Information Security
CS 3002

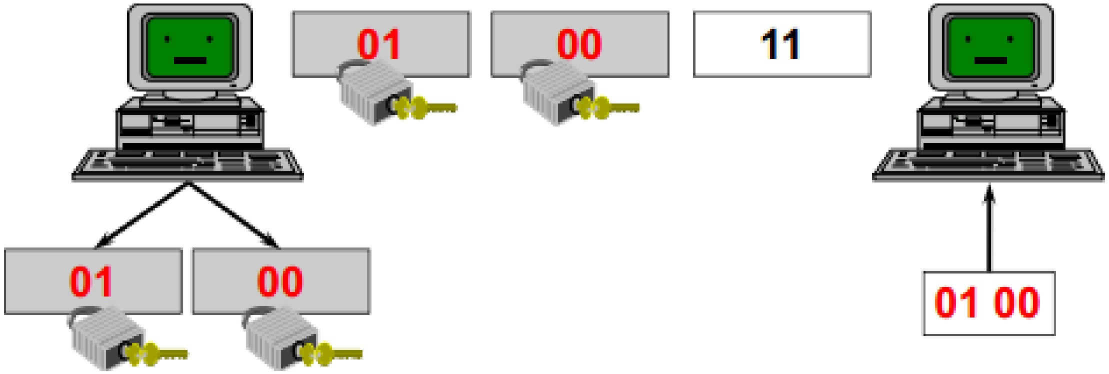
Dr. Haroon Mahmood

Assistant Professor

NUCES Lahore

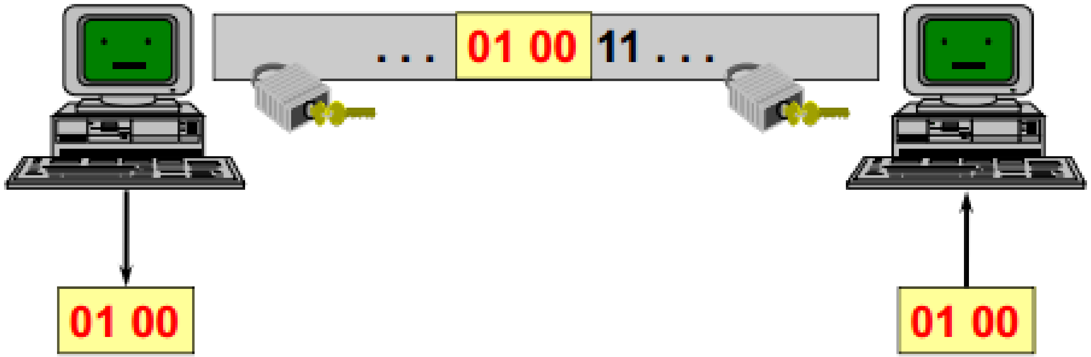
Message/data security

- authentication (single), integrity and privacy self contained in the message
- possibility of non repudiation
- requires modification of applications



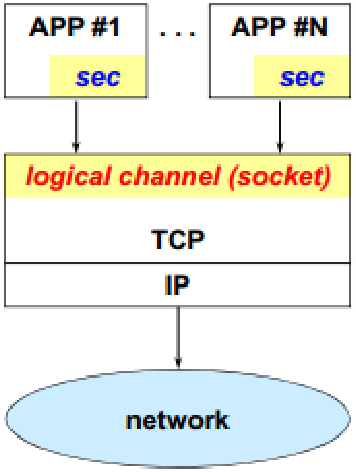
Channel Security

- authentication (single or mutual), integrity and privacy only during the transit inside the communication channel
- no possibility of non repudiation
- requires no (or small) modification of applications



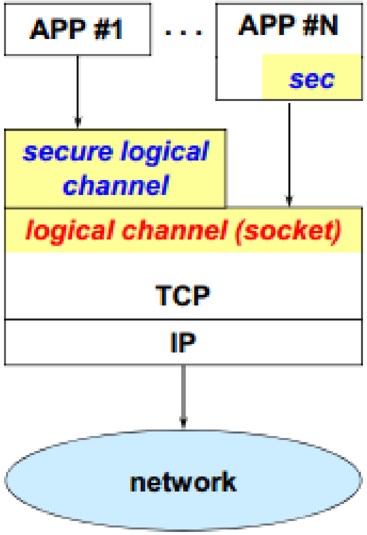
Security internal to applications

- each application implements security internally
- the common part is limited to the communication channels (socket)
- possible implementation errors (inventing security protocols is not simple!)
- does not guarantee interoperability



Security External to Applications

- the session level would be the ideal one to be used to implement many security functions
- … but it does not exist in TCP/IP!
- a “secure session” level was proposed:
  - it simplifies the work of application developers
  - it avoids implementation errors
  - it is up to the application to select it (or not)

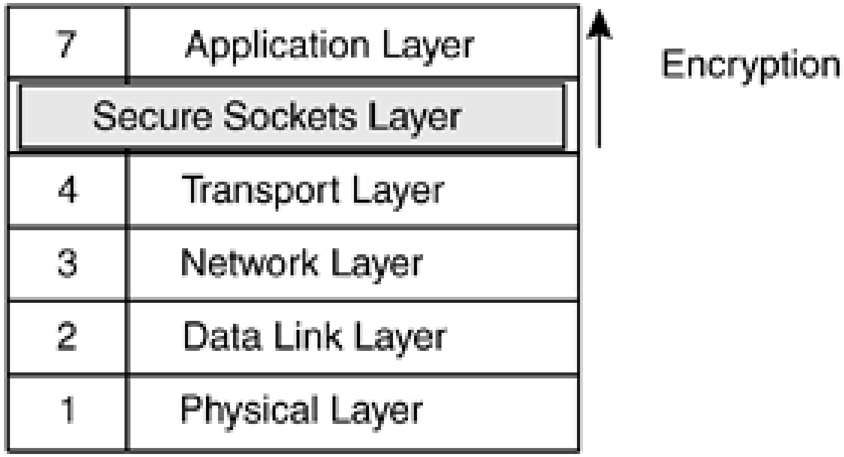


SSL what is it?

- Security at layer 4 (Transport layer)
- Secure Sockets Layer (SSL)
- secure transport channel (session level):
  - peer authentication (server, server+client)
  - message confidentiality
  - message authentication and integrity
  - protection against replay attacks
- Easily applicable to all protocols based on TCP:
  - HTTP, SMTP, FTP, TELNET, ...
  - e.g. the famous secure HTTP (https://....) = 443/TCP

SSL/TLS

- Philosophy of SSL: Easier to deploy something if no changes in OS required
- Application's API (Socket) is interface to SSL: Hence secure socket layer
- API to SSL is the superset of API to TCP
- SSL/TLS operate above TCP. OS doesn't change applications do!



7 Application Layer

Secure Sockets Layer

4 Transport Layer

3 Network Layer

2 Data Link Layer

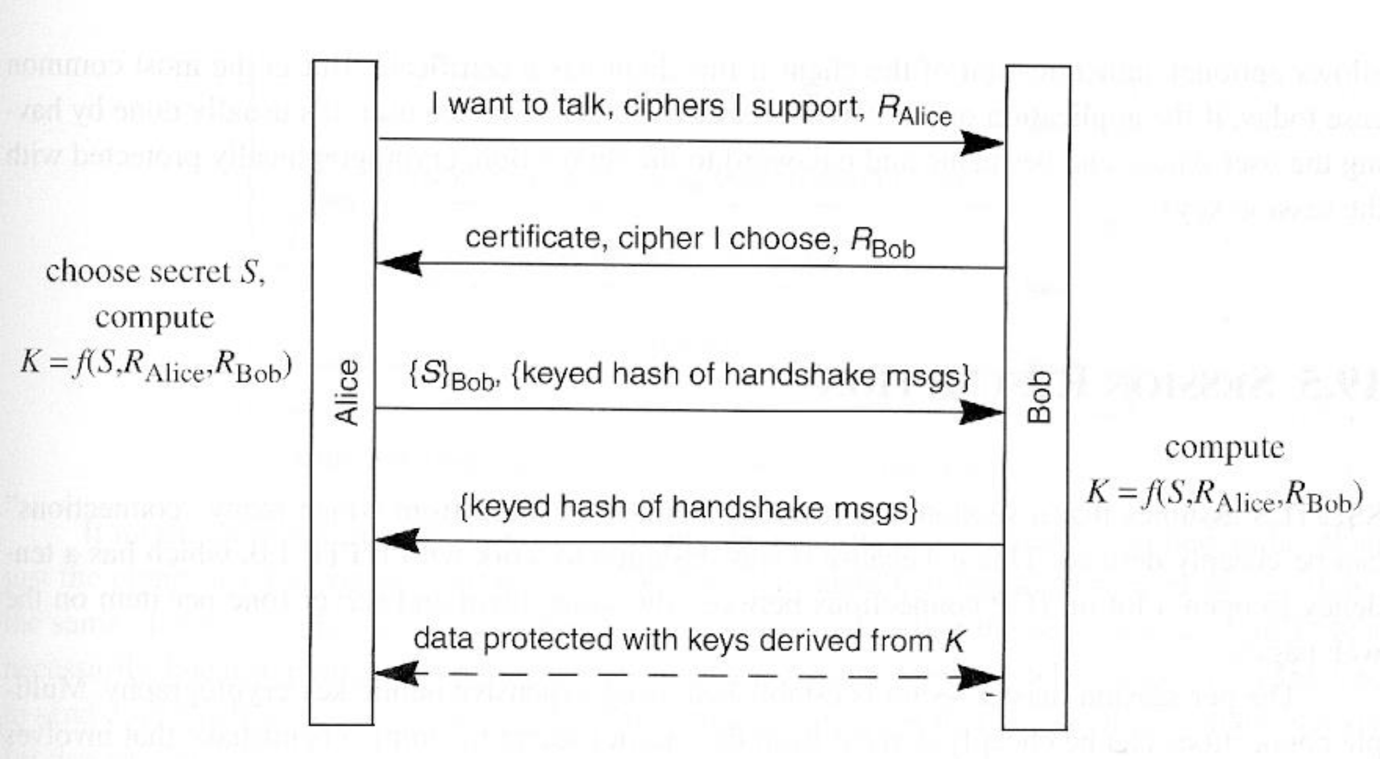
1 Physical Layer

Encryption

SSL handshake

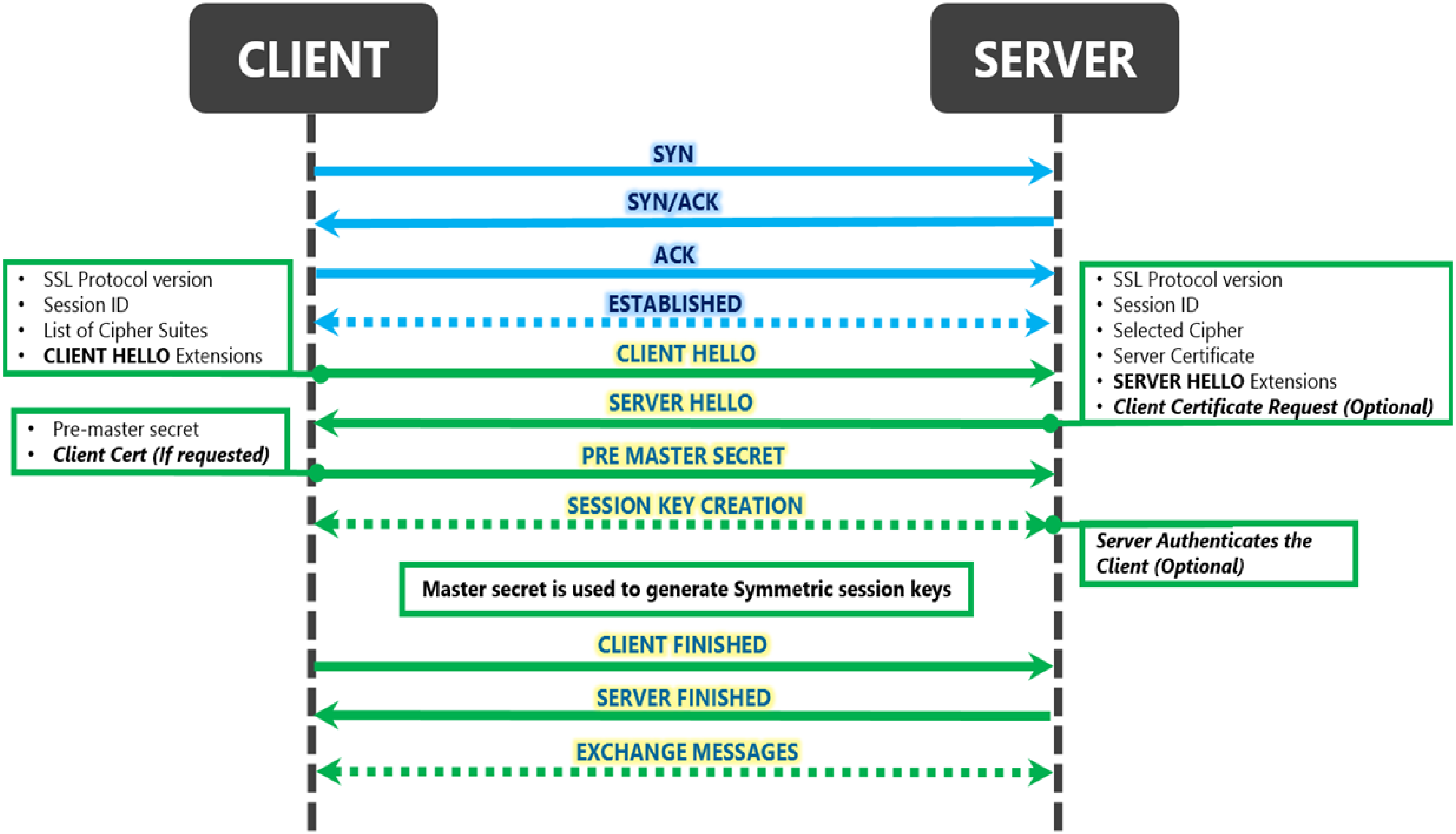
- agree on a set of algorithms for confidentiality, integrity and authentication
- exchange random numbers between the client and the server to be used for the subsequent generation of the keys
- establish a symmetric key by means of public key operations, e.g. RSA
- negotiate the session-id
- exchange the necessary certificates

SSL handshake simplified


I want to talk, ciphers I support, R_{Alice}
certificate, cipher I choose, R_{Bob}
choose secret S, compute $K = f(S, R_{Alice}, R_{Bob})$
Alice
Bob
$\{S\}_{Bob}$, {keyed hash of handshake msgs}
compute $K = f(S, R_{Alice}, R_{Bob})$
{keyed hash of handshake msgs}
data protected with keys derived from K

- **Secrets are: Pre-master key S, Master Key K**
- **Server authentication**
- **Client authentication by password (optional)**

SSL handshake in detail



- SSL Protocol version
- Session ID
- List of Cipher Suites
- **CLIENT HELLO** Extensions

- SSL Protocol version
- Session ID
- Selected Cipher
- Server Certificate
- **SERVER HELLO** Extensions
- ***Client Certificate Request (Optional)***

- Pre-master secret
- ***Client Cert (If requested)***

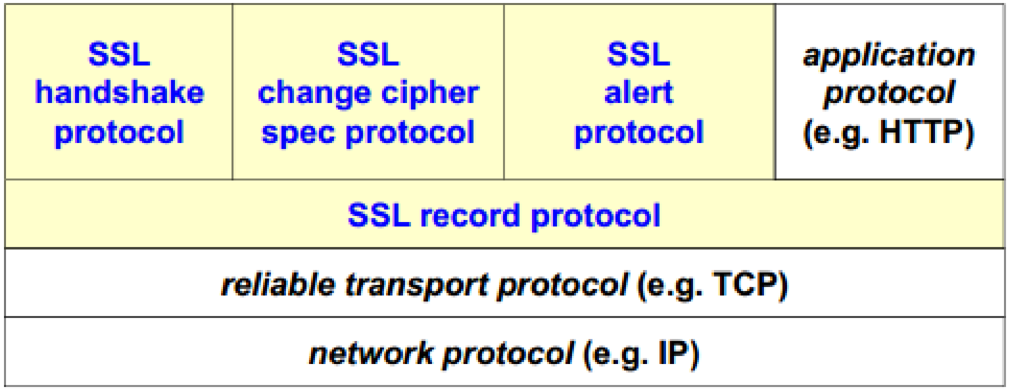
Server Authenticates the Client (Optional)

Master secret is used to generate Symmetric session keys

Key Terms

- HELLO Extensions: request extended functionality by sending data in the extensions field.
  - E.g: max_fragment_length, status request
  - The server may not oblige
  - Client may abort the handshake
- Pre-shared Secret (key): generated by client OR directly obtained from the key exchange. E.g: (DH: g^{ab} mod p)
- Master keys: generated from the pre-shared secret + random.client + random.server by applying a PRF
- Master key = PRF (pre-shared secret, random.client, random.server)
- PRF = Pseudo Random Function

SSL V3 Architecture



SSL handshake protocol

SSL change cipher spec protocol

SSL alert protocol

application protocol (e.g. HTTP)

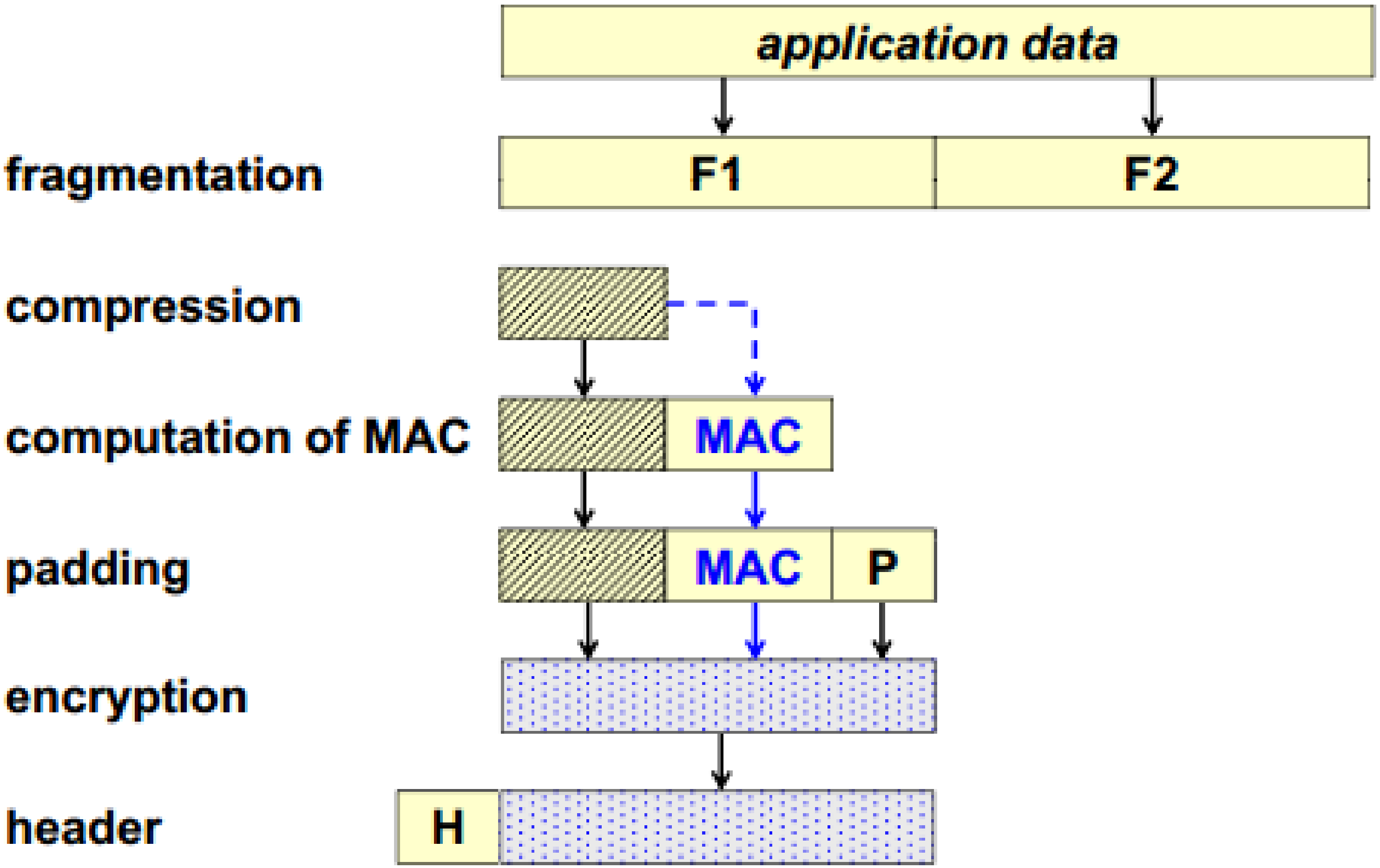
SSL record protocol

reliable transport protocol (e.g. TCP)

network protocol (e.g. IP)

- **Handshake: enables the SSL or TLS client and server to establish the secret keys with which they communicate**
- **Change cipher spec: indicates the usage of secret key for data communication**
- **Alert: signal problems with SSL connection, give current status**
- **Record protocol: permits the encapsulation of higher level protocols**

SSL3/TLS record protocol



application data

fragmentation

F1

F2

compression

computation of MAC

MAC

padding

MAC

P

encryption

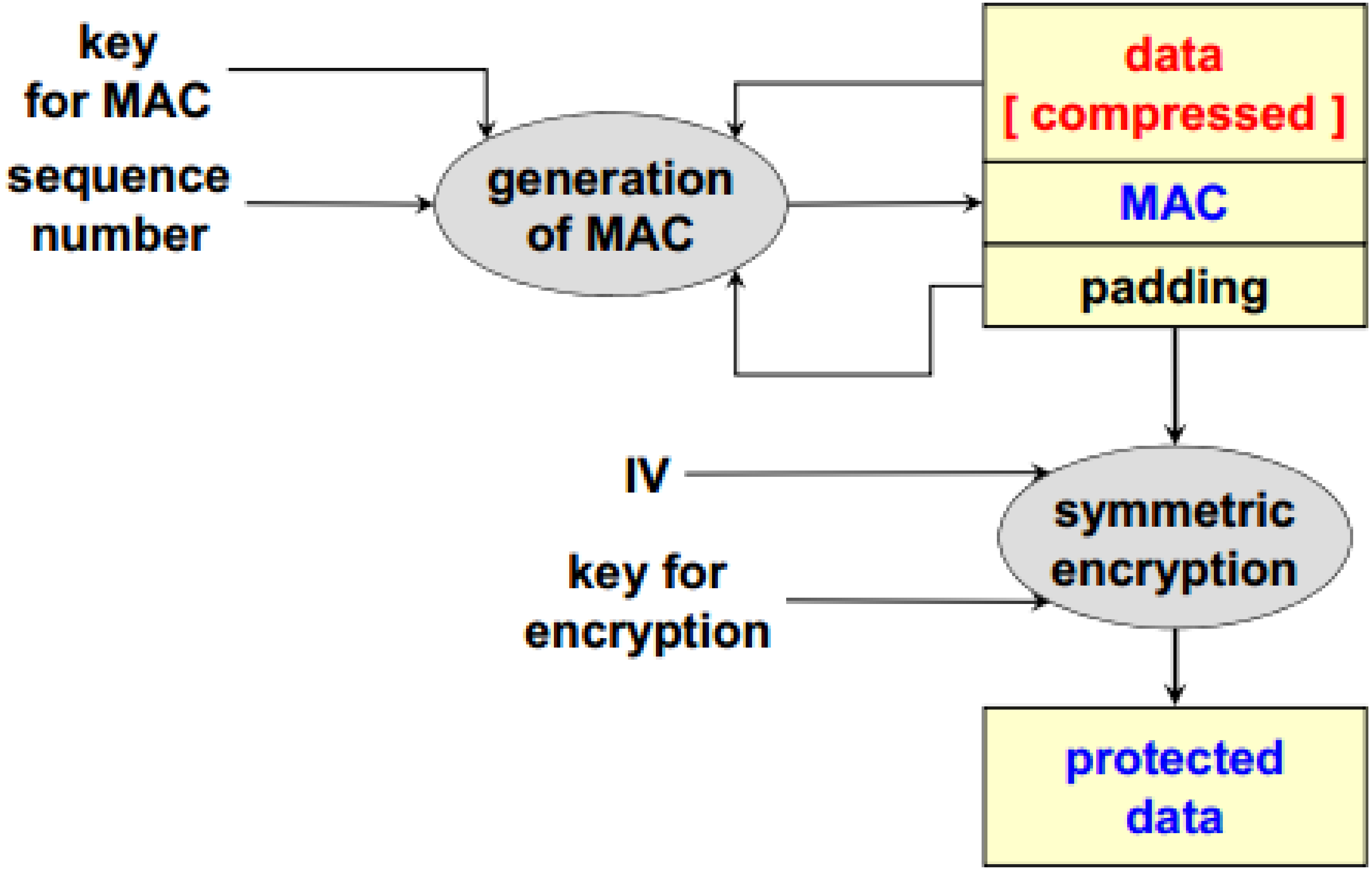
header

H

SSL MAC computation

- MAC = message_digest (key, seq_number | type | version | length | fragment)
- message_digest
  - depends on the chosen algorithm
- key
  - sender-write-key or receiver-read-key
- seq_number
  - 32-bit integer
- Type
  - Type of record
    - change cipher spec (20)
    - alert (21)
    - Handshake (22)
    - Application data (23)
- length
  - length of the fragment/plaintext

Data protection in SSL



SSL-3 new features with respect to SSL-2

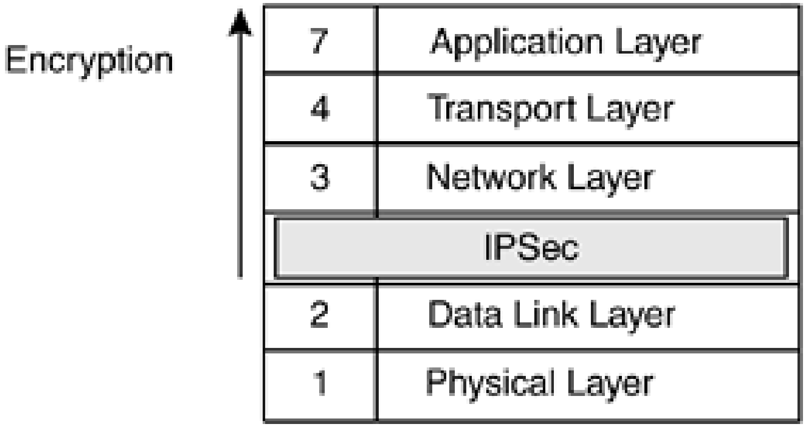
- data compression:
  - optional
  - Done before encryption

- data encryption is optional: in order to have only authentication and integrity

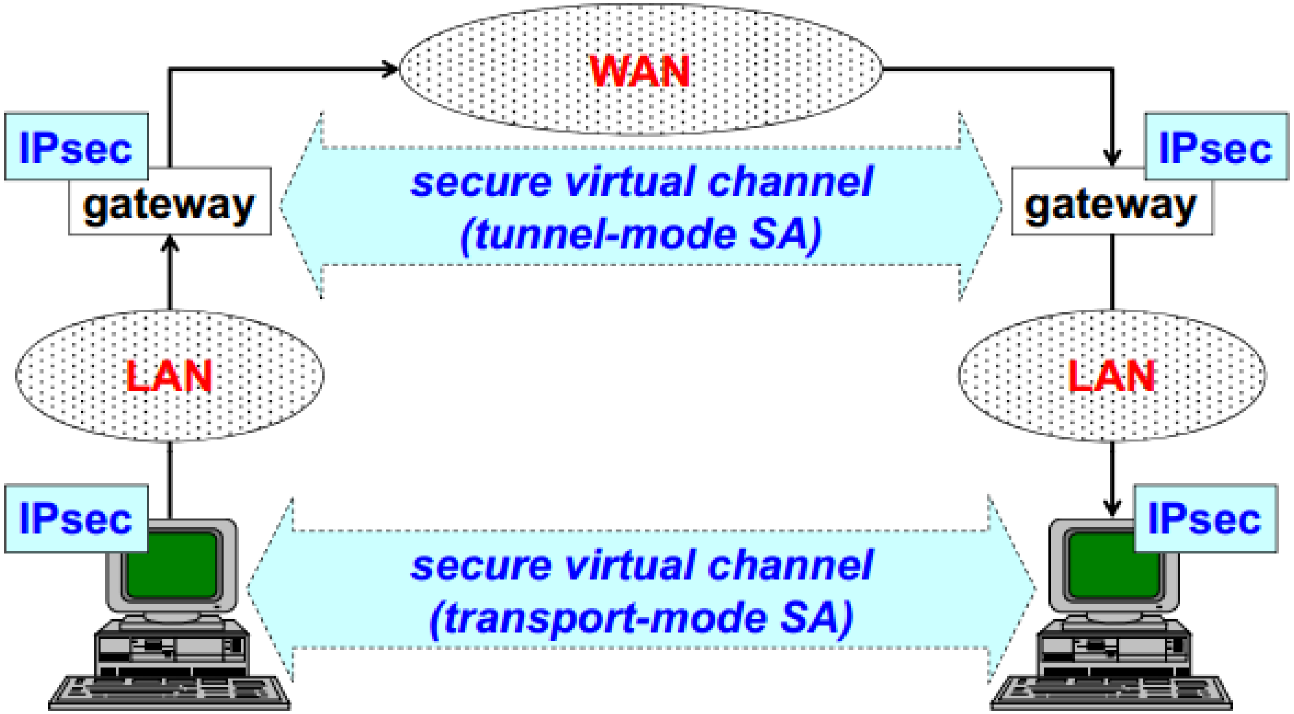
- possibility to re-negotiate the SSL connection:
  - periodical change of keys
  - change of the algorithms

IPsec

- Philosophy of IPsec: implementing security within the operating systems automatically causes applications to be protected without changing applications
- IPsec is within the OS. OS changes, applications and API to TCP don't.

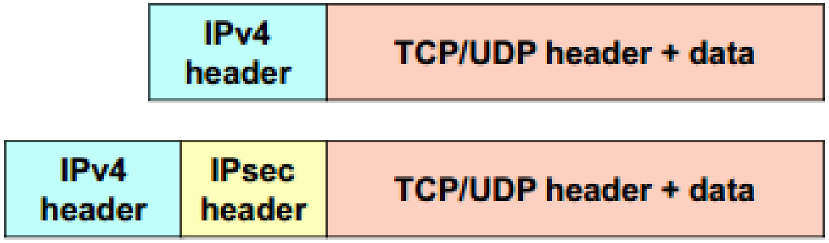


End-to-end security with basic VPN



Transport mode IPsec

- used for end-to-end security, that is used by hosts, not gateways (exception: traffic for the gateway itself e.g: SNMP, ICMP)
- pro: computationally light
- con: no protection of header variable fields

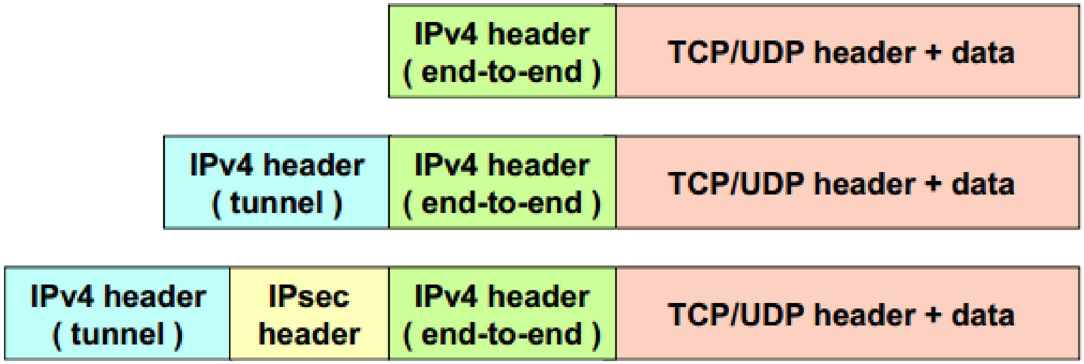


IPv4 header | TCP/UDP header + data

IPv4 header | IPsec header | TCP/UDP header + data

Tunnel Mode IPsec

- used to create a VPN, usually by gateways
- Gateway-to-gateway mode
- pro: protection of header variable fields
- con: computationally heavy



IPv4 header (end-to-end) | TCP/UDP header + data

IPv4 header (tunnel) | IPv4 header (end-to-end) | TCP/UDP header + data

IPv4 header (tunnel) | IPsec header | IPv4 header (end-to-end) | TCP/UDP header + data

IPsec

- Security at layer 3
- IPsec ensures:
  - Confidentiality, integrity, and authenticity
- Allows secure communication in the Internet
- Independent from the application or higher protocols
- Network-layer security instead of application-layer security
  - Compatible with schemes providing security at the application layer
    - Can be applied simultaneously

IPSec

- Further advantages:
  - Can be applied to all network traffic
  - Routers/firewalls vendors can implement it (Can't implement SSL)
  - Transparent to the applications
  - Transparent to the users

- Limitations:
  - Limited to IP Addresses
  - Has no concept of application users

Applications of IPsec

- Secure connection among different branches of the same company
  - Virtual Private Network (VPN)

- Secure remote access to an Intranet through the (insecure) Internet
  - Allows secure remote workers

- Secure communication between peers

- Adding security for electronic commerce applications

IPsec overview

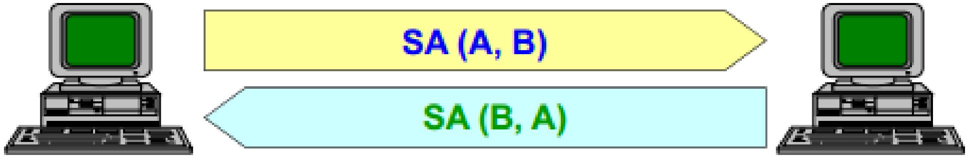
- IETF architecture for L3 security in IPv4 / IPv6:

- definition of two specific packet types:
  - AH (Authentication Header)
    - for integrity, authentication, no replay
  - ESP (Encapsulating Security Payload)
    - for confidentiality, integrity, authentication, no replay

- protocol for key exchange:
  - IKE (Internet Key Exchange)

Security Association

- Establishment of shared security attributes between sender and receiver to support secure communication
- Usually considered unidirectional
- contain all the information required for execution of various network security services
- Three SA identification parameters
  - Security parameter index
  - IP destination address
  - Security protocol identifier



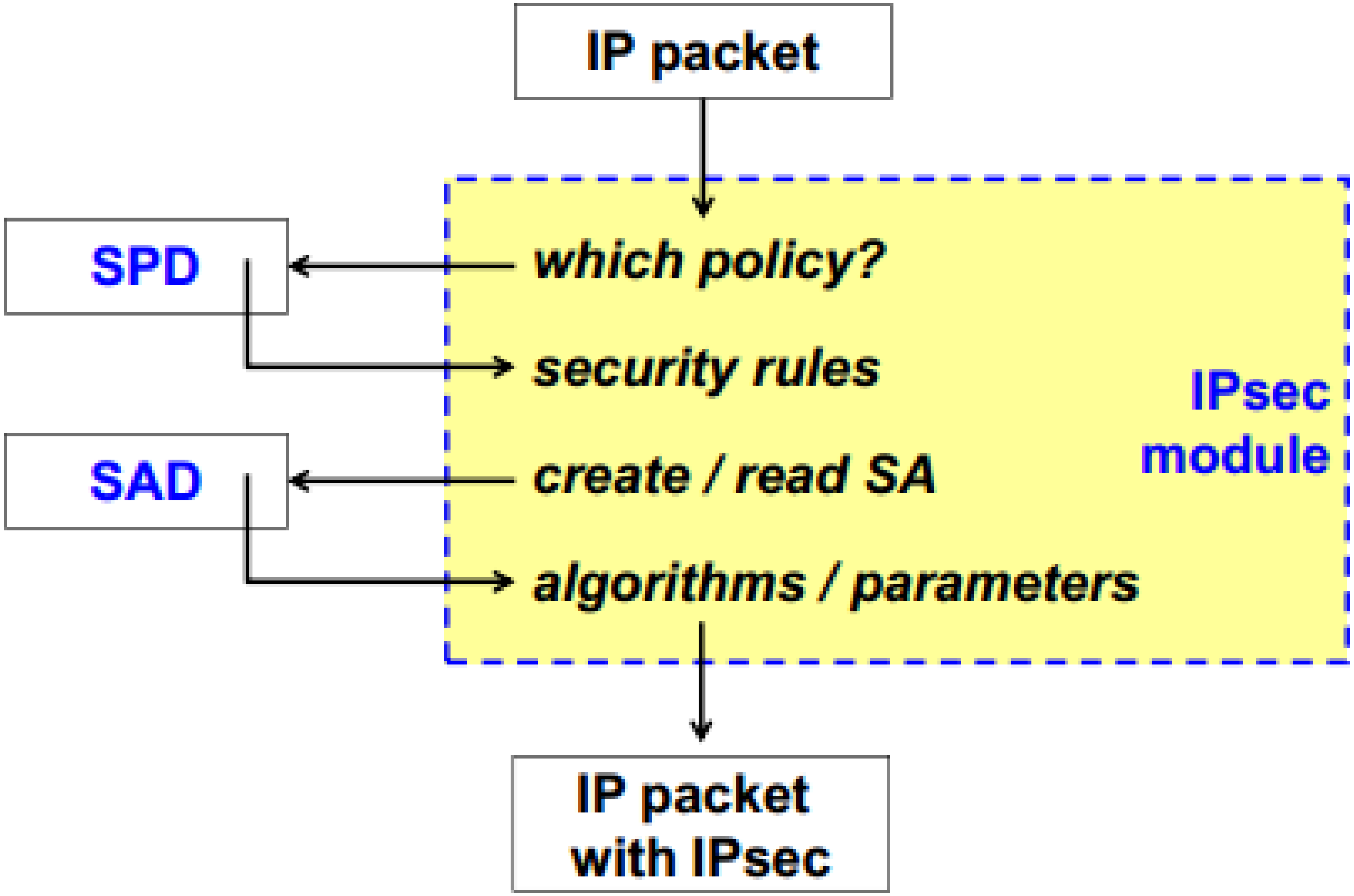
- two SA are needed to get complete protection of a bidirectional packet flow in IPsec

IPsec local databases

- SAD (SA Database)
  - list of active SA and their characteristics (algorithms, keys, parameters)
  - maintained by user-processes

- SPD (Security Policy Database)
  - list of security policies to apply to the different packet flows
  - a-priori configured (e.g. manually) or connected to an automatic system (e.g. ISPS, Internet Security Policy System)

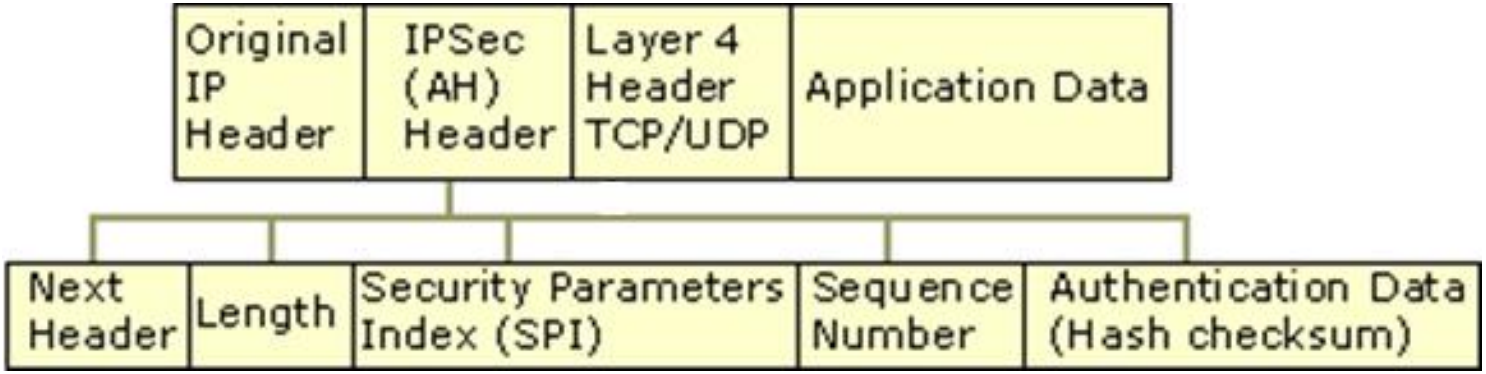
How IPsec works (sending)



AH

- Authentication Header
- mechanism (first version, RFC-1826):
  - data integrity and sender authentication
  - compulsory support of keyed-MD5 (RFC-1828)
  - optional support of keyed-SHA-1 (RFC-1852)
- mechanism (second version, RFC-2402):
  - data integrity, sender authentication and protection from replay attack
  - HMAC-MD5
  - HMAC-SHA-1

AH Packet

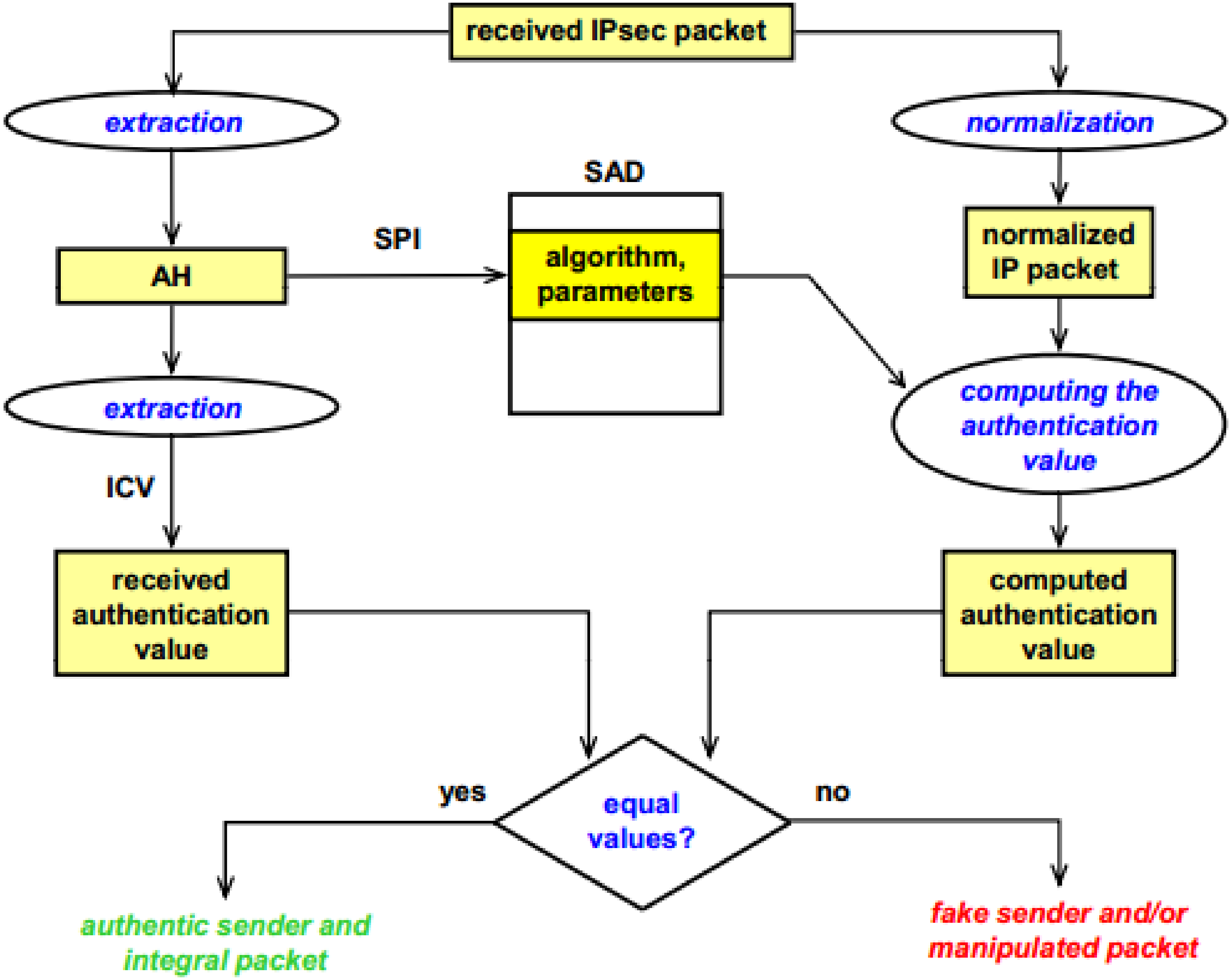


Original IP Header | IPSec (AH) Header | Layer 4 Header TCP/UDP | Application Data

Next Header | Length | Security Parameters Index (SPI) | Sequence Number | Authentication Data (Hash checksum)

- **Next header: identifies the nature of the payload (TCP/UDP)**
- **Length: Indicates the length of the AH header**
- **SPI: Identifies the correct security association for the communication**
- **Sequence Number: Provides anti-replay protection for the SA**
- **Auth. Data: contains the Integrity Check Value (ICV) that is used to verify the integrity of the message. The receiver calculates the hash value and checks it against this value (calculated by the sender) to verify integrity.**

AH verification


received IPsec packet
extraction
normalization
SAD
SPI
AH
algorithm, parameters
normalized IP packet
extraction
computing the authentication value
ICV
received authentication value
computed authentication value
yes
equal values?
no
authentic sender and integral packet
fake sender and/or manipulated packet

ESP

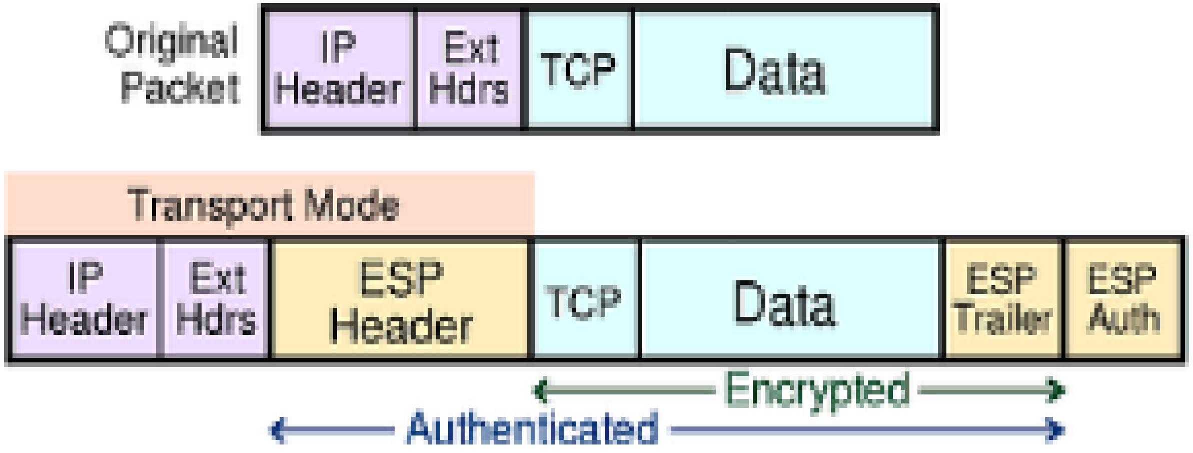
- Encapsulating Security Payload (ESP)

- first version (RFC-1827) gave only confidentiality
  - base mechanism: DES-CBC (RFC-1829)

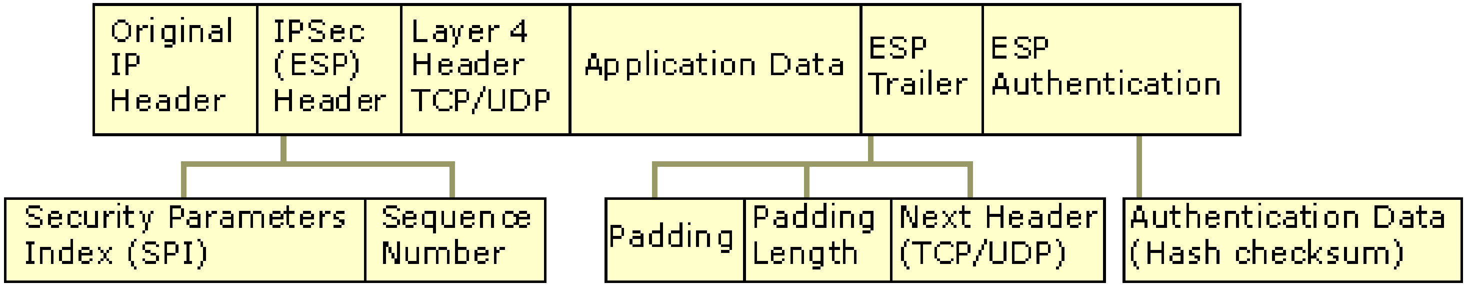
- Second version (RFC-2406):
  - provides confidentiality & authentication (but not the IP header, so the coverage is not equivalent to that of AH)

ESP in transport mode

- pro: the payload is hidden (including info needed for QoS or intrusion detection)
- con: the header remains in clear

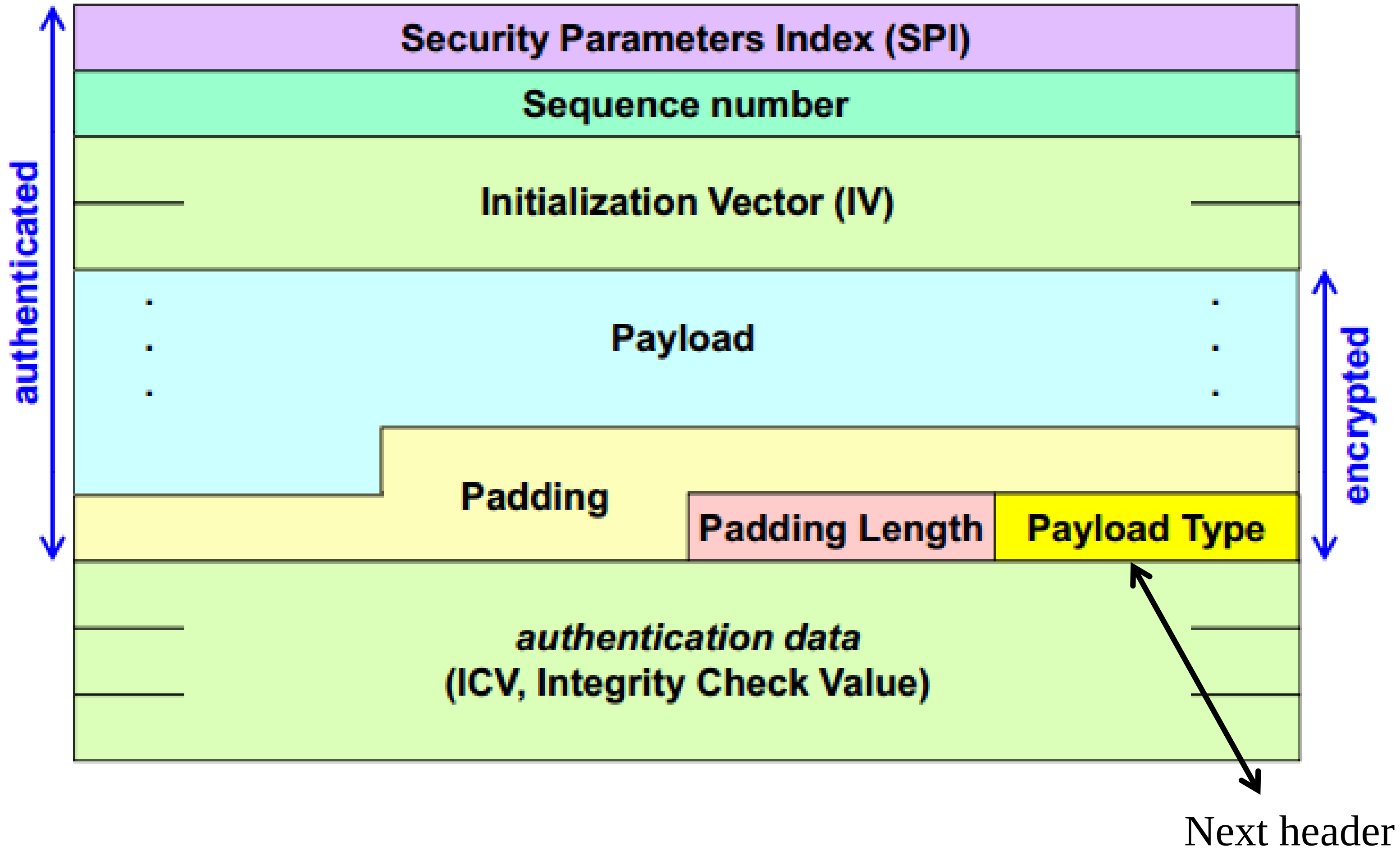


ESP Packet



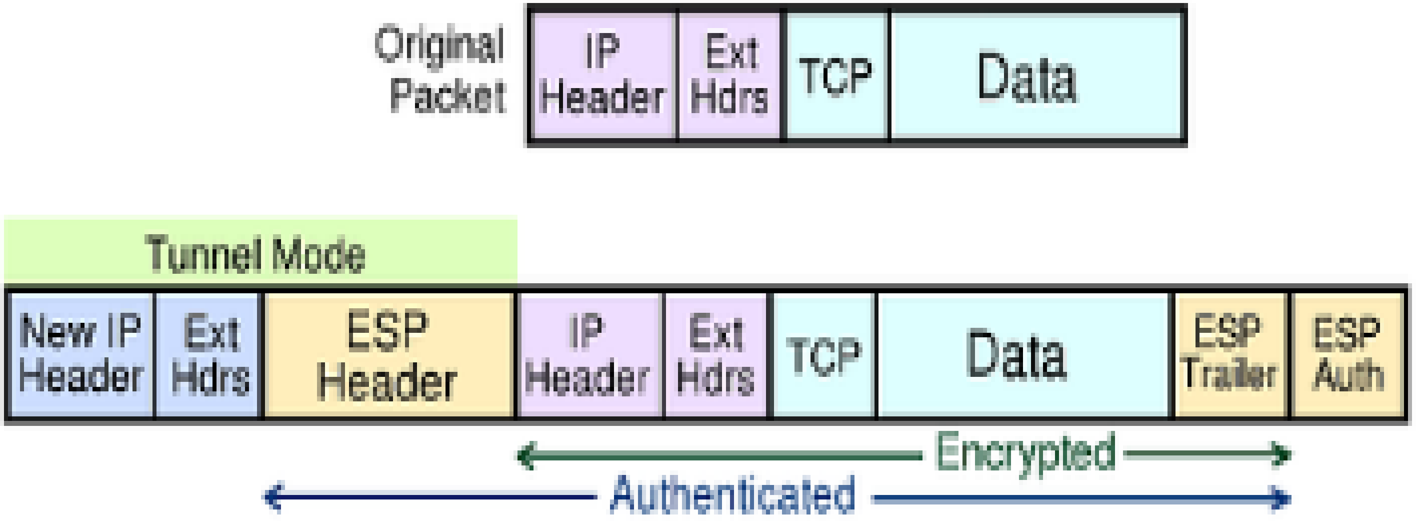
- **SPI: Identifies the correct security association for the communication**
- **Sequence number: Provides anti-replay protection for the SA**
- **Next header: Identifies the nature of the payload (TCP/UDP)**
- **Auth. Data: Contains the Integrity Check Value (ICV), and a message authentication code that is used to verify the sender's identity and message integrity. The ICV is calculated over the ESP header, the payload data and the ESP trailer**
- **Initialization Vector (IV): optional. Is after the Sequence number**

ESP Packet: Encryption & Authentication

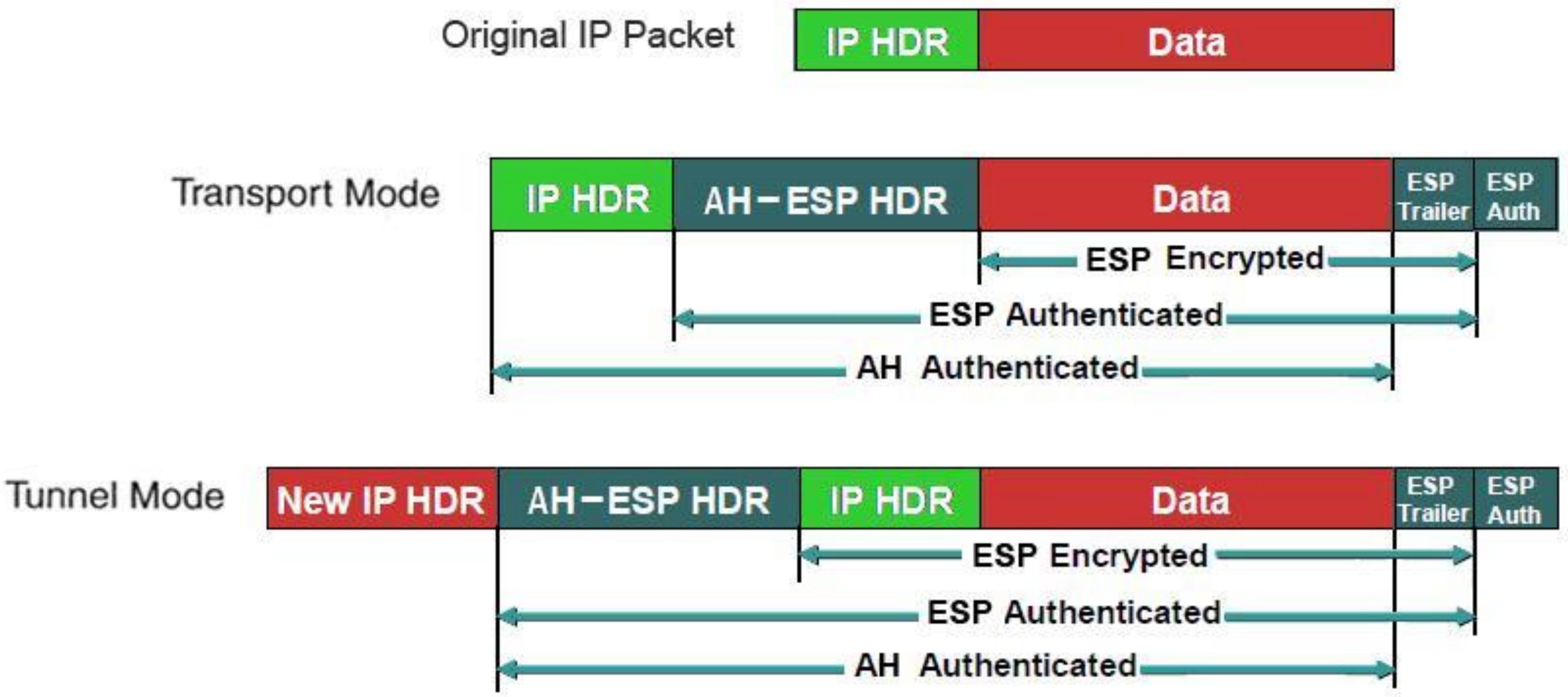


ESP tunnel mode

- pro: hides both the payload and (original) header
- con: larger packet size



IPSec: AH & ESP packet format


Original IP Packet
IP HDR
Data
Transport Mode
IP HDR
AH−ESP HDR
Data
ESP Trailer
ESP Auth
ESP Encrypted
ESP Authenticated
AH Authenticated
Tunnel Mode
New IP HDR
AH−ESP HDR
IP HDR
Data
ESP Trailer
ESP Auth
ESP Encrypted
ESP Authenticated
AH Authenticated