Block III. Internet security

Protocols: SET, SSL and IPSec

Network Security

Maria Dolores Cano Banos

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

- 4.2.1 Participants
- 4.2.2 Services
- 4.2.3 Sequence of actions
- 4.2.4 Dual Signature
- 4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

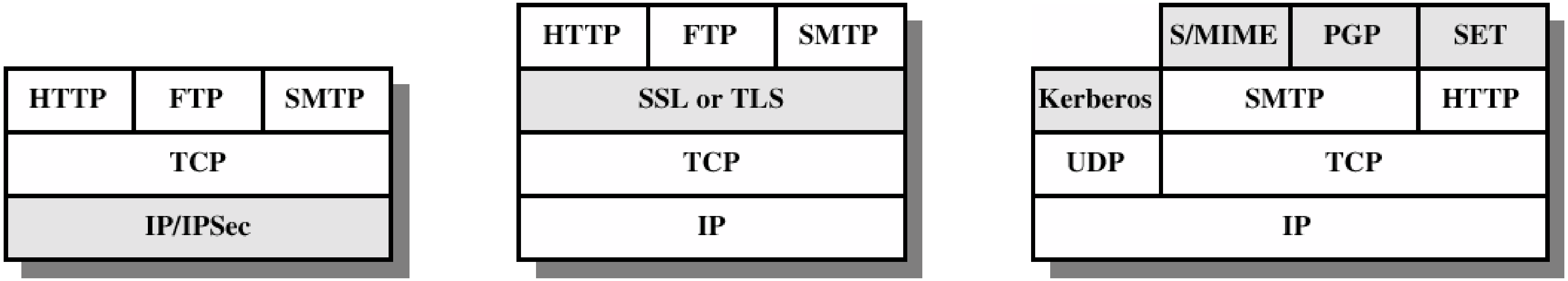
- 4.3.1 Architecture
- 4.3.2 Sessions and connections
- 4.3.3 Record Protocol
- 4.3.4 Change Cipher Spec Protocol
- 4.3.5 Alert Protocol
- 4.3.6 Handshake Protocol
- 4.3.7 Cryptographic calculations
- 4.3.8 Additional considerations

4.4 IPSec

- 4.4.1 IPSec bound protocols
- 4.4.2 Security Associations
- 4.4.3 IPSec protocols
- 4.4.4 Inter-entity authentication and security association formation

Introduction

Security tools in the TCP / IP protocol stack



HTTP | FTP | SMTP
TCP
IP/IPSec

(a) Network Level

HTTP | FTP | SMTP
SSL or TLS
TCP
IP

(b) Transport Level

S/MIME | PGP | SET
Kerberos | SMTP | HTTP
UDP | TCP
IP

(c) Application Level

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

- 4.2.1 Participants
- 4.2.2 Services
- 4.2.3 Sequence of actions
- 4.2.4 Dual Signature
- 4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

- 4.3.1 Architecture
- 4.3.2 Sessions and connections
- 4.3.3 Record Protocol
- 4.3.4 Change Cipher Spec Protocol
- 4.3.5 Alert Protocol
- 4.3.6 Handshake Protocol
- 4.3.7 Cryptographic calculations
- 4.3.8 Additional considerations

4.4 IPSec

- 4.4.1 IPSec bound protocols
- 4.4.2 Security Associations
- 4.4.3 IPSec protocols
- 4.4.4 Inter-entity authentication and security association formation

Secure Electronic Transactions

NOT IN THE EXAM

- Is a specification open of encryption and security (nineteen ninety five).
- Protects transactions with cards of credit in Internet.
- Business involved:
  - MasterCard, Visa, IBM, Microsoft, Netscape, RSA, Terisa and Verisign.
- No is a system of payment.
- Is a set of protocols of security and formats.

Contents

NOT IN THE EXAM

4.1 introduction

4.2 Secure Electronic Transactions (SET)

4.2.1 Participants
4.2.2 Services
4.2.3 Sequence of actions
4.2.4 Dual Signature
4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

4.3.1 Architecture
4.3.2 Sessions and connections
4.3.3 Record Protocol
4.3.4 Change Cipher Spec Protocol
4.3.5 Alert Protocol
4.3.6 Handshake Protocol
4.3.7 Cryptographic calculations
4.3.8 Additional considerations

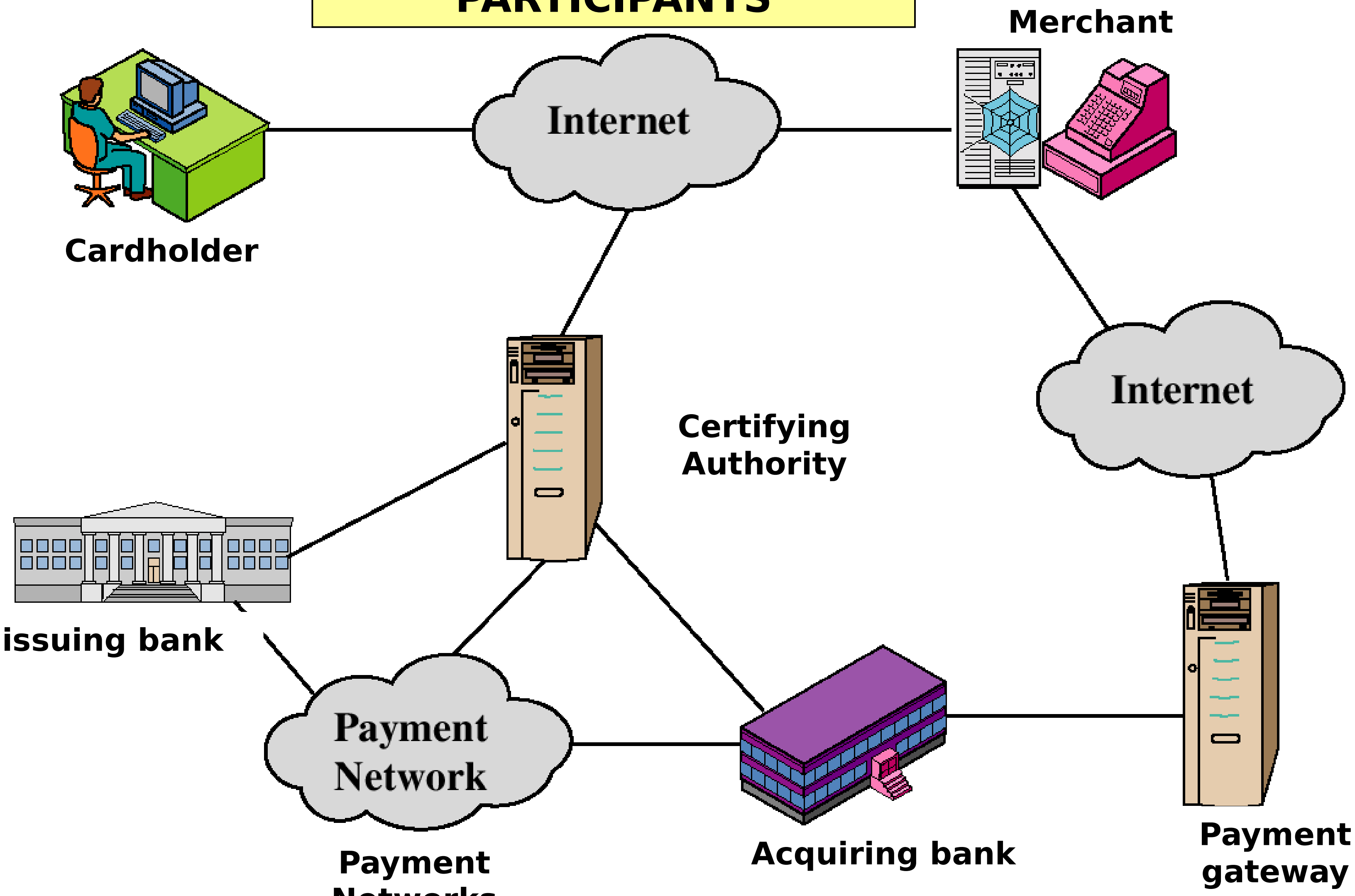
4.4 IPSec

4.4.1 IPSec bound protocols
4.4.2 Security Associations
4.4.3 IPSec protocols
4.4.4 Inter-entity authentication and security association formation

Secure Electronic Transactions

NOT IN THE EXAM

PARTICIPANTS



NOT IN THE EXAM

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

- 4.2.1 Participants
- 4.2.2 Services
- 4.2.3 Sequence of actions
- 4.2.4 Dual Signature
- 4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

- 4.3.1 Architecture
- 4.3.2 Sessions and connections
- 4.3.3 Record Protocol
- 4.3.4 Change Cipher Spec Protocol
- 4.3.5 Alert Protocol
- 4.3.6 Handshake Protocol
- 4.3.7 Cryptographic calculations
- 4.3.8 Additional considerations

4.4 IPSec

- 4.4.1 IPSec bound protocols
- 4.4.2 Security Associations
- 4.4.3 IPSec protocols
- 4.4.4 Inter-entity authentication and security association formation

Secure Electronic Transactions

NOT IN THE EXAM

SERVICES

- Authentication, X.509 v3 digital certificates
- Confidentiality, encrypted payment information
- Integrity, use of digital signature
- Payment management
- Privacy
- Immediate verification

NOT IN THE EXAM

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

- 4.2.1 Participants
- 4.2.2 Services
- 4.2.3 Sequence of actions
- 4.2.4 Dual Signature
- 4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

- 4.3.1 Architecture
- 4.3.2 Sessions and connections
- 4.3.3 Record Protocol
- 4.3.4 Change Cipher Spec Protocol
- 4.3.5 Alert Protocol
- 4.3.6 Handshake Protocol
- 4.3.7 Cryptographic calculations
- 4.3.8 Additional considerations

4.4 IPSec

- 4.4.1 IPSec bound protocols
- 4.4.2 Security Associations
- 4.4.3 IPSec protocols
- 4.4.4 Inter-entity authentication and security association formation

Secure Electronic Transactions

NOT IN THE EXAM

CONVENTIONAL SEQUENCE OF ACTIONS

1. Holder presents card to merchant
2. Merchant for card by Point of Sale Terminal (POS)
3. Transaction data through the payment network system to the issuing bank
4. Issuing bank verifies data and sends approval
5. The acquiring bank receives information, as does the POS that issues receipt
6. Merchant has money entered
7. The customer is deducted from their checking account

Secure Electronic Transactions

NOT IN THE EXAM

SEQUENCE OF ACTIONS (I)

1. Buyer opens an account and gets a VISA or MasterCard valid for SET
2. Buyer receives X.509 v3 digital certificate signed by bank. Seller must have two (signature and key exchange)
3. The customer decides to buy over the Internet (receives transaction identifier)
4. Customer checks order and sends purchase order, payment information and certificate -> SET starts
5. Merchant sends payment request to his bank

Secure Electronic Transactions

NOT IN THE EXAM

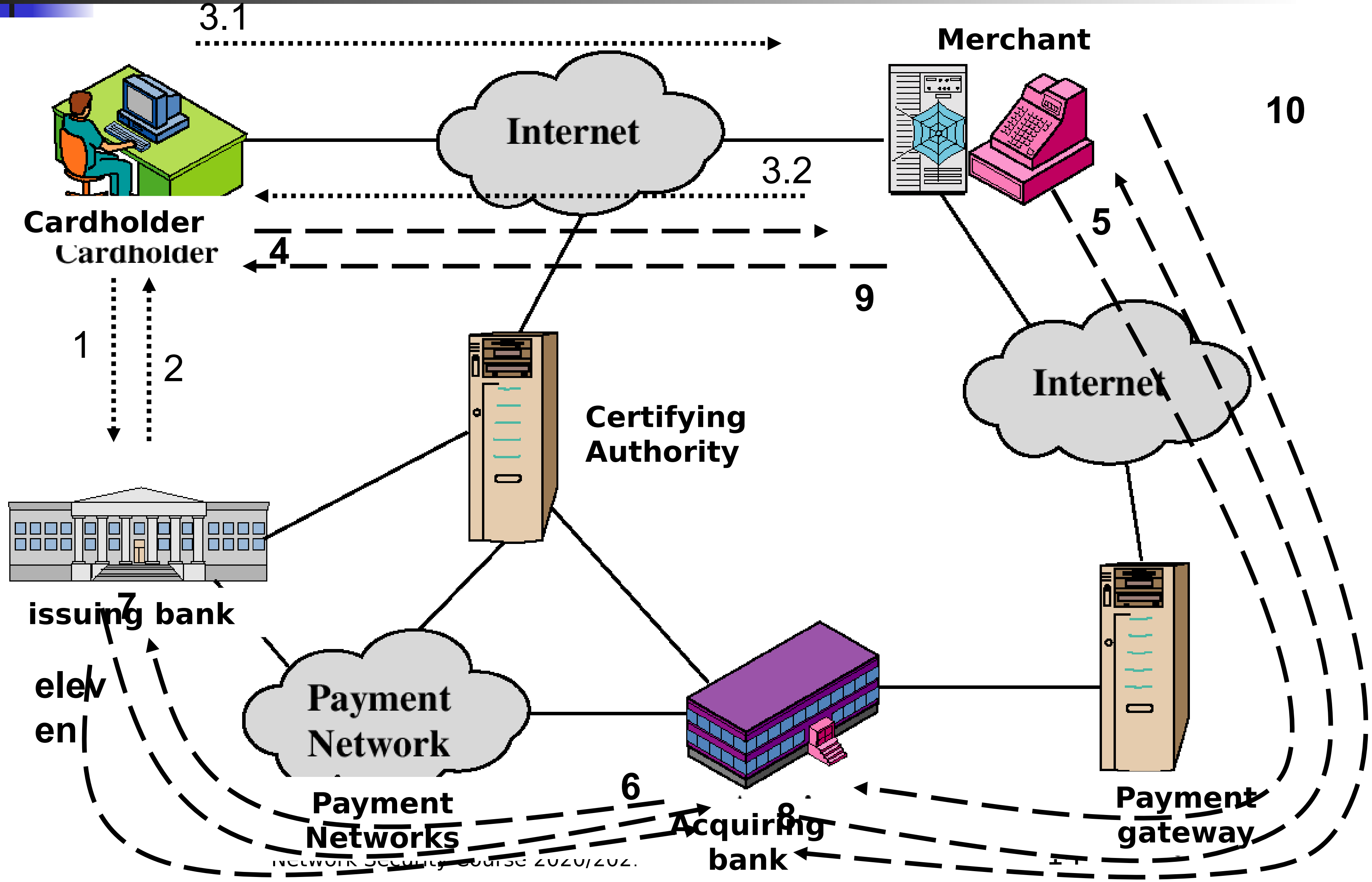
SEQUENCE OF ACTIONS (II)

6. Acquiring bank validates customer and merchant and obtains authorization from issuing bank.
7. Issuing bank authorizes payment.
8. Acquiring bank sends funds transfer witness to merchant.
9. Merchant sends receipt and merchandise to customer.
10. Merchant uses funds transfer witness to collect transaction
11. Money is deducted from the customer's account

Secure Electronic Transactions

NOT IN THE EXAM

SEQUENCE OF ACTIONS (III)



NOT IN THE EXAM

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

4.2.1 Participants
4.2.2 Services
4.2.3 Sequence of actions
4.2.4 Dual Signature
4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

4.3.1 Architecture
4.3.2 Sessions and connections
4.3.3 Record Protocol
4.3.4 Change Cipher Spec Protocol
4.3.5 Alert Protocol
4.3.6 Handshake Protocol
4.3.7 Cryptographic calculations
4.3.8 Additional considerations

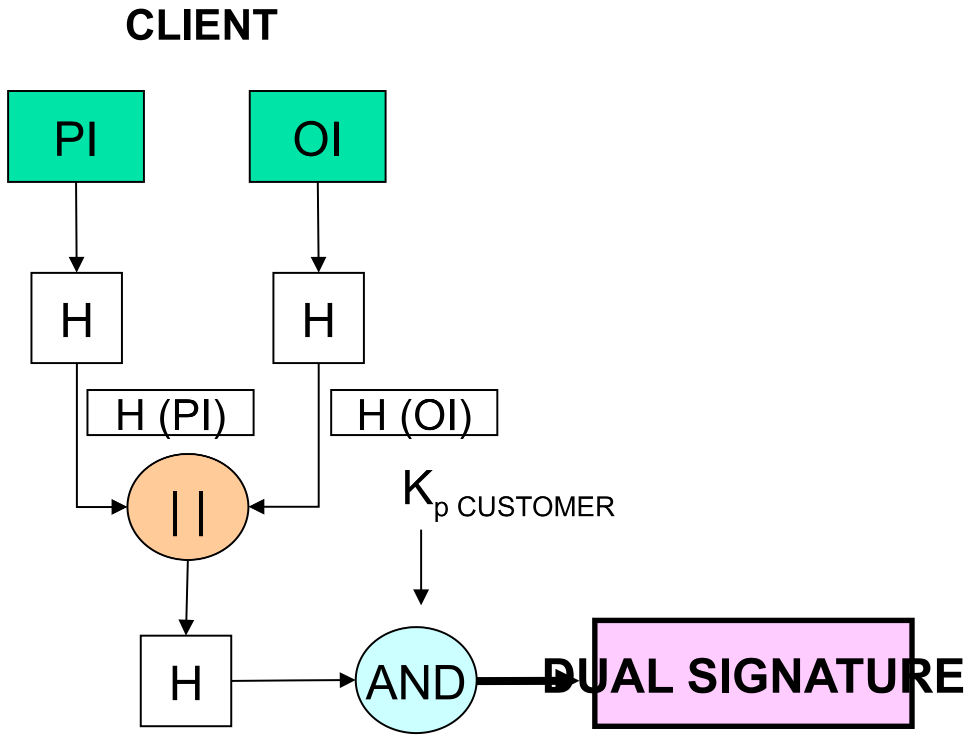
4.4 IPSec

4.4.1 IPSec bound protocols
4.4.2 Security Associations
4.4.3 IPSec protocols
4.4.4 Inter-entity authentication and security association formation

Secure Electronic Transactions

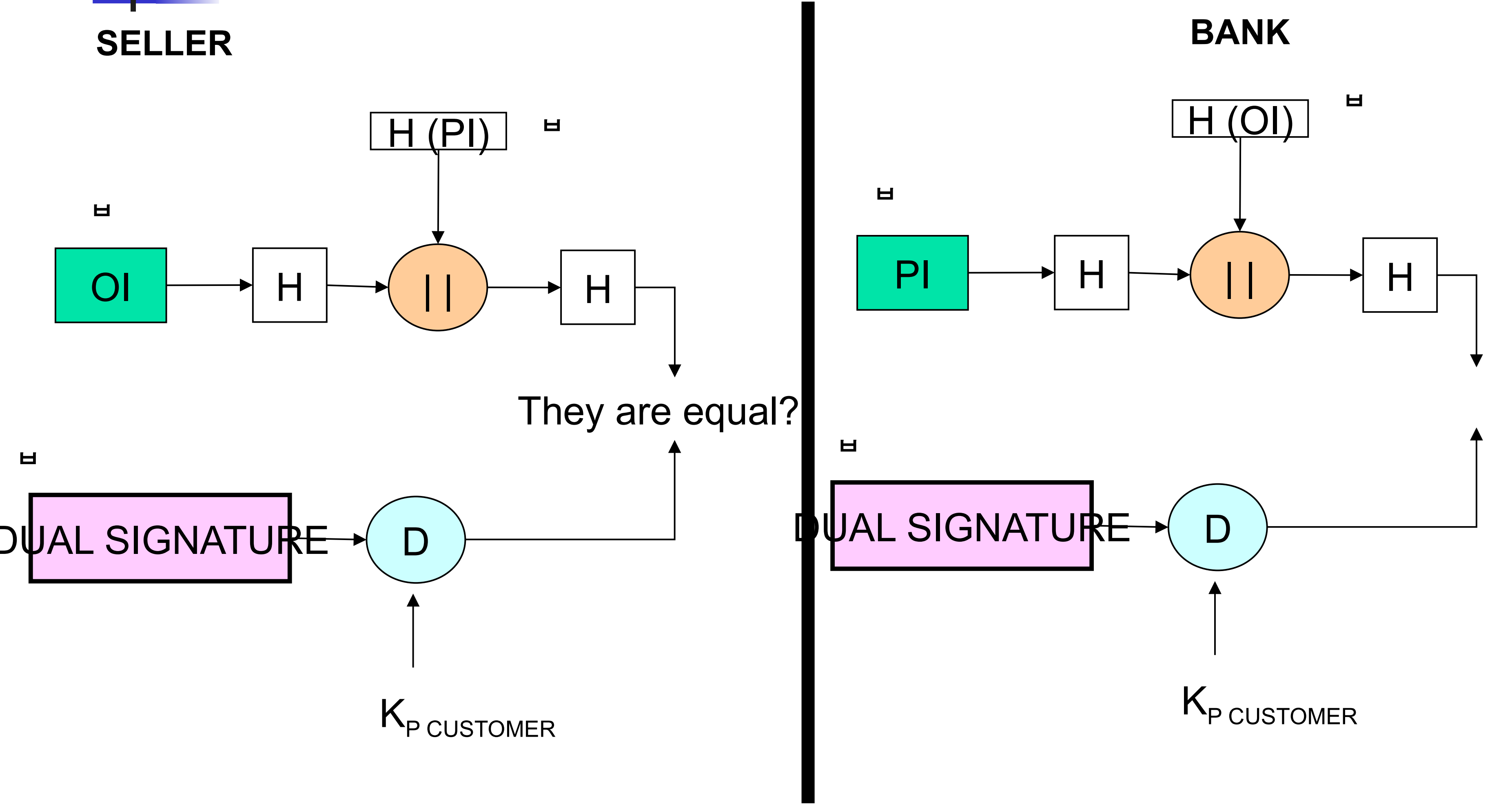
NOT IN THE EXAM

- **Dual Signature**, Purchase order (OI) and payment information (PI) in a single message
- Functioning:



Secure Electronic Transactions

NOT IN THE EXAM



NOT IN THE EXAM

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

- 4.2.1 Participants
- 4.2.2 Services
- 4.2.3 Sequence of actions
- 4.2.4 Dual Signature
- 4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

- 4.3.1 Architecture
- 4.3.2 Sessions and connections
- 4.3.3 Record Protocol
- 4.3.4 Change Cipher Spec Protocol
- 4.3.5 Alert Protocol
- 4.3.6 Handshake Protocol
- 4.3.7 Cryptographic calculations
- 4.3.8 Additional considerations

4.4 IPSec

- 4.4.1 IPSec bound protocols
- 4.4.2 Security Associations
- 4.4.3 IPSec protocols
- 4.4.4 Inter-entity authentication and security association formation

Secure Electronic Transactions

NOT IN THE EXAM

- Types of Transactions
  - Holder's record
  - Seller registration
  - **Purchase request**
  - **Payment authorization**
  - Payment capture
  - Report and certificate status
  - Purchase report
  - Undo authorization
  - Undo capture
  - Credit
  - Undo credit
  - Payment gateway certificate request
  - Batch management
  - Error message

Secure Electronic Transactions

NOT IN THE EXAM

TRANSACTION: PURCHASE REQUEST

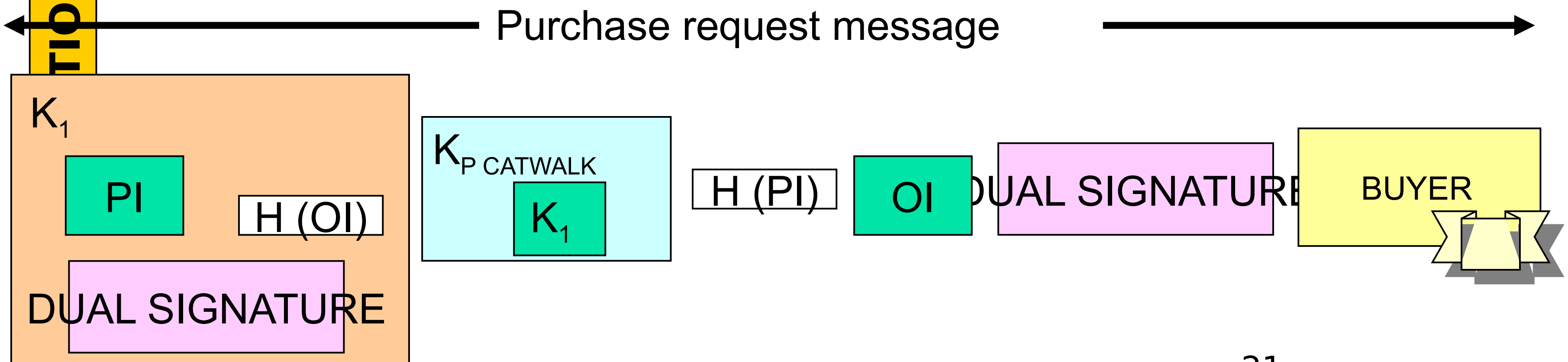
- Start request, start response, purchase request, and purchase response
- **Start request**
  - Request certificates (seller and payment gateway)
  - Includes card type, sequence number and *nonce*
- **Start response**
  - *Nonce* of the buyer, *nonce* of the next message and transaction identifier
  - Digital certificate of seller and payment gateway
  - Message signed by the seller

Secure Electronic Transactions

NOT IN THE EXAM

TION: PURCHASE REQUEST

- **Purchase request**
  - Certificate verification
  - Purchase order (Order Information, OI) together with transaction id
  - Payment information (PI) along with transaction id
  - Symmetric encryption key K_1



Secure Electronic Transactions

NOT IN THE EXAM

TRANSACTION: PURCHASE REQUEST

- **Purchase response**
  - After verifying the buyer's certificate and dual signature, the order is processed and payment information is sent to the gateway
  - Includes order acknowledgment and corresponding transaction number
  - Signed with seller's digital signature

Secure Electronic Transactions

NOT IN THE EXAM

TRANSACTION: PAYMENT AUTHORIZATION

- **Payment authorization**: Authorization request and authorization response
- Authorization request
  - Information related to the acquisition (PI, dual signature, H (OI), key K_1) obtained from the purchase request message
  - Information regarding authorization (identifier of the transaction signed with the seller's private key and encrypted with the symmetric key K_2, and key K_2 encrypted with payment gateway public key)
  - Buyer's Certificate and Seller's Certificate

Secure Electronic Transactions

NOT IN THE EXAM

TRANSACTION: PAYMENT AUTHORIZATION

- Authorization response
  - After verifying certificates
    - Get K_2 and decrypts authorization related information
    - Verify seller signature
    - Get K_1 and decrypts acquisition information
    - Verify dual signature
    - Verify transaction identifier of the seller with the one received from the buyer (contained in PI)
    - Request and receive authorization from the card issuing bank
  - Contains authorization information, catch voucher information, gateway certificate

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

- 4.2.1 Participants
- 4.2.2 Services
- 4.2.3 Sequence of actions
- 4.2.4 Dual Signature
- 4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

- 4.3.1 Architecture
- 4.3.2 Sessions and connections
- 4.3.3 Record Protocol
- 4.3.4 Change Cipher Spec Protocol
- 4.3.5 Alert Protocol
- 4.3.6 Handshake Protocol
- 4.3.7 Cryptographic calculations
- 4.3.8 Additional considerations

4.4 IPSec

- 4.4.1 IPSec bound protocols
- 4.4.2 Security Associations
- 4.4.3 IPSec protocols
- 4.4.4 Inter-entity authentication and security association formation

Secure Socket Layer

- SSL ≡ Secure Socket Layer
  - Confidentiality, integrity, authentication and non-repudiation
  - Client / server applications over reliable transport (TCP)
- Netscape created SSL (1996 v.3)
- The TLS working group was formed within the IETF
  - The first version of TLS can be seen as SSLv3.1
- Features:
  - SSL server authentication
  - SSL client authentication

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

4.2.1 Participants
4.2.2 Services
4.2.3 Sequence of actions
4.2.4 Dual Signature
4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

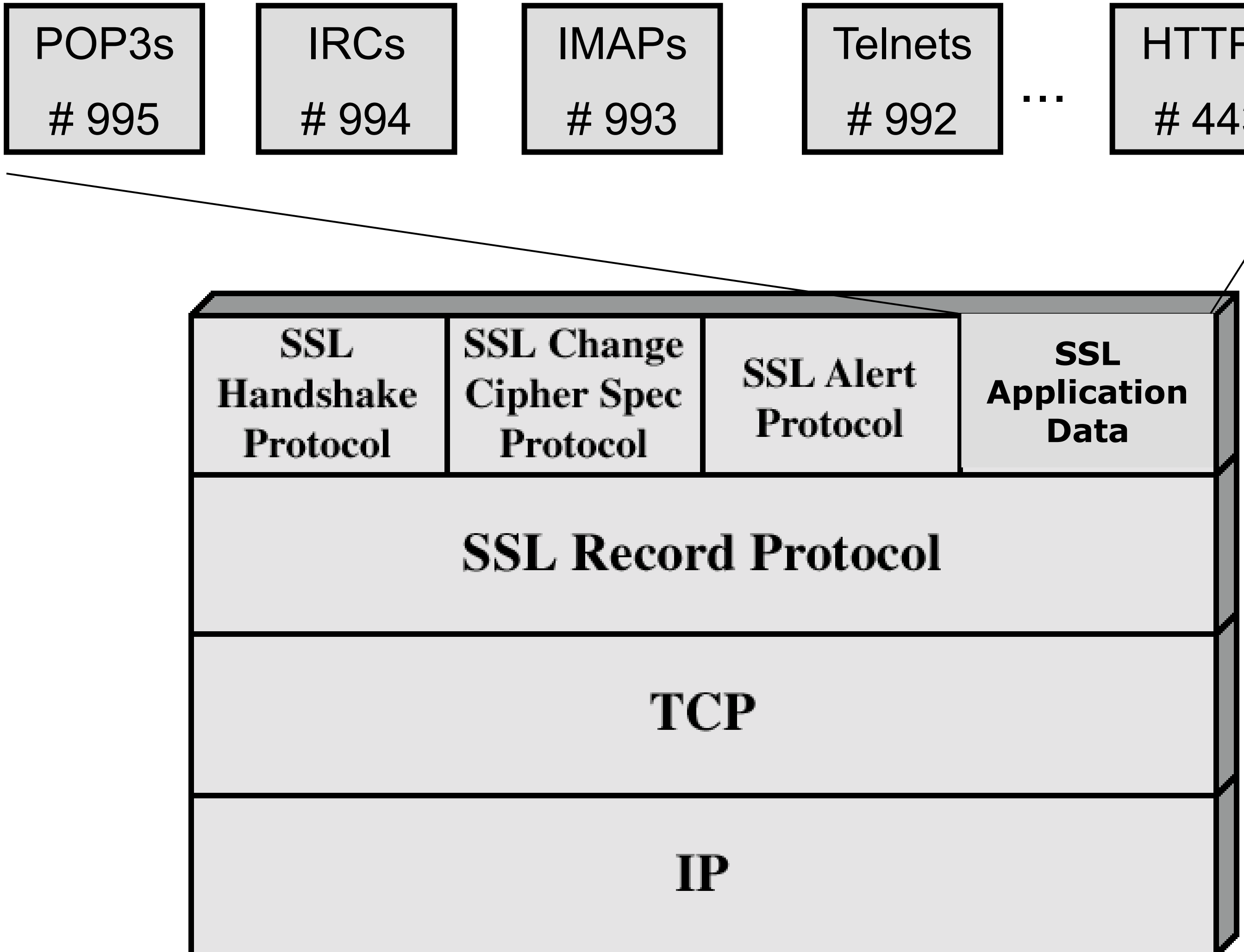
4.3.1 Architecture
4.3.2 Sessions and connections
4.3.3 Record Protocol
4.3.4 Change Cipher Spec Protocol
4.3.5 Alert Protocol
4.3.6 Handshake Protocol
4.3.7 Cryptographic calculations
4.3.8 Additional considerations

4.4 IPSec

4.4.1 IPSec bound protocols
4.4.2 Security Associations
4.4.3 IPSec protocols
4.4.4 Inter-entity authentication and security association formation

Secure Socket Layer

ARCHITECTURE



Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

- 4.2.1 Participants
- 4.2.2 Services
- 4.2.3 Sequence of actions
- 4.2.4 Dual Signature
- 4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

- 4.3.1 Architecture
- 4.3.2 Sessions and connections
- 4.3.3 Record Protocol
- 4.3.4 Change Cipher Spec Protocol
- 4.3.5 Alert Protocol
- 4.3.6 Handshake Protocol
- 4.3.7 Cryptographic calculations
- 4.3.8 Additional considerations

4.4 IPSec

- 4.4.1 IPSec bound protocols
- 4.4.2 Security Associations
- 4.4.3 IPSec protocols
- 4.4.4 Inter-entity authentication and security association formation

Secure Socket Layer

- **Sessions**
  - Session: association between client and server
    - Handshake protocol
  - Session phase parameters
    - Session identifier
    - Peer entity certificate
    - Compression method
    - Encryption specification
    - Master key
    - It is renewable

Secure Socket Layer

- **Connections**
  - Connection: Transport service
    - Each connection associated with a session
  - Connection status parameters
    - Random server and client values
    - Server writable MAC secret key
    - Client write MAC secret key
    - Server write key
    - Client write key
    - Client and Server Initialization Vector (IV)
    - Sequence number

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

4.2.1 Participants
4.2.2 Services
4.2.3 Sequence of actions
4.2.4 Dual Signature
4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

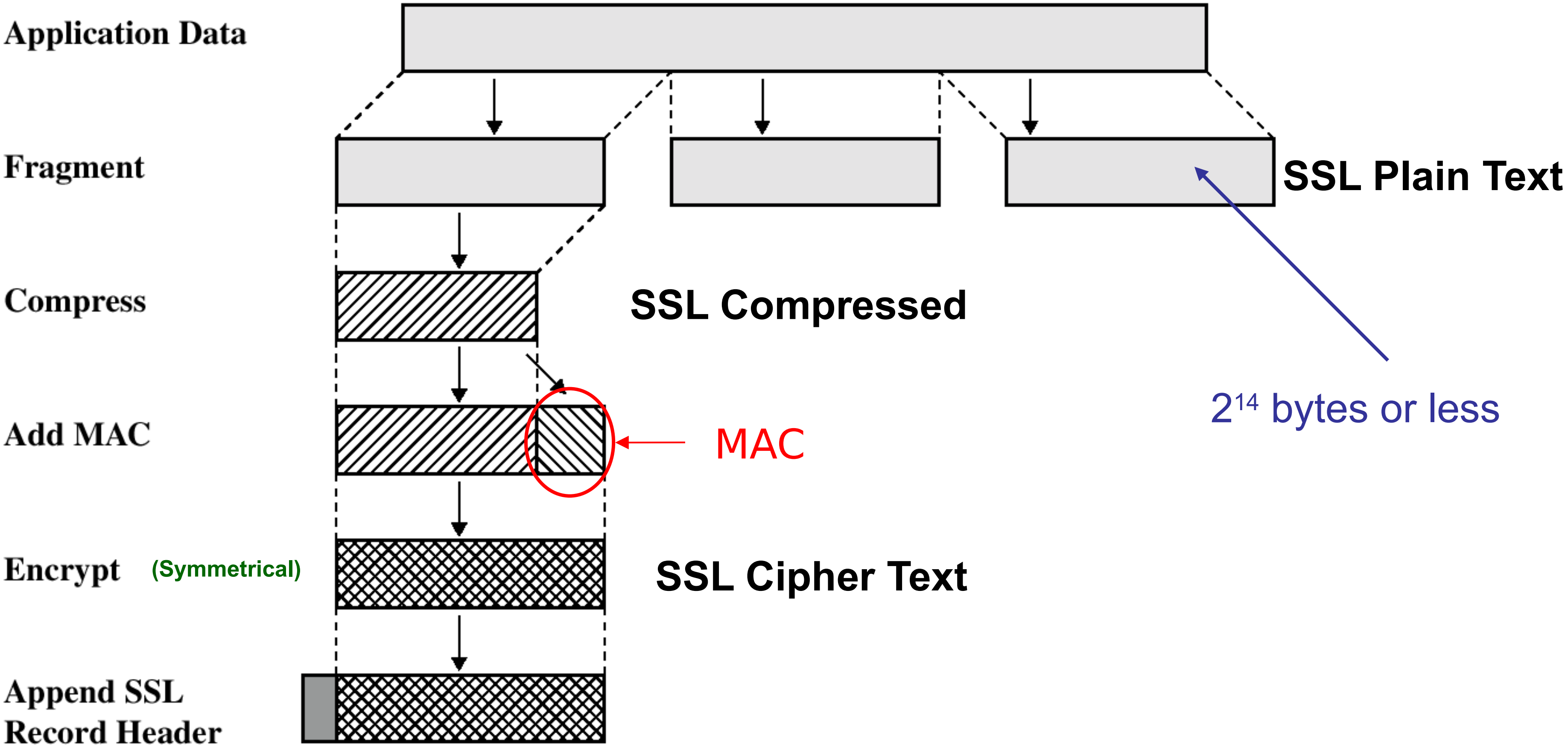
4.3.1 Architecture
4.3.2 Sessions and connections
4.3.3 Record Protocol
4.3.4 Change Cipher Spec Protocol
4.3.5 Alert Protocol
4.3.6 Handshake Protocol
4.3.7 Cryptographic calculations
4.3.8 Additional considerations

4.4 IPSec

4.4.1 IPSec bound protocols
4.4.2 Security Associations
4.4.3 IPSec protocols
4.4.4 Inter-entity authentication and security association formation

Secure Socket Layer

- **Record Protocol**: Confidentiality and integrity

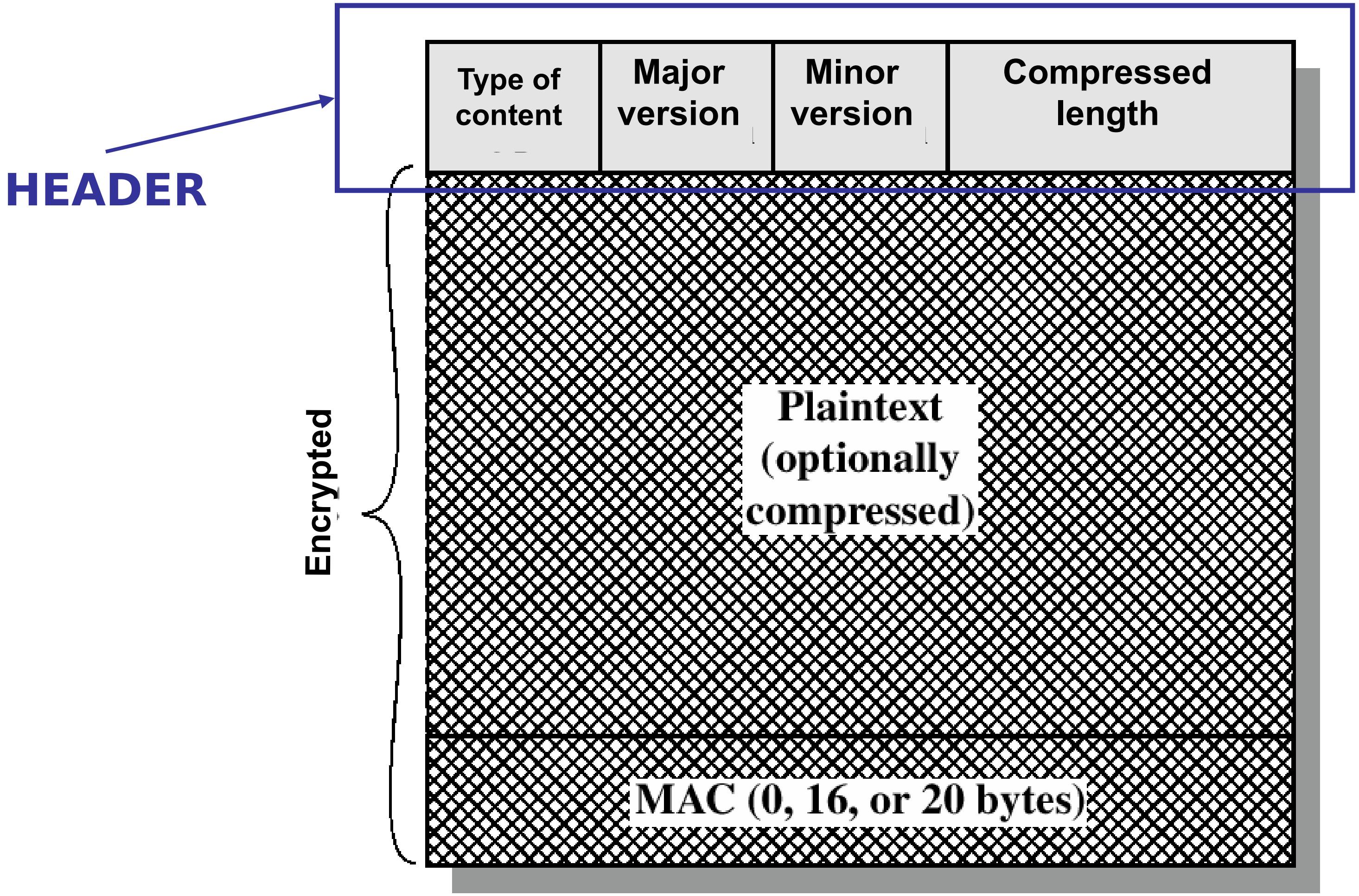


Secure Socket Layer

- MAC
  - Use shared key
  - **Hash (** key_MAC || opad || **hash (** key_MAC || ipad || seq_num || SSLCompressed.type || SSLCompressed.length || SSLCompressed.fragment**))**
  - MD5 or SHA1
- Encryption algorithms

Block Encryption		**Flow Encryption**	
Algorithm	**Size K**	**Algorithm**	**Size K**
IDEA	128	RC4-40	40
DES	56	RC4-128	128
3DES	112		
RSA	1024		
DSA	1024		
FORTEZZA	80		

Secure Socket Layer



HEADER

Type of content

Major version

Minor version

Compressed length

Plaintext (optionally compressed)

MAC (0, 16, or 20 bytes)

Encrypted

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

4.2.1 Participants
4.2.2 Services
4.2.3 Sequence of actions
4.2.4 Dual Signature
4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

4.3.1 Architecture
4.3.2 Sessions and connections
4.3.3 Record Protocol
4.3.4 Change Cipher Spec Protocol
4.3.5 Alert Protocol
4.3.6 Handshake Protocol
4.3.7 Cryptographic calculations
4.3.8 Additional considerations

4.4 IPSec

4.4.1 IPSec bound protocols
4.4.2 Security Associations
4.4.3 IPSec protocols
4.4.4 Inter-entity authentication and security association formation

Secure Socket Layer

- **Change Cipher Spec Protocol**
  - A single content message one byte of value 1
  - Objective: go from pending mode to operational mode

Contents

- 4.1 introduction
- 4.2 Secure Electronic Transactions (SET)
  - 4.2.1 Participants
  - 4.2.2 Services
  - 4.2.3 Sequence of actions
  - 4.2.4 Dual Signature
  - 4.2.5 Permitted transactions
- 4.3 Secure Socket Layer (SSL)
  - 4.3.1 Architecture
  - 4.3.2 Sessions and connections
  - 4.3.3 Record Protocol
  - 4.3.4 Change Cipher Spec Protocol
  - 4.3.5 Alert Protocol
  - 4.3.6 Handshake Protocol
  - 4.3.7 Cryptographic calculations
  - 4.3.8 Additional considerations
- 4.4 IPSec
  - 4.4.1 IPSec bound protocols
  - 4.4.2 Security Associations
  - 4.4.3 IPSec protocols
  - 4.4.4 Inter-entity authentication and security association formation

Secure Socket Layer

- **Alert protocol**
  - Objective: Transmit alerts
  - Message consists of two bytes

LEVEL (1 byte)	**ALERT (1 byte)**

- LEVEL: (1) Warning or (2) Fatal
- ALERT
  - Fatal => Unexpected message, MAC registration failed, decompression failure, negotiation failure, illegal parameter
  - Notice => Notification of closure, no certificate, wrong certificate, not allowed certificate, revoked certificate, expired certificate, unknown certificate

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

4.2.1 Participants
4.2.2 Services
4.2.3 Sequence of actions
4.2.4 Dual Signature
4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

4.3.1 Architecture
4.3.2 Sessions and connections
4.3.3 Record Protocol
4.3.4 Change Cipher Spec Protocol
4.3.5 Alert Protocol
4.3.6 Handshake Protocol
4.3.7 Cryptographic calculations
4.3.8 Additional considerations

4.4 IPSec

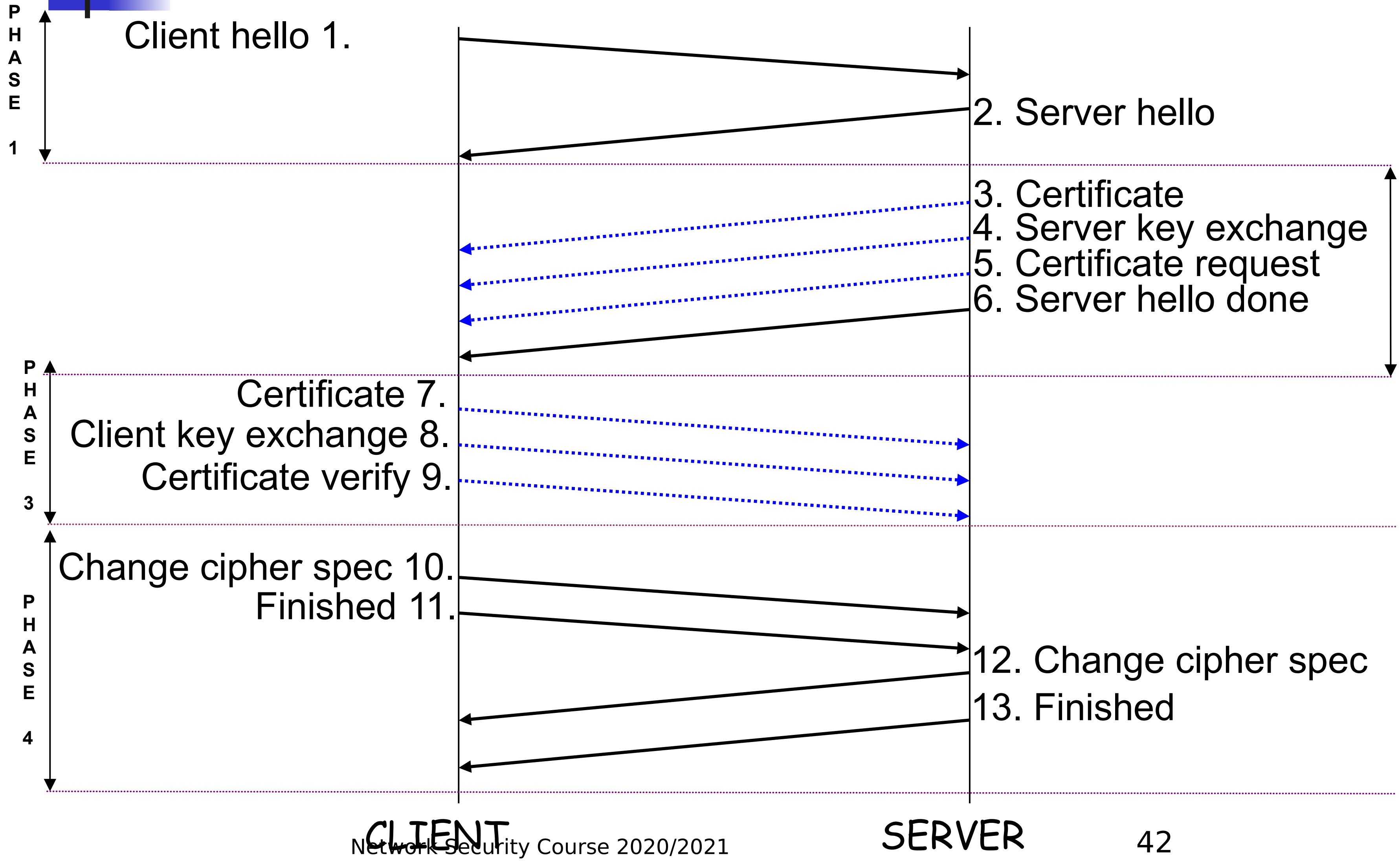
4.4.1 IPSec bound protocols
4.4.2 Security Associations
4.4.3 IPSec protocols
4.4.4 Inter-entity authentication and security association formation

Secure Socket Layer

- **Handshake Protocol**
  - Objective:
    - Client and server must agree on SSL version and compression method
    - Agreement on Encryption Specifications and Creation of Encryption Keys
    - Allows client and server authentication
  - An SSL session always begins with the handshake
  - Handshake messages

TYPE	LENGTH	CONTENT
(1 byte)	(3 bytes)	(≥ 1 byte)

Secure Socket Layer

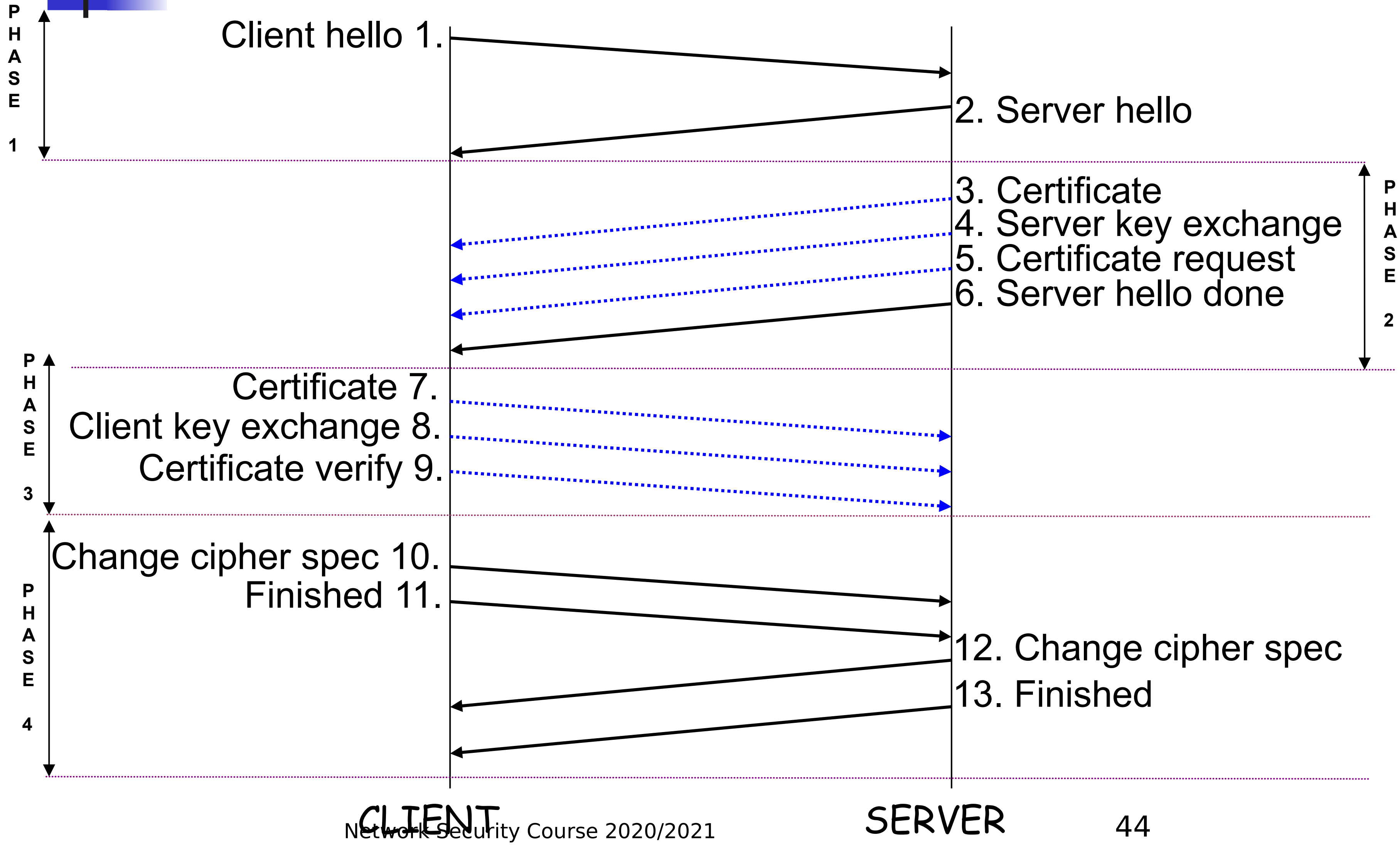


Secure Socket Layer

Phase 1 - Establishment of security capabilities (protocol version, session ID, cipher suite, compression method and n^{or} initial random).

- *Hello client*
  - Version, the highest version number of SSL that supports
  - Random value
  - Session ID, if it is ≠ 0 update the existing connection parameters or <u>create new connection within this session</u>, if = 0 indicates new connection in new session
  - Cipher suite, list of cipher suites that supports
    - Key exchange algorithm
    - Encryption specifications: encryption algorithm, encryption type, is exportable, hash size, key material, initialization vector size
  - Compression method

Secure Socket Layer



PHASE 1

Client hello 1.

2. Server hello

PHASE 2

3. Certificate

4. Server key exchange

5. Certificate request

6. Server hello done

PHASE 3

Certificate 7.

Client key exchange 8.

Certificate verify 9.

PHASE 4

Change cipher spec 10.

Finished 11.

12. Change cipher spec

13. Finished

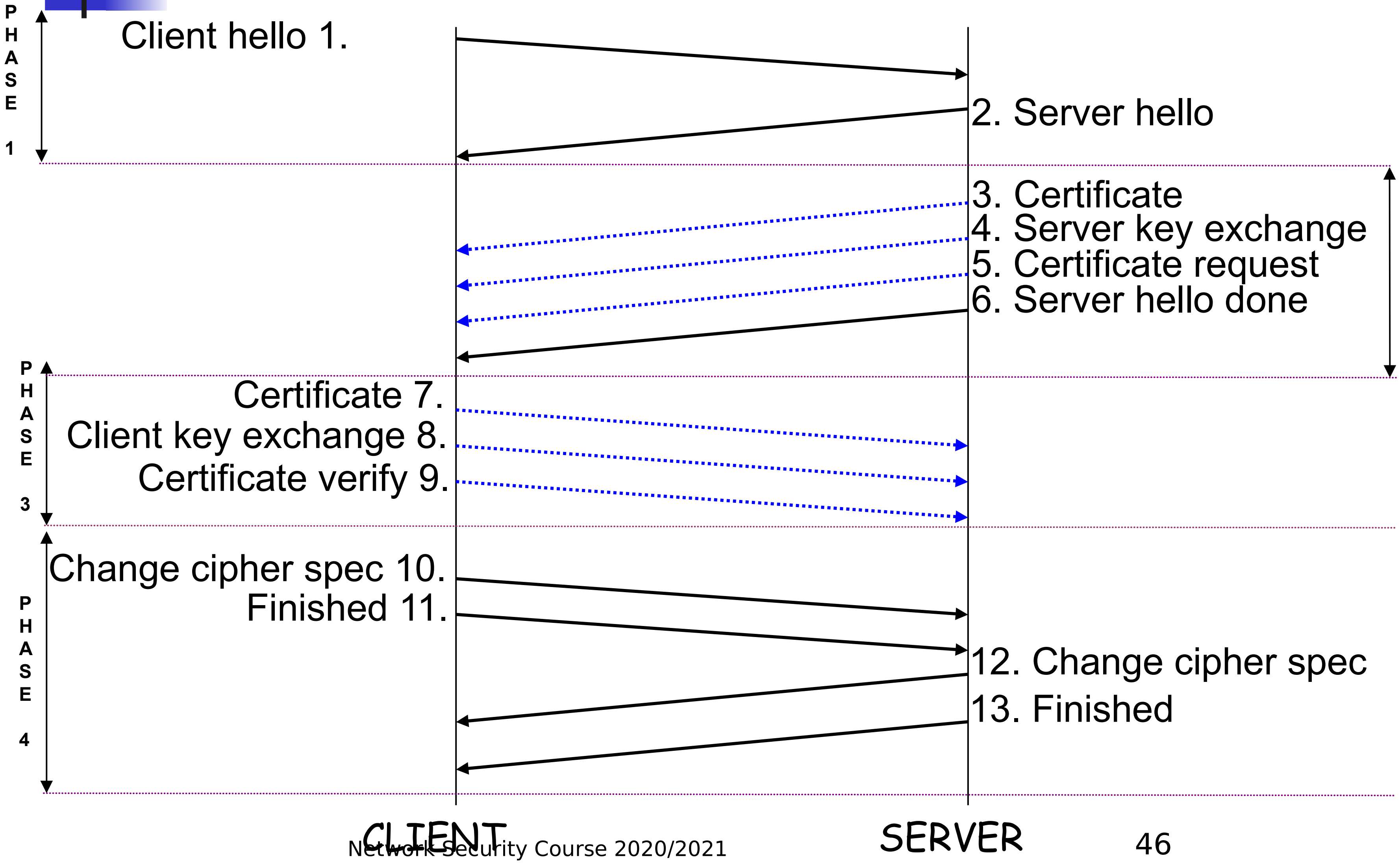
CLIENT

SERVER

Secure Socket Layer

- *Server hello*
  - Version
  - Random value
  - Session id
    - If client session id = 0 => server session id contains different value indicating that a new session has been created
    - If client session id ≠ 0 => server checks in its cache if it saves information about that connection, if so and a new connection can be created, it responds the same client session id
  - Encryption suite, chosen from among those proposed by client
  - Compression method, chosen from those proposed by the client

Secure Socket Layer



PHASE 1

Client hello 1.

2. Server hello

PHASE 2

3. Certificate

4. Server key exchange

5. Certificate request

6. Server hello done

PHASE 3

Certificate 7.

Client key exchange 8.

Certificate verify 9.

PHASE 4

Change cipher spec 10.

Finished 11.

12. Change cipher spec

13. Finished

CLIENT

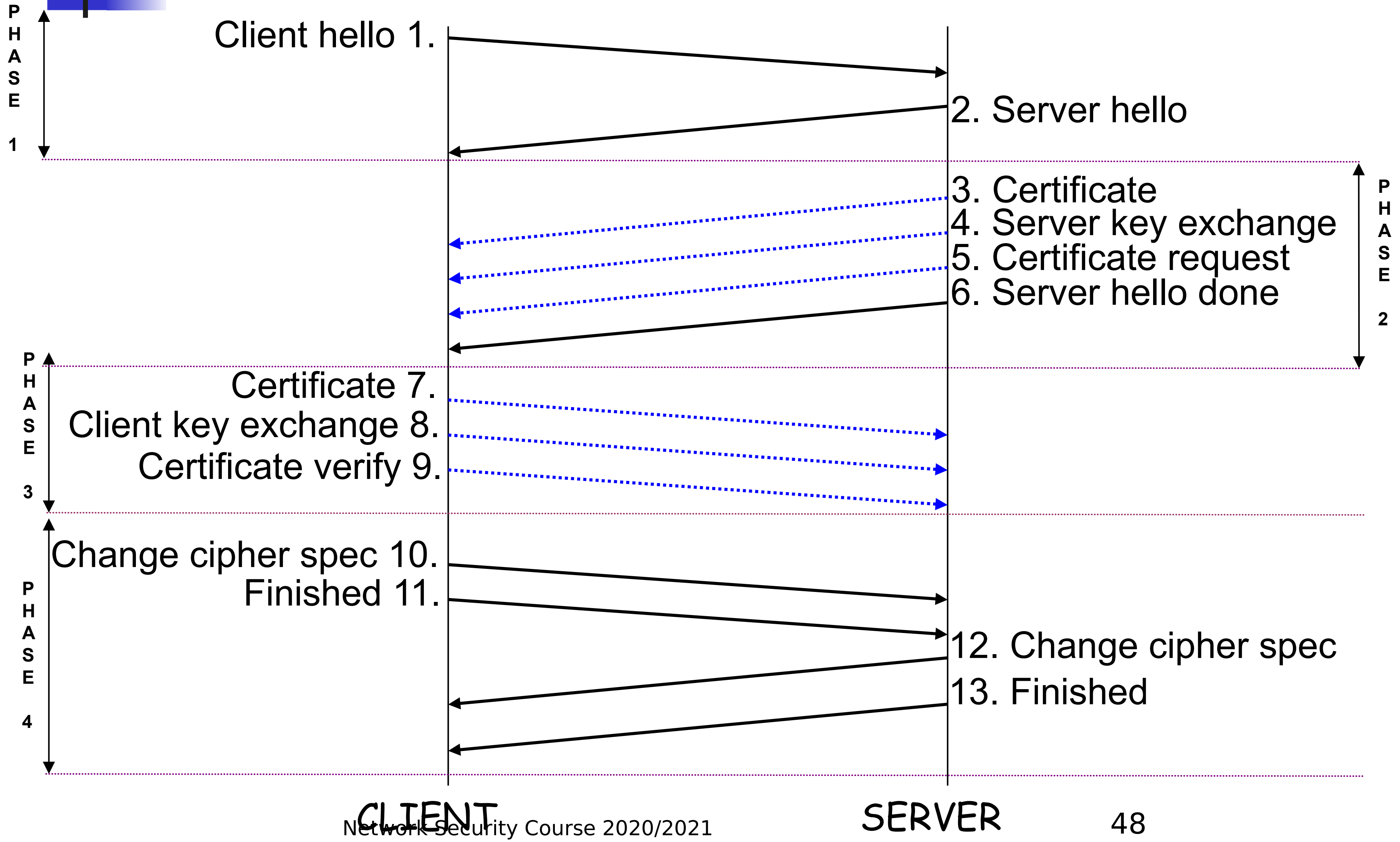
SERVER

Secure Socket Layer

Phase 2 - The server can send a certificate, key exchange and certificate request. The server signals the end of the hello message phase.

- *Certificate*, server sends its X.509 v.3 certificate
- *Server key exchange*,
  - It is not necessary if (1) server has sent certificate with Diffie-Hellman parameters or (2) RSA is used for key exchange
- *Certificate request*, request certificate from client
- *Server hello done*, indicates end of phase 2, does not contain parameters. Server awaits response from client.

Secure Socket Layer

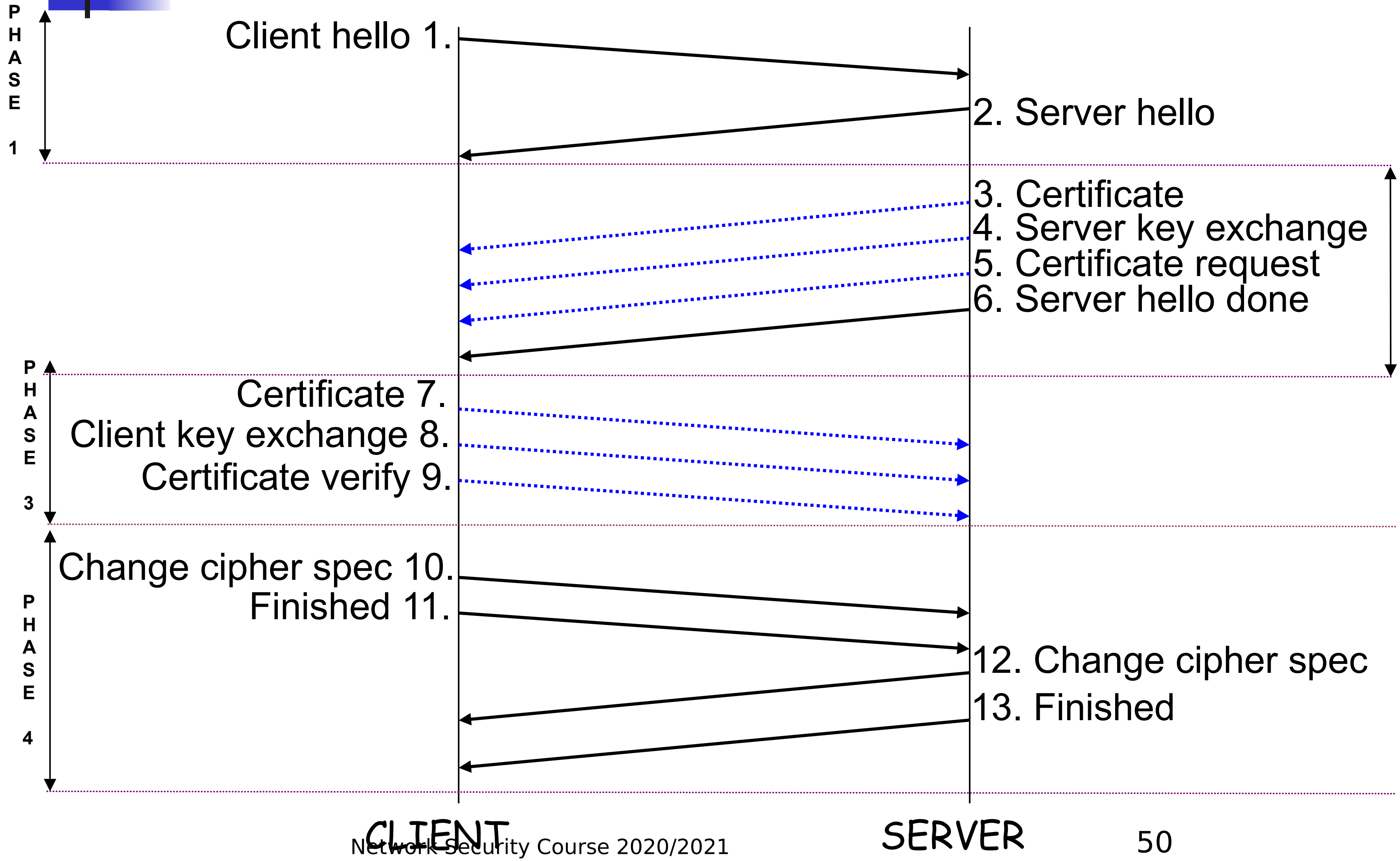


Secure Socket Layer

Phase 3 - Client sends certificate if requested, key exchange, and may send certificate verification.

- Client verifies server certificate
- Check which phase 1 parameters are acceptable
  - *Certificate*, client sends its certificate (if it does not have *not certified*)
  - *Client key exchange*, depends on type of key exchange
    - RSA, sends previous 48 bytes of master key encrypted with server public key
    - Diffie-Hellman, client public Diffie-Hellman parameters (if already in certificate content null)
  - *Certificate verify*, verify that the client has a private key in accordance with the client certificate

Secure Socket Layer



Secure Socket Layer

Phase 4 - Cipher suite exchange and handshake protocol completion.

- The secure connection establishment is completed.
  - *Change cipher spec*, goes from pending to operational mode (Change Cipher Spec protocol)
  - *Finished*, client and server send it using new algorithms and keys

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

4.2.1 Participants
4.2.2 Services
4.2.3 Sequence of actions
4.2.4 Dual Signature
4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

4.3.1 Architecture
4.3.2 Sessions and connections
4.3.3 Record Protocol
4.3.4 Change Cipher Spec Protocol
4.3.5 Alert Protocol
4.3.6 Handshake Protocol
4.3.7 Cryptographic calculations
4.3.8 Additional considerations

4.4 IPSec

4.4.1 IPSec bound protocols
4.4.2 Security Associations
4.4.3 IPSec protocols
4.4.4 Inter-entity authentication and security association formation

Secure Socket Layer

- **Master key**: one-time value (one session) of 48 bytes
- Two steps
  - Exchange of previous value $K_{previous}$
    - RSA or Diffie-Hellman
  - Calculation of the master key

$\mathbf{K}_{\mathbf{master}}$ = **MD5 (** $K_{previous}$ || **SHA (**'A' || $K_{previous}$ || clienthello.random || serverhello.random**))** ||
MD5 ($K_{previous}$ || **SHA (**'BB' || $K_{previous}$ || clienthello.random || serverhello.random**))** ||
MD5 ($K_{previous}$ || **SHA (**'CCC' || $K_{previous}$|| clienthello.random || serverhello.random**))**

Secure Socket Layer

- SSL requires for each connection:
  - server write MAC secret key
  - client write MAC secret key
  - server write key
  - client write key
  - client and server initialization vector (IV)
- They are created in that order starting with K_{master}

K_{block}= MD5 (K_{master}|| SHA ('A' || K_{master} || clienthello.random || serverhello.random)) ||
MD5 (Kmaster || SHA ('BB' || Kmaster || clienthello.random || serverhello.random)) ||
MD5 (Kmaster || SHA ('CCC' || Kmaster || clienthello.random || serverhello.random))……

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

4.2.1 Participants
4.2.2 Services
4.2.3 Sequence of actions

4.2.4 Dual Signature
4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

4.3.1 Architecture
4.3.2 Sessions and connections
4.3.3 Record Protocol
4.3.4 Change Cipher Spec Protocol

4.3.5 Alert Protocol
4.3.6 Handshake Protocol
4.3.7 Cryptographic calculations
4.3.8 Additional considerations

4.4 IPSec

4.4.1 IPSec bound protocols
4.4.2 Security Associations
4.4.3 IPSec protocols
4.4.4 Inter-entity authentication and security association formation

Secure Socket Layer

- Independence of application and dependence on transport
- Export laws
- Standardization
  - **Transport Layer Security**

Transport Layer Security

- 1996, IETF RFC 2246 (SSL with some variations)
  - Format
    - In TLS version major is 3 and minor 1
  - Message authentication code
    - Algorithm to calculate authentication code is HMAC
    - The HMAC is calculated on different fields

MESSAGE AUTHENTICATION CODE

Transport Layer Security

- HMAC algorithm

IN SSL v3

Hash (**key_MAC || opad**|| hash (**key_MAC || ipad**|| seq_num || SSLCompressed.type || SSLCompressed.length || SSLCompressed.fragment))

IN TLS

$$HMAC_K = H\,[(K^{+} \oplus opad) \,||\, H\,[(K^{+} \oplus ipad) \,||\, X]]$$

H≡ MD5 or SHA1 hash function

X ≡ Plain text

K^{+}≡ secret key padded with leading zeros until it equals length of input block of hash functions

ipad ≡ 00110110 repeated

opad ≡01011100 repeated

Transport Layer Security

- Fields on which to calculate the HMAC

IN SSL v3

Hash (MAC_key || opad || hash (MAC_key || ipad || seq_num || SSLCompressed.type || SSLCompressed.length || SSLCompressed.fragment))

IN TLS

HMAC (MAC_key, seq_num || TLSCompressed.type || **TLSCompressed.version**|| TLSCompressed.length || SSLCompressed.fragment))

Protocol version being used

Transport Layer Security

- **Alert Codes**:
  - All SSL v.3 except alert *no-certificate*
  - Additional alerts:
    - Decryption failure (*decryption-failed*)
    - Unknown Certificate Authority (unknown-ca)
    - Insufficient security (insufficient_security)

Transport Layer Security

Other differences:

- Encryption Suite: All symmetric encryption and key exchange techniques available in SSL v.3 except for Fortezza.
- Certificates: does not include Fortezza
- Filling:
  - In SSL minimum padding so that the total size of the data to be encrypted is a multiple of the length of the encrypted block (DES -> 512 bits)
  - With TLS it can be anything (max 255 bytes = 2040 bits)

Transport Layer Security

- TLS uses the PRF (Pseudo Random Function) function for master key expansion

PRF (secret, label, seed) = P_MD5 (S1, label || seed) ⊕ P_SHA-1 (S2, label || seed)

P_hash = HMAC_hash (secret, A (1) || seed) || HMAC_hash (secret, A (2) || seed) || HMAC_hash (secret, A (1) || seed) || ...

A (n):

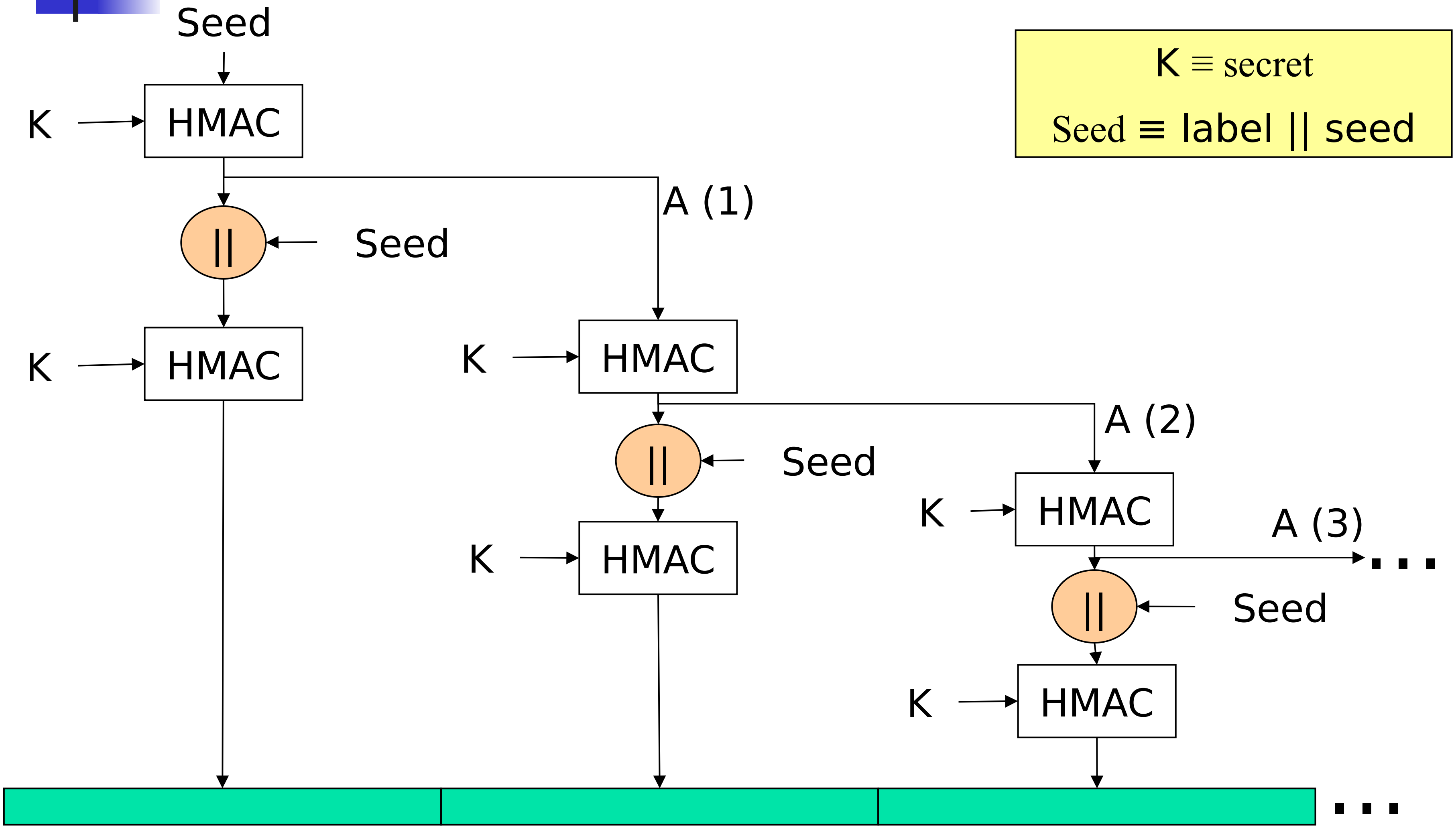
A (0) = seed

A (1) = HMAC_hash (secret, A (0))

...

A (i) = HMAC_hash (secret, A (i-1))

Transport Layer Security



K ≡ secret

Seed ≡ label || seed

P_HASH FUNCTION

Transport Layer Security

Master key creation

- Pre_master_secret $K_{previous}$ same as SSL v.3
- Master key K_{master} 48 bytes:

K_{master}= PRF ($K_{previous}$, “Master secret”, clientHello.random || serverHello.random)

K_{block}= PRF (K_{master}, "Key expansion", SecurityParameters.server_random || SecurityParameters.client_random)

Contents

4.1 introduction

4.2 Secure Electronic Transactions (SET)

4.2.1 Participants
4.2.2 Services
4.2.3 Sequence of actions
4.2.4 Dual Signature
4.2.5 Permitted transactions

4.3 Secure Socket Layer (SSL)

4.3.1 Architecture
4.3.2 Sessions and connections
4.3.3 Record Protocol
4.3.4 Change Cipher Spec Protocol
4.3.5 Alert Protocol
4.3.6 Handshake Protocol
4.3.7 Cryptographic calculations
4.3.8 Additional considerations

4.4 IPSec

4.4.1 IPSec bound protocols
4.4.2 Security Associations
4.4.3 IPSec protocols
4.4.4 Inter-entity authentication and security association formation

IPSec

- IPSec (IP Security Protocol) is a set of open standards that work together to guarantee between peer entities at the network level:
  - Confidentiality
  - Integrity
  - Authentication

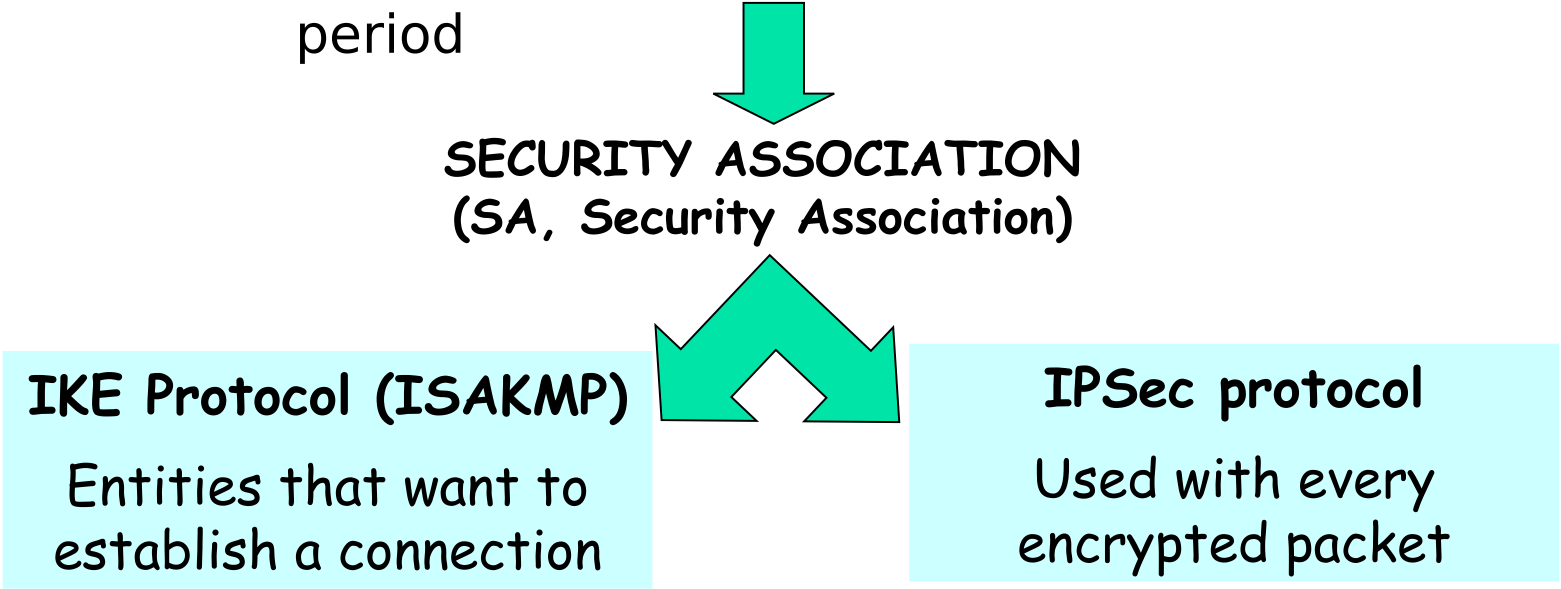
IPSec

- IPSec protocols:
  - **Authentication Header (AH)**
  - **Encapsulation Security Payload (ESP)**
  - Encryption: DES, 3DES, AES, ...
  - Hash functions: HMAC, MD5 or SHA1
  - Digital signature: RSA or shared secret
  - Key exchange: via CA (certificates) or Diffie-Hellman
  - Negotiation of security associations:
    - IKE (Internet Key Exchange)
    - ISAKMP (Internet Security Association and Key Management Protocol)

IPSec

ASSOCIATIONS

- Entities decide what security services they need
- Negotiation process between entities begins
  - Set of common algorithms for authentication, encryption and / or summary functions + validity period



SECURITY ASSOCIATION (SA, Security Association)

IKE Protocol (ISAKMP)

Entities that want to establish a connection

IPSec protocol

Used with every encrypted packet

IPSec

- Associations
  - Simplex
  - IPSec associations depend on the type of protocol
  - Security Parameter Index (SPI)

SPI	@IP dest.	ESP and/or AH

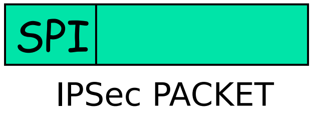
Unique identification of the security association

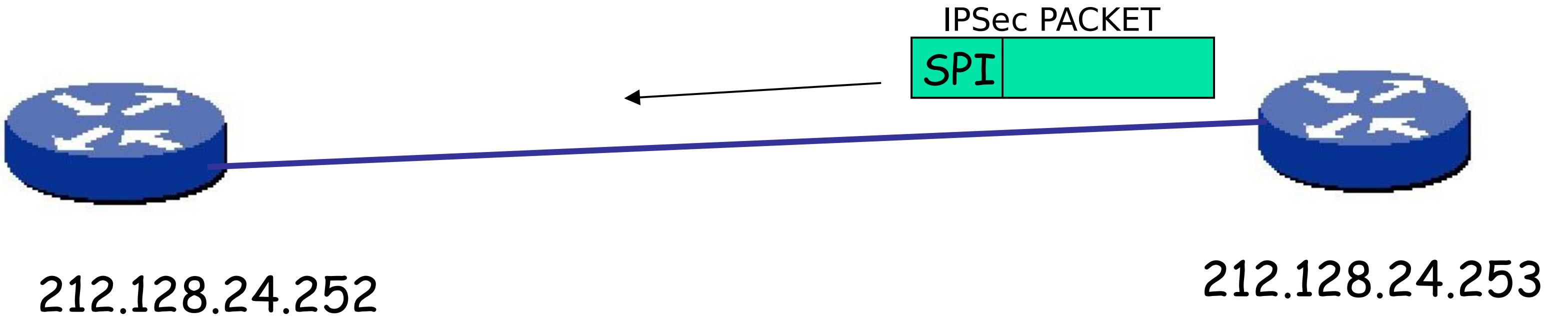
  - Security Association Database

IPSec

1. SPI + @ IPdestination + IPSec protocol
2. SAD query

1. SA? -> SPI
2.

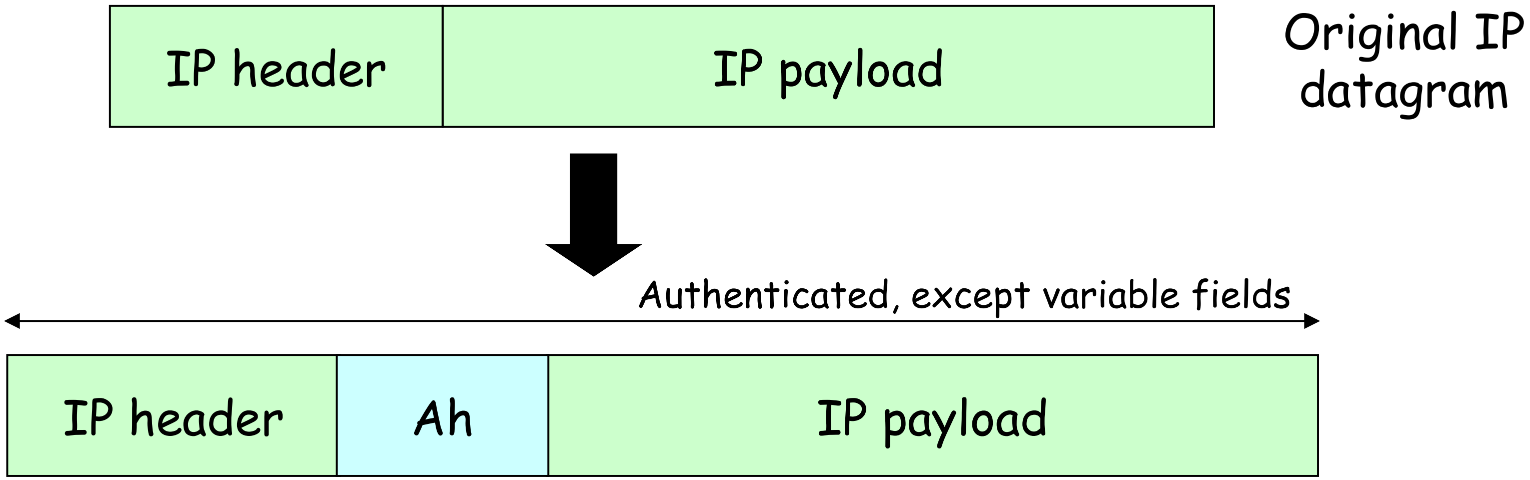




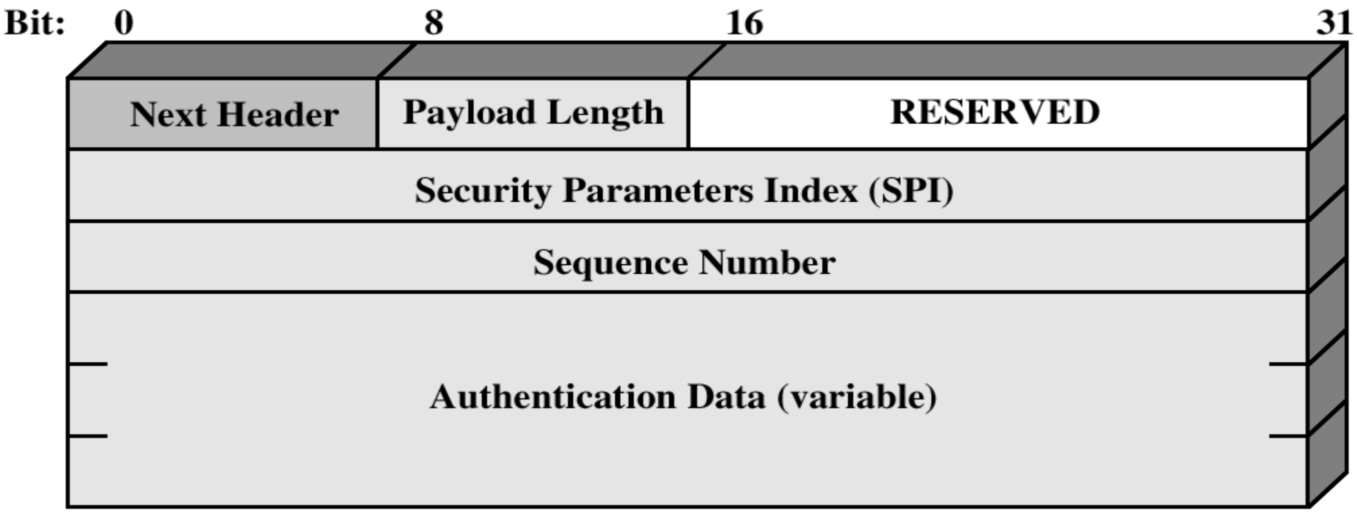
IPSec

- **AH protocol** (RFC 2402)
  - Data integrity
  - Authenticationrn from the data source
  - Service against forwardingíor packages (optional)

Interesting Traffic ≡ AH header added



IPSec



Bit: 0 8 16 31

Next Header | Payload Length | RESERVED

Security Parameters Index (SPI)

Sequence Number

Authentication Data (variable)

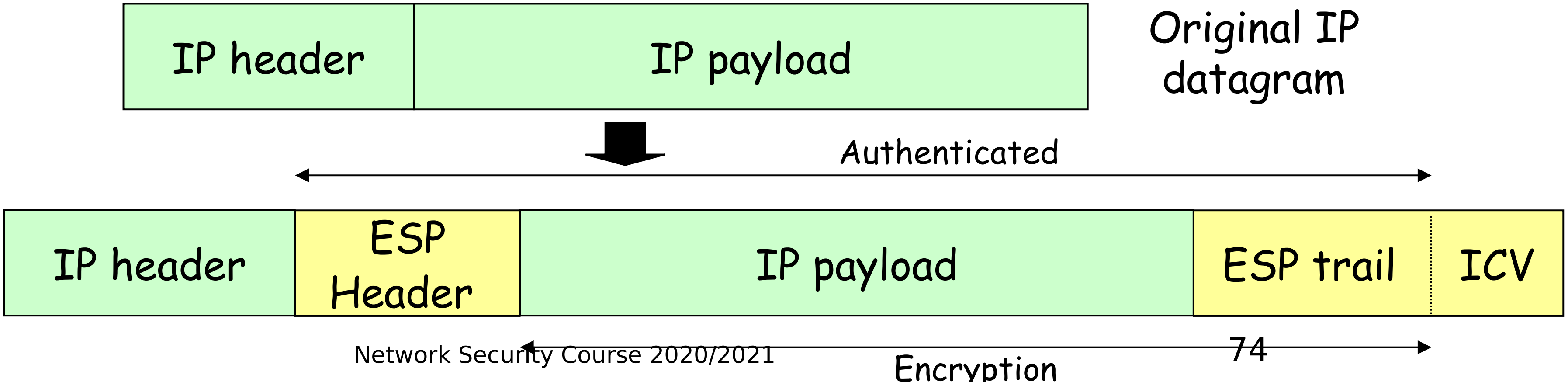
- Next header (6 -> TCP, 17 -> UDP)
- Payload Length, header length in 32-bit words (-2)
- Reserved
- Security Parameters Index (SPI)
- Sequence Number Field, prevent packet forwarding attacks
- Authentication Data, contains Integrity Check Value (ICV) (multiple of 32 bits always -> padding)

IPSec

- ICV is calculated using MAC:
  - Use DES, 3DES or AES
  - Use MD5 or SHA1
  - Use shared key
  - Full IP packet MAC (including AH header fields) bypassing variable fields such as TTL that would go to zero
  - Each entity in the secure association calculates the ICV separately and then checks

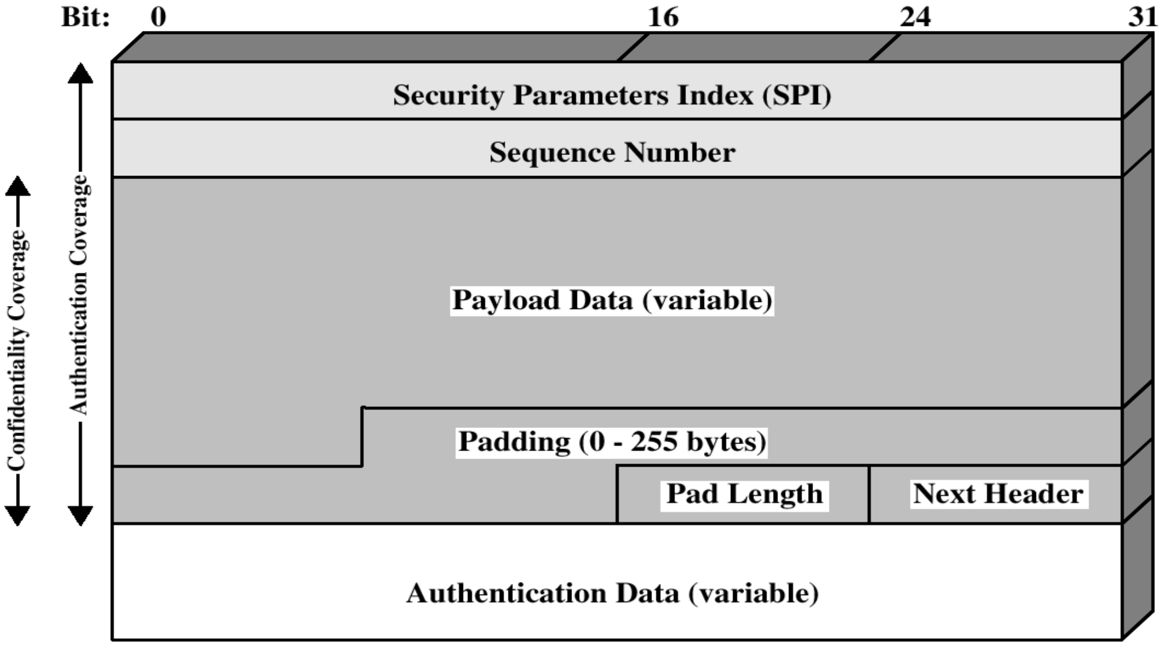
IPSec

- **ESP protocol** (RFC 2406)
  - Confidentiality
  - Data source authentication (optional)
  - Integrity (optional)
  - Counter-forwarding service (optional)
- ESP encapsulates the original IP datagram (complete or not)



IPSec

- ESP header
  - SPI
  - Sequence Numbe
- Payload data (original IP datagram or part of it)
- ESP trail
  - Padding
  - Padding Length
  - Next Header
  - ICV

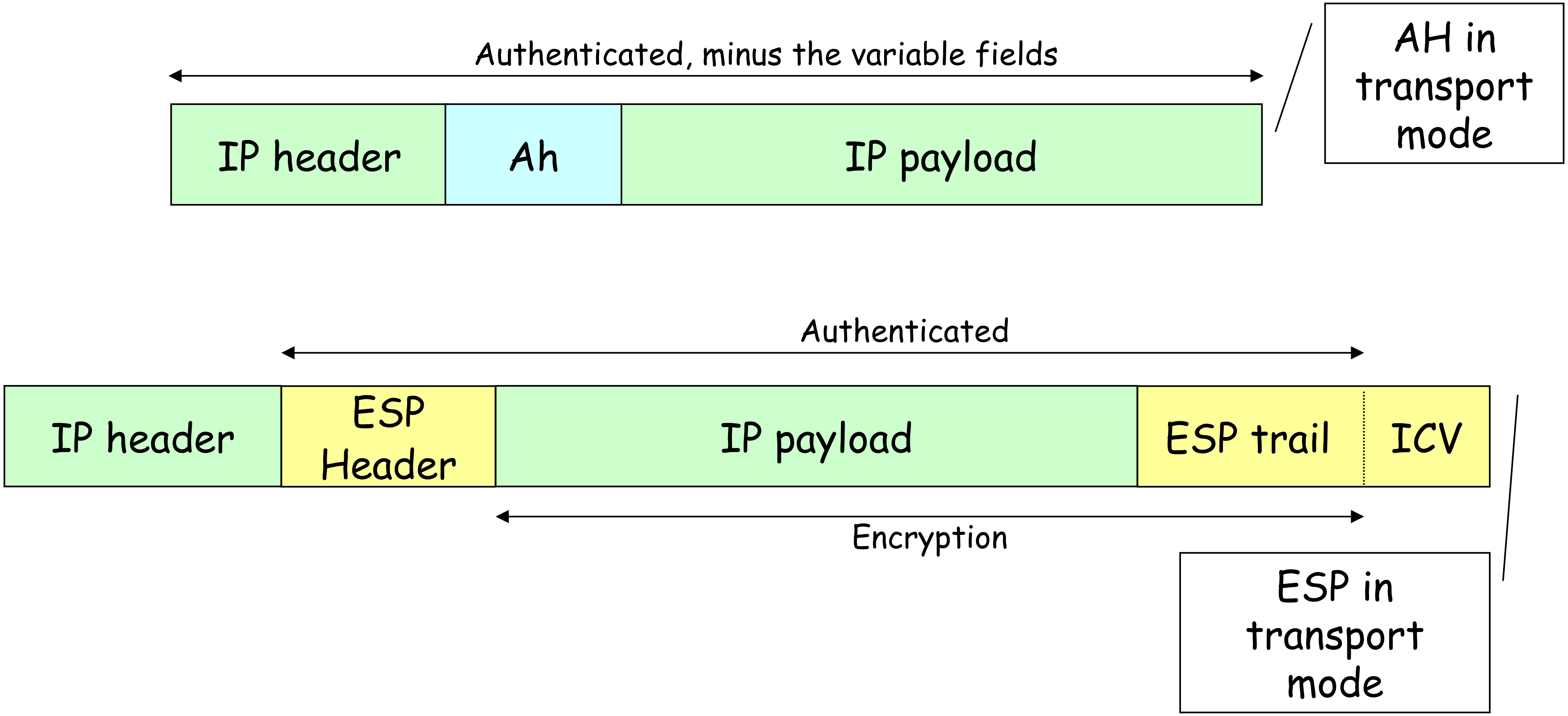


IPSec

- **Transport mode**
  - End-to-end connections between a host and a device that acts as such
- **Tunnel mode**
  - Between gateways, or between a host connecting to a security gateway
  - IP header is copied and shifted to the left
  - New IP header is formed with the copy

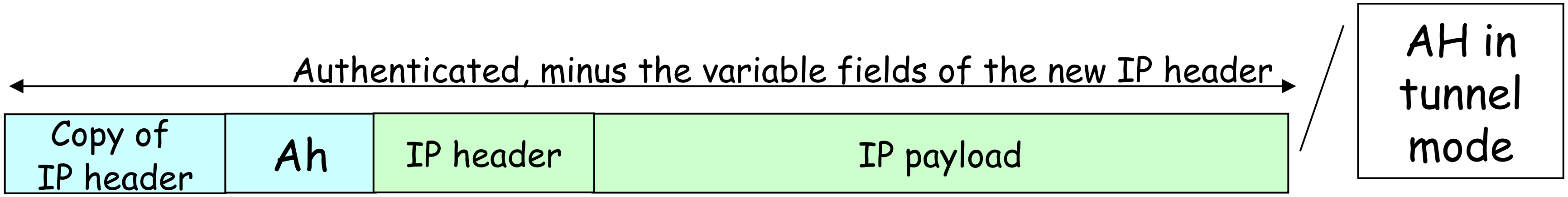
IPSec

- AH and ESP in transport mode



IPSec

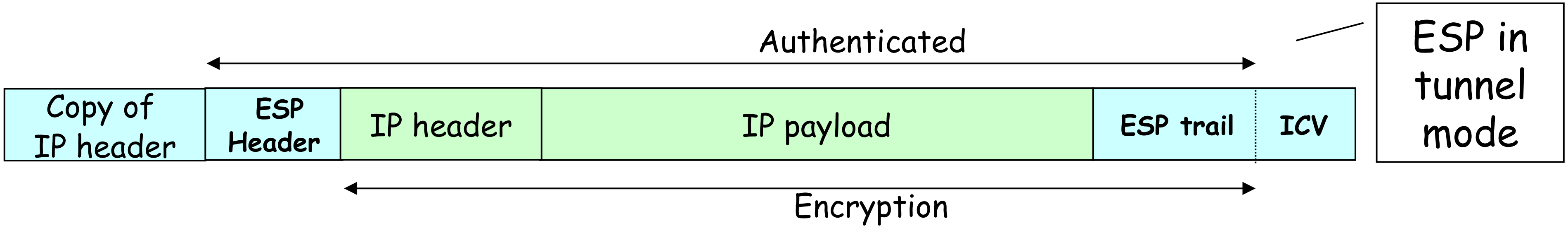
- AH and ESP in tunnel mode



Authenticated, minus the variable fields of the new IP header

Copy of IP header | Ah | IP header | IP payload

AH in tunnel mode



Authenticated

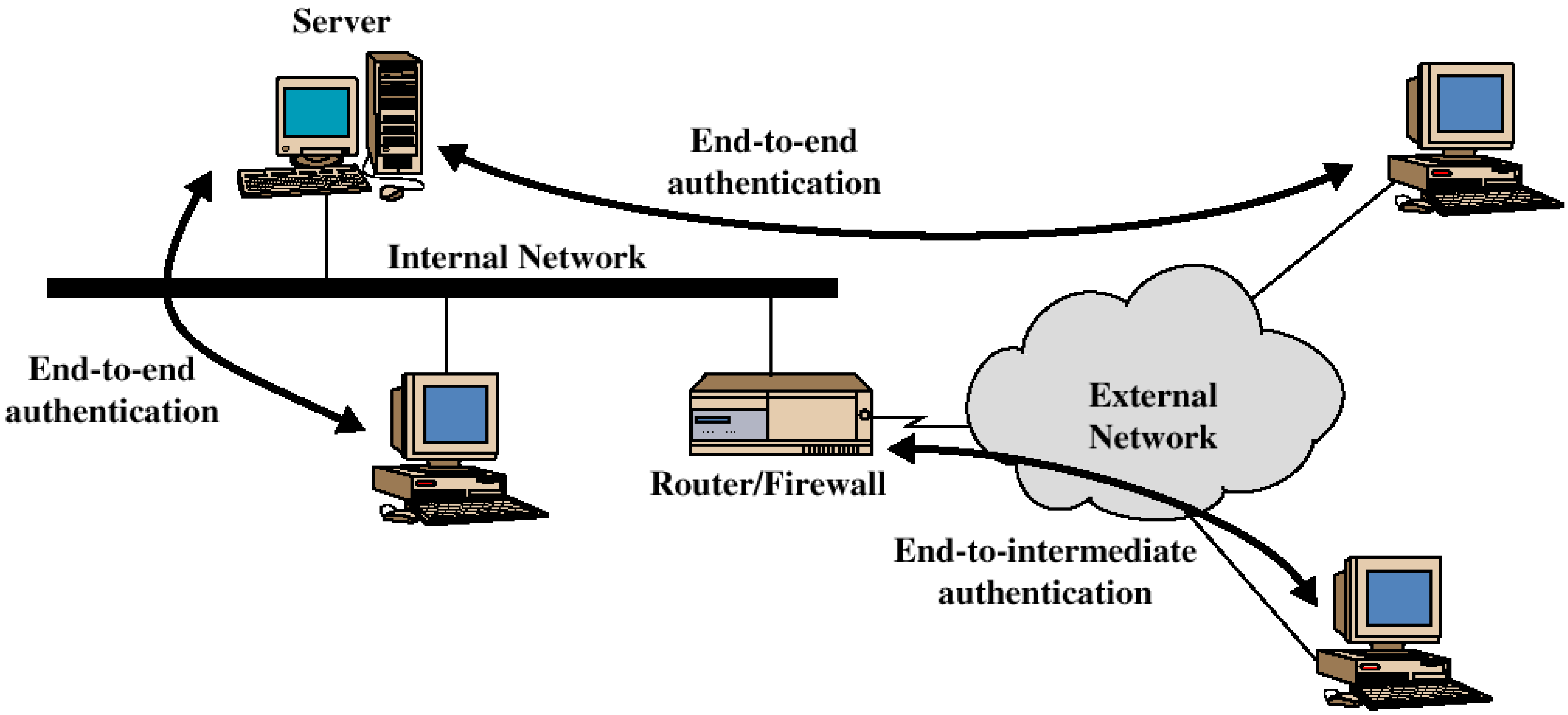
Copy of IP header | ESP Header | IP header | IP payload | ESP trail | ICV

Encryption

ESP in tunnel mode

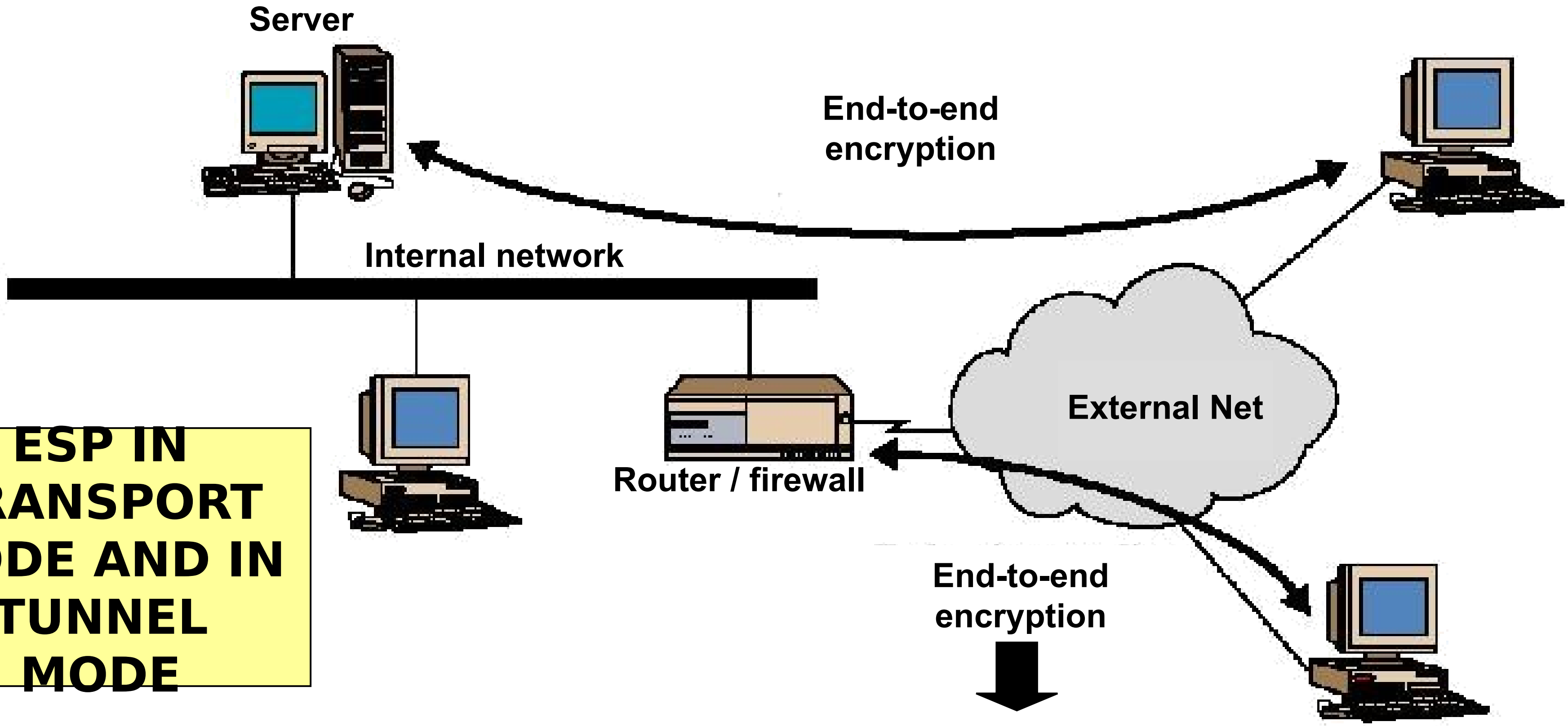
IPSec

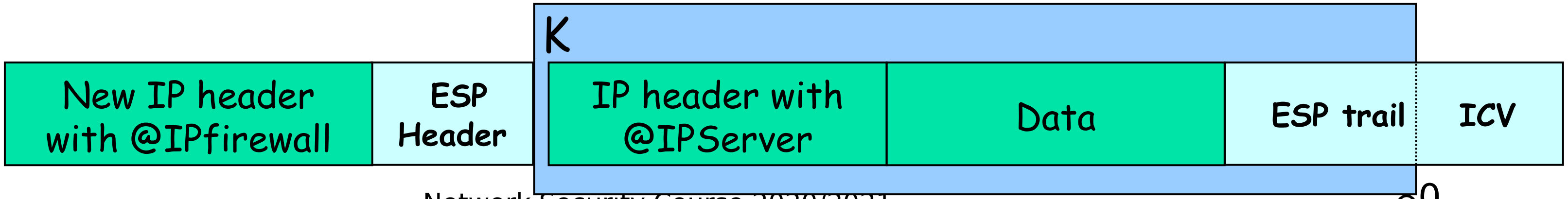
AH IN TRANSPORT MODE AND IN TUNNEL MODE


Server
End-to-end authentication
Internal Network
End-to-end authentication
Router/Firewall
External Network
End-to-intermediate authentication

IPSec

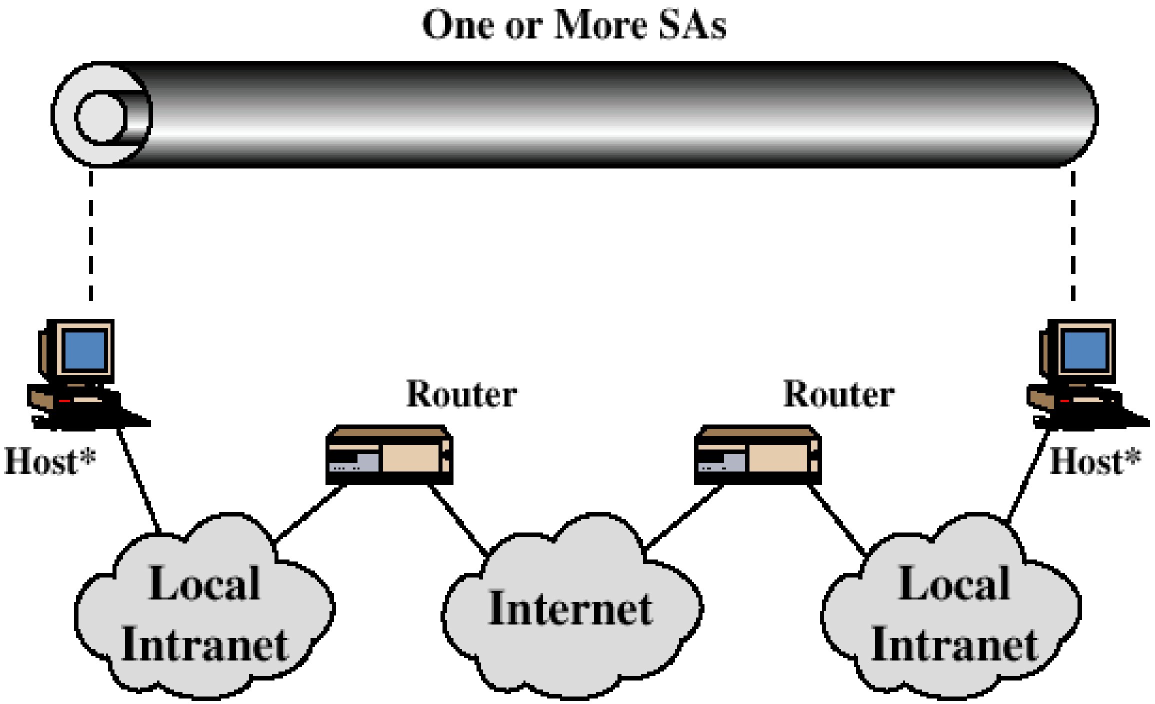
ESP IN TRANSPORT MODE AND IN TUNNEL MODE





IPSec

BASIC COMBINATIONS OF SECURITY ASSOCIATIONS

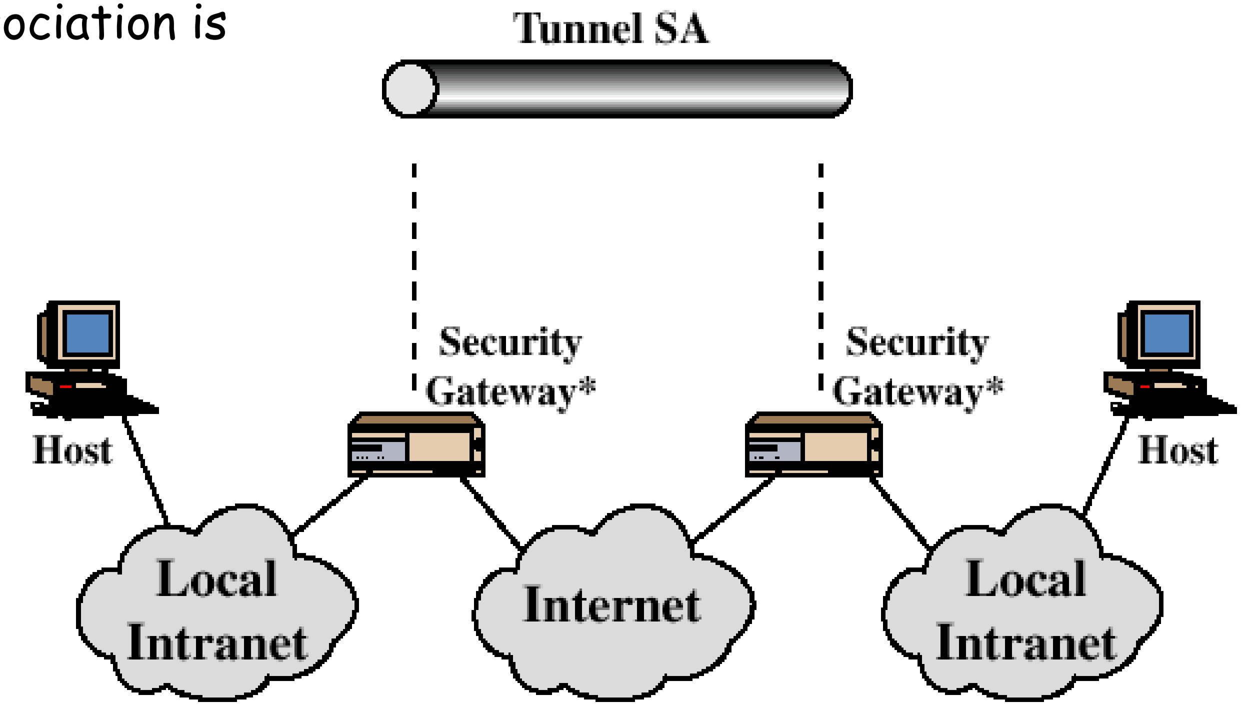


- AH in transport mode
- ESP in tunnel mode
- AH followed by ESP in transport mode
- Any one above within an AH or ESP in tunnel mode

IPSec

BASIC COMBINATIONS OF SECURITY ASSOCIATIONS

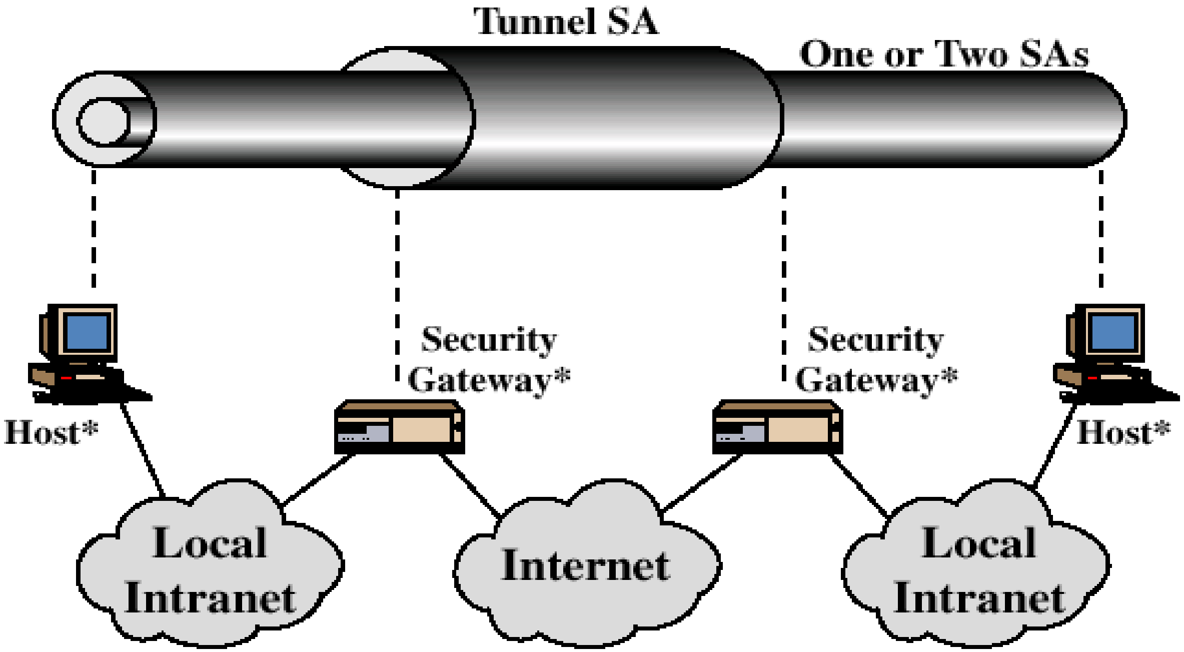
- Illustrates tunnel mode in a virtual private network (VPN)
- Only one security association is needed
- AH, ESP or ESP with authentication option



IPSec

BASIC COMBINATIONS OF SECURITY ASSOCIATIONS

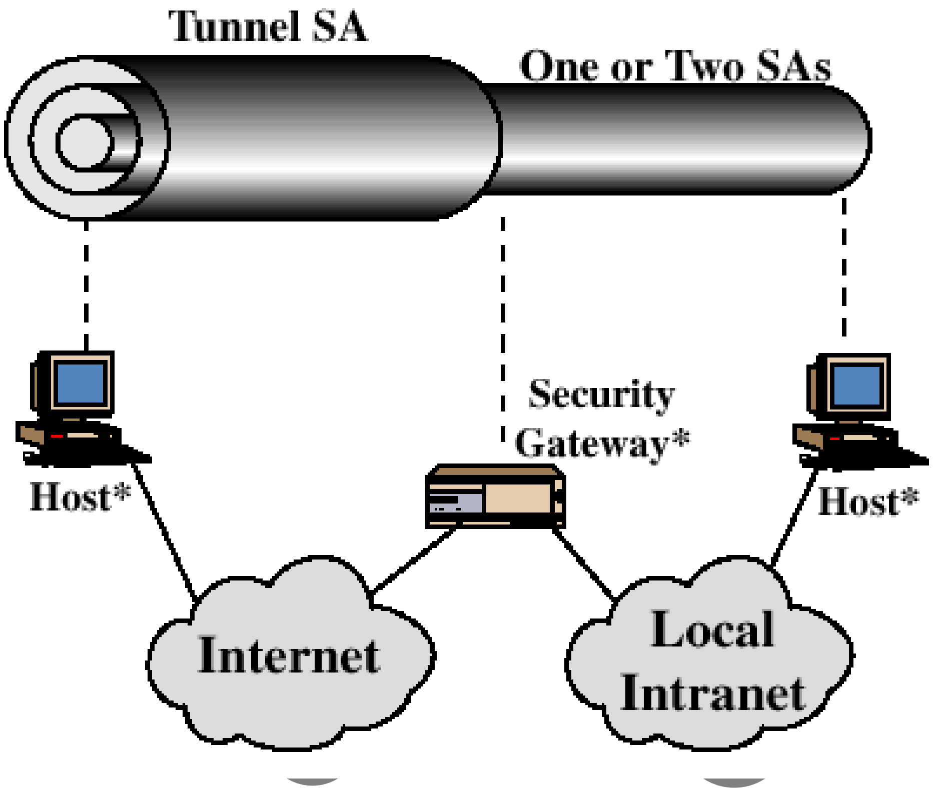
- Builds on the previous case adding end-to-end security
- All combinations of the above cases



IPSec

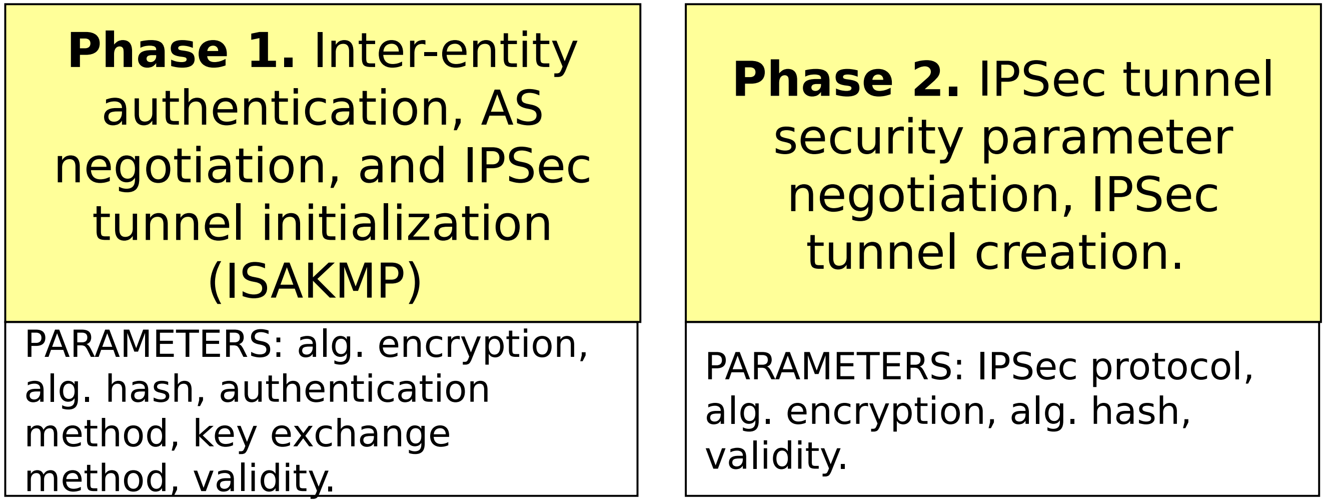
BASIC COMBINATIONS OF SECURITY ASSOCIATIONS

- Support for a remote host that wants to access a firewalled organization and then access some server behind the firewall



IPSec

- Formation of security associations
  - Two-phase IKE protocol



Phase 1. Inter-entity authentication, AS negotiation, and IPSec tunnel initialization (ISAKMP)

PARAMETERS: alg. encryption, alg. hash, authentication method, key exchange method, validity.

Phase 2. IPSec tunnel security parameter negotiation, IPSec tunnel creation.

PARAMETERS: IPSec protocol, alg. encryption, alg. hash, validity.