Chapter 18

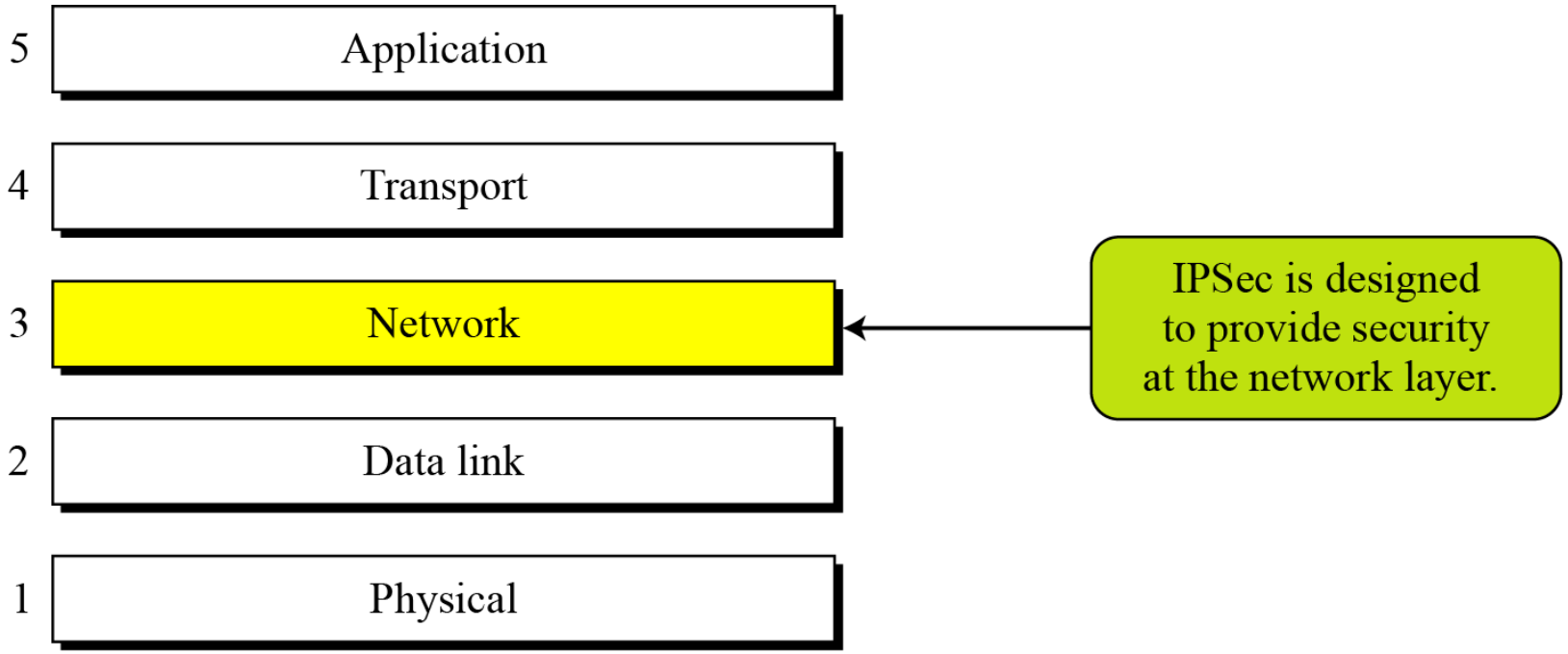
Security at the Network Layer: IPSec

Chapter 18

Objectives

- To define the architecture of IPSec
- To discuss the application of IPSec in transport and tunnel modes
- To discuss how IPSec can be used to provide only authentication
- To discuss how IPSec can be used to provide both confidentiality and authentication
- To define Security Association and explain how it is implemented for IPSec
- To define Internet Key Exchange and explain how it is used by IPSec.

Figure 18.1 *TCP/IP Protocol Suite and IPSec*



18-1 TWO MODES

IPSec operates in one of two different modes: transport mode or tunnel mode.

Topics discussed in this section:

18.1.1 Transport Mode

In transport mode, IPSec protects what is delivered from the transport layer to the network layer.

Note

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

Figure 18.2 *IPSec in transport mode*

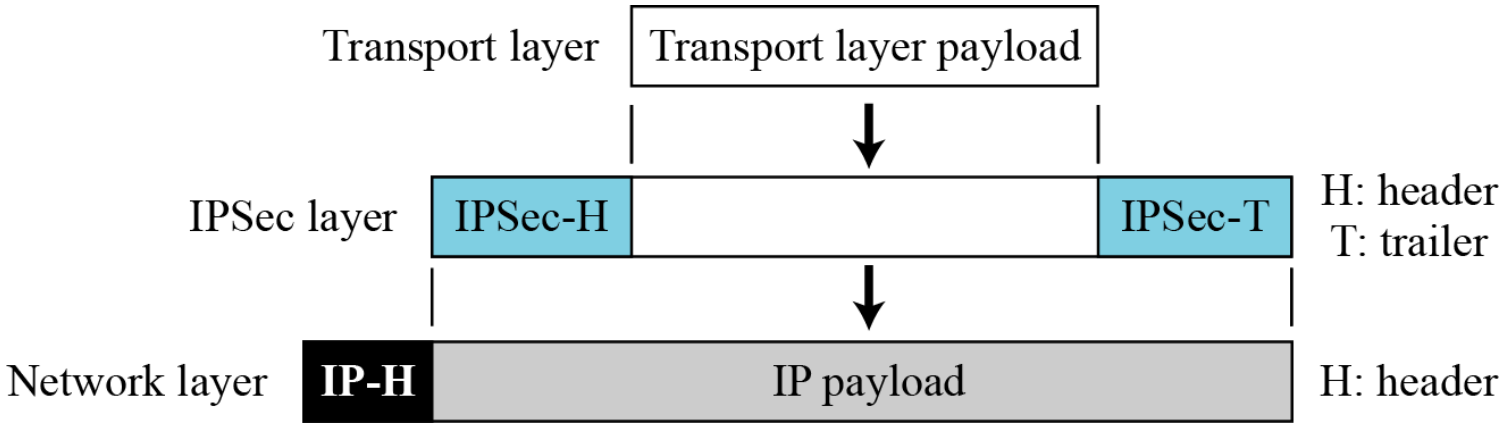
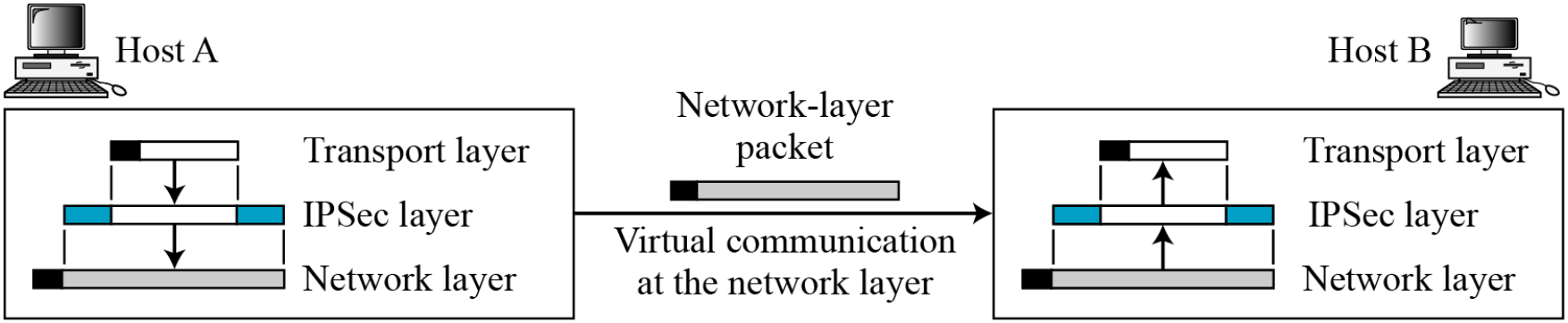


Figure 18.3 *Transport mode in action*



18.1.2 Tunnel Mode

In tunnel mode, IPSec protects the entire IP packet. It takes an IP packet, including the header, applies IPSec security methods to the entire packet, and then adds a new IP header.

Note

IPSec in tunnel mode protects the original IP header.

18.1.2 (Continued)

Figure 18.4 *IPSec in tunnel mode*

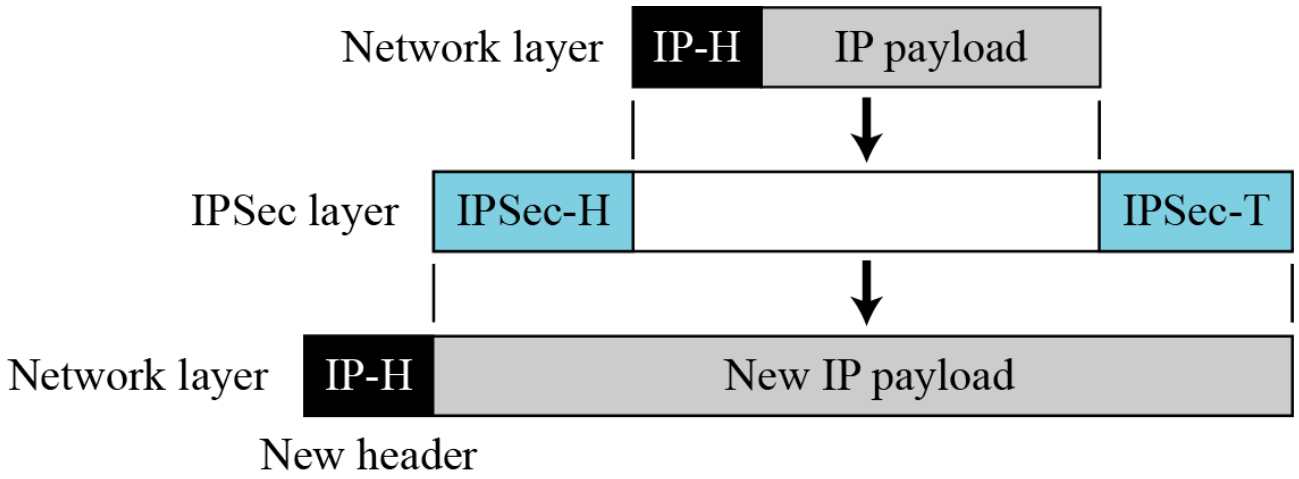
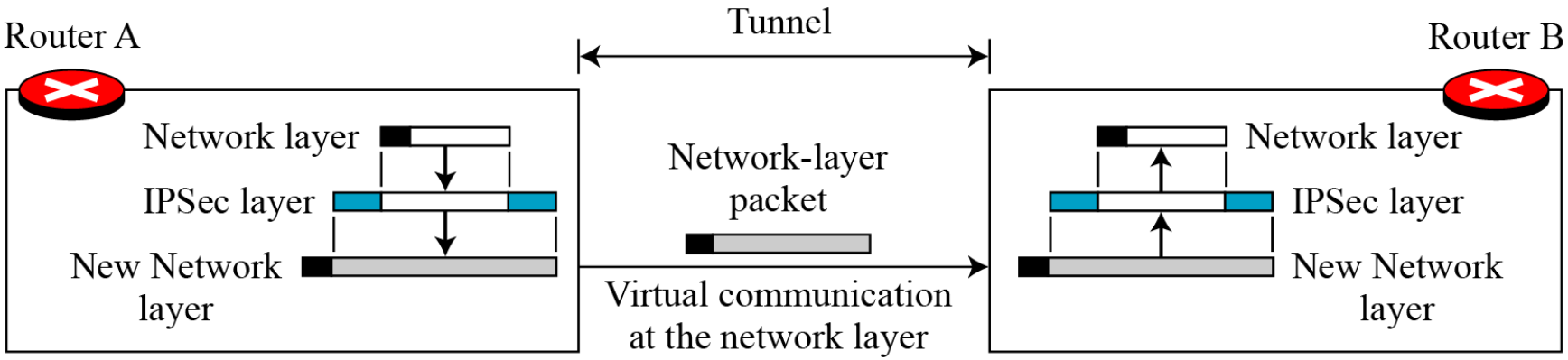
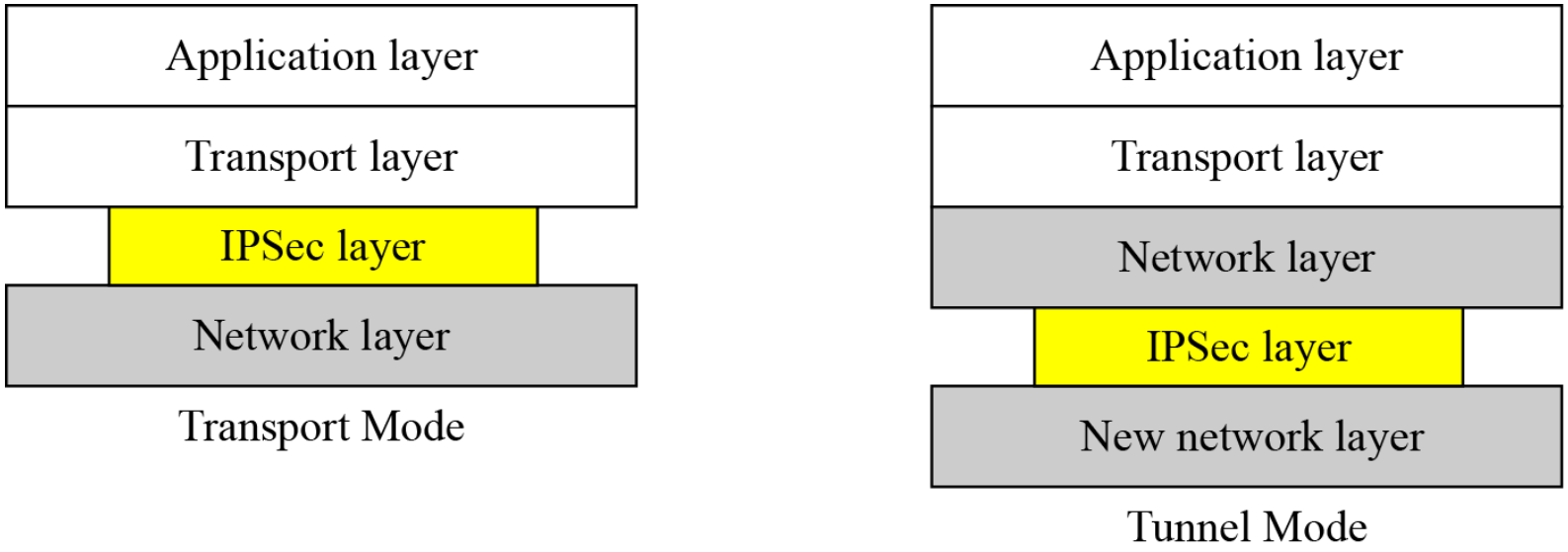


Figure 18.5 *Tunnel mode in action*



18.1.3 Comparison

Figure 18.6 *Transport mode versus tunnel mode*



18-2 TWO SECURITY PROTOCOL

IPSec defines two protocols—the Authentication Header (AH) Protocol and the Encapsulating Security Payload (ESP) Protocol—to provide authentication and/or encryption for packets at the IP level.

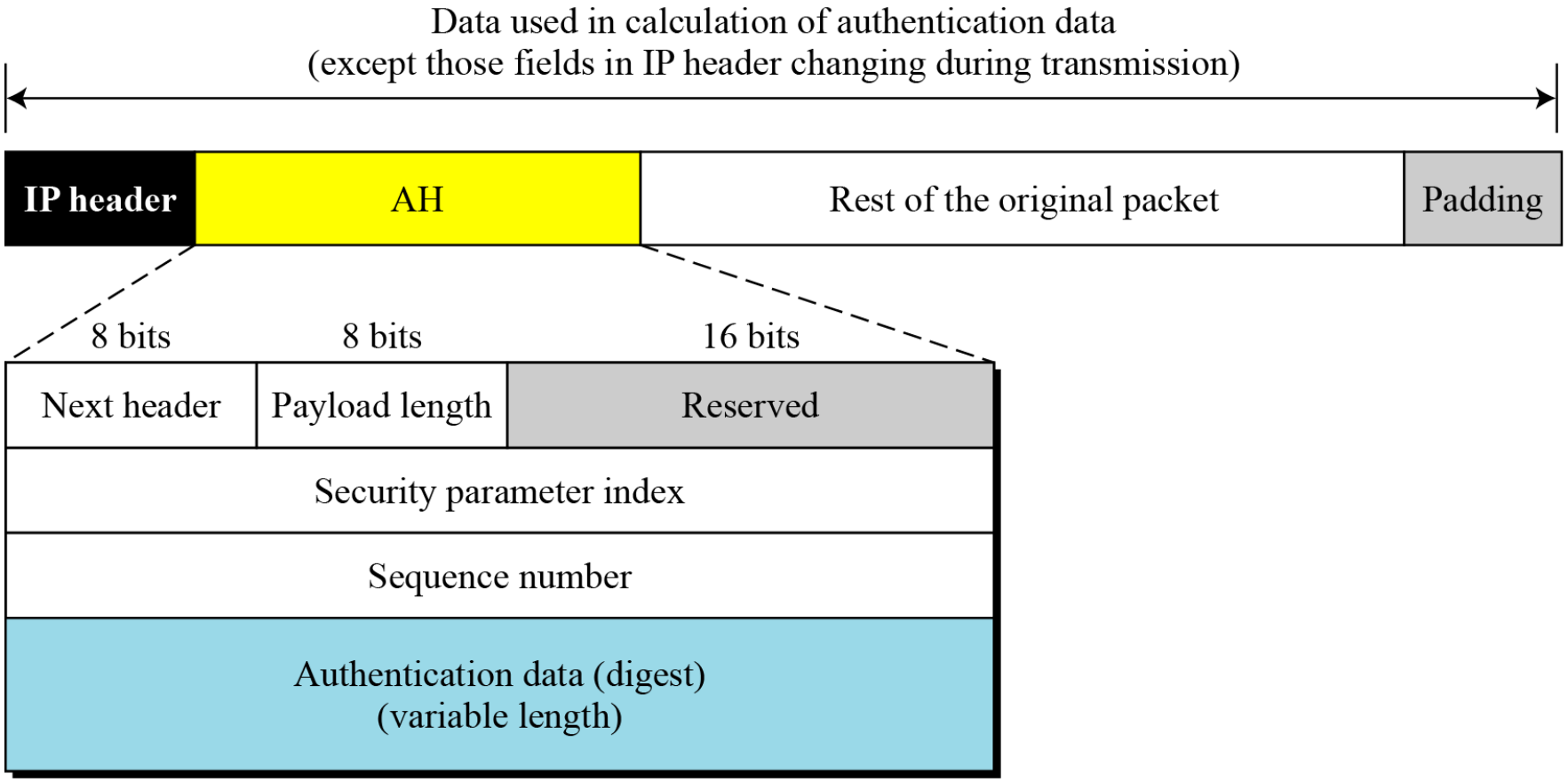
Topics discussed in this section:

18.2.1 Authentication Header (AH)

Note

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

Figure 18.7 *Authentication Header (AH) protocol*

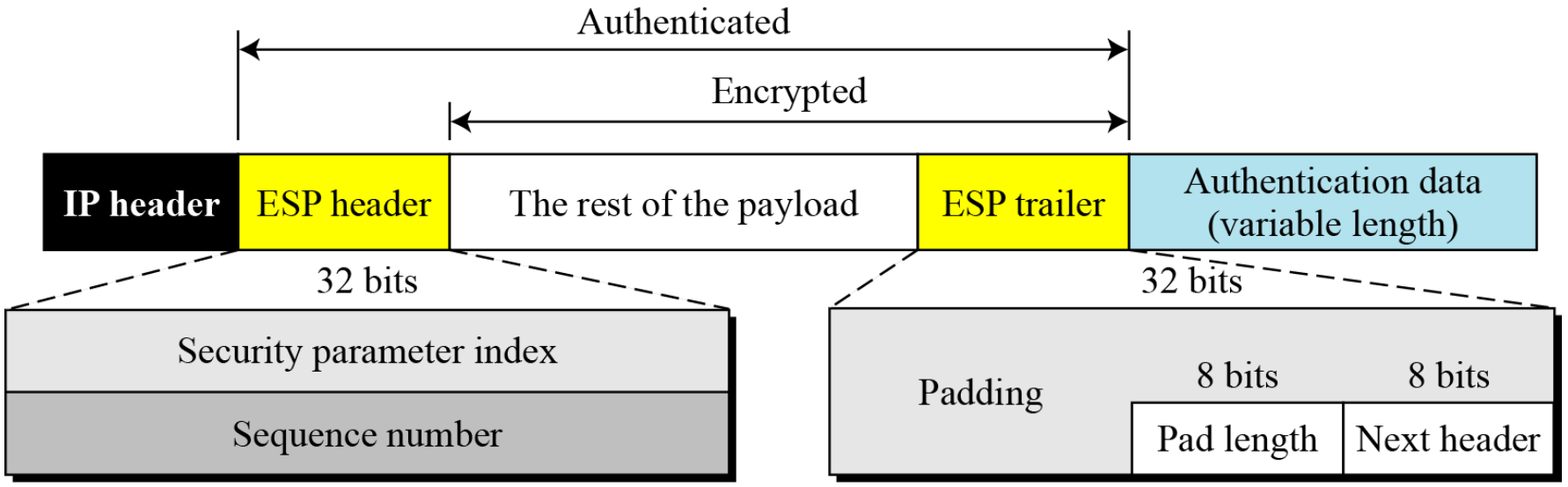


18.2.2 Encapsulating Security Payload (ESP)

Note

ESP provides source authentication, data integrity, and privacy.

Figure 18.8 *ESP*



18.2.3 IPv4 and IPv6

IPSec supports both IPv4 and IPv6. In IPv6, however, AH and ESP are part of the extension header.

18.2.4 AH versus ESP

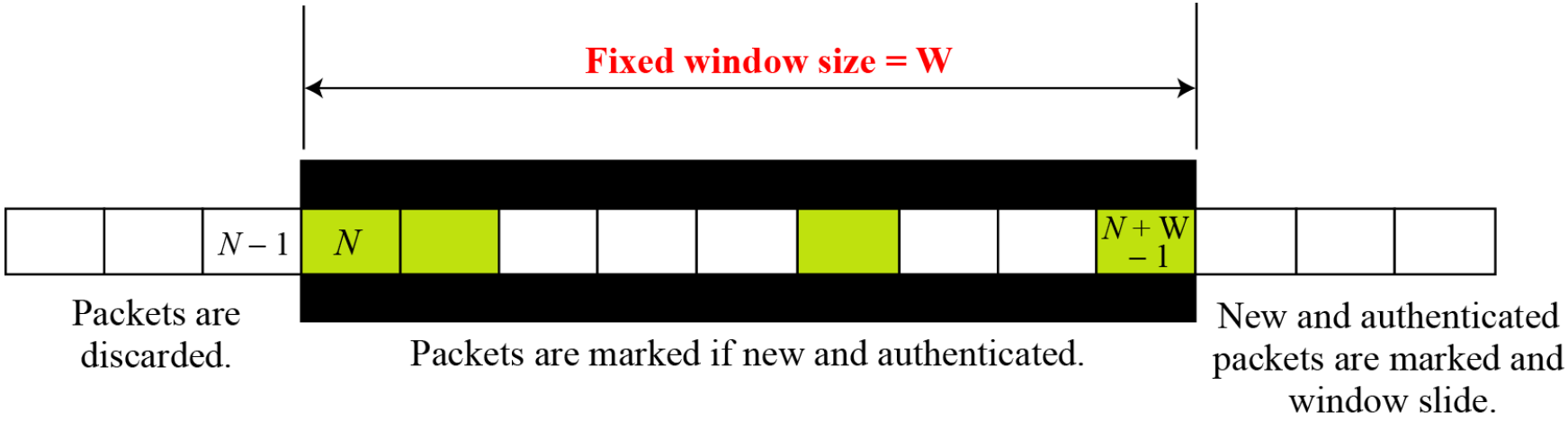
The ESP protocol was designed after the AH protocol was already in use. ESP does whatever AH does with additional functionality (privacy).

18.2.5 Services Provided by IPSec

Table 18.1 *IPSec services*

Services	*AH*	*ESP*
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 18.9 *Replay window*



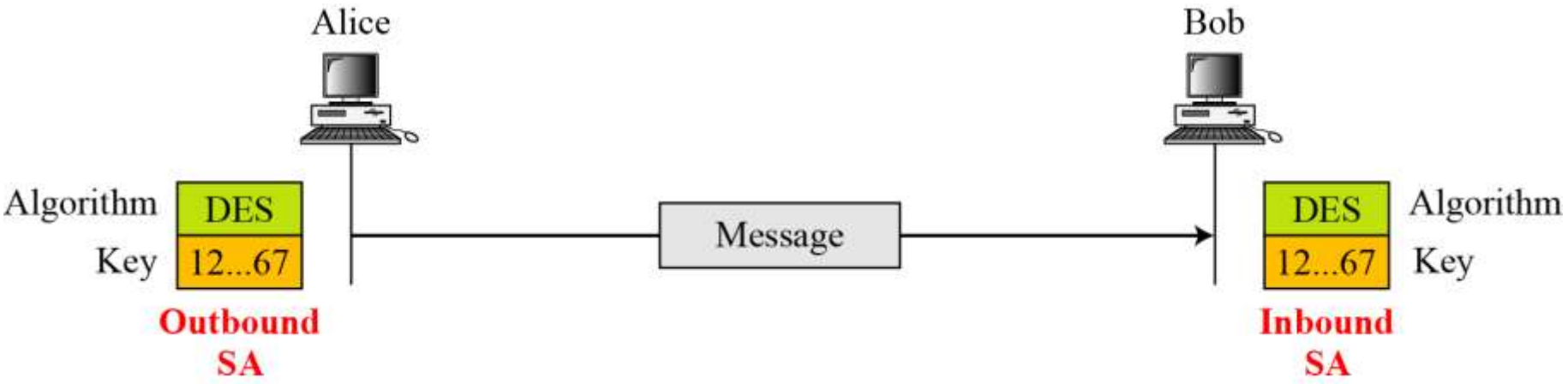
18-3 SECURITY ASSOCIATION

Security Association is a very important aspect of IPSec. IPSec requires a logical relationship, called a Security Association (SA), between two hosts. This section first discusses the idea and then shows how it is used in IPSec.

Topics discussed in this section:

18.3.1 Idea of Security Association

Figure 18.10 *Simple SA*



18.3.2 Security Association Database (SAD)

Figure 18.11 *SAD*

Index	SN	OF	ARW	AH/ESP	LT	Mode	MTU
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							

Security Association Database

Legend:

SPI: Security Parameter Index

DA: Destination Address

AH/ESP: Information for either one

P: Protocol

Mode: IPSec Mode Flag

SN: Sequence Number

OF: Overflow Flag

ARW: Anti-Replay Window

LT: Lifetime

MTU: Path MTU (Maximum Transfer Unit)

18.3.2 (Continued)

Table 18.2 *Typical SA Parameters*

Parameters	*Description*
Sequence Number Counter	This is a 32-bit value that is used to generate sequence numbers for the AH or ESP header.
Sequence Number Overflow	This is a flag that defines a station's options in the event of a sequence number overflow.
Anti-Replay Window	This detects an inbound replayed AH or ESP packet.
AH Information	This section contains information for the AH protocol: 1. Authentication algorithm 2. Keys 3. Key lifetime 4. Other related parameters
ESP Information	This section contains information for the ESP protocol: 1. Encryption algorithm 2. Authentication algorithm 3. Keys 4. Key lifetime 5. Initiator vectors 6. Other related parameters
SA Lifetime	This defines the lifetime for the SA.
IPSec Mode	This defines the mode, transport or tunnel.
Path MTU	This defines the path MTU (fragmentation).

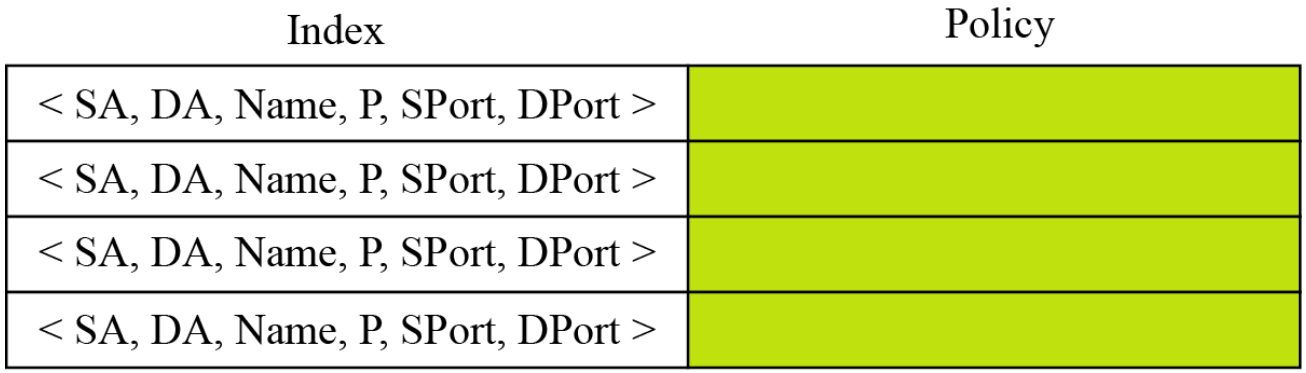
18-4 SECURITY POLICY

Another import aspect of IPSec is the Security Policy (SP), which defines the type of security applied to a packet when it is to be sent or when it has arrived. Before using the SAD, discussed in the previous section, a host must determine the predefined policy for the packet.

Topics discussed in this section:

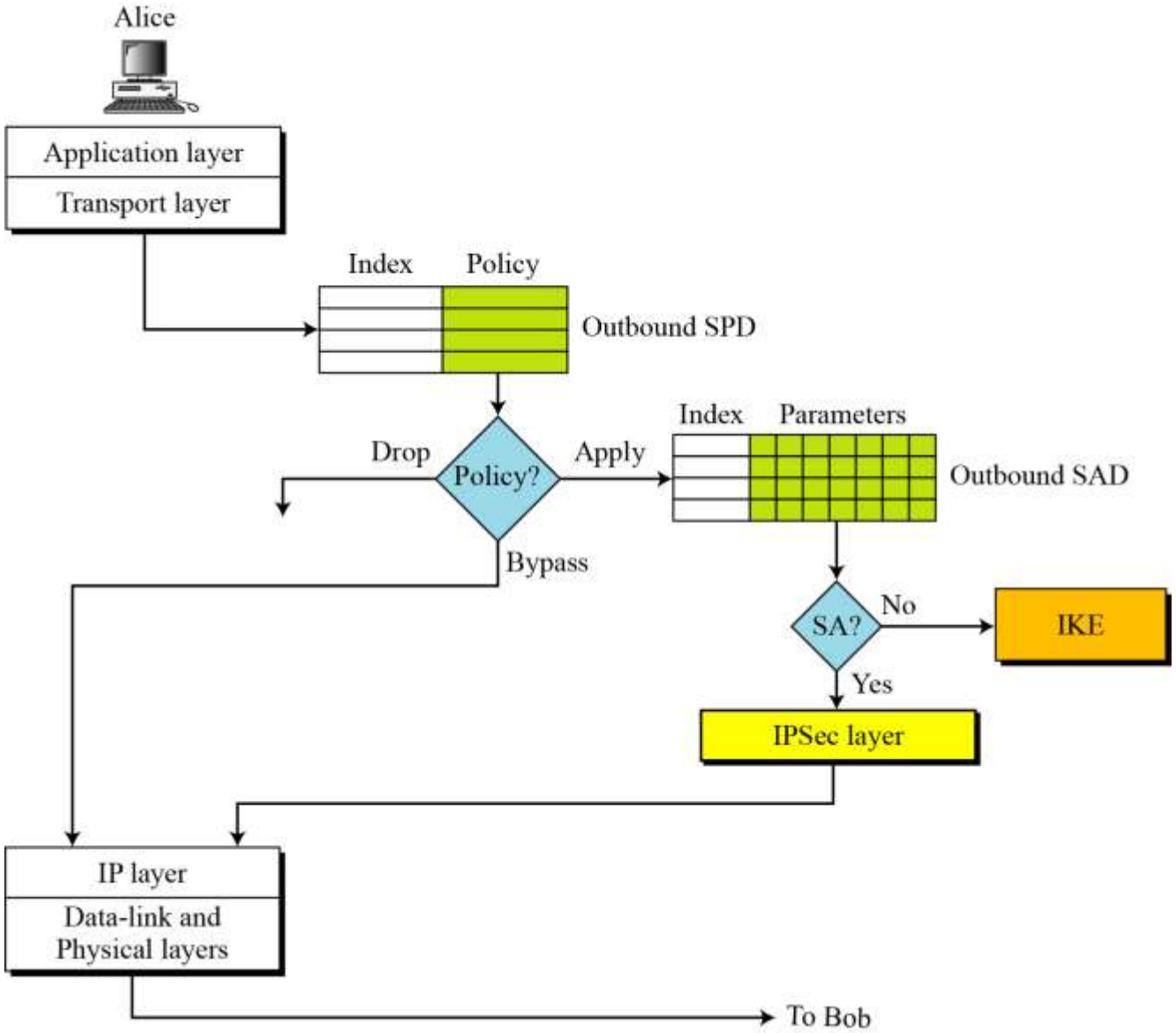
18.4.1 Security Policy Database

Figure 18.12 *Connection identifiers*



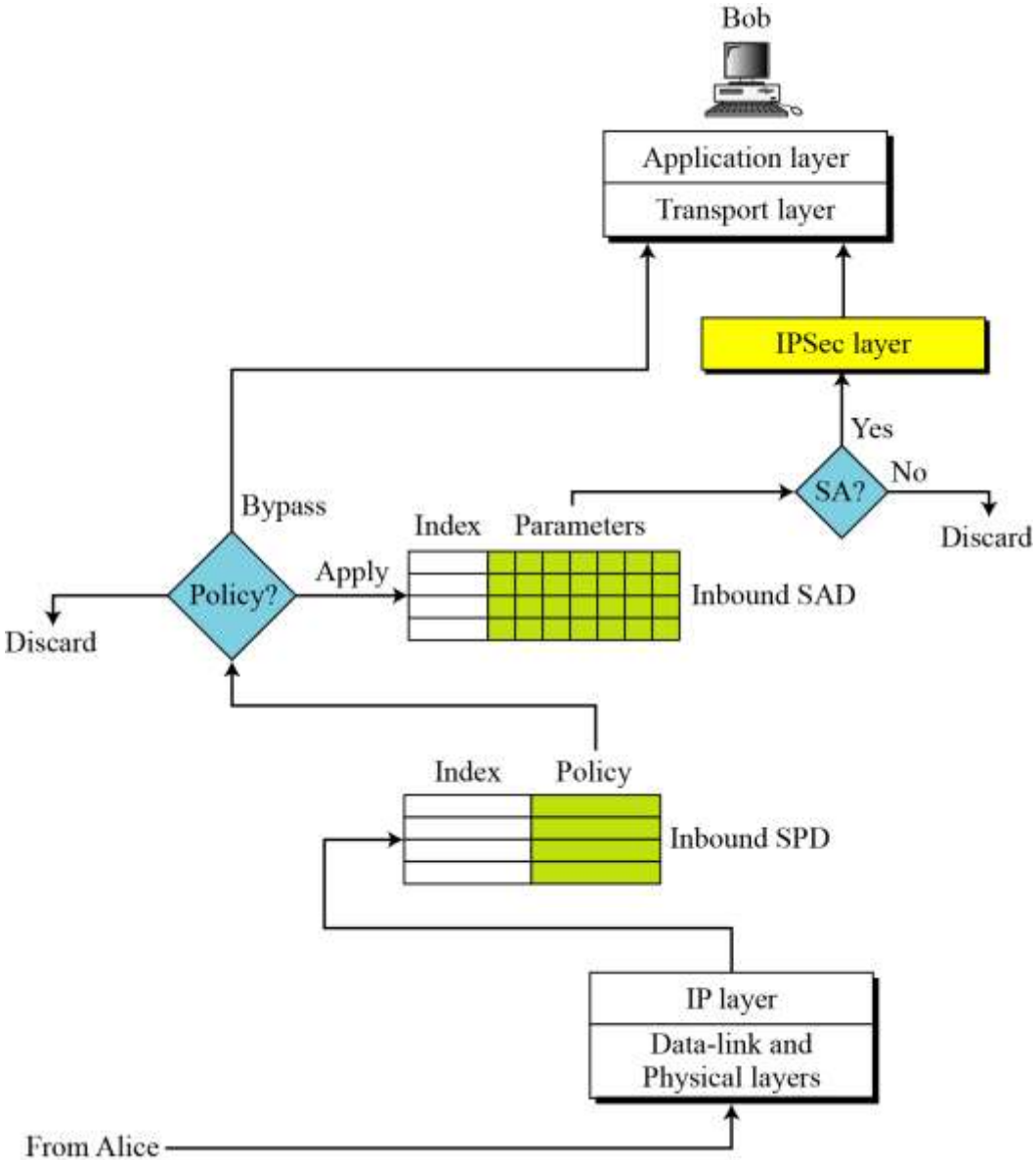
18.4.1 (Continued)

Figure 18.13 *Outbound processing*



18.4.1 (Continued)

Figure 18.14 *Inbound processing*



18-5 INTERNET KEY EXCHANGE (IKE)

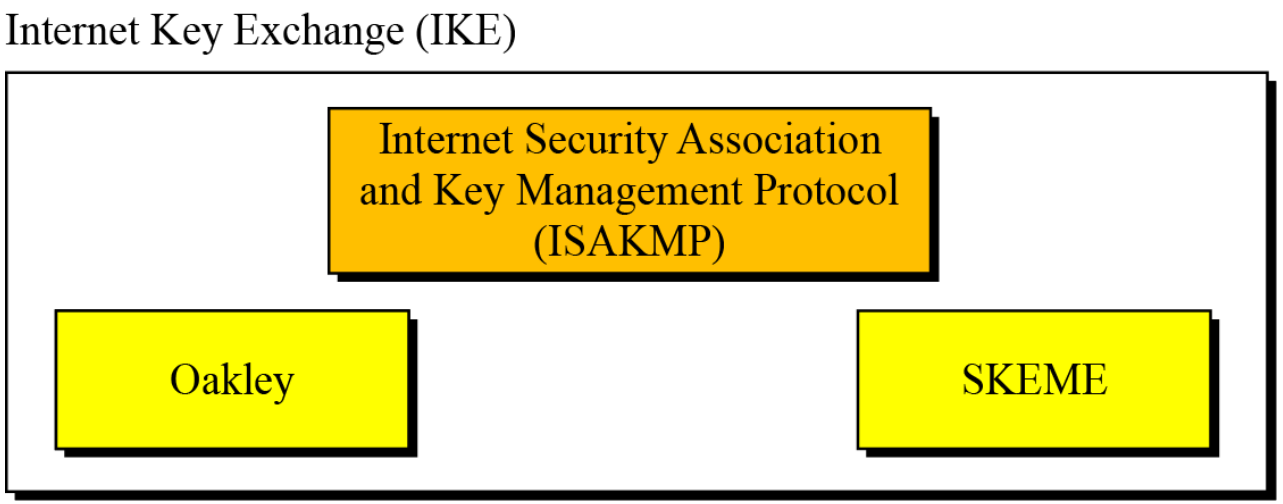
The Internet Key Exchange (IKE) is a protocol designed to create both inbound and outbound Security Associations.

Topics discussed in this section:

Note

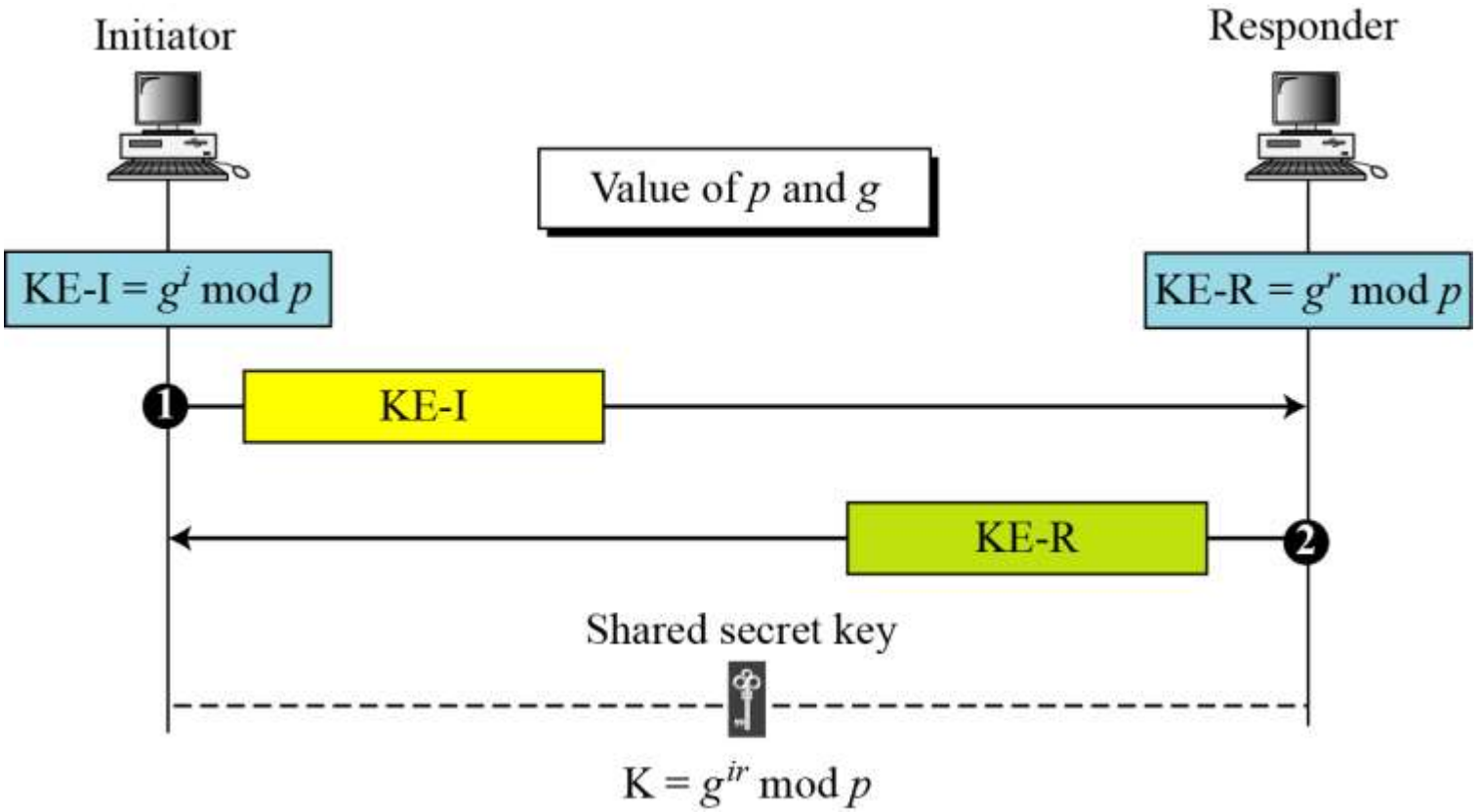
IKE creates SAs for IPSec.

Figure 18.15 *IKE components*



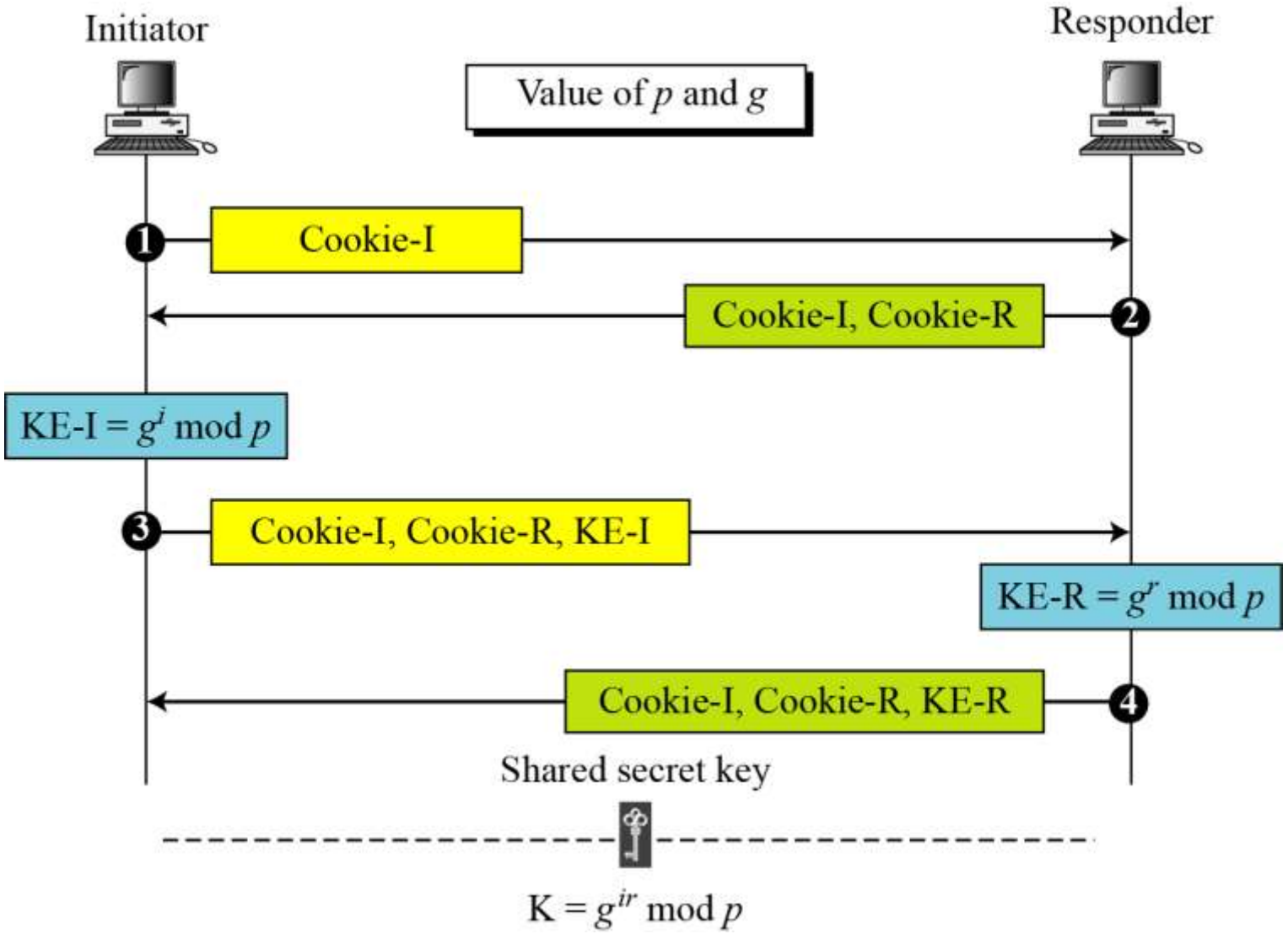
18.5.1 Improved Diffie-Hellman

Figure 18.16 *Diffie-Hellman key exchange*



18.5.1 (Continued)

Figure 18.17 *Diffie-Hellman with cookies*



18.5.1 Continued

Note

To protect against a clogging attack, IKE uses cookies.

Note

To protect against a replay attack, IKE uses nonces.

Note

To protect against man-in-the-middle attack, IKE requires that each party shows that it possesses a secret.

18.5.2 IKE Phases

Note

IKE is divided into two phases: phase I and phase II. Phase I creates SAs for phase II; phase II creates SAs for a data exchange protocol such as IPSec..

18.5.3 Phases and Modes

Figure 18.18 *IKE Phases*

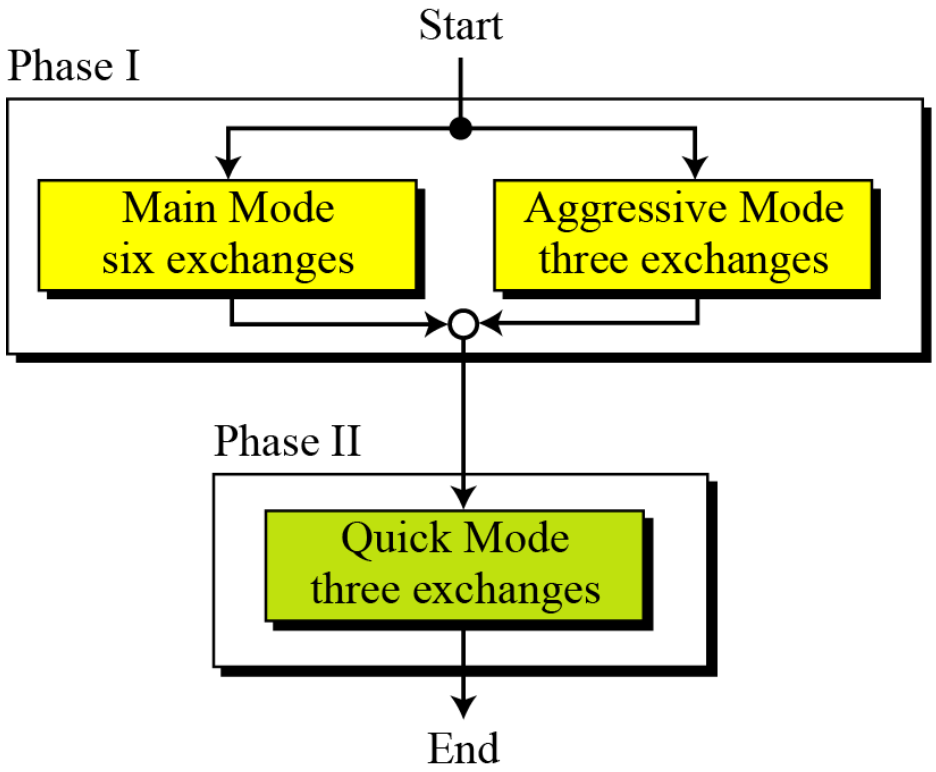
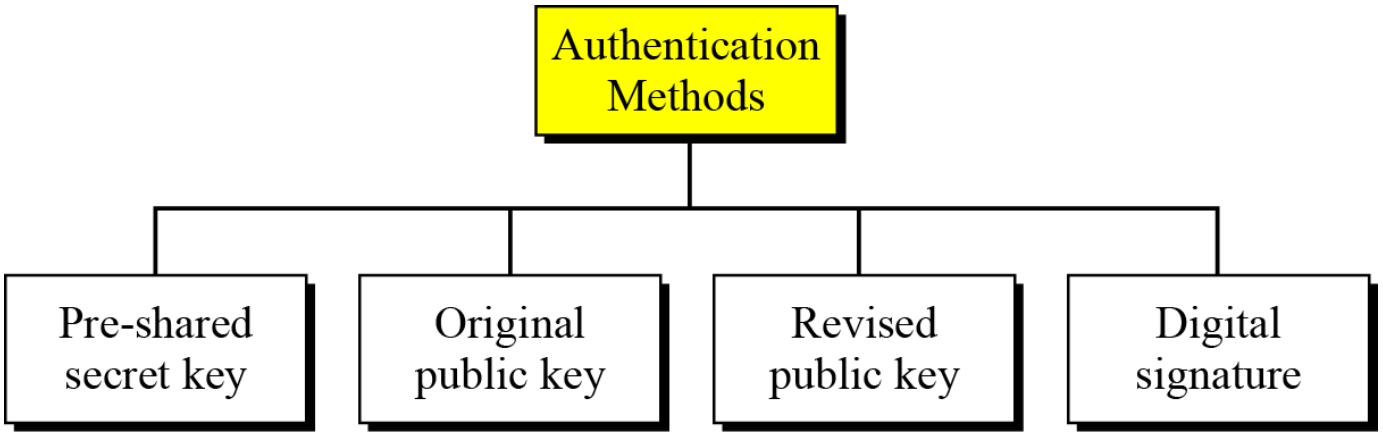


Figure 18.19 *Main-mode or aggressive-mode methods*

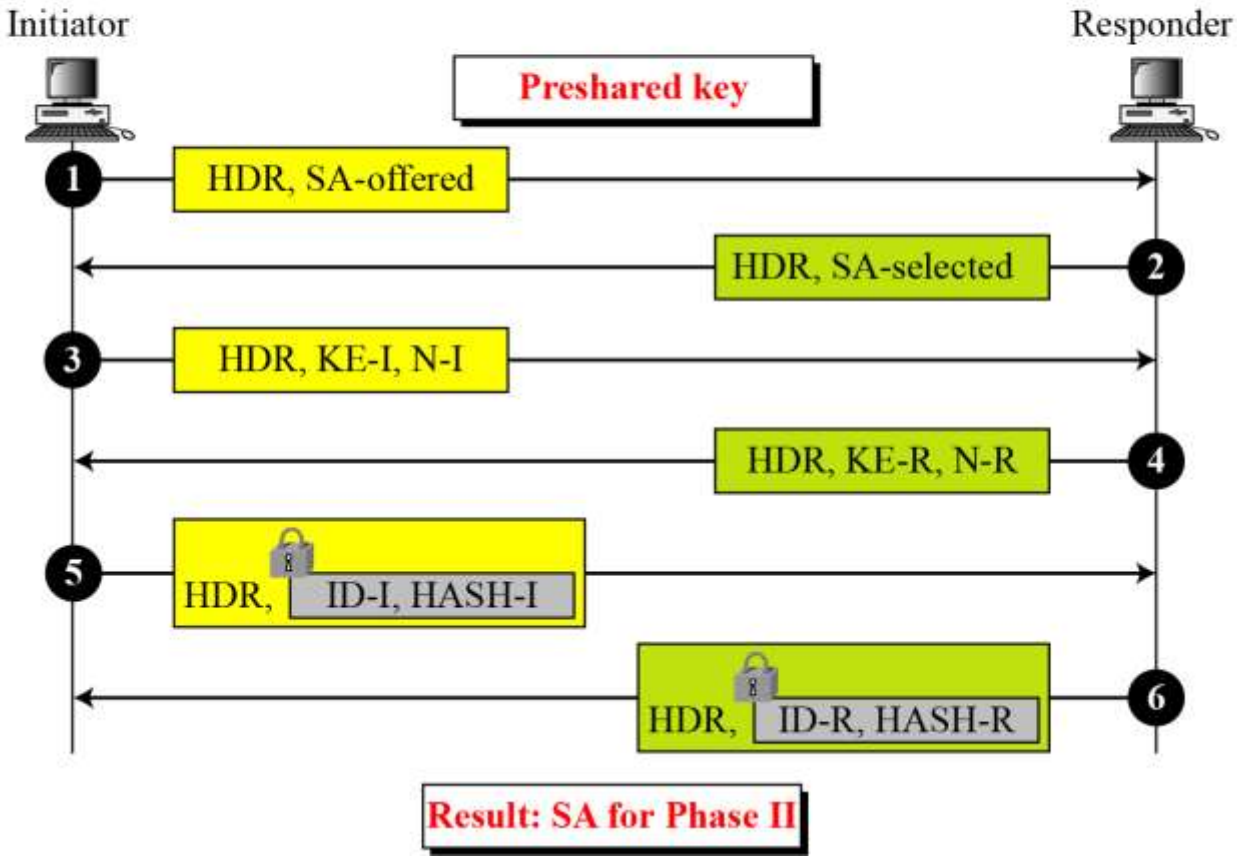


18.5.4 Phase I: Main Mode

Figure 18.20 *Main mode, preshared secret-key method*

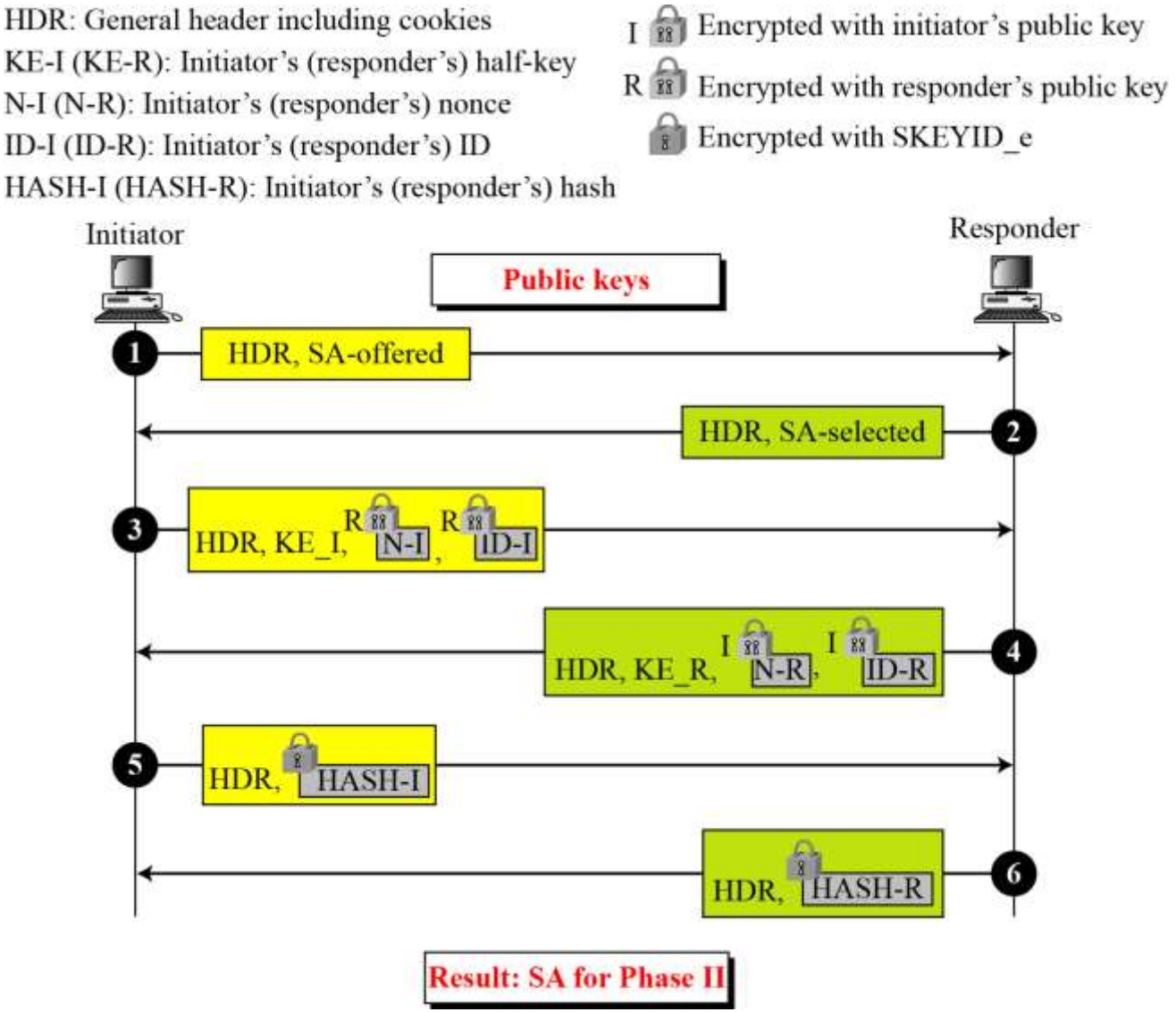
KE-I (KE-R): Initiator's (responder's) half-key
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's) hash

HDR: General header including cookies
Encrypted with SKEYID_e



18.5.4 (Continued)

Figure 18.21 *Main mode, original public-key method*



18.5.4 (Continued)

Figure 18.22 *Main mode, revised public-key method*

HDR: General header including cookies
KE-I (KE-R): Initiator's (responder's) half-key
Cert-I (Cert-R): Initiator's (responder's) certificate
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's) hash

I Encrypted with initiator's public key
R Encrypted with responder's public key
R Encrypted with responder's secret key
I Encrypted with initiator's secret key
Encrypted with SKEYID_e

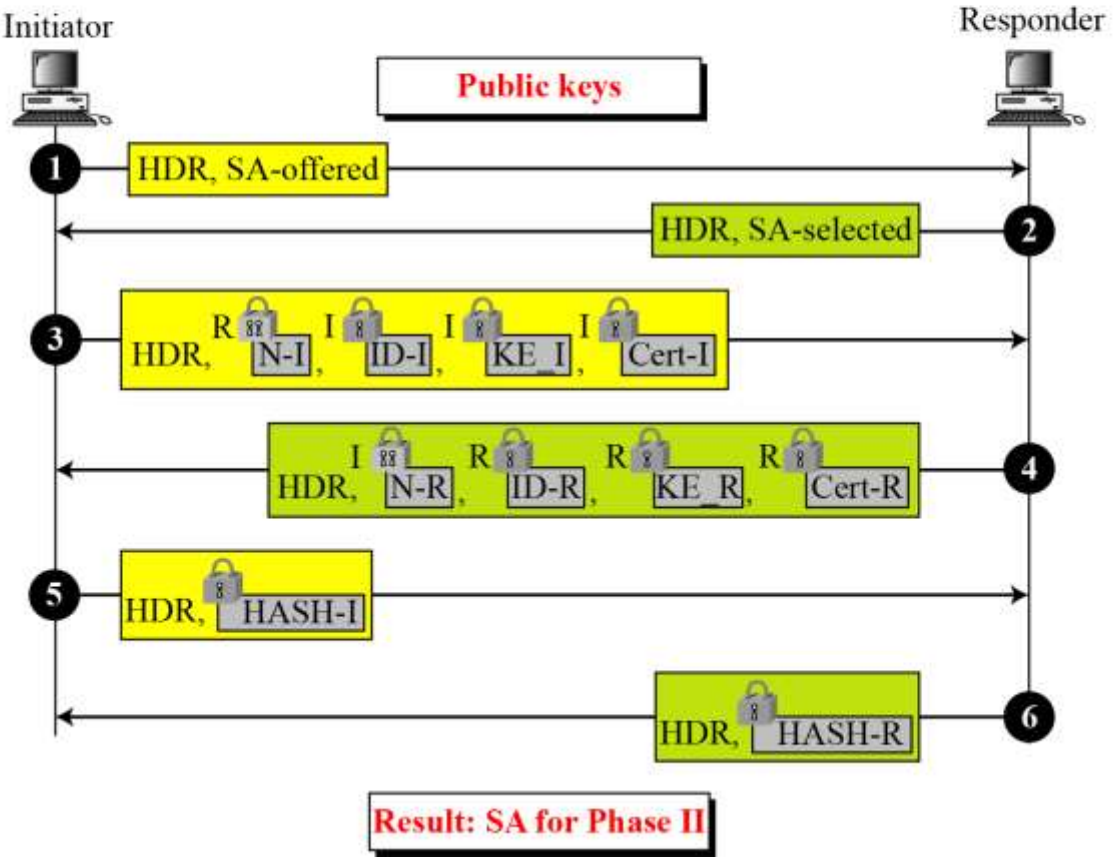
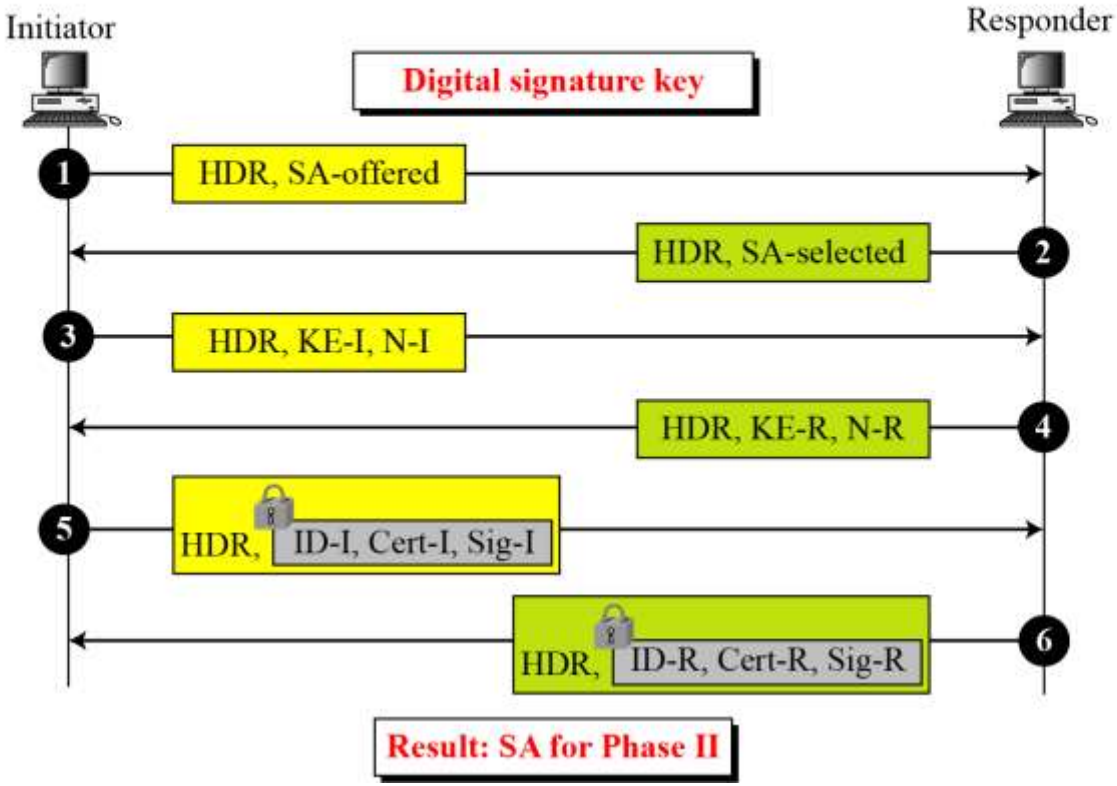


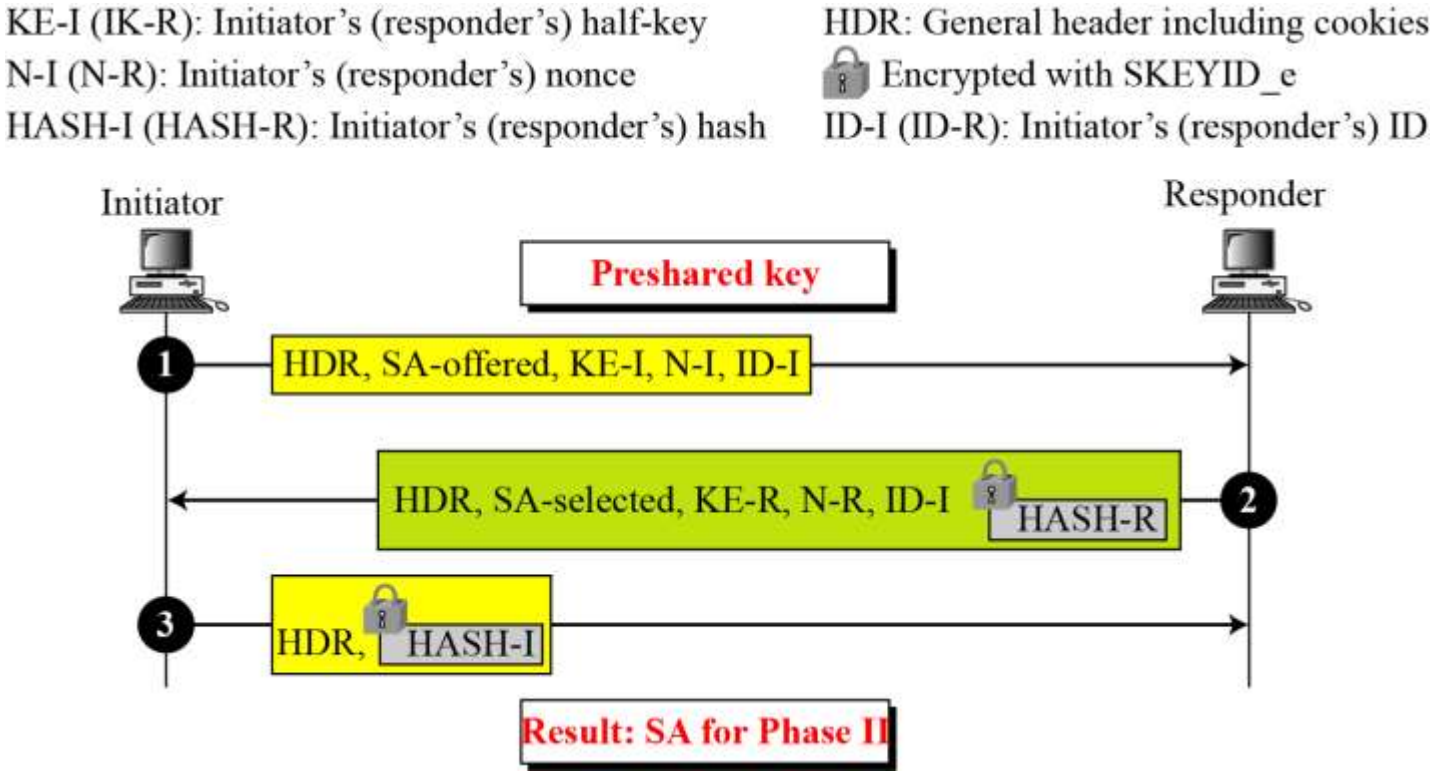
Figure 18.23 *Main mode, digital signature method*

HDR: General header including cookies
Sig-I: Initiator's signature on messages 1-4
Sig-R: Initiator's signature on messages 1-5
Cert-I (Cert-R): Initiator's (responder's) certificate
N-I (N-R): Initiator's (responder's) nonce
KE-I (KE-R): Initiator's (responder's) half-key
ID-I (ID-R): Initiator's (responder's) ID
Encrypted with SKEYID_e



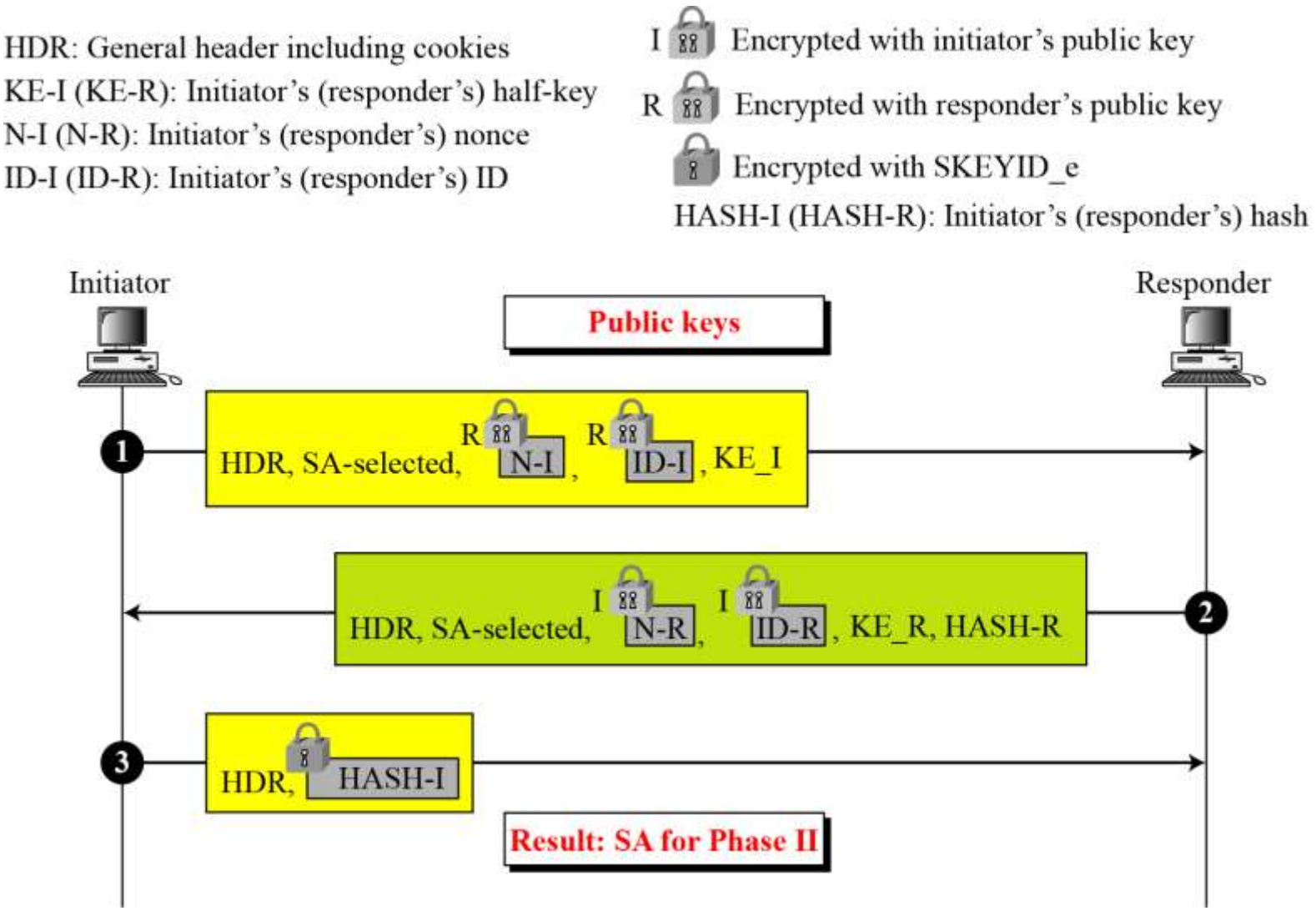
18.5.5 Phase I: Aggressive Mode

Figure 18.24 *Aggressive mode, preshared-key method*



18.5.5 (Continued)

Figure 18.25 *Aggressive mode, original public-key method*



18.5.5 (Continued)

Figure 18.26 *Aggressive mode, revised public-key method*

HDR: General header including cookies
KE-I (KE-R): Initiator's (responder's) half-key
Cert-I (Cert-R): Initiator's (responder's) certificate
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's) hash

I Encrypted with initiator's public key
R Encrypted with responder's public key
R Encrypted with responder's secret key
I Encrypted with initiator's secret key
Encrypted with SKEYID_e

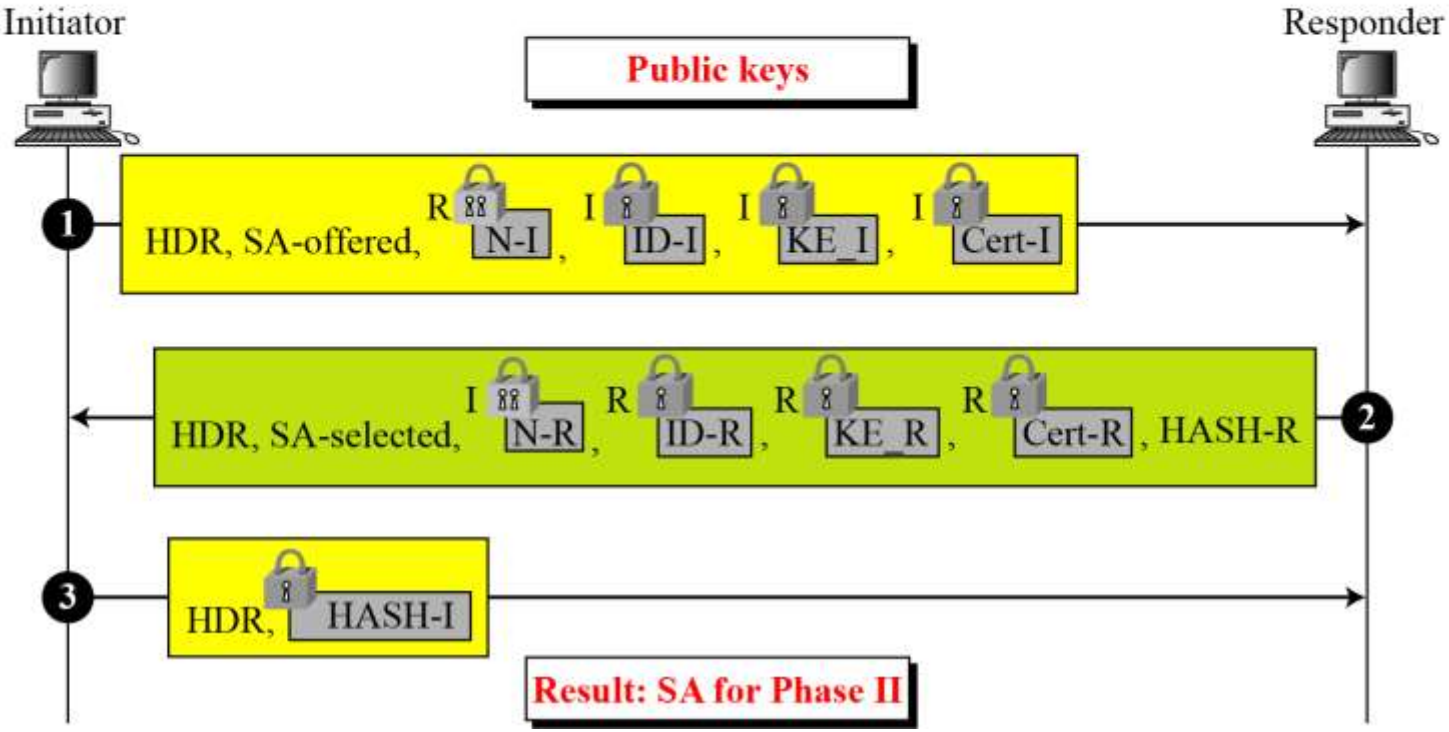
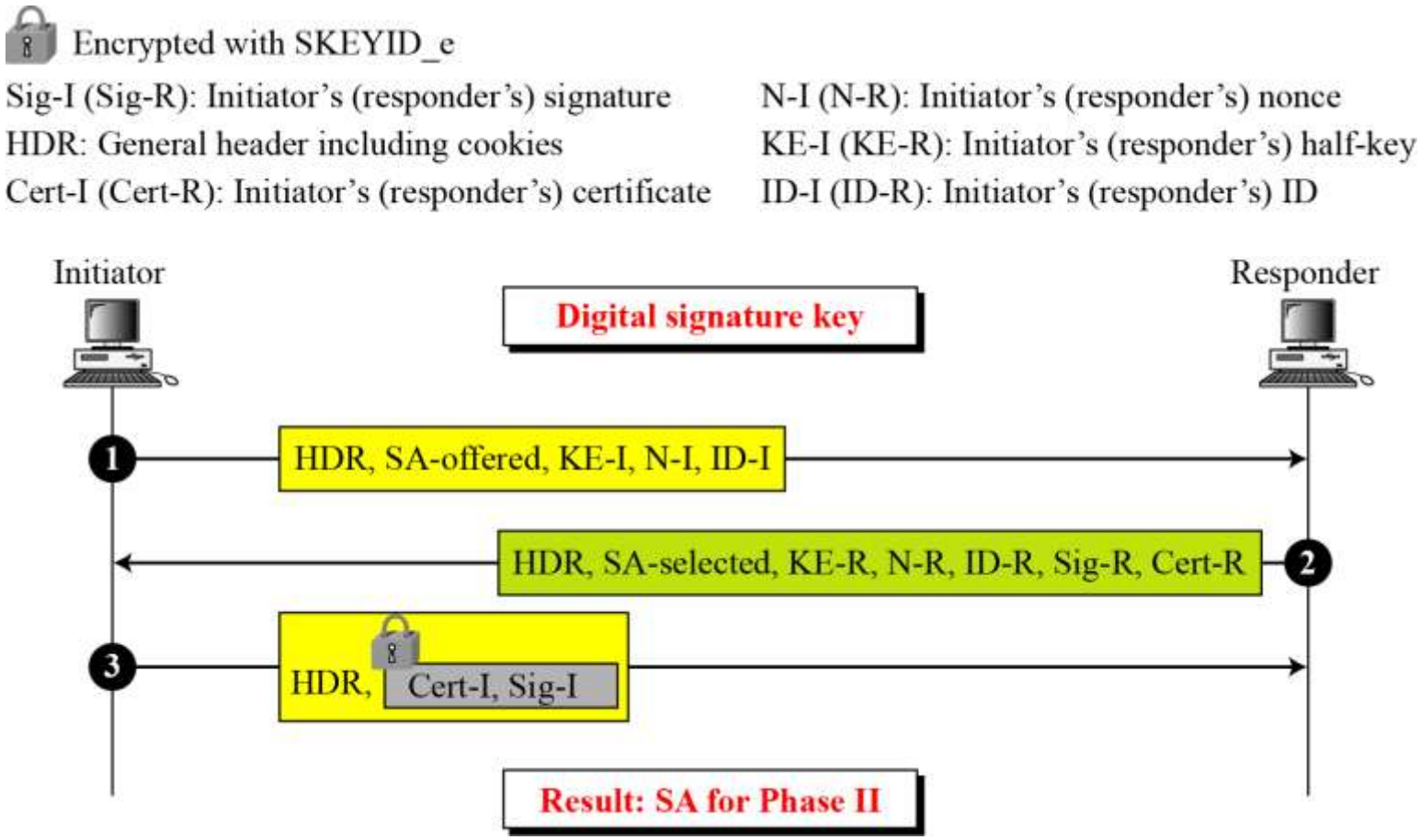
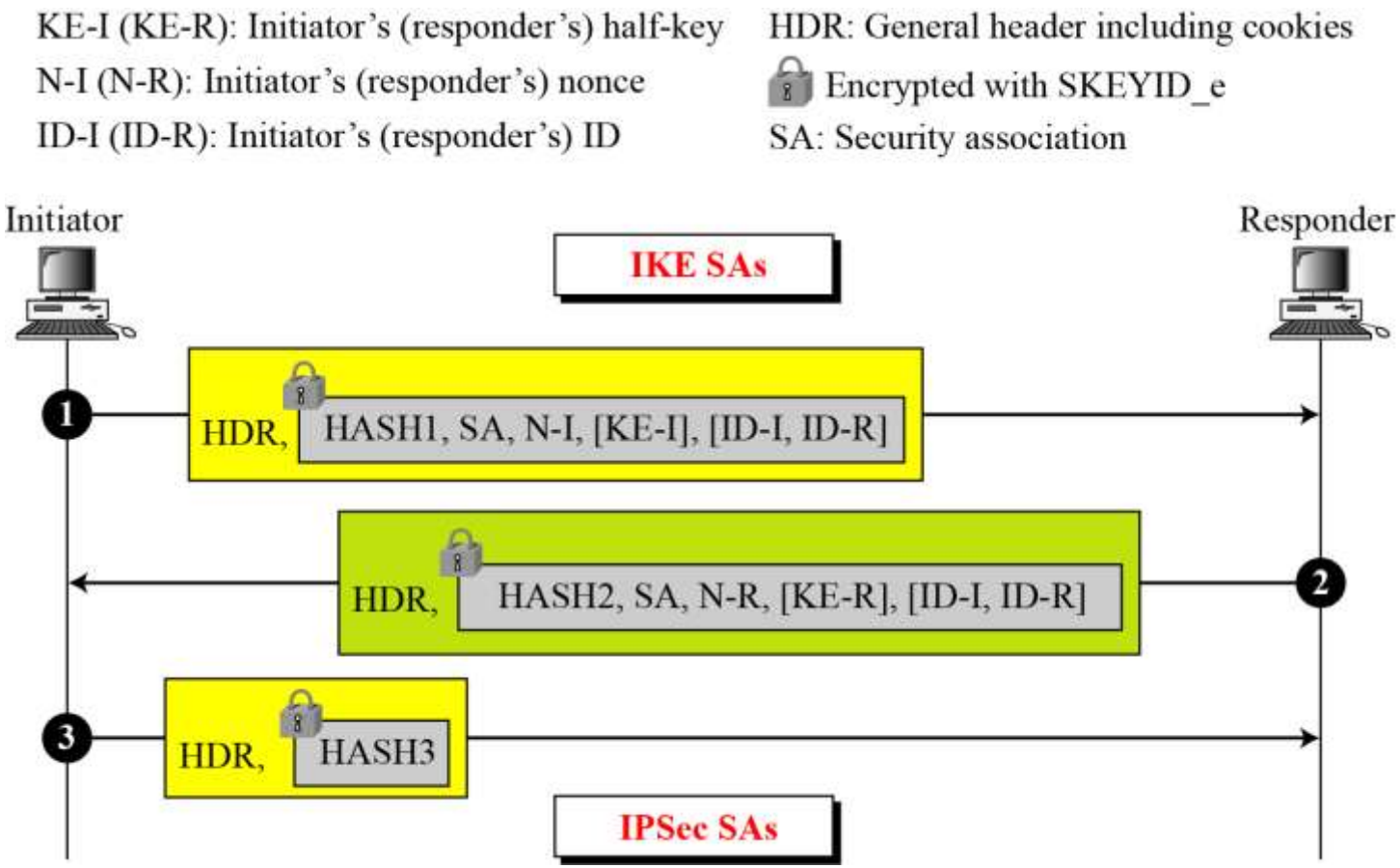


Figure 18.27 *Aggressive mode, digital signature method*



18.5.6 Phase II: Quick Mode

Figure 18.28 *Quick mode*



18.5.7 SA Algorithms

Table 18.3 *Diffie-Hellman groups*

Value	*Description*
1	Modular exponentiation group with a 768-bit modulus
2	Modular exponentiation group with a 1024-bit modulus
3	Elliptic curve group with a 155-bit field size
4	Elliptic curve group with a 185-bit field size
5	Modular exponentiation group with a 1680-bit modulus

18.5.7 Continued

Table 18.4 *Hash Algorithms*

Value	*Description*
1	MD5
2	SHA
3	Tiger
4	SHA2-256
5	SHA2-384
6	SHA2-512

Table 18.5 *Encryption algorithms*

Value	*Description*
1	DES
2	IDEA
3	Blowfish
4	RC5
5	3DES
6	CAST
7	AES

18-6 ISKAMP

The ISAKMP protocol is designed to carry messages for the IKE exchange.

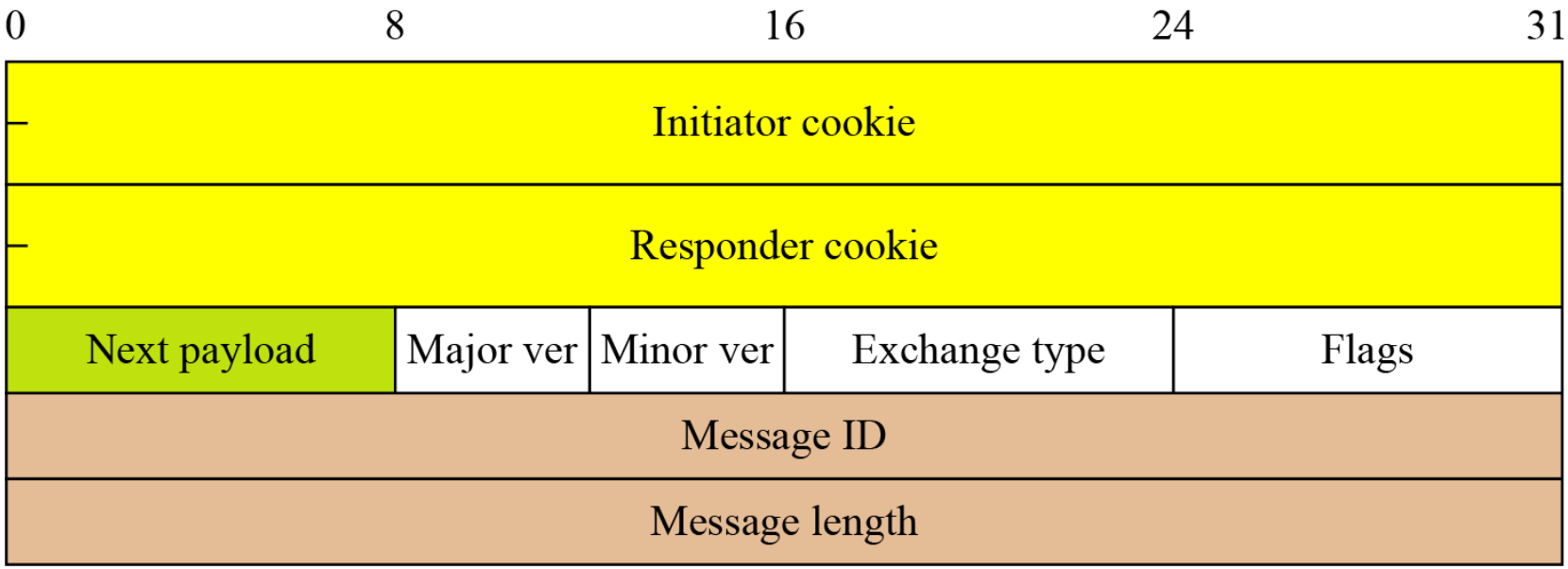
Topics discussed in this section:

18.6.1 General Header

18.6.2 Payloads

18.6.1 General Header

Figure 18.29 *ISAKMP general header*



18.6.2 Payloads

Table 18.6 *Payloads*

Types	*Name*	*Brief Description*
0	None	Used to show the end of the payloads
1	SA	Used for starting the negotiation
2	Proposal	Contains information used during SA negotiation
3	Transform	Defines a security transform to create a secure channel
4	Key Exchange	Carries data used for generating keys
5	Identification	Carries the identification of communication peers
6	Certification	Carries a public-key certificate
7	Certification Request	Used to request a certificate from the other party
8	Hash	Carries data generated by a hash function
9	Signature	Carries data generated by a signature function
10	Nonce	Carries randomly generated data as a nonce
11	Notification	Carries error message or status associated with an SA
12	Delete	Carries one more SA that the sender has deleted
13	Vendor	Defines vendor-specification extensions

18.6.2 (Continued)

Figure 18.30 *Generic payload header*

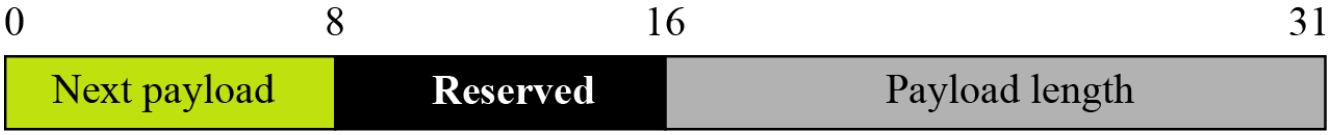
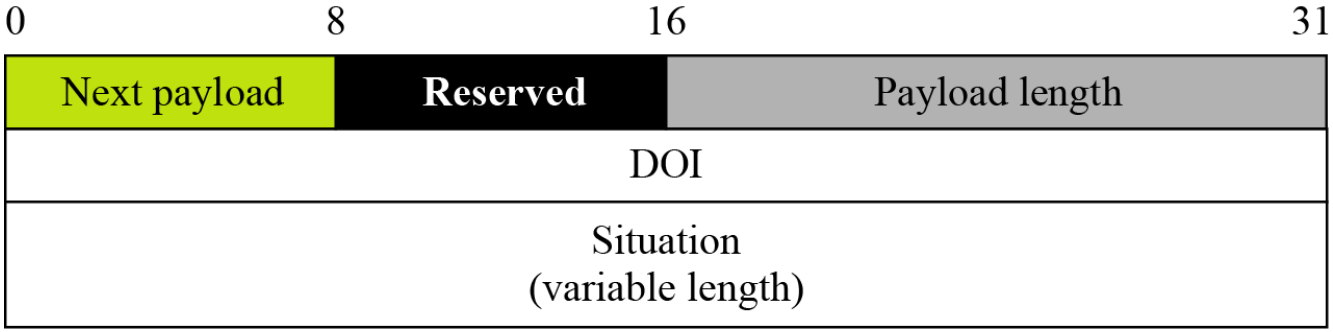


Figure 18.31 *SA payload*



DOI - Domain of Interpretation

Figure 18.32 *Proposal payload*

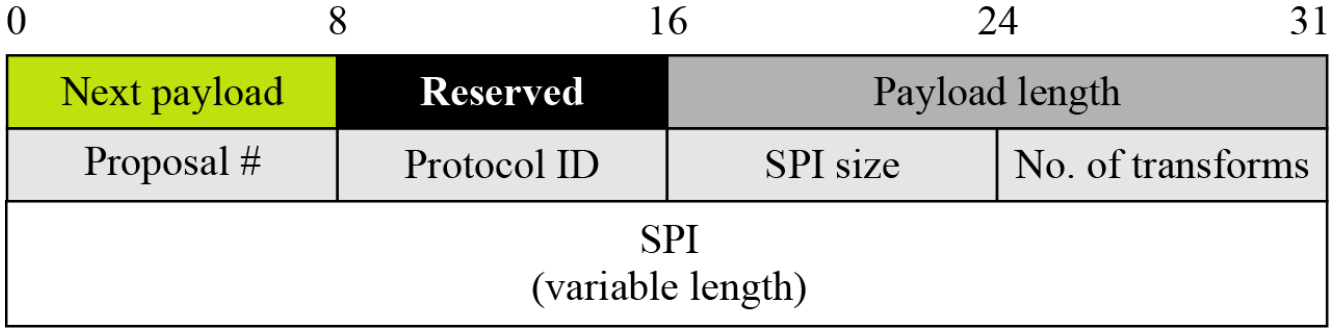


Figure 18.33 *Transform payload*

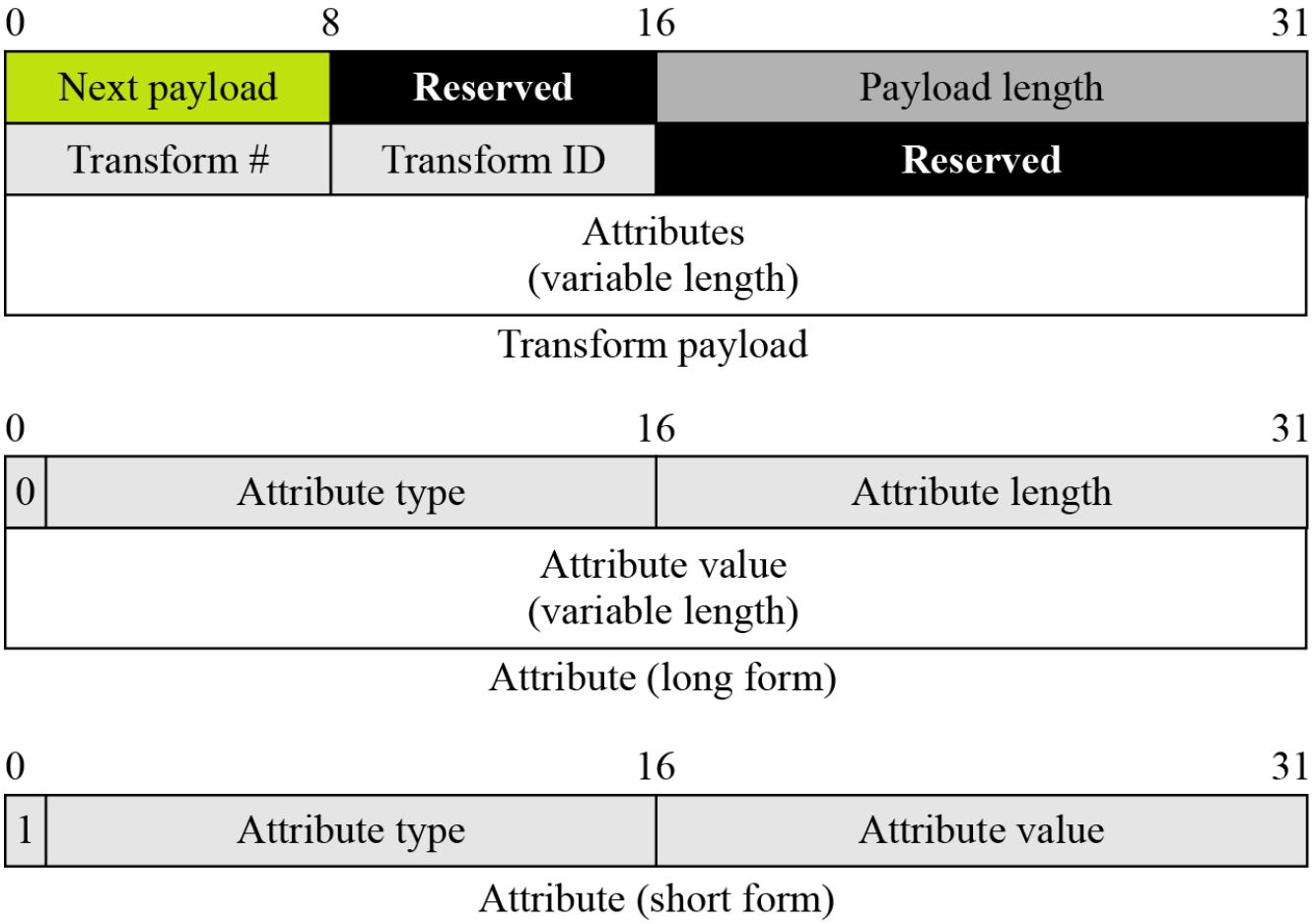


Figure 18.34 *Key-exchange payload*

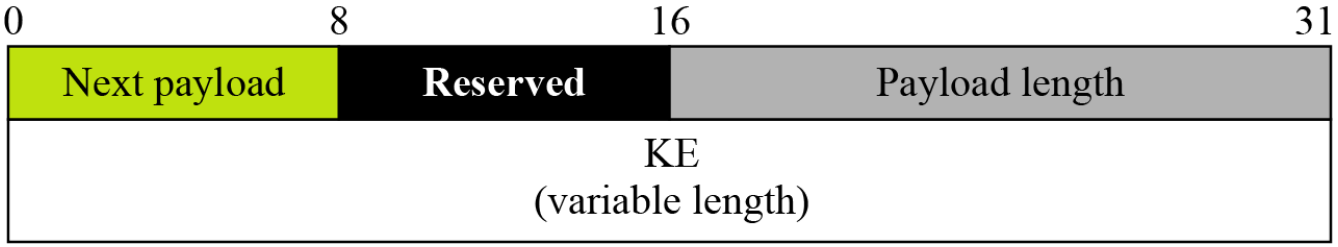


Figure 18.35 *Identification payload*

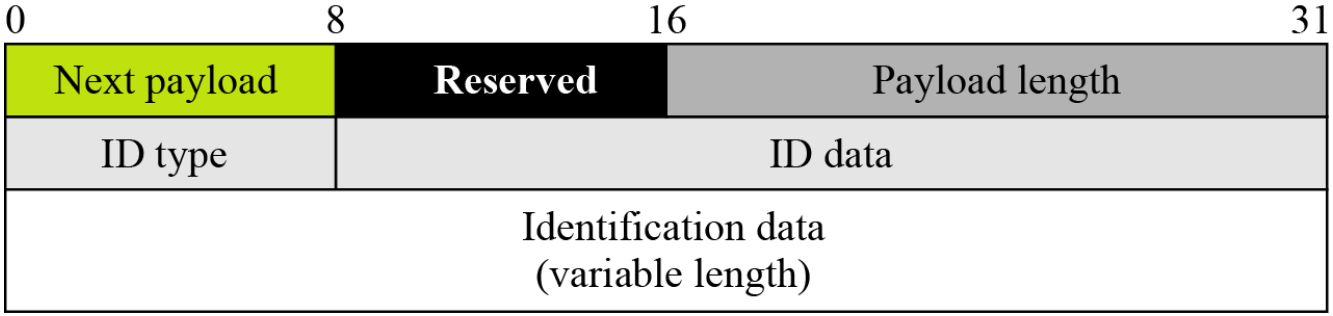


Figure 18.36 *Certification payload*

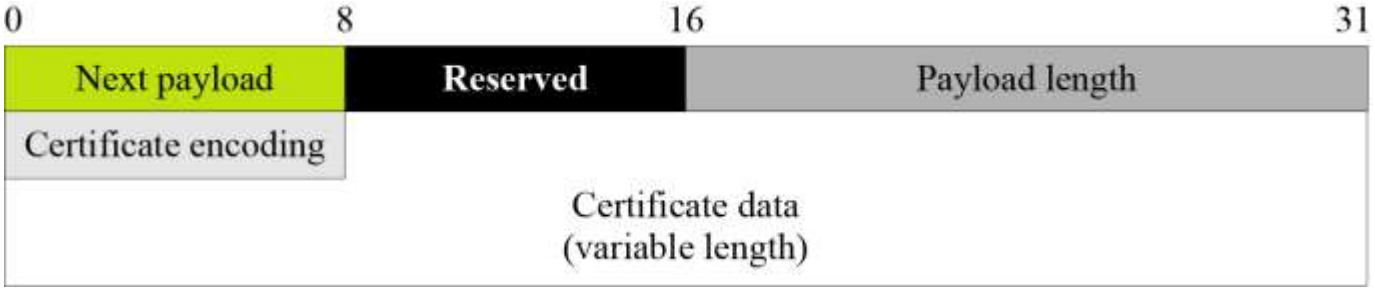


Table 18.7 *Certification types*

Value	*Type*
0	None
1	Wrapped X.509 Certificate
2	PGP Certificate
3	DNS Signed Key
4	X.509 Certificate —Signature
5	X.509 Certificate—Key Exchange
6	Kerberos Tokens
7	Certification Revocation List
8	Authority Revocation List
9	SPKI Certificate
10	X.509 Certificate—Attribute

Figure 18.37 *Certification request payload*

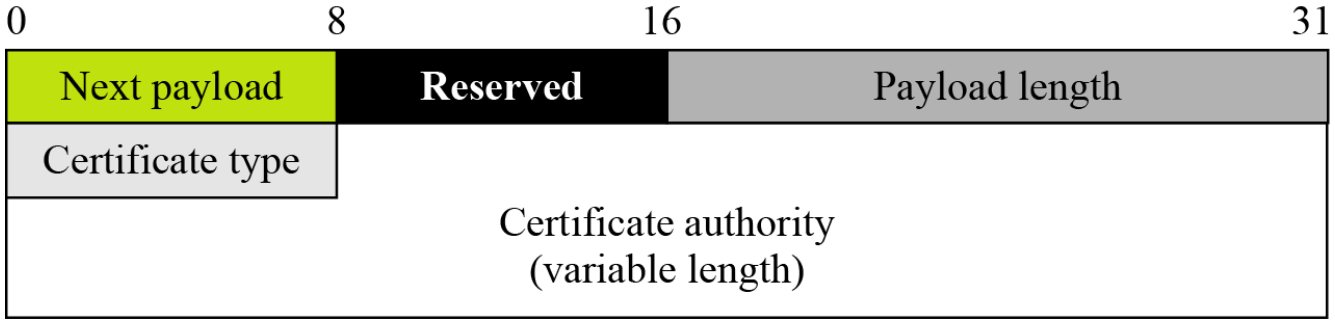


Figure 18.38 *Hash payload*

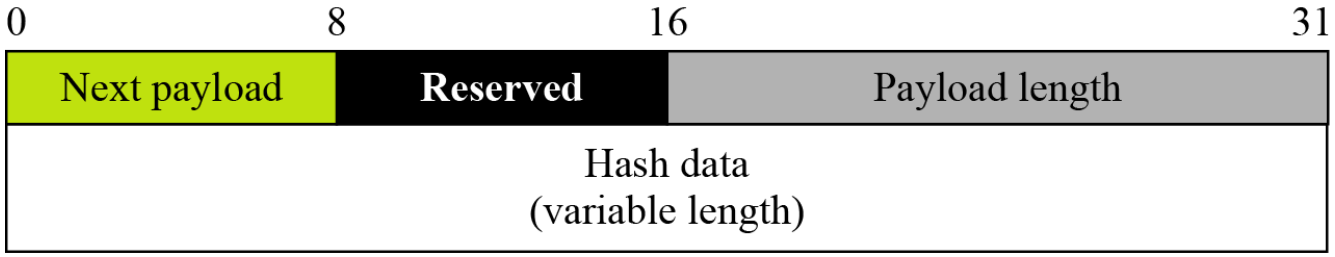


Figure 18.39 *Signature payload*

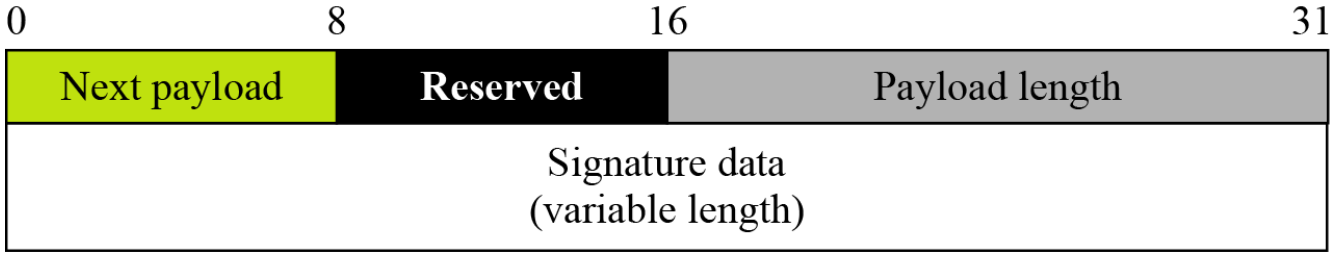


Figure 18.40 *Nonce payload*

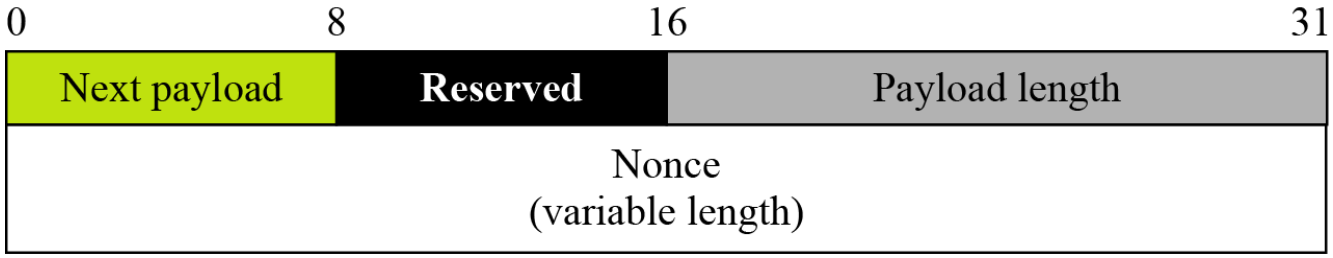


Figure 18.41 *Notification payload*

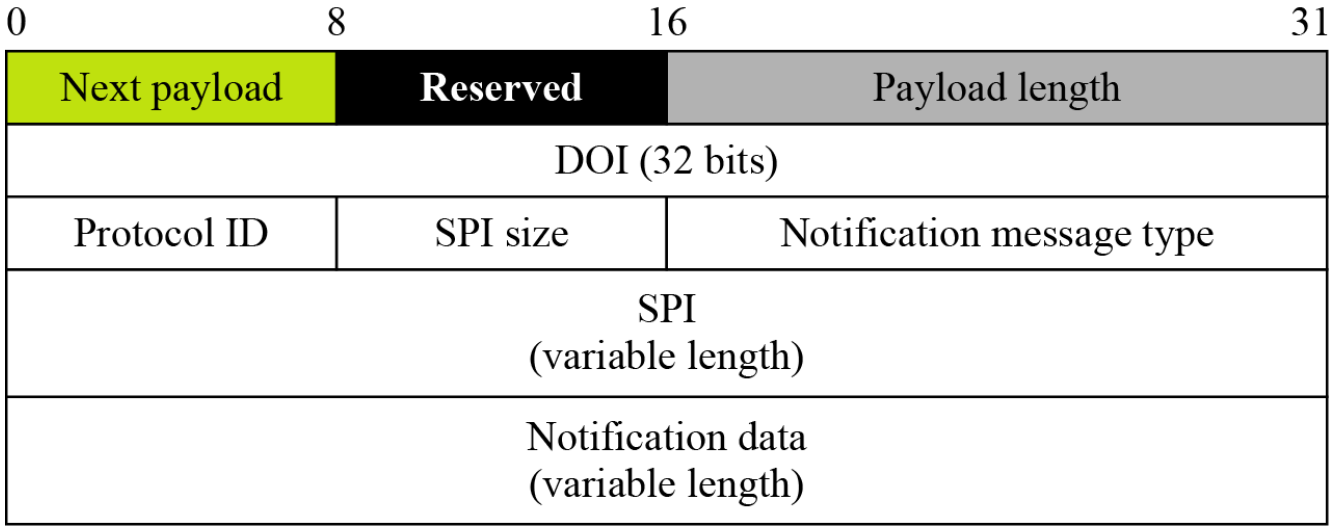


Table 18.8 *Notification types*

Value	*Description*	*Value*	*Description*
1	INVALID-PAYLOAD-TYPE	8	INVALID-FLAGS
2	DOI-NOT-SUPPORTED	9	INVALID-MESSAGE-ID
3	SITUATION-NOT-SUPPORTED	10	INVALID-PROTOCOL-ID
4	INVALID-COOKIE	11	INVALID-SPI
5	INVALID-MAJOR-VERSION	12	INVALID-TRANSFORM-ID
6	INVALID-MINOR-VERSION	13	ATTRIBUTE-NOT-SUPPORTED
7	INVALID-EXCHANGE-TYPE	14	NO-PROPOSAL-CHOSEN

Table 18.8 *Notification types (Continued)*

Value	*Description*	*Value*	*Description*
15	BAD-PROPOSAL-SYNTAX	23	INVALID-HASH-INFORMATION
16	PAYLOAD-MALFORMED	24	AUTHENTICATION-FAILED
17	INVALID-KEY-INFORMATION	25	INVALID-SIGNATURE
18	INVALID-ID-INFORMATION	26	ADDRESS-NOTIFICATION
19	INVALID-CERT-ENCODING	27	NOTIFY-SA-LIFETIME
20	INVALID-CERTIFICATE	28	CERTIFICATE-UNAVAILABLE
21	CERT-TYPE-UNSUPPORTED	29	UNSUPPORTED EXCHANGE-TYPE
22	INVALID-CERT-AUTHORITY	30	UNEQUAL-PAYLOAD-LENGTHS

Table 18.9 *Status notification values*

Value	*Description*
16384	CONNECTED
24576-32767	DOI-specific codes

Figure 18.42 *Delete payload*

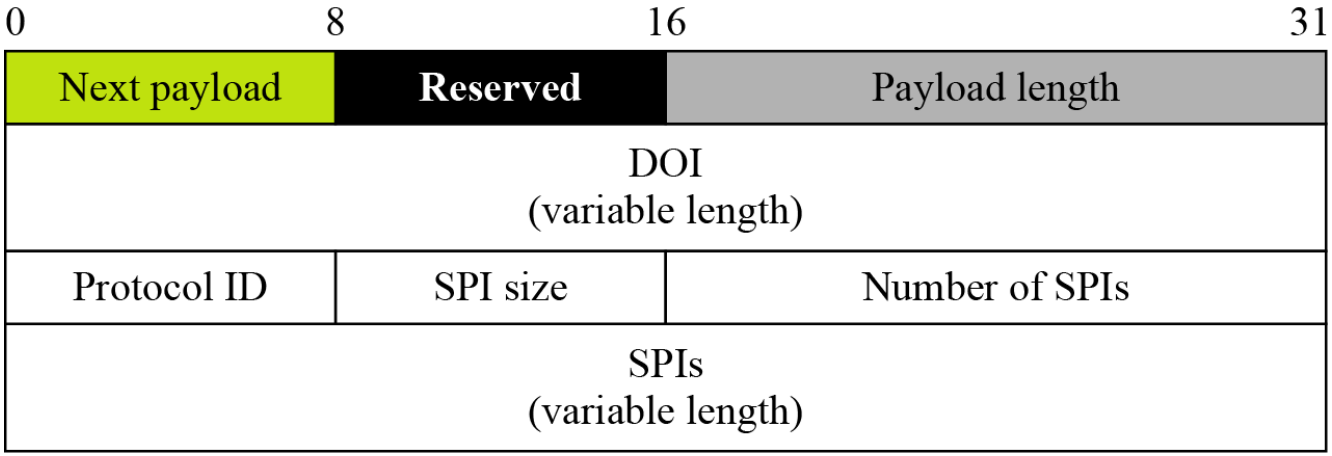


Figure 18.43 *Vendor payload*