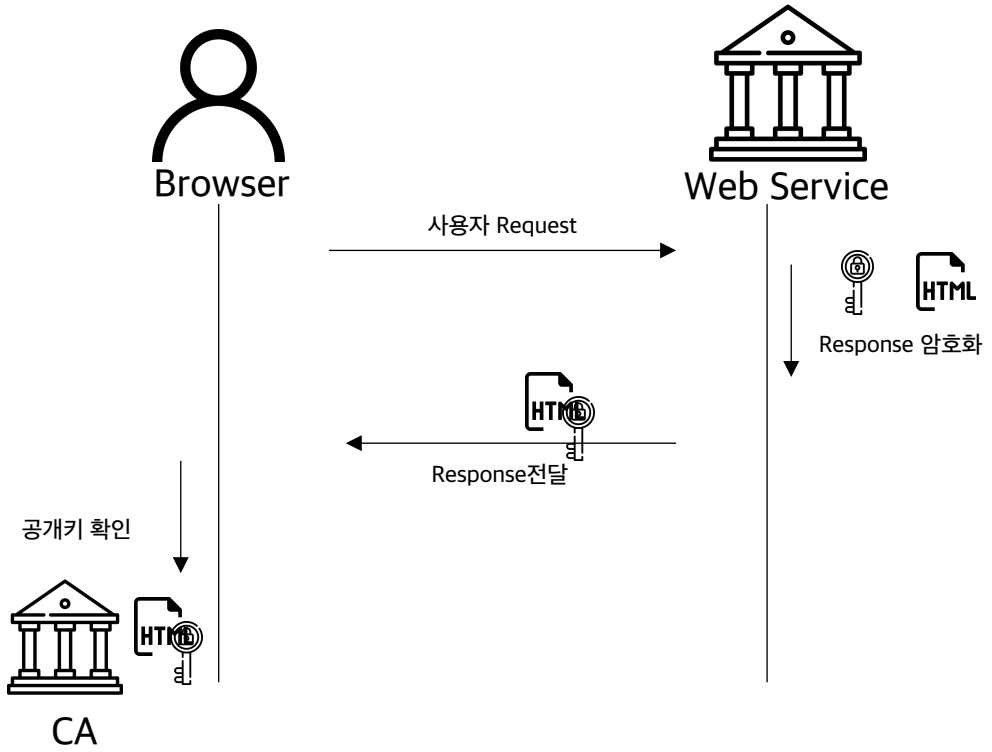
공인인증서

공인인증서는 RSA-2048 알고리즘으로
발행된 X509표준의 인증서이다.
공개키와 개인키를 포함하여 발급되며
개인키는 사용자의 PC에 저장되고 공개키는
사용자와 AD(Active Directory)에 모두
저장되는 형태이다.



SSL 인증서

HTTPS는 HTTP Protocol 위에 데이터 송수신 데이터의 안전한 전송을 위해서 진행하는 SSL 인증 전송 방식이다.
SSL 인증서는 CA 기관을 통하거나 Self-Signed 방식을 통해서 발급이 누구든 가능하다.



```
String pwd = "password";

KeyPair keyPair = generator.generateKeyPair();
PrivateKey privateKey = keyPair.getPrivate();

String EncryptedPwd = encrypt(privateKey, pwd);

public String encrypt(PrivateKey priv, String pwd){


  Cipher cipher = Cipher.getInstance("RSA");
  cipher.init(Cipher.ENCRYPT_MODE, priv);
  byte[] bytePlain = cipher.doFinal(plainText.getBytes());

  String enc_pwd = Base64.getEncoder().encodeToString(bytePlain);
  return enc_pwd
}
```

RSA 암호화 방식

```
String pwd = "password";
String EncryptedPwd = encrypt(pwd);

public String encrypt(String pwd){

  MessageDigest md = MessageDigest.getInstance("SHA-256");

  md.update(pwd.getBytes());

  String hex = String.format("%x",new BigInteger(1, md.digest()));
  return hexPwd
}
```

Hash 방식