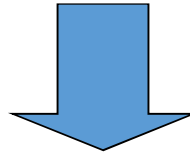


# 암호화와 인증서

# 암호화의 역할

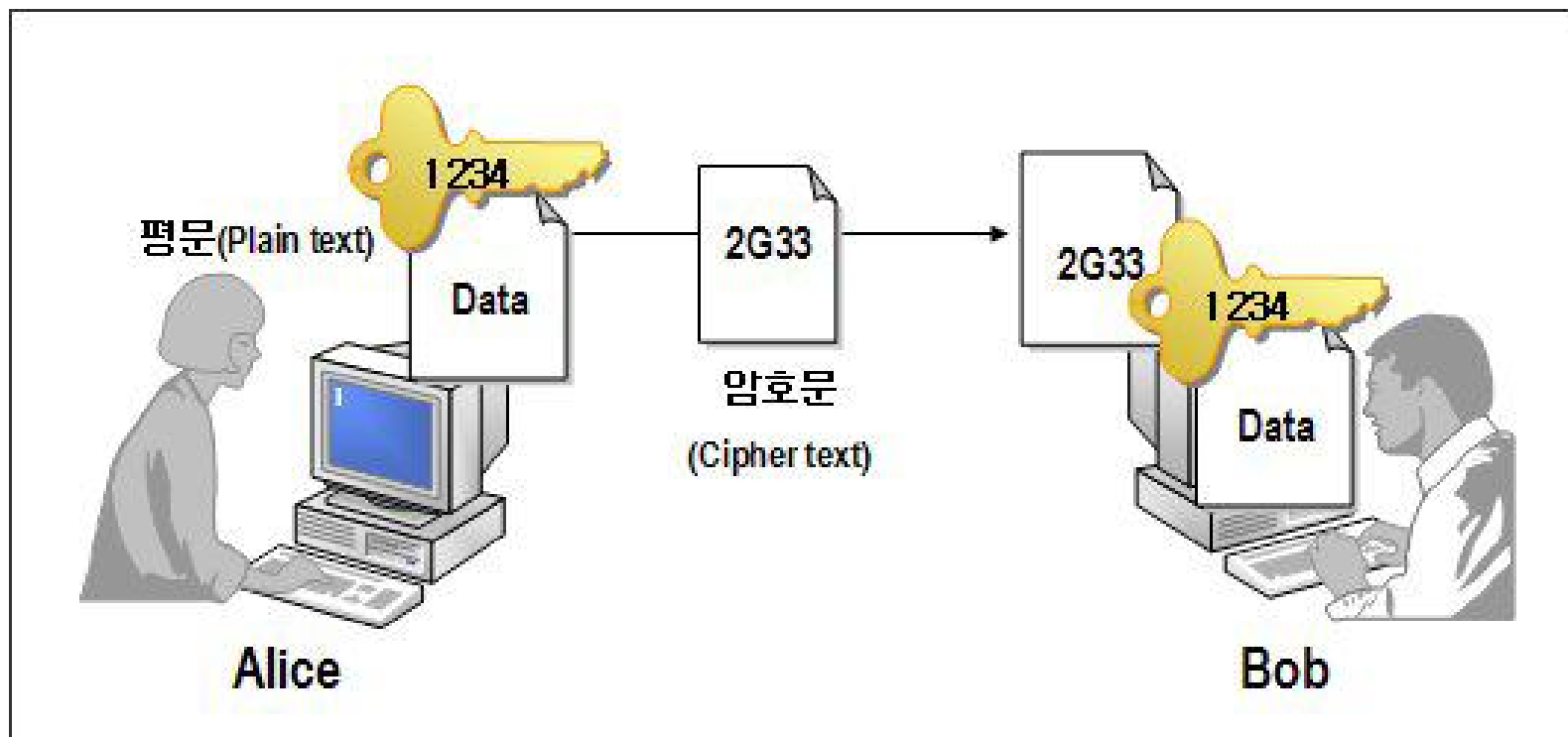
- Confidentiality (기밀성)
- Authenticity (신뢰성)
- Integrity (무결성)



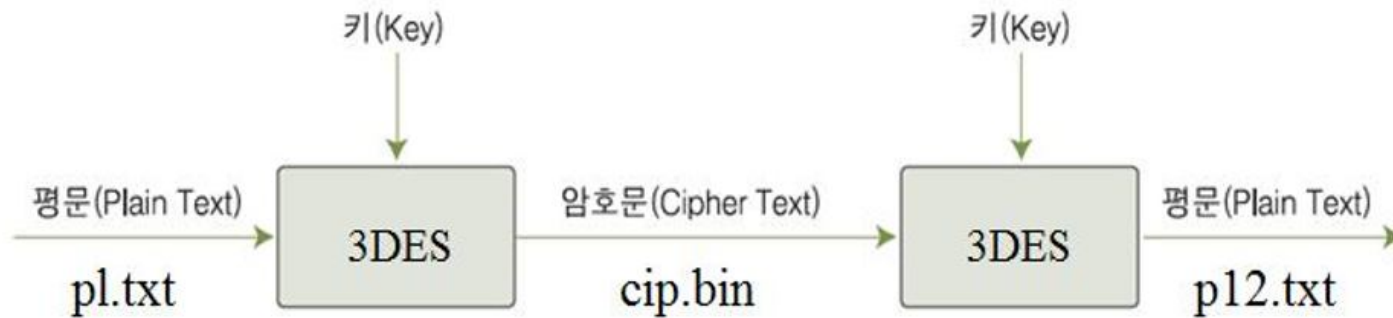
Key를 이용한 암호화 기술로서 해결

# Symmetric Key

- Encryption Key = Decryption Key



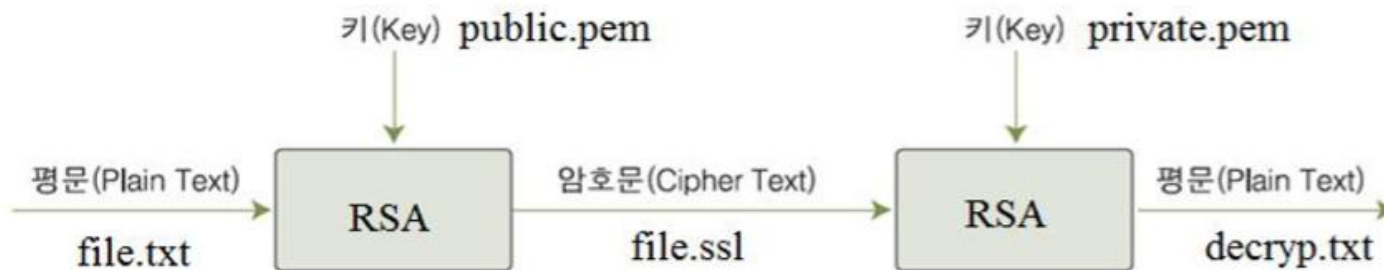
- DES, 3DES 등의 프로토콜이 해당됨



암호화	<pre>echo 'this is the plain text' &gt; pl.txt openssl enc -e -des3 -salt -in pl.txt -out cip.bin</pre> <ul style="list-style-type: none"> <li>· <code>enc -e -des3</code> : des3 방식으로 암호화</li> <li>· <code>-in pl.txt -out cip.bin</code> : pl.txt 파일의 암호화 한 결과를 cip.bin 파일에 저장</li> </ul>
복호화	<pre>openssl enc -d -des3 -in cip.bin -out p2.txt</pre> <ul style="list-style-type: none"> <li>· <code>enc -d -des3</code> : cip.bin 파일을 p2.txt 파일로 복호화</li> </ul>

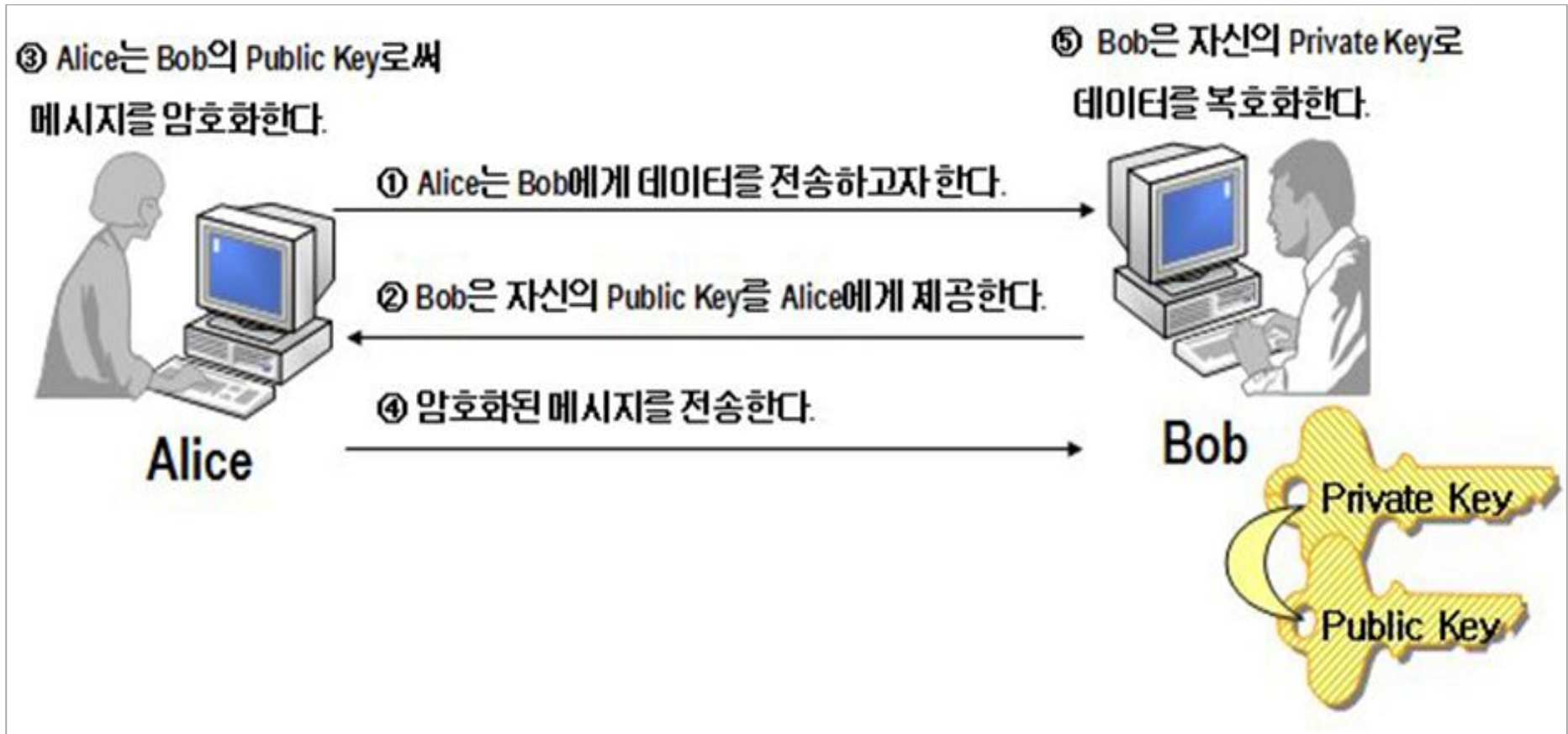
# Public Key

- Encryption Key  $\neq$  Decryption Key
  - 비대칭키, 공용키, 페어(Pair)키에 해당함
- Public Key와 Private Key로 구성
  - Public Key(공용키) = 공개가 되는 키
  - Private Key(개인키) = 오직 발행주체만이 가지는 비밀키
- RSA 프로토콜

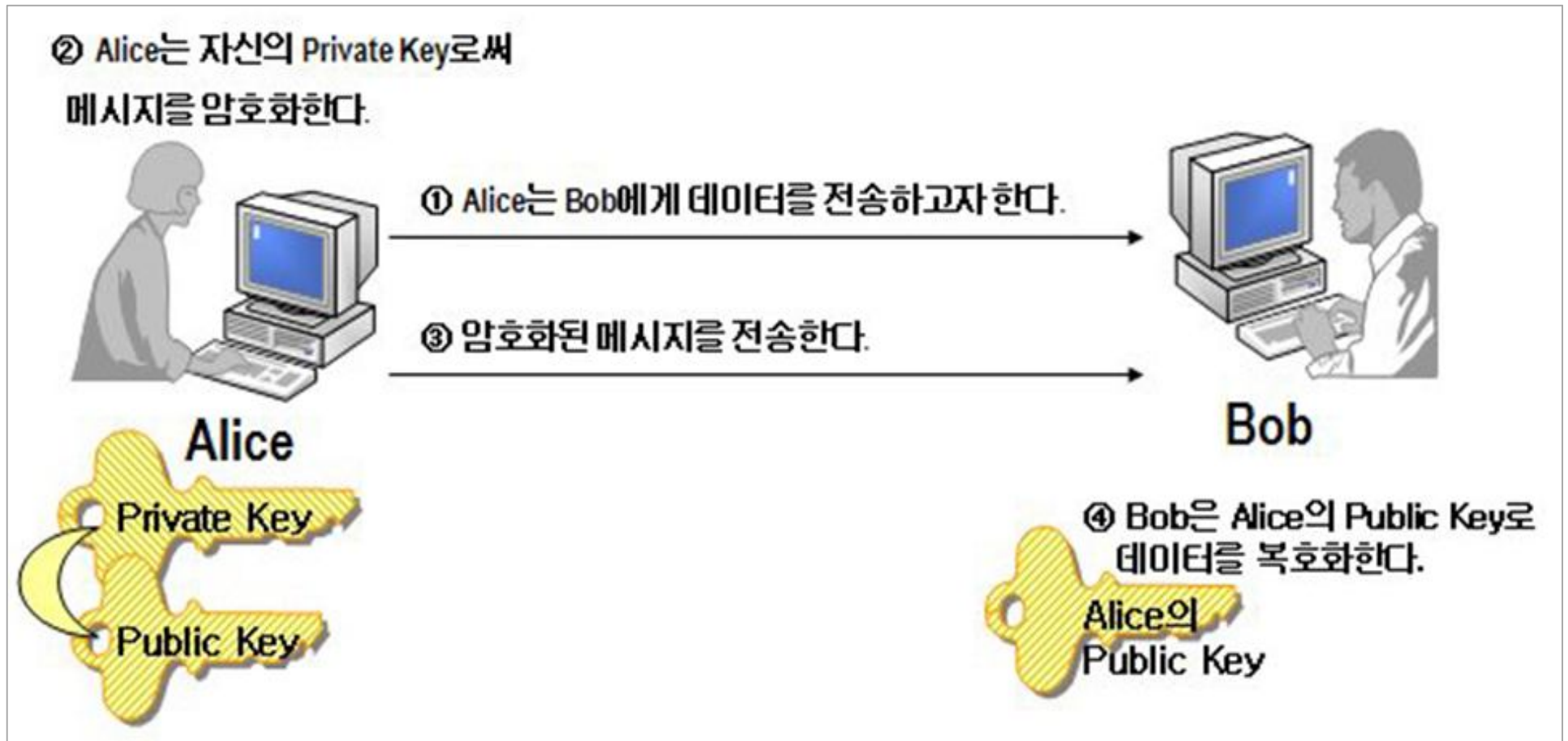


개인키 생성	<pre>openssl genrsa -out private.pem 1024</pre> <ul style="list-style-type: none"> <li>• RSA로 개인키 private.pem 생성, • private.pem 키는 1024bit 길이로 이 숫자가 높을수록 안전</li> </ul>
공개키 생성	<pre>openssl rsa -in private.pem -out public.pem -outform PEM -pubout</pre> <ul style="list-style-type: none"> <li>• 개인키 private.pem 에 대한 공개키 public.pem 생성</li> <li>• 공개키를 자신에게 정보를 제공할 사람에게 전송</li> </ul>
공개키로 암호화	<pre>echo 'coding everybody' &gt; file.txt openssl rsautl -encrypt -inkey public.pem -pubin -in file.txt -out file.ssl</pre> <ul style="list-style-type: none"> <li>• file.txt의 내용을 RSA방식으로 암호화한 파일 file.ssl 생성</li> </ul>
개인키로 복호화	<pre>openssl rsautl -decrypt -inkey private.pem -in file.ssl -out decryp.txt</pre> <ul style="list-style-type: none"> <li>• 개인키 private.pem로 file.ssl의 복호화한 결과를 decryp.txt로 생성</li> </ul>

## Public Key 암호화(Confidentiality 제공)

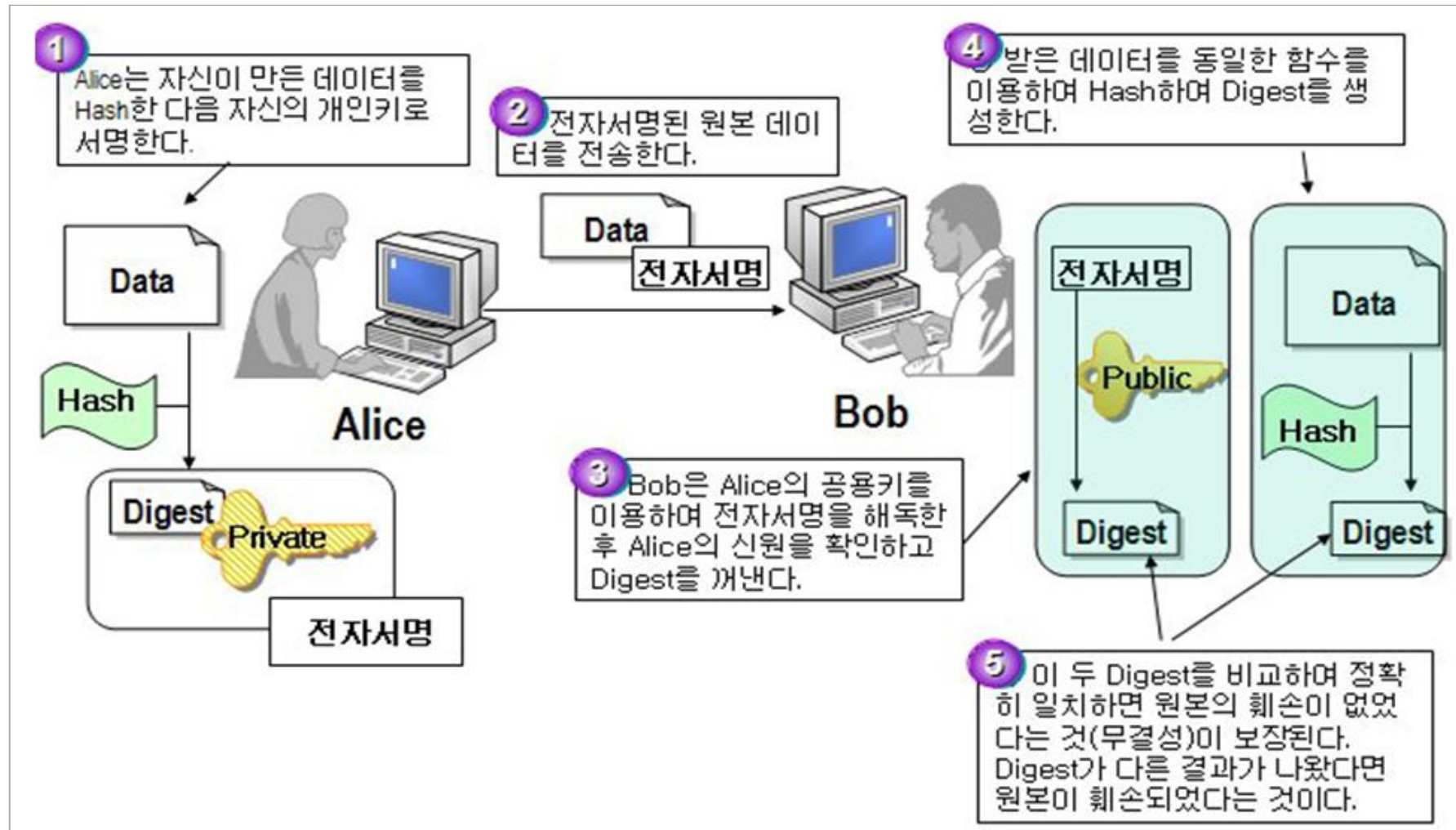


## Private Key 암호화(Authenticity 제공)



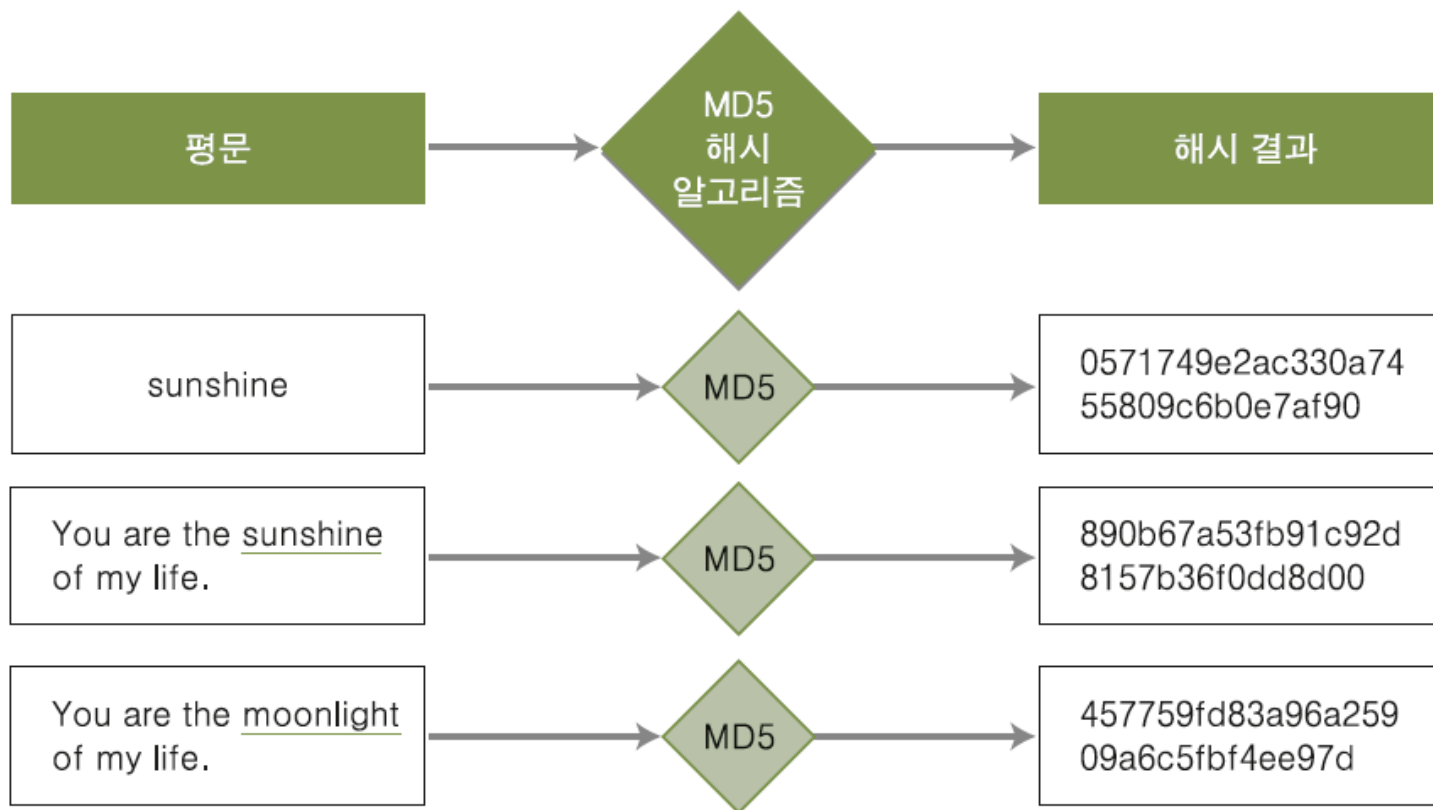


# Hash Function & Digest(Integrity 제공)



# Hash Algorithm

- 해시 : 하나의 문자열을, 이를 상징하는 더 짧은 길이의 값이나 키로 변환하는 것



■ MD 알고리즘

- MD2, MD4, MD5 이렇게 세 가지가 있음.
- MD5 알고리즘은 MD4의 확장판으로, MD4보다 속도가 빠르지는 않지만 데이터 보안성에 있어 더 많은 확신을 제공

■ SHA 알고리즘

- 160비트의 값을 생성하는 해시 함수로, MD4가 발전한 형태
- MD5보다 조금 느리지만 좀더 안전한 것으로 알려져 있음

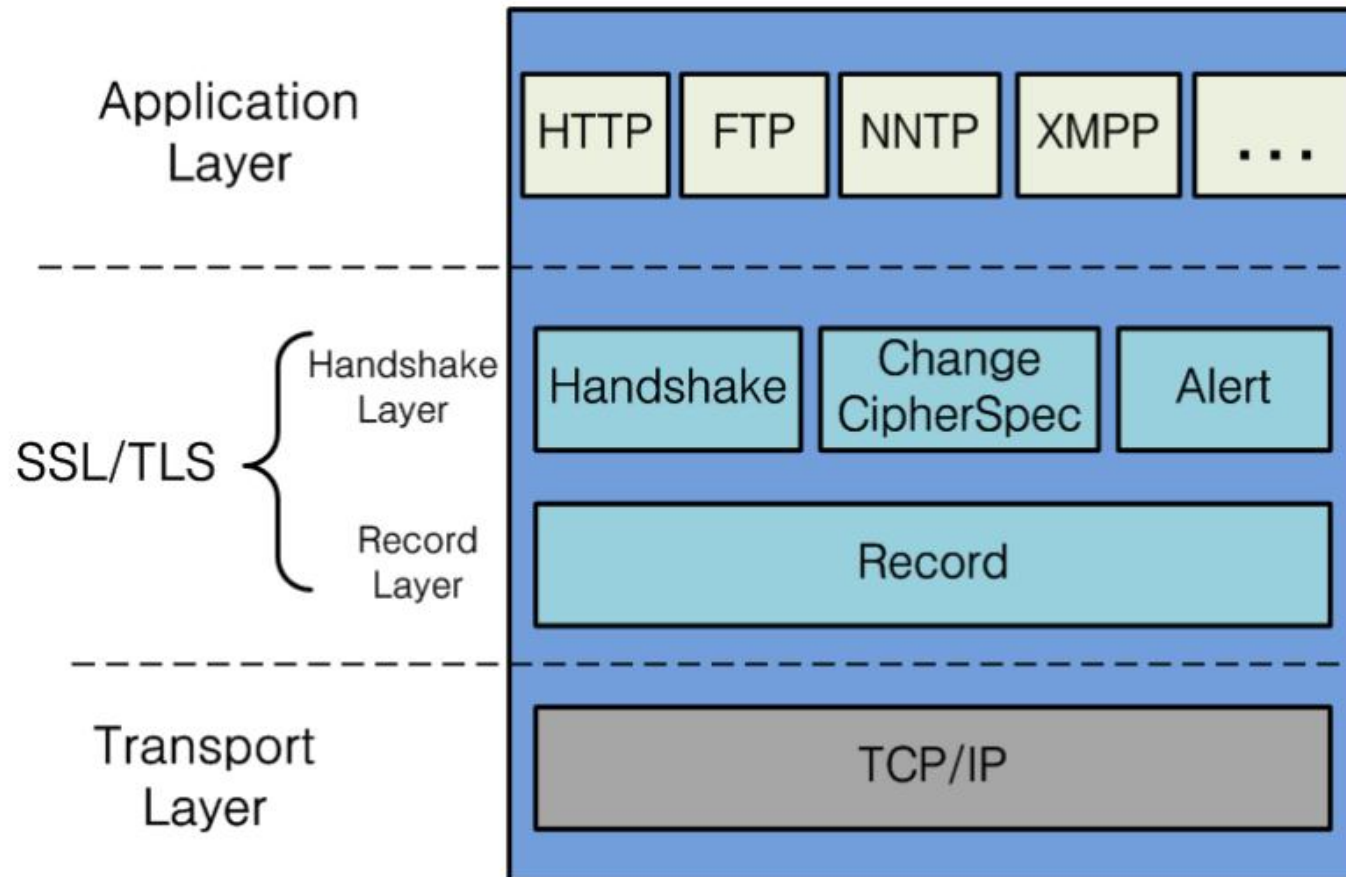
알고리즘	블록 크기	해시 결과값 길이	해시 강도
SHA-1	512비트	160비트	0.625
SHA-256	512비트	256비트	1
SHA-384	1024비트	384비트	1.5
SHA-512	1024비트	512비트	2

# SSL/TLS

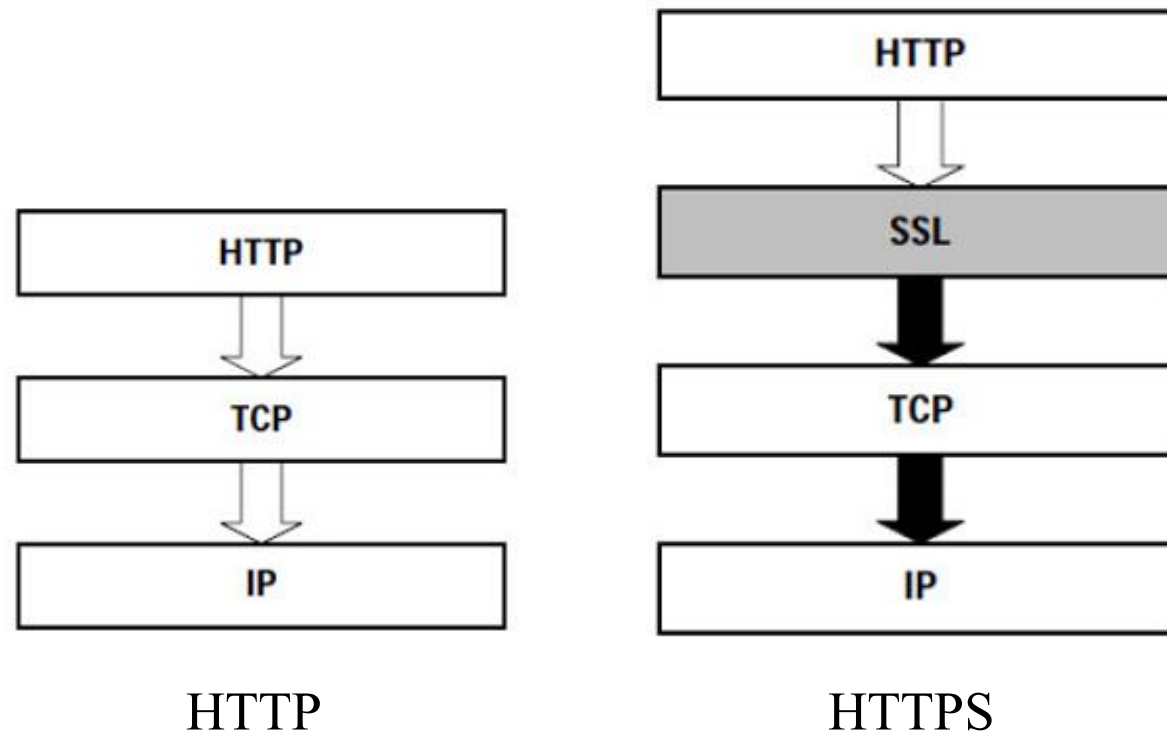
- 응용계층을 보호하는 프로토콜
- 웹 브라우저에 기본적으로 탑재되어 있음
- IETF에서 TLS 1.0 발표, SSL 3.1 이라고도 함
- TLS 프로토콜은 SSL 프로토콜을 사용할 수 있도록 구성
- SSL/TLS 프로토콜이라는 용어 사용

SSL(Secure Socket Layer)	TLS(Transport Layer Security)
응용계층을 보호하는 프로토콜	전송 계층 상위에서 동작하여 응용계층 암호화 HTTP 패킷 보호가 주용도

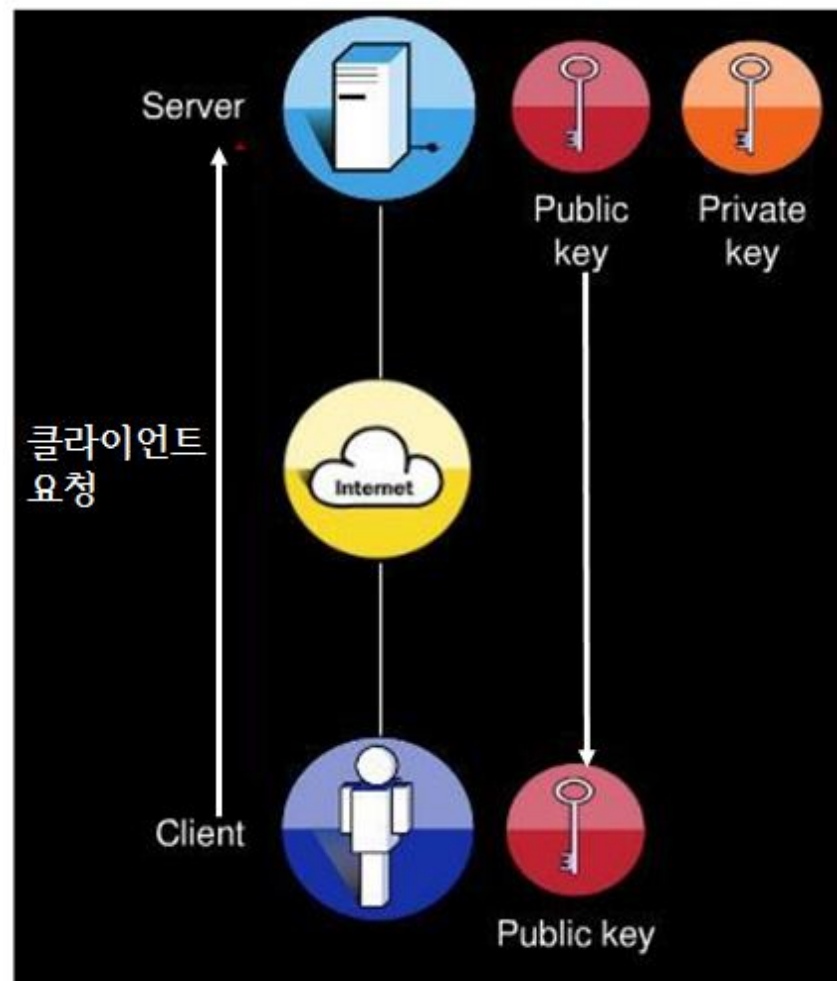
# SSL/TLS Protocol stack



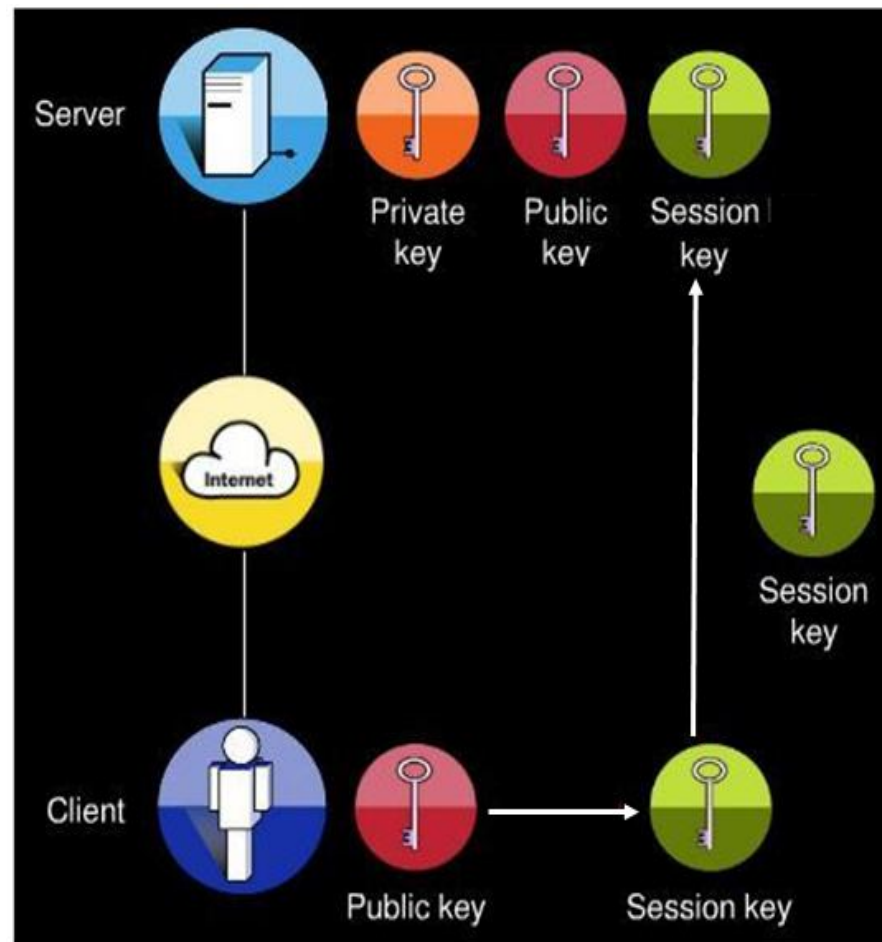
- HTTP와 HTTPS



# SSL/TLS Processing

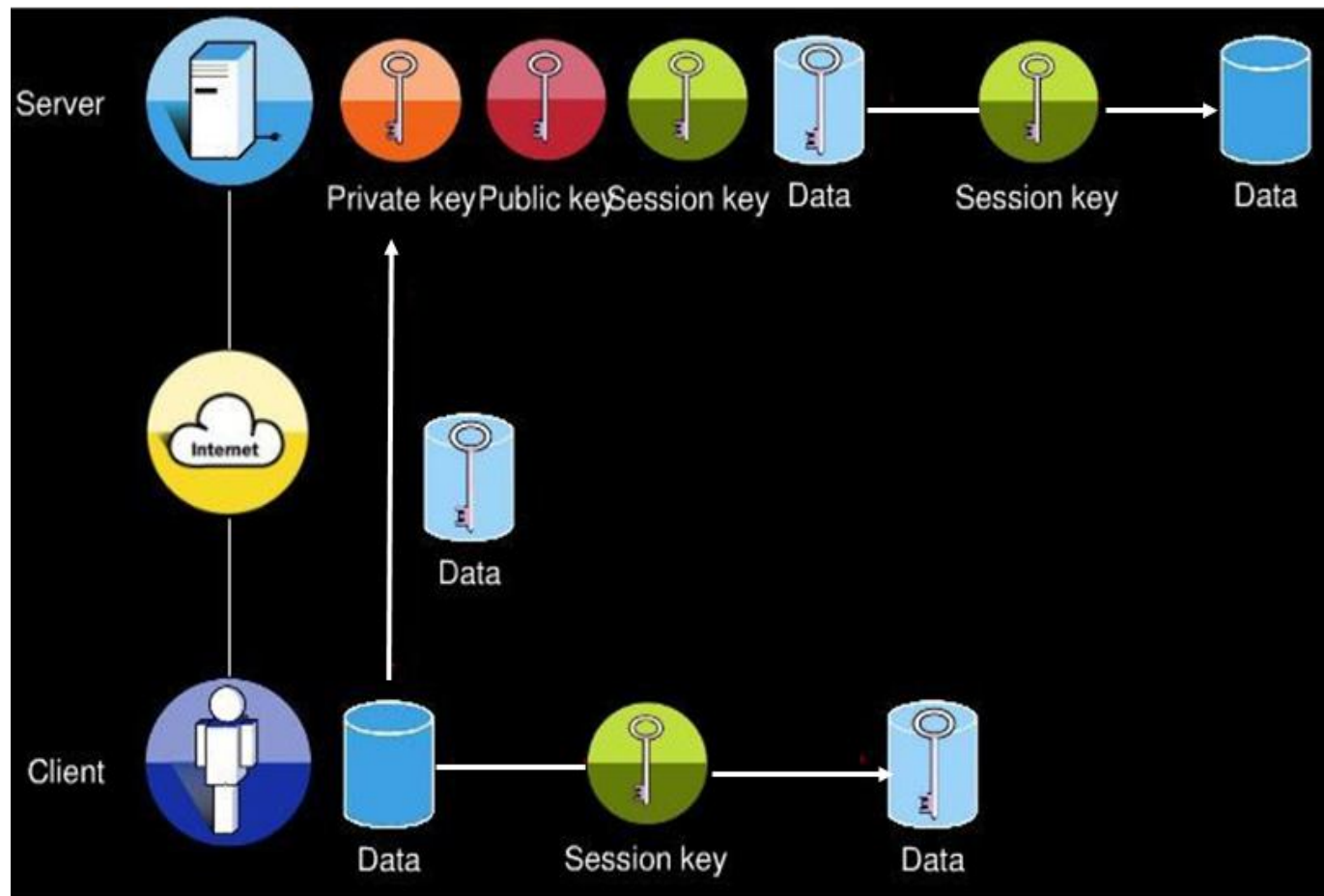


# SSL/TLS Processing





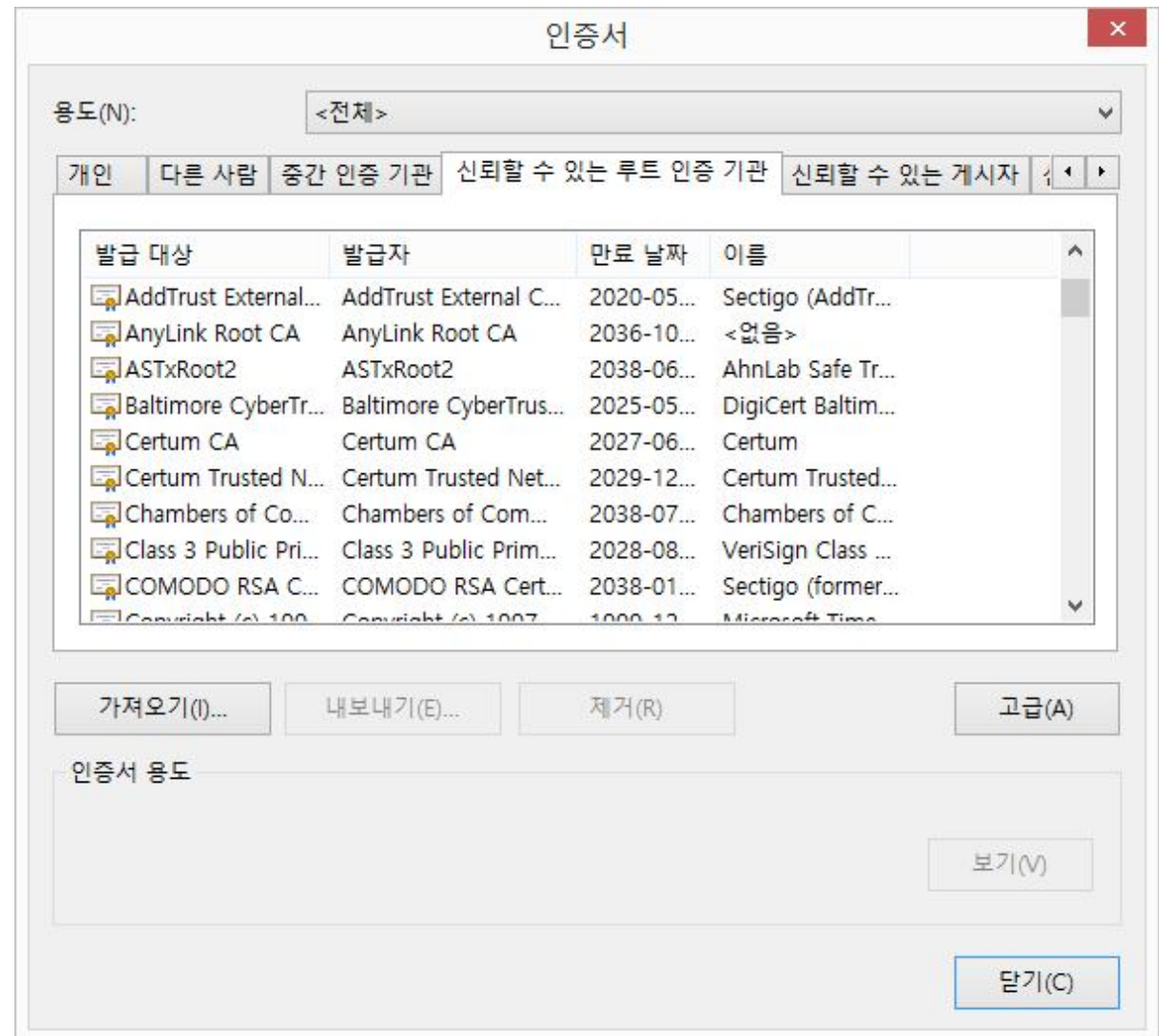
# SSL/TLS Processing

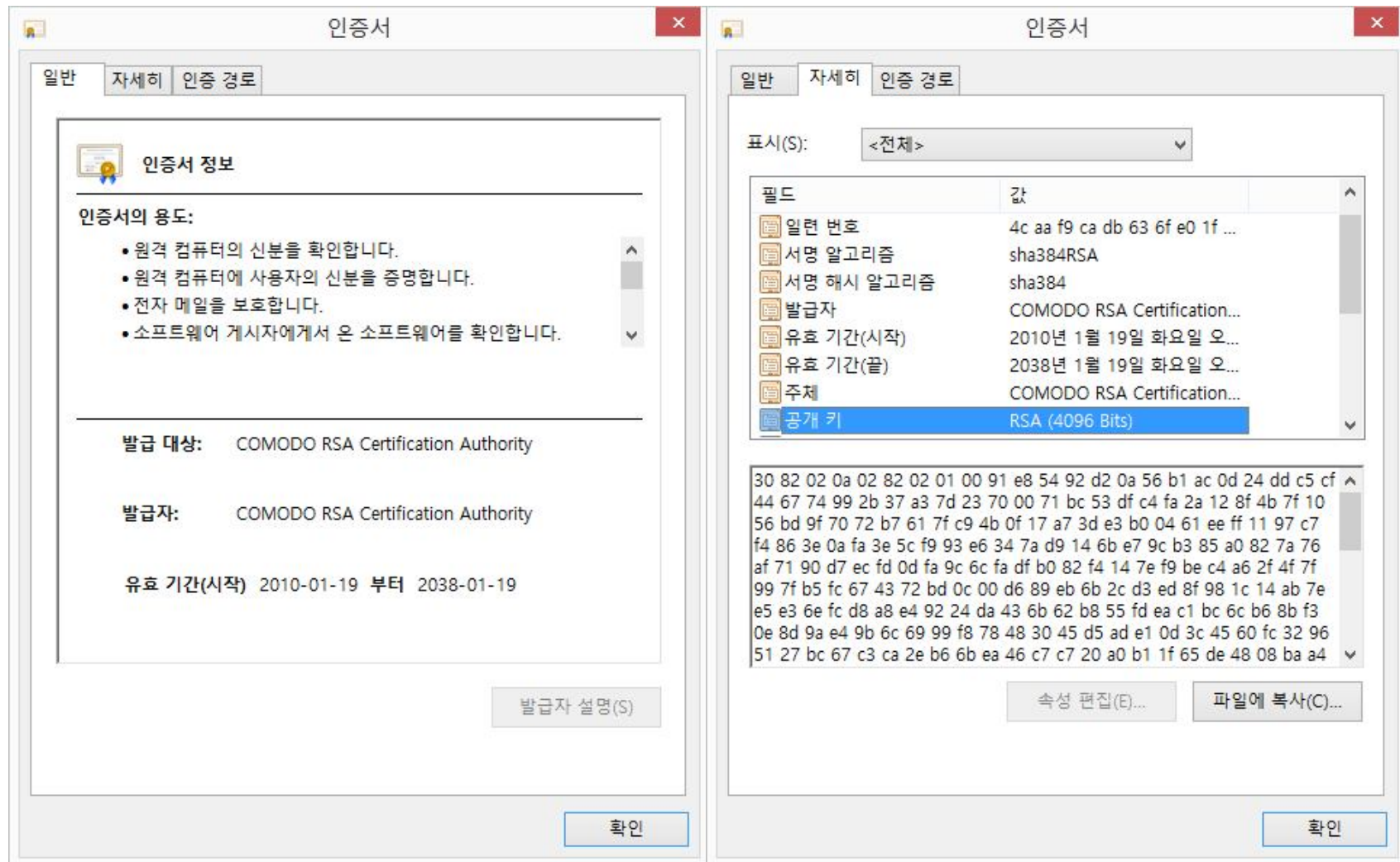


# 공개 키 기반 구조(Public Key Infrastructure, PKI)

- 메시지의 암호화 및 전자서명을 제공하는 복합적인 보안 시스템환경
- 공개 키를 효과적으로 운용하기 위해 정해진 많은 규격이나 선택사항
  - 디지털 인증서 생성, 관리, 저장, 배분, 취소 (폐지)
- 에 필요한 하드웨어, 소프트웨어, 사람, 정책 및 절차라고 정의
- 공인인증서가 신분증과 같은 효력을 발휘하려면 검증 기관 필요.
- 공개 키 기반 구조는 ‘인터넷에서 신분증을 검증해주는 관청’ 역할.
- 인증 기관(Certification Authority, CA)에서 공인인증서를 이용하여 증명 가능

# X.509 인증서





인증기관(CA)



웹브라우저 회사

Web서버



클라이언트



이 웹 사이트의 보안 인증서에 문제가 있습니다.

이 웹 사이트에서 제시한 보안 인증서는 만료되었거나 아직 유효하지 않습

문제가 있는 인증서를 통해 사용자를 속이거나 사용자가 서버로 보내는 데 가로챌 수도 있습니다.

이 웹 페이지를 닫고 이 웹 사이트를 계속 탐색하지 않는 것이 좋습니다.

✓ 이 웹 페이지를 닫으려면 여기를 클릭하십시오.

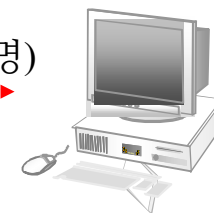
✗ 이 웹 사이트를 계속 탐색합니다(권장하지 않음).

⌵ 추가 정보



Web서버

⑦ 인증서 제출 (Web서버 공개키/CA 전자서명)



⑧ 인증서  
신뢰확인

클라이언트

