UNIT-5

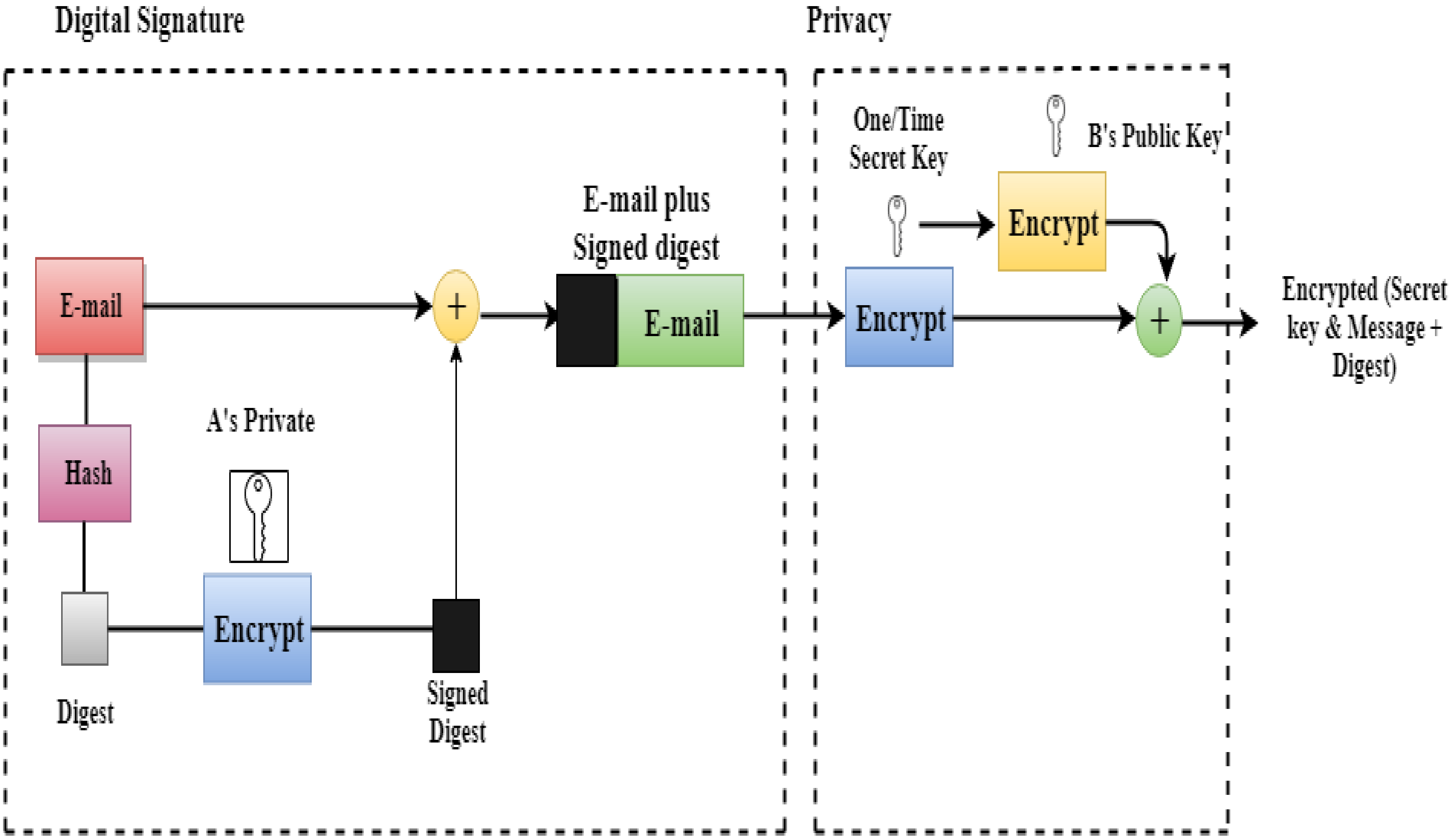
E-MAIL Security

Pretty Good Privacy

- PGP stands for Pretty Good Privacy (PGP) which is invented by Phil Zimmermann.
- PGP was designed to provide all four aspects of security, i.e., privacy, integrity, authentication, and non-repudiation in the sending of email.

- PGP uses a digital signature (a combination of hashing and public key encryption) to provide integrity, authentication, and non-repudiation. PGP uses a combination of secret key encryption and public key encryption to provide privacy. Therefore, we can say that the digital signature uses one hash function, one secret key, and two private-public key pairs.
- PGP provides authentication through the use of Digital Signature.

- It provides confidentiality through the use of symmetric block encryption.

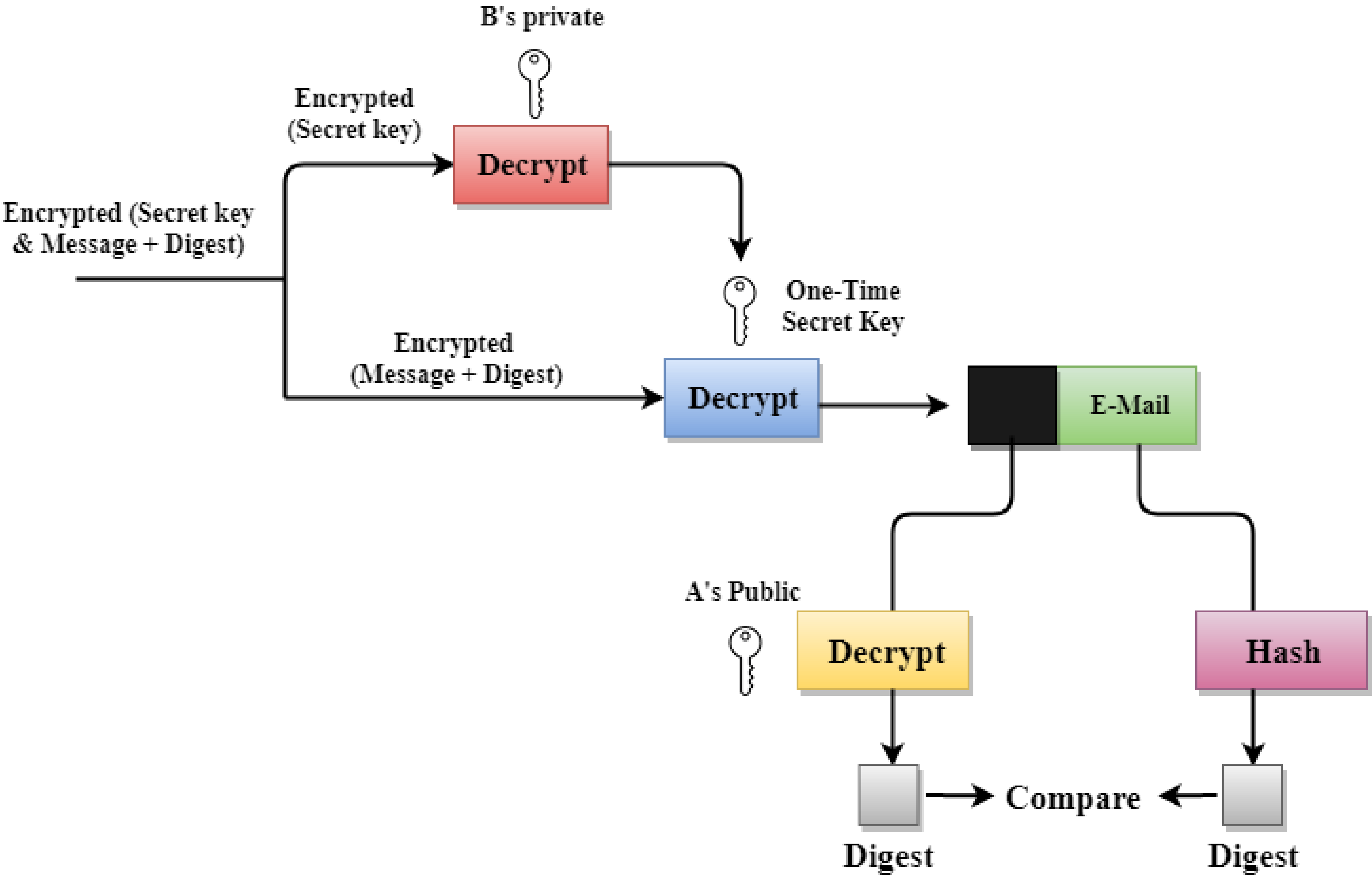


PGP at the Sender site (A)

- The e-mail message is hashed by using a hashing function to create a digest.
- The digest is then encrypted to form a signed digest by using the sender's private key, and then signed digest is added to the original email message.
- The original message and signed digest are encrypted by using a one-time secret key created by the sender.

- The secret key is encrypted by using a receiver's public key.
- Both the encrypted secret key and the encrypted combination of message and digest are sent together.

PGP at the Receiver site (B)



- The receiver receives the combination of encrypted secret key and message digest is received.
- The encrypted secret key is decrypted by using the receiver's private key to get the one-time secret key.
- The secret key is then used to decrypt the combination of message and digest.
- The digest is decrypted by using the sender's public key, and the original message is hashed by using a hash function to create a digest.

- Both the digests are compared if both of them are equal means that all the aspects of security are preserved.

What Is S/MIME

- **S/MIME or Secure/Multipurpose Internet Mail Extension** is a technology widely used by corporations that enhances **email security** by providing **encryption**, which protects the content of email messages from unwanted access. It also adds digital signatures, which confirm that you are the authentic sender of the message, making it a powerful weapon against many email-based attacks.

- **S/MIME Uses**
- S/MIME can be used to:
- Check that the email you sent has not been tampered with by a third party.
- Create digital signatures to use when signing emails.
- Encrypt all emails.
- Check the email client you’re using.

- **How Does S/MIME Work?**
- To operate, S/MIME employs mathematically related public and private keys. This technology is based on asymmetric cryptography. Because the two keys are mathematically related, a message that was encrypted with the public key (which is, of course, published) can only be decrypted using the private key (which is kept secret).

- When someone clicks “send” on an email, S/MIME sending agent software encrypts the message with the recipient’s public key, and the receiving agent decrypts it with the recipient’s private key. Needless to say, both the sender and the recipient must support S/MIME.
- You may be aware that SMTP-based Internet email does not provide message security. An SMTP (Simple Mail Transfer Protocol) internet email message can be read by anyone who sees it as it travels or views it where it is stored. S/MIME uses encryption to tackle these issues.

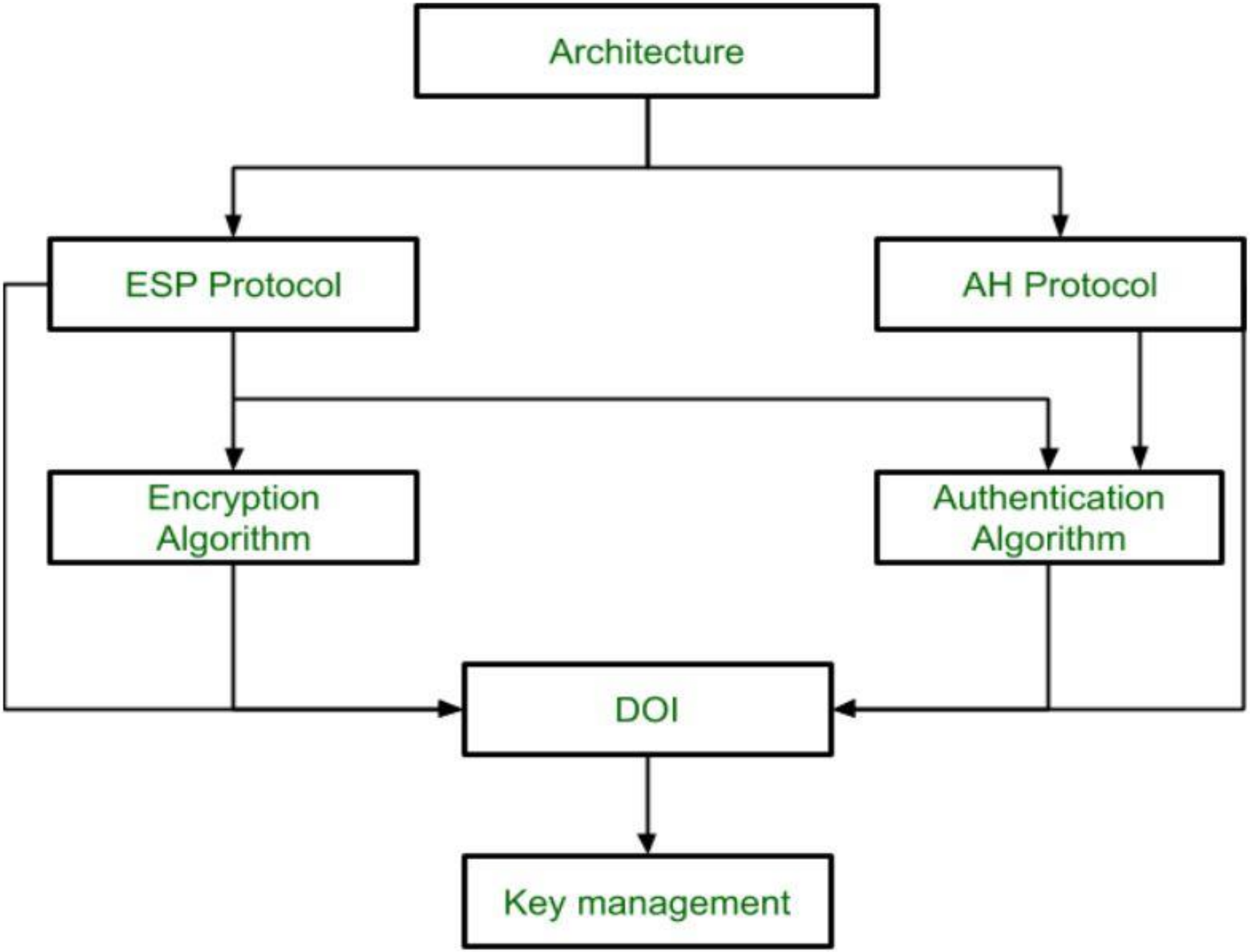
- **Functions of S/MIME:**
- Authentication
- Non repudiation
- Data integrity
- Privacy
- Data Security

IPSec Architecture

- **IPSec (IP Security) architecture** uses two protocols to secure the traffic or data flow. These protocols are ESP (Encapsulation Security Payload) and AH (Authentication Header). IPSec Architecture includes protocols, algorithms, DOI, and Key Management. All these components are very important in order to provide the three main services:

- Confidentiality
- Authentication
- Integrity

- **IP Security Architecture:**


Architecture
ESP Protocol
AH Protocol
Encryption Algorithm
Authentication Algorithm
DOI
Key management

- **1. Architecture:** Architecture or IP Security Architecture covers the general concepts, definitions, protocols, algorithms, and security requirements of IP Security technology.
- **2. ESP Protocol:** ESP(Encapsulation Security Payload) provides a confidentiality service. Encapsulation Security Payload is implemented in either two ways:
- ESP with optional Authentication.
- ESP with Authentication.

- **3. Encryption algorithm: The encryption** algorithm is the document that describes various encryption algorithms used for Encapsulation Security Payload.
- **4. AH Protocol:** AH (Authentication Header) Protocol provides both Authentication and Integrity service. Authentication Header is implemented in one way only: Authentication along with Integrity. Authentication Header covers the packet format and general issues related to the use of AH for packet authentication and integrity.

- **5. Authentication Algorithm:** The authentication Algorithm contains the set of documents that describe the authentication algorithm used for AH and for the authentication option of ESP.
- **6. DOI (Domain of Interpretation):** DOI is the identifier that supports both AH and ESP protocols. It contains values needed for documentation related to each other.
- **7. Key Management:** Key Management contains the document that describes how the keys are exchanged between sender and receiver.

COMBINING SECURITY ASSOCIATIONS

- An individual SA can implement either the AH or ESP protocol but not both. Sometimes a particular traffic flow will call for the services provided by both AH and ESP. Further, a particular traffic flow may require IPsec services between hosts and, for that same flow, separate services between security gateways, such as firewalls. In all of these cases, multiple SAs must be employed for the same traffic flow to achieve the desired IPsec services.

- **Case 1.** All security is provided between end systems that implement IPsec. For any two end systems to communicate via a SA, they must share the appropriate secret keys. Among the possible combinations are
- **a.** AH in transport mode
- **b.**ESP in transport mode
- **c.**ESP followed by AH in transport mode (an ESP SA inside an AH SA)
- **d.** Any one of a, b, or c inside an AH or ESP in tunnel mode
- We have already discussed how these various combinations can be used support to authentication, encryption, authentication before encryption, and authentication after encryption.

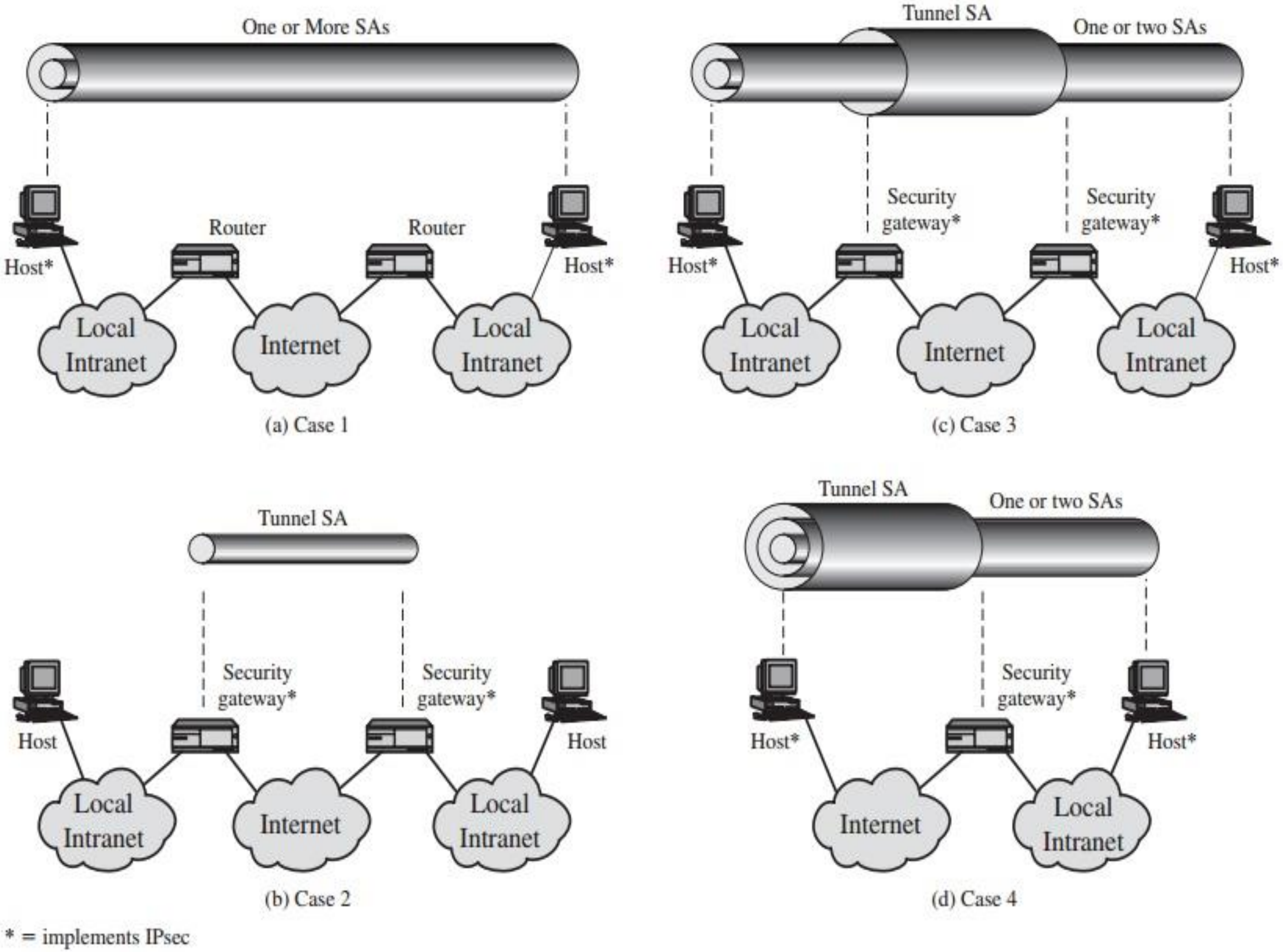


Figure 19.10 Basic Combinations of Security Associations

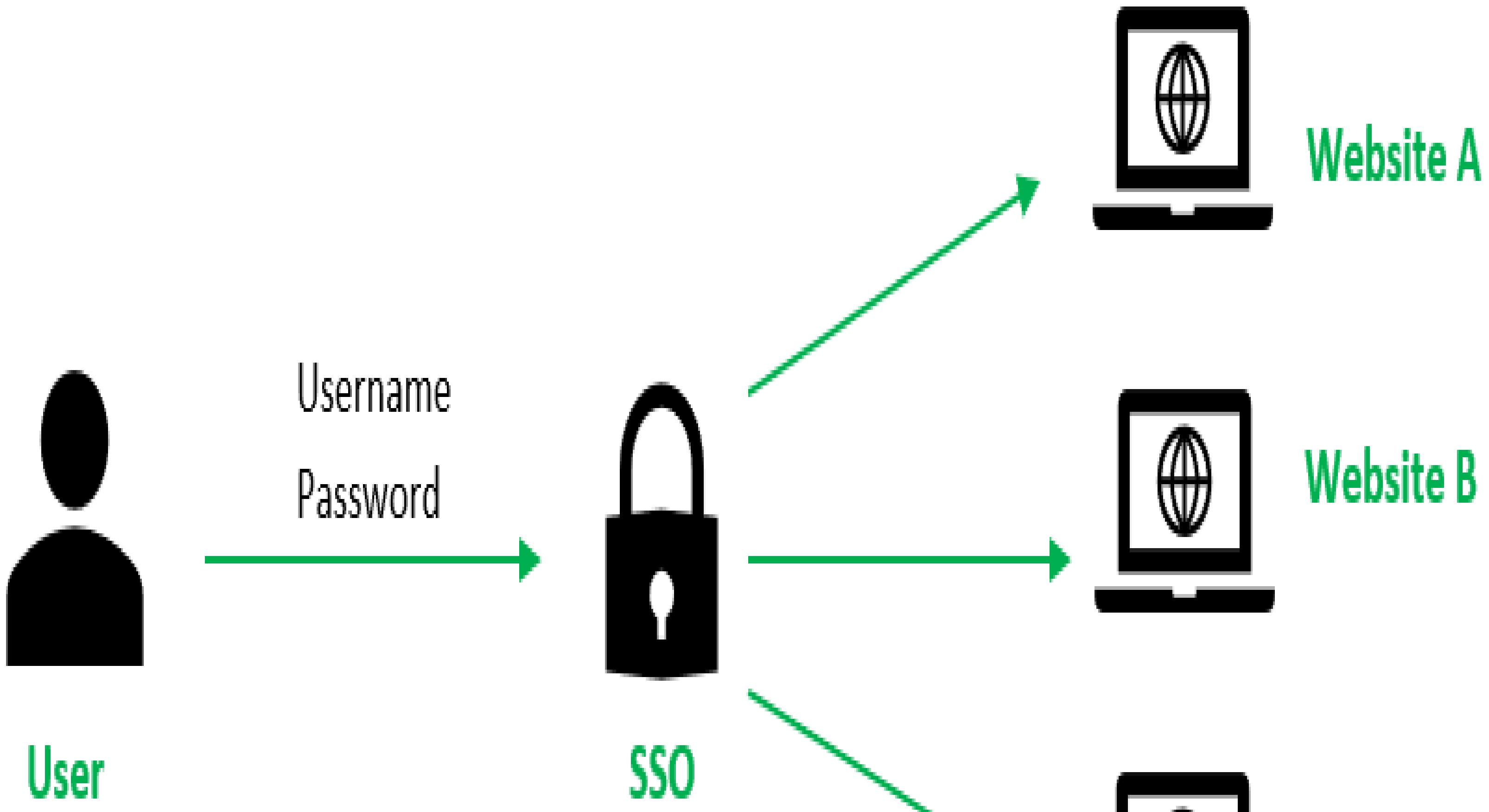
- **Case 2.** Security is provided only between gateways (routers, firewalls, etc.) and no hosts implement IPsec. This case illustrates simple virtual private network support. The security architecture document sp ecifies that only a single tunnel SA is needed for this case. The tunnel could support AH, ESP, or ESP with the authenti- cation option. Nested tunnels are not required, because the IPsec services apply to the entire inner packet.

- **Case 3.** This builds on case 2 by adding end-to-end security. The same combi- nations discussed for cases 1 and 2 are allowed here. The gateway-to-gateway tunnel provides either authentication, confidentiality, or both for all traffic between end systems. When the gateway-to-gateway tunnel is ESP, it also provides a limited form of traffic confidentiality. Individual hosts can implement any additional IPsec ser- vices required for given applications or given users by means of end-to-end SAs.

- **Case 4.** This provides support for a remote host that uses the Internet to reach an organization's firewall and then to gain access to some server or workstation behind the firewall. Only tunnel mode is required between the remote host and the firewall.As in case 1, one or two SAs may be used between the remote host and the local host.

Introduction of Single Sign On (SSO)

- **Single Sign On (SSO)** is an authentication scheme where users can securely authenticate and gain access to multiple applications and websites by only logging in with a single username and password.
  For example, logging in to your Google account once will allow you to access Google applications such as Google Docs, Gmail, and Google Drive.


Username
Password
User
SSO
Website A
Website B
Website C

- Without SSO solution, the website maintains a database of login credentials – username and passwords. Each time the user login to the website, it checks the user’s credentials against its database and authenticates the user.
- With the SSO solution, the website does not store login credentials in its database. Instead, SSO makes use of a shared cluster of authentication servers where users are only required to enter their login credentials once for authentication. With this feature of one login and multiple access, it is crucial to protect login credentials in SSO systems.

- Hence it is highly recommended to integrate SSO with other strong authentication means such as smart tokens or one-time passwords to achieve multi-factor authentication.