ELECTRONIC MAIL SECURITY

INTRODUCTION

- In all virtually distributed environments, **electronic mail** is the most heavily used network-based application.
- Users expect to be able to, and do, send e-mail to others who are connected directly or indirectly to the Internet.
- With the explosively growing reliance on e-mail, there grows a demand for authentication and confidentiality services, which is provide by:
  - Pretty Good Privacy (PGP)
  - S/MIME

PRETTY GOOD PRIVACY (PGP)

- PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.
- Developed by Phil Zimmermann
  - Selected the best available cryptographic algorithms as building blocks
  - Integrated these algorithms into a general-purpose application that is independent of operating system and processor and that is based on a small set of easy-to-use commands
  - Made the package and its documentation, including the source code, freely available via the Internet, bulletin boards, and commercial networks
  - Entered into an agreement with a company to provide a fully compatible, low-cost commercial version of PGP.

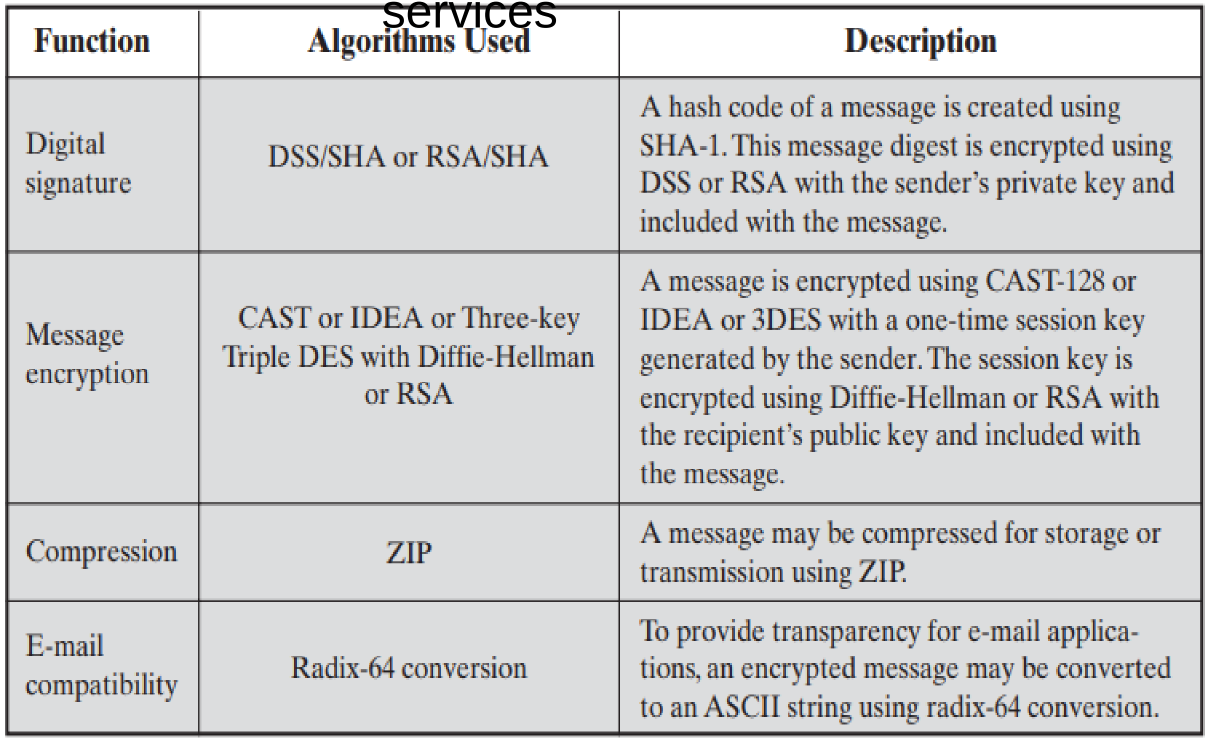
PGP Growth

- It is available free worldwide in versions that run on a variety of platforms, including Windows, UNIX, Macintosh, and many more.
- The commercial version satisfies users who want a product that comes with vendor support.
- It is based on algorithms that have survived extensive public review and are considered extremely secure.
- It has a wide range of applicability.
- It was not developed by nor is it controlled by any governmental or standards organization.
- PGP is now on an Internet standards track (RFC 3156; MIME Security with OpenPGP).

Operational Description of PGP

- The actual operation of PGP consists of four services:
  - authentication,
  - confidentiality,
  - compression, and
  - e-mail compatibility

Summary of PGP services



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.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

PGP Authentication

- Combination of SHA-1 and RSA provides an effective digital signature scheme
  - Because of the strength of RSA the recipient is assured that only the possessor of the matching private key can generate the signature
  - Because of the strength of SHA-1 the recipient is assured that no one else could generate a new message that matches the hash code
- As an alternative, signatures can be generated using DSS/SHA-1
- Detached signatures are supported
  - Each person's signature is independent and therefore applied only to the document.

PGP Confidentiality

- Provided by encrypting messages to be transmitted or to be stored locally as files
    - In both cases the symmetric encryption algorithm CAST-128 may be used
    - Alternatively IDEA or 3DES may be used
    - The 64-bit cipher feedback (CFB) mode is used
- In PGP each symmetric key is used only once
    - Although referred to as a session key, it is in reality a one-time key
    - Session key is bound to the message and transmitted with it
    - To protect the key, it is encrypted with the receiver’s public key
- As an alternative to the use of RSA for key encryption, PGP uses ElGamal, a variant of Diffie-Hellman that provides encryption/decryption.

PGP Confidentiality and Authentication

- Both services may be used for the same message
  - First a signature is generated for the plaintext message and prepended to the message
  - Then the plaintext message plus signature is encrypted using CAST-128 (or IDEA or 3DES) and the session key is encrypted using RSA (or ElGamal)
- When both services are used:
  - The sender first signs the message with its own private key
  - Then encrypts the message with a session key
  - And finally encrypts the session key with the recipient’s public key

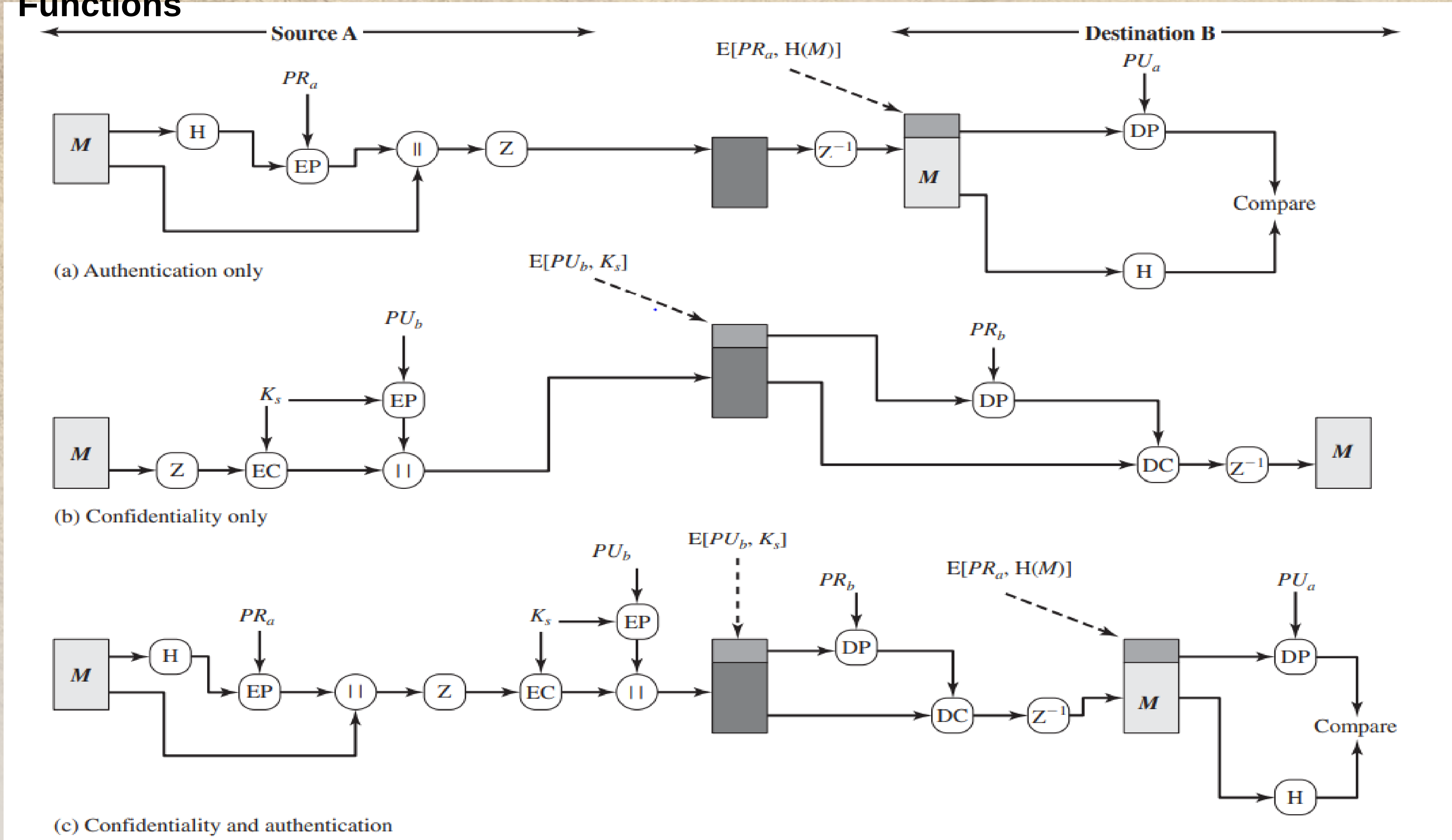
PGP Compression

- As a default, PGP compresses the message after applying the signature but before encryption
  - This has the benefit of saving space both for e-mail transmission and for file storage
  - The placement of the compression algorithm is critical
    - Applying the hash function and signature after compression would constrain all PGP implementations to the same version of the compression algorithm
    - Message encryption is applied after compression to strengthen cryptographic security
  - The compression algorithm used is ZIP

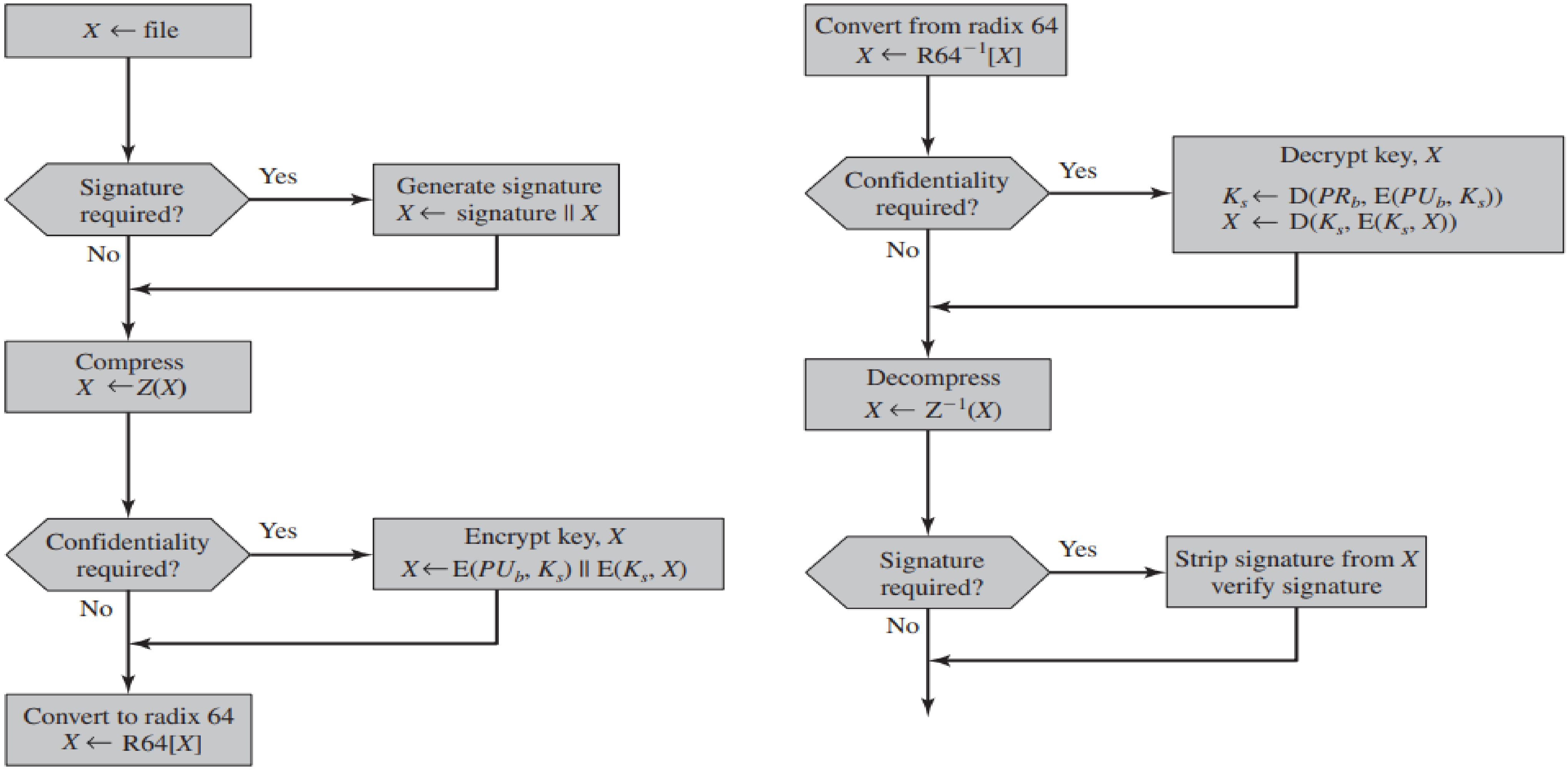
PGP E-mail Compatibility

- Many electronic mail systems only permit the use of blocks consisting of ASCII text
    - To accommodate this restriction, PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters
    - The scheme used for this purpose is radix-64 conversion
        - Each group of three octets of binary data is mapped into four ASCII characters
        - This format also appends a CRC to detect transmission errors

PGP Cryptographic Functions



Transmission and Reception of PGP Messages



$X \leftarrow$ file
Signature required?
Yes
No
Generate signature $X \leftarrow$ signature $\|$ X
Compress $X \leftarrow Z(X)$
Confidentiality required?
Yes
No
Encrypt key, X
$X \leftarrow E(PU_b, K_s) \| E(K_s, X)$
Convert to radix 64 $X \leftarrow R64[X]$
Convert from radix 64 $X \leftarrow R64^{-1}[X]$
Confidentiality required?
Yes
No
Decrypt key, X
$K_s \leftarrow D(PR_b, E(PU_b, K_s))$
$X \leftarrow D(K_s, E(K_s, X))$
Decompress $X \leftarrow Z^{-1}(X)$
Signature required?
Yes
No
Strip signature from X verify signature

(a) Generic transmission diagram (from A)

(b) Generic reception diagram (to B)

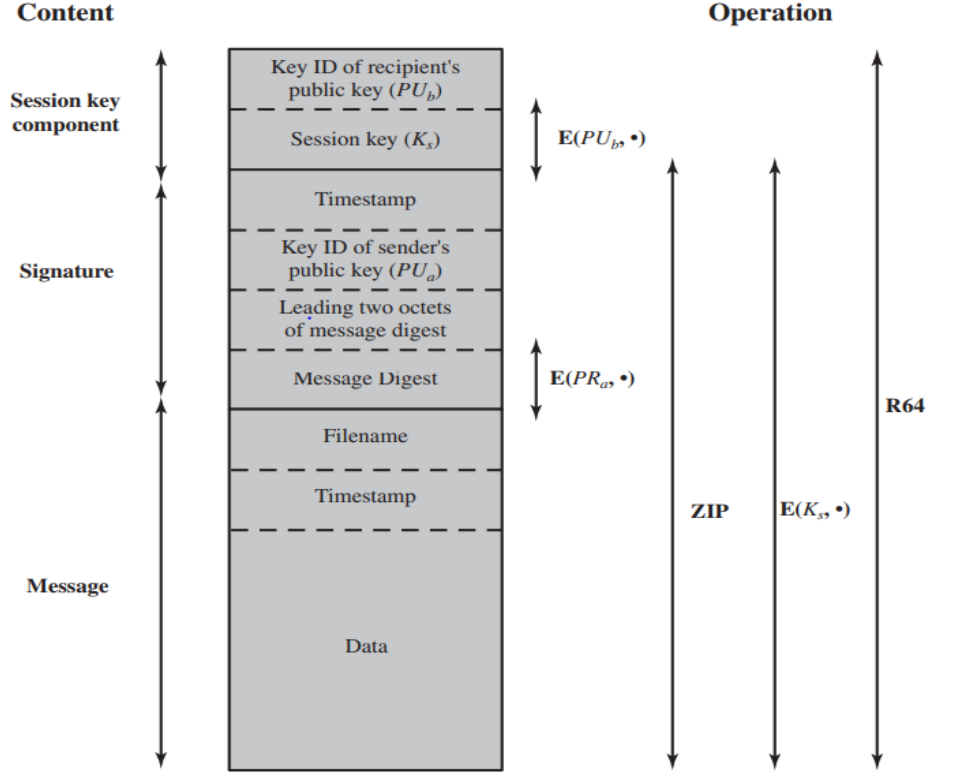
PGP Session Keys

- Need a session key for each message
  - of varying sizes: 56-bit DES, 128-bit CAST or IDEA, 168-bit Triple-DES
- Uses random inputs taken from
  - actual keys hit
  - keystroke timing of a user

PGP Public & Private Keys

- Since many public/private keys may be in use, need to identify which is actually used to encrypt session key in a message
  - could send full public-key with every message
  - but this is inefficient
- rather use a key identifier based on key
  - is least significant 64-bits of the key
  - will very likely be unique
- also use key ID in signatures

General Format PGP Message (From A To B)



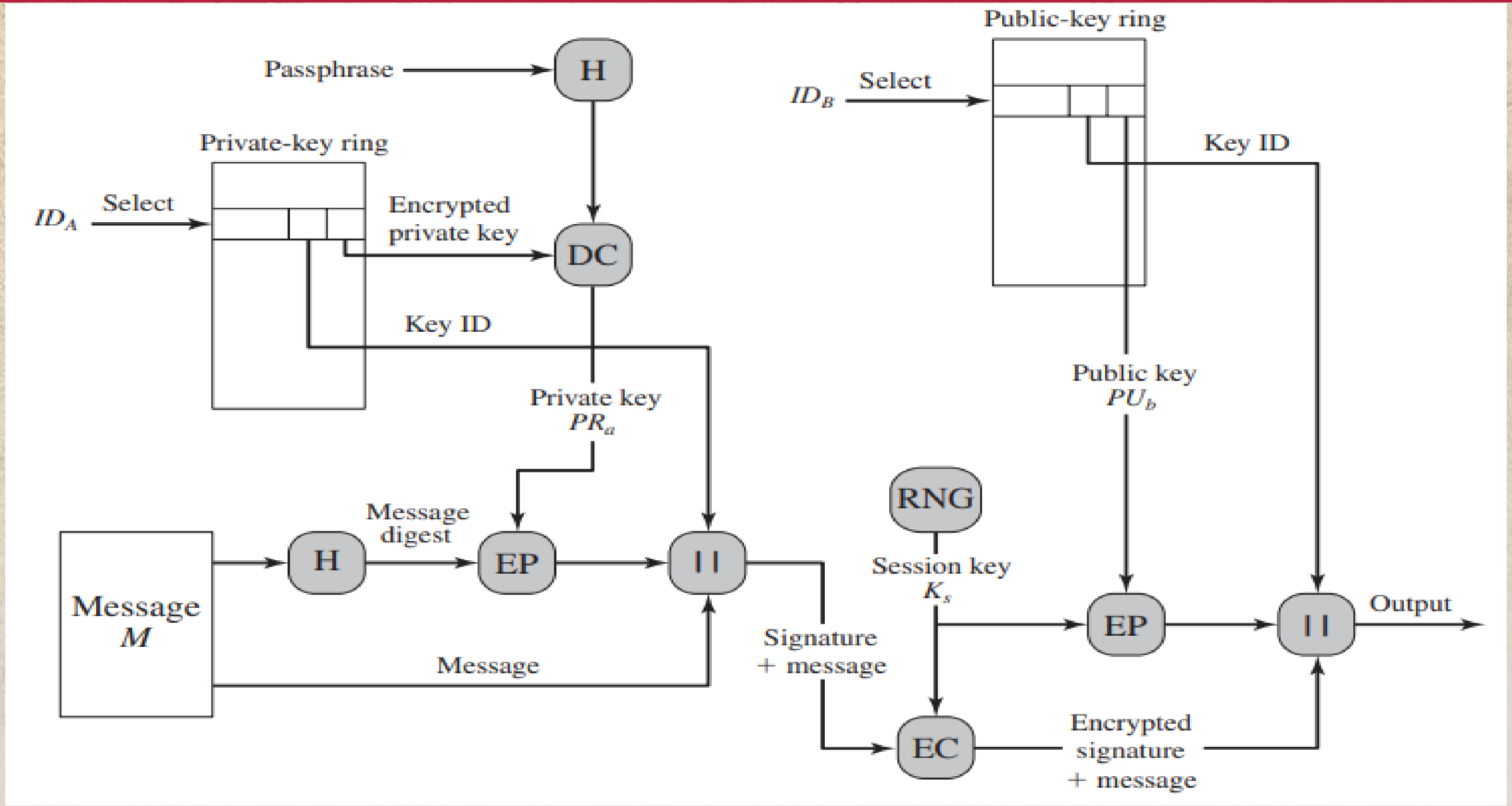
Notation:

E(PU_b, •)	=	encryption with user b's public key
E(PR_a, •)	=	encryption with user a's private key
E(K_s, •)	=	encryption with session key
ZIP	=	Zip compression function
R64	=	Radix-64 conversion function

PGP Key Rings

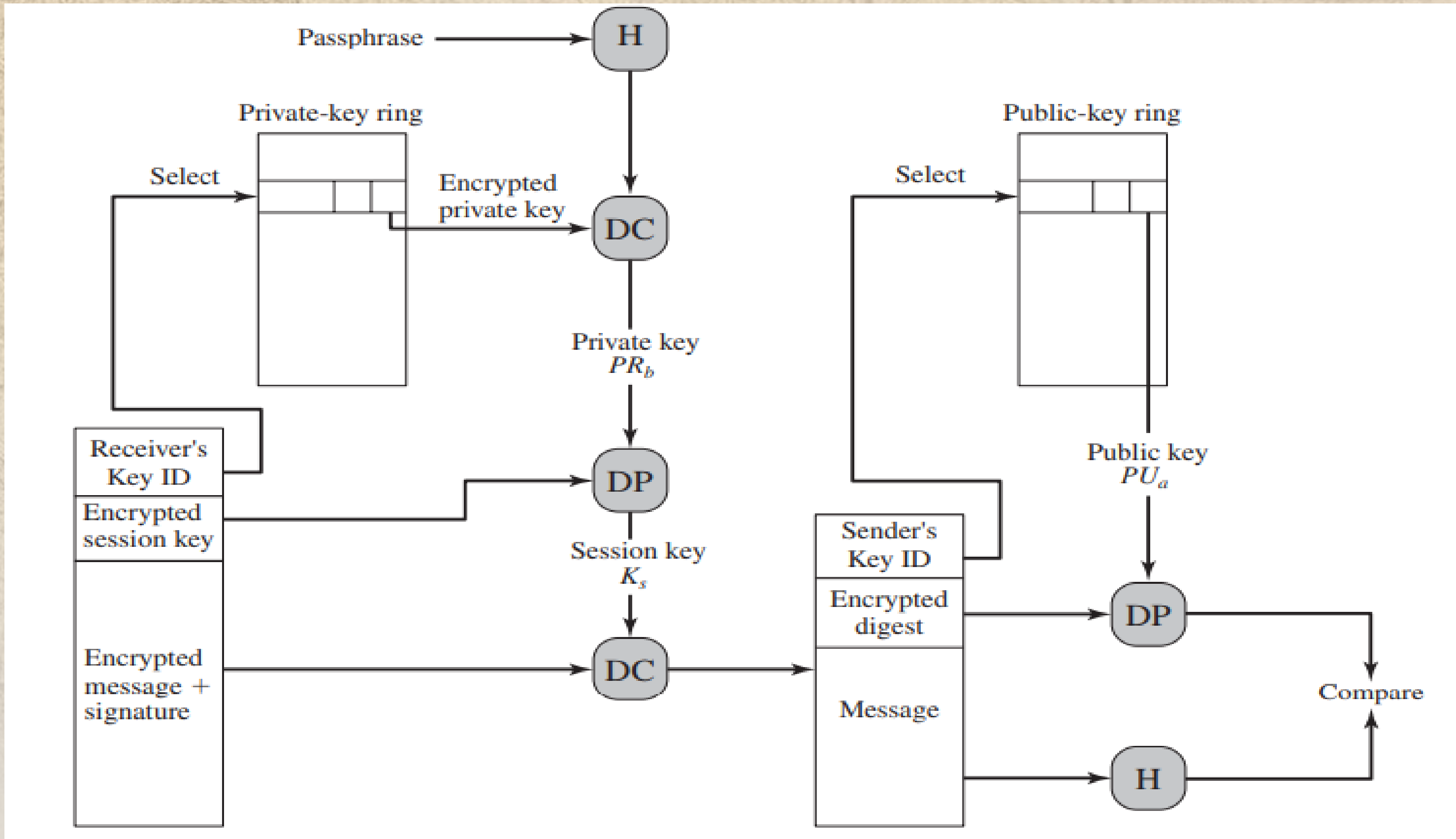
- Each PGP user has a pair of keyrings:
  - public-key ring contains all the public-keys of other PGP users known to this user, indexed by key ID
  - private-key ring contains the public/private key pair(s) for this user, indexed by key ID & encrypted keyed from a hashed passphrase
- Security of private keys thus depends on the pass-phrase security

PGP Message Generation



PGP Message Generation (from User A to User B: no compression or radix-64 conversion)

PGP Message Reception



PGP Message Reception (from User A to User B; no compression or radix-64 conversion)

SECURE/MULTIPURPOSE INTERNET MAIL EXTENSION (S/MIME)

- A security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security
- Defined in:
  - RFCs 3370, 3850, 3851, 3852

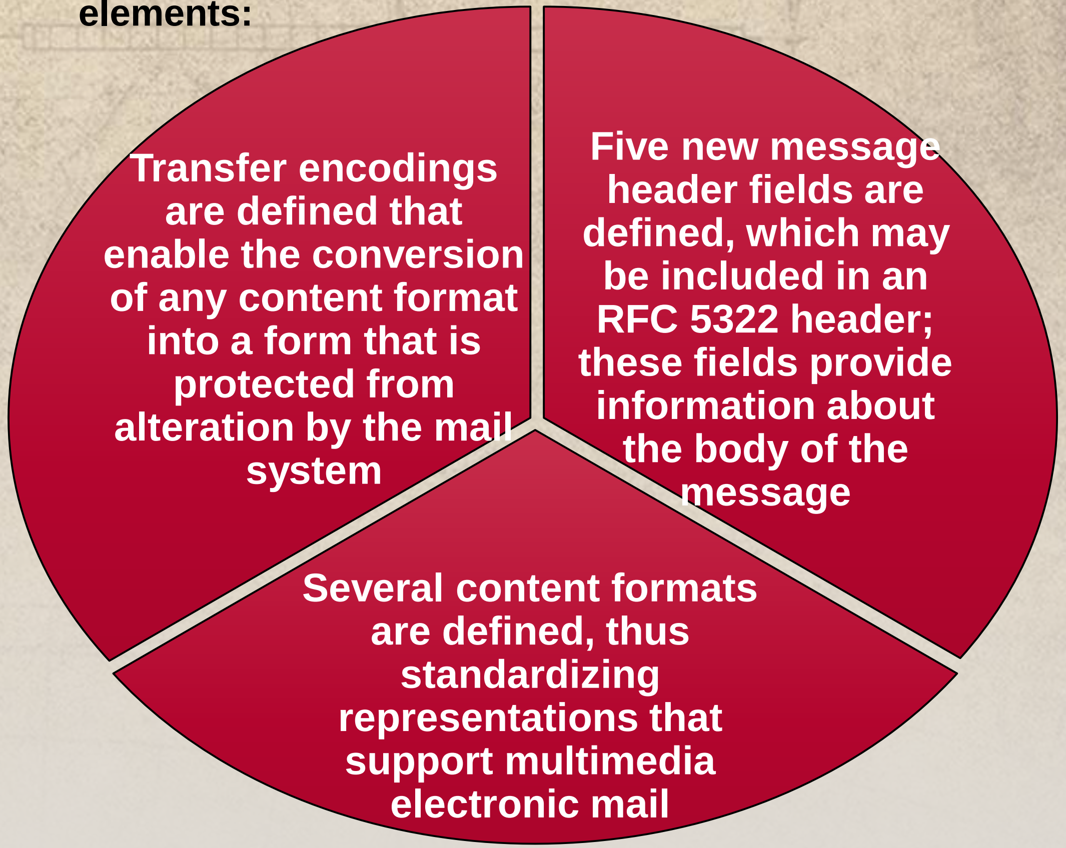
RFC 5322

- Defines a format for text messages that are sent using electronic mail
- Messages are viewed as having an envelope and contents
  - The envelope contains whatever information is needed to accomplish transmission and delivery
  - The contents compose the object to be delivered to the recipient
  - RFC 5322 standard applies only to the contents
- The content standard includes a set of header fields that may be used by the mail system to create the envelope

Multipurpose Internet Mail Extensions (MIME)

- An extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP)
  - Is intended to resolve these problems in a manner that is compatible with existing RFC 5322 implementations
  - The specification is provided in RFCs 2045 through 2049

MIME specification includes the following elements:



Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system

Five new message header fields are defined, which may be included in an RFC 5322 header; these fields provide information about the body of the message

Several content formats are defined, thus standardizing representations that support multimedia electronic mail

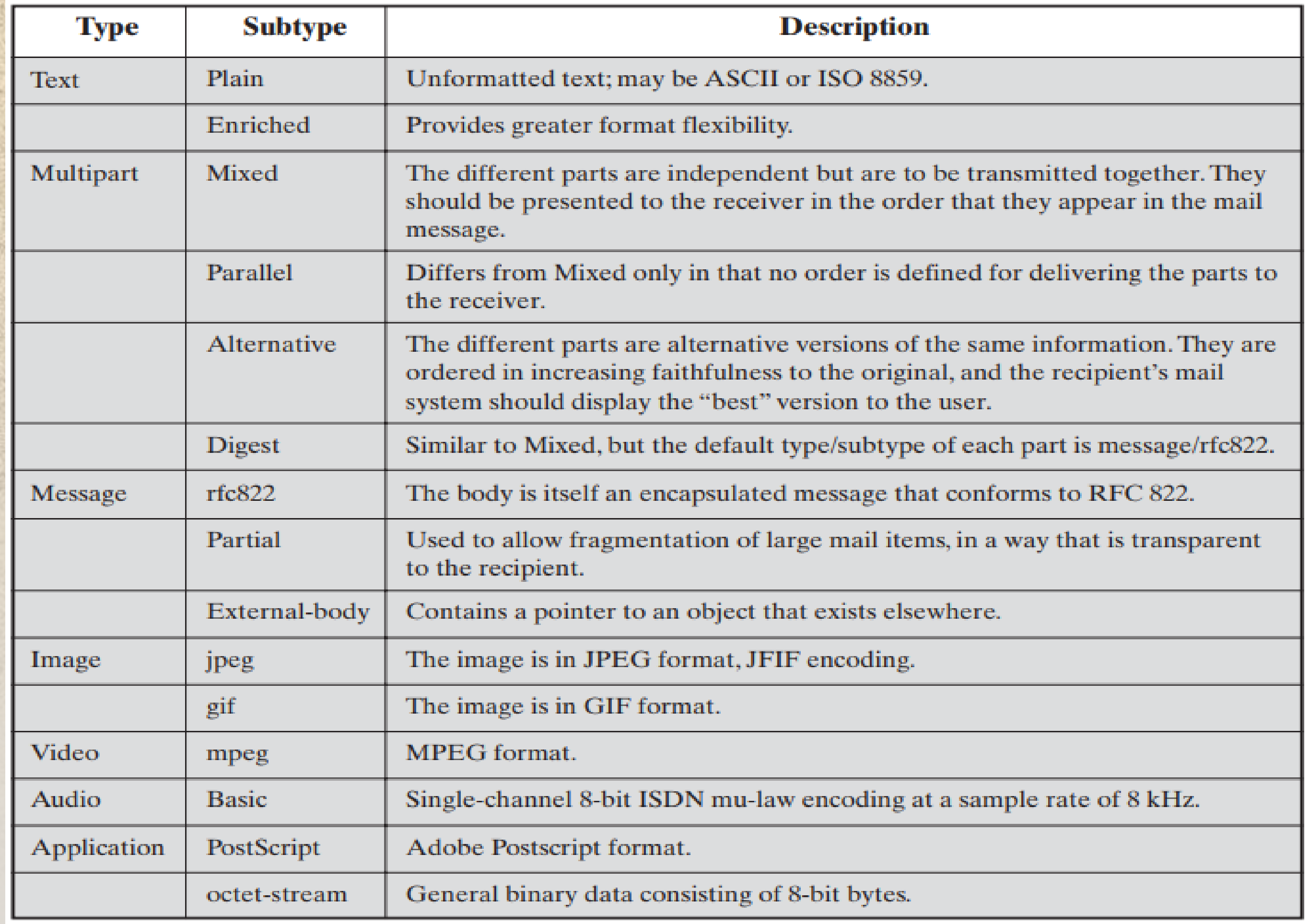
The Five Header Fields Defined in MIME

- MIME-Version
  - Must have the parameter value 1.0
  - This field indicates that the message conforms to RFCs 2045 and 2046
- Content-Type
  - Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner

The Five Header Fields Defined in MIME (conti…)

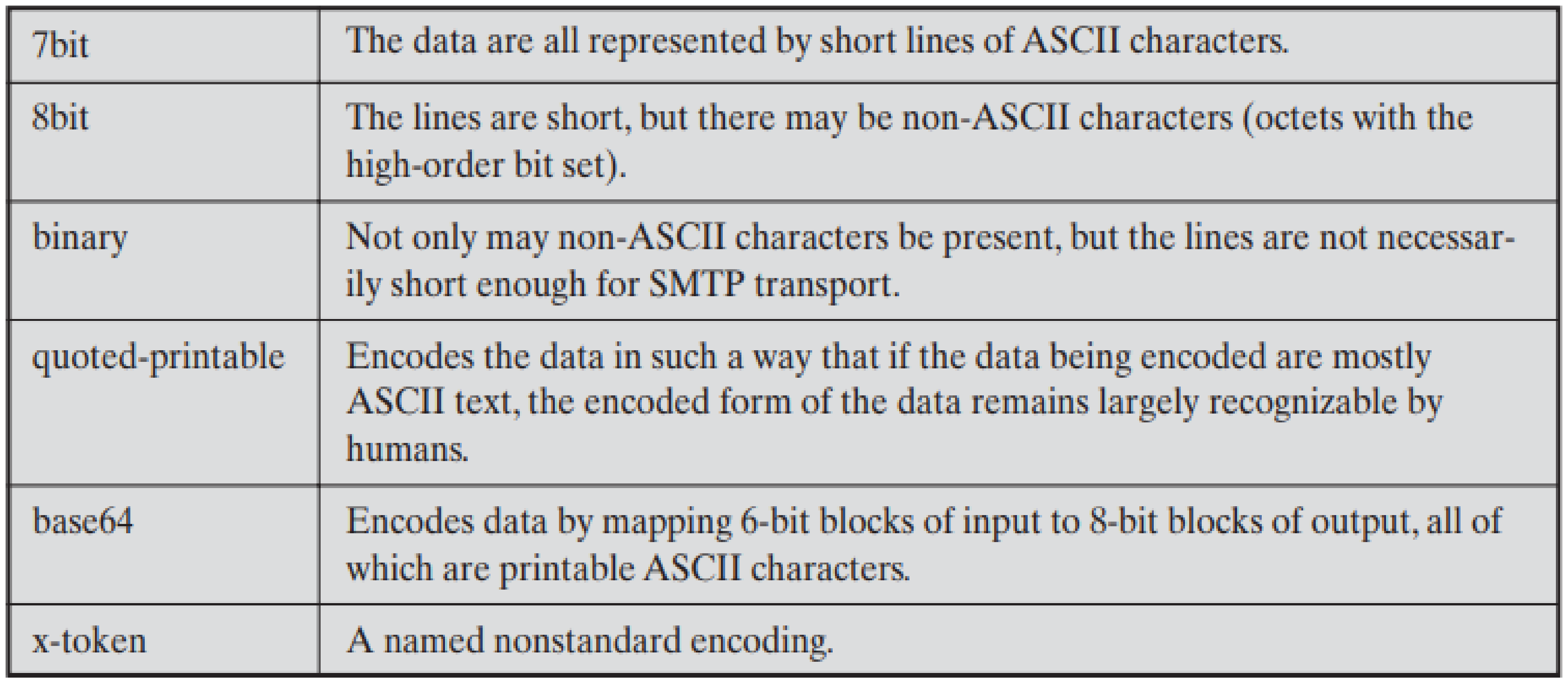
- Content-Transfer-Encoding
  - Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport
- Content-ID
  - Used to identify MIME entities uniquely in multiple contexts
- Content-Description
  - A text description of the object with the body; this is useful when the object is not readable

MIME Content Types



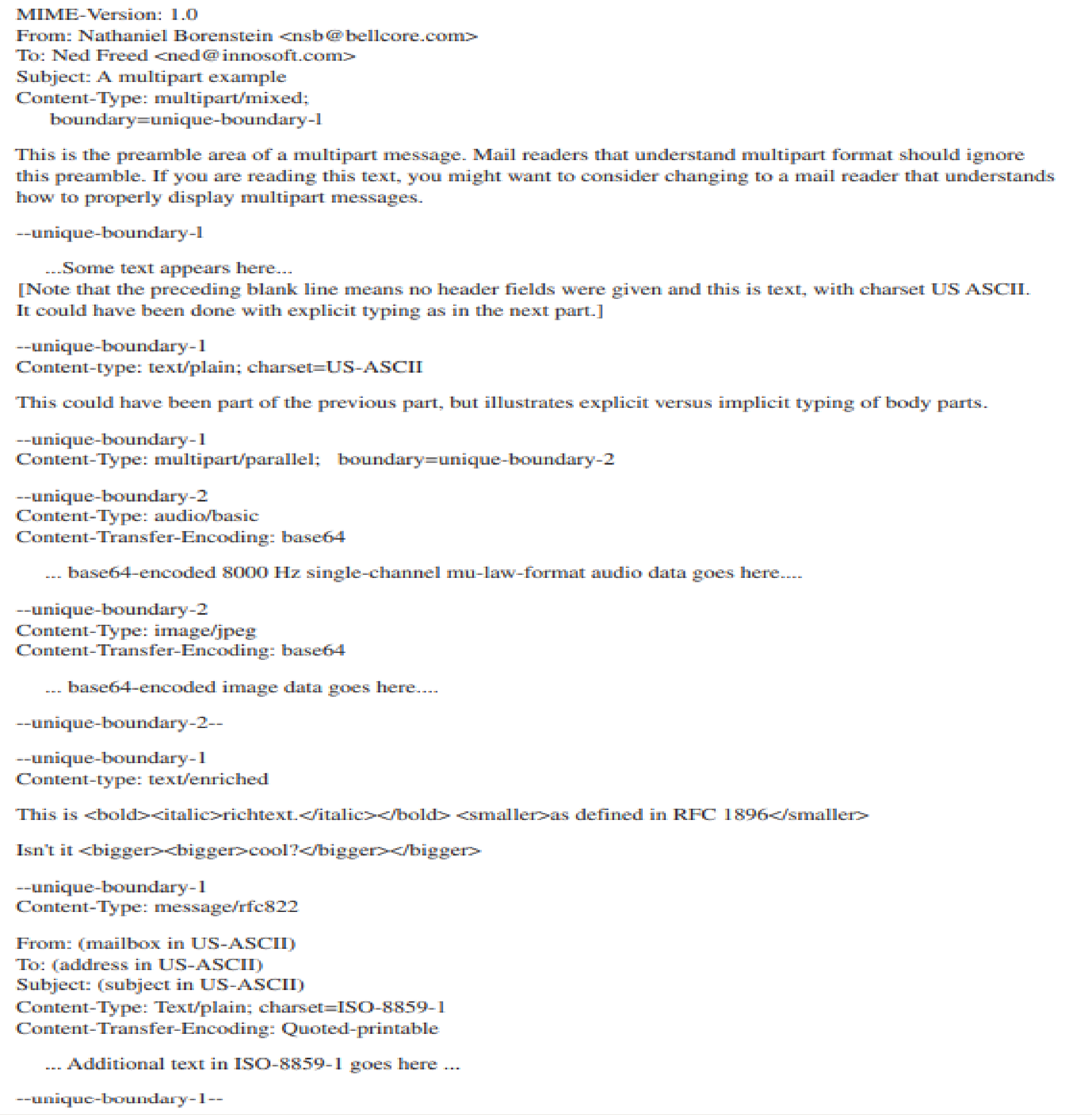
Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.
	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript format.
	octet-stream	General binary data consisting of 8-bit bytes.

MIME Transfer Encodings



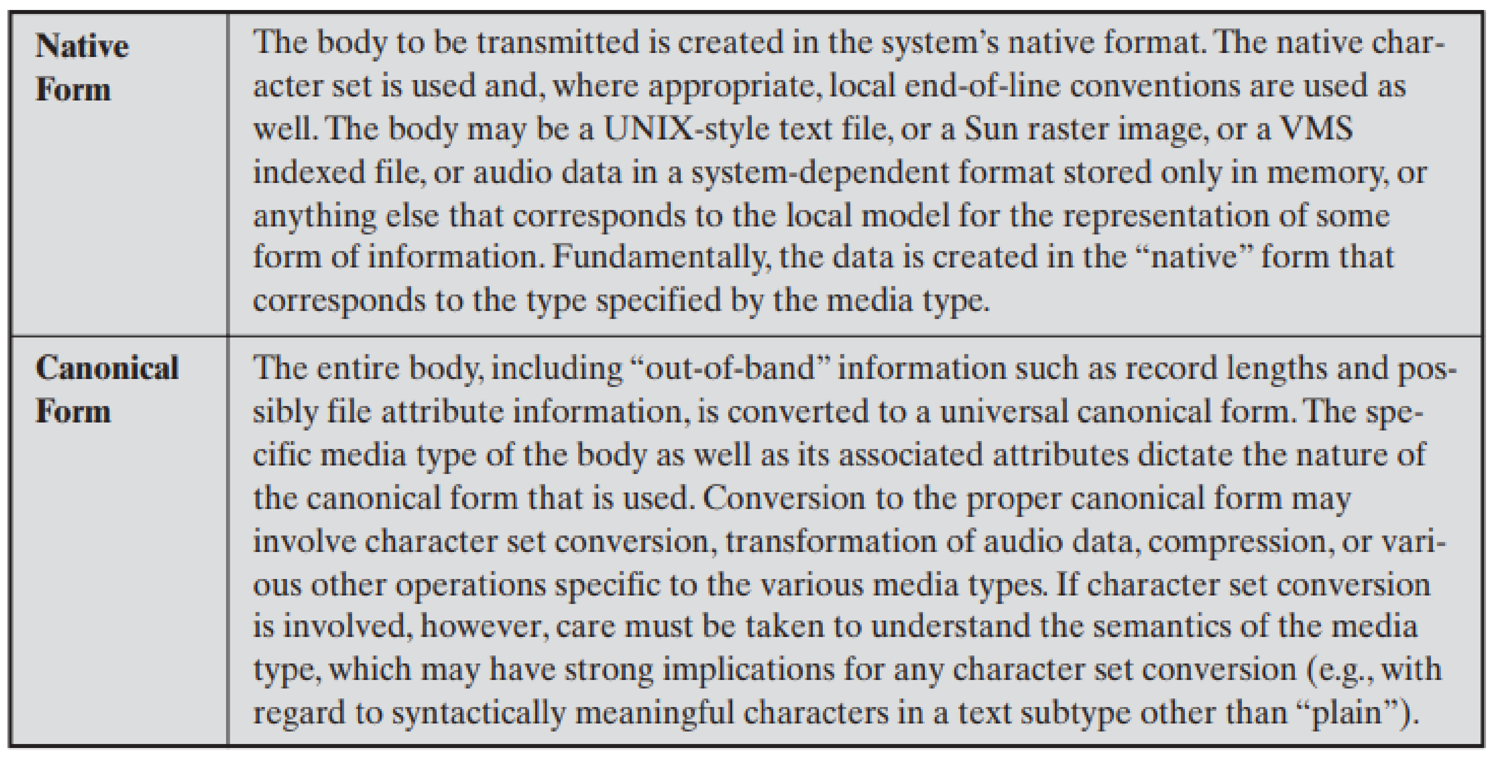
7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters (octets with the high-order bit set).
binary	Not only may non-ASCII characters be present, but the lines are not necessarily short enough for SMTP transport.
quoted-printable	Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

Example MIME Message Structure



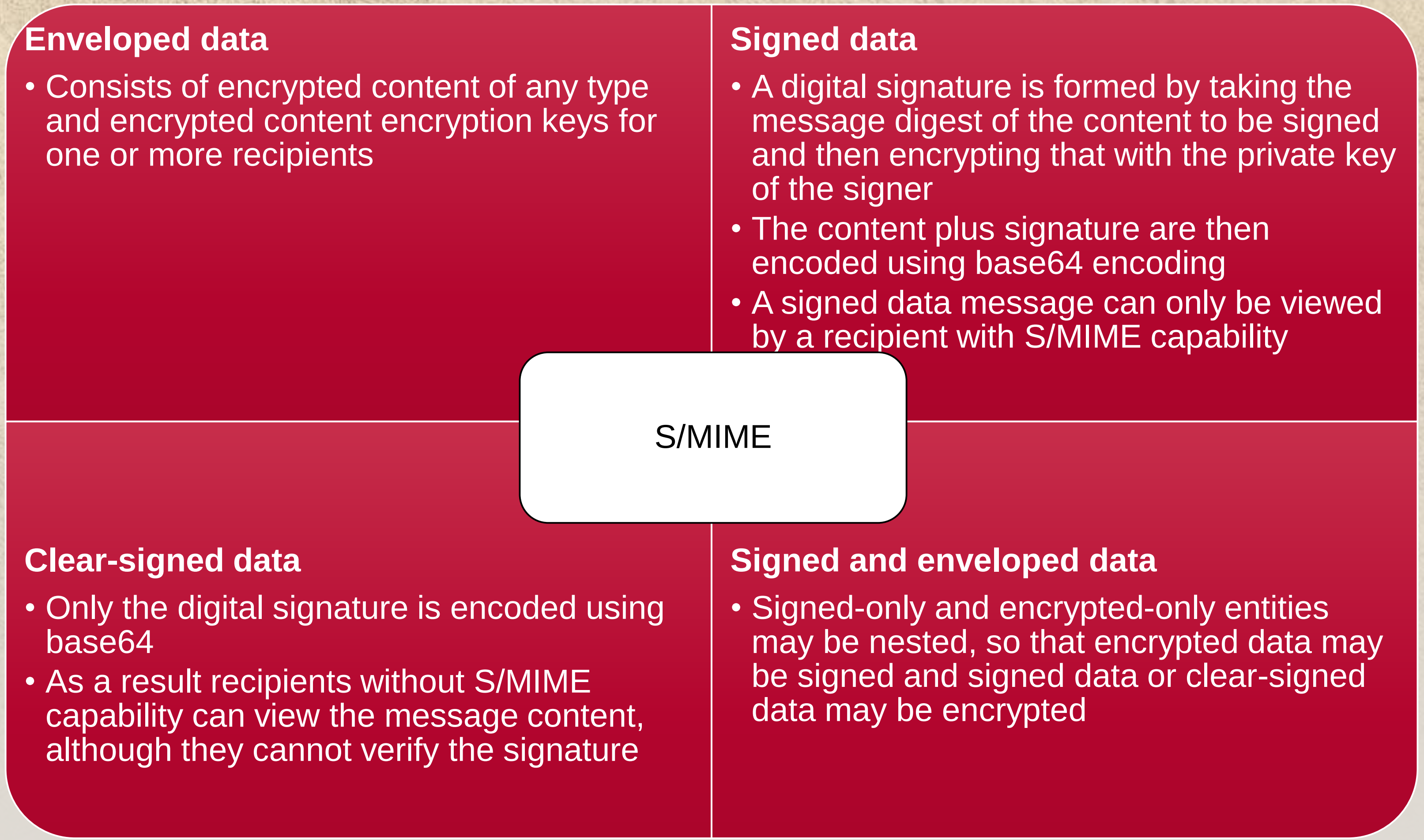
```
MIME-Version: 1.0
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A multipart example
Content-Type: multipart/mixed;
     boundary=unique-boundary-1

This is the preamble area of a multipart message. Mail readers that understand multipart format should ignore
this preamble. If you are reading this text, you might want to consider changing to a mail reader that understands
how to properly display multipart messages.

--unique-boundary-1

    ...Some text appears here...
[Note that the preceding blank line means no header fields were given and this is text, with charset US ASCII.
It could have been done with explicit typing as in the next part.]

--unique-boundary-1
Content-type: text/plain; charset=US-ASCII

This could have been part of the previous part, but illustrates explicit versus implicit typing of body parts.

--unique-boundary-1
Content-Type: multipart/parallel;   boundary=unique-boundary-2

--unique-boundary-2
Content-Type: audio/basic
Content-Transfer-Encoding: base64

    ... base64-encoded 8000 Hz single-channel mu-law-format audio data goes here....

--unique-boundary-2
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

    ... base64-encoded image data goes here....

--unique-boundary-2--

--unique-boundary-1
Content-type: text/enriched

This is <bold><italic>richtext.</italic></bold> <smaller>as defined in RFC 1896</smaller>

Isn't it <bigger><bigger>cool?</bigger></bigger>

--unique-boundary-1
Content-Type: message/rfc822

From: (mailbox in US-ASCII)
To: (address in US-ASCII)
Subject: (subject in US-ASCII)
Content-Type: Text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: Quoted-printable

    ... Additional text in ISO-8859-1 goes here ...

--unique-boundary-1--
```

Native And Canonical Form



Native Form	The body to be transmitted is created in the system's native format. The native character set is used and, where appropriate, local end-of-line conventions are used as well. The body may be a UNIX-style text file, or a Sun raster image, or a VMS indexed file, or audio data in a system-dependent format stored only in memory, or anything else that corresponds to the local model for the representation of some form of information. Fundamentally, the data is created in the "native" form that corresponds to the type specified by the media type.
Canonical Form	The entire body, including "out-of-band" information such as record lengths and possibly file attribute information, is converted to a universal canonical form. The specific media type of the body as well as its associated attributes dictate the nature of the canonical form that is used. Conversion to the proper canonical form may involve character set conversion, transformation of audio data, compression, or various other operations specific to the various media types. If character set conversion is involved, however, care must be taken to understand the semantics of the media type, which may have strong implications for any character set conversion (e.g., with regard to syntactically meaningful characters in a text subtype other than "plain").

S/MIME Functionality



S/MIME

Enveloped data

- Consists of encrypted content of any type and encrypted content encryption keys for one or more recipients

Signed data

- A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer
- The content plus signature are then encoded using base64 encoding
- A signed data message can only be viewed by a recipient with S/MIME capability

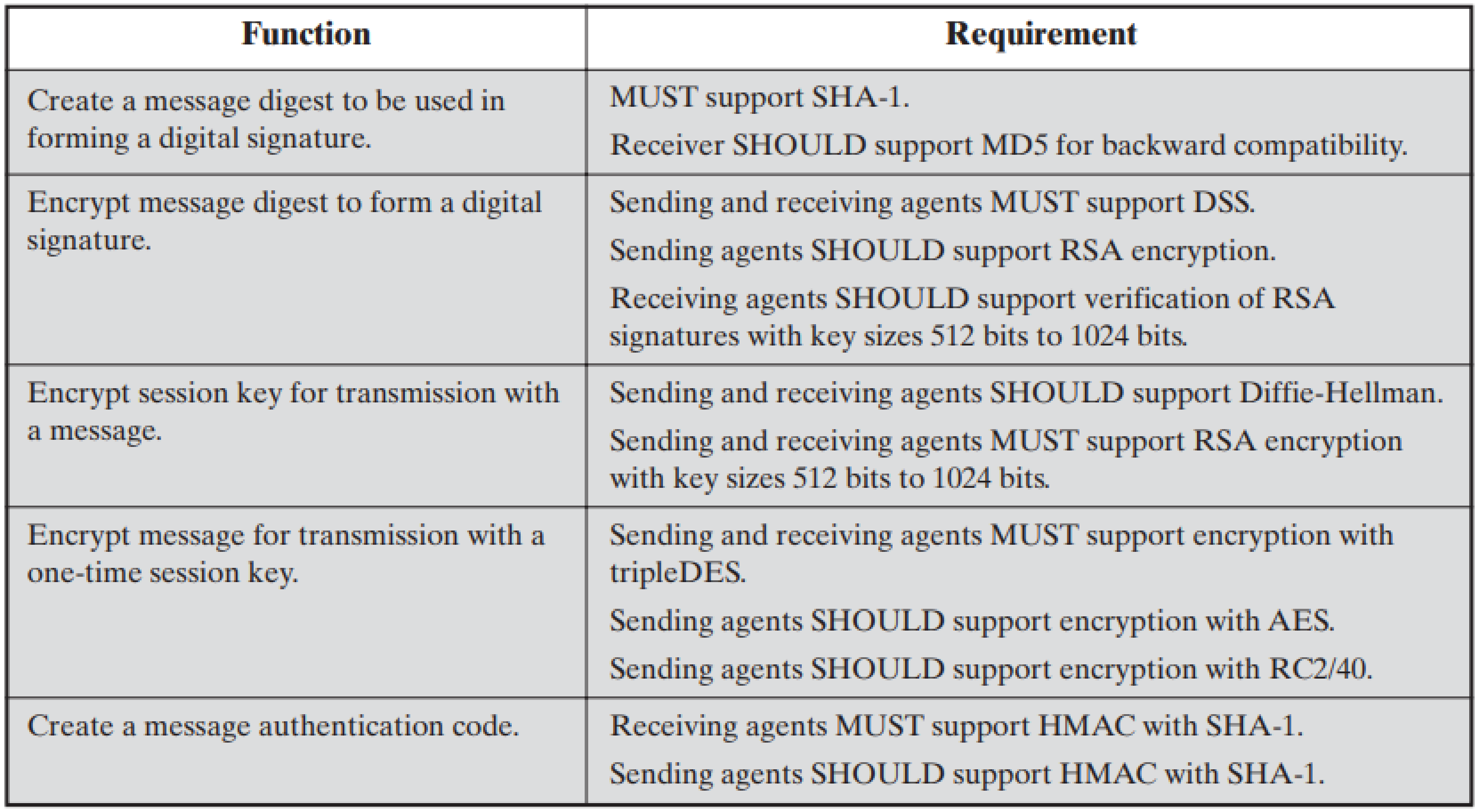
Clear-signed data

- Only the digital signature is encoded using base64
- As a result recipients without S/MIME capability can view the message content, although they cannot verify the signature

Signed and enveloped data

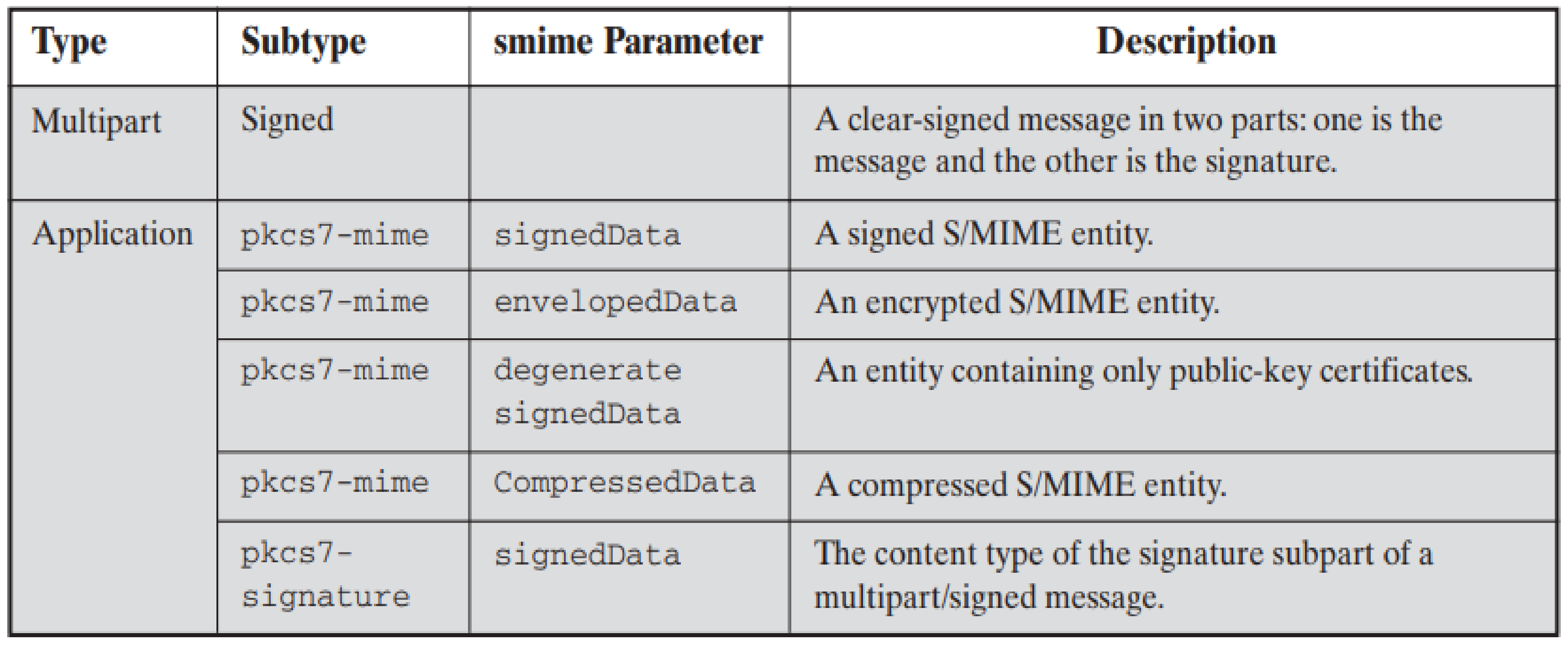
- Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted

Cryptographic Algorithms Used in S/MIME



Function	Requirement
Create a message digest to be used in forming a digital signature.	MUST support SHA-1. Receiver SHOULD support MD5 for backward compatibility.
Encrypt message digest to form a digital signature.	Sending and receiving agents MUST support DSS. Sending agents SHOULD support RSA encryption. Receiving agents SHOULD support verification of RSA signatures with key sizes 512 bits to 1024 bits.
Encrypt session key for transmission with a message.	Sending and receiving agents SHOULD support Diffie-Hellman. Sending and receiving agents MUST support RSA encryption with key sizes 512 bits to 1024 bits.
Encrypt message for transmission with a one-time session key.	Sending and receiving agents MUST support encryption with tripleDES. Sending agents SHOULD support encryption with AES. Sending agents SHOULD support encryption with RC2/40.
Create a message authentication code.	Receiving agents MUST support HMAC with SHA-1. Sending agents SHOULD support HMAC with SHA-1.

S/MIME Content Types



Type	Subtype	smime Parameter	Description
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the signature.
Application	pkcs7-mime	signedData	A signed S/MIME entity.
	pkcs7-mime	envelopedData	An encrypted S/MIME entity.
	pkcs7-mime	degenerate signedData	An entity containing only public-key certificates.
	pkcs7-mime	CompressedData	A compressed S/MIME entity.
	pkcs7-signature	signedData	The content type of the signature subpart of a multipart/signed message.

Securing a MIME Entity

- S/MIME secures a MIME entity with a signature, encryption, or both
- The MIME entity is prepared according to the normal rules for MIME message preparation
  - The MIME entity plus some security-related data, such as algorithm identifiers and certificates, are processed by S/MIME to produce what is known as a PKCS object
  - A PKCS object is then treated as message content and wrapped in MIME
- In all cases the message to be sent is converted to canonical form

EnvelopedData

- The steps for preparing an envelopedData MIME entity are:
  - Generate a pseudorandom session key for a particular symmetric encryption algorithm
  - For each recipient, encrypt the session key with the recipient’s public RSA key
  - For each recipient, prepare a block known as RecipientInfo that contains an identifier of the recipient’s public-key certificate, an identifier of the algorithm used to encrypt the session key, and the encrypted session key
  - Encrypt the message content with the session key

SignedData

- The steps for preparing a signedData MIME entity are:
  - Select a message digest algorithm (SHA or MD5)
  - Compute the message digest (hash function) of the content to be signed
  - Encrypt the message digest with the signer’s private key
  - Prepare a block known as SignerInfo that contains the signer’s public key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and the encrypted message digest

Clear Signing

- Achieved using the multipart content type with a signed subtype
- This signing process does not involve transforming the message to be signed
- Recipients with MIME capability but not S/MIME capability are able to read the incoming message

S/MIME CERTIFICATE PROCESSING

- S/MIME uses public-key certificates that conform to version 3 of X.509
- The key-management scheme used by S/MIME is in some ways a hybrid between a strict X.509 certification hierarchy and PGP’s web of trust
- S/MIME managers and/or users must configure each client with a list of trusted keys and with certificate revocation lists
  - The responsibility is local for maintaining the certificates needed to verify incoming signatures and to encrypt outgoing messages
- The certificates are signed by certification authorities

User Agent Role

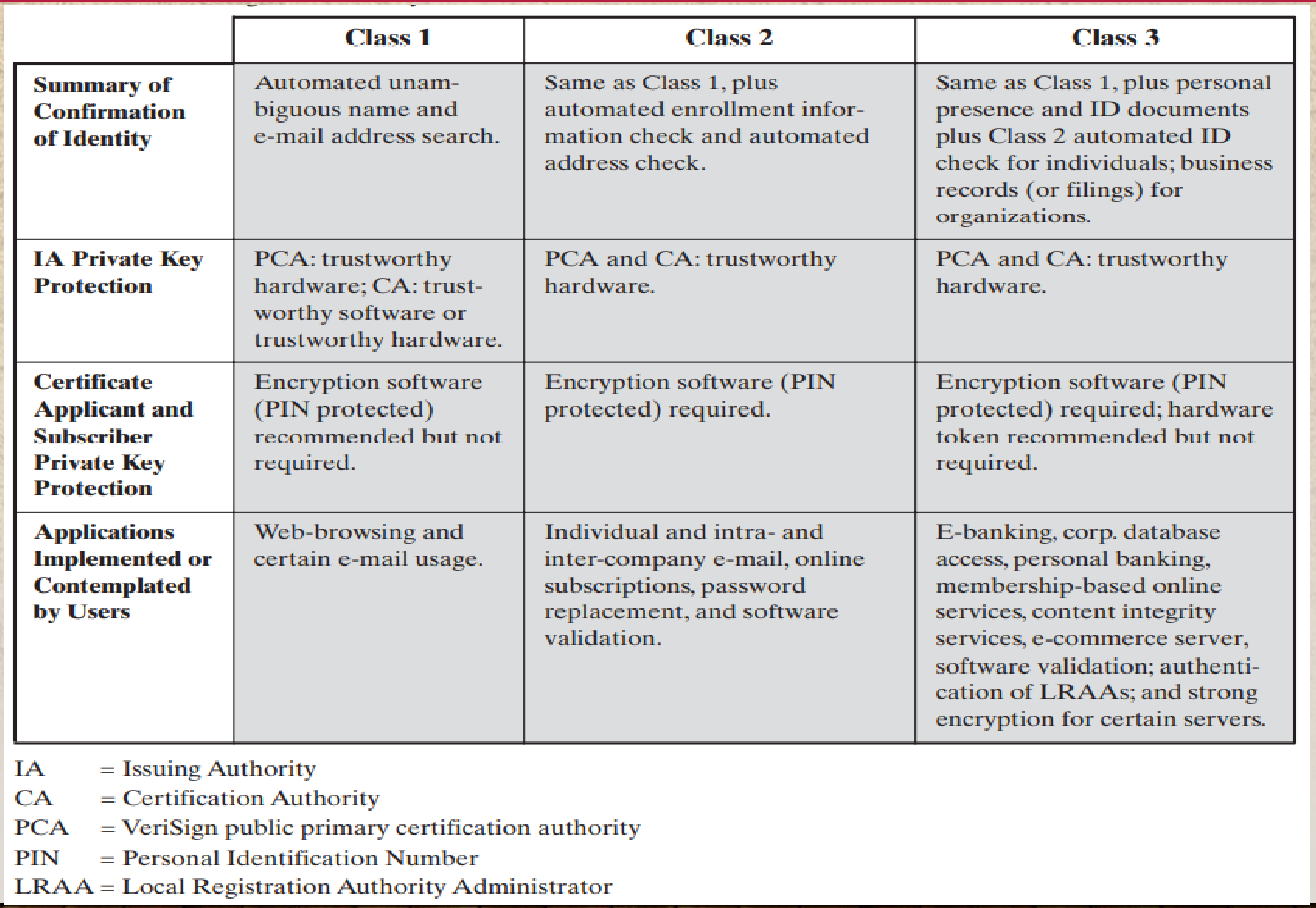
- An S/MIME user has several key-management functions to perform.

1. **Key generation:**
   - The user of some related administrative utility MUST be capable of generating separate Diffie-Hellman and DSS key pairs and SHOULD be capable of generating RSA key pairs.
   - A user agent SHOULD generate RSA key pairs with a length in the range of 768 to 1024 bits and MUST NOT generate a length of less than 512 bits.
2. **Registration:**
   - A user's public key must be registered with a certification authority in order to receive an X.509 public-key certificate.
3. **Certificate storage and retrieval:**
   - A user requires access to a local list of certificates in order to verify incoming signatures and to encrypt outgoing messages.

VeriSign Certificates

- VeriSign provides a certification authority (CA) service that is intended to be compatible with S/MIME and a variety of other applications
- Issues X.509 certificates with the product name VeriSign Digital ID
- At a minimum, each Digital ID contains:
  - Owner’s public key
  - Owner’s name or alias
  - Expiration date of the Digital ID
  - Serial number of the Digital ID
  - Name of the certification authority that issued the Digital ID
  - Digital signature of the certification authority that issued the Digital ID

VeriSign Public-key Certificate Classes



	Class 1	Class 2	Class 3
Summary of Confirmation of Identity	Automated unambiguous name and e-mail address search.	Same as Class 1, plus automated enrollment information check and automated address check.	Same as Class 1, plus personal presence and ID documents plus Class 2 automated ID check for individuals; business records (or filings) for organizations.
IA Private Key Protection	PCA: trustworthy hardware; CA: trustworthy software or trustworthy hardware.	PCA and CA: trustworthy hardware.	PCA and CA: trustworthy hardware.
Certificate Applicant and Subscriber Private Key Protection	Encryption software (PIN protected) recommended but not required.	Encryption software (PIN protected) required.	Encryption software (PIN protected) required; hardware token recommended but not required.
Applications Implemented or Contemplated by Users	Web-browsing and certain e-mail usage.	Individual and intra- and inter-company e-mail, online subscriptions, password replacement, and software validation.	E-banking, corp. database access, personal banking, membership-based online services, content integrity services, e-commerce server, software validation; authentication of LRAAs; and strong encryption for certain servers.

IA	=	Issuing Authority
CA	=	Certification Authority
PCA	=	VeriSign public primary certification authority
PIN	=	Personal Identification Number
LRAA	=	Local Registration Authority Administrator

ENHANCED SECURITY SERVICES

- Three enhanced security services have been proposed in an Internet draft:
  - Signed receipt
    - Returning a signed receipt provides proof of delivery to the originator of a message and allows the originator to demonstrate to a third party that the recipient received the message
  - Security labels
    - A set of security information regarding the sensitivity of the content that is protected by S/MIME encapsulation
  - Secure mailing lists
    - An S/MIME Mail List Agent (MLA) can take a single incoming message, perform the recipient-specific encryption for each recipient, and forward the message

Thank you

IP SECURITY

IP SECURITY

- Have a range of application specific security mechanisms
  - eg. S/MIME, PGP, Kerberos, SSL/HTTPS
- However, there are security concerns that cut across protocol layers
- Would like security implemented by the network for all applications
- IP-level security encompasses three functional areas:
  - authentication,
  - confidentiality, and
  - key management

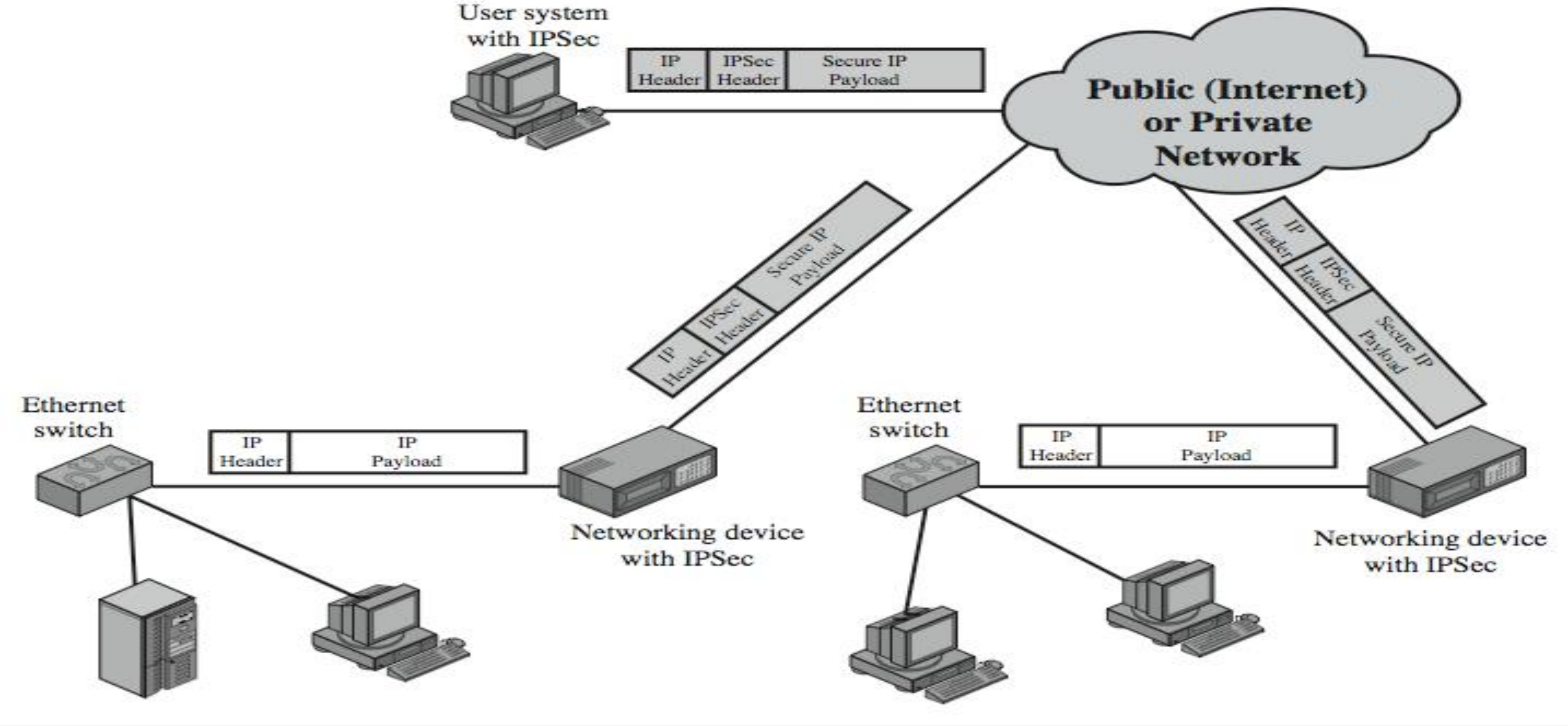
IP SECURITY OVERVIEW

- In 1994, the Internet Architecture Board (IAB) issued a report titled "Security in the Internet Architecture"
  - report identified key areas for security mechanisms
- To provide security
  - IAB included authentication and encryption as necessary security features in IPv4 and IPv6
- IPsec specification exists as a set of Internet standards

APPLICATIONS OF IPsec

- It provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet.
- Examples:
  - Secure branch office connectivity over the Internet
  - Secure remote access over the Internet
  - Establishing extranet and intranet connectivity with partners
  - Enhancing electronic commerce security
- The principal feature of IPsec that it can encrypt and/or authenticate all traffic at the IP level

IP SECURITY USES



BENEFITS OF IPsec

- in a firewall/router provides strong security to all traffic crossing the perimeter
- in a firewall/router is resistant to bypass
- is below transport layer, hence transparent to applications
- can be transparent to end users
- can provide security for individual users
- secures routing architecture

IP SECURITY ARCHITECTURE

- Specification is quite complex, with groups:
  - Architecture
    - RFC4301 *Security Architecture for Internet Protocol*
  - Authentication Header (AH)
    - RFC4302 *IP Authentication Header*
  - Encapsulating Security Payload (ESP)
    - RFC4303 *IP Encapsulating Security Payload (ESP)*
  - Internet Key Exchange (IKE)
    - RFC4306 *Internet Key Exchange (IKEv2) Protocol*
  - Cryptographic algorithms

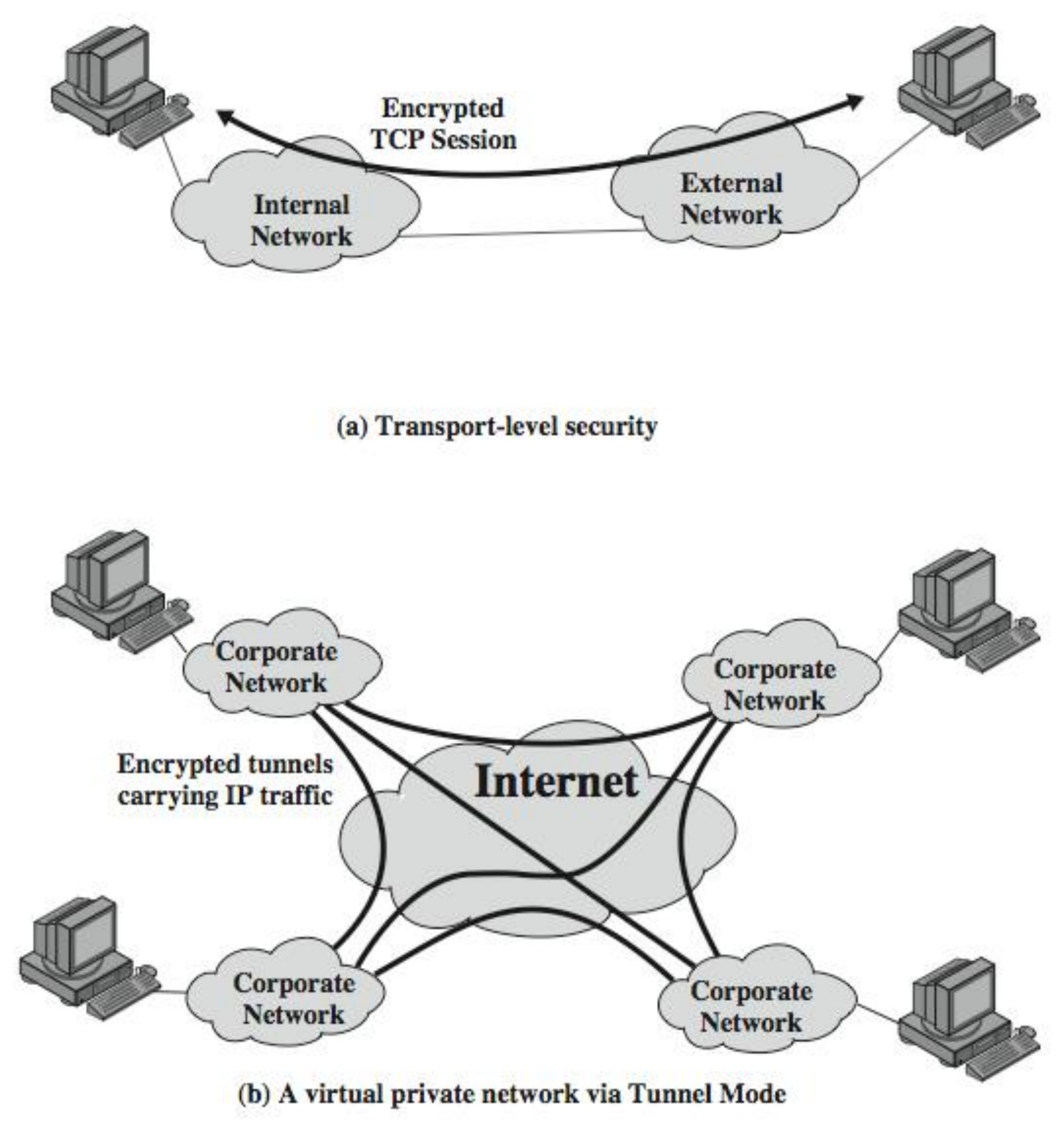
IPsec SERVICES

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets
  - a form of partial sequence integrity
- Confidentiality (encryption)
- Limited traffic flow confidentiality

TRANSPORT AND TUNNEL MODES

- Transport Mode
  - to encrypt & optionally authenticate IP data
  - can do traffic analysis but is efficient
  - good for ESP host to host traffic
- Tunnel Mode
  - encrypts entire IP packet
  - add new header for next hop
  - no routers on way can examine inner IP header
  - good for VPNs, gateway to gateway security

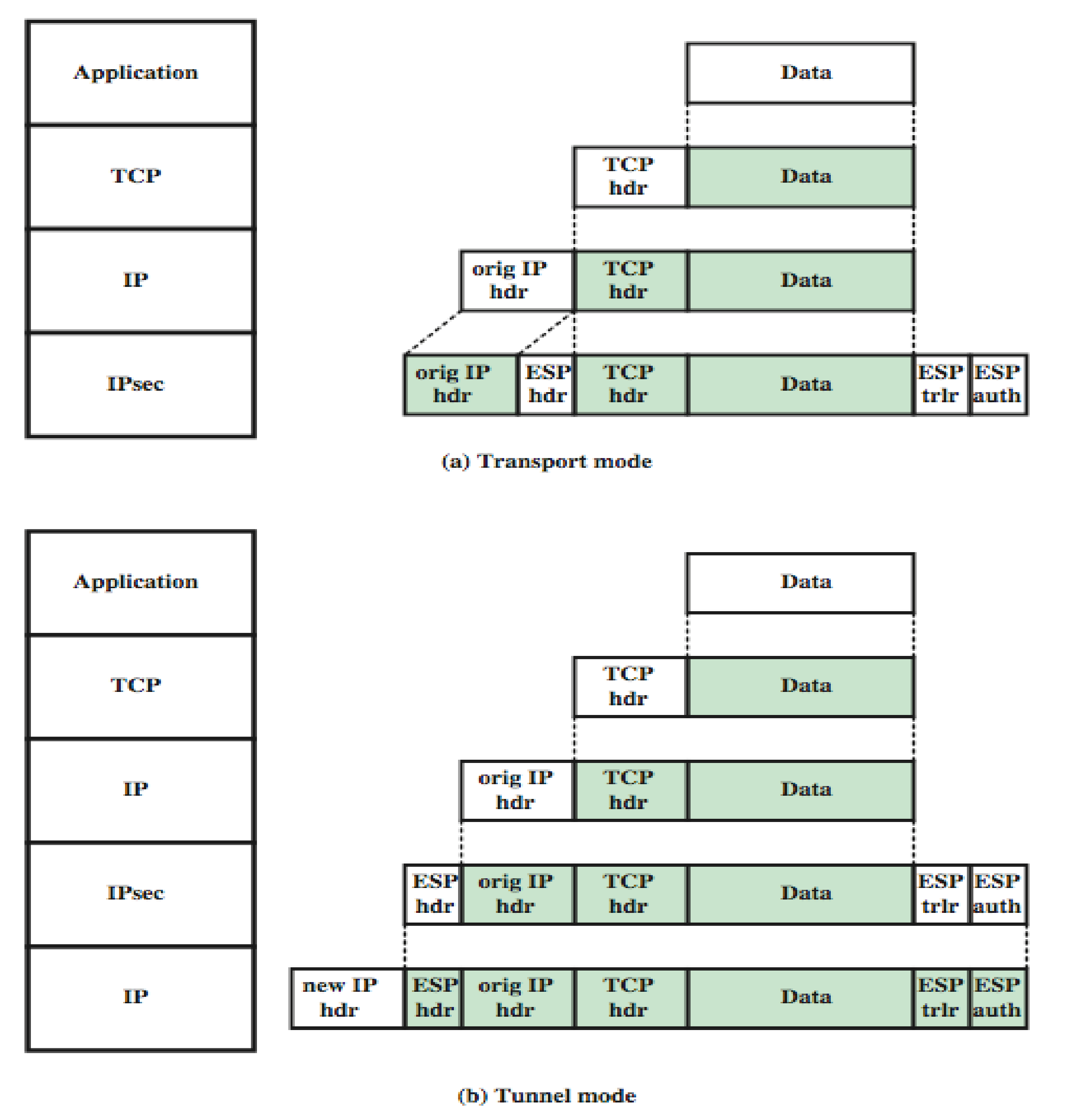
TRANSPORT AND TUNNEL MODES



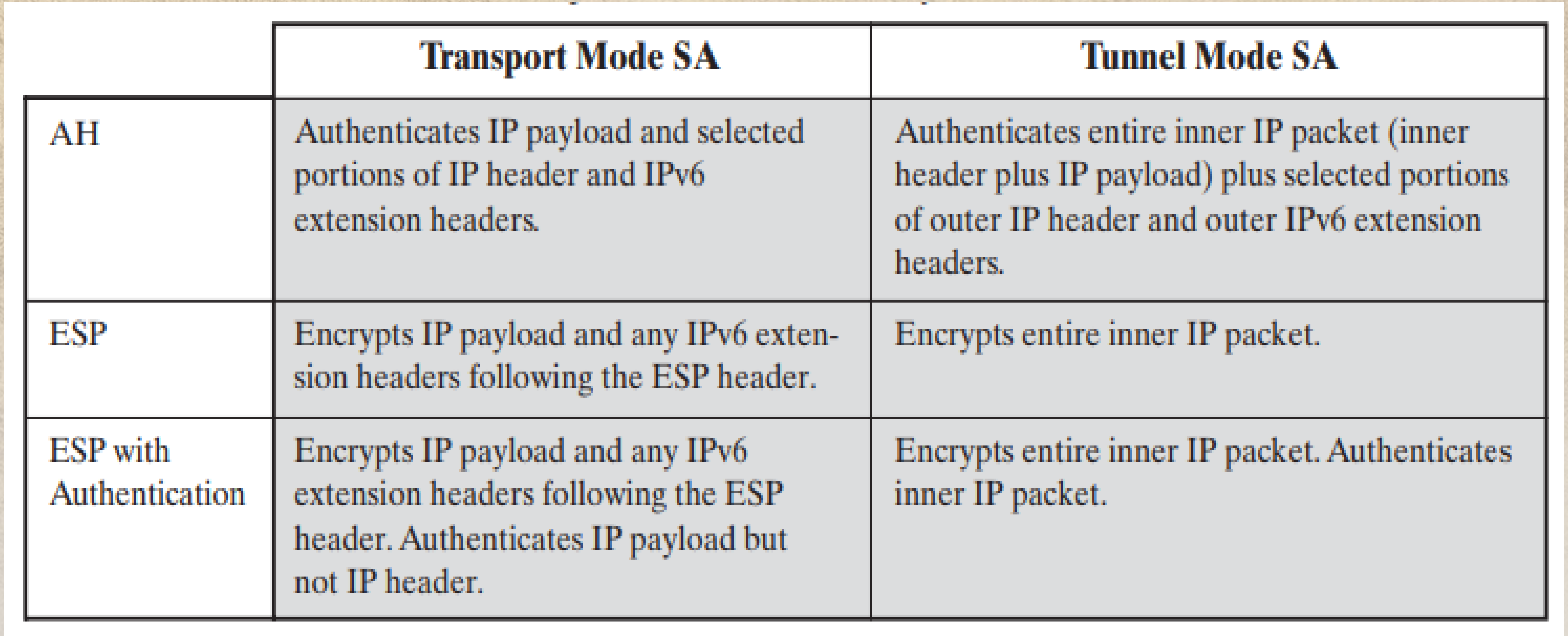
(a) Transport-level security

(b) A virtual private network via Tunnel Mode

TRANSPORT AND TUNNEL MODE PROTOCOLS



TUNNEL MODE AND TRANSPORT MODE FUNCTIONALITY

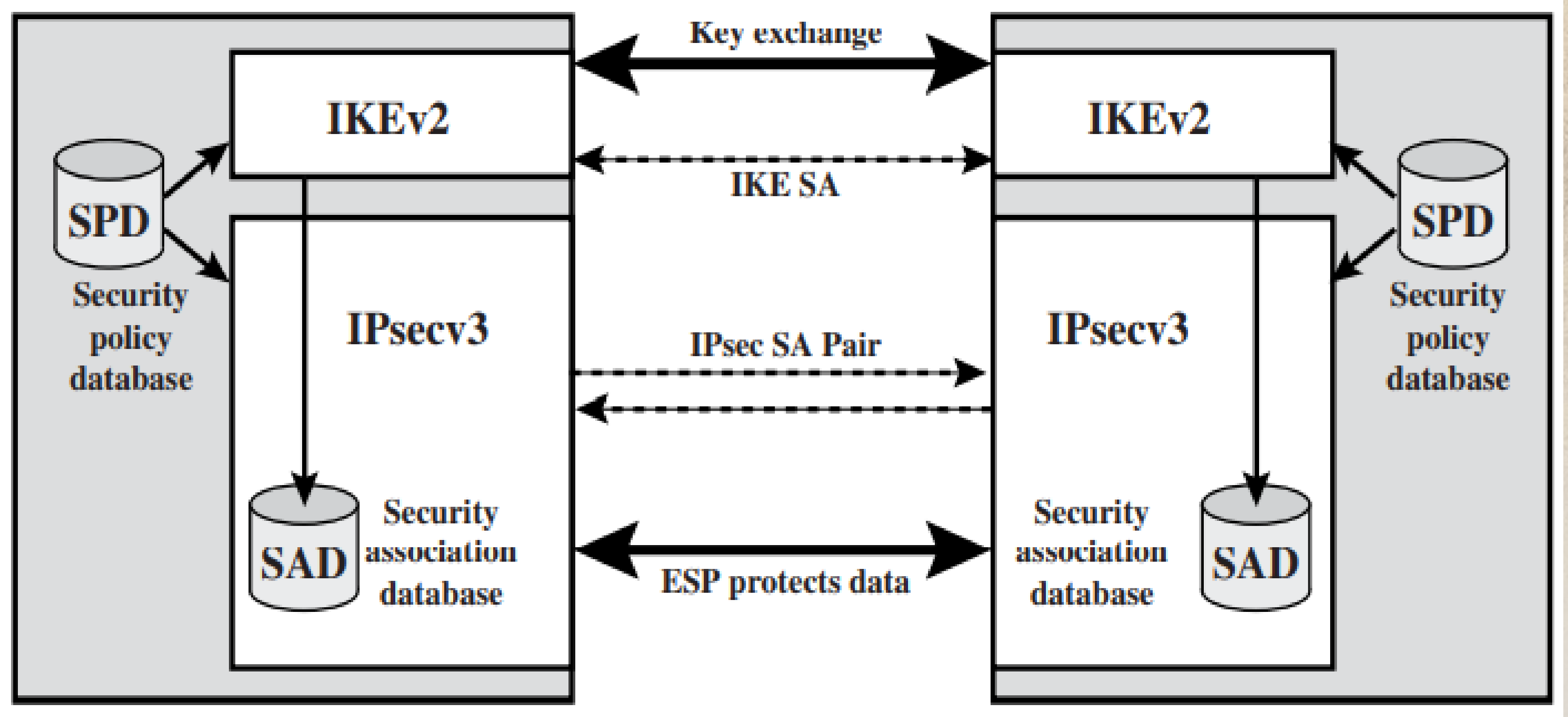


	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts entire inner IP packet.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts entire inner IP packet. Authenticates inner IP packet.

SECURITY ASSOCIATIONS

- a one-way relationship between sender & receiver that affords security for traffic flow
- defined by 3 parameters:
  - Security Parameters Index (SPI)
  - IP Destination Address
  - Security Protocol Identifier
- has a number of other parameters
  - seq no, AH & EH info, lifetime etc
- have a database of Security Associations

IPsec ARCHITECTURE

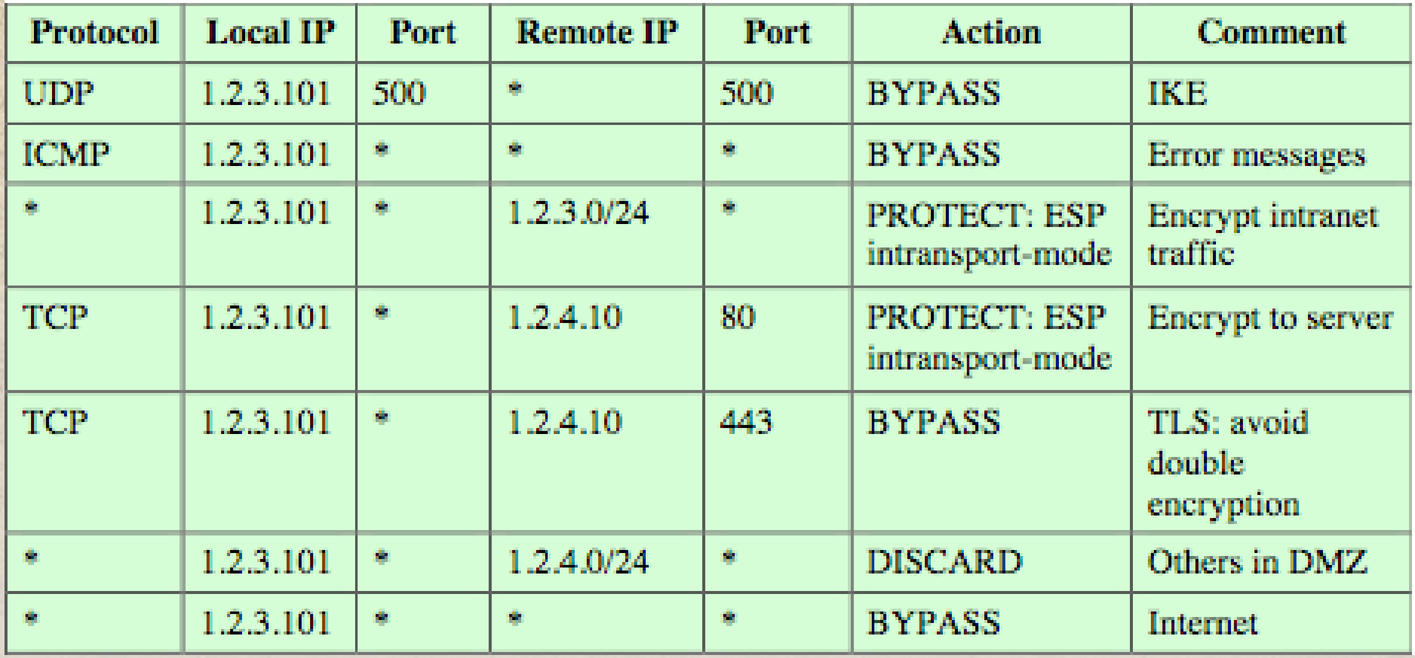


SECURITY ASSOCIATION DATABASE

- A security association is normally defined by the following parameters in an SAD entry:
  - Security Parameter Index
  - Sequence Number Counter
  - Sequence Counter Overflow
  - Anti-Replay Window
  - AH Information
  - ESP Information
  - Lifetime of this Security Association
  - IPsec Protocol Mode
  - Path MTU

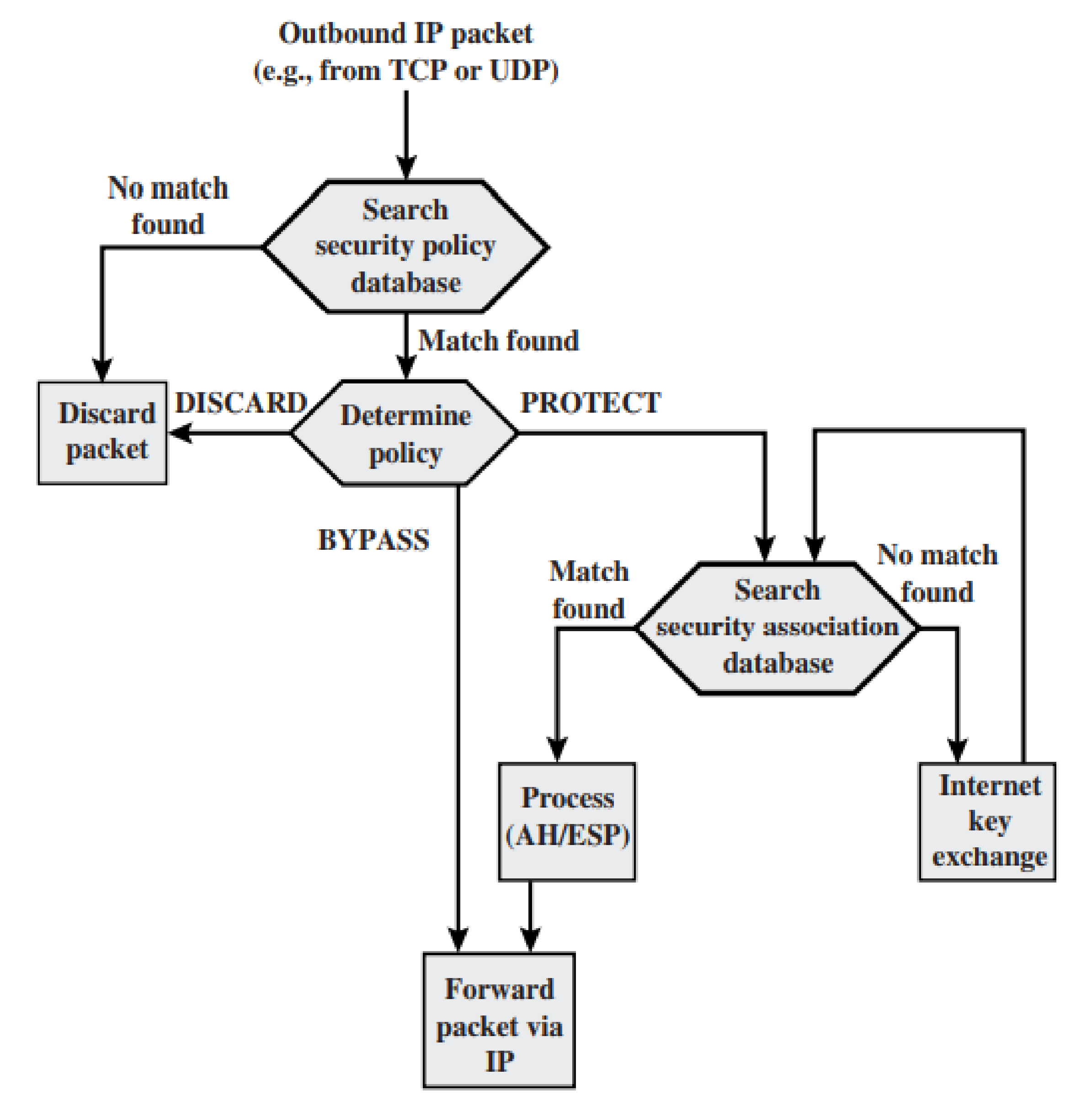
SECURITY POLICY DATABASE

- relates IP traffic to specific SAs
  - match subset of IP traffic to relevant SA
  - use selectors to filter outgoing traffic to map
  - based on local & remote IP addresses, next layer protocol, name, local & remote ports

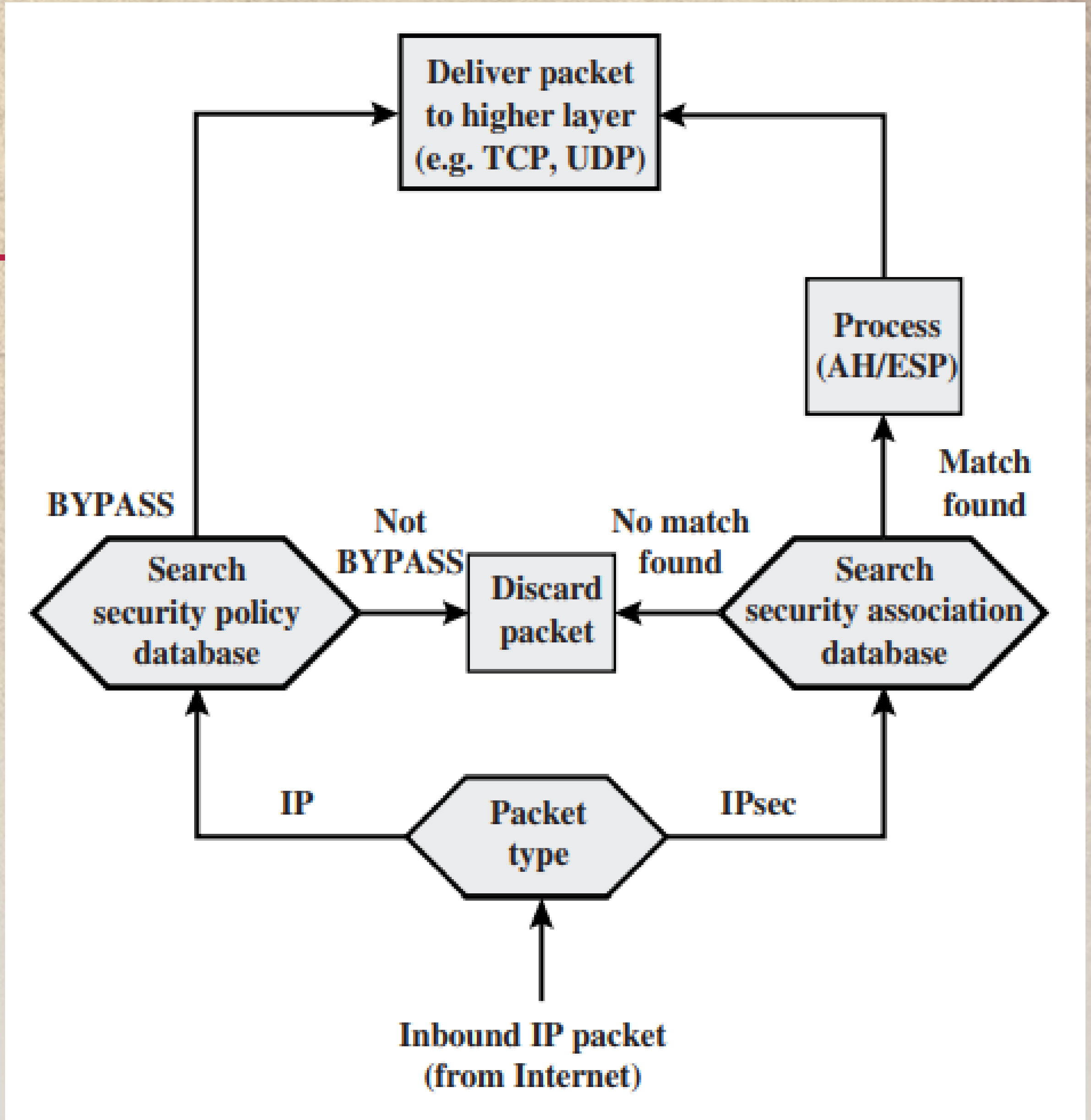


Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

PROCESSING MODEL FOR OUTBOUND PACKETS



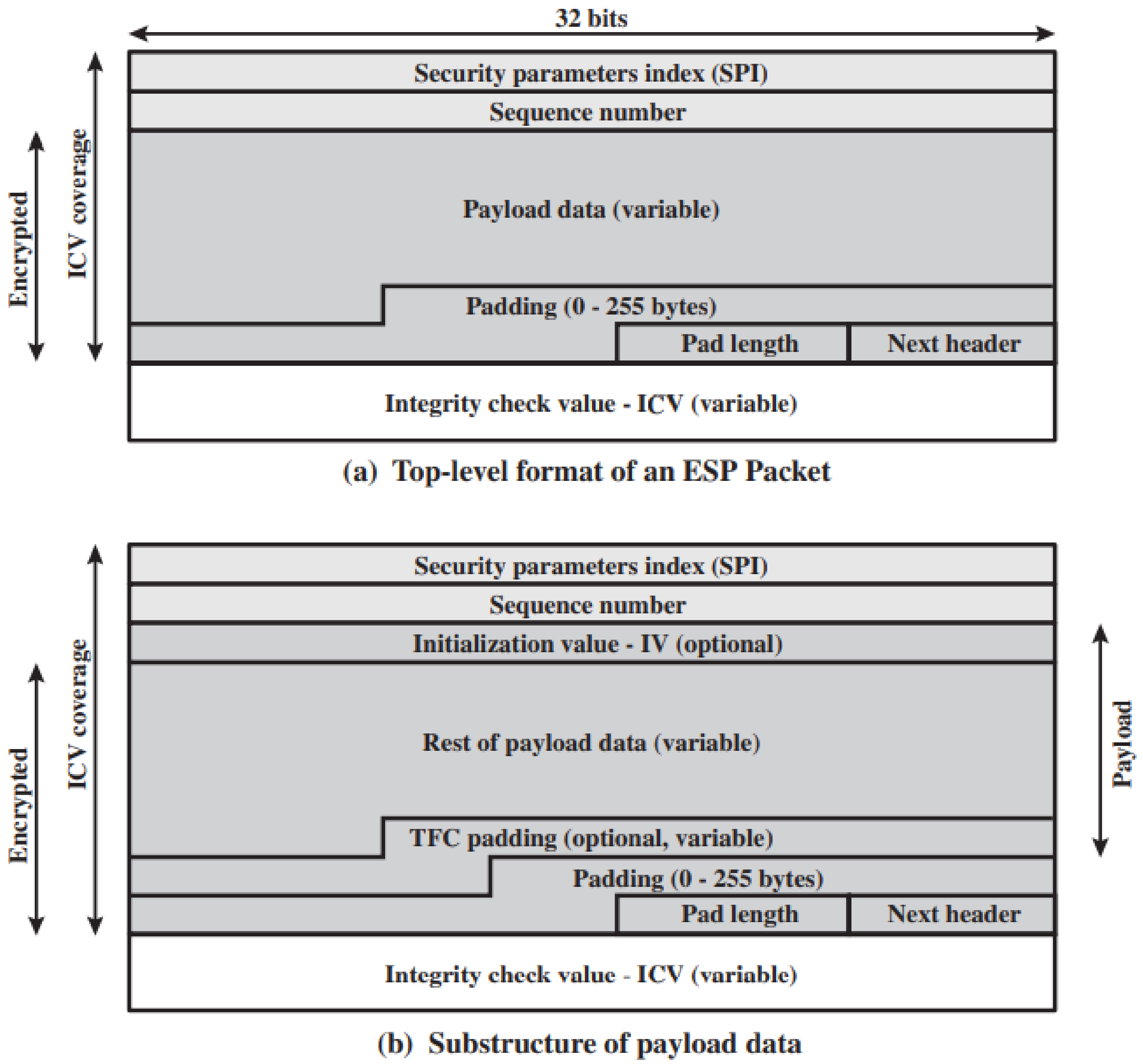
PROCESSING MODEL FOR INBOUND PACKETS


Deliver packet to higher layer (e.g. TCP, UDP)
Process (AH/ESP)
Match found
BYPASS
Not BYPASS
No match found
Search security policy database
Discard packet
Search security association database
IP
Packet type
IPsec
Inbound IP packet (from Internet)

ENCAPSULATING SECURITY PAYLOAD (ESP)

- provides message content confidentiality, data origin authentication, connectionless integrity, an anti-replay service, limited traffic flow confidentiality
- services depend on options selected when establish Security Association (SA), net location
- can use a variety of encryption & authentication algorithms

ESP PACKET FORMAT


32 bits
Encrypted
ICV coverage
Security parameters index (SPI)
Sequence number
Payload data (variable)
Padding (0 - 255 bytes)
Pad length
Next header
Integrity check value - ICV (variable)

(a) Top-level format of an ESP Packet

Encrypted
ICV coverage
Payload
Security parameters index (SPI)
Sequence number
Initialization value - IV (optional)
Rest of payload data (variable)
TFC padding (optional, variable)
Padding (0 - 255 bytes)
Pad length
Next header
Integrity check value - ICV (variable)

(b) Substructure of payload data

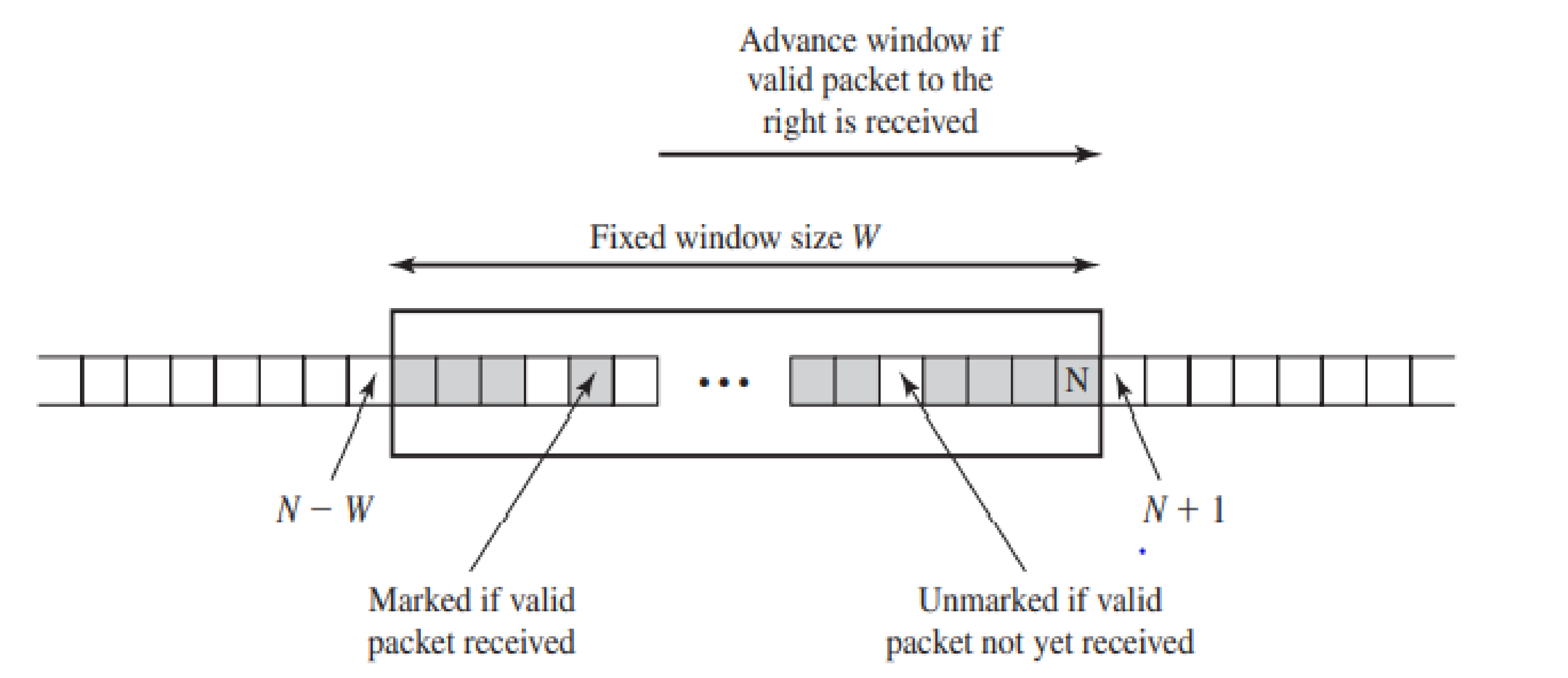
ENCRYPTION & AUTHENTICATION ALGORITHMS & PADDING

- ESP can encrypt payload data, padding, pad length, and next header fields
  - if needed have IV at start of payload data
- ESP can have optional ICV for integrity
  - is computed after encryption is performed
- ESP uses padding
  - to expand plaintext to required length
  - to align pad length and next header fields
  - to provide partial traffic flow confidentiality

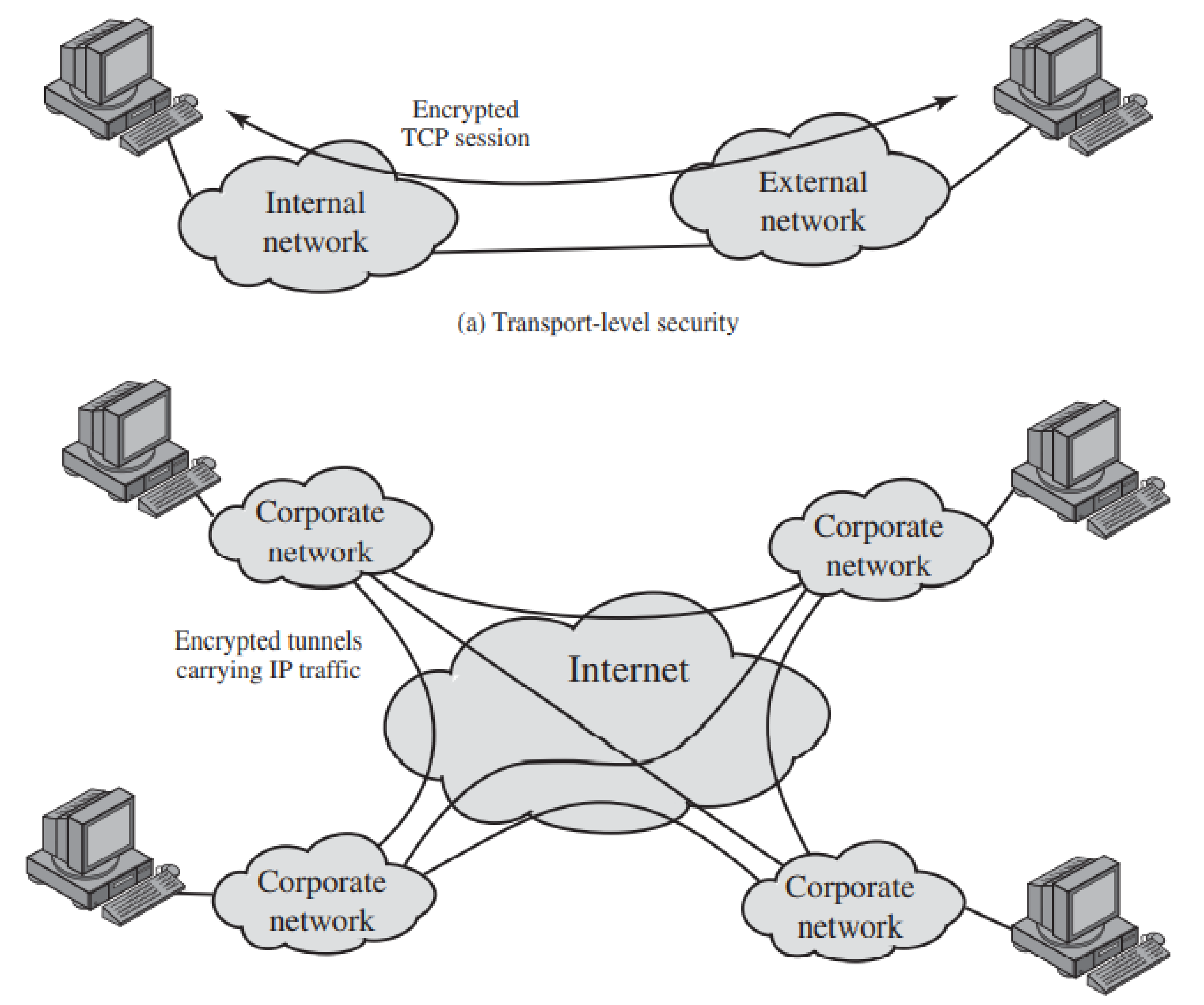
ANTI-REPLAY SERVICE

- replay is when attacker resends a copy of an authenticated packet
- use sequence number to thwart this attack
- sender initializes sequence number to 0 when a new SA is established
  - increment for each packet
  - must not exceed limit of $2^{32} - 1$
- receiver then accepts packets with seq no within window of $(N - W + 1)$

ANTI-REPLAY MECHANISM


Advance window if valid packet to the right is received
Fixed window size W
N
• • •
$N - W$
$N + 1$
Marked if valid packet received
Unmarked if valid packet not yet received

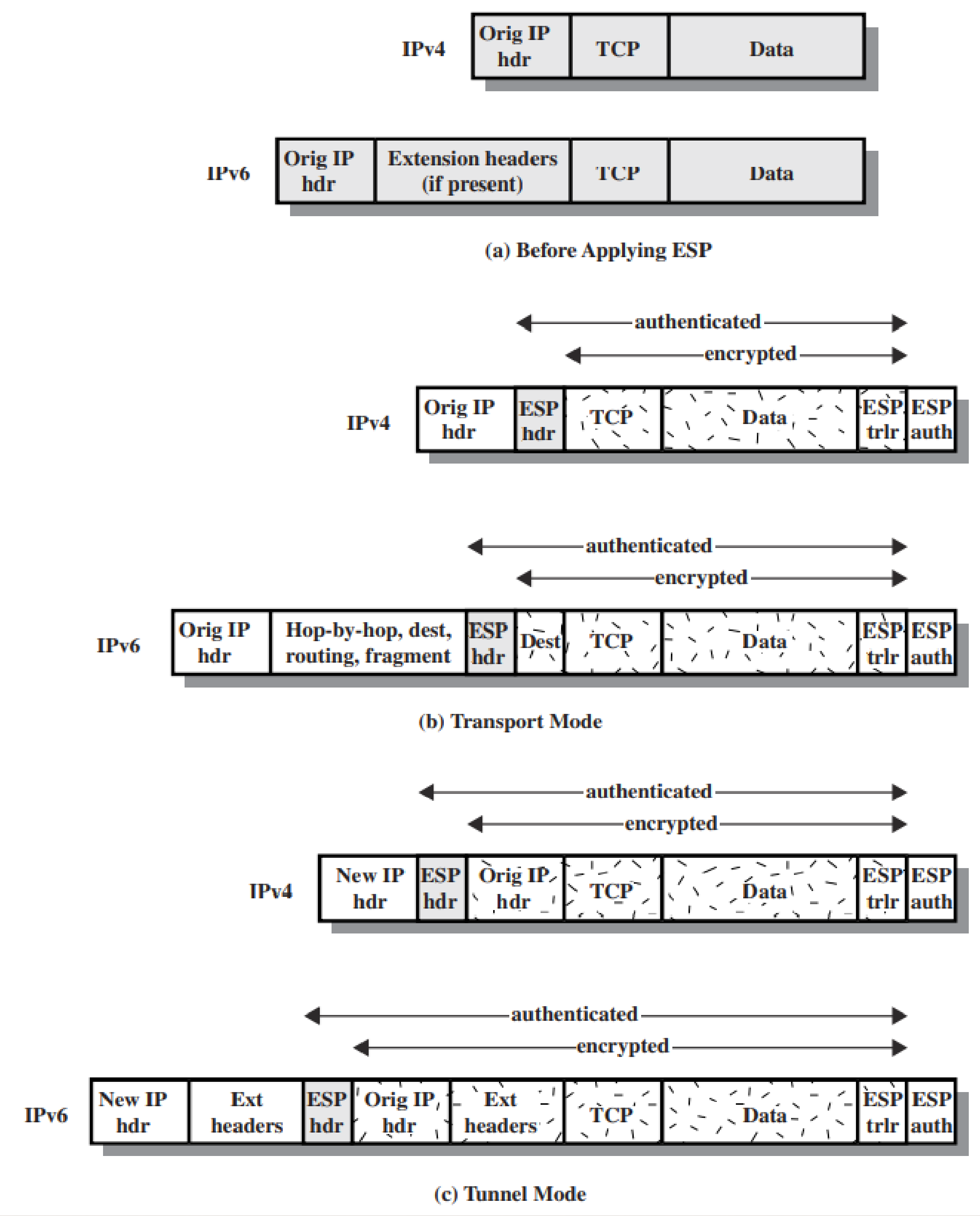
TRANSPORT AND TUNNEL MODES



(a) Transport-level security

(b) A virtual private network via tunnel mode

SCOPE OF ESP ENCRYPTION AND AUTHENTICATION



(a) Before Applying ESP

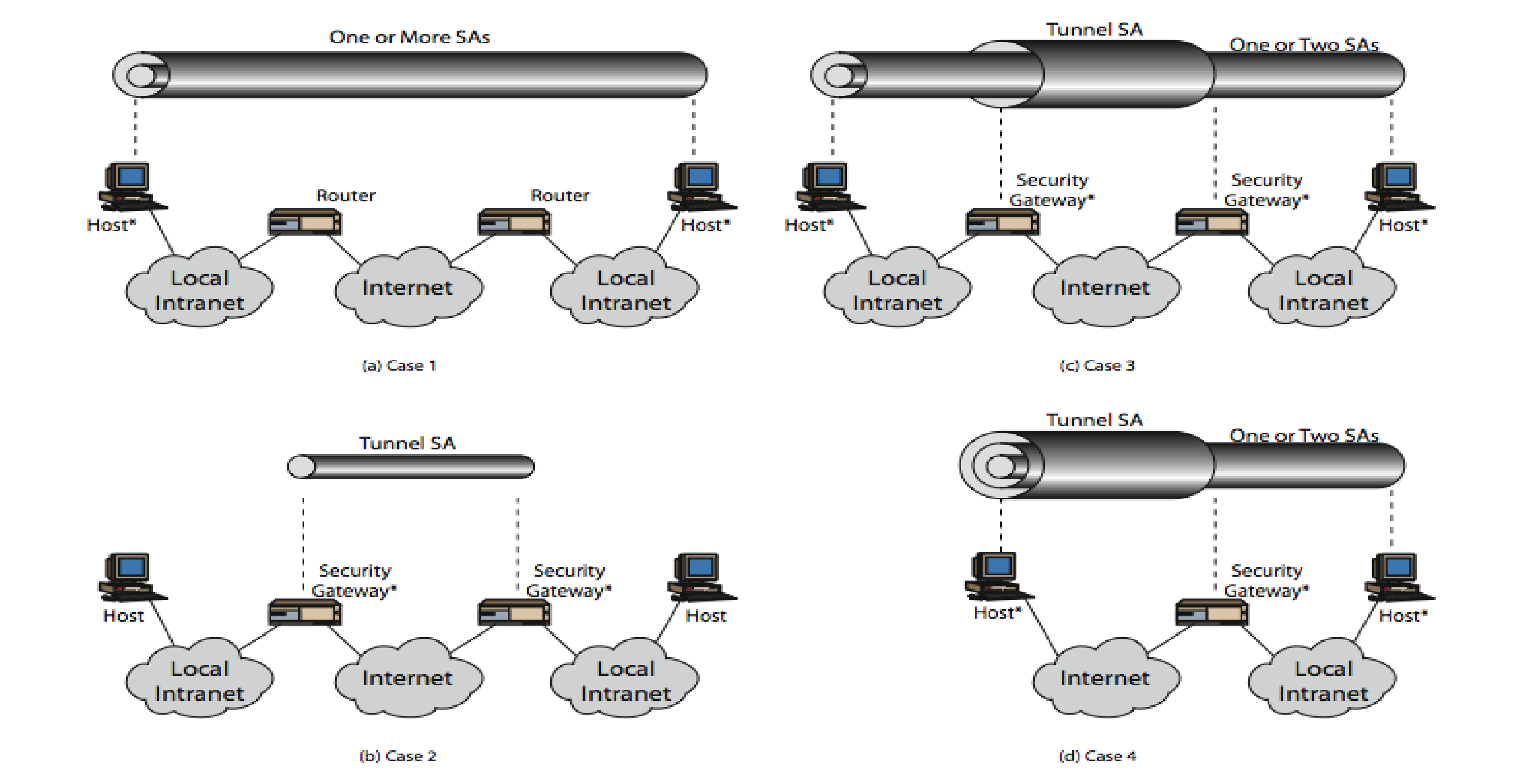
(b) Transport Mode

(c) Tunnel Mode

COMBINING SECURITY ASSOCIATIONS

- SA's can implement either AH or ESP
- to implement both, need to combine SA's
  - form a security association bundle
  - may terminate at different or same endpoints
  - combined by
    - transport adjacency
    - iterated tunneling
- combining authentication & encryption
  - ESP with authentication, bundled inner ESP & outer AH, bundled inner transport & outer ESP

COMBINING SECURITY ASSOCIATIONS



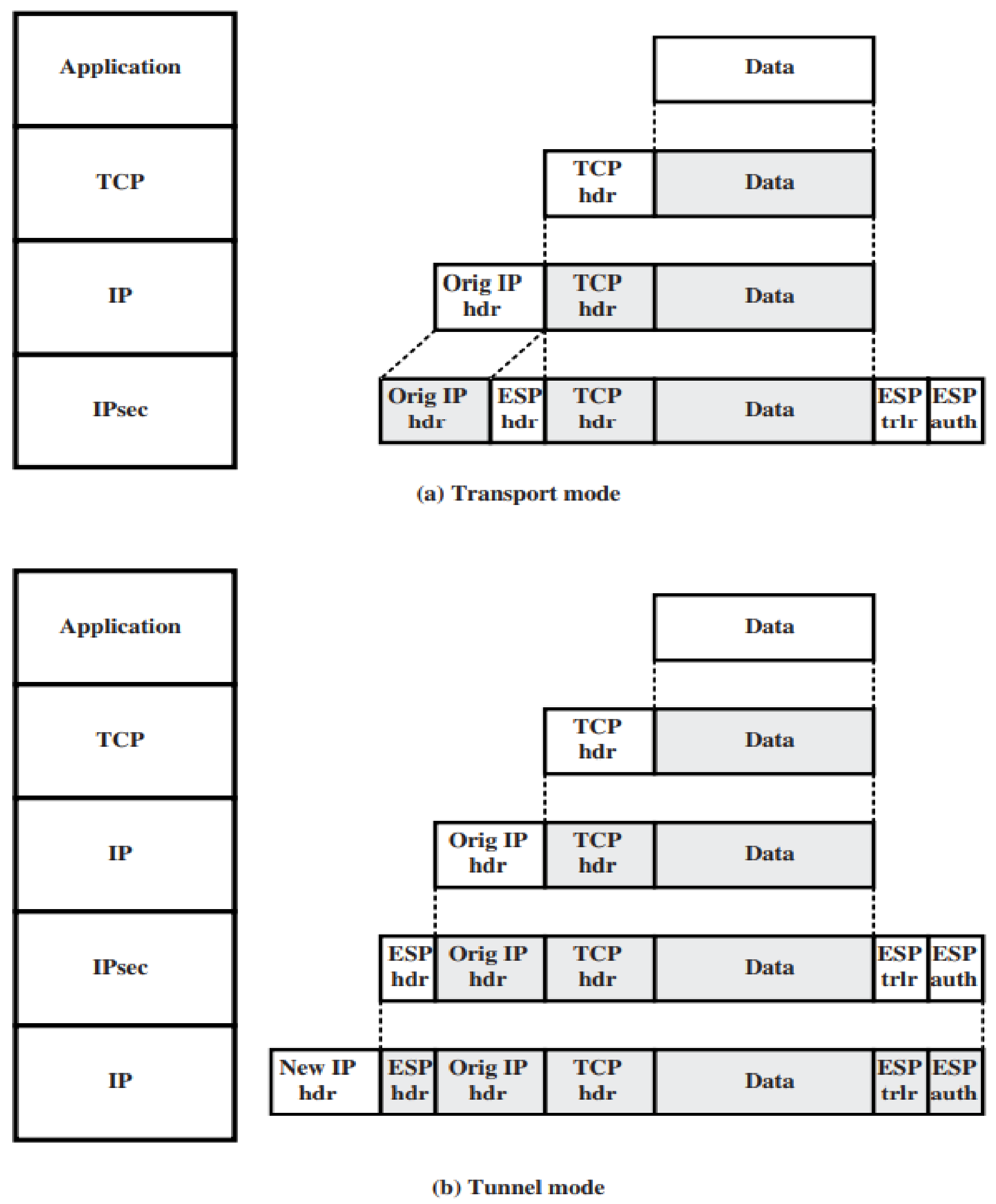
(a) Case 1

(c) Case 3

(b) Case 2

(d) Case 4

PROTOCOL OPERATION FOR ESP



(a) Transport mode

(b) Tunnel mode

INTERNET KEY EXCHANGE

- handles key generation & distribution
- typically need 2 pairs of keys
  - 2 per direction for AH & ESP
- manual key management
  - sysadmin manually configures every system
- automated key management
  - automated system for on demand creation of keys for SA’s in large systems
  - has Oakley & ISAKMP elements

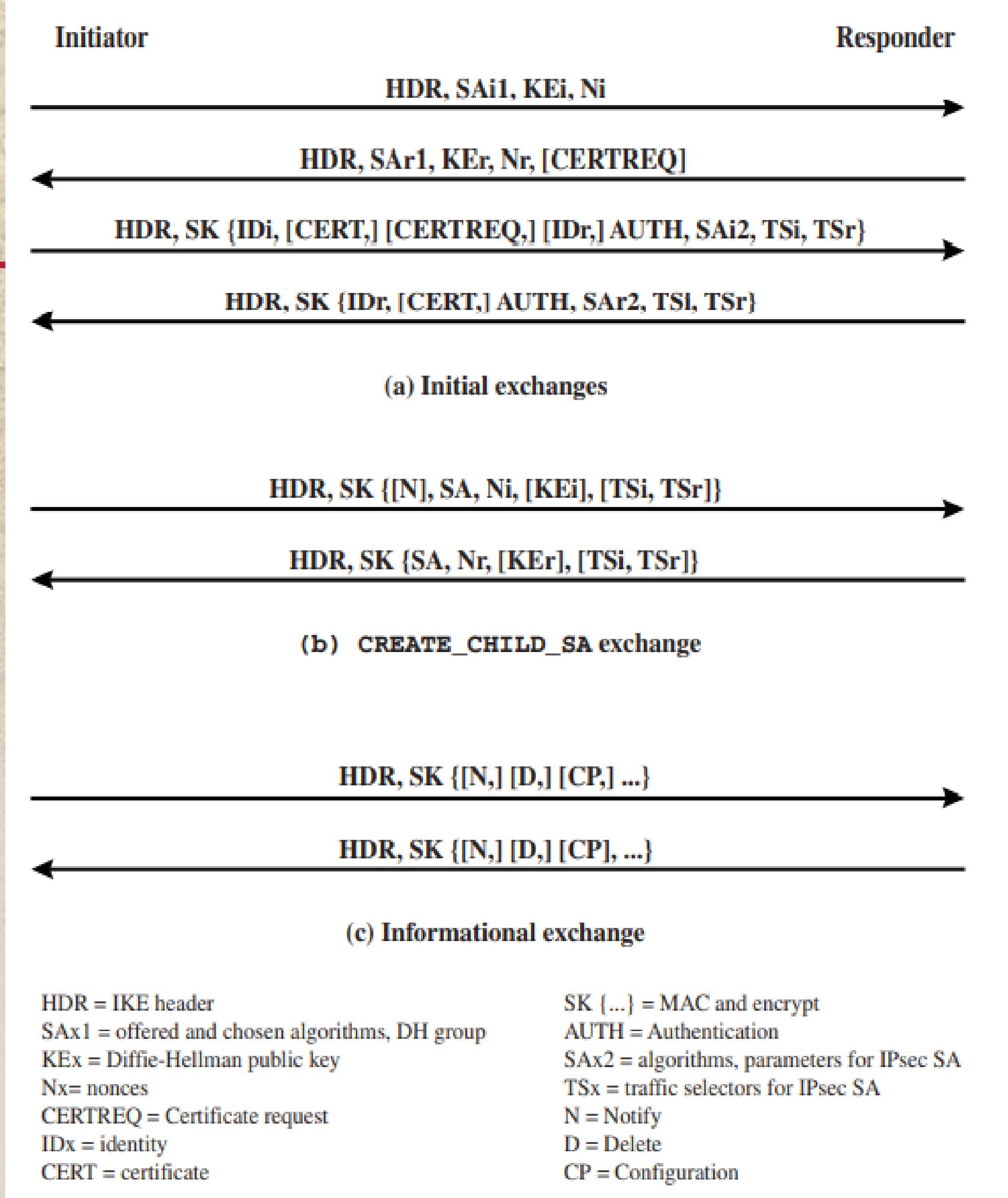
OAKLEY

- a key exchange protocol
- based on Diffie-Hellman key exchange
- adds features to address weaknesses
  - no info on parties, man-in-middle attack, cost
  - so adds cookies, groups (global params), nonces, DH key exchange with authentication
- can use arithmetic in prime fields or elliptic curve fields

ISAKMP

- Internet Security Association and Key Management Protocol
- provides framework for key management
- defines procedures and packet formats to establish, negotiate, modify, & delete SAs
- independent of key exchange protocol, encryption algorithm & authentication method
- IKEv2 no longer uses Oakley & ISAKMP terms, but basic functionality is same

IKEV2 EXCHANGES


Initiator
Responder
HDR, SAi1, KEi, Ni
HDR, SAr1, KEr, Nr, [CERTREQ]
HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr}
HDR, SK {IDr, [CERT,] AUTH, SAr2, TSi, TSr}

(a) Initial exchanges

HDR, SK {[N], SA, Ni, [KEi], [TSi, TSr]}
HDR, SK {SA, Nr, [KEr], [TSi, TSr]}

(b) CREATE_CHILD_SA exchange

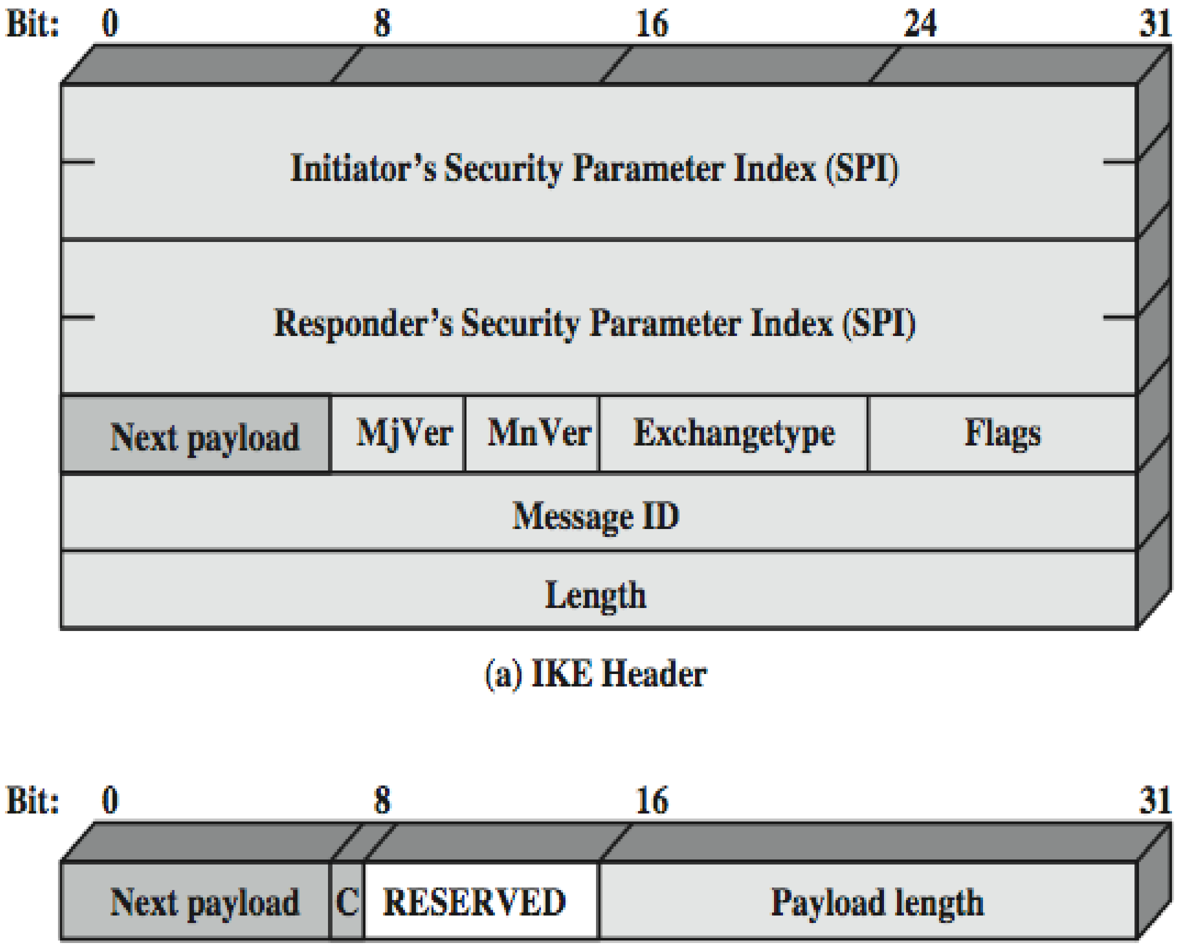
HDR, SK {[N,] [D,] [CP,] ...}
HDR, SK {[N,] [D,] [CP], ...}

(c) Informational exchange

HDR = IKE header
SAx1 = offered and chosen algorithms, DH group
KEx = Diffie-Hellman public key
Nx= nonces
CERTREQ = Certificate request
IDx = identity
CERT = certificate

SK {...} = MAC and encrypt
AUTH = Authentication
SAx2 = algorithms, parameters for IPsec SA
TSx = traffic selectors for IPsec SA
N = Notify
D = Delete
CP = Configuration

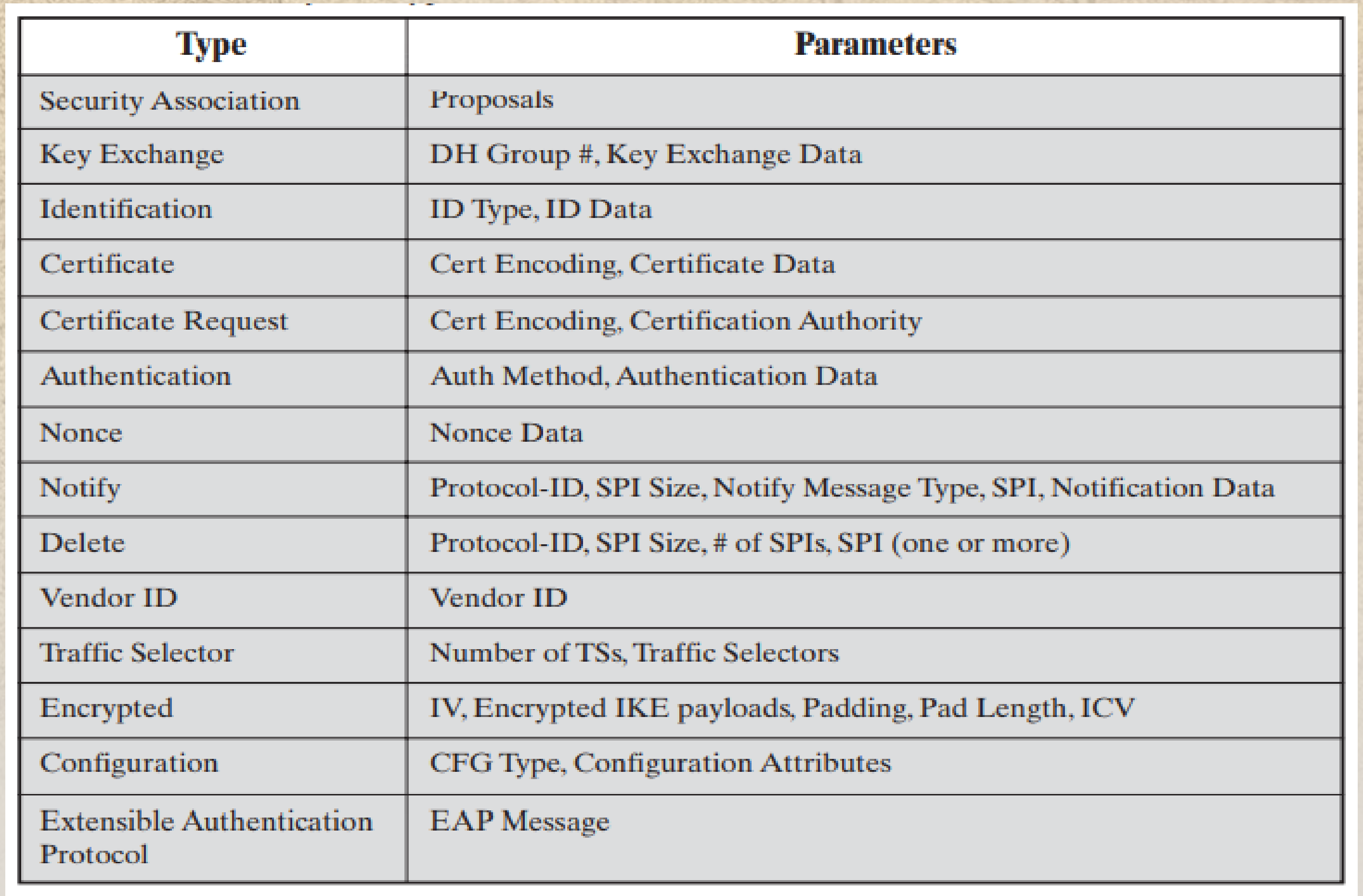
IKE FORMATS



(a) IKE Header

(b) Generic Payload Header

IKE PAYLOAD TYPES



Type	Parameters
Security Association	Proposals
Key Exchange	DH Group #, Key Exchange Data
Identification	ID Type, ID Data
Certificate	Cert Encoding, Certificate Data
Certificate Request	Cert Encoding, Certification Authority
Authentication	Auth Method, Authentication Data
Nonce	Nonce Data
Notify	Protocol-ID, SPI Size, Notify Message Type, SPI, Notification Data
Delete	Protocol-ID, SPI Size, # of SPIs, SPI (one or more)
Vendor ID	Vendor ID
Traffic Selector	Number of TSs, Traffic Selectors
Encrypted	IV, Encrypted IKE payloads, Padding, Pad Length, ICV
Configuration	CFG Type, Configuration Attributes
Extensible Authentication Protocol	EAP Message

IKE PAYLOADS & EXCHANGES

- have a number of ISAKMP payload types:
  - Security Association, Key Exchange, Identification, Certificate, Certificate Request, Authentication, Nonce, Notify, Delete, Vendor ID, Traffic Selector, Encrypted, Configuration, Extensible Authentication Protocol
- payload has complex hierarchical structure
- may contain multiple proposals, with multiple protocols & multiple transforms

CRYPTOGRAPHIC SUITES

- variety of cryptographic algorithm types
- to promote interoperability have
  - RFC4308 defines VPN cryptographic suites
    - VPN-A matches common corporate VPN security using 3DES & HMAC
    - VPN-B has stronger security for new VPNs implementing IPsecv3 and IKEv2 using AES
  - RFC4869 defines four cryptographic suites compatible with US NSA specs
    - provide choices for ESP & IKE
    - AES-GCM, AES-CBC, HMAC-SHA, ECP, ECDSA

SUMMARY

- have considered:
  - IPSec security framework
  - IPSec security policy
  - ESP
  - combining security associations
  - internet key exchange
  - cryptographic suites used

Thank
you