

## UNIT - V

# *Security at the Application Layer: PGP and S/MIME*



## Objectives

- ☐ To explain the general structure of an e-mail application program
- ☐ To discuss how PGP can provide security services for e-mail
- ☐ To discuss how S/MIME can provide security services for e-mail
- ☐ To define trust mechanism in both PGP and S/MIME
- ☐ To show the structure of messages exchanged in PGP and S/MIME

## 16-1 E-MAIL

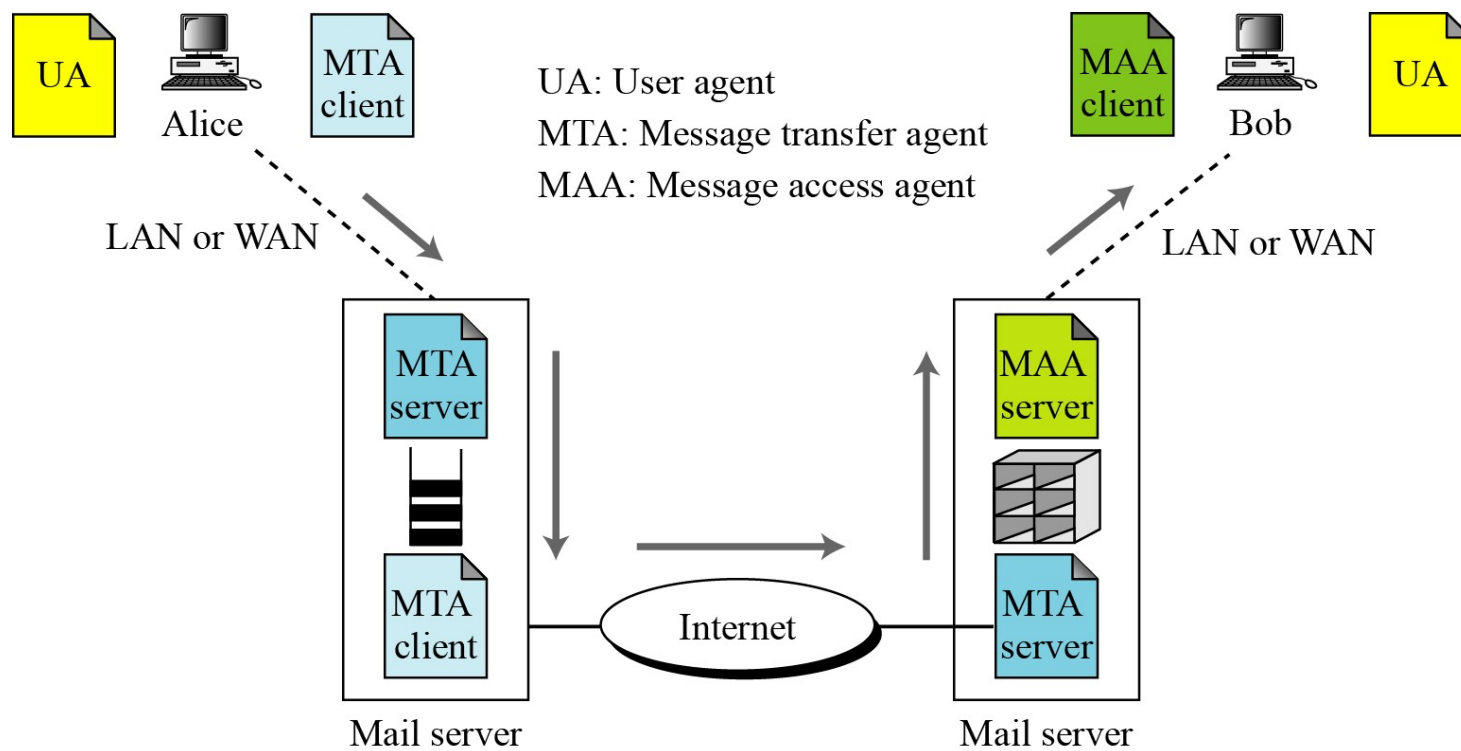
*Let us first discuss the electronic mail (e-mail) system in general.*

*Topics discussed in this section:*

16.1.1 E-mail Architecture

16.1.2 E-mail Security

Figure 16.1 E-mail architecture



## Cryptographic Algorithms

### Note

In e-mail security, the sender of the message needs to include the name or identifiers of the algorithms used in the message.

## Certificates

*It is obvious that some public-key algorithms must be used for e-mail security.*

## *Cryptographic Secrets*

### *Note*

In e-mail security, the encryption/decryption is done using a symmetric-key algorithm, but the secret key to decrypt the message is encrypted with the public key of the receiver and is sent with the message.

*Pretty Good Privacy (PGP) can be used to create a secure e-mail message or to store a file securely for future retrieval.*

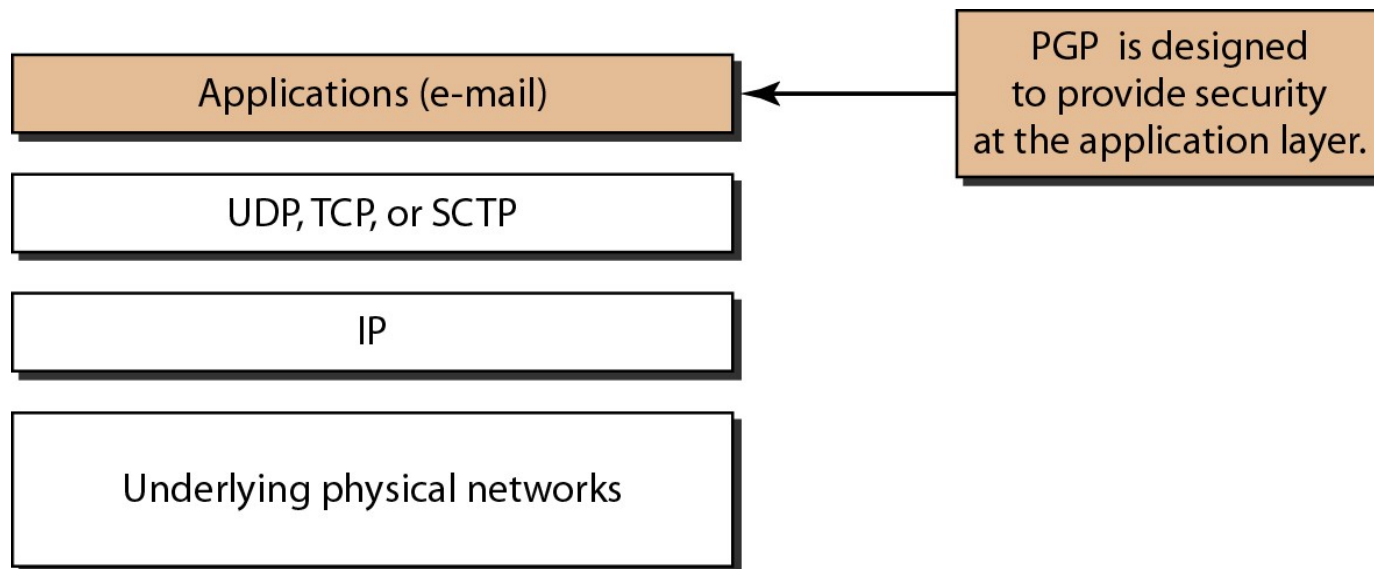
*One of the protocols to provide security at the application layer is Pretty Good Privacy (PGP).*

*PGP is designed to create authenticated and confidential e-mails.*

---

**Figure 32.19** *Position of PGP in the TCP/IP protocol suite*

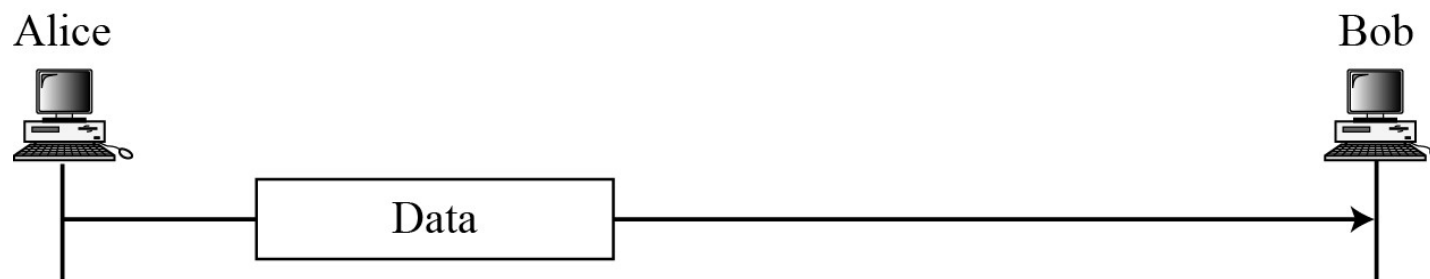
---





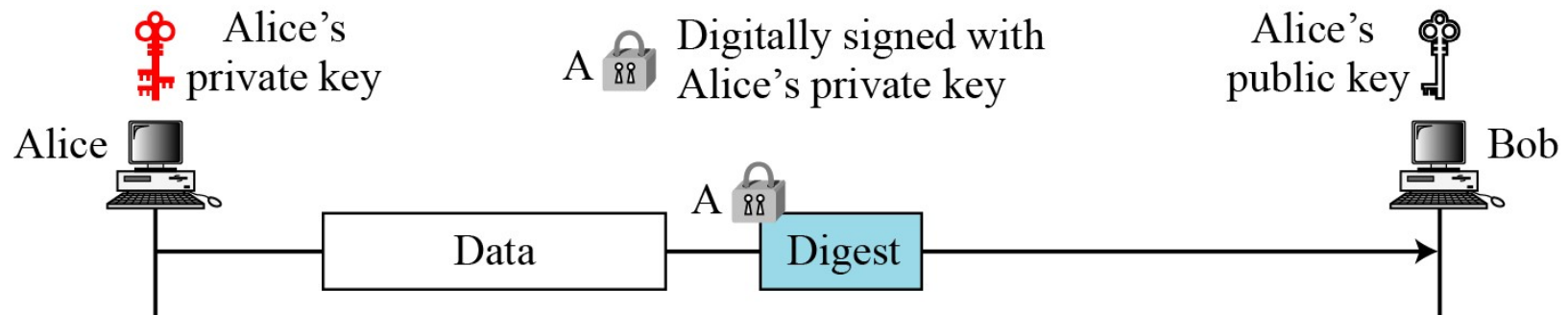
## *Plaintext*

Figure 16.2 *A plaintext message*



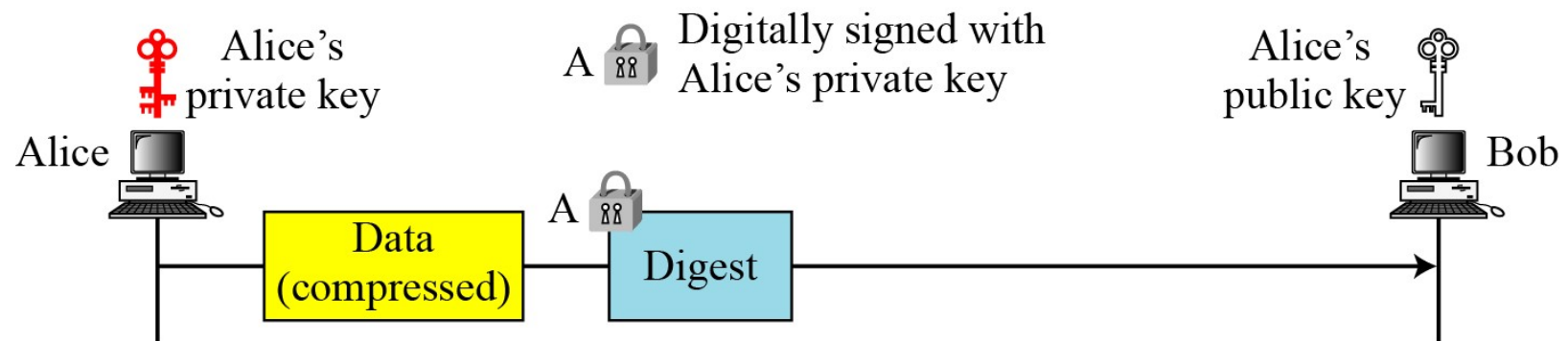
## Message Integrity

Figure 16.3 *An authenticated message*

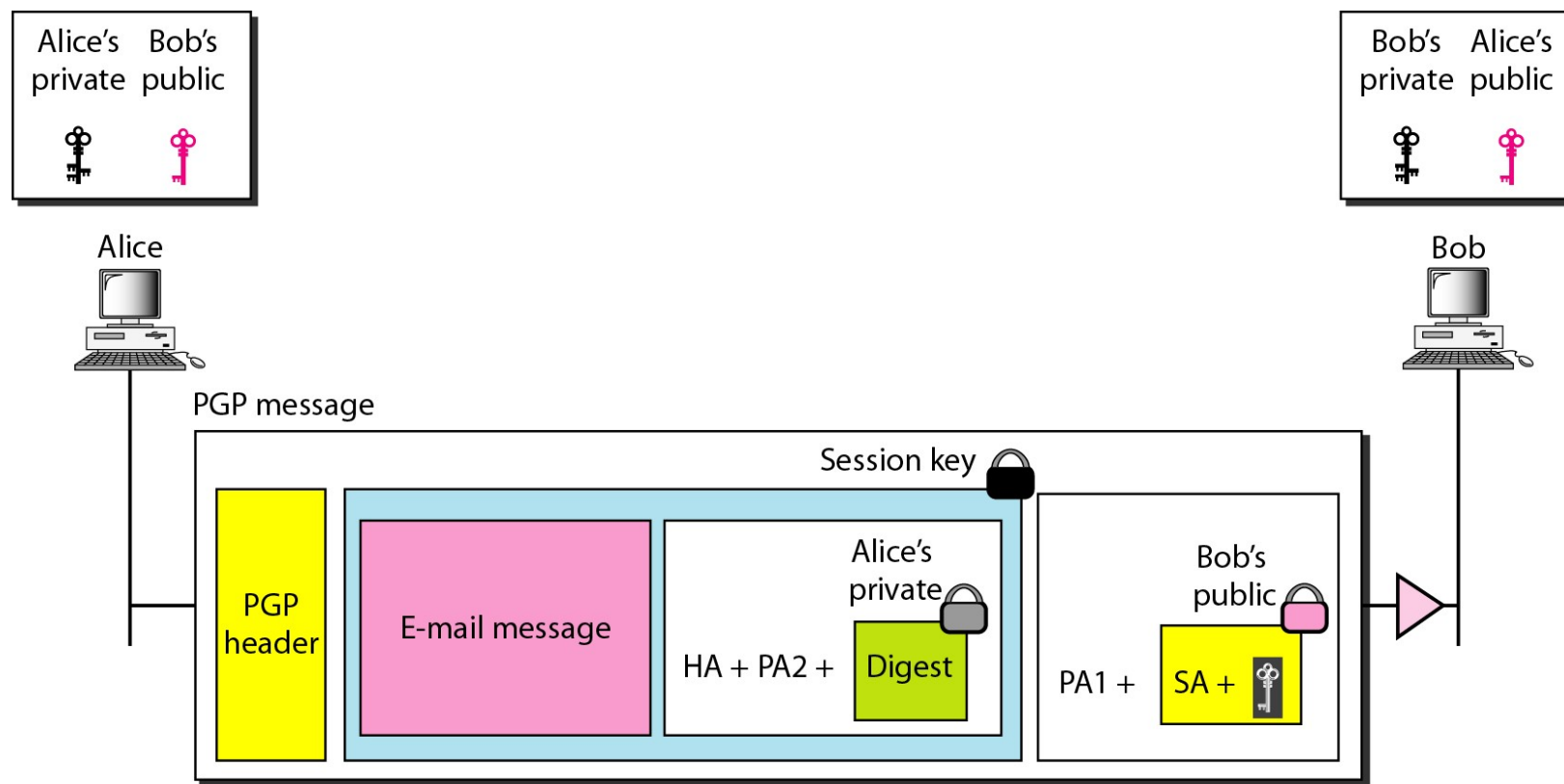


## Compression

Figure 16.4 *A compressed message*



**Figure 32.20** *A scenario in which an e-mail message is authenticated and encrypted*



PA1: Public-key algorithm 1 (for encrypting session key)

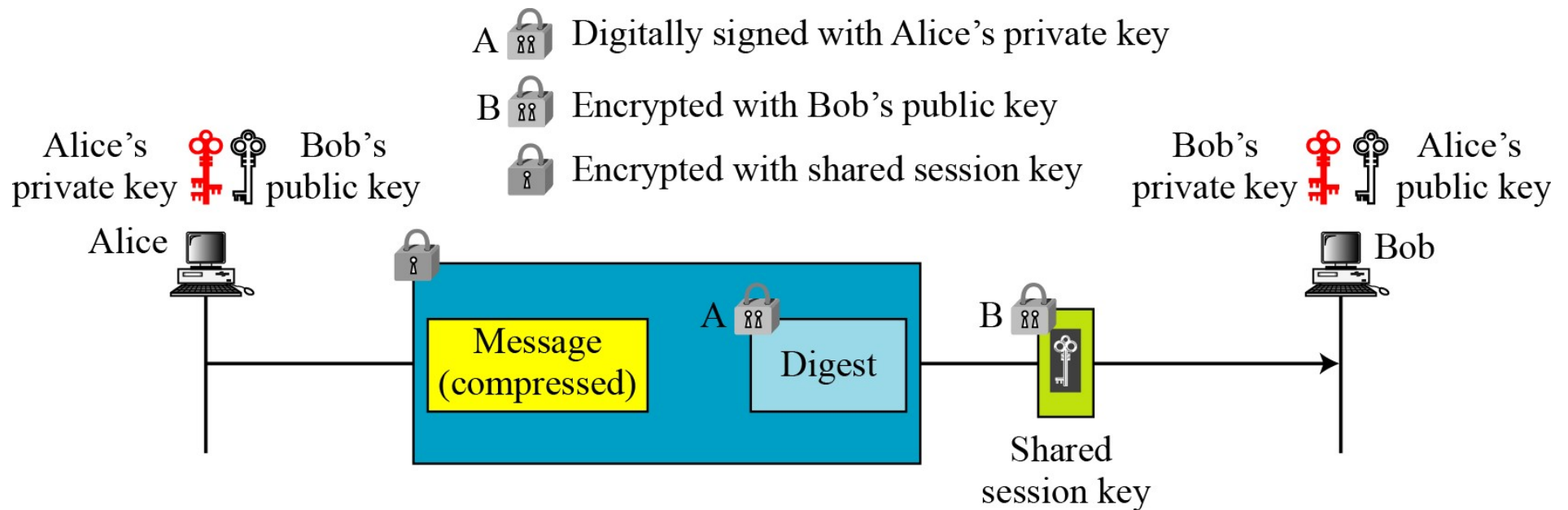
PA2: Public-key algorithm (for encrypting the digest)

SA: Symmetric-key algorithm identification (for encrypting message and digest)

HA: Hash algorithm identification (for creating digest)

# Confidentiality with One-Time Session Key

Figure 16.5 A confidential message



## *Code Conversion*

*Another service provided by PGP is code conversion.  
PGP uses Radix-64 conversion.*

## *Segmentation*

*PGP allows segmentation of the message.*



## PGP Algorithms

**Table 16.1** *Public-key algorithms*

<i>ID</i>	<i>Description</i>
1	RSA (encryption or signing)
2	RSA (for encryption only)
3	RSA (for signing only)
16	ElGamal (encryption only)
17	DSS
18	Reserved for elliptic curve
19	Reserved for ECDSA
20	ElGamal (for encryption or signing)
21	Reserved for Diffie-Hellman
100–110	Private algorithms

**Table 16.2** *Symmetric-key algorithms*

<i>ID</i>	<i>Description</i>
0	No Encryption
1	IDEA
2	Triple DES
3	CAST-128
4	Blowfish
5	SAFER-SK128
6	Reserved for DES/SK
7	Reserved for AES-128
8	Reserved for AES-192
9	Reserved for AES-256
100–110	Private algorithms



**Table 16.3** *Hash Algorithms*

<i>ID</i>	<i>Description</i>
1	MD5
2	SHA-1
3	RIPE-MD/160
4	Reserved for double-width SHA
5	MD2
6	TIGER/192
7	Reserved for HAVAL
100–110	Private algorithms

**Table 16.4** *Compression methods*

<i>ID</i>	<i>Description</i>
0	Uncompressed
1	ZIP
2	ZLIP
100–110	Private methods

## *X.509 Certificates*

*Protocols that use X.509 certificates depend on the hierarchical structure of the trust.*

### *Note*

In X.509, there is a single path from the fully trusted authority to any certificate.

## *PGP Certificates*

*In PGP, there is no need for CAs; anyone in the ring can sign a certificate for anyone else in the ring.*

### *Note*

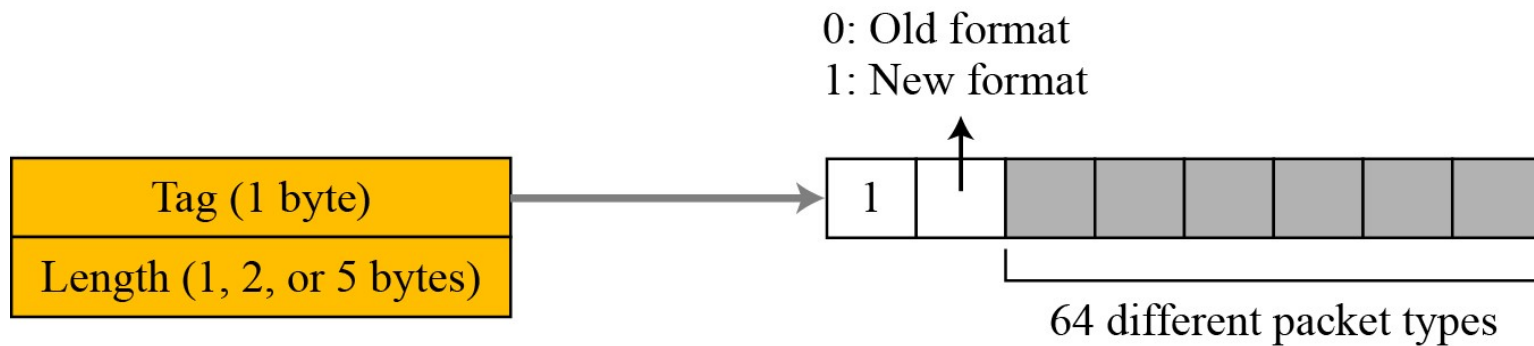
In PGP, there can be multiple paths from fully or partially trusted authorities to any subject.

## *Trusts and Legitimacy*

*The entire operation of PGP is based on **introducer trust**, the **certificate trust**, and the **legitimacy of the public keys**.*

*It may become necessary for an entity to revoke his or her public key from the ring. This may happen if the owner of the key feels that the key is compromised (stolen, for example) or just too old to be safe.*

Figure 16.12 *Format of packet header*



**Table 16.12** *Some commonly used packet types*

<i>Value</i>	<i>Packet type</i>
1	Session key packet encrypted using a public key
2	Signature packet
5	Private-key packet
6	Public-key packet
8	Compressed data packet
9	Data packet encrypted with a secret key
11	Literal data packet
13	User ID packet

**Figure 16.13** *Literal data packet*

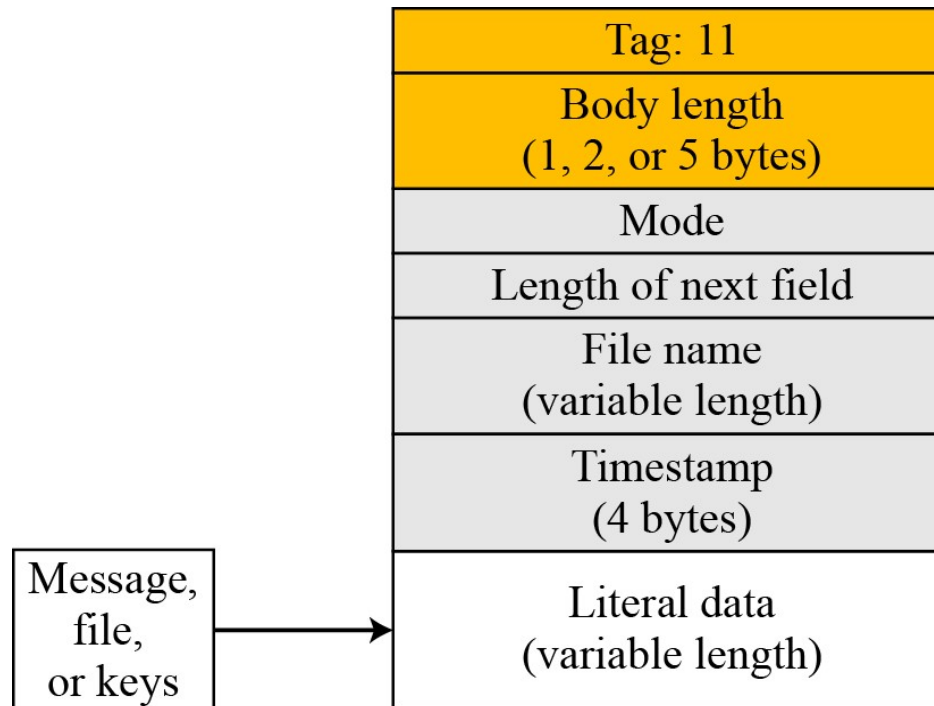




Figure 16.14 *Compressed data packet*

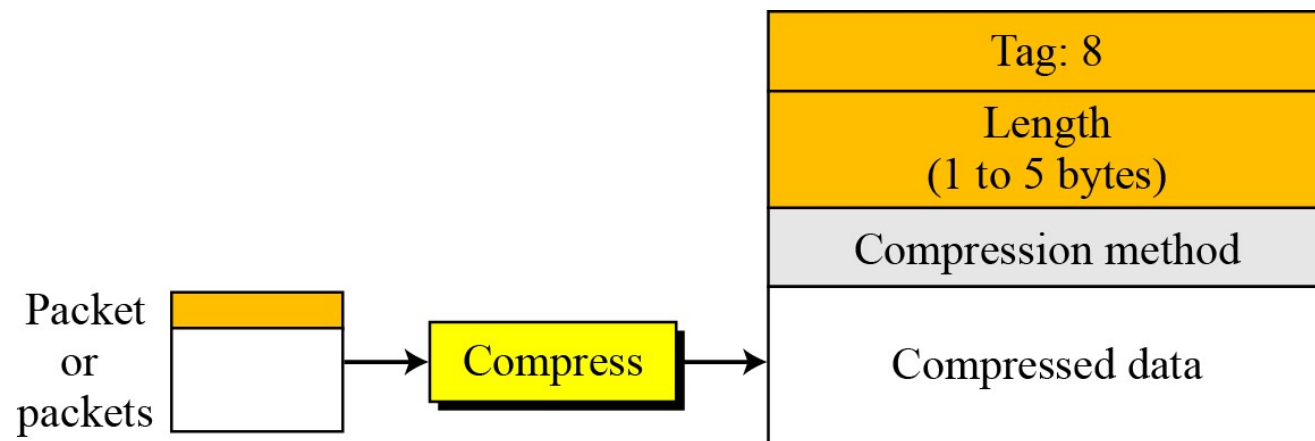


Figure 16.15 Encrypted data packet

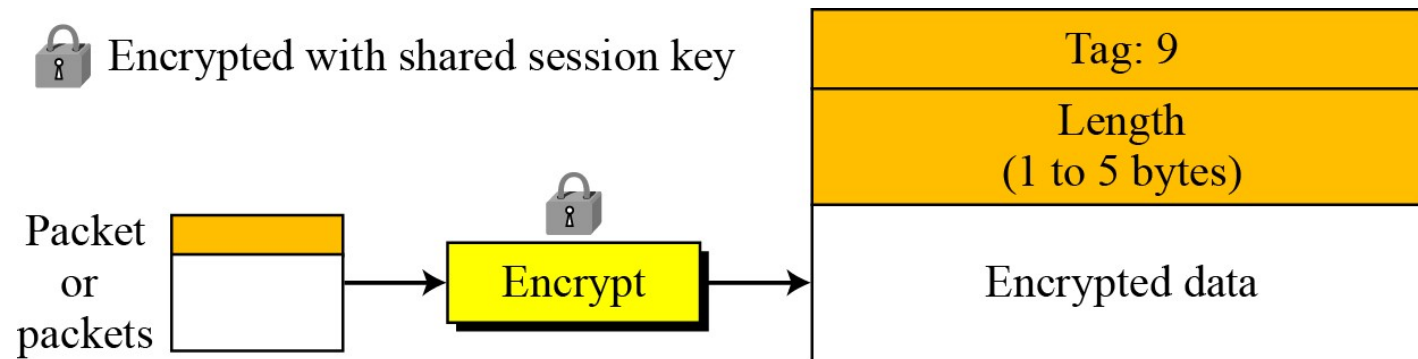

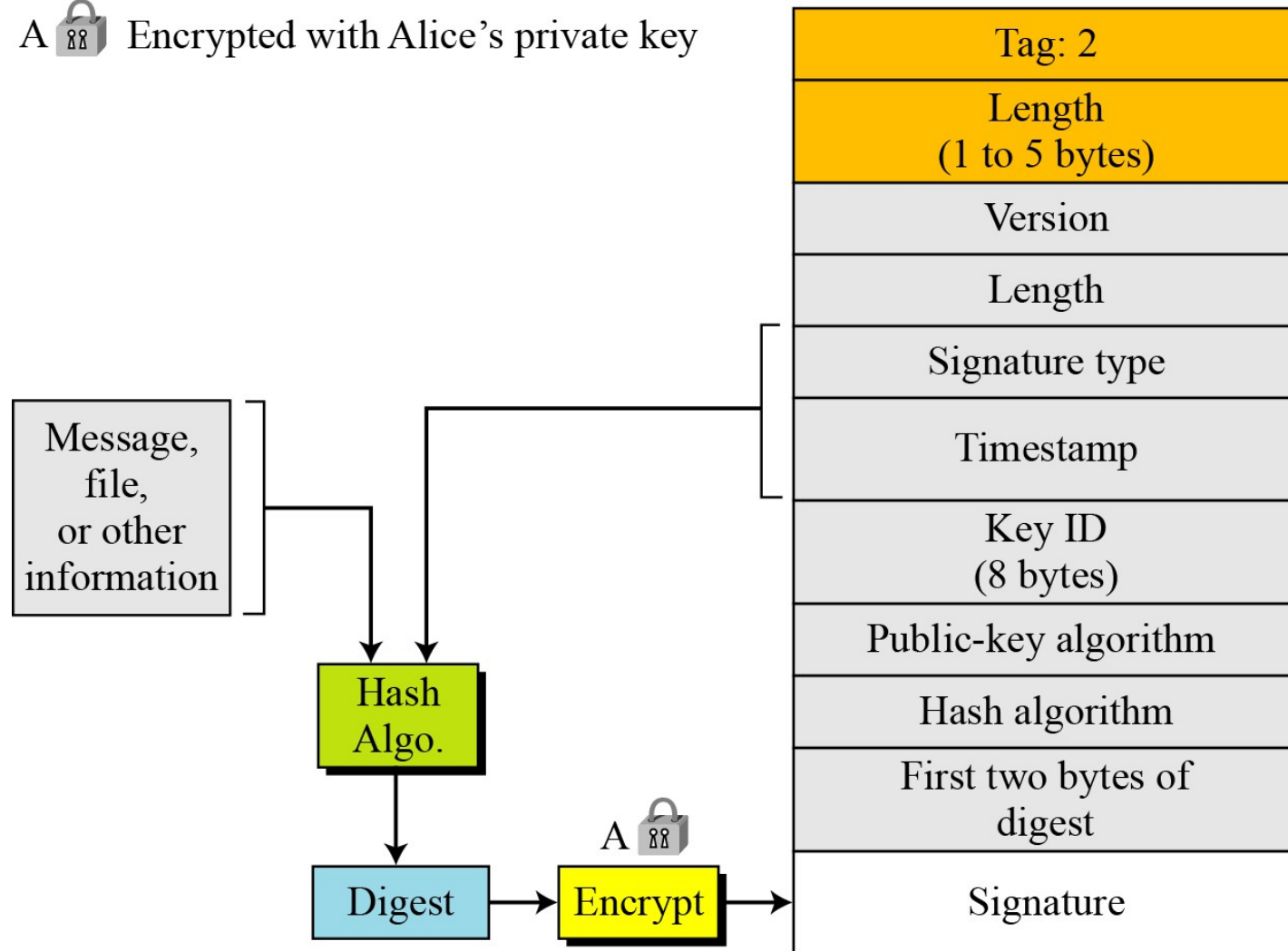


Figure 16.16 Signature packet

A  Encrypted with Alice's private key



**Table 16.13** *Some signature values*

<i>Value</i>	<i>Signature</i>
0x00	Signature of a binary document (message or file).
0x01	Signature of a text document (message or file).
0x10	Generic certificate of a user ID and public-key packet. The signer does not make any particular assertion about the owner of the key.
0x11	Personal certificate of a user ID and public-key packet. No verification is done on the owner of the key.
0x12	Casual certificate of a User ID and public-key packet. Some casual verification done on the owner of the key.
0x13	Positive certificate of a user ID and public-key packet. Substantial verification done.
0x30	Certificate revocation signature. This removes an earlier certificate (0x10 through 0x13).

Figure 16.17 Session-key packet

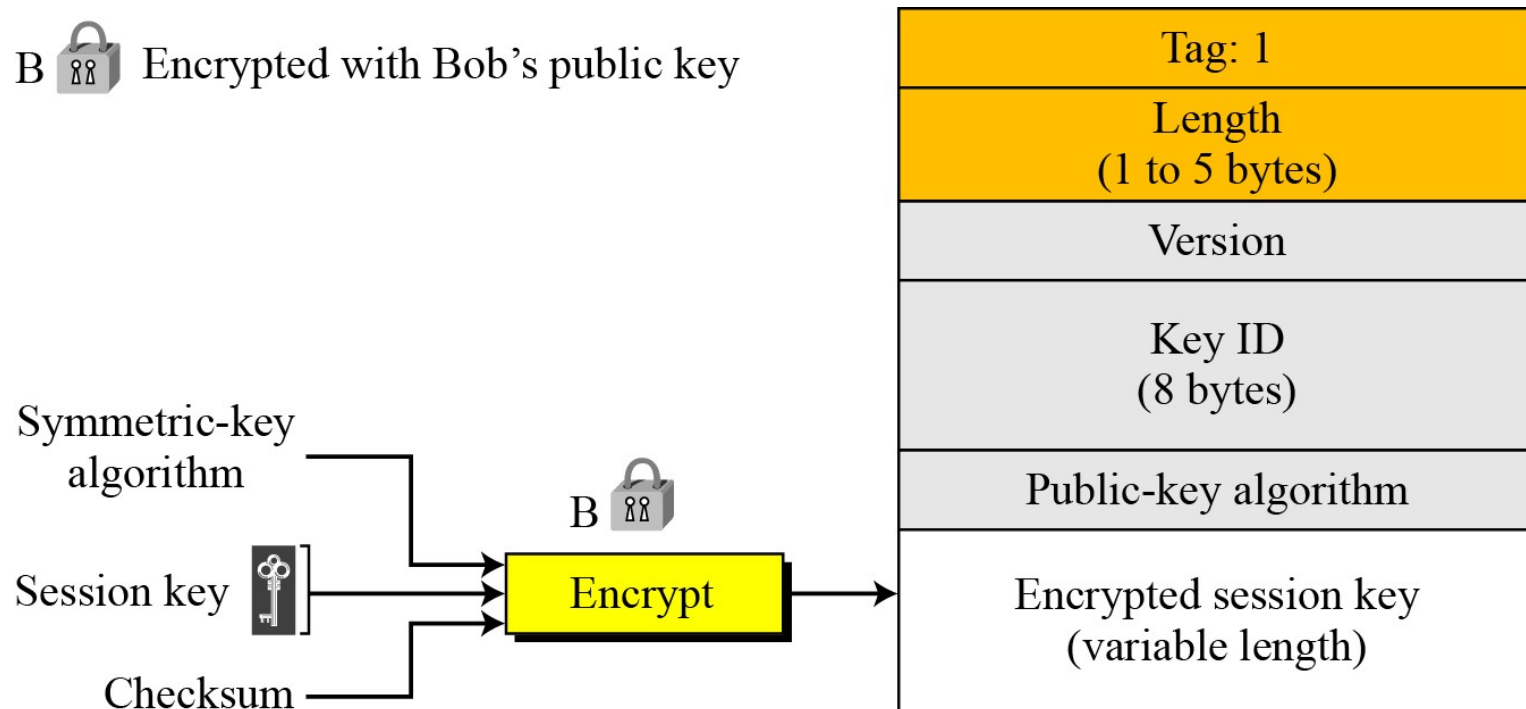


Figure 16.18 *Public-key packet*

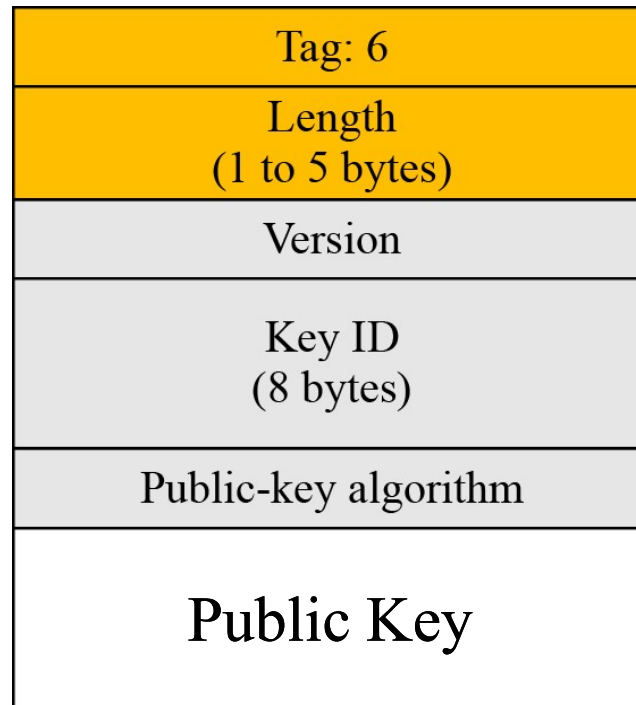


Figure 16.19 *User ID packet*

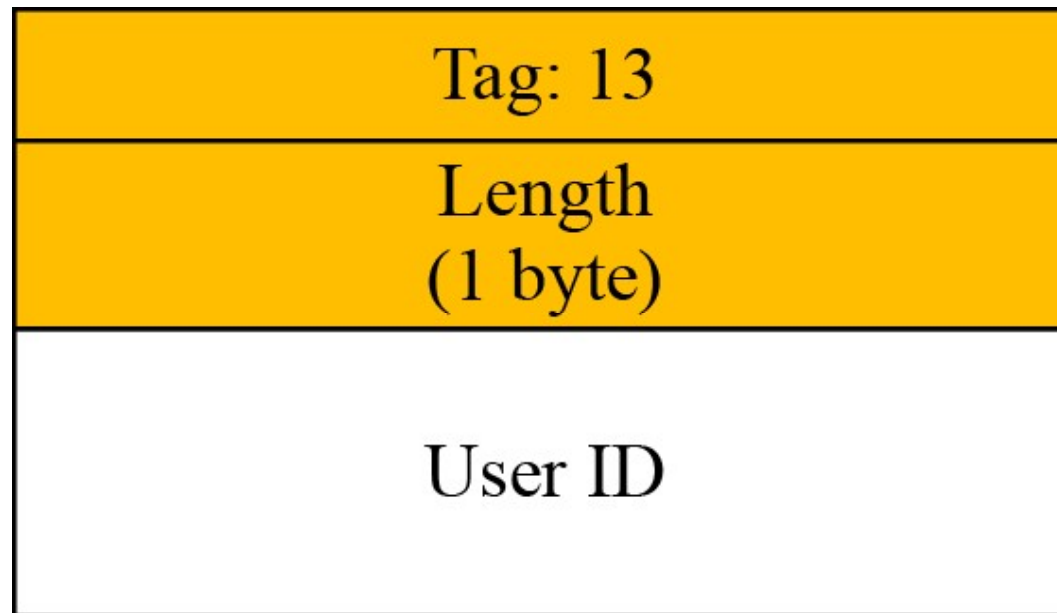


Figure 16.20 *Encrypted message*

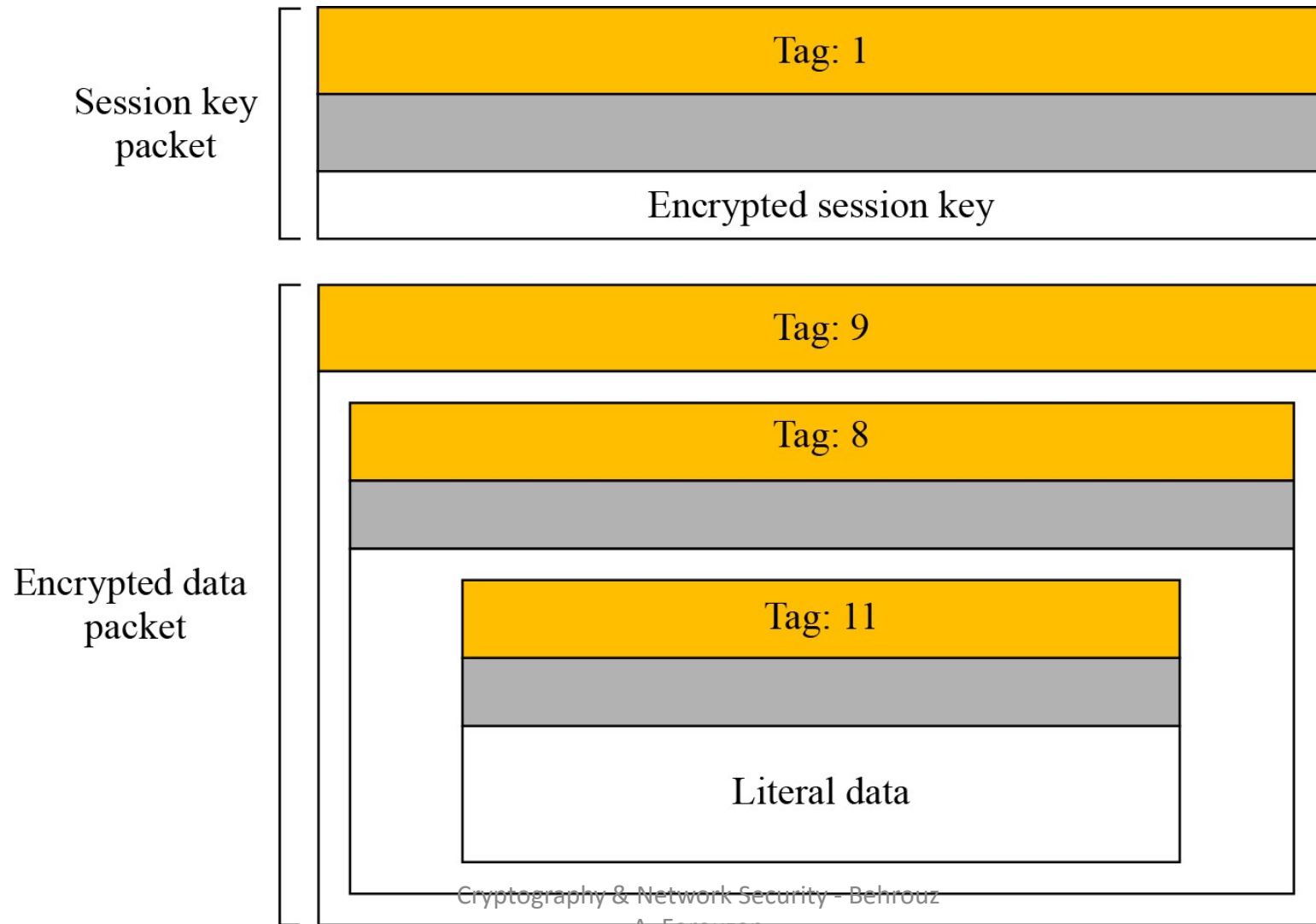




Figure 16.21 *Signed message*

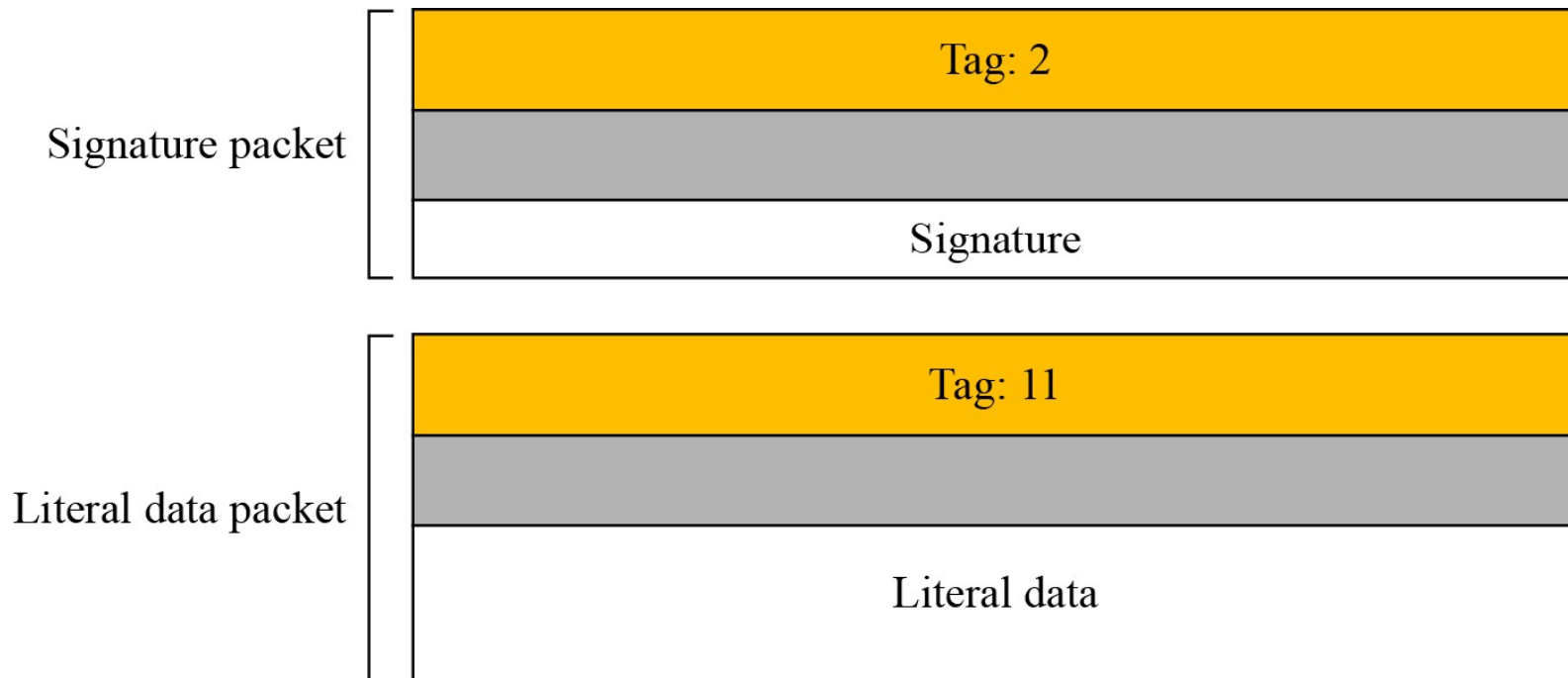
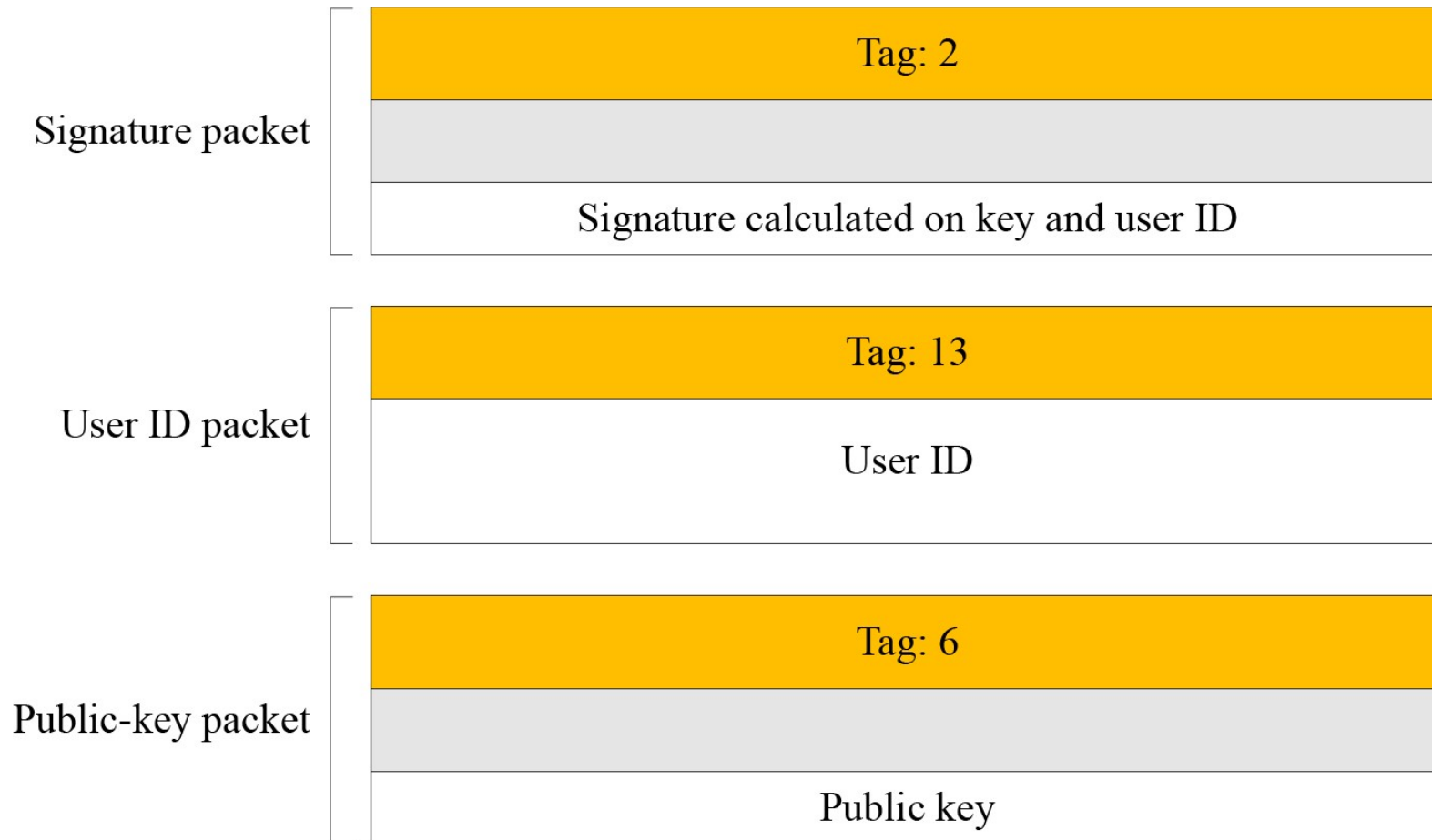


Figure 16.22 *Certificate message*



## 16-3 S/MIME

*Another security service designed for electronic mail is Secure/Multipurpose Internet Mail Extension (S/MIME). The protocol is an enhancement of the Multipurpose Internet Mail Extension (MIME) protocol.*

*Topics discussed in this section:*

16.3.1 MIME

16.3.2 S/MIME

16.3.3 Applications of S/MIME

Figure 16.23 *MIME*

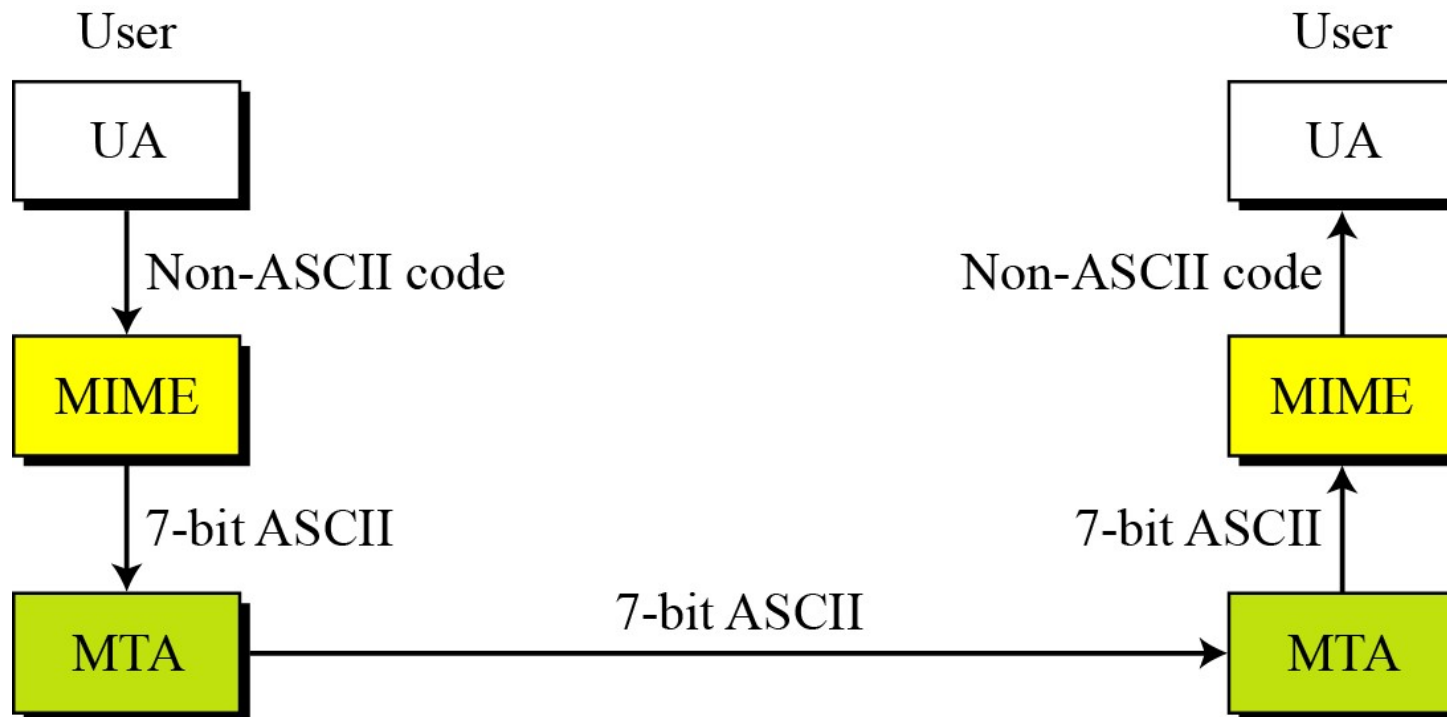
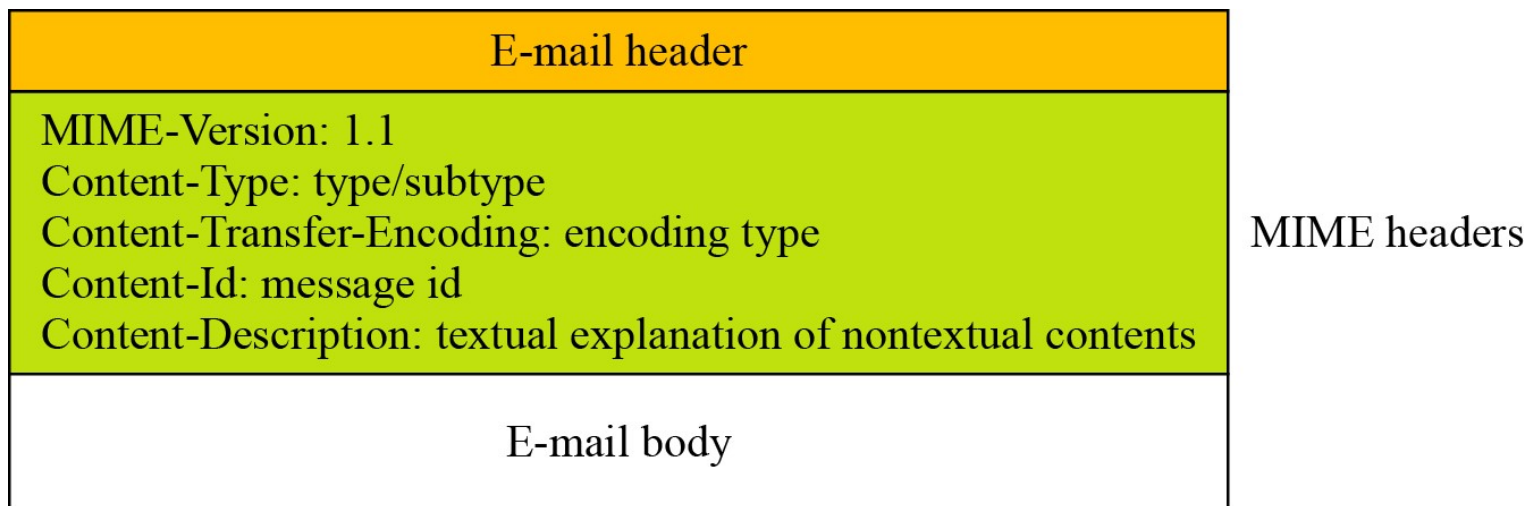


Figure 16.24 *Teledesic*



## *MIME-Version*

*This header defines the version of MIME used. The current version is 1.1.*

**MIME-Version: 1.1**

## *Content-Type*

*The content type and the content subtype are separated by a slash. Depending on the subtype, the header may contain other parameters.*

**Content-Type: <type / subtype; parameters>**



### 16.3.1 Continued

**Table 16.14** *Data types and subtypes in MIME*

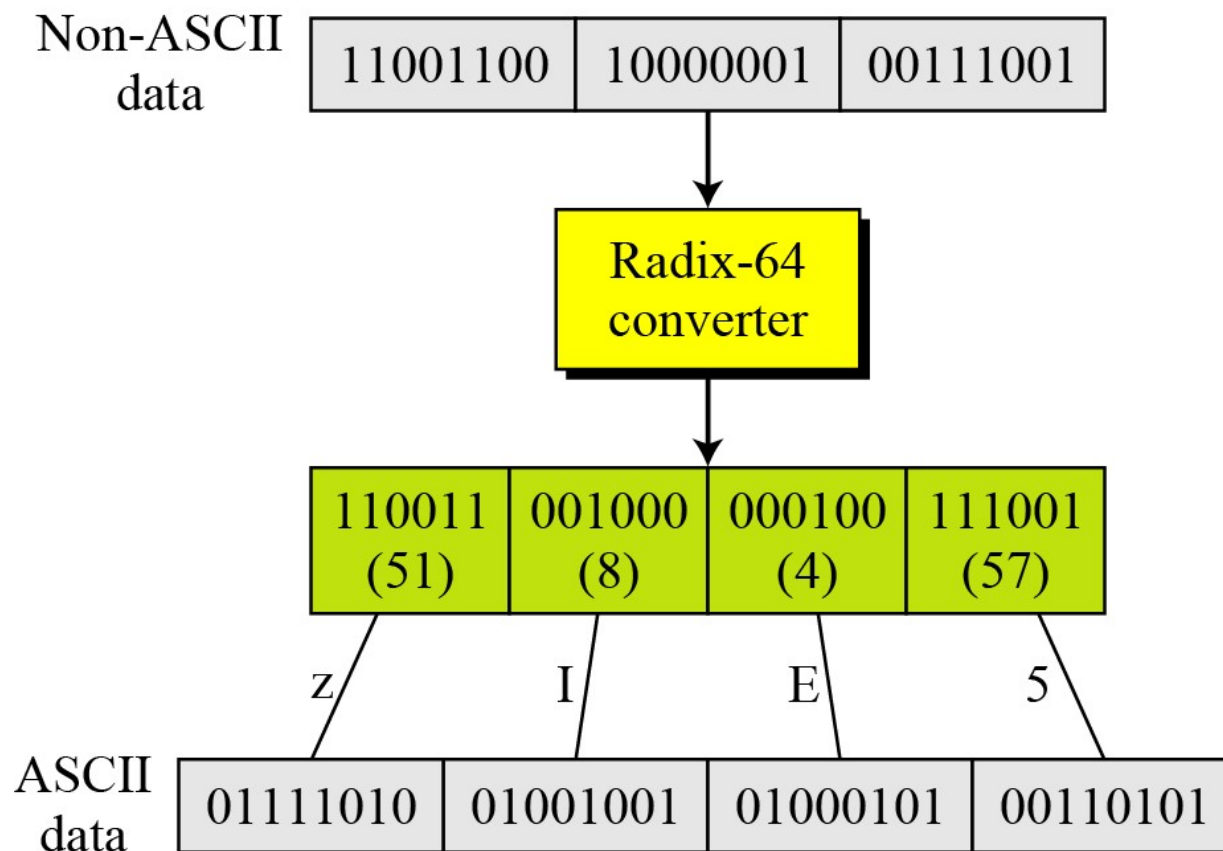
Type	Subtype	Description
	Plain	Unformatted.
	HTML	HTML format.
Multipart	Mixed	Body contains ordered parts of different data types.
	Parallel	Same as above, but no order.
	Digest	Similar to Mixed, but the default is message/RFC822.
	Alternative	Parts are different versions of the same message.
Message	RFC822	Body is an encapsulated message.
	Partial	Body is a fragment of a bigger message.
	External-Body	Body is a reference to another message.
Image	JPEG	Image is in JPEG format.
	GIF	Image is in GIF format.
Video	MPEG	Video is in MPEG format.
Audio	Basic	Single channel encoding of voice at 8 KHz.
Application	PostScript	Adobe PostScript.
	Octet-stream	General binary data (eight-bit bytes).

**Table 16.15** *Content-transfer-encoding*

<i>Type</i>	<i>Description</i>
7bit	NVT ASCII characters and short lines.
8bit	Non-ASCII characters and short lines.
Binary	Non-ASCII characters with unlimited-length lines.
Radix-64	6-bit blocks of data are encoded into 8-bit ASCII characters using Radix-64 conversion.
Quoted-printable	Non-ASCII characters are encoded as an equal sign followed by an ASCII code.



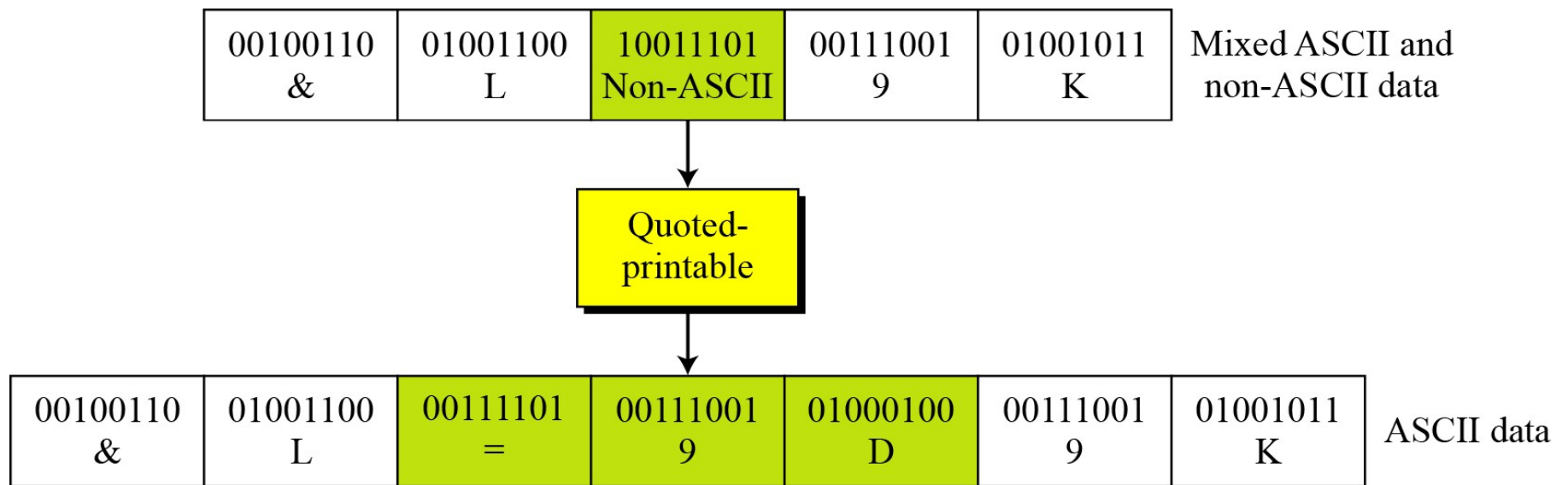
Figure 16.25 Radix-64 conversion



**Table 16.16** Radix-64 encoding table

<i>Value</i>	<i>Code</i>	<i>Value</i>	<i>Code</i>	<i>Value</i>	<i>Code</i>	<i>Value</i>	<i>Code</i>	<i>Value</i>	<i>Code</i>	<i>Value</i>	<i>Code</i>
0	<b>A</b>	11	<b>L</b>	22	<b>W</b>	33	<b>h</b>	44	<b>s</b>	55	<b>3</b>
1	<b>B</b>	12	<b>M</b>	23	<b>X</b>	34	<b>i</b>	45	<b>t</b>	56	<b>4</b>
2	<b>C</b>	13	<b>N</b>	24	<b>Y</b>	35	<b>j</b>	46	<b>u</b>	57	<b>5</b>
3	<b>D</b>	14	<b>O</b>	25	<b>Z</b>	36	<b>k</b>	47	<b>v</b>	58	<b>6</b>
4	<b>E</b>	15	<b>P</b>	26	<b>a</b>	37	<b>l</b>	48	<b>w</b>	59	<b>7</b>
5	<b>F</b>	16	<b>Q</b>	27	<b>b</b>	38	<b>m</b>	49	<b>x</b>	60	<b>8</b>
6	<b>G</b>	17	<b>R</b>	28	<b>c</b>	39	<b>n</b>	50	<b>y</b>	61	<b>9</b>
7	<b>H</b>	18	<b>S</b>	29	<b>d</b>	40	<b>o</b>	51	<b>z</b>	62	<b>+</b>
8	<b>I</b>	19	<b>T</b>	30	<b>e</b>	41	<b>p</b>	52	<b>0</b>	63	<b>/</b>
9	<b>J</b>	20	<b>U</b>	31	<b>f</b>	42	<b>q</b>	53	<b>1</b>		
10	<b>K</b>	21	<b>V</b>	32	<b>g</b>	43	<b>r</b>	54	<b>2</b>		

Figure 16.26 *Quoted-printable*



*S/MIME adds some new content types to include security services to the MIME. All of these new types include the parameter “application/pkcs7-mime,” in which “pkcs” defines “**Public Key Cryptography Specification.**”*

### *Cryptographic Message Syntax (CMS)*

*To define how security services, such as confidentiality or integrity, can be added to MIME content types, S/MIME has defined Cryptographic Message Syntax (CMS). The syntax in each case defines the exact encoding scheme for each content type.*

Figure 16.27 Signed-data content type

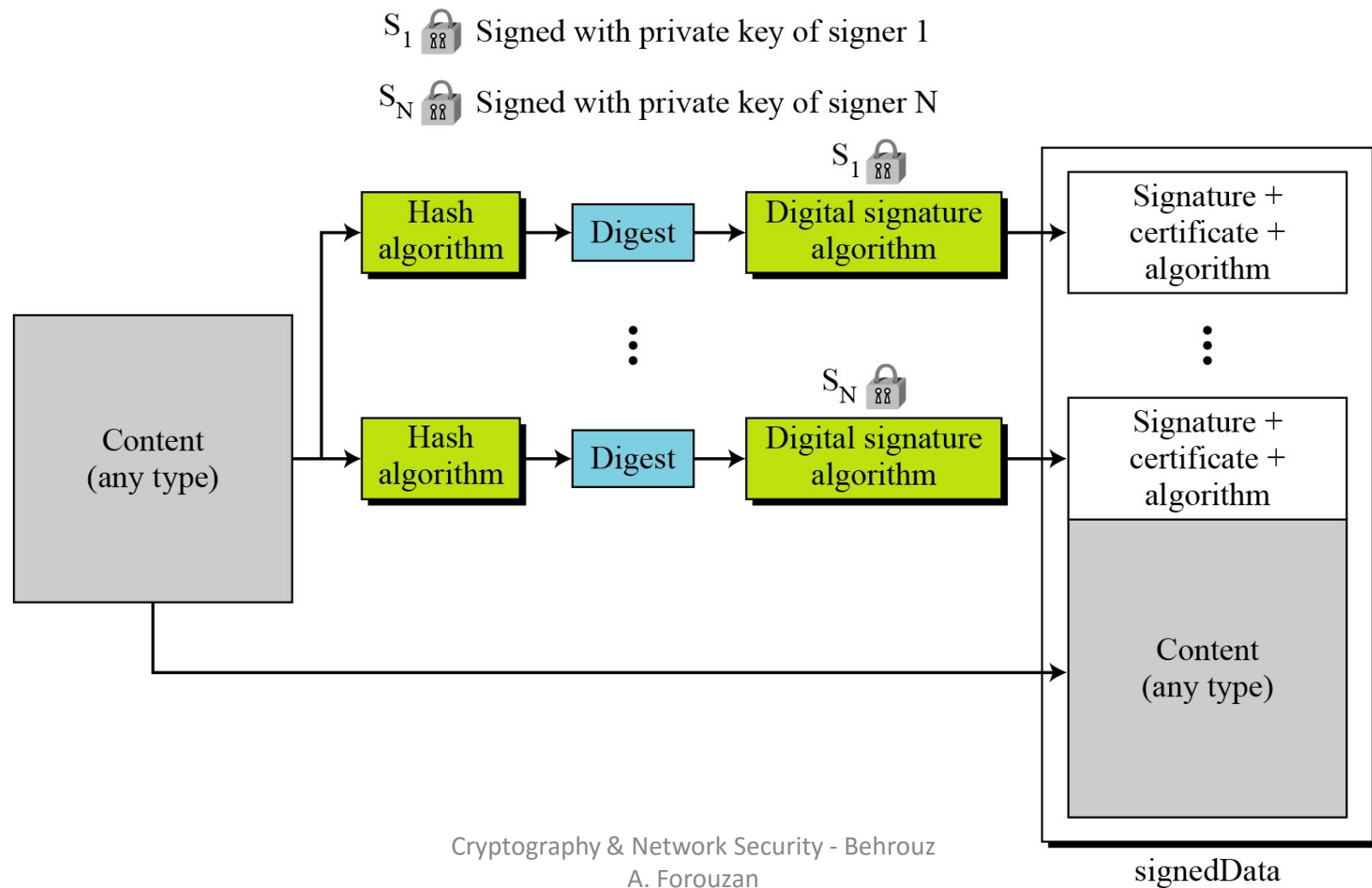


Figure 16.28 Enveloped-data content type

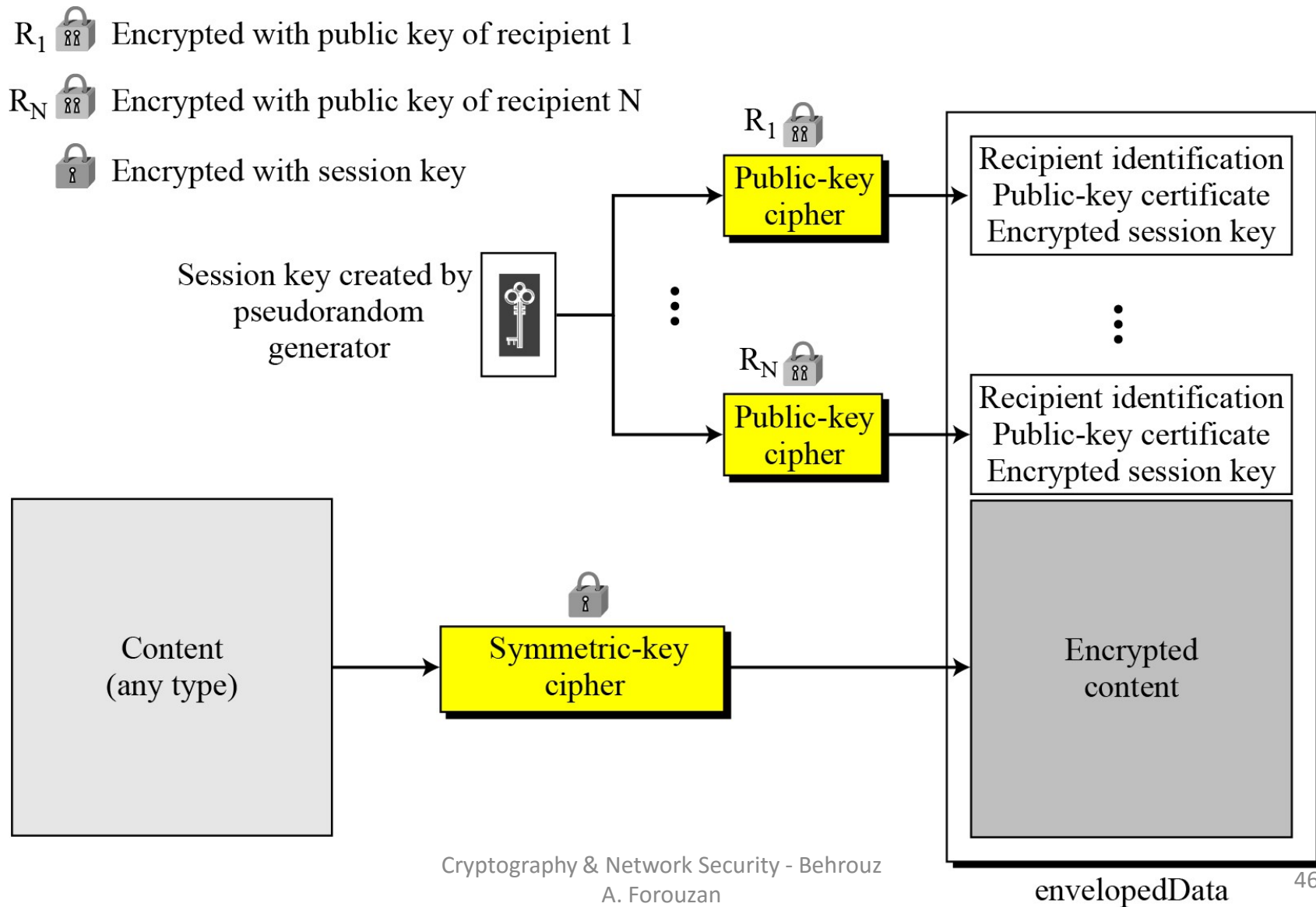
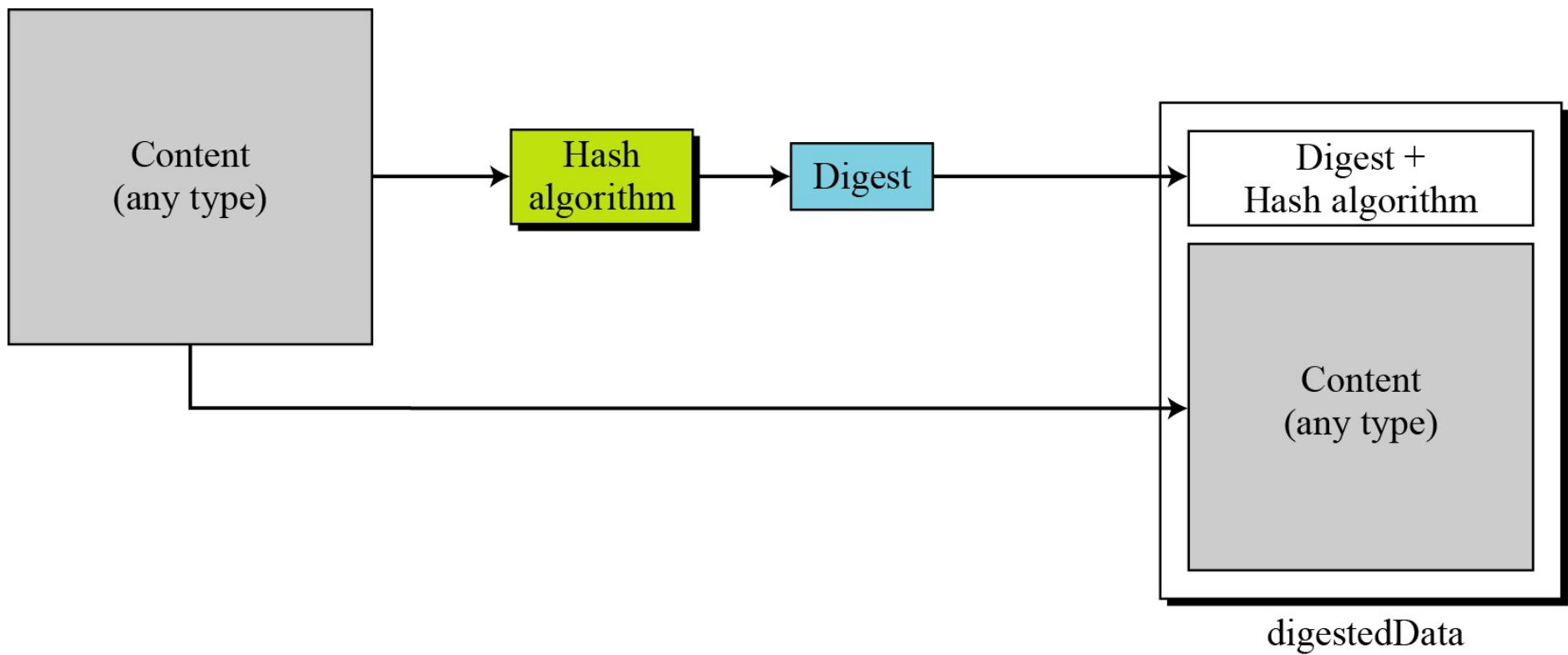




Figure 16.29 *Digest-data content type*

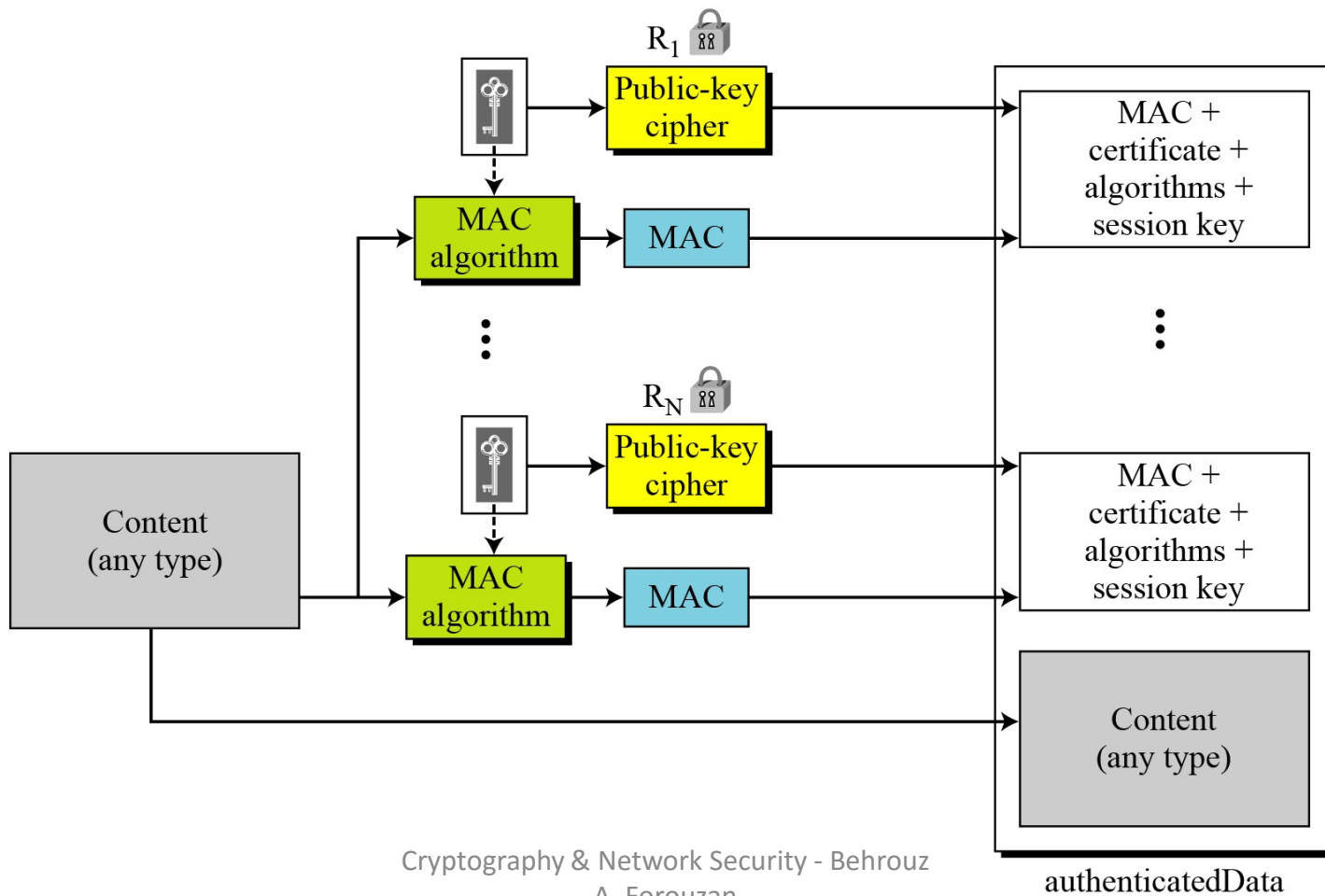


### 16.3.2 Continued

Figure 16.30 *Authenticated-data content type*

$R_1$   Encrypted with public key of recipient 1

$R_N$   Encrypted with public key of recipient N





## Cryptographic Algorithms

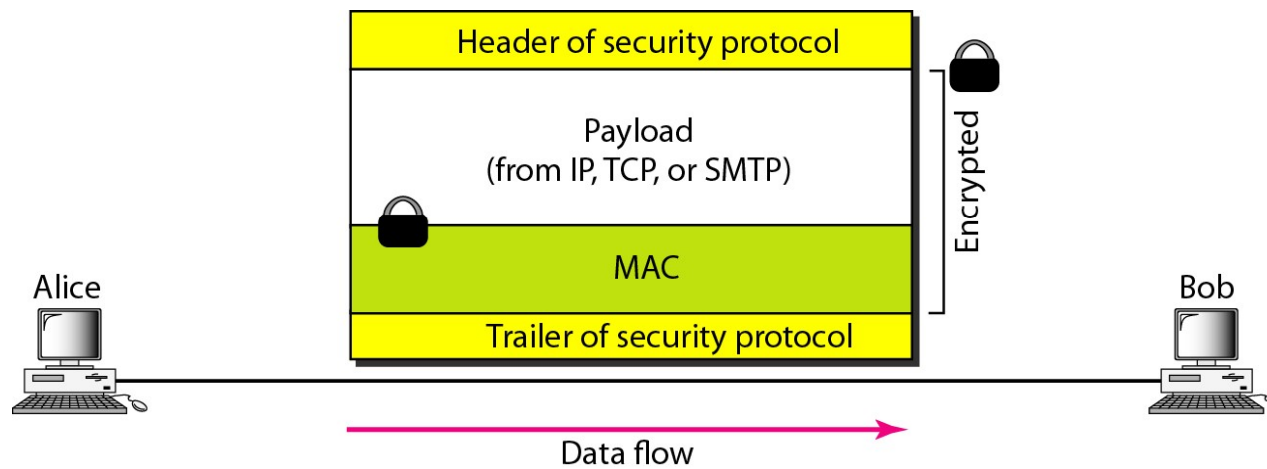
*S/MIME defines several cryptographic algorithms. The term “**must**” means an absolute requirement; the term “**should**” means recommendation.*

**Table 16.17** Cryptographic algorithm for S/MIME

Algorithm	Sender <i>must support</i>	Receiver <i>must support</i>	Sender <i>should support</i>	Receiver <i>should support</i>
Content-encryption algorithm	Triple DES	Triple DES		1. AES 2. RC2/40
Session-key encryption algorithm	RSA	RSA	Diffie-Hellman	Diffie-Hellman
Hash algorithm	SHA-1	SHA-1		MD5
Digest-encryption algorithm	DSS	DSS	RSA	RSA
Message-authentication algorithm		HMAC with SHA-1		

# Security in the Internet: IPSec, SSL/TLS, PGP, VPN, and Firewalls

Figure 32.1 *Common structure of three security protocols*



## 32-1 IPSecurity (IPSec)

*IPSecurity (IPSec) is a collection of protocols designed by the Internet Engineering Task Force (IETF) to provide security for a packet at the network level.*

### Topics discussed in this section:

Two Modes

Two Security Protocols

Security Association

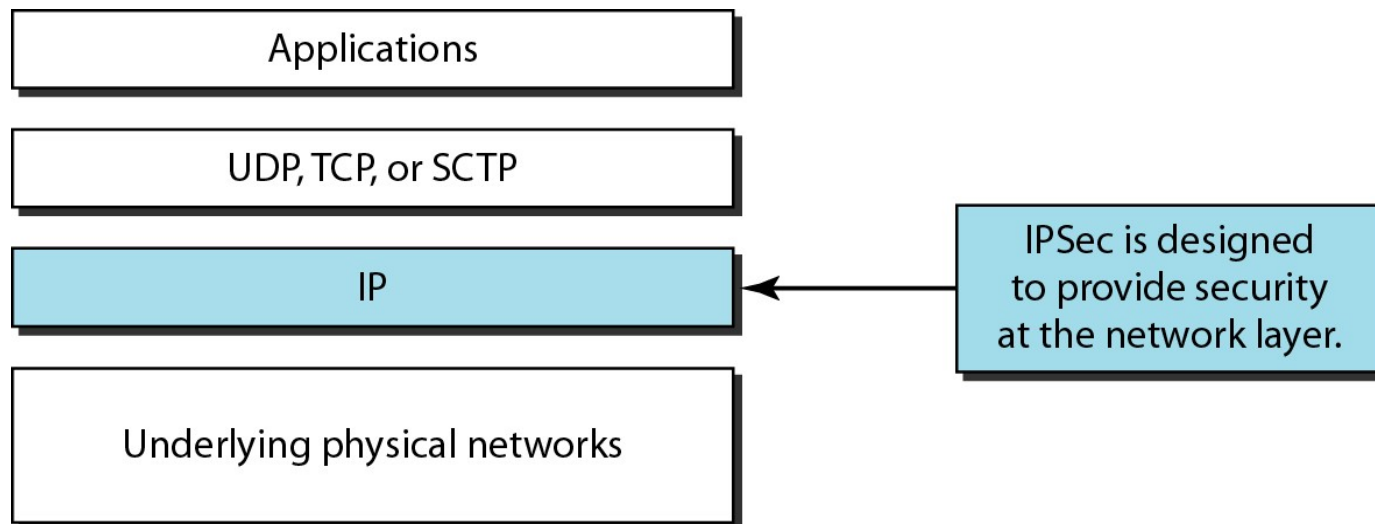
Internet Key Exchange (IKE)

Virtual Private Network

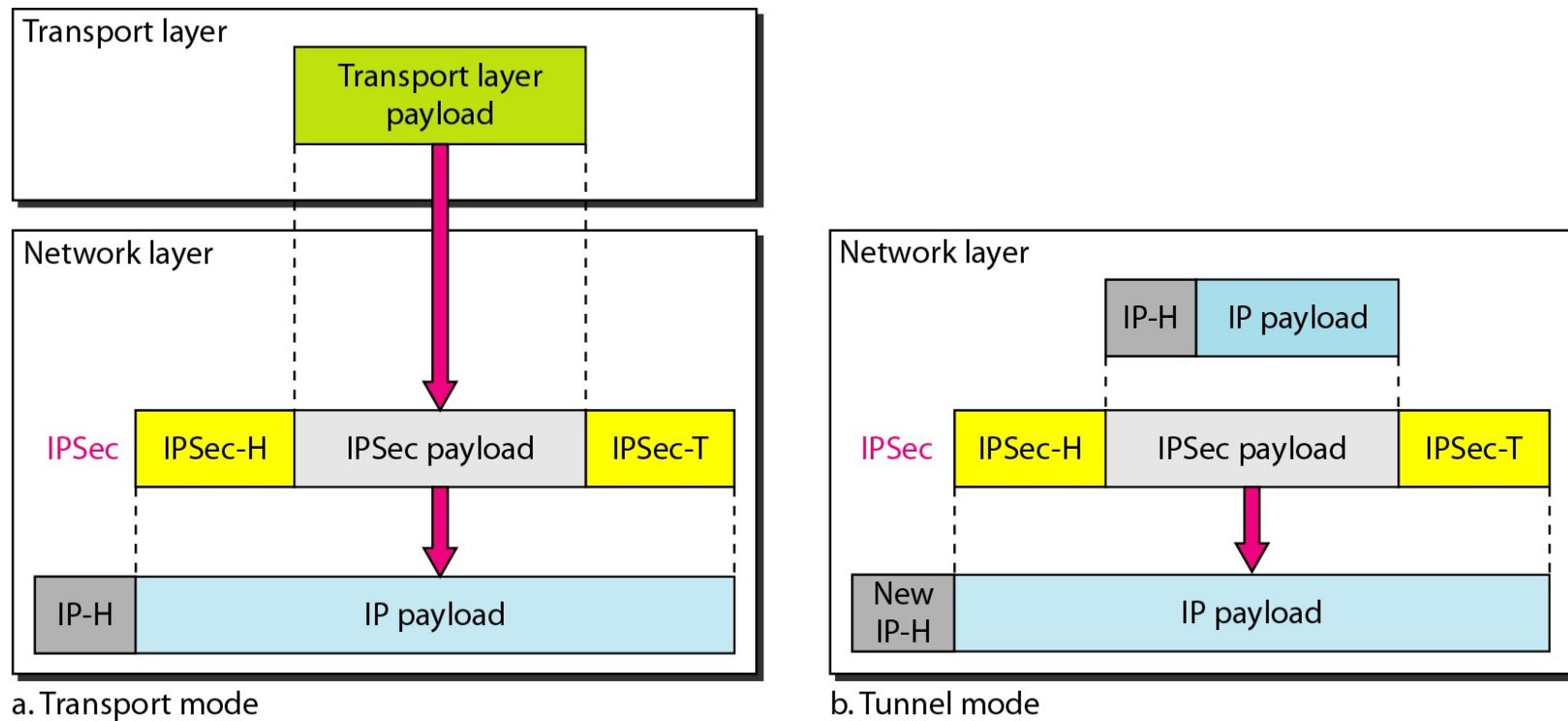
---

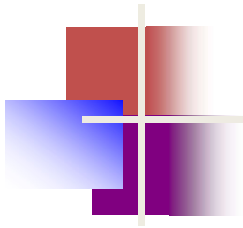
Figure 32.2 *TCP/IP protocol suite and IPSec*

---



**Figure 32.3** *Transport mode and tunnel modes of IPSec protocol*





## *Note*

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

Figure 32.4 *Transport mode in action*

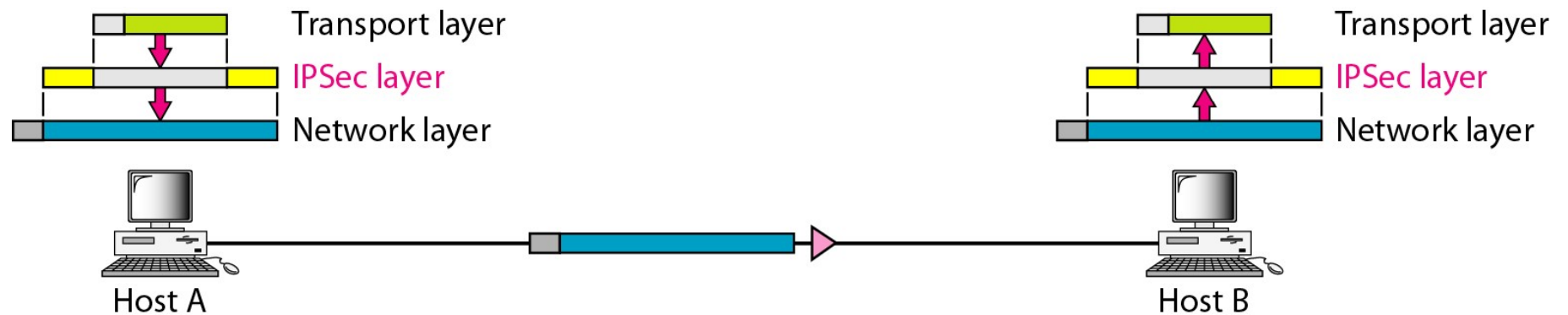
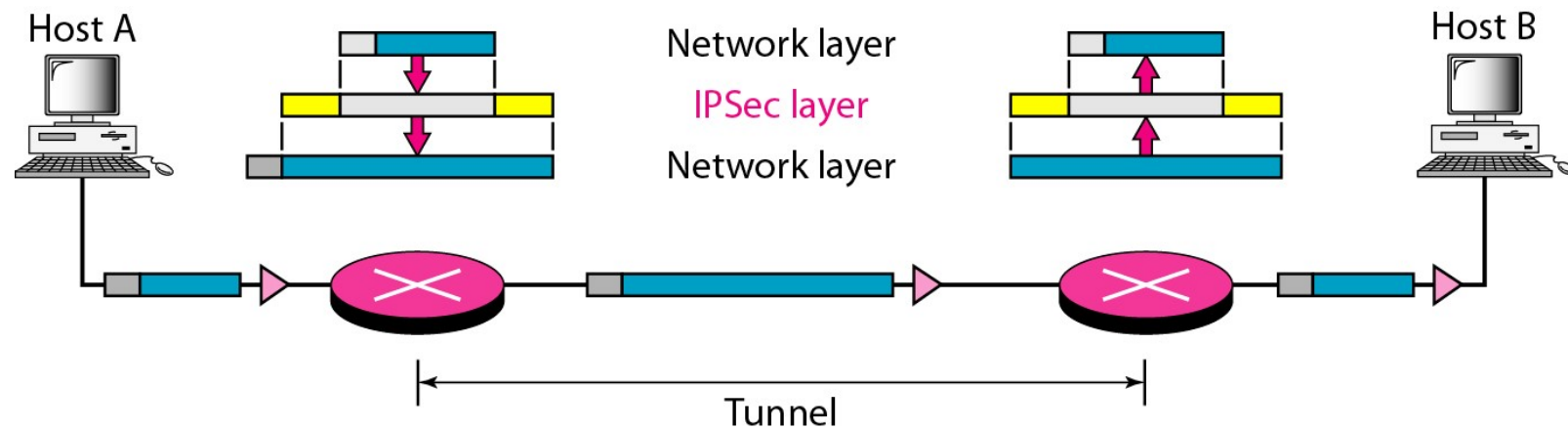
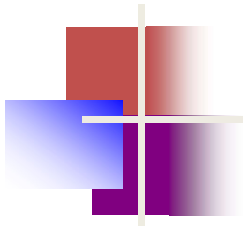




Figure 32.5 *Tunnel mode in action*

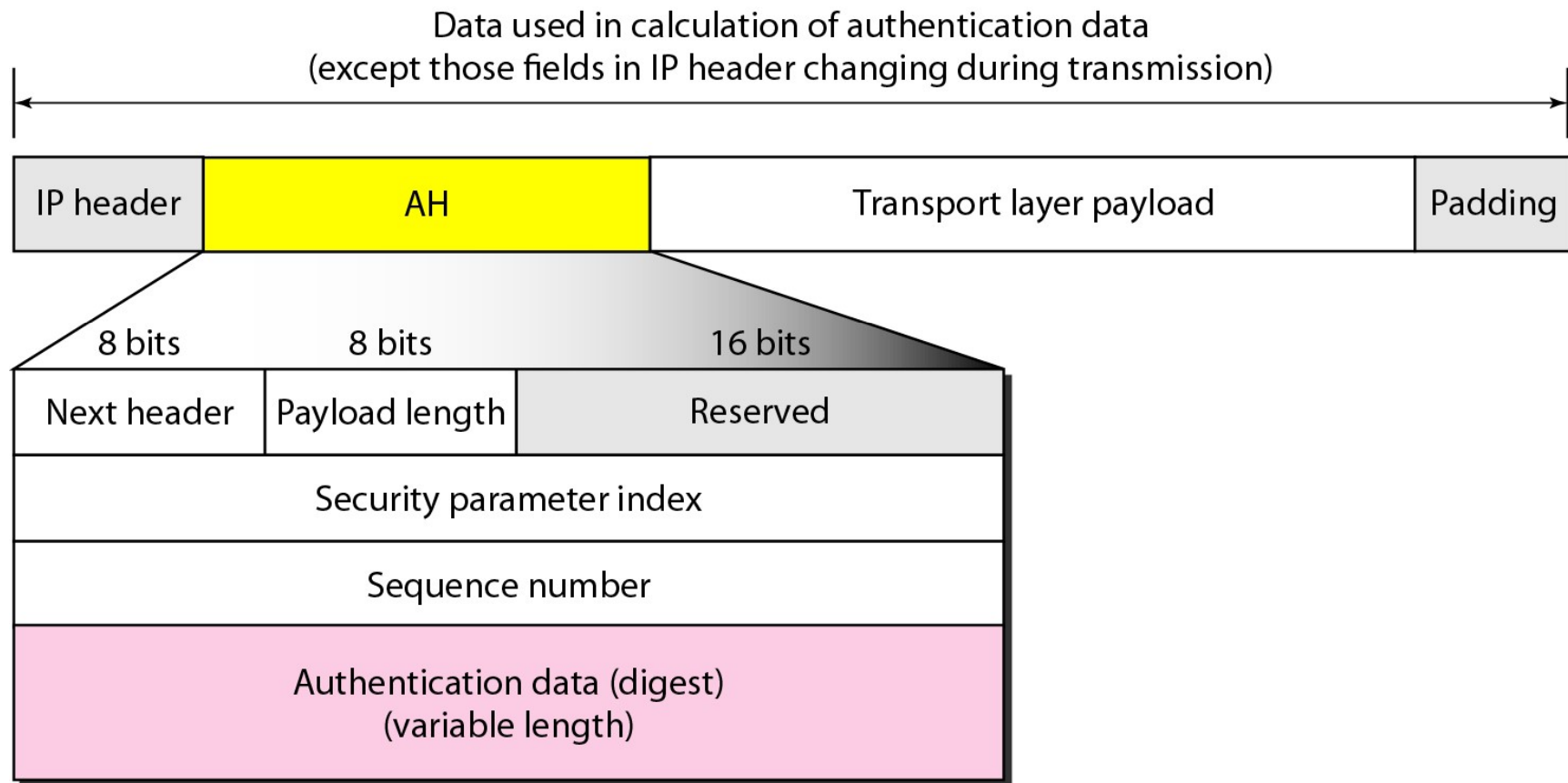


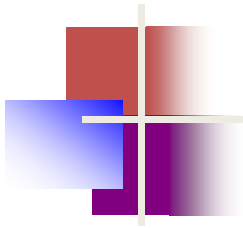


### *Note*

IPSec in tunnel mode protects the original IP header.

**Figure 32.6** *Authentication Header (AH) Protocol in transport mode*

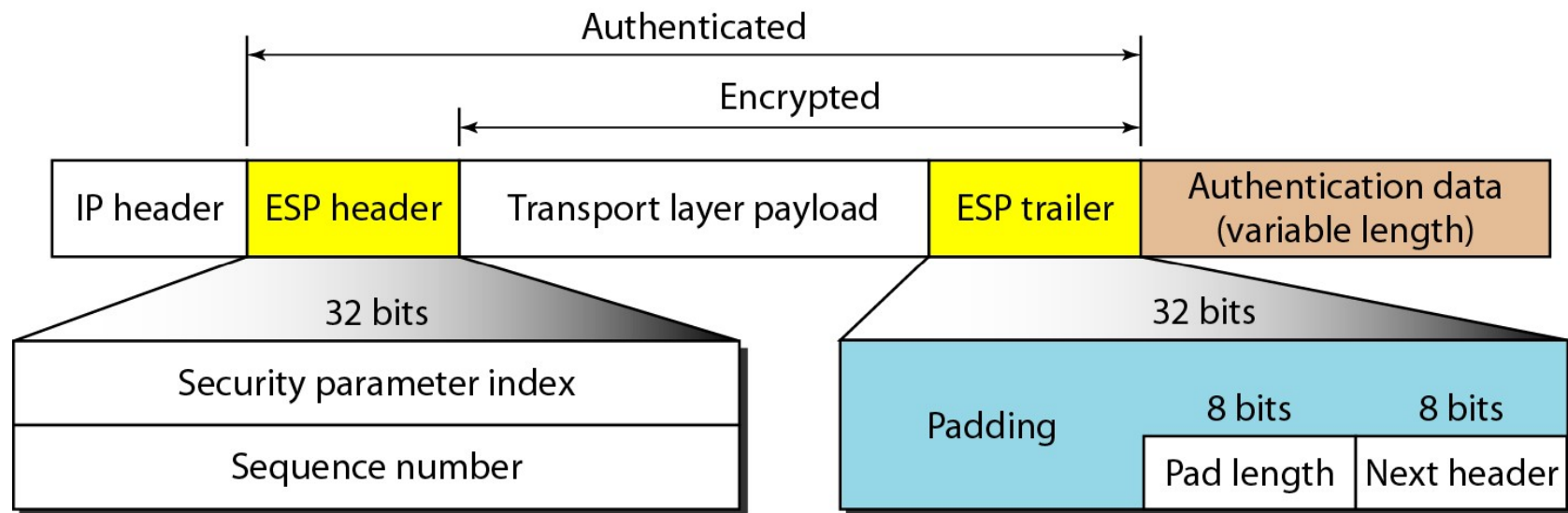


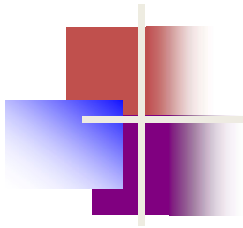


### *Note*

The AH Protocol provides source authentication and data integrity,  
but not privacy.

**Figure 32.7** *Encapsulating Security Payload (ESP) Protocol in transport mode*





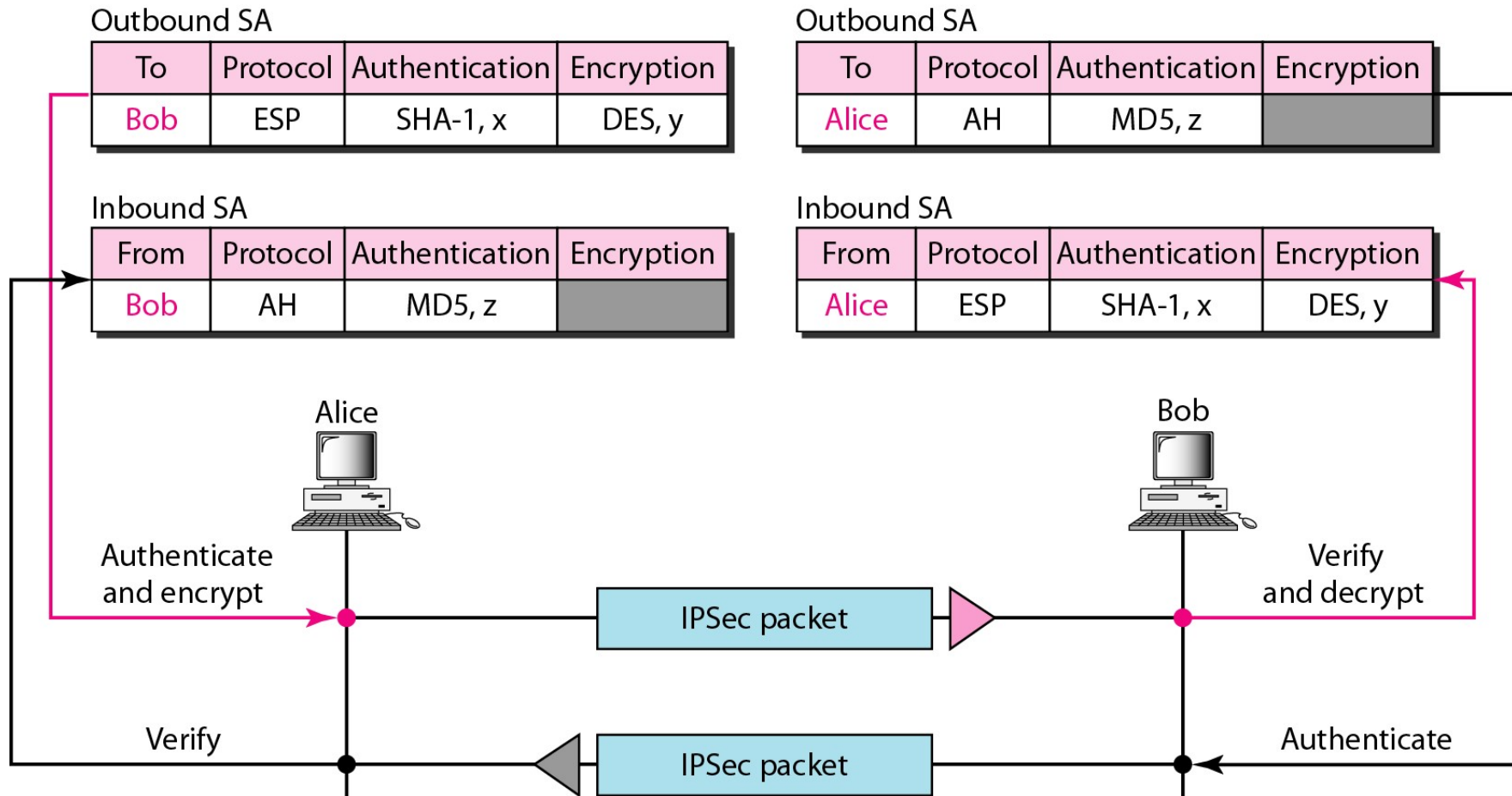
### *Note*

ESP provides source authentication, data integrity, and privacy.

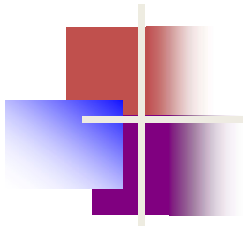
**Table 32.1** *IPSec services*

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

**Figure 32.8** *Simple inbound and outbound security associations*







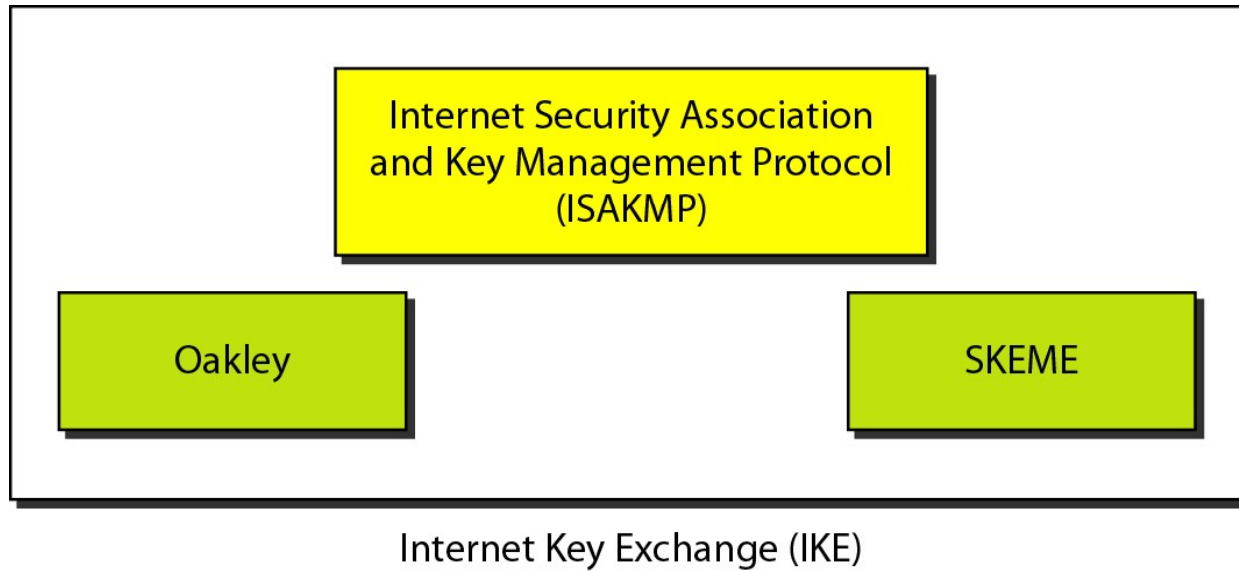
*Note*

IKE creates SAs for IPSec.

---

Figure 32.9 *IKE components*

---



**Table 32.2** *Addresses for private networks*

<i>Prefix</i>	<i>Range</i>	<i>Total</i>
10/8	10.0.0.0 to 10.255.255.255	$2^{24}$
172.16/12	172.16.0.0 to 172.31.255.255	$2^{20}$
192.168/16	192.168.0.0 to 192.168.255.255	$2^{16}$

Figure 32.10 *Private network*

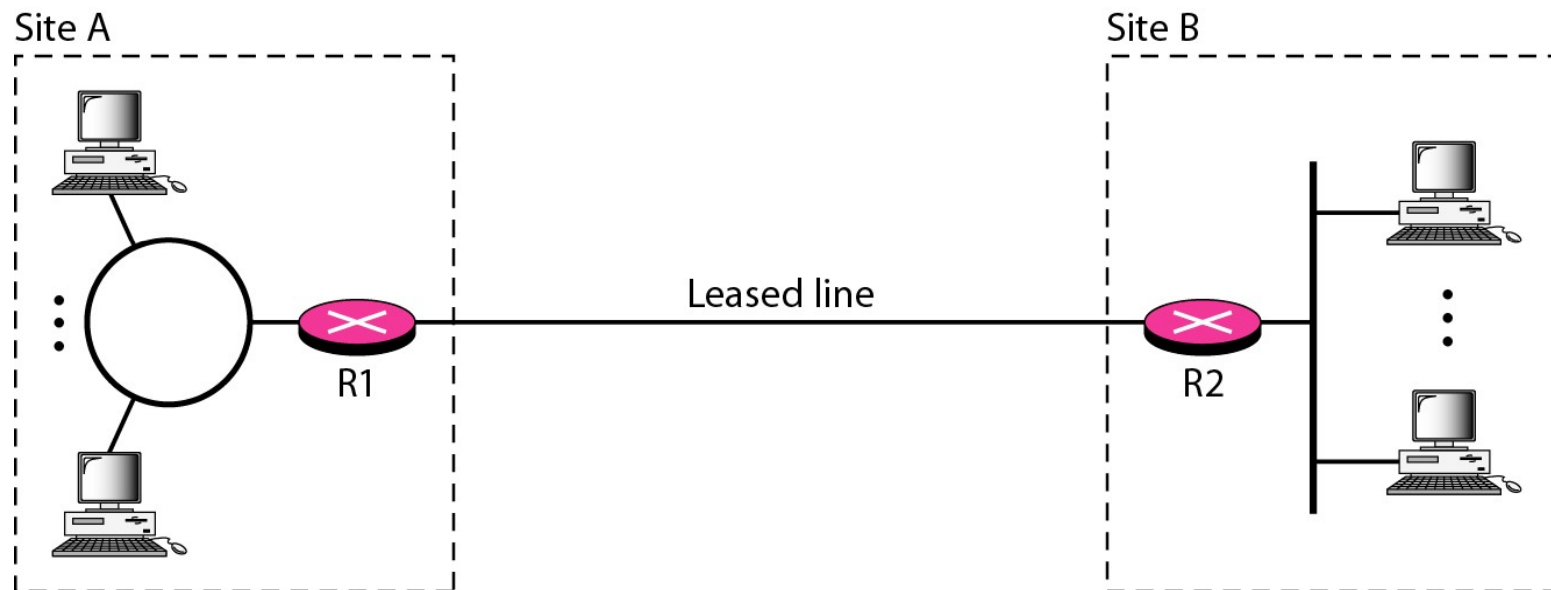


Figure 32.11 *Hybrid network*

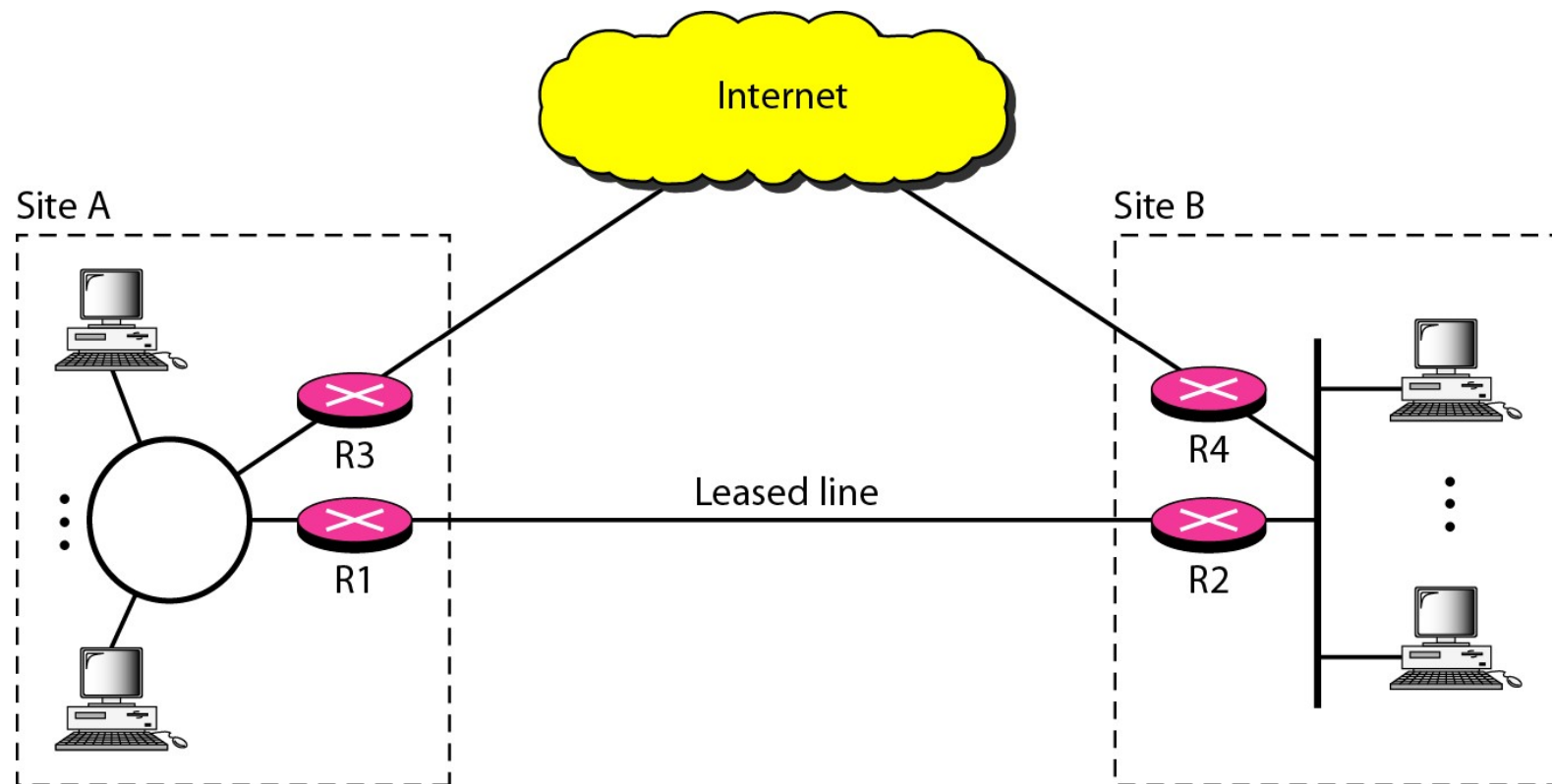


Figure 32.12 *Virtual private network*

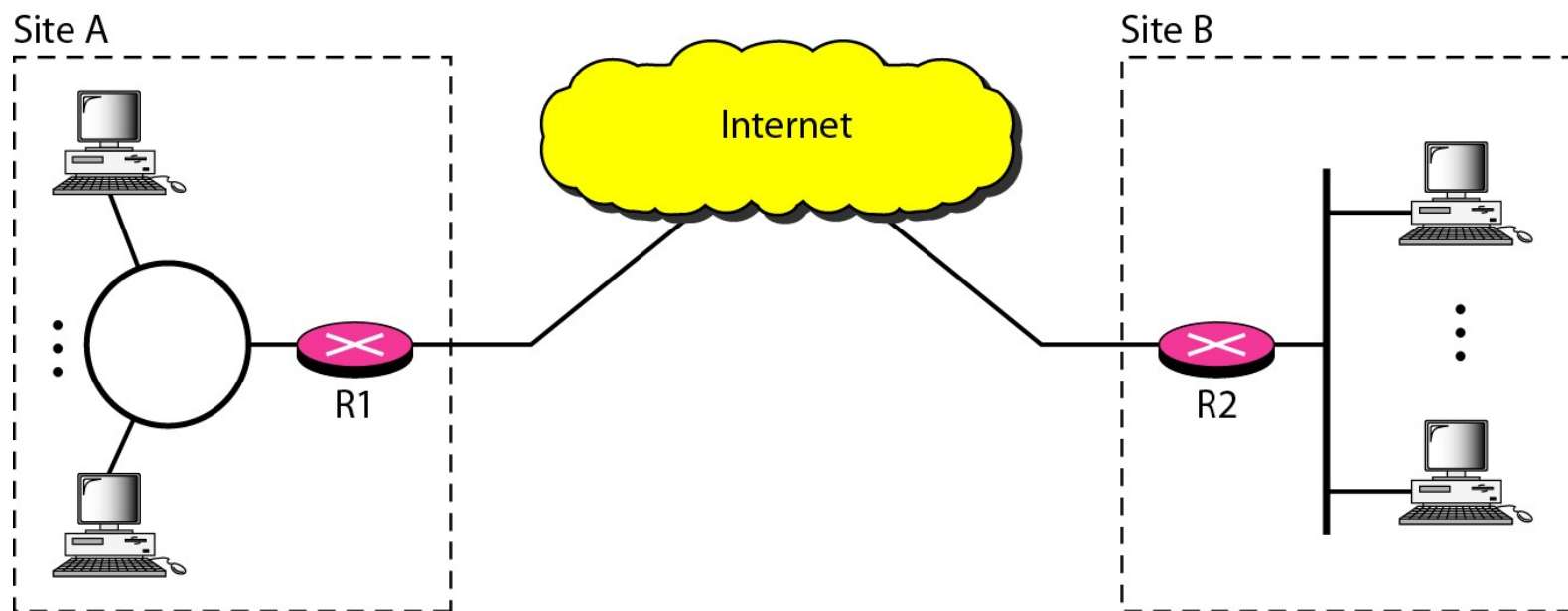
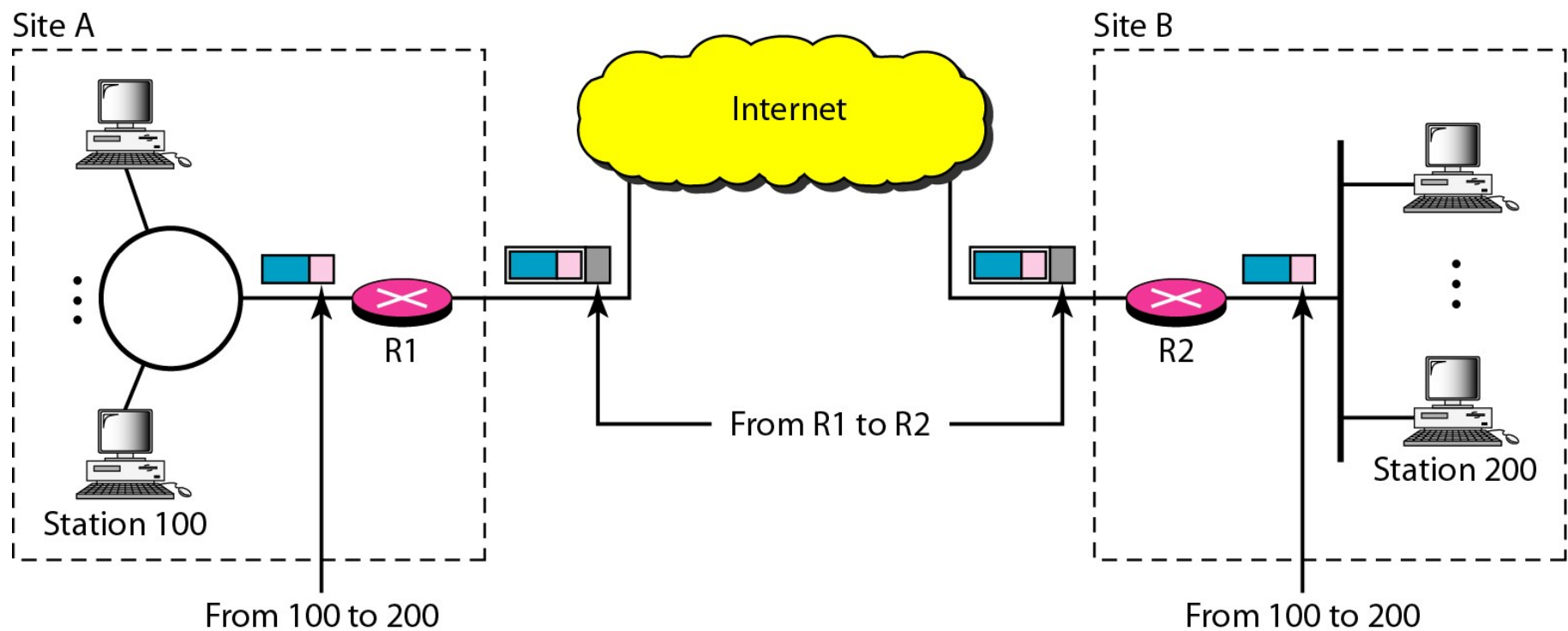


Figure 32.13 *Addressing in a VPN*



*Two protocols are dominant today for providing security at the transport layer: the Secure Sockets Layer (SSL) Protocol and the Transport Layer Security (TLS) Protocol. The latter is actually an IETF version of the former.*

### Topics discussed in this section:

SSL Services

Security Parameters

Sessions and Connections

Four Protocols

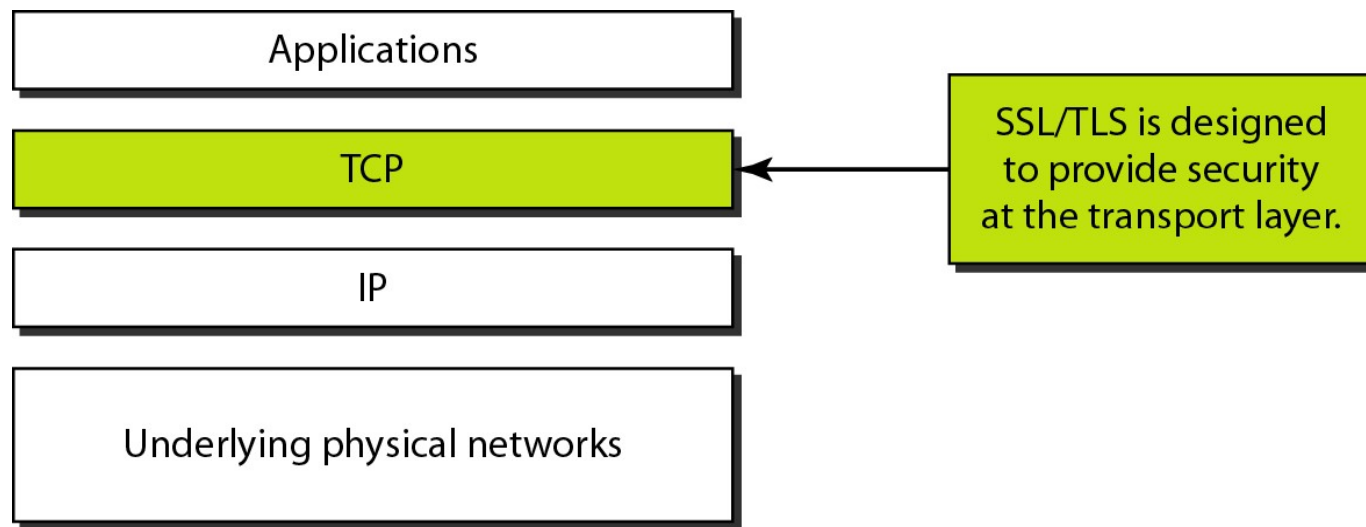
Transport Layer Security



---

Figure 32.14 *Location of SSL and TLS in the Internet model*

---

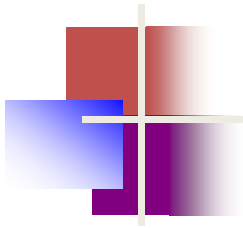


**Table 32.3** *SSL cipher suite list*

<i>Cipher Suite</i>	<i>Key Exchange Algorithm</i>	<i>Encryption Algorithm</i>	<i>Hash Algorithm</i>
SSL_NULL_WITH_NULL_NULL	NULL	NULL	NULL
SSL_RSA_WITH_NULL_MD5	RSA	NULL	MD5
SSL_RSA_WITH_NULL_SHA	RSA	NULL	SHA
SSL_RSA_WITH_RC4_128_MD5	RSA	RC4_128	MD5
SSL_RSA_WITH_RC4_128_SHA	RSA	RC4_128	SHA
SSL_RSA_WITH_IDEA_CBC_SHA	RSA	IDEA_CBC	SHA
SSL_RSA_WITH_DES_CBC_SHA	RSA	DES_CBC	SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES_EDE_CBC	SHA
SSL_DH_anon_WITH_RC4_128_MD5	DH_anon	RC4_128	MD5
SSL_DH_anon_WITH_DES_CBC_SHA	DH_anon	DES_CBC	SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	DH_anon	3DES_EDE_CBC	SHA

**Table 32.3** *SSL cipher suite list (continued)*

<i>Cipher Suite</i>	<i>Key Exchange Algorithm</i>	<i>Encryption Algorithm</i>	<i>Hash Algorithm</i>
SSL_DHE_RSA_WITH_DES_CBC_SHA	DHE_RSA	DES_CBC	SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE_RSA	3DES_EDE_CBC	SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA	DHE_DSS	DES_CBC	SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DHE_DSS	3DES_EDE_CBC	SHA
SSL_DH_RSA_WITH_DES_CBC_SHA	DH_RSA	DES_CBC	SHA
SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA	DH_RSA	3DES_EDE_CBC	SHA
SSL_DH_DSS_WITH_DES_CBC_SHA	DH_DSS	DES_CBC	SHA
SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA	DH_DSS	3DES_EDE_CBC	SHA
SSL_FORTEZZA_DMS_WITH_NULL_SHA	FORTEZZA_DMS	NULL	SHA
SSL_FORTEZZA_DMS_WITH_FORTEZZA_CBC_SHA	FORTEZZA_DMS	FORTEZZA_CBC	SHA
SSL_FORTEZZA_DMS_WITH_RC4_128_SHA	FORTEZZA_DMS	RC4_128	SHA



### *Note*

The client and the server have six different cryptography secrets.

Figure 32.15 *Creation of cryptographic secrets in SSL*

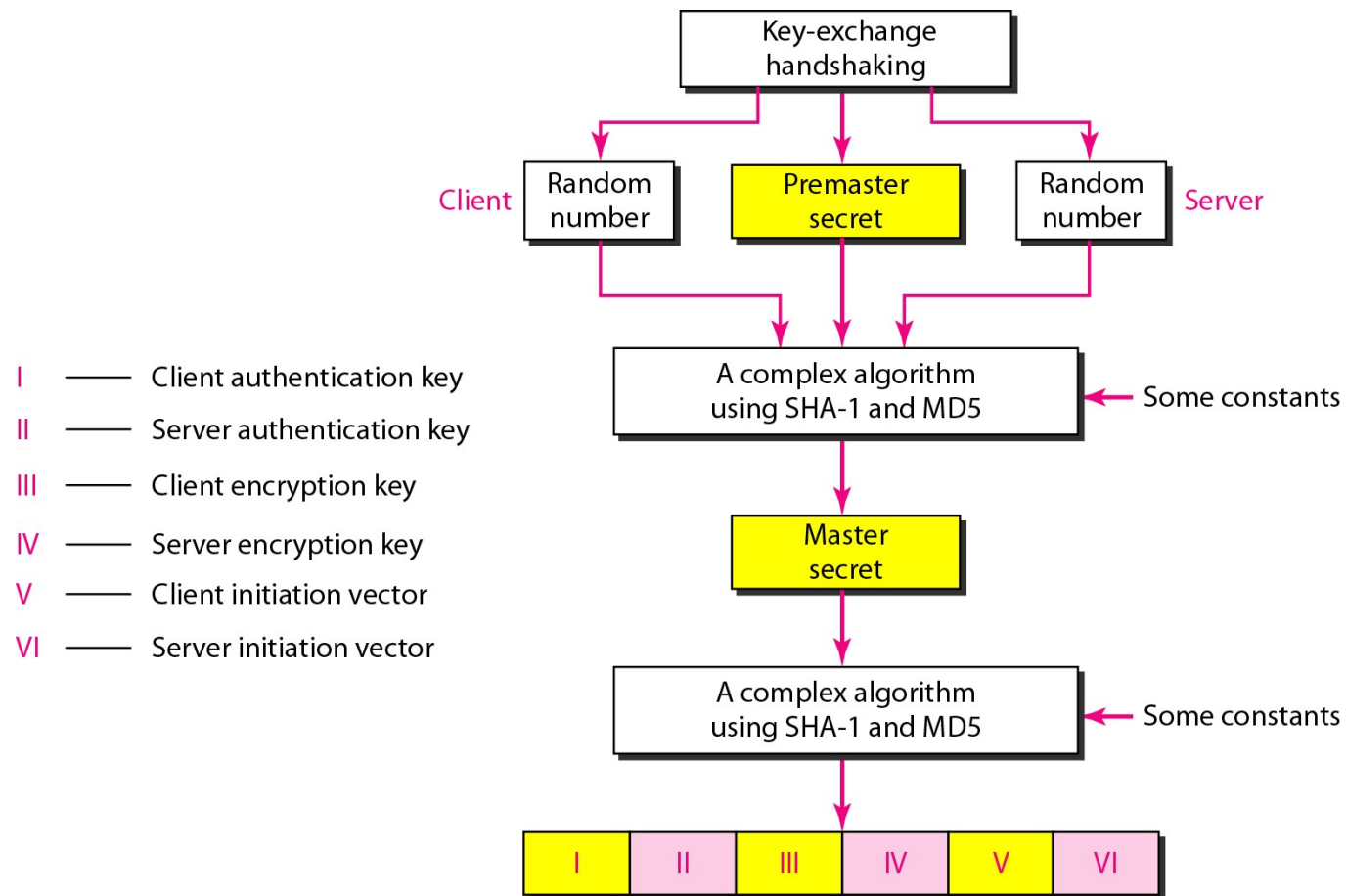


Figure 32.16 *Four SSL protocols*

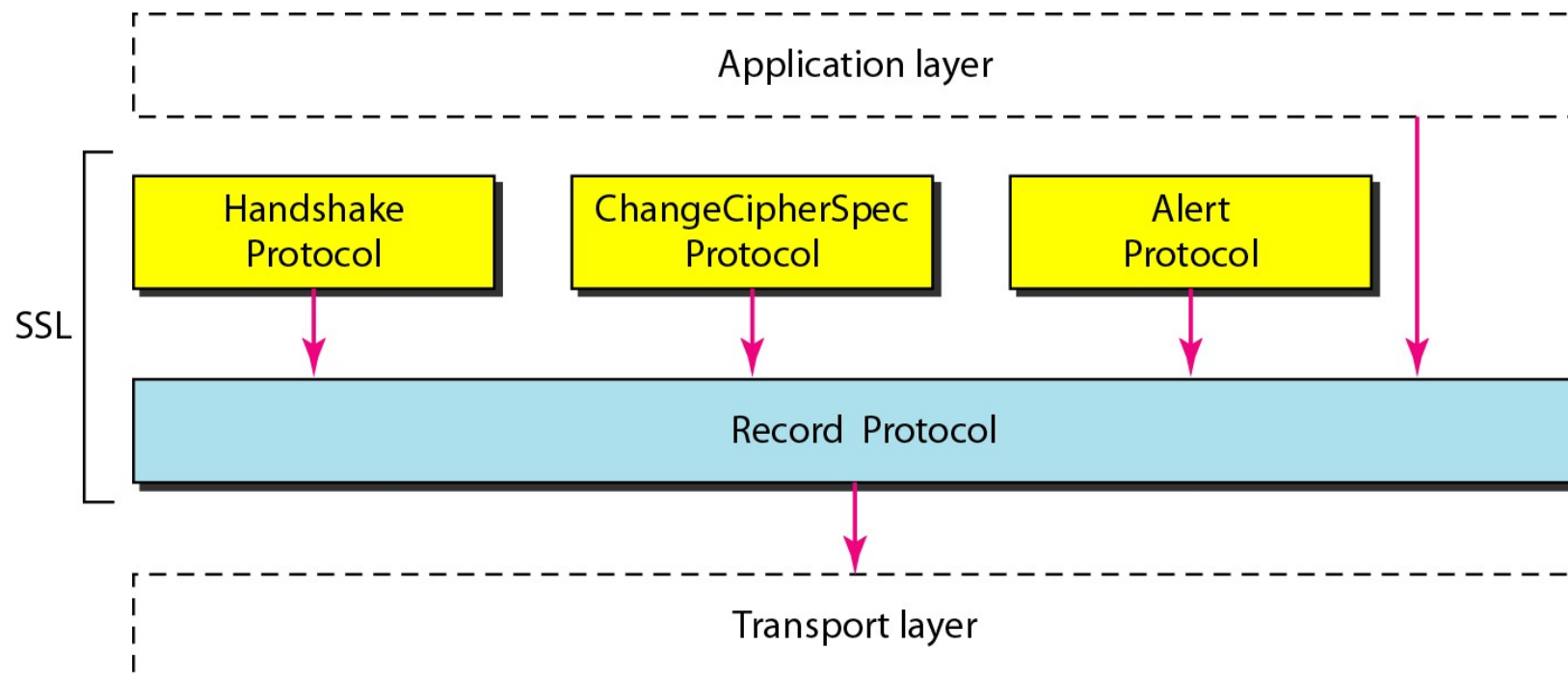


Figure 32.17 *Handshake Protocol*

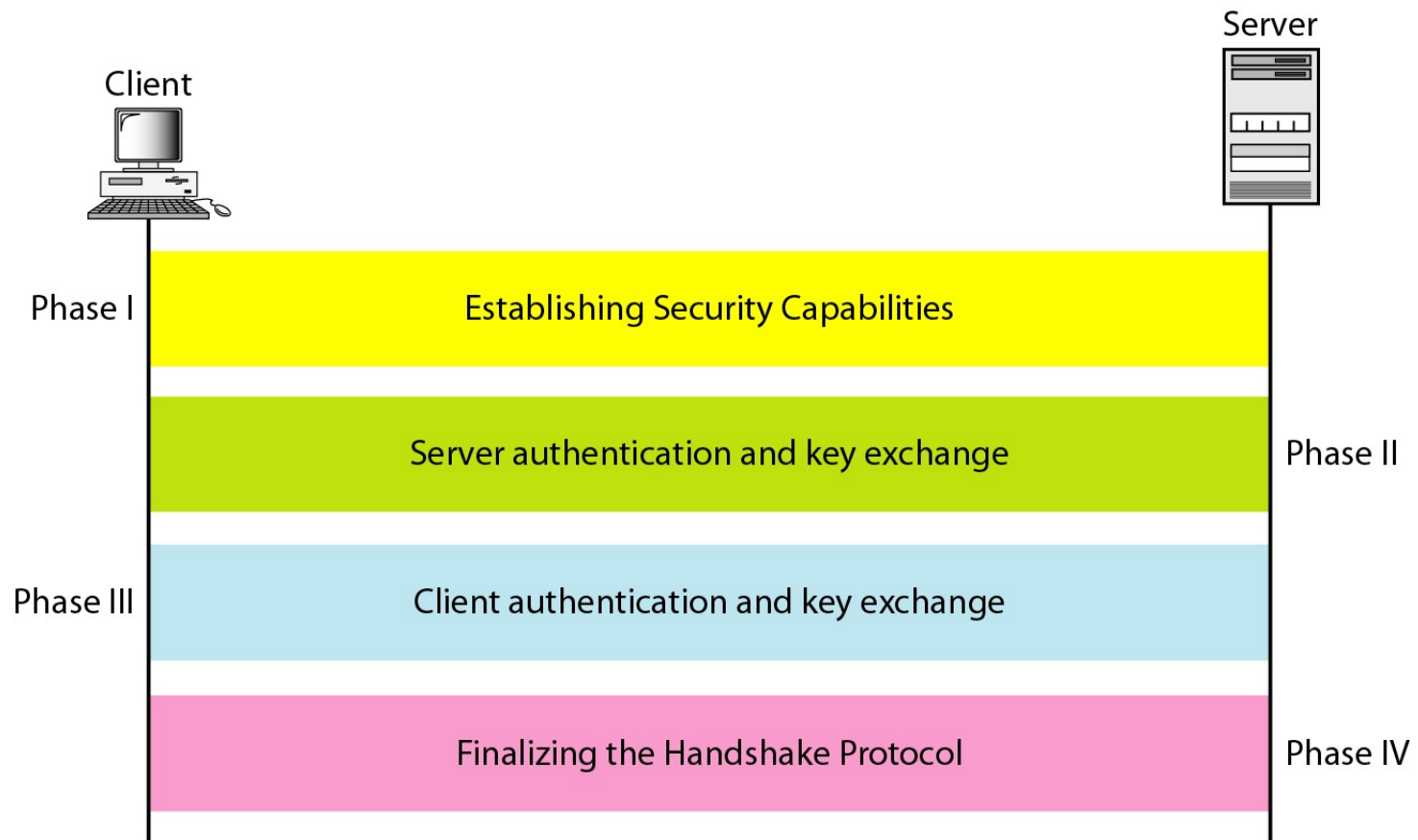


Figure 32.18 *Processing done by the Record Protocol*

