

# 12장 네트워크 보안

---

정보보호이론

Spring 2015



고려대학교  
KOREA UNIVERSITY

# 12.1 SSL/TLS 프로토콜

---

두 가지 암호모듈 적용방안

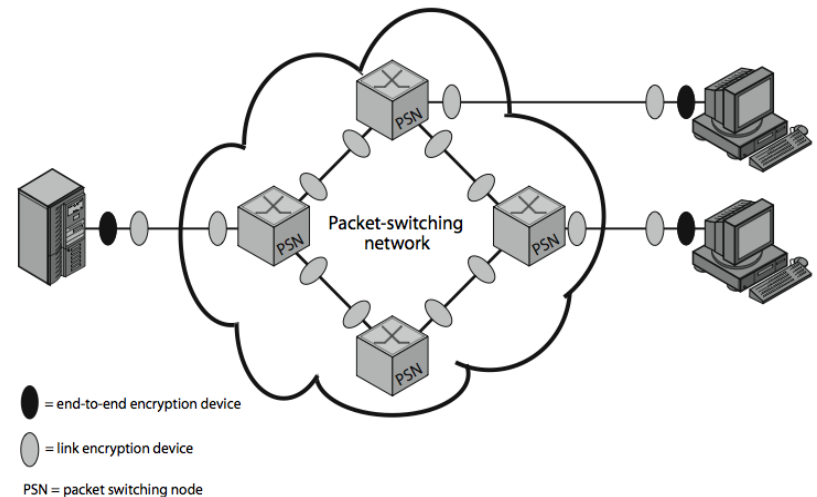
## ■ 링크 암호화

- ✗ encryption occurs independently on every link
- ✗ implies must decrypt traffic between links
- ✗ requires many devices, but paired keys

## ■ end-to-end encryption

- ✗ encryption occurs between original source and final destination
- ✗ need devices at each end

with shared keys



# 12.1 SSL/TLS 프로토콜

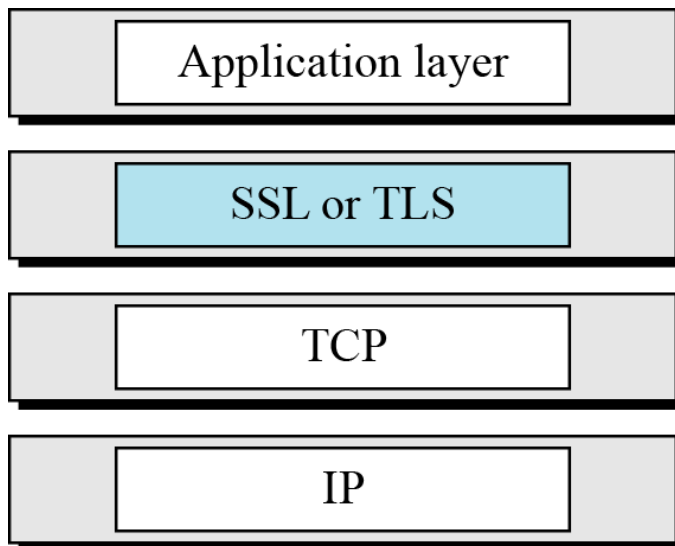
---

- When using end-to-end encryption must leave headers in clear
  - ✗ so network can correctly route information
  - ✗ hence although contents protected, traffic pattern flows are not
- ideally want both at once
  - ✗ end-to-end protects data contents over entire path and provides authentication
  - ✗ link protects traffic flows from monitoring

# 12.1 SSL/TLS 프로토콜

---

- SSL/TLS Provides auth., data confidentiality, and data integrity bet'n client and server.
- Application programs such as HTTP can encapsulate their data in SSL packets. → https://.. to allow HTTP messages to be encapsulated in SSL packets.



**HTTPS** is Hypertext Transfer Protocol Secure (HTTPS) is a combination of the Hypertext Transfer Protocol (HTTP) with SSL/TLS protocol to provide encrypted communication and secure identification of a network web server. HTTPS connections are often used for payment transactions on the World Wide Web and for sensitive transactions in corporate information systems.

## 12.1 SSL/TLS 프로토콜

---

- SSL is designed to provide security and compression services to data generated from the application layer.
- Uses TCP to provide a reliable end-to-end service
- Services : Fragmentation, Compression, Message Integrity, Confidentiality, Framing (A header is added to the encrypted payload, which is then passed to a reliable transport layer protocol.)
- Ex: <https://www.bankofamerica.com/>
  - ✕ EV(Extended Validation Certificate)

# 12.1 SSL/TLS 프로토콜

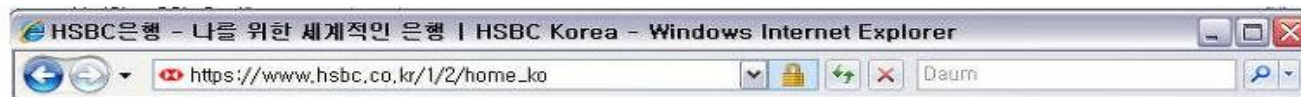
---

## ■ EV(Extended Validation Certificate)

- ✗ Commercial pressures have led some CAs to introduce "domain validation only" SSL certificates for which minimal verification is performed of the details in the certificate.

### ✗ EVSSL

- ▶ 특정 웹 사이트가 공인인증기관으로부터 엄격한 심사규정을 통해 실존 여부를 검증
- ▶ 개인정보를 입력하는 페이지는 암호화하여 안전하게 전송하고 있다는 것을 고객들이 직접 눈으로 웹 사이트 주소창을 통해 확인
- ▶ 전세계적으로 75%이상의 브라우저들이 EV SSL 인증서를 지원, 국내에서도 대형 금융, 쇼핑몰, 포털사이트 등 사용업체 증가



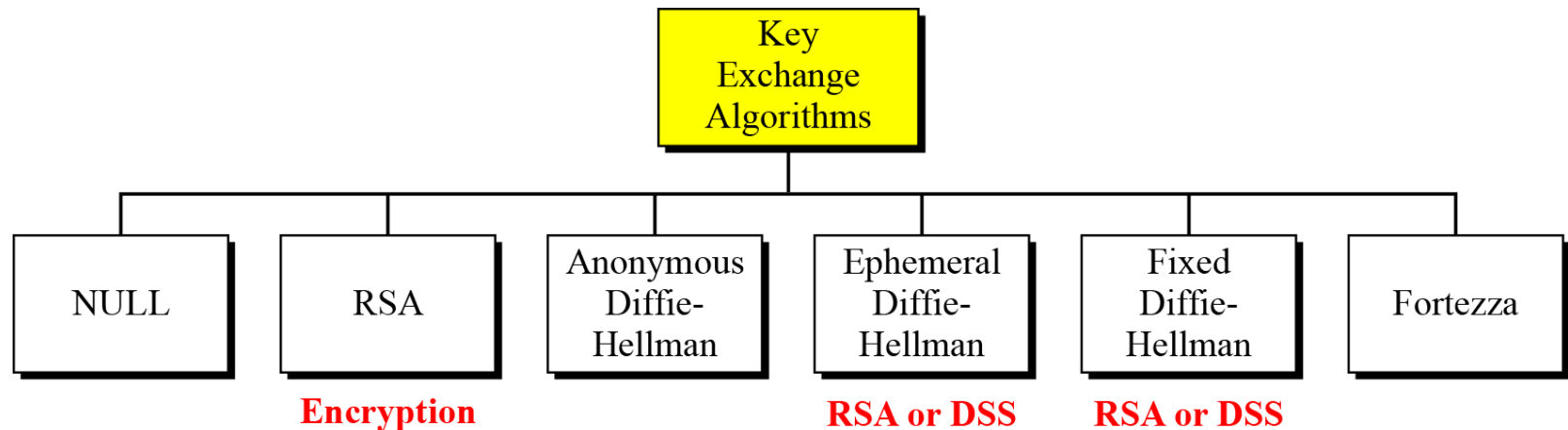
(EV SSL 인증서인 경우)



## 12.1.2 SSL 프로토콜 구조

---

Key-exchange methods (to generate pre-master secret)



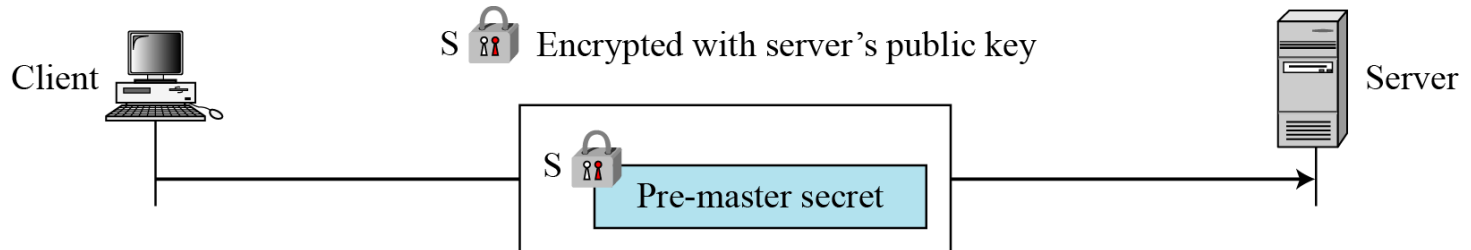
### ■ NULL

✗ There is no key exchange in this method. No pre-master secret is established between the client and the server.

➔ To generate a session key, need the pre-master secret

## 12.1.2 SSL 프로토콜 구조

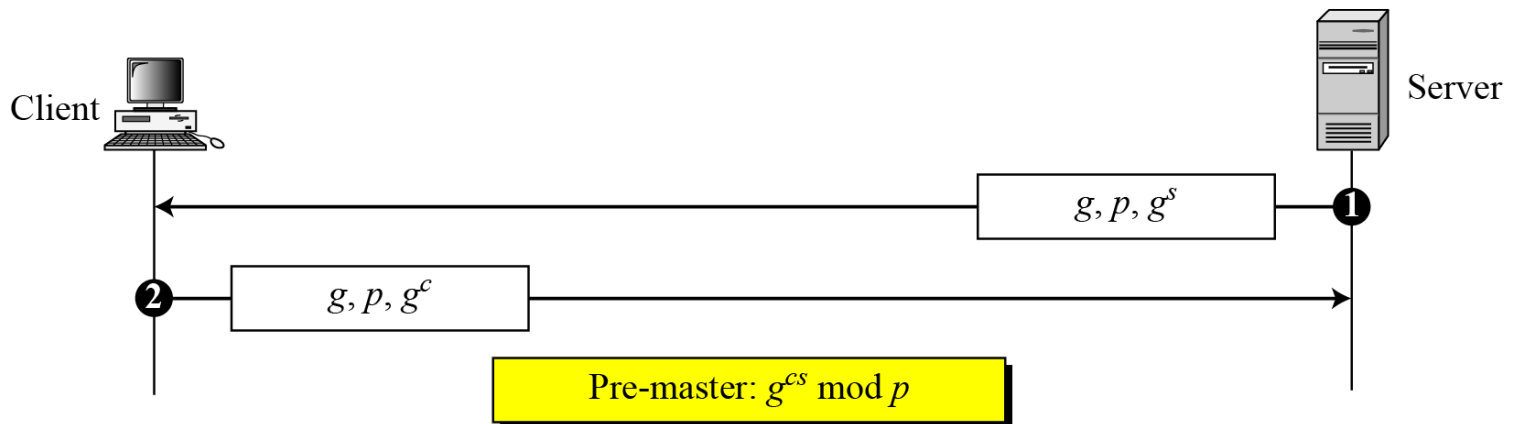
### ■ RSA key exchange; server public key



✗ Pre-master secret : 48-byte random number created by the client

### ■ Anonymous Diffie-Hellman key exchange

✗ Insecure one

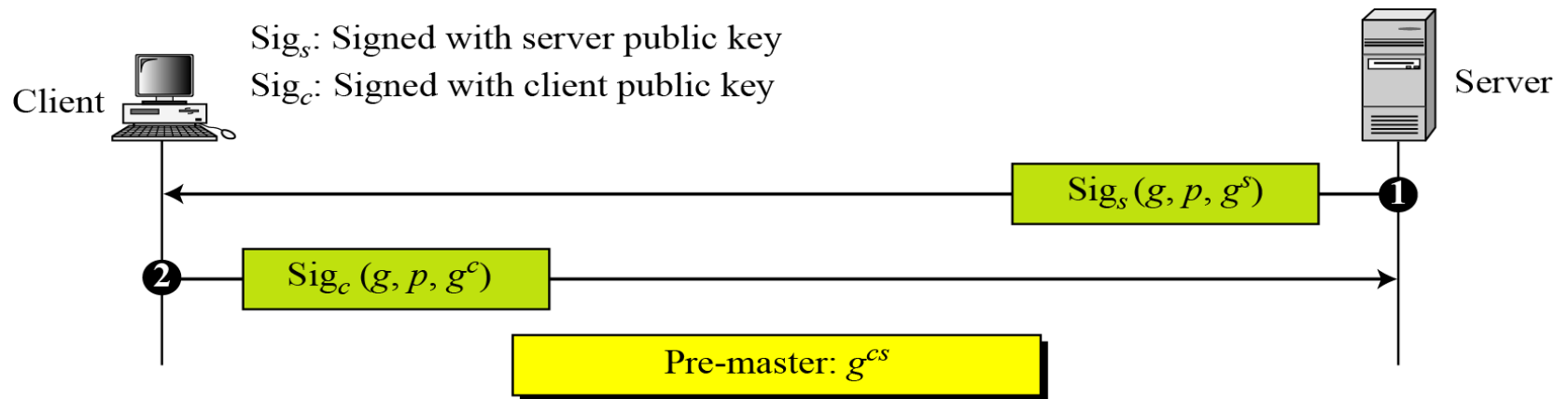




## 12.1.2 SSL 프로토콜 구조

### ■ Ephemeral Diffie-Hellman key exchange

✗ Need certificate



### ■ Fixed Diffie-Hellman

✗ DH half key inserted in a certificate

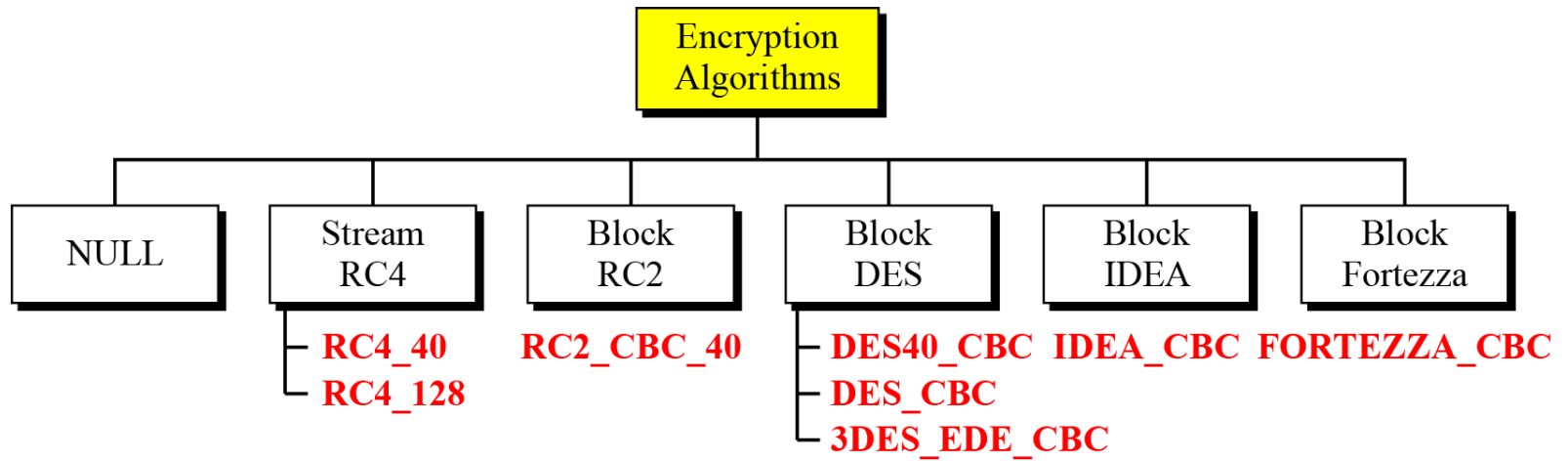
### ■ Fortezza

✗ Fortezza is a registered trademark of the U.S. National Security Agency (NSA). It is a family of security protocols developed for the Defense Department.

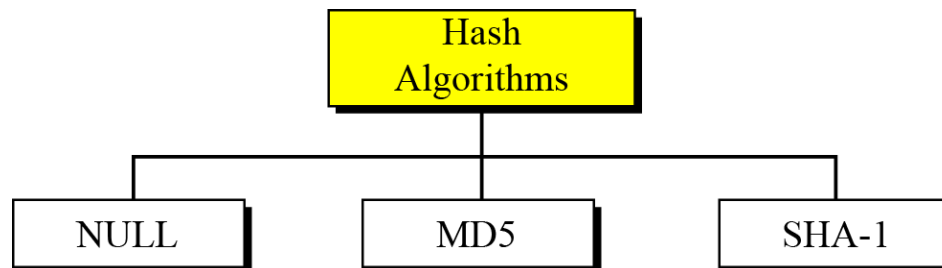
## 12.1.2 SSL 프로토콜 구조

---

### Encryption/decryption algorithms



### Hash algorithms



## 12.1.2 SSL 프로토콜 구조

---

### Cipher Suite

- The combination of key exchange, hash, and encryption algorithms defines a cipher suite for each SSL session. See Table 17.1

SSL\_DHE\_RSA\_WITH\_DES\_CBC\_SHA

✕ Ephemeral DH with RSA sig. cert., DES\_CBC, & SHA

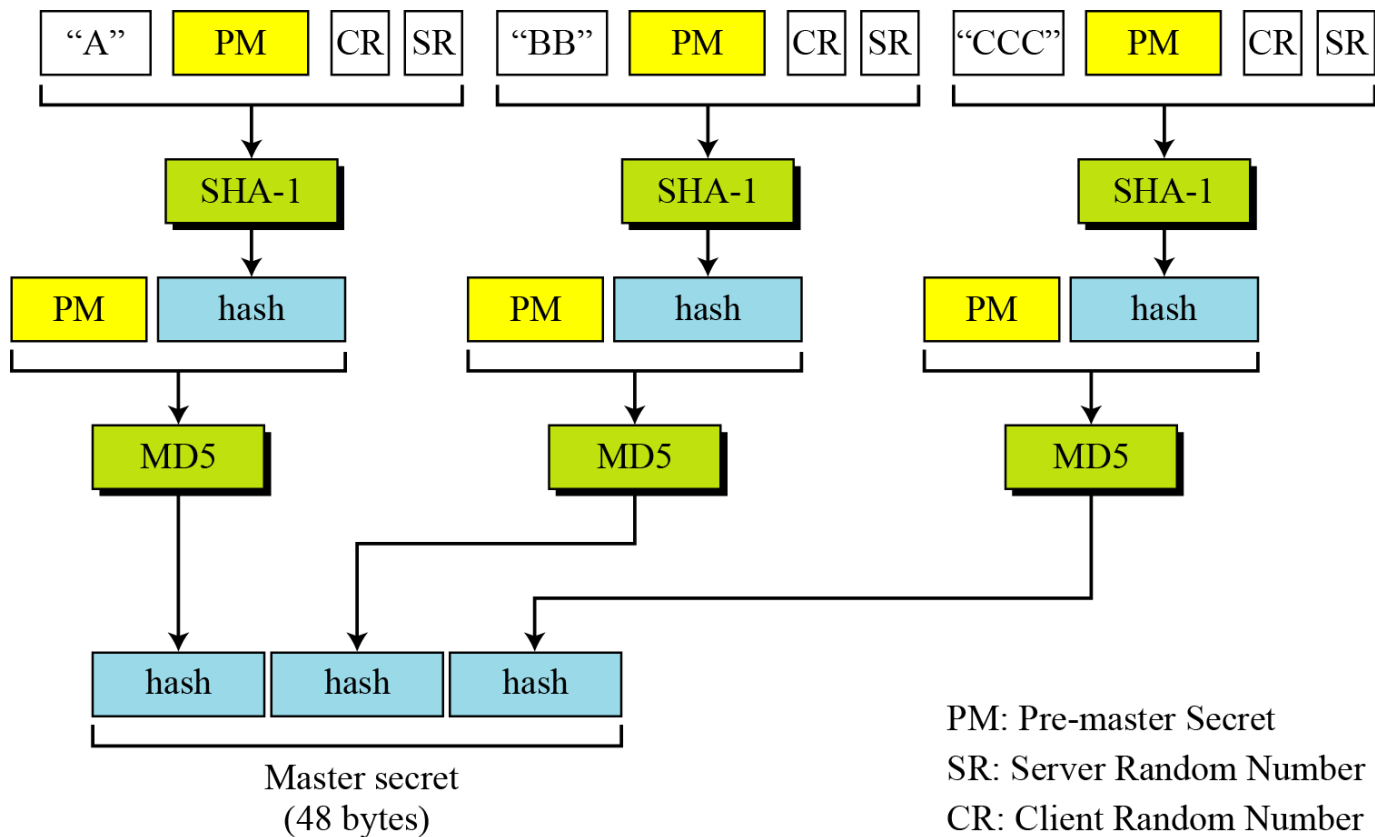
### ■ Compression Algorithms

✕ Compression is optional in SSLv3. No specific compression algorithm is defined for SSLv3. Therefore, the default compression method is NULL.

## 12.1.2 SSL 프로토콜 구조

### Cryptographic Parameter Generation

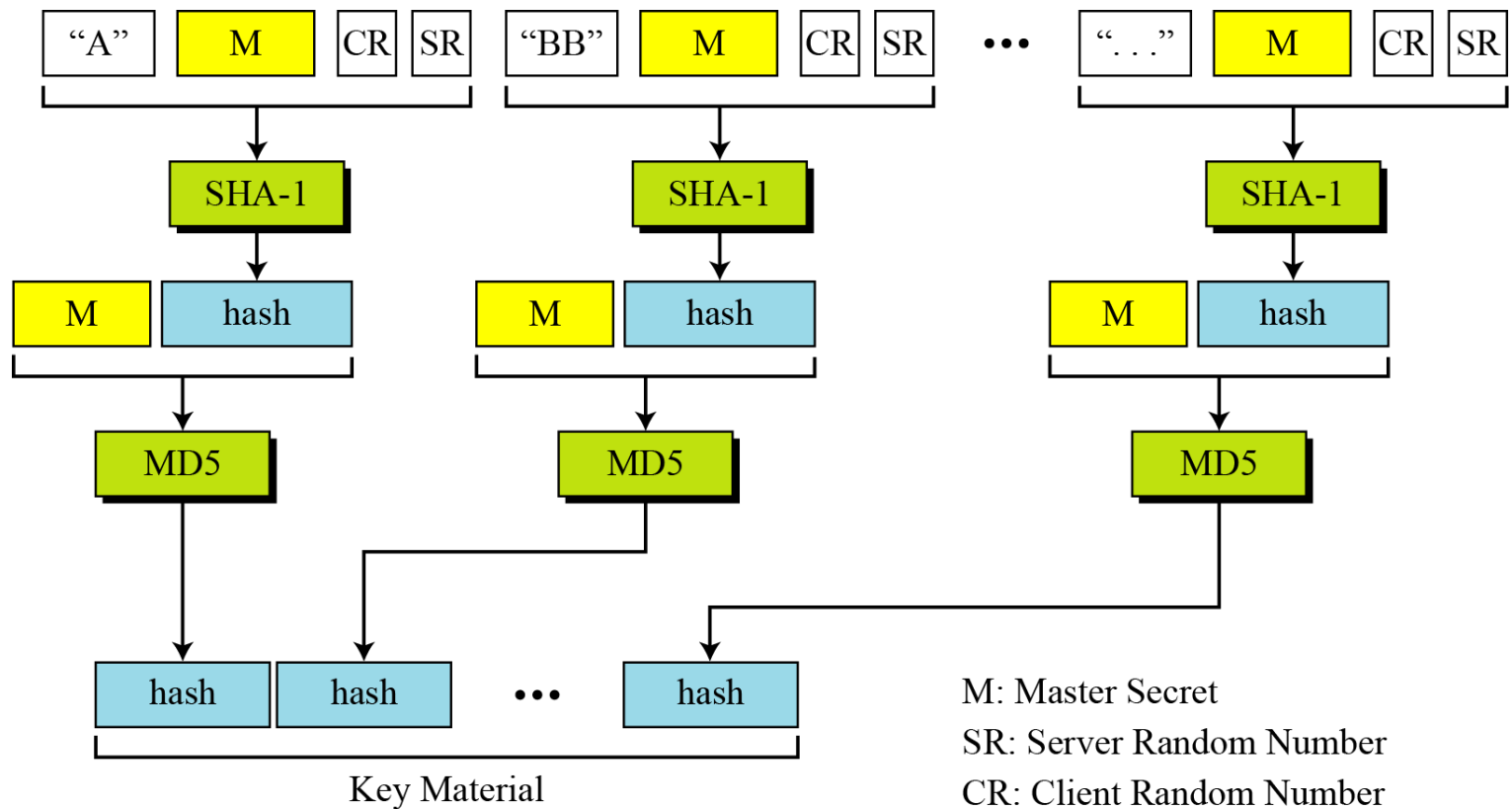
- Calculation of 48-byte master secret from pre-master secret



## 12.1.2 SSL 프로토콜 구조

### Cryptographic Parameter Generation

#### ■ Calculation of key material from master secret



## 12.1.2 SSL 프로토콜 구조

---

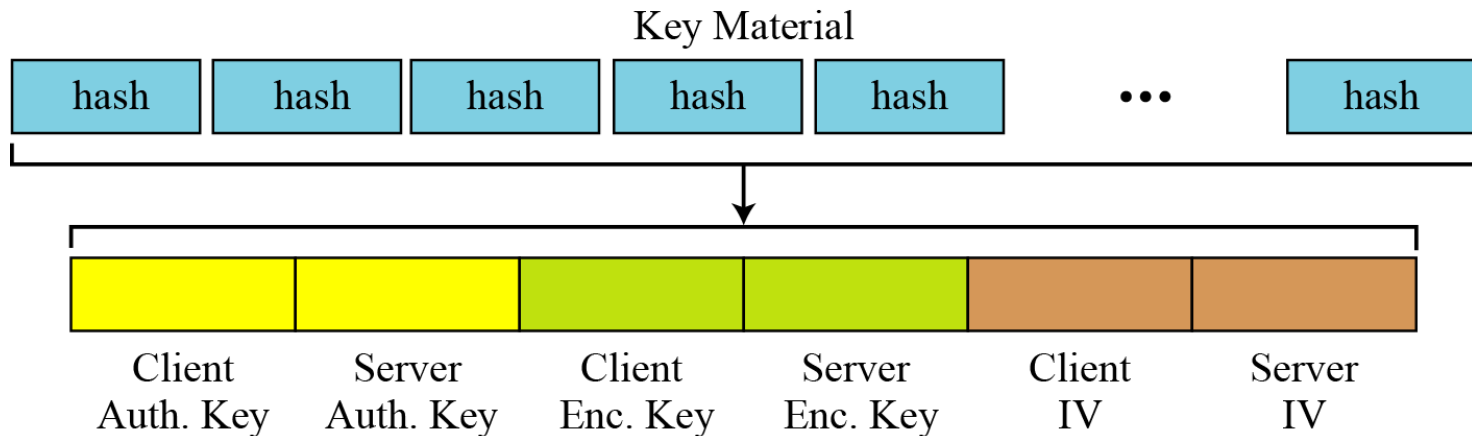
### Cryptographic Parameter Generation

- Extractions of cryptographic secrets from key material

Auth. Key: Authentication Key

Enc. Key: Encryption Key

IV: Initialization Vector

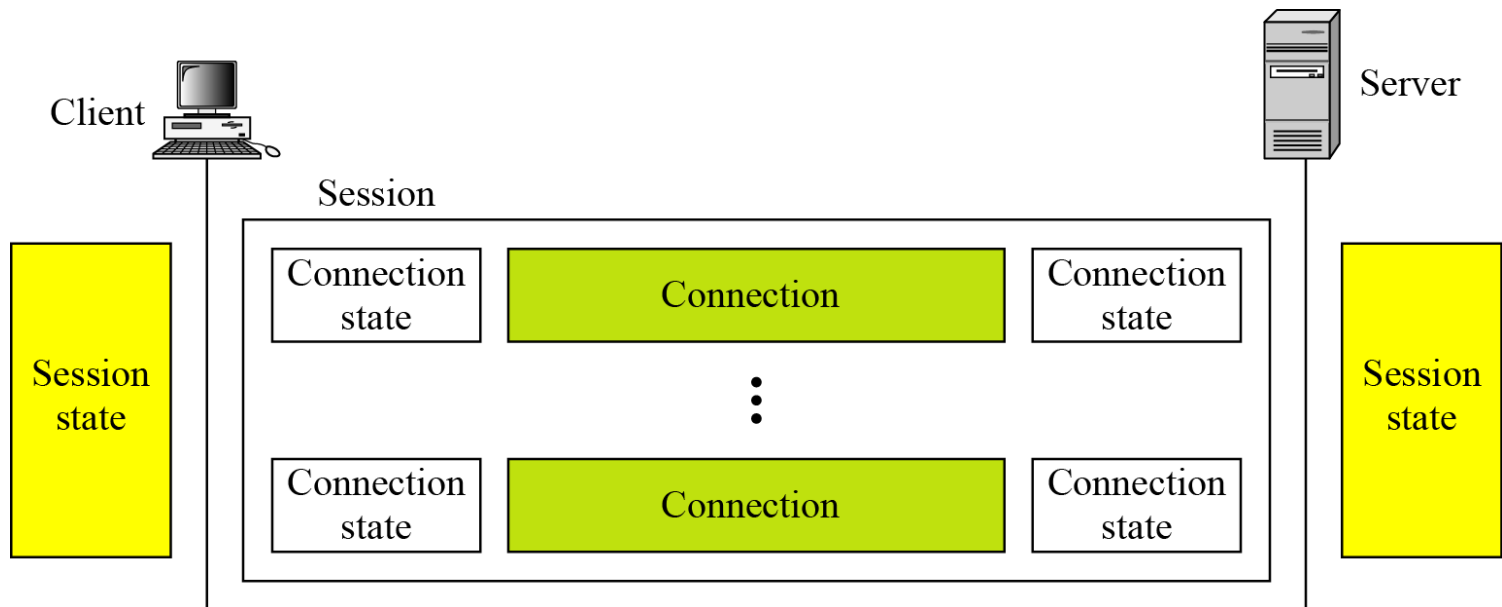


## 12.1.2 SSL 프로토콜 구조

---

### Sessions and Connections

- In a session, one party has the role of a client and the other the role of a server; in a connection, both parties have equal roles, they are peers.



## 12.1.2 SSL 프로토콜 구조

---

### Sessions and Connections

#### ■ Session state parameters

<i>Parameter</i>	<i>Description</i>
Session ID	A server-chosen 8-bit number defining a session.
Peer Certificate	A certificate of type X509.v3. This parameter may be empty (null).
Compression Method	The compression method.
Cipher Suite	The agreed-upon cipher suite.
Master Secret	The 48-byte secret.
Is resumable	A yes-no flag that allows new connections in an old session.

- ✕ Separation of a session from a connection : no negotiation necessary and saving in the high cost to generate a master secret.



## 12.1.2 SSL 프로토콜 구조

---

### Sessions and Connections

#### ■ Connection state parameters

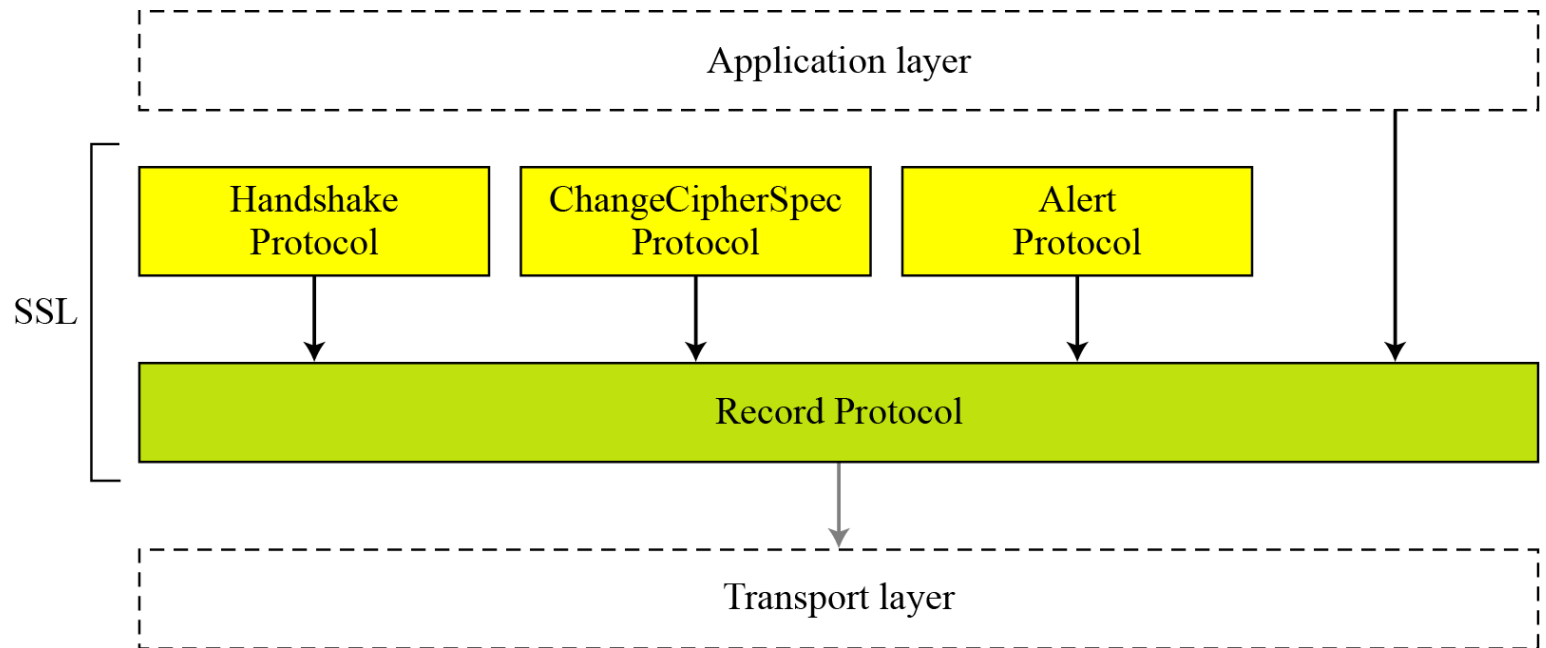
✕ The read secrets for the client are the same as the write secrets for the server and vice versa.

<i>Parameter</i>	<i>Description</i>
Server and client random numbers	A sequence of bytes chosen by the server and client for each connection.
Server write MAC secret = client read MAC secret	The outbound server MAC key for message integrity. The server uses it to sign; the client uses it to verify.
Client write MAC secret = server read MAC secret	The outbound client MAC key for message integrity. The client uses it to sign; the server uses it to verify.
Server write secret	The outbound server encryption key for message integrity.
Client write secret	The outbound client encryption key for message integrity.
Initialization vectors	The block ciphers in CBC mode use initialization vectors (IVs). One initialization vector is defined for each cipher key during the negotiation, which is used for the first block exchange. The final cipher text from a block is used as the IV for the next block.
Sequence numbers	Each party has a sequence number. The sequence number starts from 0 and increments. It must not exceed $2^{64} - 1$ .

## 12.1.2 SSL 프로토콜 구조

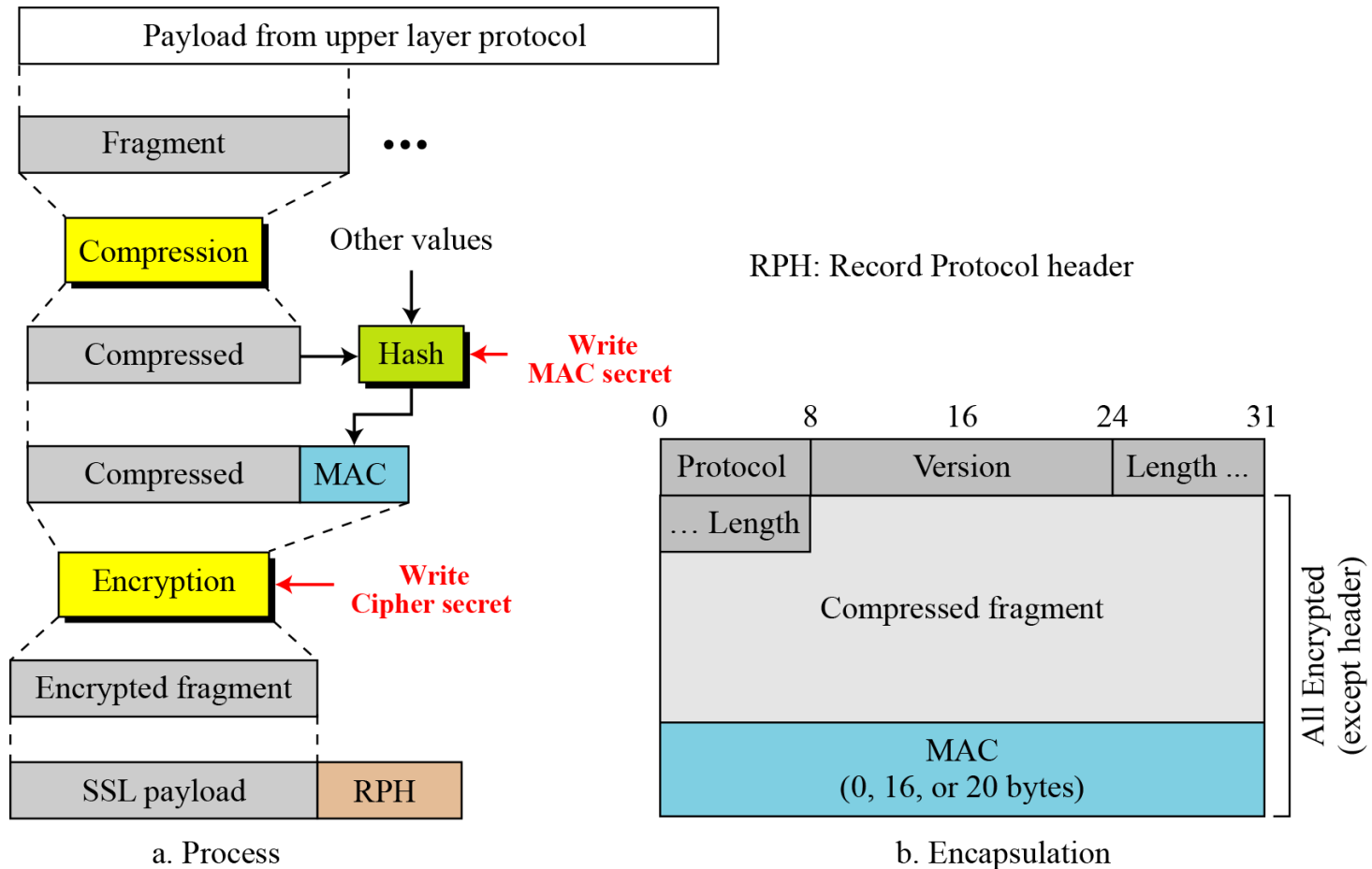
---

### ■ Four SSL protocols



## 12.1.2.1 Record 프로토콜

- Carries messages from the upper layer



## 12.1.2.1 Record 프로토콜

---

```
■ struct {  
    ContentType type;  
    ProtocolVersion version;  
    uint16 length;  
    opaque fragment[TLSPlaintext.length];  
} TLSPlaintext;
```

type

The higher level protocol used to process the enclosed fragment.

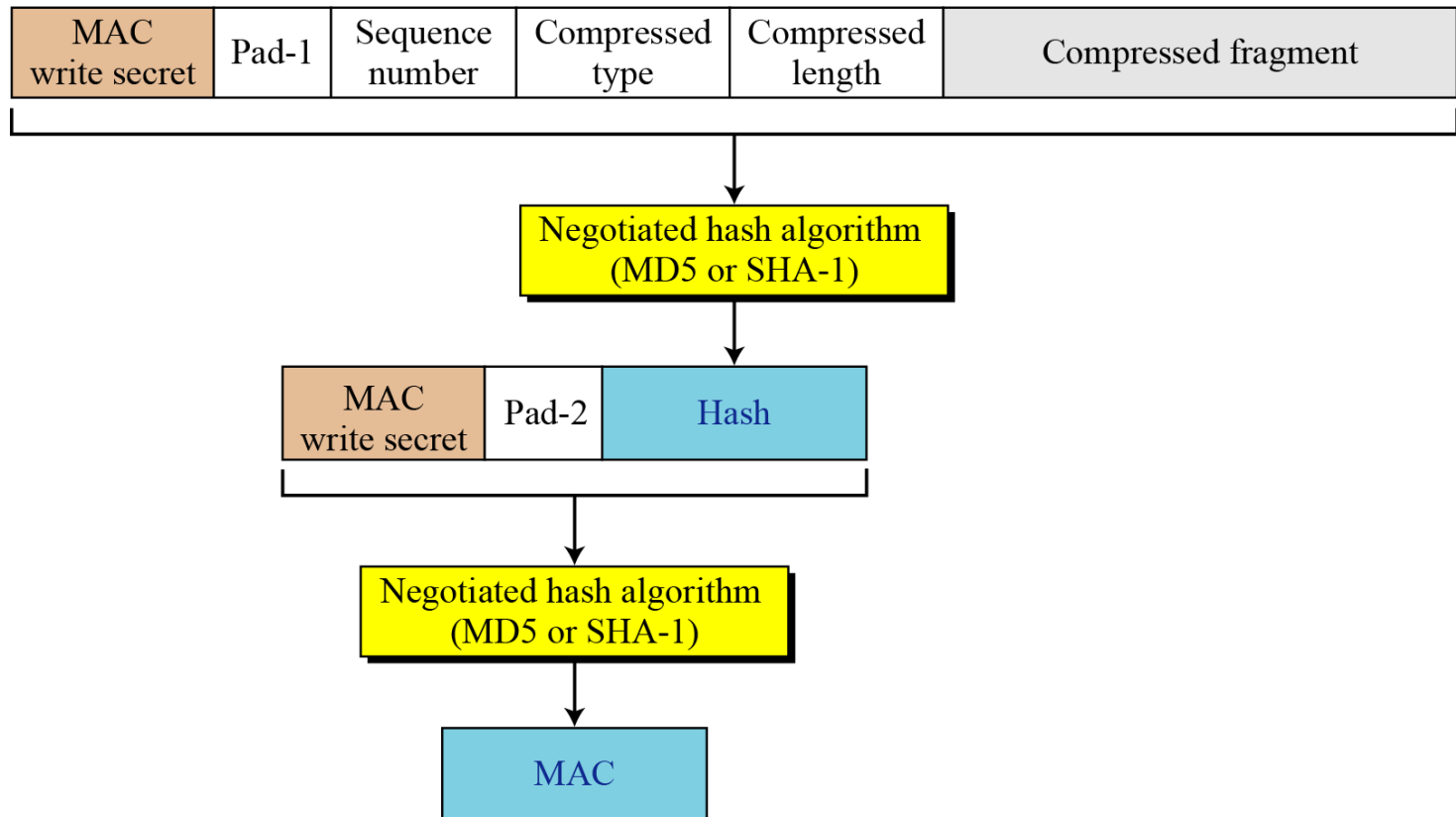
- 20 : ChangeCipherSpec
- 21 : Alert protocol
- 22 : Handshake protocol
- 23 : Application protocol data

## 12.1.2.1 Record 프로토콜

### ■ Calculation of MAC

Pad-1: Byte 0x36 (00110110) repeated 48 times for MD5 and 40 times for SHA-1

Pad-2: Byte 0x5C (01011100) repeated 48 times for MD5 and 40 times for SHA-1



## 12.1.2.2 Alert 프로토콜

---

### ■ Report errors and abnormal conditions

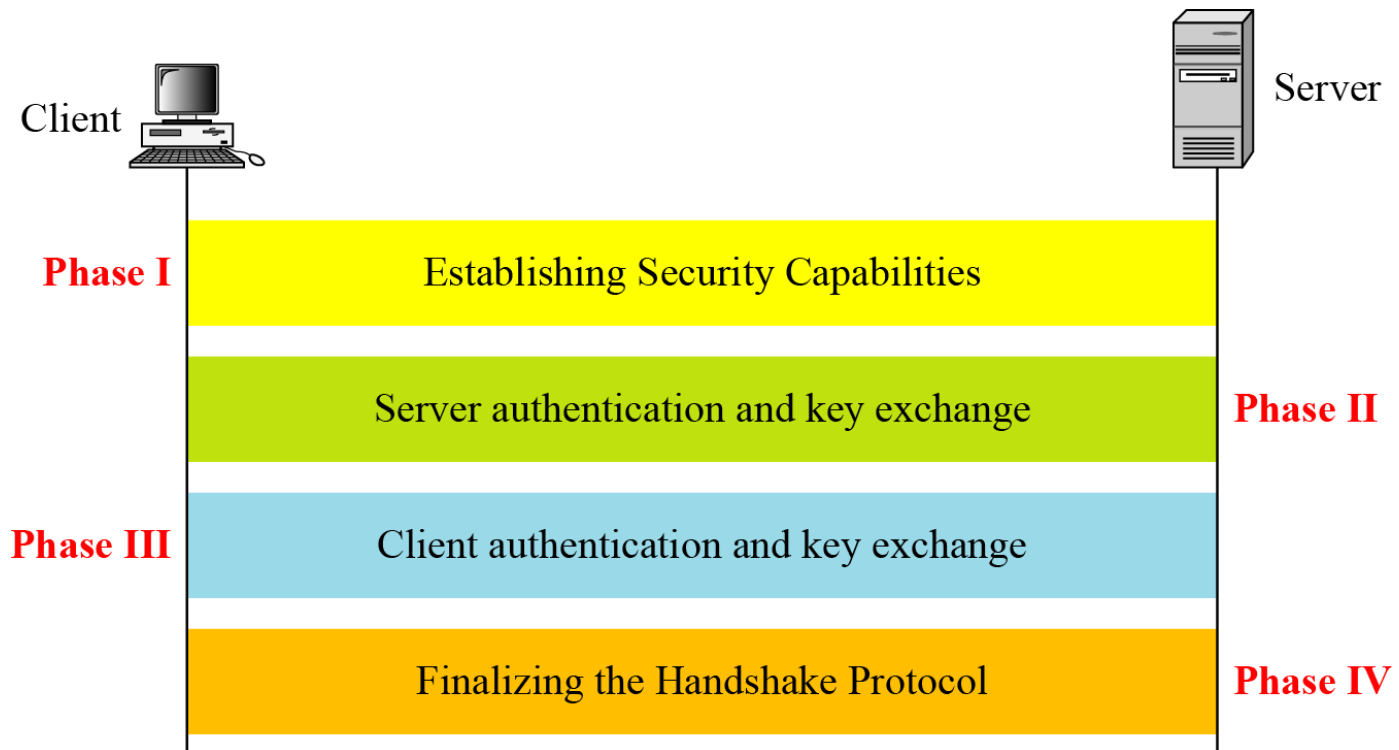
<i>Value</i>	<i>Description</i>	<i>Meaning</i>
0	<i>CloseNotify</i>	Sender will not send any more messages.
10	<i>UnexpectedMessage</i>	An inappropriate message received.
20	<i>BadRecordMAC</i>	An incorrect MAC received.
30	<i>DecompressionFailure</i>	Unable to decompress appropriately.
40	<i>HandshakeFailure</i>	Sender unable to finalize the handshake.
41	<i>NoCertificate</i>	Client has no certificate to send.
42	<i>BadCertificate</i>	Received certificate corrupted.
43	<i>UnsupportedCertificate</i>	Type of received certificate is not supported.
44	<i>CertificateRevoked</i>	Signer has revoked the certificate.
45	<i>CertificateExpired</i>	Certificate expired.
46	<i>CertificateUnknown</i>	Certificate unknown.
47	<i>IllegalParameter</i>	An out-of-range or inconsistent field.

## 12.1.2.3 Handshake 프로토콜

---

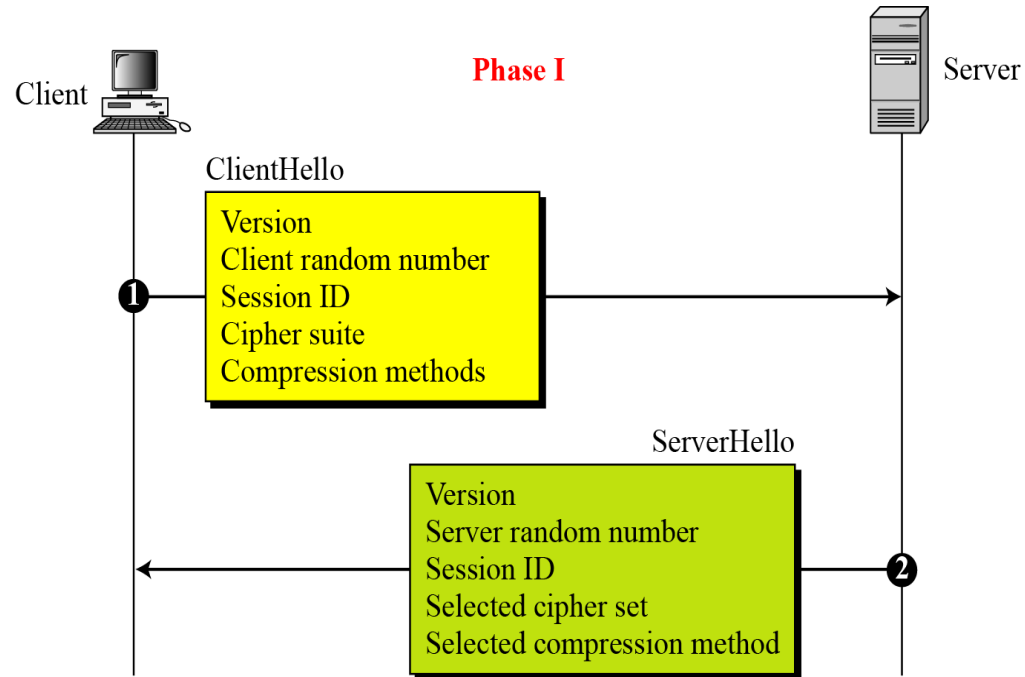
### ■ Handshake Protocol

- ✕ The negotiation of the cipher suite and the generation of cryptographic secrets



## 12.1.2.3 Handshake 프로토콜

### ■ Phase I



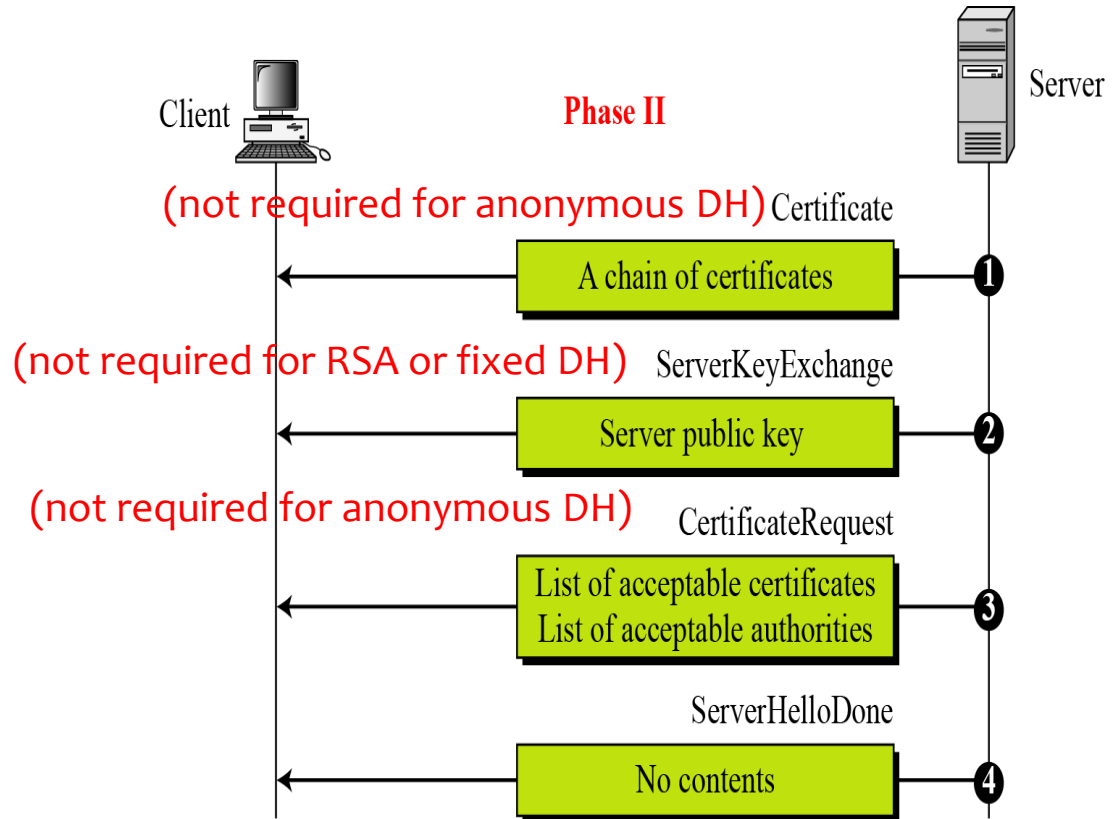
✕ After Phase I, the client and server know the following:

- ▶ The version of SSL
- ▶ The algorithms for key exchange, message authentication, and encryption
- ▶ The compression method
- ▶ The two random numbers for key generation



## 12.1.2.3 Handshake 프로토콜

### ■ Phase II

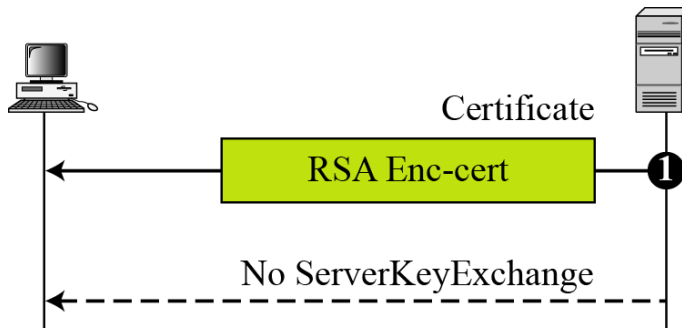


✕ After Phase II,

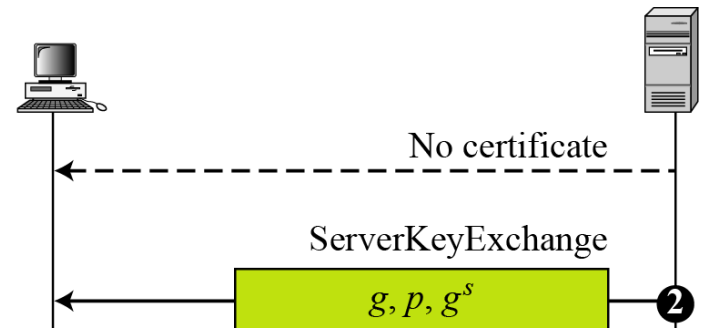
- ▶ The server is authenticated to the client.
- ▶ The client knows the public key of the server if required.

## 12.1.2.3 Handshake 프로토콜

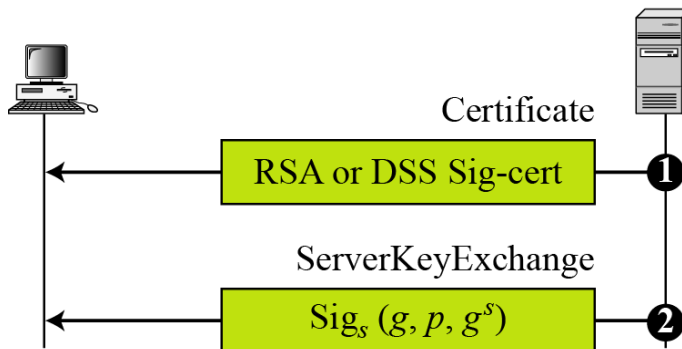
### ■ Four cases in Phase II



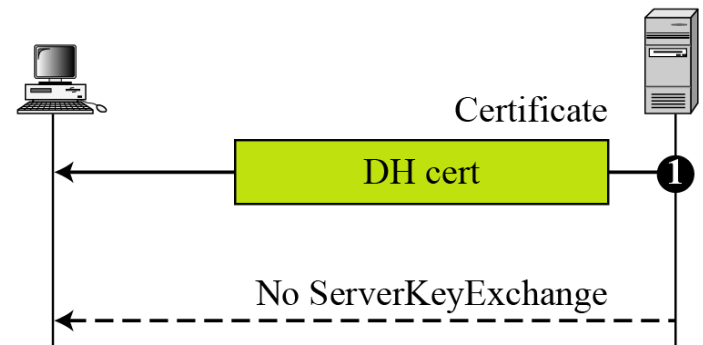
a. RSA



b. Anonymous DH



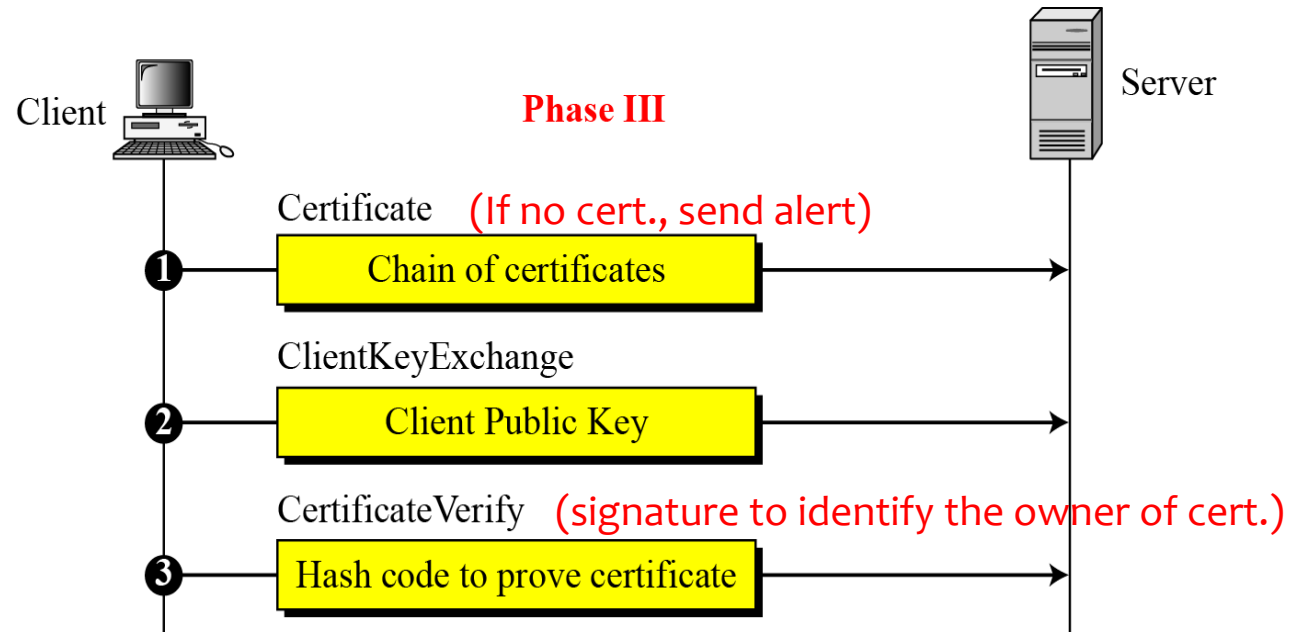
c. Ephemeral DH



d. Fixed DH

## 12.1.2.3 Handshake 프로토콜

### ■ Phase III



#### ✗ Certificate Verify


- ▶ The client verifies it owns the PK by signing M.
- ▶ Fixed DH does not work in this way.

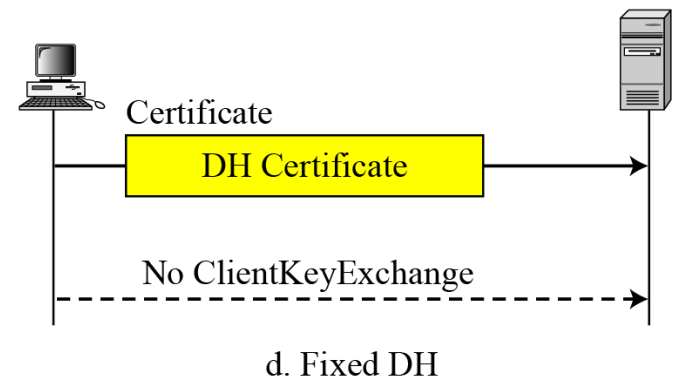
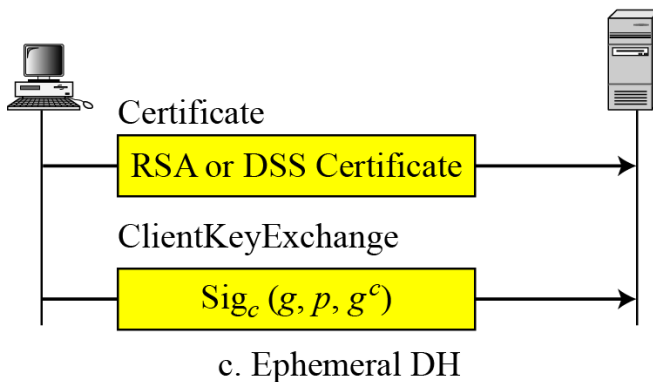
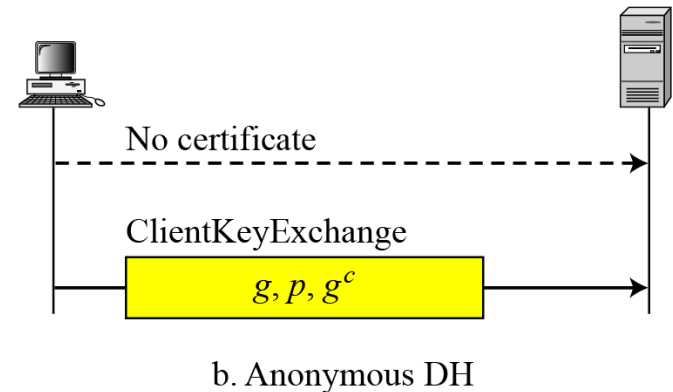
#### ✗ After Phase III,

- ▶ The client is authenticated for the server.
- ▶ Both the client and the server know the pre-master secret.

## 12.1.2.3 Handshake 프로토콜

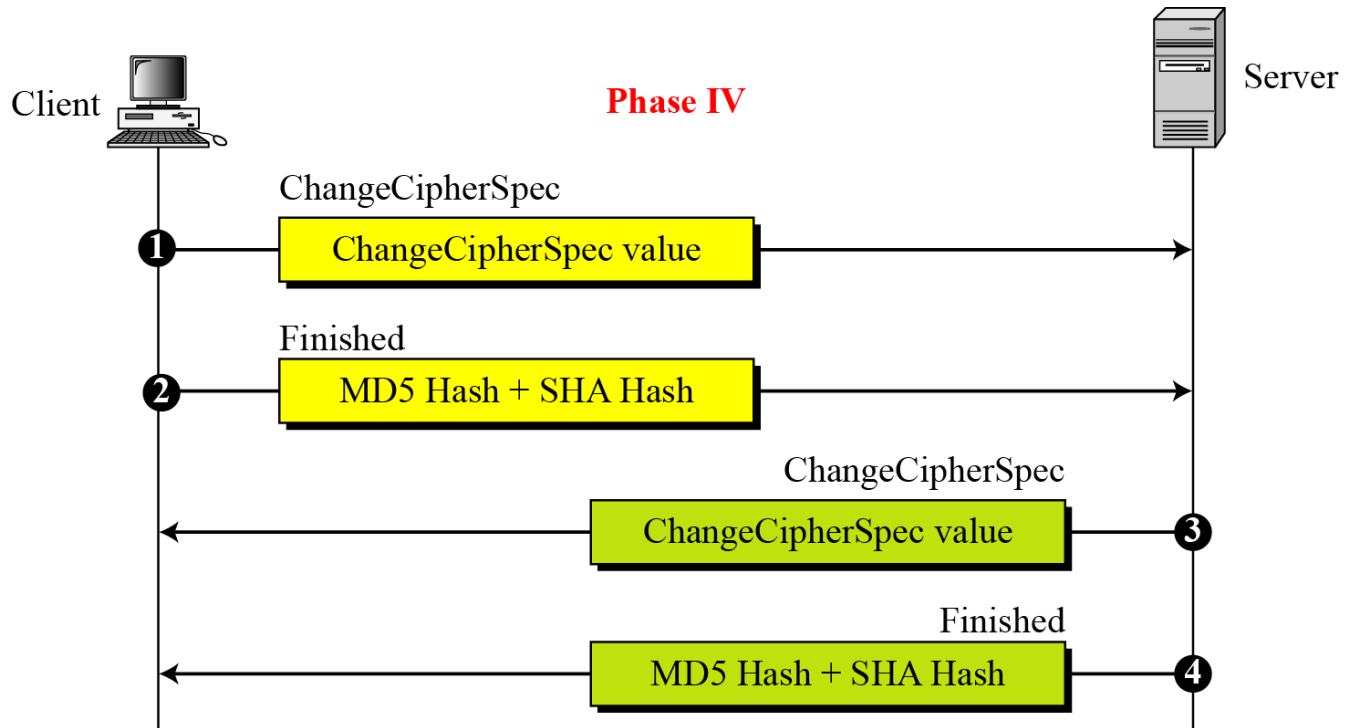
### ■ Four cases in Phase III

 S encrypted with server's public key  
Sig<sub>c</sub>: Signed with client's public key



## 12.1.2.3 Handshake 프로토콜

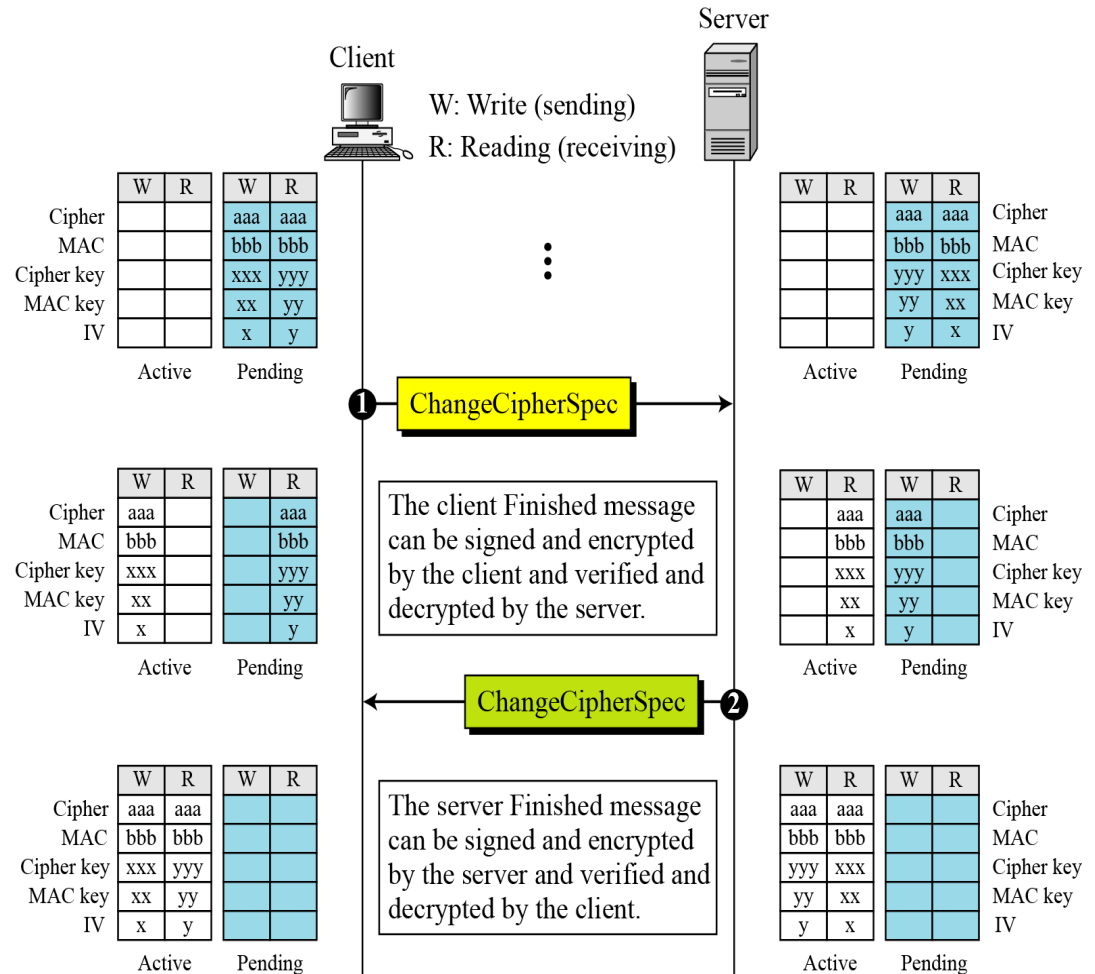
### ■ Phase IV



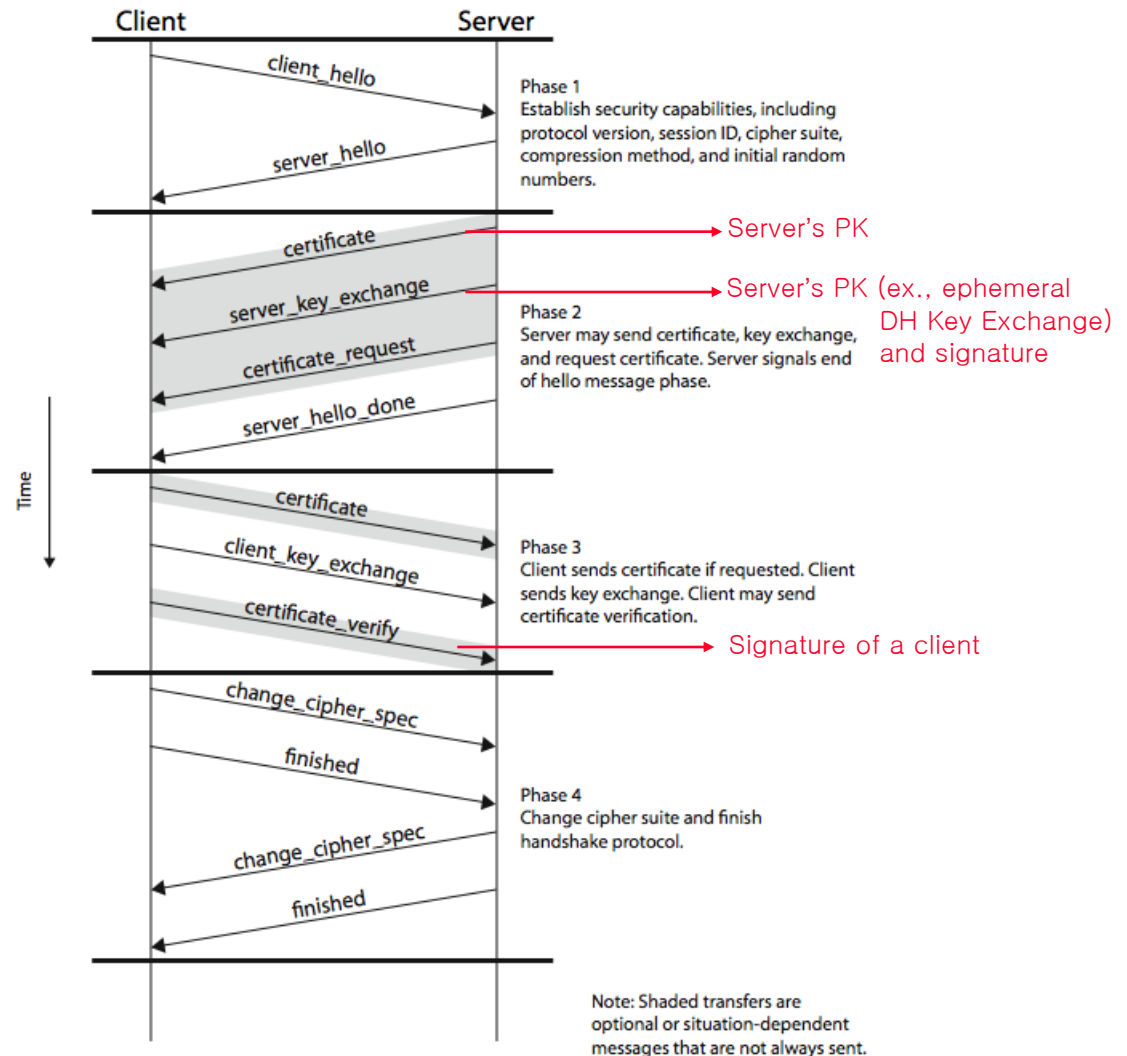
✕ After Phase IV, the client and server are ready to exchange data.

## 12.1.2.4 ChangeCipherSpec 프로토콜

- The parameters established in Handshake protocol can be used only after ChangeCipherSpec message.
- Movement of parameters from pending state to active state



## 12.1.2.3 Handshake 프로토콜 Overview



## 12.2 IPSec

---

### ■ Why IPSec?

1. Not all client/server programs are protected at the application layer : PGP only protects e-mail.
2. Not all client/server programs use TCP : wireless applications are using UDP.
3. Many programs, such as routing, directly use the services of IP.

### ■ IP security is a collection of protocols by IETF to provide security for a packet at the network layer (Internet Protocol or IP layer) : authenticated and confidential packets

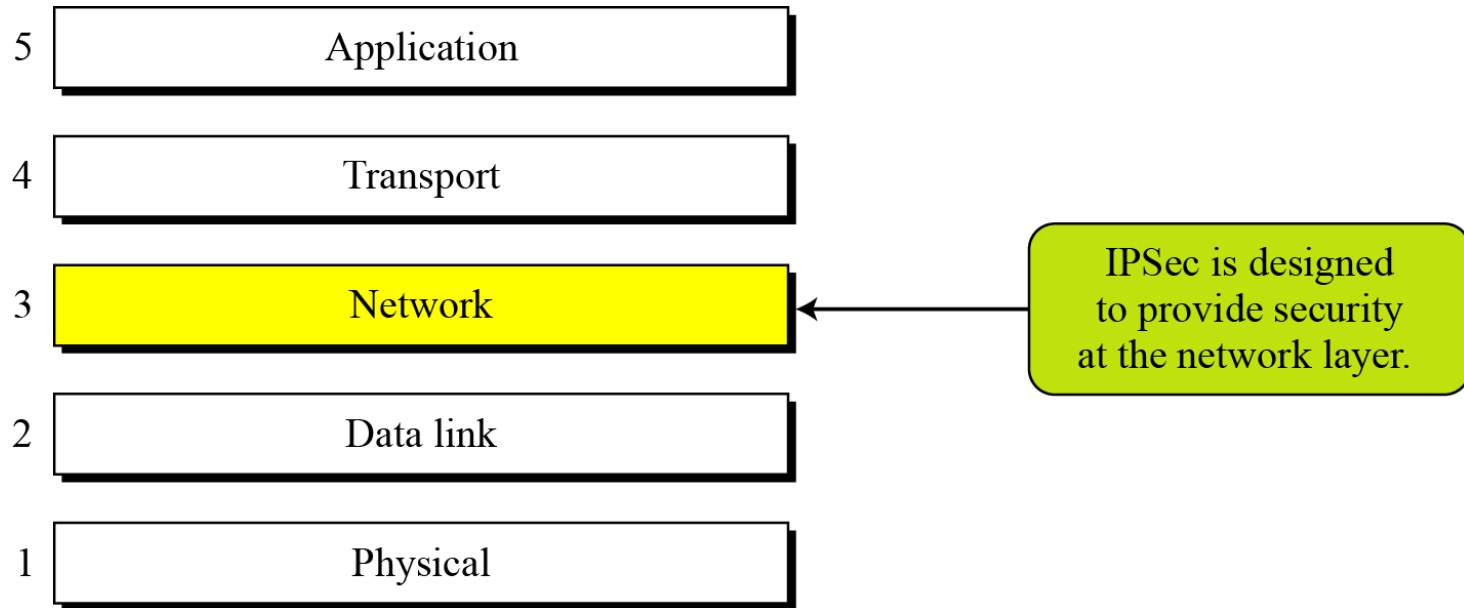
### ■ Useful for private TCP/IP network



## 12.2.1 IPSec 소개

---

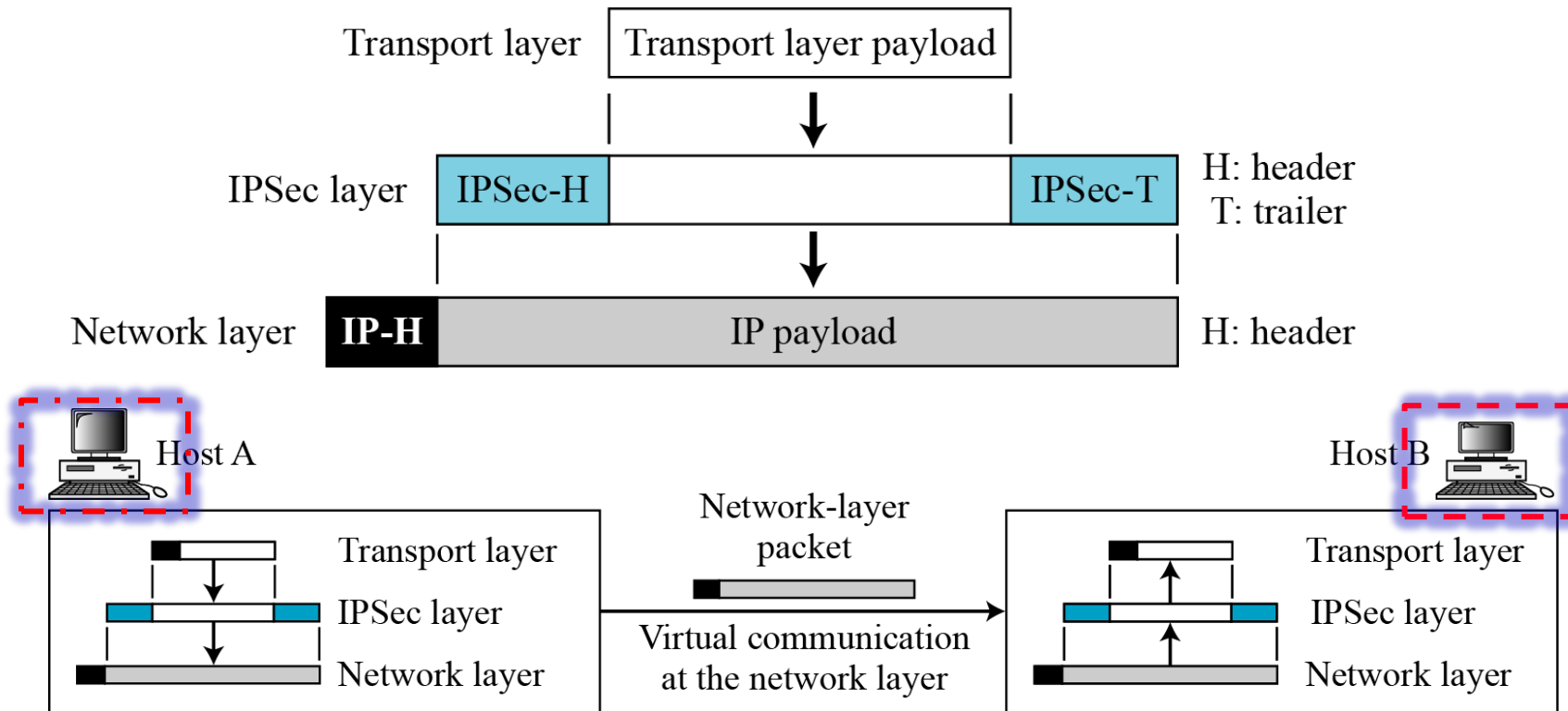
### ■ TCP/IP Protocol Suite and IPSec



## 12.2.1 IPSec 소개- Transport and tunnel modes

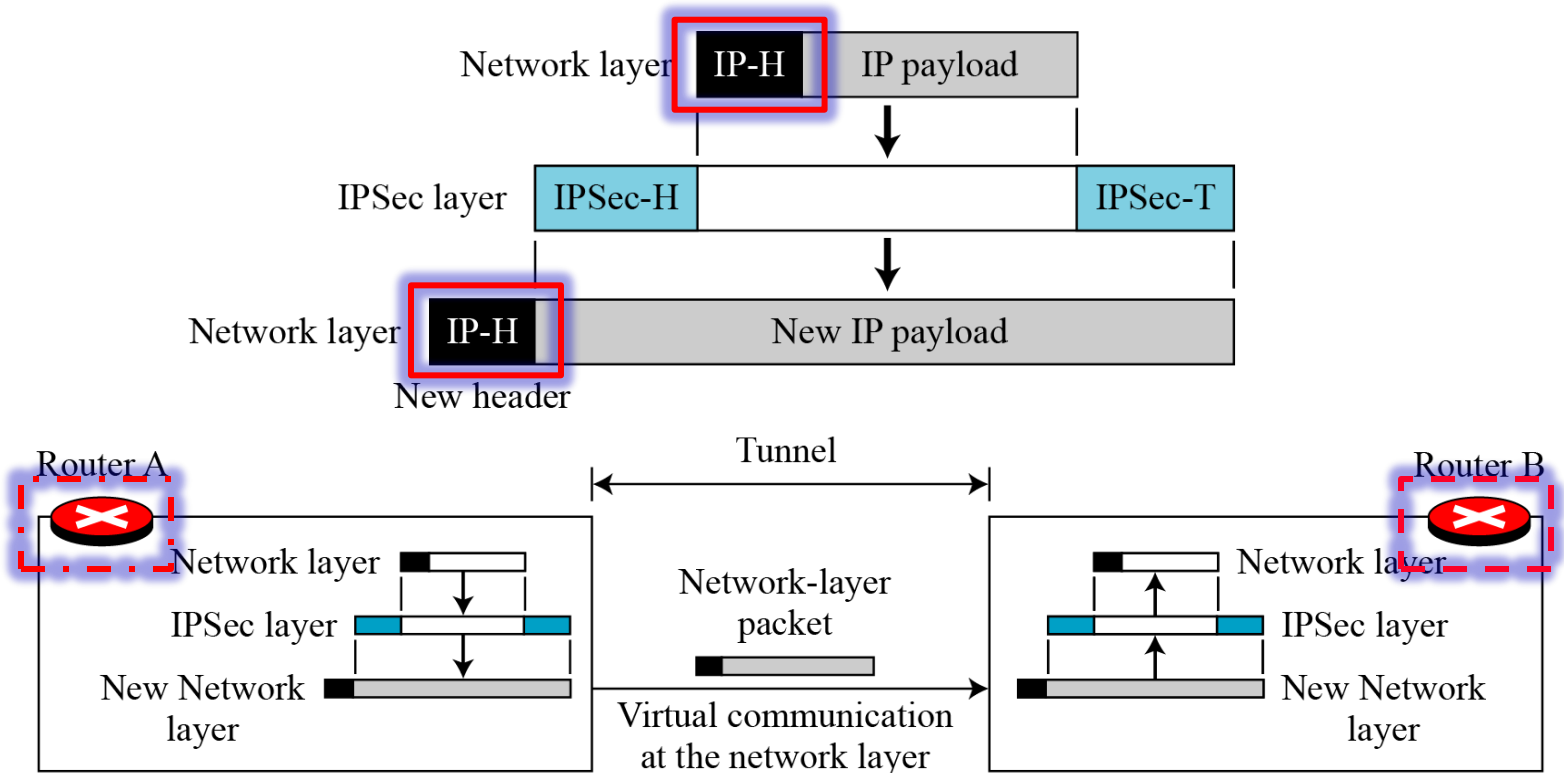
- In **transport mode**, IPSec protects what is delivered from the transport layer to the network layer.

✕ IPSec in transport mode does not protect the IP header; it only protects the information coming from the transport layer.



## 12.2.1 IPSec 소개- Transport and tunnel modes

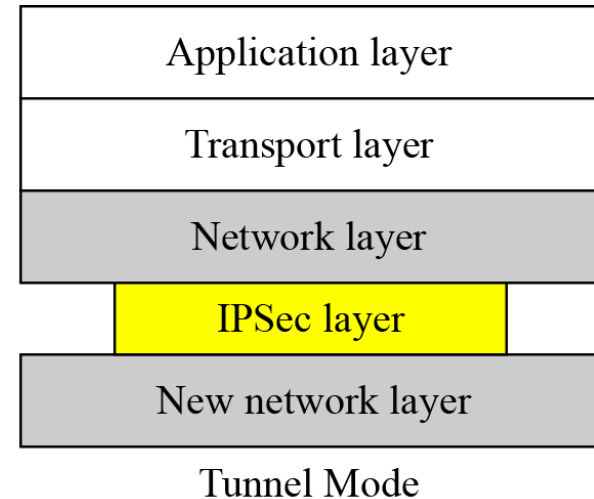
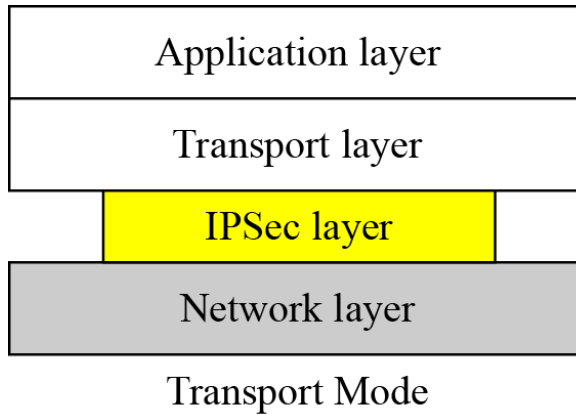
- In **tunnel mode**, IPSec protects the entire IP packet. It takes an IP packet, including the header, applies IPSec security methods to the entire packet, and then adds a new IP header.



## 12.2.1 IPSec 소개- Transport and tunnel modes

---

### ■ Transport mode versus tunnel mode



## 12.3 AH(Authentication Header) 프로토콜

---

- IPSec defines two protocols—the Authentication Header (AH) Protocol and the Encapsulating Security Payload (ESP) Protocol to provide authentication and/or encryption for packets at the IP level.
- IP Header :

IP Header Field || Source IP || Destination IP

- Attacker : spoofing, sniffing, session hijacking
- IPSec is to add cryptographic protection to IP Header

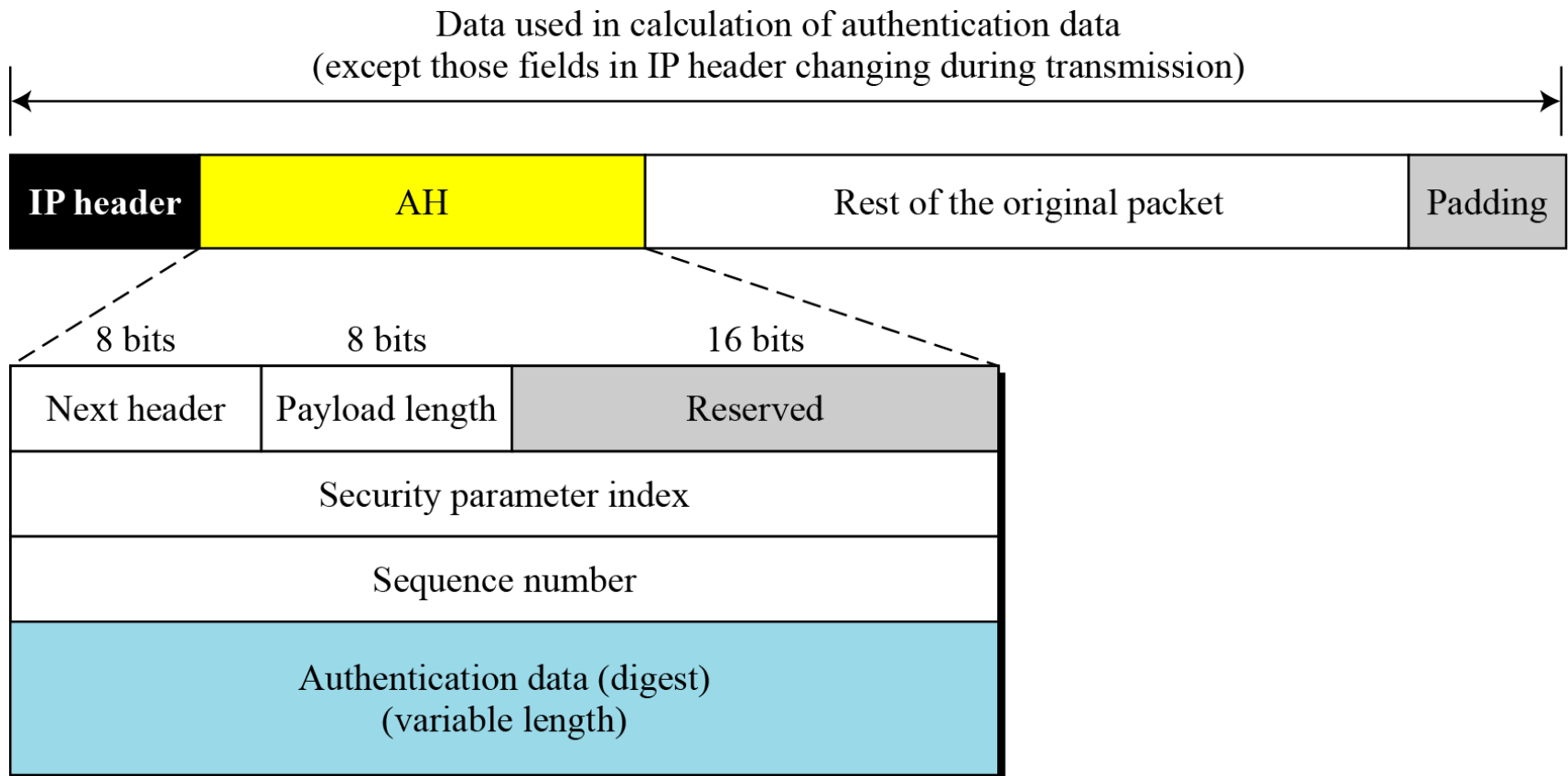
## 12.3 AH(Authentication Header) 프로토콜

---

- The AH protocol provides source authentication and data integrity, but not privacy.
- IP Header || AH
- **Next header** ← protocol field in IP header that defines type of payload (TCP, UDP, etc.)
- **Payload length** : the length of the **AH**
- **Security Parameters Index** (SPI)
  - ✗ Specifies the cryptographic algo. used for the auth.
- **Authentication Data** or Integrity Check Value (ICV)
  - ✗ Hashed value of the entire IP datagram

# 12.3 AH(Authentication Header) 프로토콜

## ■ The AH protocol



# 12.4 ESP(Encapsulating Security Payload)

## 프로토콜

---

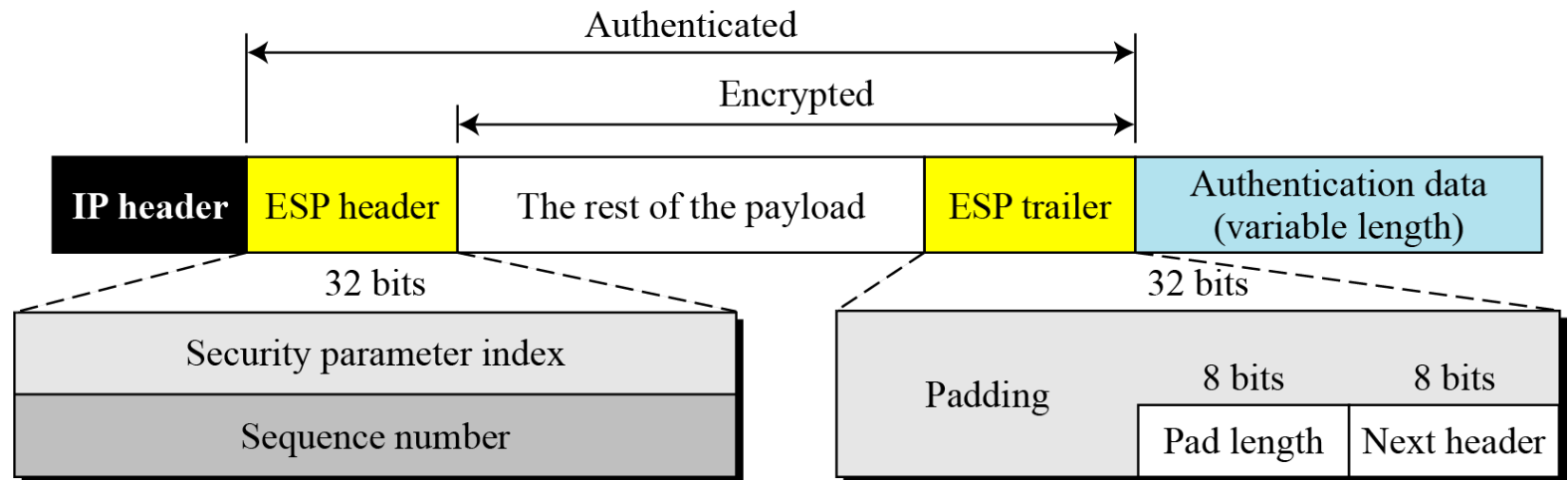
- ESP provides source authentication, data integrity, and privacy.
- Optional service for IPSec
- Achieved by Encapsulating Security Payload (ESP)
- SPI specifies encryption algo.
- Payload data is the ciphertext of the confidential data



# 12.4 ESP(Encapsulating Security Payload)

## 프로토콜

- ESP provides source authentication, data integrity, and privacy.



## 18.2.3 IPv4 and IPv6

---

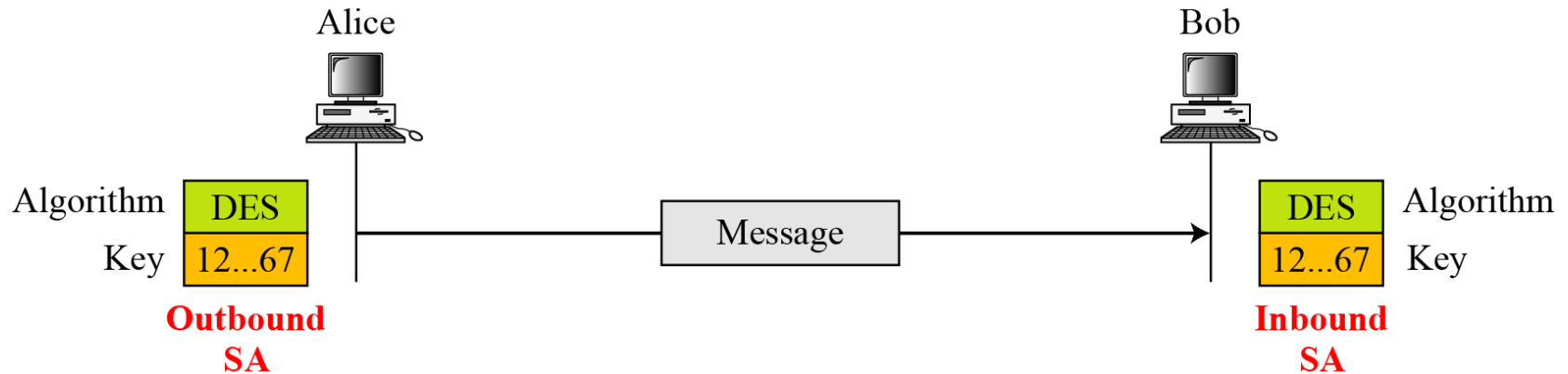
- IPSec supports both IPv4 and IPv6. In IPv6, however, AH and ESP are part of the extension header.
- The ESP protocol was designed after the AH protocol was already in use. ESP does whatever AH does with additional functionality (privacy).

<i>Services</i>	<i>AH</i>	<i>ESP</i>
Access control	yes	yes
Message authentication (message integrity)	yes	yes
Entity authentication (data source authentication)	yes	yes
Confidentiality	<b>no</b>	yes
Replay attack protection	yes	yes

## 12.5 SA(Security Association)

---

- Security Association is a very important aspect of IPSec. IPSec requires a logical relationship, called a Security Association (SA), between two hosts



# 12.5 SA(Security Association)

---

## ■ Typical SA Parameters

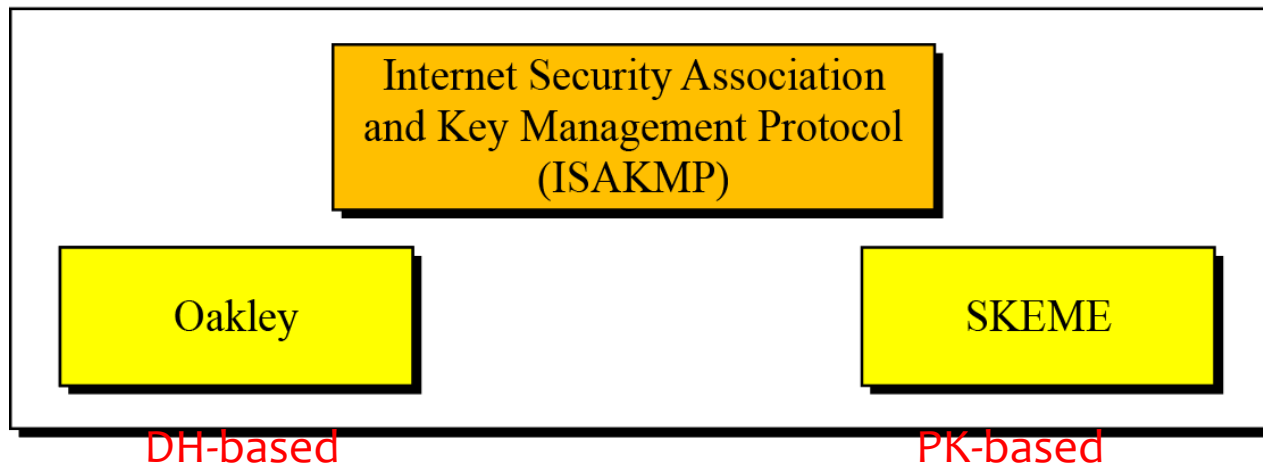
Sequence Number Counter	This is a 32-bit value that is used to generate sequence numbers for the AH or ESP header.
Sequence Number Overflow	This is a flag that defines a station's options in the event of a sequence number overflow.
Anti-Replay Window	This detects an inbound replayed AH or ESP packet.
AH Information	This section contains information for the AH protocol: 1. Authentication algorithm 2. Keys 3. Key lifetime 4. Other related parameters
ESP Information	This section contains information for the ESP protocol: 1. Encryption algorithm 2. Authentication algorithm 3. Keys 4. Key lifetime 5. Initiator vectors 6. Other related parameters
SA Lifetime	This defines the lifetime for the SA.
IPSec Mode	This defines the mode, transport or tunnel.
Path MTU	This defines the path MTU (fragmentation).

## 12.6.1 IKE(Internet Key Exchange)

---

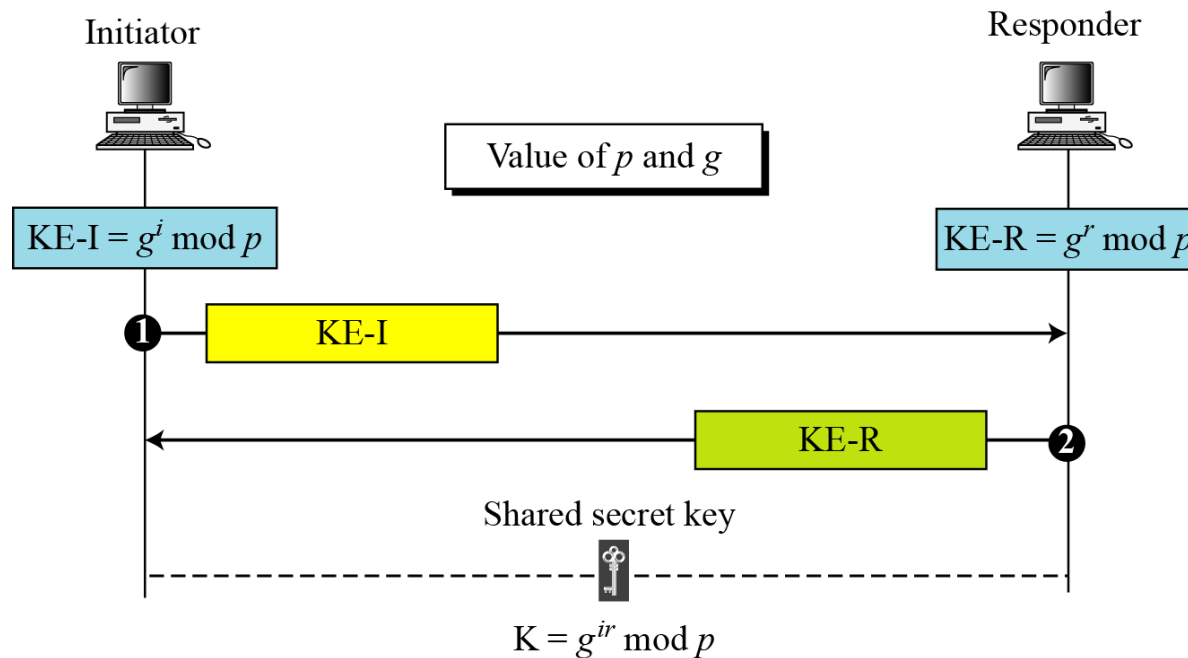
- The Internet Key Exchange (IKE) is a protocol designed to create both inbound and outbound Security Associations.
- ✗ IKE creates SAs for IPSec.
- IKE components

Internet Key Exchange (IKE)



# 12.6.1 IKE(Internet Key Exchange)- Improved Diffie-Hellman

## ■ Diffie-Hellman key exchange



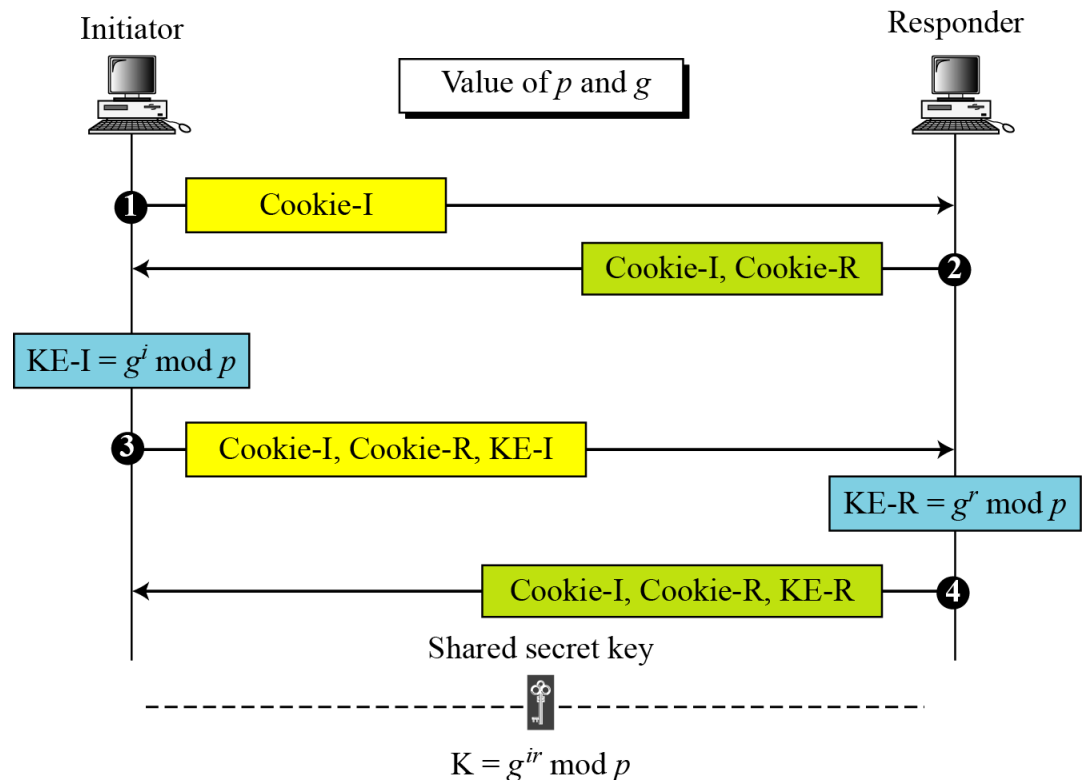
## ■ Clogging Attack (or DOS attack)

- ✗ Eve sends many half-keys to Bob.
- ✗ To protect against a clogging attack, IKE uses cookies.

# 12.6.1 IKE(Internet Key Exchange)- Improved Diffie-Hellman

## Diffie-Hellman with cookies

✗ The cookie here is the hash of ID of the peer, a random known to the party that generates the cookie, and a timestamp.



- To protect against a replay attack, IKE uses nonces.
- To protect against man-in-the-middle attack, IKE requires that each party shows that it possesses a secret.

## 12.6.1 IKE(Internet Key Exchange)- IKE Phases

---

- IKE is divided into two phases: phase I and phase II. Phase I creates SAs for phase II; phase II creates SAs for a data exchange protocol such as IPSec..

