1

IP Security Architecture

1. **Definition**: IPSec is a framework of protocols designed to secure data transmitted over IP networks by providing encryption, authentication, and integrity.
2. **Key Components**:
    - **Authentication Header (AH)**: Verifies the integrity and authenticity of the data.
    - **Encapsulating Security Payload (ESP)**: Encrypts data for confidentiality and can also provide authentication.
3. **Standards**:
    - Defined by several **RFCs**, including:
        - RFC 2401: Overview of IPSec architecture.
        - RFC 2402: Packet authentication.
        - RFC 2406: Packet encryption.
        - RFC 2408: Key management.
4. **Implementation**: IPSec works as **extension headers** in IP packets, added after the main IP header. It is mandatory in IPv6 and optional in IPv4.
5. **Services**: IPSec ensures:
    - **Confidentiality**: Protects data from unauthorized access.
    - **Integrity**: Ensures data is not altered during transmission.
    - **Authentication**: Verifies the sender's identity.
    - **Key Management**: Safely generates and exchanges cryptographic keys.

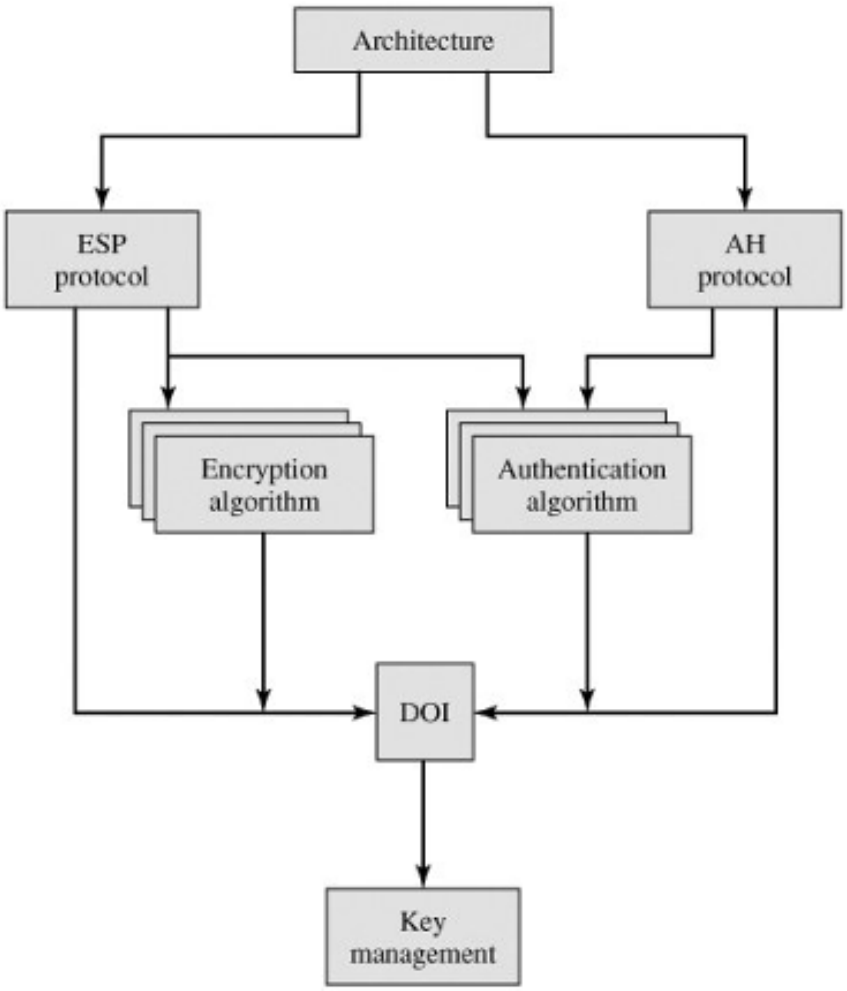
- **Architecture:** Covers the general concepts, security requirements, definitions, and mechanisms defining IPSec technology.

[Page 489]

- **Encapsulating Security Payload (ESP):** Covers the packet format and general issues related to the use of the ESP for packet encryption and, optionally, authentication.
- **Authentication Header (AH):** Covers the packet format and general issues related to the use of AH for packet authentication.
- **Encryption Algorithm:** A set of documents that describe how various encryption algorithms are used for ESP.
- **Authentication Algorithm:** A set of documents that describe how various authentication algorithms are used for AH and for the authentication option of ESP.
- **Key Management:** Documents that describe key management schemes.
- **Domain of Interpretation (DOI):** Contains values needed for the other documents to relate to each other. These include identifiers for approved encryption and authentication algorithms, as well as operational parameters such as key lifetime.

Figure 16.2. IPSec Document Overview

(This item is displayed on page 488 in the print version)



2

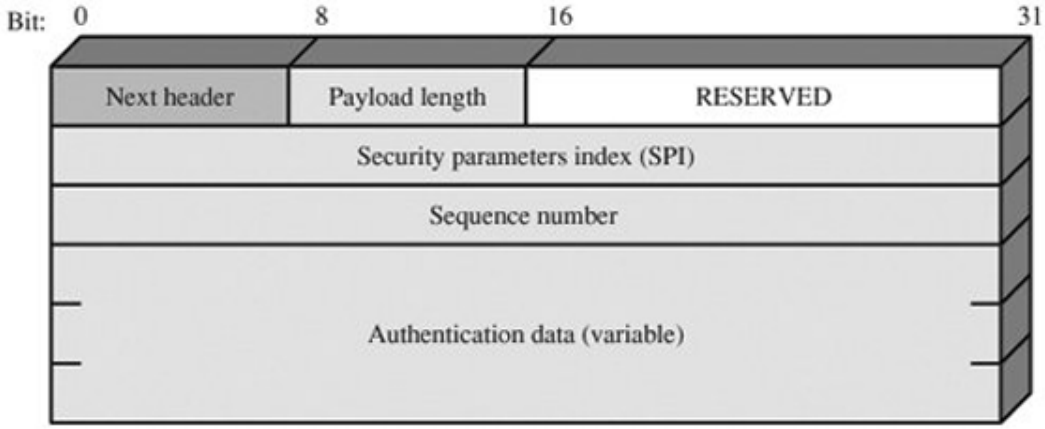
IPSec Authentication Header (AH) - Key Points for 5 Marks

1. **Purpose**:
    - Ensures **data integrity** (no changes during transit) and **authentication** (verifies sender identity).
    - Protects against attacks like **address spoofing** and **replay attacks**.
2. **Structure**:
    - **Next Header**: Identifies the type of header following AH.
    - **Payload Length**: Indicates the size of AH.
    - **Reserved**: Space for future use.
    - **Security Parameters Index (SPI)**: Identifies the security association.
    - **Sequence Number**: Prevents replay attacks by maintaining a unique counter for each packet.
    - **Authentication Data**: Contains a Message Authentication Code (MAC) to verify packet integrity.
3. **Replay Attack Protection**:
    - Sequence numbers prevent duplicate packets from being reused by attackers.
    - A sliding **window mechanism** (default size 64) tracks valid sequence numbers to detect and discard duplicates.
4. **Key Requirement**:
    - Both sender and receiver must share a **secret key** to generate and verify the MAC.
5. **Significance**:
    - AH ensures packets are not tampered with, verifies their origin, and protects against packet duplication, enhancing overall network security.

he Authentication Header consists of the following fields (Figure 16.3):

- **Next Header (8 bits):** Identifies the type of header immediately following this header.
- **Payload Length (8 bits):** Length of Authentication Header in 32-bit words, minus 2. For example, the default length of the authentication data field is 96 bits, or three 32-bit words. With a three-word fixed header, there are a total of six words in the header, and the Payload Length field has a value of 4.
- **Reserved (16 bits):** For future use.
- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** A monotonically increasing counter value, discussed later.
- **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value (ICV), or MAC, for this packet, discussed later.

Figure 16.3. IPSec Authentication Header



3

Transport and Tunnel Modes

Both AH and ESP support two modes of use: transport and tunnel mode. The operation of these two modes is best understood in the context of a description of AH and ESP, which are covered in Sections 16.3 and 16.4, respectively. Here we provide a brief overview.

Transport Mode

Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet. Examples include a TCP or UDP segment or an ICMP packet, all of which operate directly above IP in a host protocol stack. Typically, transport mode is used for end-to-end communication between two hosts (e.g., a client and a server, or two workstations). When a host runs AH or ESP over IPv4, the payload is the data that normally follow the IP header. For IPv6, the payload is the data that normally follow both the IP header and any IPv6 extensions headers that are present, with the possible exception of the destination options header, which may be included in the protection.

ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header. AH in transport mode authenticates the IP payload and selected portions of the IP header.

Tunnel Mode

Tunnel mode provides protection to the entire IP packet. To achieve this, after the AH or ESP fields are

---

added to the IP packet, the entire packet plus security fields is treated as the payload of new "outer" IP packet with a new outer IP header. The entire original, or inner, packet travels through a "tunnel" from one point of an IP network to another; no routers along the way are able to examine the inner IP header. Because the original packet is encapsulated, the new, larger packet may have totally different source and destination addresses, adding to the security. Tunnel mode is used when one or both ends of an SA are a security gateway, such as a firewall or router that implements IPSec. With tunnel mode, a number of hosts on networks behind firewalls may engage in secure communications without implementing IPSec. The unprotected packets generated by such hosts are tunneled through external networks by tunnel mode SAs set up by the IPSec software in the firewall or secure router at the boundary of the local network.

4

Encapsulating Security Payload (ESP) - Key Points for 5 Marks

1. **Purpose**:
    - ESP provides **data confidentiality** (encrypting the content of packets) and optional **authentication** (verifying the source and integrity of packets).
    - It also offers **limited traffic flow confidentiality** by hiding the actual size of the data.
2. **ESP Packet Structure**:
    - **Security Parameters Index (SPI)**: Identifies the security association.

- **Sequence Number**: Prevents replay attacks by ensuring a unique counter for each packet.
- **Payload Data**: Encrypted transport-level data or an entire IP packet, depending on the mode (Transport or Tunnel).
- **Padding & Pad Length**: Aligns data for encryption and conceals actual payload length.
- **Next Header**: Indicates the type of data in the payload (e.g., TCP, UDP).
- **Authentication Data**: Verifies packet integrity using a Message Authentication Code (MAC).

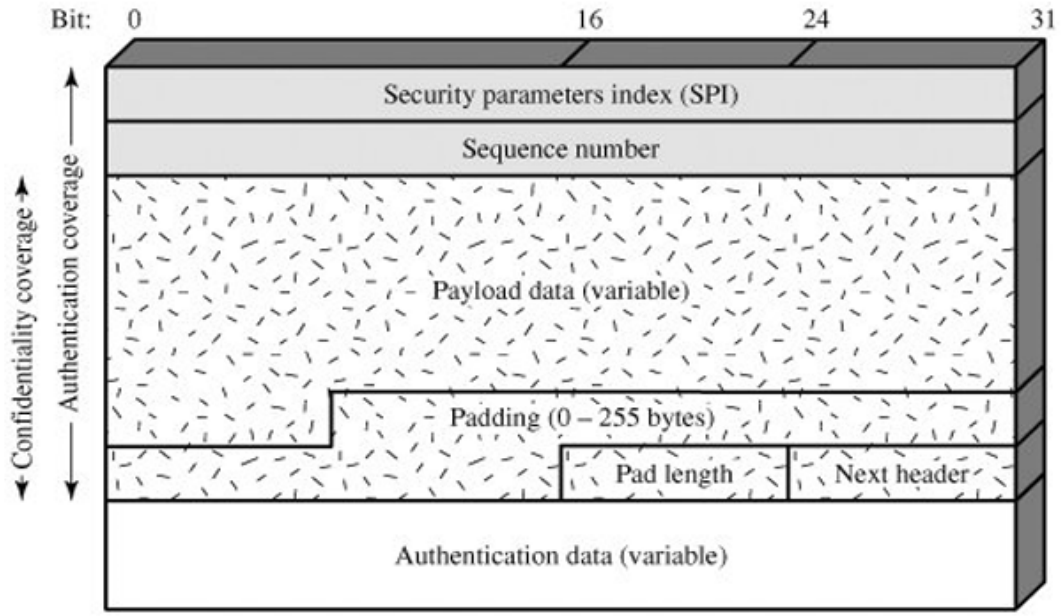
3. **Encryption and Authentication**:
   - Payload data and some fields (Padding, Pad Length, and Next Header) are encrypted.
   - Algorithms like DES, Triple DES, RC5, Blowfish, and AES are supported for encryption.
   - ESP also supports MAC algorithms like HMAC-MD5 and HMAC-SHA-1 for integrity verification.
4. **Modes of Operation**:
   - **Transport Mode**: Encrypts only the payload of the IP packet.
   - **Tunnel Mode**: Encrypts the entire IP packet for secure communication between networks.
5. **Padding Uses**:
   - Ensures data alignment for encryption algorithms.
   - Adds extra bytes to hide the actual length of the data, enhancing traffic flow confidentiality.

ESP Format

Figure 16.7 shows the format of an ESP packet. It contains the following fields:

- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** A monotonically increasing counter value; this provides an anti-replay function, as discussed for AH.
- **Payload Data (variable):** This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.

Figure 16.7. IPSec ESP format



- **Padding (0255 bytes):** The purpose of this field is discussed later.
- **Pad Length (8 bits):** Indicates the number of pad bytes immediately preceding this field.
- **Next Header (8 bits):** Identifies the type of data contained in the payload data field by identifying the first header in that payload (for example, an extension header in IPv6, or an upper-layer protocol such as TCP).

- **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

5

Key Management: Oakley and ISAKMP (Simplified for 5 Marks)

1. **Purpose of Key Management**:
   - Key management is essential in IPsec for securely exchanging and managing cryptographic keys between devices.
   - Oakley and ISAKMP are protocols used to handle this process efficiently.

2. **Oakley Protocol**:
    - Based on the **Diffie-Hellman key exchange**, which allows two parties to securely generate a shared secret key over an insecure network.
    - It enhances security by including features like:
        - **Perfect Forward Secrecy**: Ensures that even if a long-term key is compromised, past session keys remain secure.
        - **Key Freshness**: Uses unique values (nonces) to ensure that keys are not reused.
3. **ISAKMP (Internet Security Association and Key Management Protocol)**:
    - Provides a **framework** for negotiating, establishing, and managing Security Associations (SAs), which define how two devices will communicate securely.
    - Defines the format and process for exchanging keying information but does not specify the actual key exchange mechanism (e.g., Oakley can be used with it).
4. **Integration of Oakley and ISAKMP**:
    - Oakley handles **key generation** securely.
    - ISAKMP manages **key negotiation, authentication, and SA management**, ensuring a structured way to establish secure communication.
5. **Key Features**:
    - Supports multiple encryption and authentication methods.
    - Ensures interoperability between different systems.
    - Allows secure and efficient key exchanges, preventing attacks like replay or key reuse.

Summary:

Oakley provides the secure mechanism for key generation, while ISAKMP acts as the framework for managing the secure communication process. Together, they enable robust key management in IPsec systems.

6

SSL Protocol Stack (Simplified for 5 Marks)

The **Secure Sockets Layer (SSL)** protocol ensures secure communication over the internet. It operates as a layered stack with three key components:

---

1. **Handshake Protocol**:
    - Establishes a secure connection between the client (browser) and server.
    - Includes steps like:
        - **Authentication**: The server proves its identity using a digital certificate.
        - **Key Exchange**: The client and server agree on encryption keys using methods like RSA or Diffie-Hellman.
    - Negotiates encryption algorithms and session details before data exchange begins.

---

2. **Record Protocol**:
    - Handles the **secure transmission of data** between the client and server.
    - Ensures:
        - **Confidentiality**: Encrypts the data using the negotiated encryption key.
        - **Integrity**: Adds a Message Authentication Code (MAC) to verify that the data hasn't been altered in transit.

---

3. **Alert Protocol**:
    - Manages error notifications and connection issues.
    - Types of alerts:
        - **Warning Alerts**: Inform about non-critical issues (e.g., certificate expiry).
        - **Fatal Alerts**: Indicate critical problems (e.g., handshake failure), terminating the connection.

Key Features of SSL Protocol Stack:

- Provides **encryption, authentication, and data integrity**.
- Uses symmetric encryption for fast data transfer and public key cryptography for secure key exchange.
- Operates between the **application layer (e.g., HTTP)** and the **transport layer (TCP)**, securing web traffic.

Summary:

The SSL protocol stack ensures secure communication through **Handshake Protocol** for connection setup, **Record Protocol** for encrypted data transfer, and **Alert Protocol** for error handling. It protects data from eavesdropping and tampering.

7

Handshake Protocol Actions (Simplified for 5 Marks)

The **Handshake Protocol** is a part of SSL/TLS used to establish a secure connection between a client (e.g., web browser) and a server (e.g., website). Here's a simple explanation of its actions:

---

1. **Client Hello**:
    - The client starts by sending a message to the server.
    - It includes:
        - A list of supported encryption algorithms (cipher suites).
        - Random data for generating secure keys.

---

2. **Server Hello**:
    - The server responds with:
        - Its chosen encryption algorithm from the client's list.
        - Its own random data.
        - A **digital certificate** proving the server's identity (e.g., domain ownership).

---

3. **Key Exchange**:
    - The client verifies the server's certificate to ensure it is trustworthy.
    - A secure key exchange occurs:
        - The client sends a **pre-master key** encrypted with the server's public key.

        - Both client and server use this key and random data to generate a **session key** for encryption.

---

4. **Finished Message**:
    - The client and server confirm that secure keys and algorithms are ready.
    - They exchange a "finished" message encrypted with the session key, ensuring everything is working correctly.

---

5. **Secure Connection Established**:
    - Once the handshake is complete, the client and server start secure communication using the session key.
    - All data exchanged is now encrypted and protected.

Summary:

The **Handshake Protocol** establishes a secure connection by exchanging encryption details and verifying identities. Key steps include **Client Hello**, **Server Hello**, **Key Exchange**, and **Finished Message**, ensuring data confidentiality and integrity.

8

SET: Secure Electronic Transaction

SET is a protocol designed to ensure secure online transactions, especially for credit card payments. It protects sensitive information and ensures transactions are trustworthy.

Features of SET

1. **Confidentiality**: Encrypts credit card and personal details to protect them from unauthorized access.
2. **Authentication**: Verifies the identities of buyers, merchants, and banks involved in a transaction.
3. **Integrity**: Ensures that the data exchanged during a transaction is not altered.
4. **Interoperability**: Works across different platforms and systems, making it versatile for online transactions.

Benefits of SET

1. **Secure Transactions**: Protects sensitive information, reducing the risk of fraud.
2. **Trustworthy Payments**: Builds confidence between customers and merchants.
3. **Global Standard**: Accepted worldwide, enabling secure international transactions.
4. **Prevention of Fraud**: Uses encryption and authentication to prevent unauthorized activities.

Requirements for SET

1. **Digital Certificates**: Used to verify identities (issued by trusted Certificate Authorities).
2. **Encryption**: Protects payment details and transaction data during communication.
3. **Secure Software**: Both merchants and customers need software that supports SET.
4. **Payment Gateway**: Links the merchant with the bank to process transactions securely.

Components of Secure Electronic Commerce

1. **Cardholder**: The customer making the online purchase.
2. **Merchant**: The seller providing goods or services online.
3. **Issuer**: The bank that issued the card to the customer.
4. **Acquirer**: The merchant's bank responsible for processing the payment.
5. **Payment Gateway**: A secure system connecting the merchant's and cardholder's banks.
6. **Certificate Authority (CA)**: Issues digital certificates to verify identities.

Summary:

SET provides **confidentiality, authentication, and integrity** for online credit card transactions, making them secure. It benefits users by preventing fraud and enabling trustworthy global payments through encryption and certificates.

9

Intrusion Detection (Simplified for 5 Marks)

1. **Purpose**: Intrusion detection identifies unauthorized activity in a system after prevention mechanisms fail, acting as a second line of defense.
2. **Benefits**:
    - Detects and responds to intruders quickly, minimizing damage.
    - Acts as a deterrent to discourage intrusions.
    - Gathers data on attack techniques to improve security systems.
3. **Principle**: It assumes intruder behavior differs from legitimate user behavior in detectable ways, though there may be overlaps.
4. **Challenges**:
    - **False Positives**: Legitimate users incorrectly flagged as intruders.
    - **False Negatives**: Intruders not detected due to overly strict rules.
5. **Key Findings**:
    - Detecting "masqueraders" (unauthorized users) is easier by analyzing deviations from normal user patterns.
    - Detecting "misfeasors" (authorized users acting improperly) or "clandestine users" is more complex and may need advanced techniques.

10

Viruses and Worms Simplified

1. Nature of Worms:

- Worms are a type of malicious software (malware).
- Unlike viruses, worms **do not need a host file** to spread.
- They are **self-replicating** and can spread automatically across networks without user action.

2. Worms:

- A worm spreads by exploiting vulnerabilities in systems or networks.
- It replicates itself, consuming system resources like memory and bandwidth, potentially causing network slowdowns or system crashes.
- Examples include **Morris Worm** and **Conficker**.

3. State of Worm Technology:
Worms have evolved with modern technology:

- **Simple Worms**: Early worms spread without much harm, focusing on replication.
- **Advanced Worms**: Modern worms can carry **payloads** like stealing data, installing backdoors, or delivering ransomware.
- **Network Worms**: Target specific network vulnerabilities to spread rapidly across systems.
- **Polymorphic Worms**: Change their code structure to evade detection by antivirus programs.

Summary:

Worms are malicious programs that self-replicate and spread without user interaction. They exploit system vulnerabilities to harm networks, consume resources, and evade detection through advanced techniques like polymorphism.

11

Dual Signature and Payment Processing Simplified

Dual Signature:

- **Purpose**: Ensures the integrity and confidentiality of payment information in online transactions.
- It **links two pieces of information** securely:
    1. **Order Information (OI)** for the merchant.
    2. **Payment Information (PI)** for the bank.
- Ensures that:
    - The **merchant cannot see payment details** (PI).
    - The **bank cannot see order details** (OI).

How it Works:

- The customer creates a **digital signature** by hashing OI and PI separately, then combining the hashes into one and encrypting it with their private key.
- This **dual signature** guarantees that neither OI nor PI can be tampered with.

Payment Processing:

1. **Customer Initiates Transaction**:
    - Sends OI, PI, and the dual signature to the merchant.
2. **Merchant Processes Order**:
    - Verifies the OI and forwards the PI and dual signature to the bank.
3. **Bank Verifies Payment**:
    - Checks the dual signature and processes the payment.
4. **Completion**:
    - Bank confirms the transaction to the merchant.
    - Merchant ships the goods to the customer.

Key Benefits:

- **Confidentiality**: Keeps payment and order details separate and private.
- **Integrity**: Ensures that no data is altered during the transaction.
- **Authentication**: Verifies the customer's identity securely.

Dual Signature and Payment Processing together ensure secure and efficient online transactions.