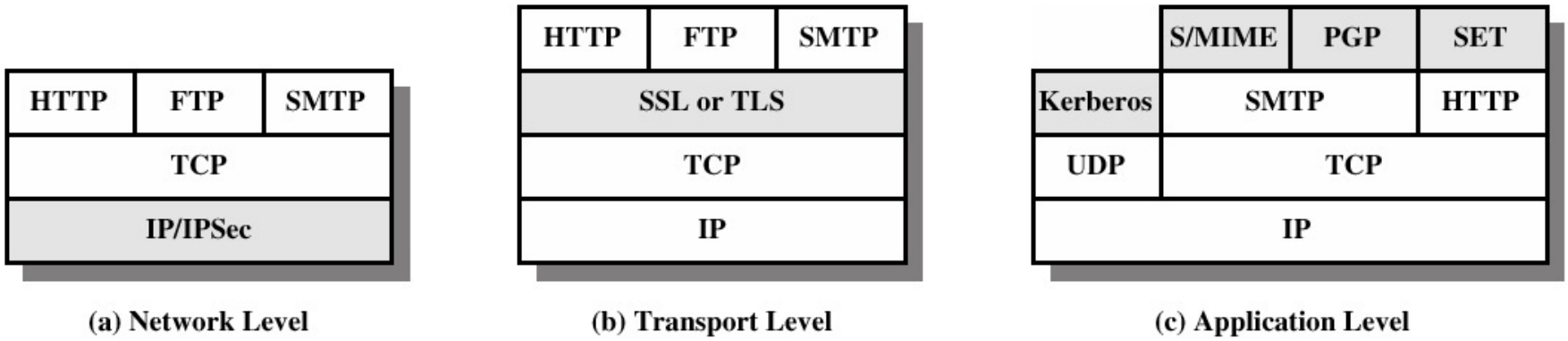
Secure Protocols

Different Secure Protocol Solutions at various layers of the Internet Protocol stack



Application Layer Solutions -- Secure Email

- Pretty Good Privacy (PGP)
- Secure MIME (S/MIME)

S/MIME

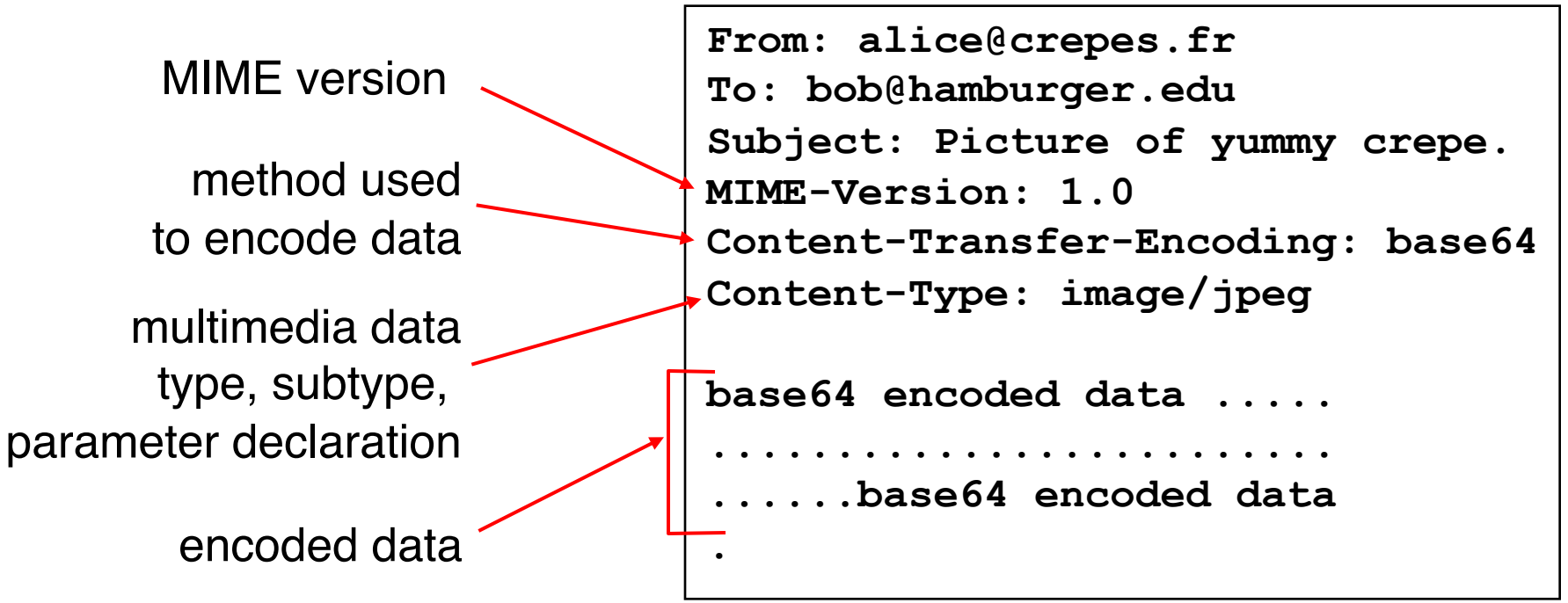
- Secure/Multipurpose Internet Mail Extension
- S/MIME will probably emerge as the industry standard.
- PGP for personal e-mail security

Simple Mail Transfer Protocol (SMTP, RFC 822)

- **SMTP Limitations - Cannot transmit, or has a problem with:**
  - executable files, or other binary files (jpeg image)
  - “international language” characters (non-ASCII)
  - messages over a certain size
  - lines longer than a certain length (72 to 254 characters)

Message format: multimedia extensions

- MIME: multimedia mail extension, RFC 2045, 2056
- additional lines in msg header declare MIME content type



MIME types

Text

- example subtypes: **`plain`**, **`html`**

Image

- example subtypes: **`jpeg`**, **`gif`**

Audio

- example subtypes: **`basic`** (8-bit mu-law encoded), **`32kadpcm`** **(32 kbps coding)**

Video

- example subtypes: **`mpeg`**, **`quicktime`**

Application

- other data that must be processed by reader before "viewable"
- example subtypes: **`msword`**, **`octet-stream`**

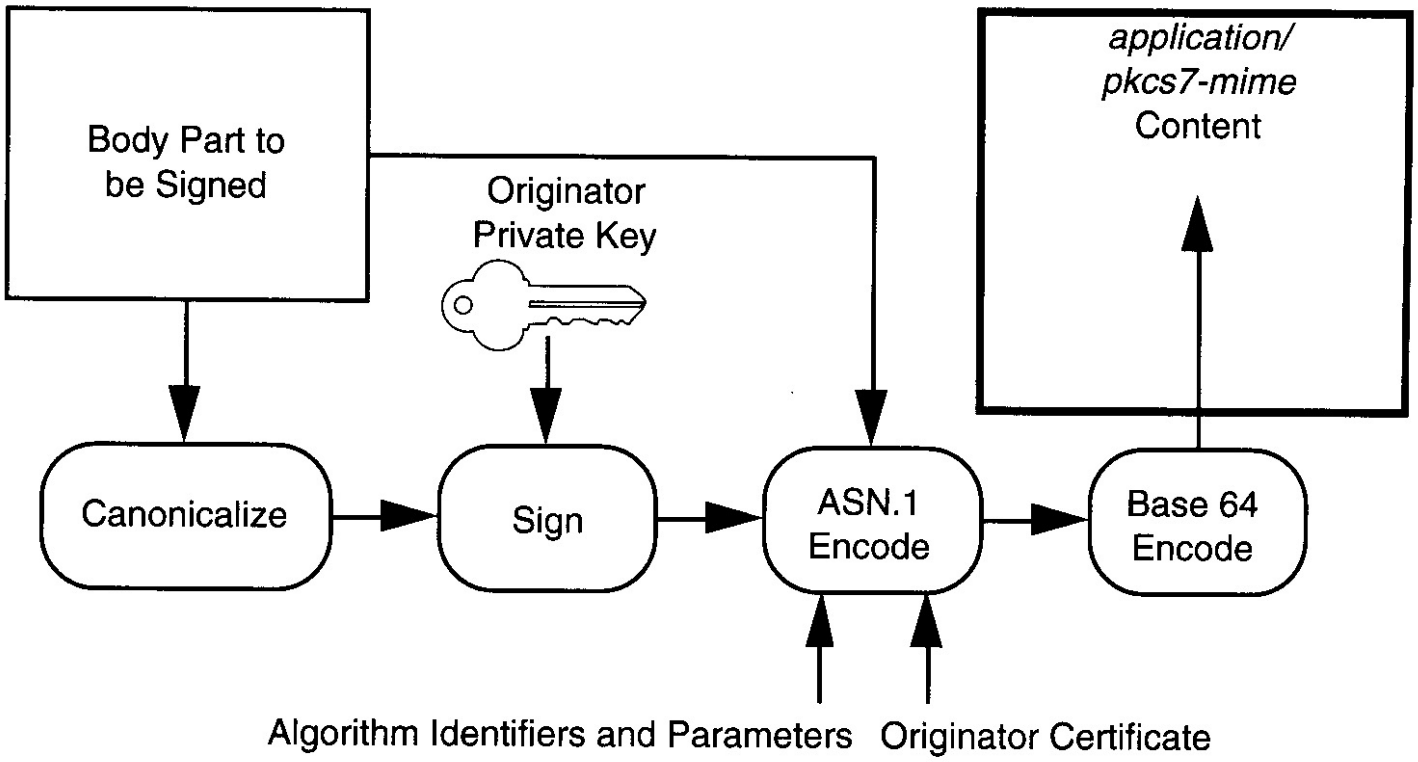
S/MIME Functions

- **Enveloped Data:** Encrypted content and encrypted session keys for recipients.
- **Signed Data:** Message Digest encrypted with private key of "signer" ; then encode the original message and digital signature using base64.
- **Clear-Signed Data:** Signed but the original message is not encoded in base64.
- **Signed and Enveloped Data:** Various orderings for encrypting and signing.

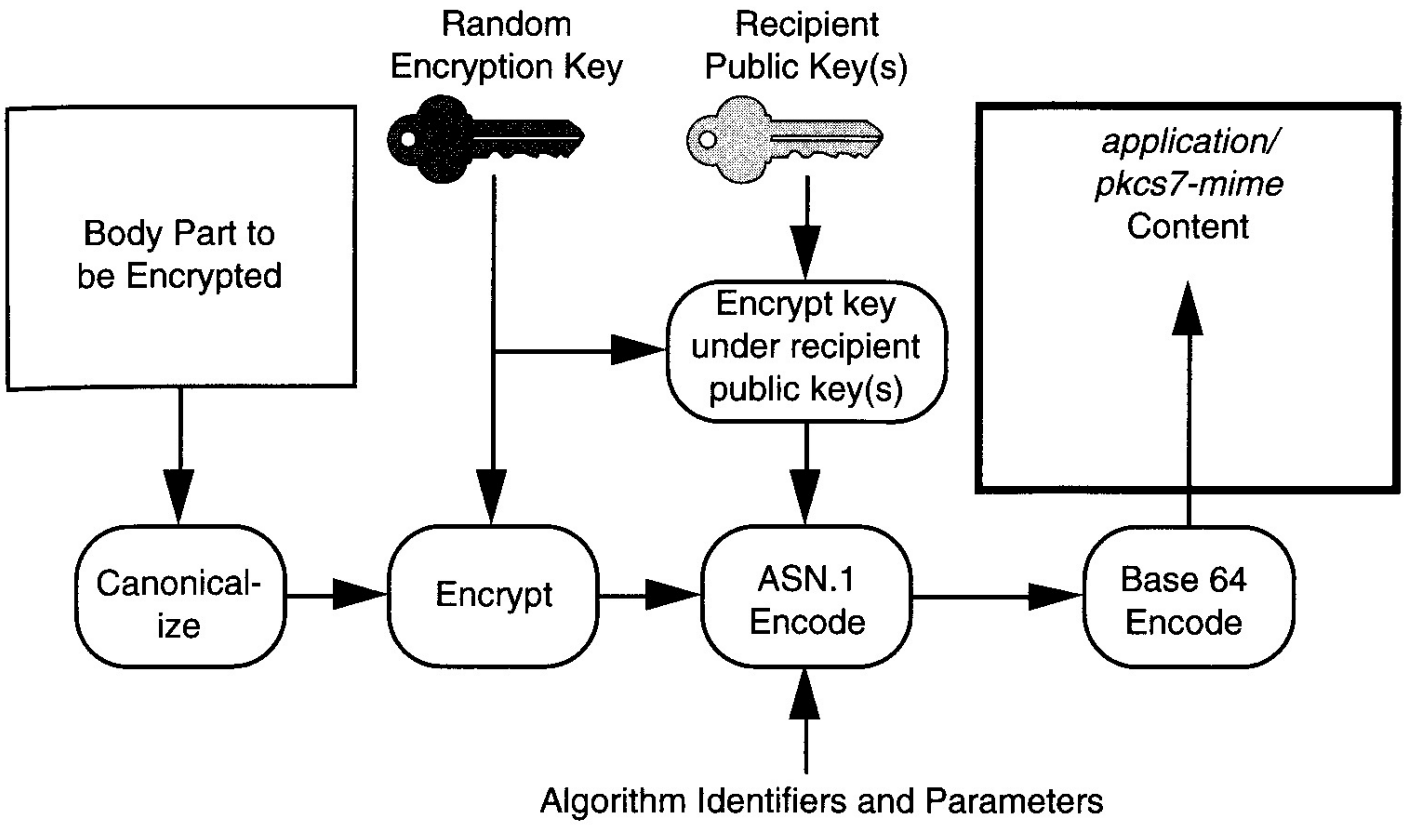
Algorithms Used

- **Message Digesting:** SHA-256 (MUST), SHA-1 and MD5 (SHOULD)
- **Digital Signatures:** DSS
- **Secret-Key Encryption:** Triple-DES, RC2/40 (exportable)
- **Public-Private Key Encryption:** RSA with key sizes of 512 and 1024 bits, and it can also use a variant of Diffie-Hellman, i.e. ElGamal, for the encryption/decryption of session keys.
- S/MIME uses Public-Key Certificates - X.509 version 3 signed by Certification Authority
  - **Registration** - Public keys must be registered with X.509 CA.

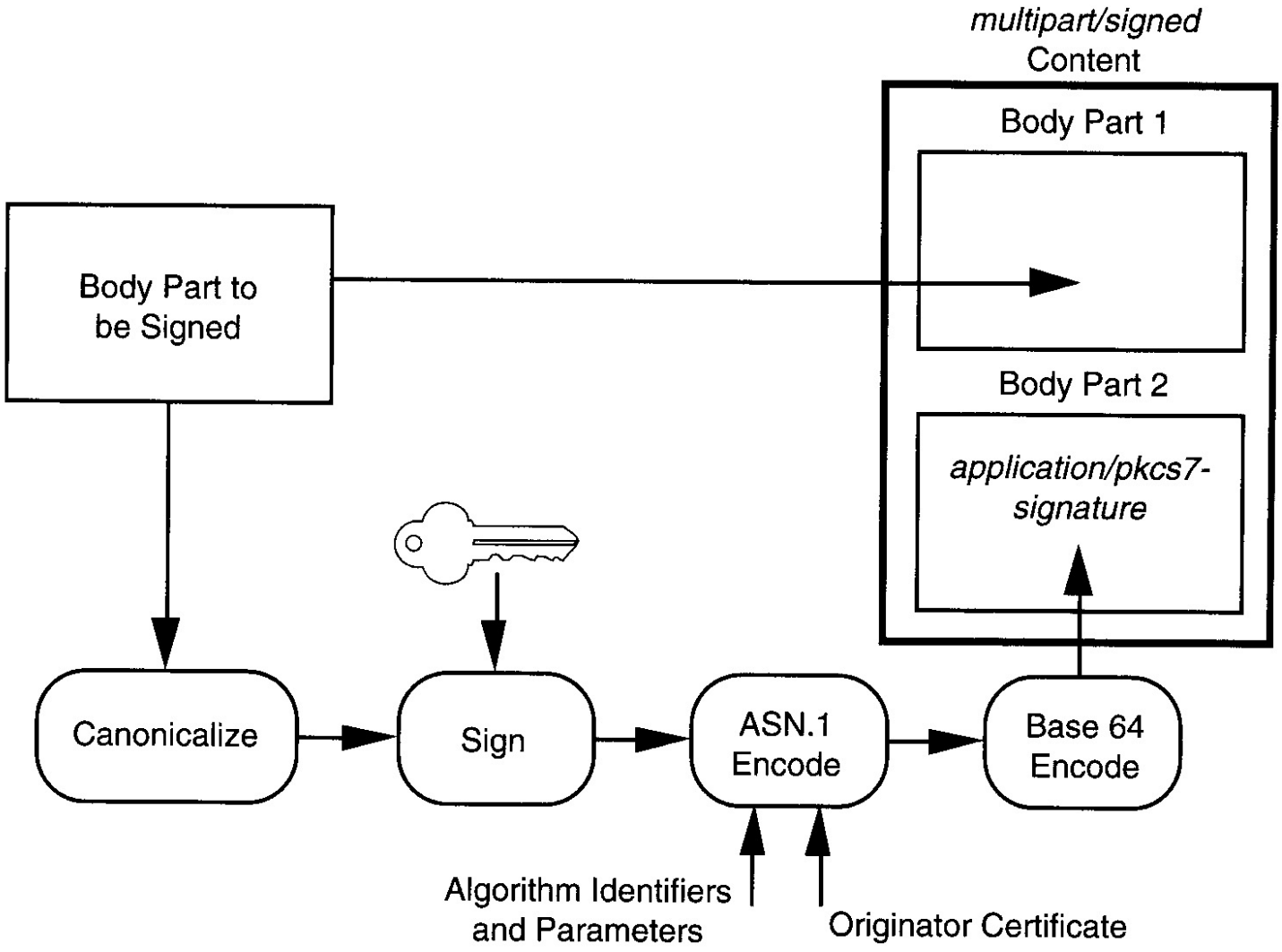
S/MIME Digital Signature Generation



S/MIME Encryption Process



S/MIME Digital Signature with multipart/signed

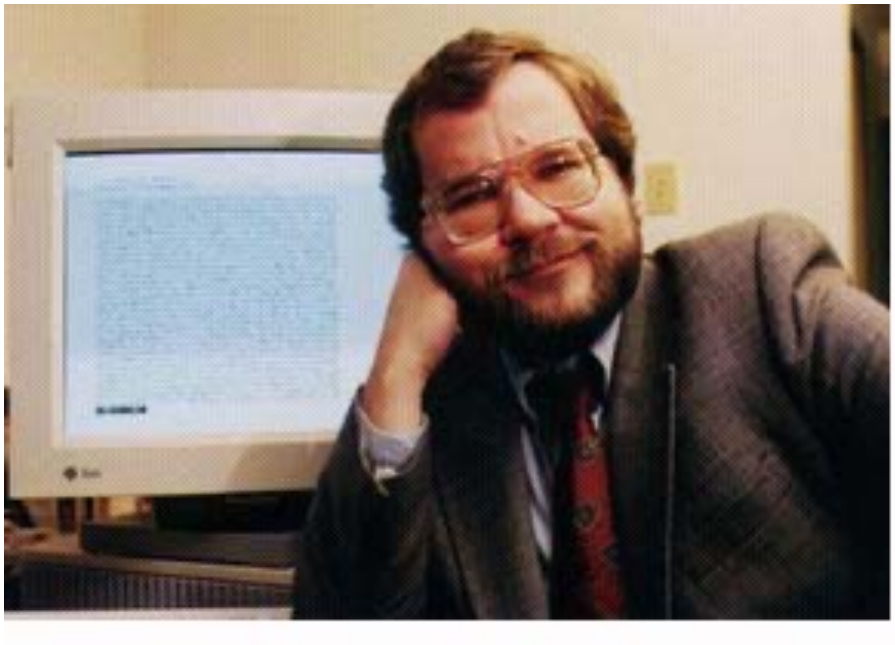


Pretty Good Privacy

- 1991 – Creation of a single person, Phil Zimmermann
- Provides confidentiality and authentication services for electronic mail and file storage applications
- Selected best available cryptographic algorithms
- Integrated these algorithms into a general purpose application
- Source code and doc freely available on the net
- Agreement with company (Viacrypt) for low cost commercial version

Phil Zimmermann

- Target of three year criminal investigation
- Gave software away to friend who put it on the Internet in 1991
- Intended to give individuals "the right to be let alone”
- US export restrictions violated – same class as ammunitions and nuclear weapons
- Government dropped the case in 1996



“PGP has spread like a prairie fire, fanned by countless people who fervently want their privacy restored in the information age”

- Phil Zimmermann, testifying before the US Senate, 1996

Summary of 5 PGP Services

authentication →

confidentiality →

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key, and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key, and included with the message.
Compression	ZIP	A message may be compressed, for storage or transmission, using ZIP.
Email compatibility	Radix 64 conversion	To provide transparency for email applications, an encrypted message may be converted to an ASCII string using radix 64 conversion.
Segmentation	—	To accommodate maximum message size limitations, PGP performs segmentation and reassembly.

Compression – Save Space

- PGP compresses (ZIP) the message after applying the signature but before encryption (default)
- Better to sign an uncompressed message
- Security is greater if message is encrypted after compression

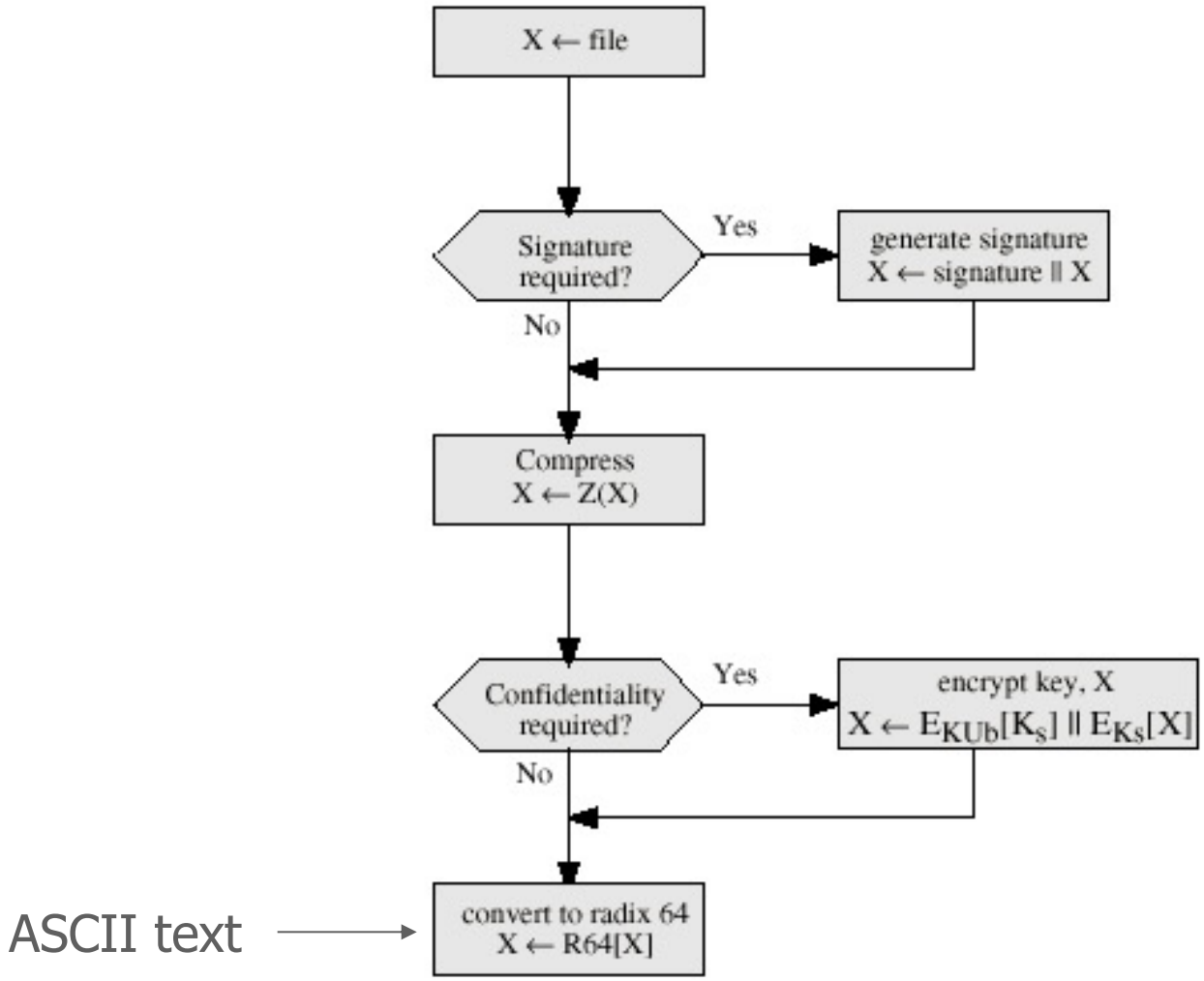
E-mail Compatibility

- Part or all of block consists of a stream of arbitrary 8-bit octets
- Many mail systems only allow ASCII text
- PGP converts raw binary stream to a stream of printable ASCII characters
- Radix-64 conversion – 3 binary => 4 ASCII

Stream Of Printable ASCII Chars

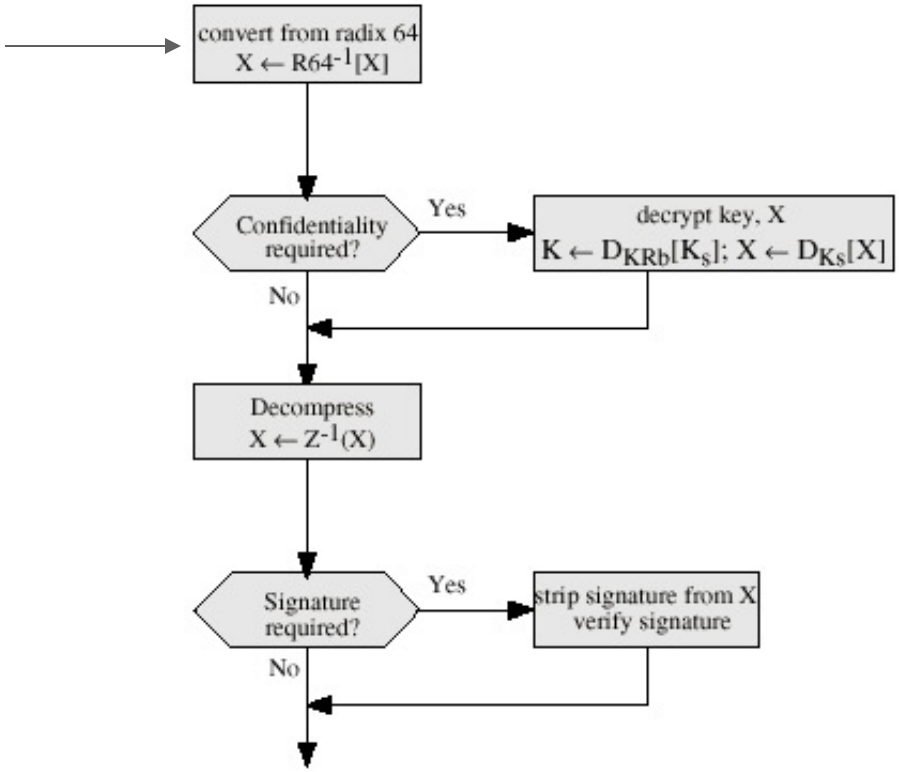
```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3i
mQBNAi23Dv0AAAECAMm6GNU3nqebKr3HW/fmrEhMlrFkwuZ6KHIYEat92nYfQIUj
lRLgj3TPHTRIMbswyTdaIJA7OvkSgxETLBCExX0ABRG0K0FuZHJlYXMgUmllZ2Vy
IDwxMDAxMTEuMzU0MEBjb21wdXNlcnZlLmNvbT4=
=8t7f
-----END PGP PUBLIC KEY BLOCK-----
```

Generic Transmission Diagram



Generic Reception Diagram

ASCII text
to binary



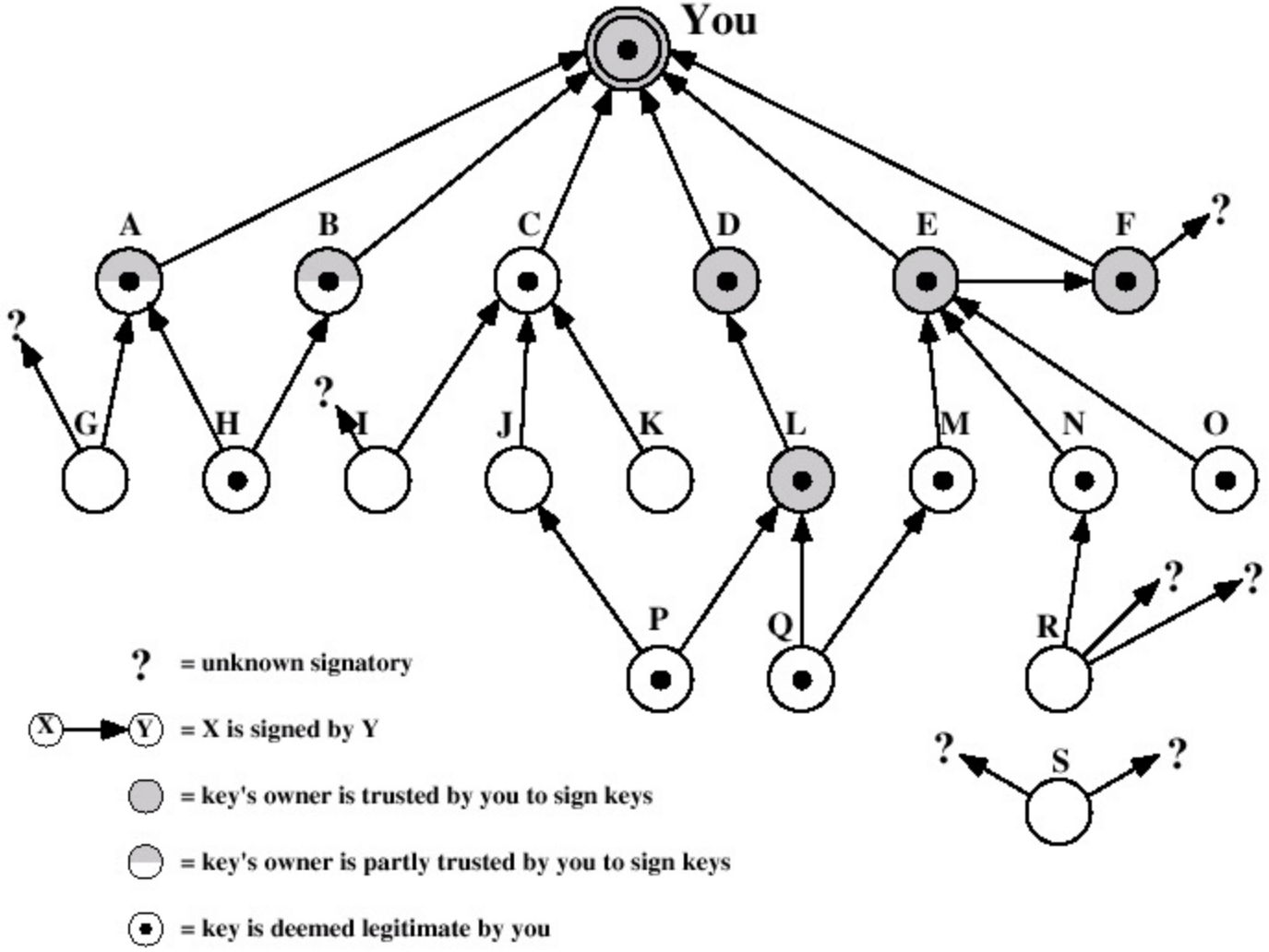
Segmentation

- Maximum message length restrictions in e-mail
- PGP automatically subdivides a large message into segments small enough to mail separately
- PGP reassembles entire original block at the receiving end

Summary of 5 PGP Services

- Authentication
- Confidentiality
- Compression
- E-Mail Compatibility
- Segmentation

PGP Trust Model Example



Security Solution as the Transport Layer: Secure Socket Layer (SSL) and Transport Layer Security (TLS)

SSL and TLS

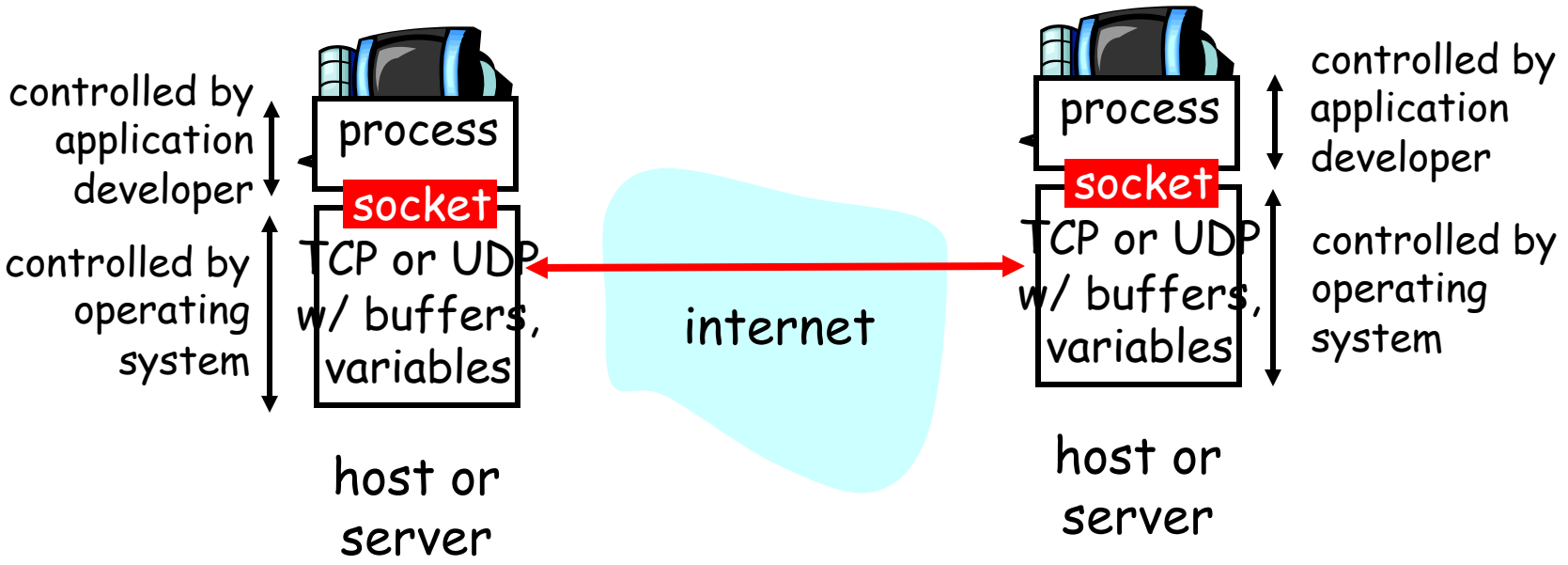
- SSL was originated by Netscape
- TLS working group was formed within IETF
- First version of TLS can be viewed as an SSLv3.1

Application and Operating System Interface --- the “Socket” interface

Socket: a door between application process and end-end-transport protocol (UCP or TCP)

TCP service: reliable transfer of bytes from one process to another

UDP service: unreliable transfer of datagrams from one process to another



SSL Architecture

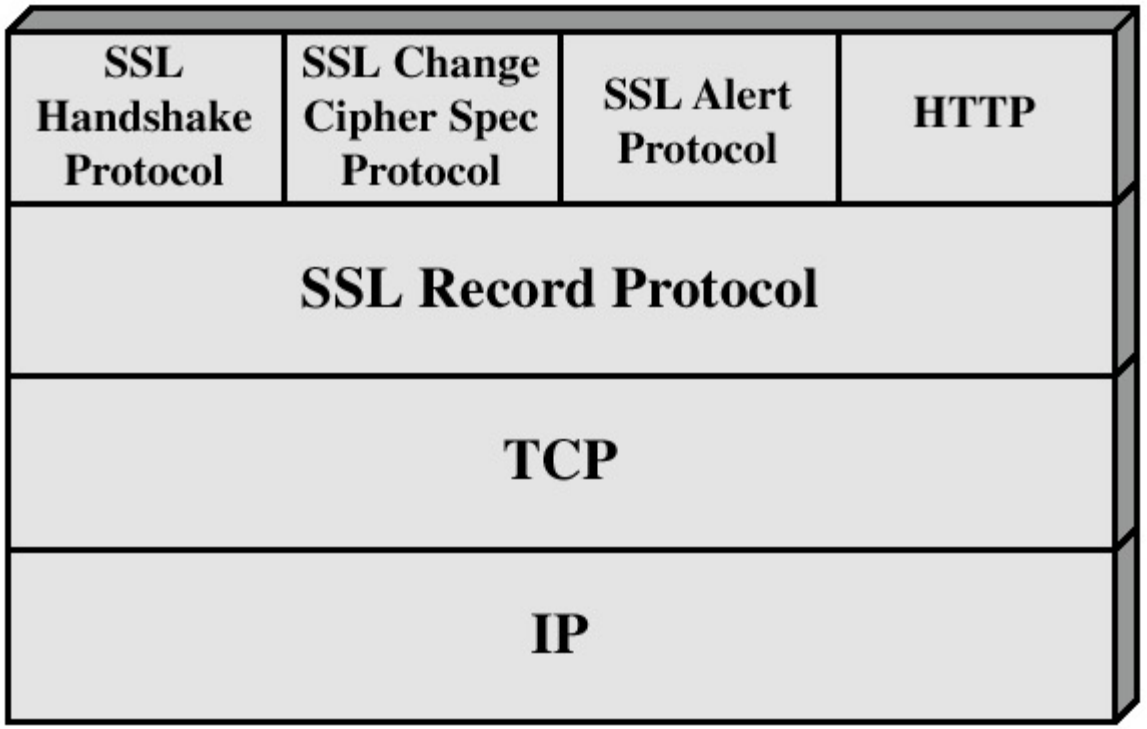
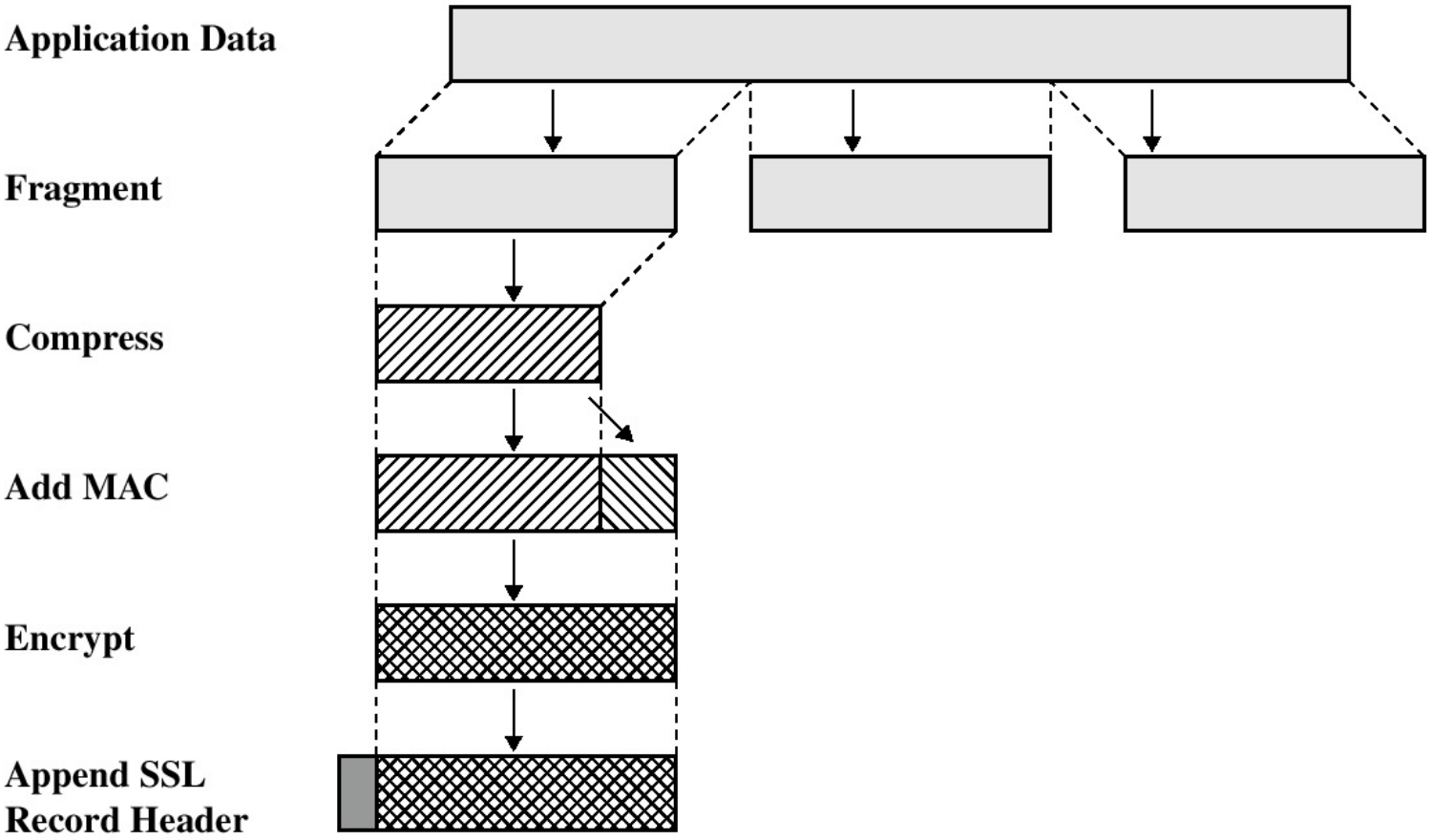
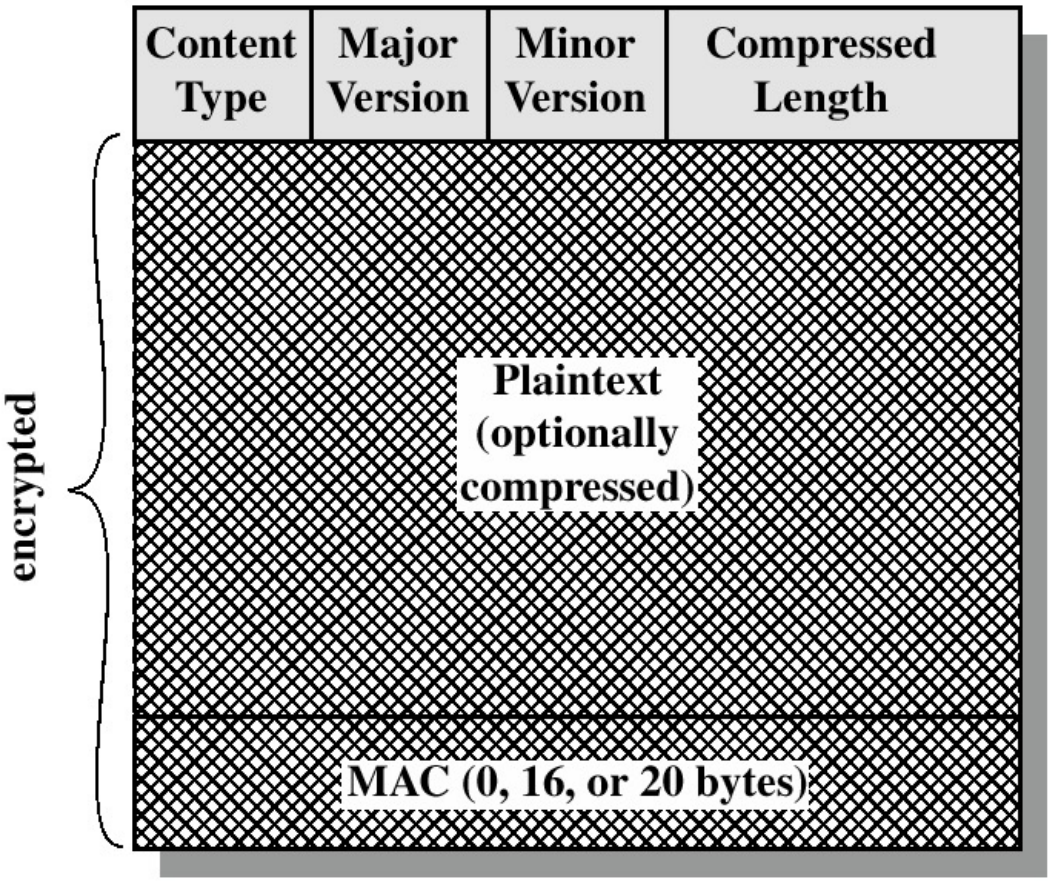


Figure 7.2 SSL Protocol Stack

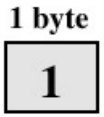
SSL Record Protocol Operation



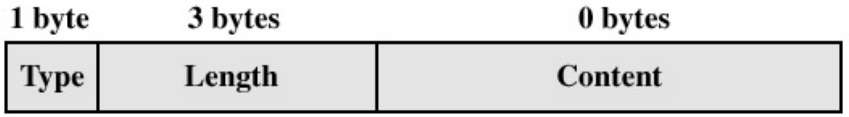
SSL Record Format



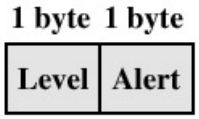
SSL Record Protocol Payload



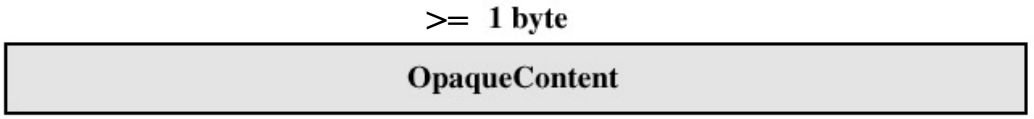
(a) Change Cipher Spec Protocol



(c) Handshake Protocol



(b) Alert Protocol

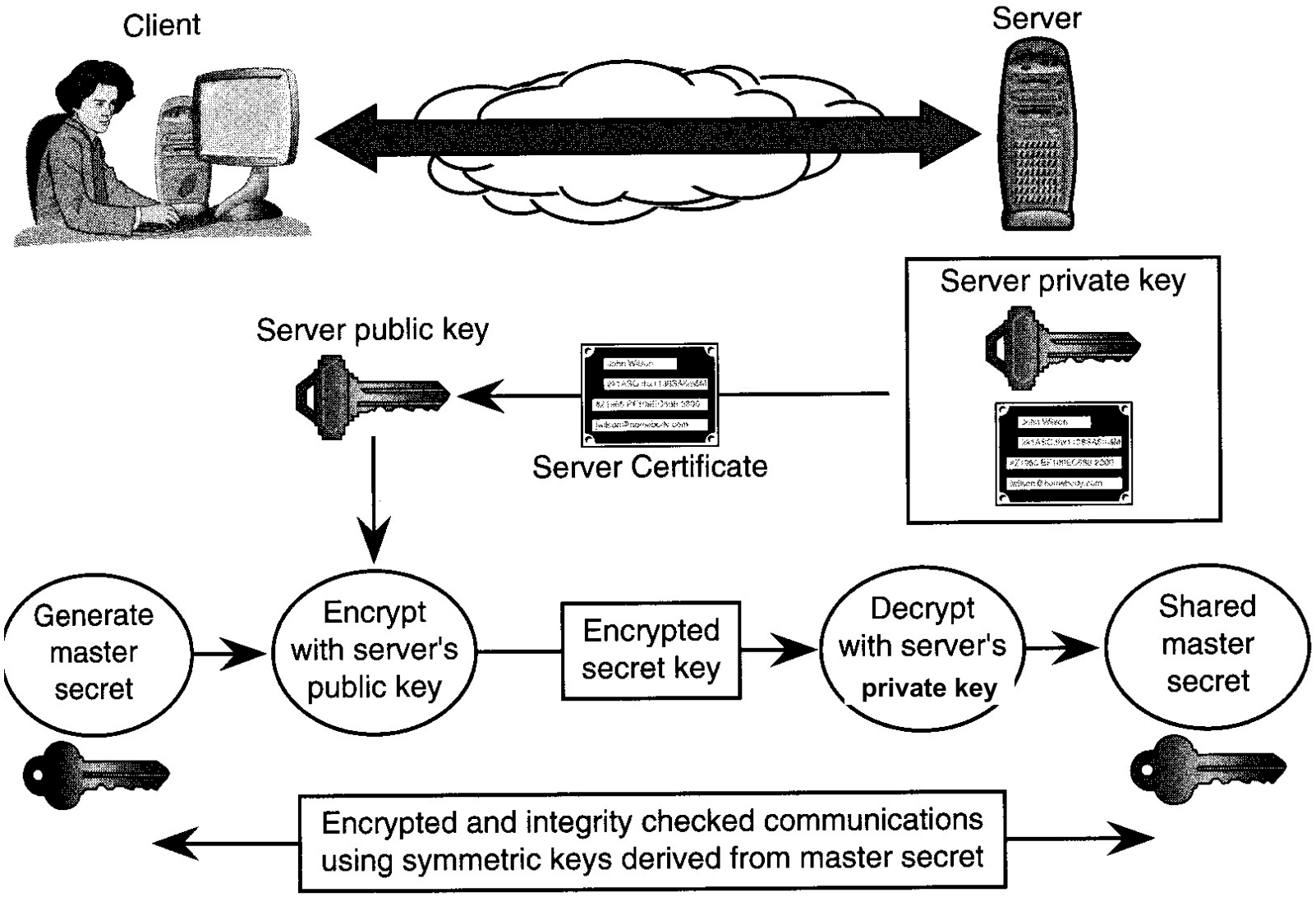


(d) Other Upper-Layer Protocol (e.g., HTTP)

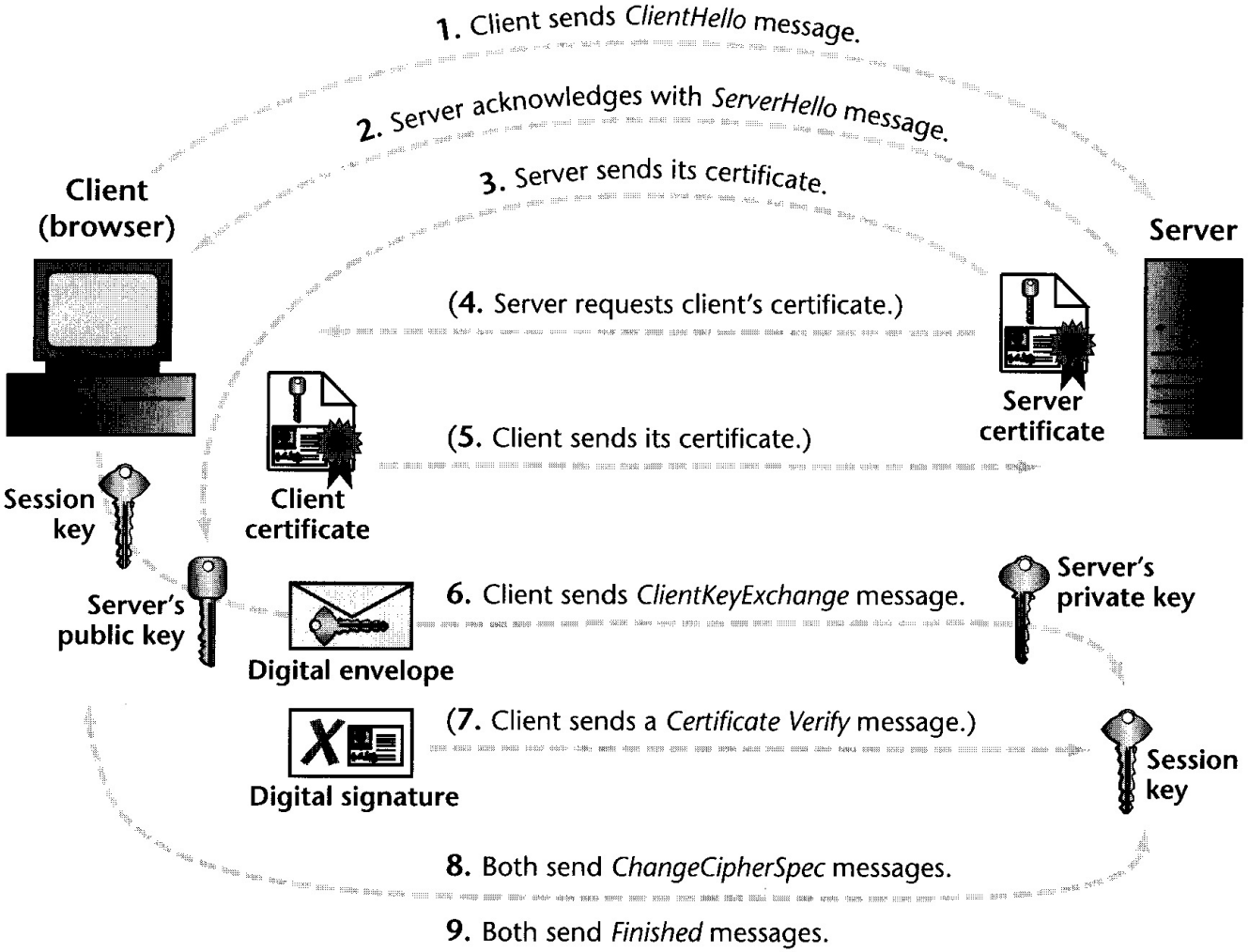
Handshake Protocol

- The most complex part of SSL.
- Allows the server and client to authenticate each other.
- Negotiate encryption, MAC algorithm and cryptographic keys.
- Used before any application data are transmitted.

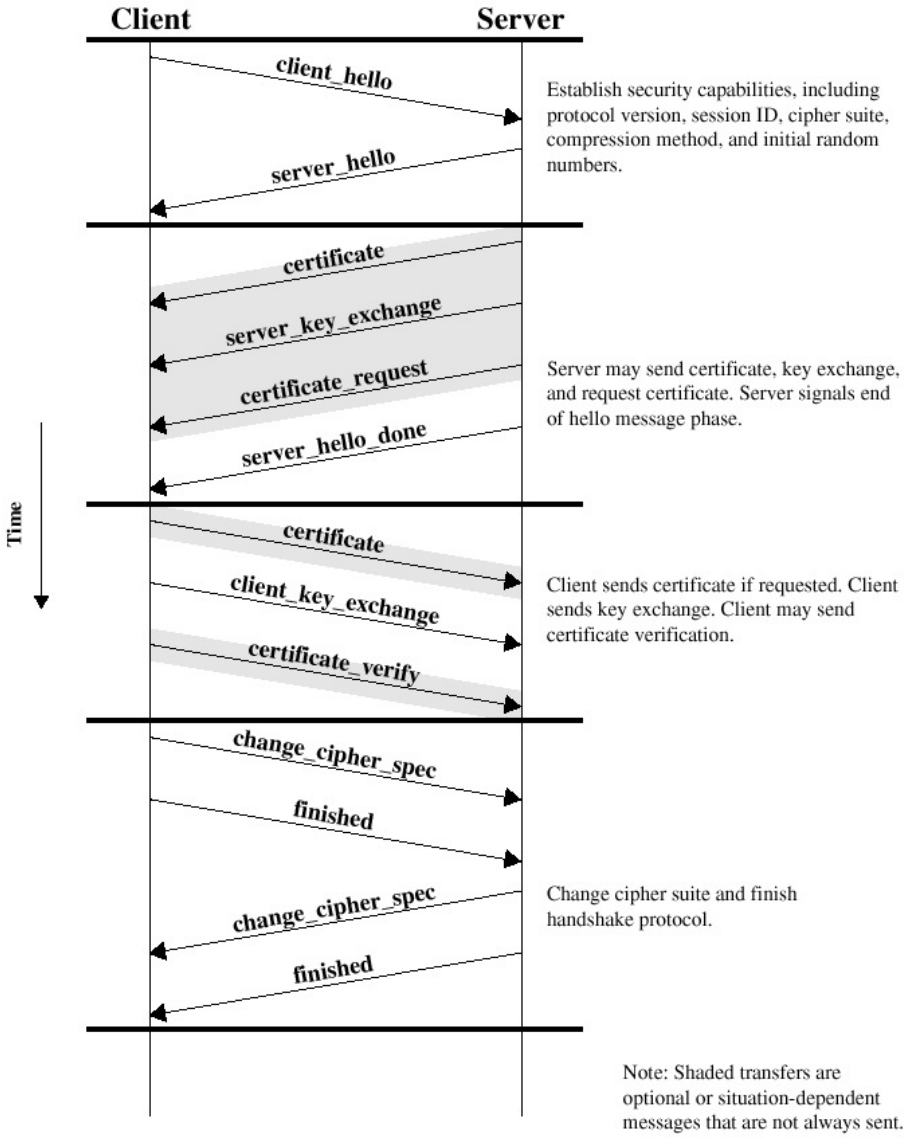
SSL Server Authentication, Encryption and Integrity Checking



A simplified SSL Handshake



Handshake Protocol Action



Transport Layer Security (TLS)

- The same record format as the SSL record format.
- Defined in RFC 2246, 4346, 5246.
- Similar to SSLv3.
- Differences in the:
  - version number
  - message authentication code
  - pseudorandom function
  - alert codes
  - cipher suites
  - client certificate types
  - certificate_verify and finished message
  - cryptographic computations
  - Padding
- RFC 4347,5238, Datagram TLS (DTLS), ``SSL running on unreliable Transport Protocols", e.g. UDP or DCCP"
- TLS v1.3, a new standardized version since v1.2 (released about a decade ago) has been finalized as of March 21, 2018 !!

Major Changes in TLS v1.3

- Support 1-RTT Handshakes
- Also support 0-RTT handshake for previously visited server
  - BUT without replay protection nor Perfect Forward Secrecy (PFS)
- Remove support for weak and lesser-used named Elliptic curves in ECC.
- Remove support for MD5 and SHA1
- Require Digital Signatures even when a previous configuration is used
- Dropping support for many insecure or obsolete features including compression, renegotiation, AES-CBC mode etc
- Prohibit SSL or RC4 negnotiation for backwards compatibility

See Eric Rescoria's talk on TLS1.3 for details:

http://web.stanford.edu/class/ee380/Abstracts/151118.html

The crypto-library developed by Mozilla and used by its Firefox browser enabled TLS 1.3 by default in Feb 2017

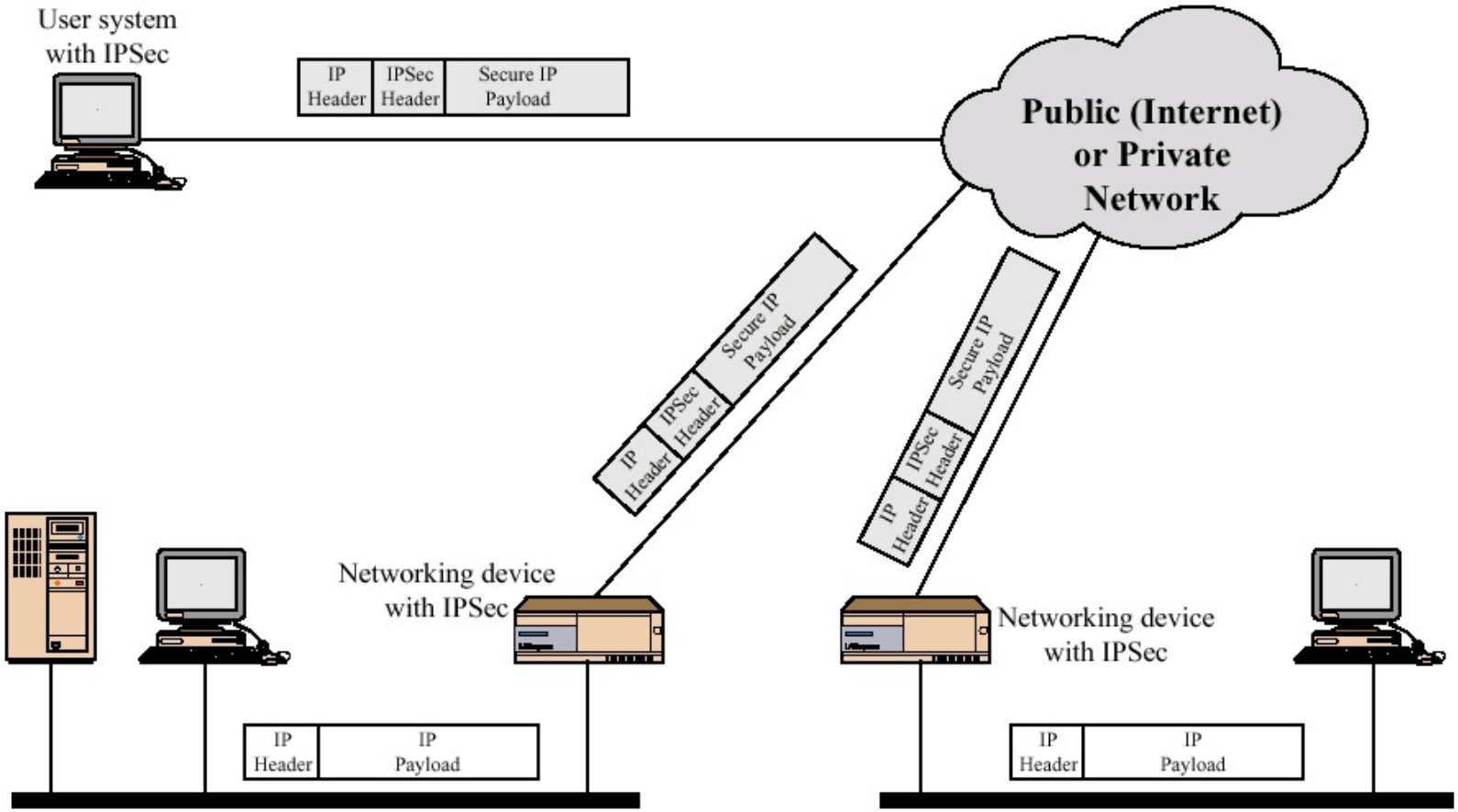
Security Solution as the Network Layer: IPsec

IP Security Overview

- RFC2401(1998), RFC4301(2005): Security in the Internet Architecture
- Identified key needs:
  - secure network infrastructure from unauthorized monitoring
  - control network traffic
  - secure end-to-end user traffic using encryption and authentication
- CERT – most serious attacks are IP spoofing and eavesdropping/packet sniffing
- Next generation IP includes authentication and encryption
- IPv6
- IPSec “supposed to be” a mandatory part of IPv6
  - Not true in real world
- Available with IPv4

Application of IPsec

- Secure branch office connectivity over the Internet
- Secure remote access over the Internet
- Establishing extranet and intranet connectivity with partners
- Enhancing electronic commerce security



Pros and Cons of IPsec

Benefits:

- Strong security for all traffic when crossing the perimeter (assuming it is implemented in a firewall or router)
- Below the transport layer (TCP, UDP) and transparent to applications
- Transparent to the end users
- Provides security for individual users – offsite workers, VPN

Drawbacks:

- Provide a host-to-host security solution rather than an end-user-to-end-user ;
- Require Operating System changes
- Not interwork with some existing/deployed networking technologies, esp. those muddle with Layer 4 and higher protocol elements, e.g.
  - Network Address Translation (NAT) boxes (some solutions do exist)
    - http://www.ietf.org/rfc/rfc3947.txt?number=3947
    - http://www.ietf.org/rfc/rfc3948.txt?number=3948
  - Load-balancers

IPSec Services

- Provides security services at the IP layer
- Enables a system to:
  - select required security protocols
  - determine algorithms to use
  - setup needed keys
- Two Protocols:
  - Authentication protocol – designated by the authentication header (AH)
  - Encryption/Authentication protocol – designated by the format of the packet, Encapsulating Security Payload (ESP); it is a mechanism for providing integrity and confidentiality to IP datagrams

IPsec Services

	AH	ESP (encryption only)	ESP (encryption plus authentication)
Access control	✔	✔	✔
Connectionless integrity	✔		✔
Data origin authentication	✔		✔
Rejection of replayed packets	✔	✔	✔
Confidentiality		✔	✔
Limited traffic flow confidentiality		✔	✔

Encapsulating Security Payload

- Provides confidentiality services
- Confidentiality of message contents and limited traffic flow confidentiality
- ESP can also provide the same authentication services as AH

Authentication Header

- Provide support for data integrity and authentication of IP packets
- Undetected modification in transit is impossible
- Prevents address spoofing attacks
- Guards against replay attacks
- Based on the use of a message authentication code (MAC) so two parties must share a key

ESP and AH Algorithms

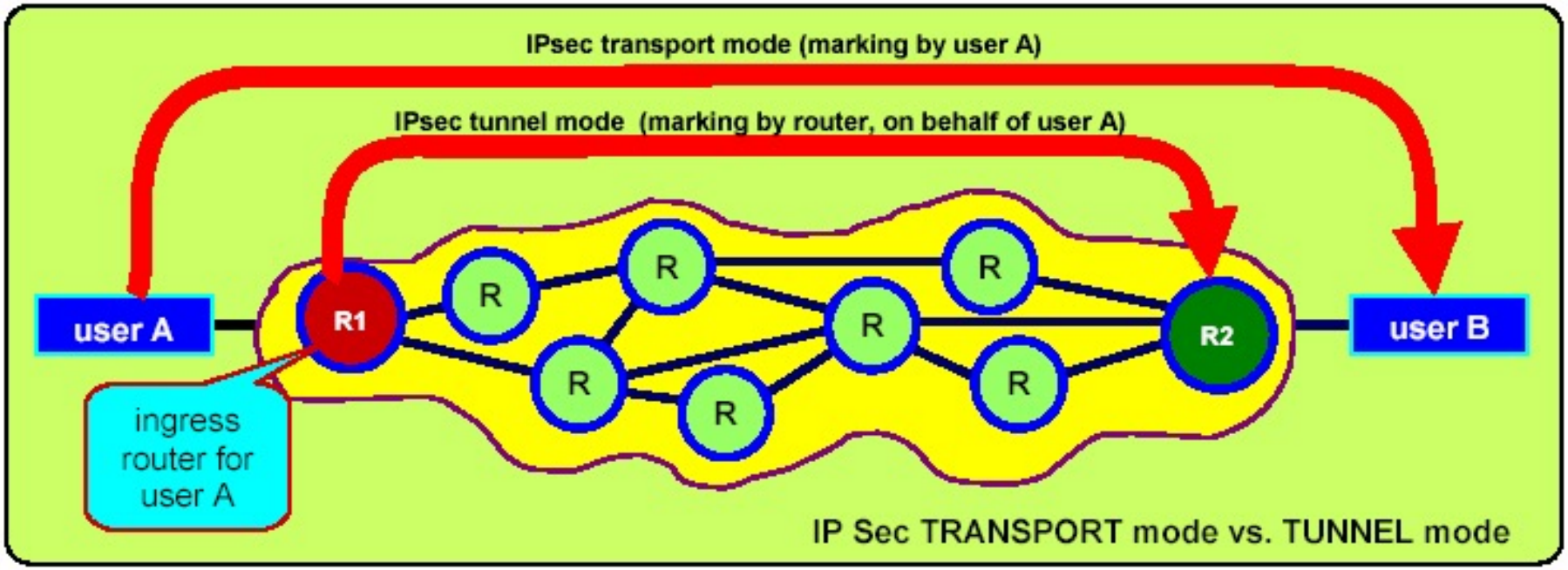
- Implementation must support DES in cipher block chaining (CBC) mode
- Other algorithms have been assigned identifiers in the standards document ;
- Others:
  3DES, PC5, IDA, 3IDEA, CAST, Blowfish
- ESP and AH both support use of a 96bit MAC
  - HMAC value is calculated but only first 96 bits are used
    - HMAC-MD5-96
    - HMAC-SHA-1-96

Transport Vs. Tunnel Modes

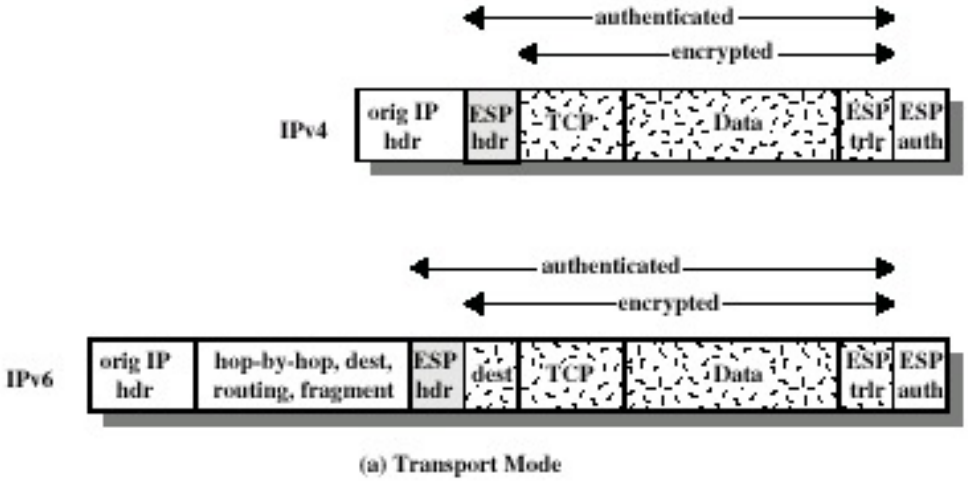
- SA supports two modes:

  Transport – protection for the upper layer protocols, with partial IP header protection.

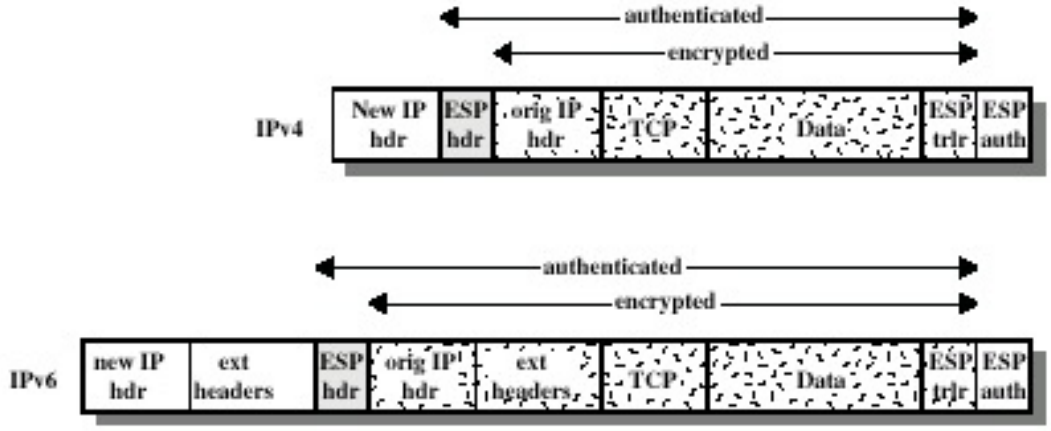
  Tunnel – protection for the entire IP packet



Scope of ESP Encryption



(a) Transport Mode



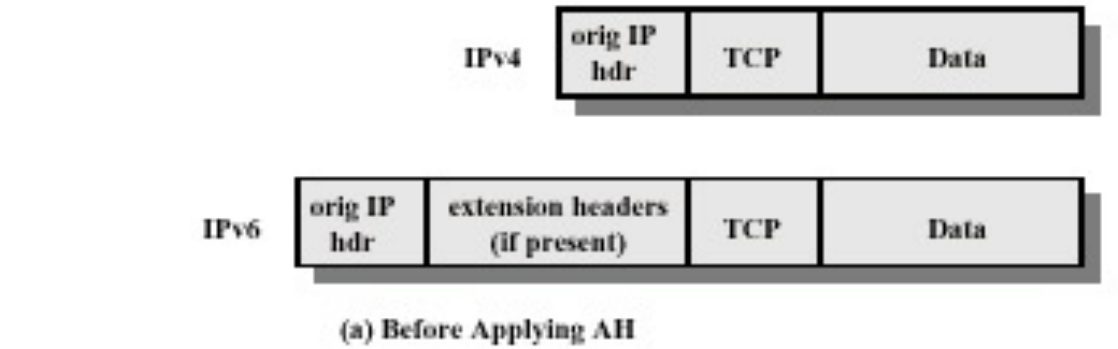
(b) Tunnel Mode

Transport Mode Vs. Tunnel Mode

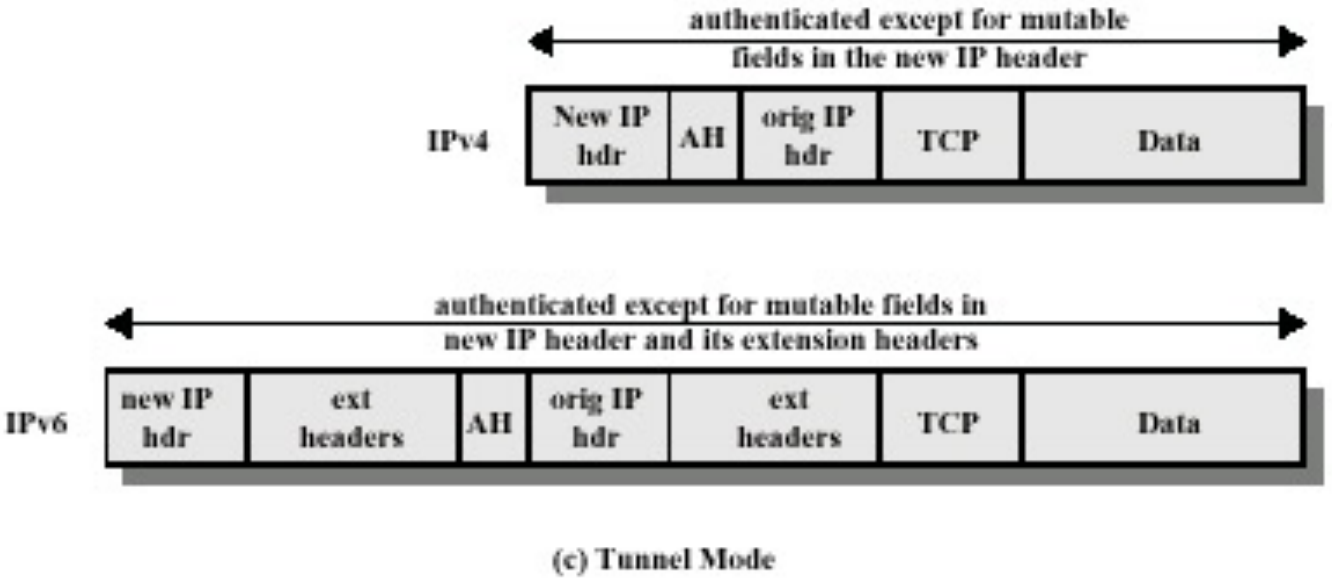
- Protection extends to the payload of an IP packet
- Primarily for upper layer protocols – TCP, UDP, ICMP
- Mostly used for end-to-end communication
- Encrypts and/or authenticates the payload, but not all of the IP header

- Protection for the entire packet
- Add new *outer* IP packet with a new outer header
- AH or ESP fields are added to the IP packet and entire packet is treated as payload of the outer packet
- Packet travels through a *tunnel* from point to point in the network

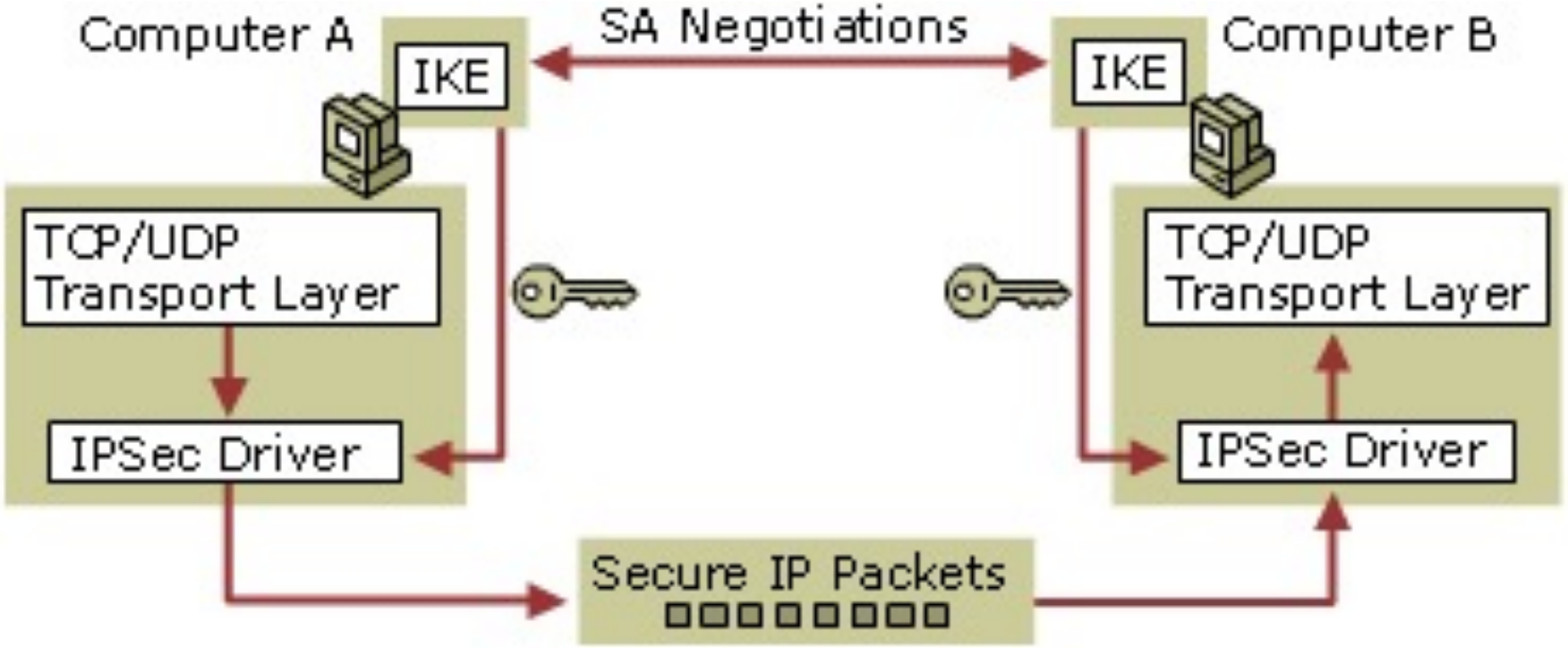
Scope of AH Authentication



Scope of AH Authentication

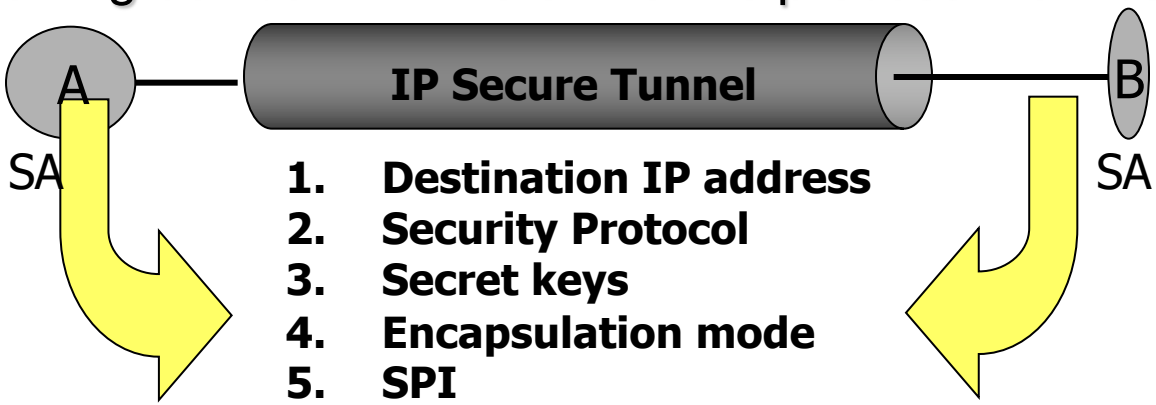


(c) Tunnel Mode


Computer A
IKE
SA Negotiations
IKE
Computer B
TCP/UDP
Transport Layer
TCP/UDP
Transport Layer
IPSec Driver
IPSec Driver
Secure IP Packets

Security Associations

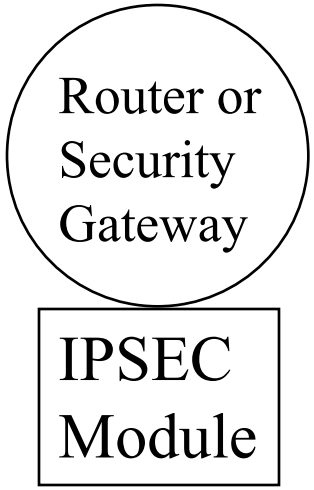
- Security Association (SA) – is a one-way relationship between a sender and a receiver (end-hosts or routers) that defines the security services that are provided to a user
- System administrator or network/security designer decides where to set them up.
- A SA is uniquely identified by:
  - Destination IP address – address of the destination endpoint of the SA (end user system or firewall/router)
  - Security protocol – whether association is AH or ESP. Defines key size, lifetime and crypto algorithms (transforms)
  - Security parameter index (SPI) – bit string that provides the receiving device with info on how to process the incoming traffic



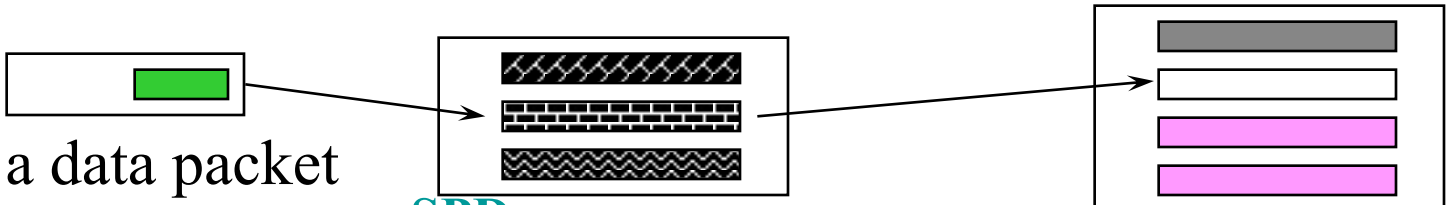
Security Associations

- SA is unidirectional
- It defines the operations that occur in the transmission in one direction only
- Bi-directional transport of traffic requires a pair of SAs (e.g., secure tunnel)
  - These two SAs use the same characteristics but employ different keys
- Requirements are stored in two databases: security policy database (SPD) and security association database (SAD)

SPD and SAD (Outgoing Packets)



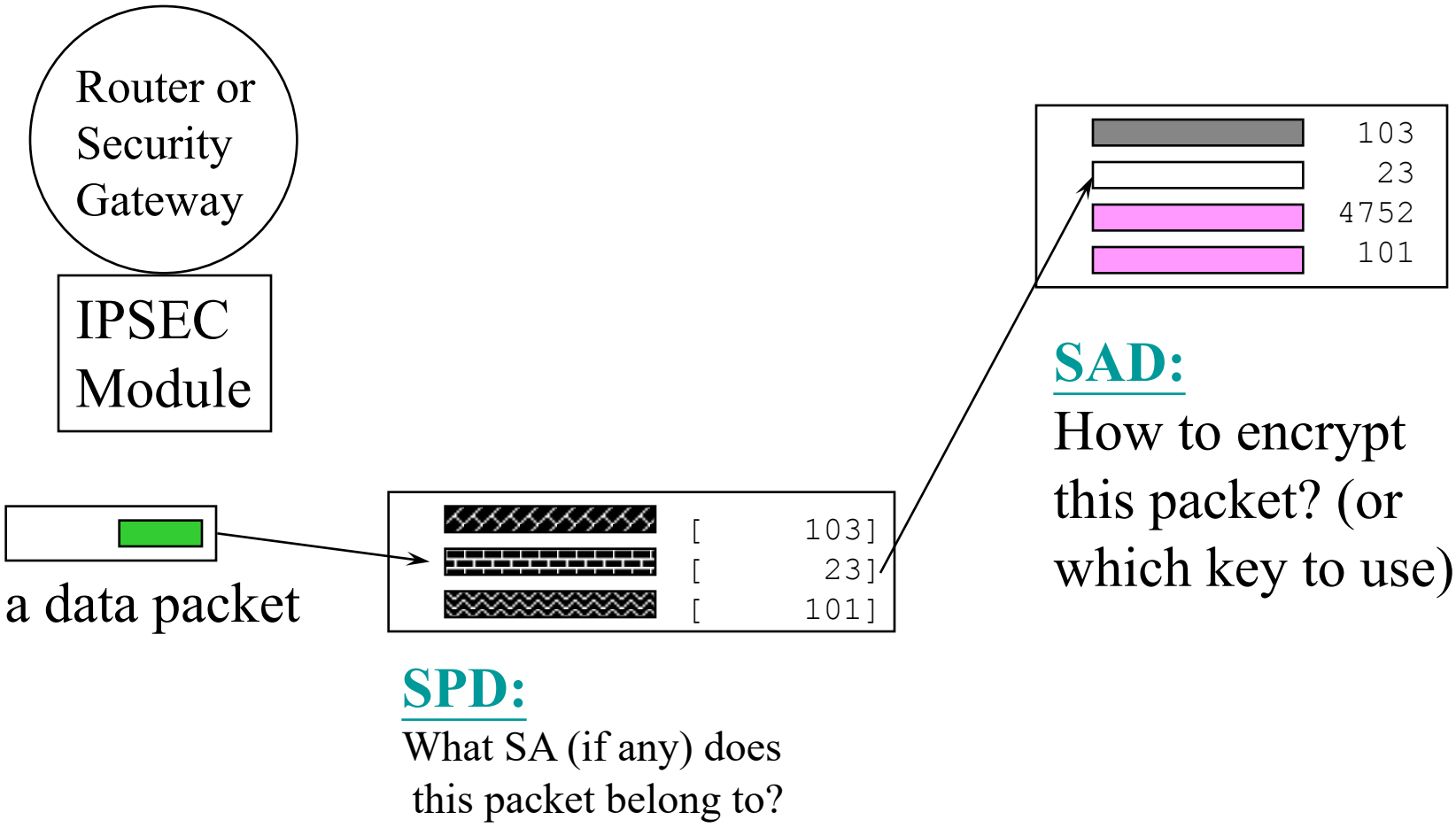
Security Policy Database (SPD):
which packet to do what?
Security Association Database (SAD):
how to do it?



SPD:
Which SA (if any) does this packet belong to ? If none, should a new SA be setup to protect this packet ?

SAD:
How to encrypt this packet? (e.g. which key to use, Use DES ? AES ?

SPI (Security Parameter Index)

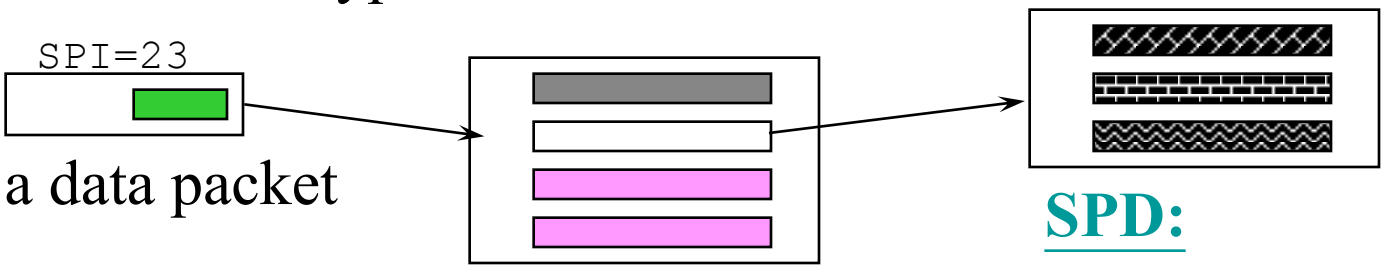


SPD and SAD (Incoming Packets)

Router or Security Gateway

IPSEC Module

I can not see the selector information yet because of the encryption!!



a data packet

SAD:
Which SA to use to decrypt the packet

SPD:
Is this violating my local policy?

Security Policy Database

- Considerable flexibility in way IPSec services are applied to IP traffic
- Can discriminate between traffic that is afforded IPSec protection and traffic allowed to bypass IPSec
- The Security Policy Database (SPD) is the means by which IP traffic is related to specific SAs
- Each entry defines a subset of IP traffic and points to an SA for that traffic
- These selectors are used to filter outgoing traffic in order to map it into a particular SA

Security Policy Database

- Destination IP address
- Source IP address
- User ID
- Data sensitivity level – secret or unclassified
- Transport layer protocol
- IPSec protocol – AH or ESP or AH/ESP
- Source and destination ports
- IPv6 class
- IPv6 flow label
- IPv4 type of service (TOS)

Security Policy Database

Outbound processing for each packet:

1. Compare fields in the packet header to find a matching SPD entry
2. Determine the SA and its associated SPI
   - If the required SA has not been setup, establish one
3. Do the required IPSec processing as specified in the SAD

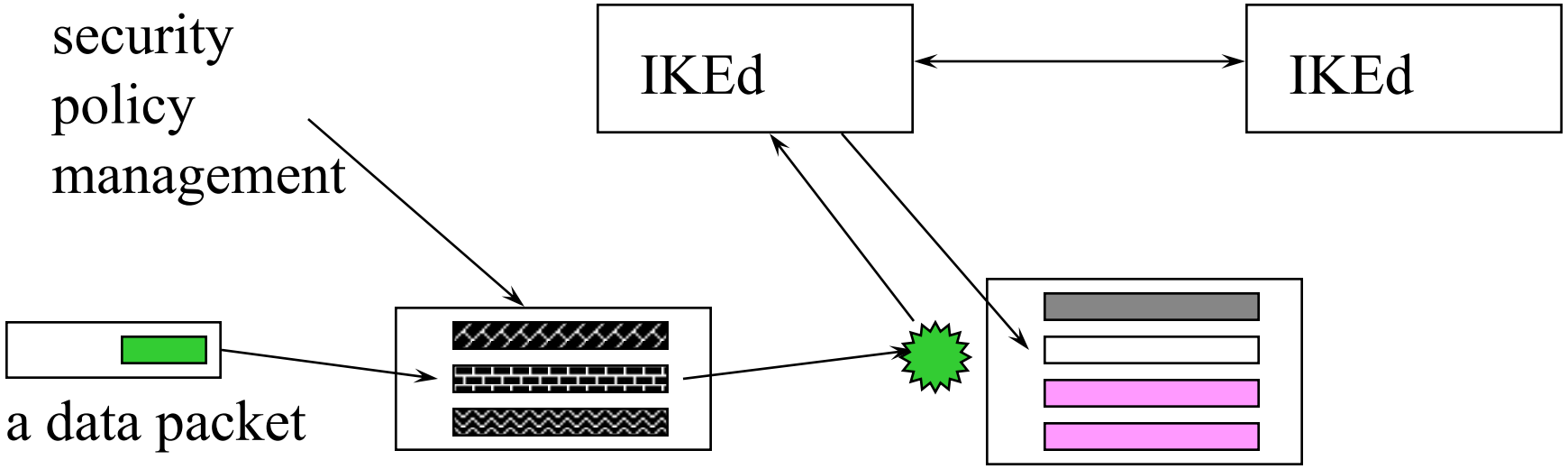
Security Association Database

- Each IPSec implementation has a Security Association Database (SAD)
- SAD defines the parameters association with each SA, indexed by SPI
- SAD stores pairs of SA, since SAs are unidirectional

Security Association Database

- Sequence number counter (for replay window protection)
- Anti-replay window
- Sequence counter overflow
- AH information
- ESP information
- Lifetime of this SA
- IPSec protocol mode – tunnel, transport
- Path MTU

Traffic Driven SA Establishment



SPD:
Should this packet be authenticated….

SAD:
How to authenticate this packet…...

Key Management

- AH and ESP require encryption and authentication keys
- Key Management is needed for determination and distribution of secret keys
  - a Process to negotiate and establish IPSec SA's between two entities
  - Support both Manual and Automatic modes of key management
  - Manual Keying
    - Mandatory
    - Useful when IPSec developers are debugging
    - Keys exchanged offline (phone, email, etc.)
    - Set up SPI and negotiate parameters
  - Automatic Keying
    - Traffic Driven ; dynamically setup a SA when an outbound packet requires IPsec protection according to the Security Policy Database (SPD) specification

Authomatic Key Management via Internet Key Exchange - IKE

- The protocol for automatic key management for IPsec:
  - Internet Security Association and Key Management Protocol (ISAKMP) ; also referred as IKE – Internet Key Exchange
- Define procedures and packet formats to establish, negotiate, modify and delete SAs
- Define payloads for exchanging key generation and authentication data

- Used when an outbound packet does not have an SA
- Two phases:
  - Phase 1: Establish an IKE SA
  - Phase 2: Use IKE SA to negotiate IPSec SAs
- IKE SA used to define encryption & authentication of IKE traffic
- Multiple IPSec SAs can be established with one IKE SA
- IKE SA bidirectional

IKE protocol (cont'd)

- IKEv1 supports both pre-shared secret and Diffie-Hellman key-exchange
- Also supports the use of public-key/ digital certificate for peer identity authentication
- IKEv1 is notorious for
  - its many different modes of operations (e.g. main vs. aggressive, different ways of using public-keys)

  => Severe implementation complexity
- The Standardization of IKEv2 has recently completed by IETF ;
  - IKEv2 simplifies IKEv1 by getting rid of many not-that-useful modes while adding some practically useful functions, e.g.
    - support dynamic address assignment ;
    - Provide transport of EAP messages to better authentication using an AAA-infrastructure (RADIUS/ DIAMETER)
  - IKEv2 is NOT backward compatible with IKEv1

Summary IPSec

- Key exchange and encryption are separate
  - New encryption algorithms can be added
- Complex – a lot of flexibility & options