Lecture 8.2

Internet Security

Dr. Vandana Kushwaha

Department of Computer Science

Institute of Science, BHU, Varanasi

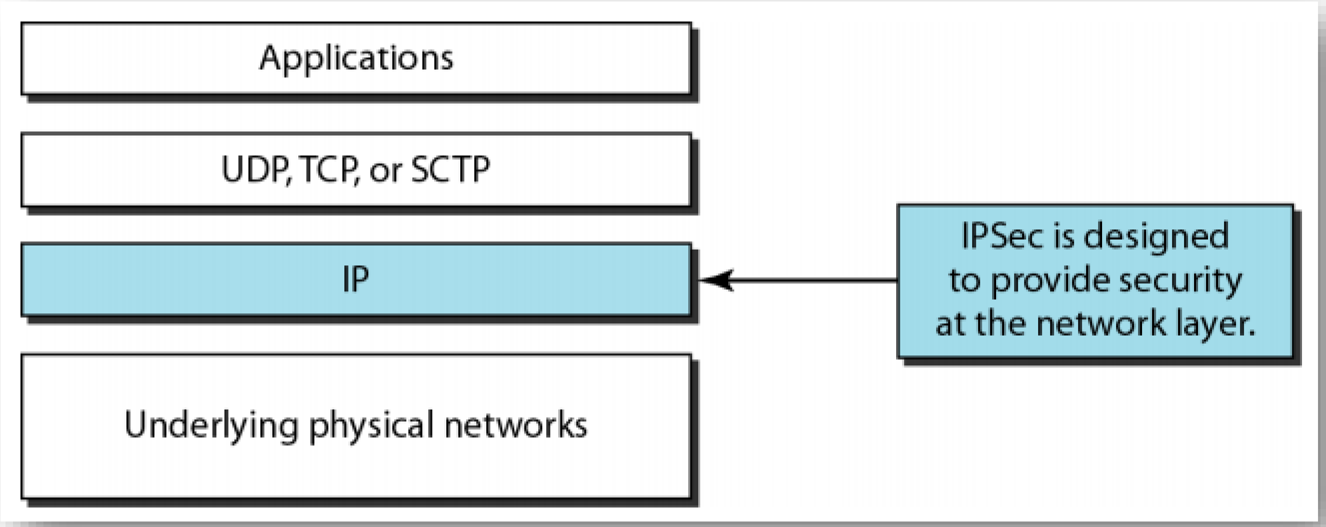
Introduction

- Certain security aspects particularly: **privacy** and **message authentication**, can be applied to the **Network**, **Transport**, and **Application layers** of the **TCP/IP Internet model**.
- **IPSec protocol** can add **Authentication** and **Confidentiality** to the **IP protocol.**
- **SSL(or TLS)** can do the same for the **TCP protocol.**
- **PGP** can do it for the **SMTP protocol (e-mail).**
  - ***Network Layer:** IPSec protocol*
  - ***Transport Layer:** SSL/TLS*
  - ***Application Layer:** PGP*

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 layer.**

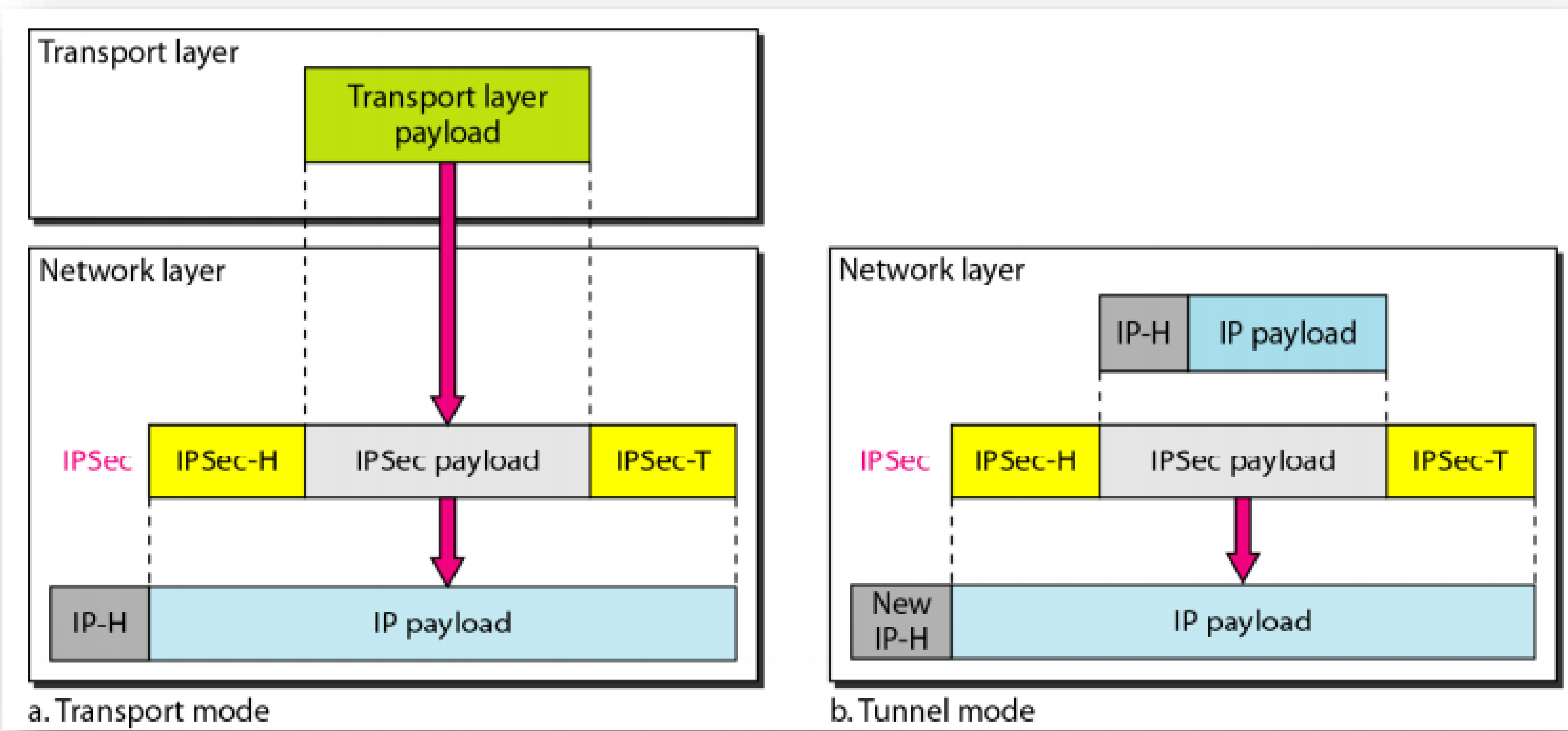
- **IPSec** helps to create **authenticated** and **confidential packets** for the **IP layer.**



Two Modes of IPSec

- **IPSec operates** in one of **two different modes**:

  1. ***Transport mode***
  2. ***Tunnel mode***

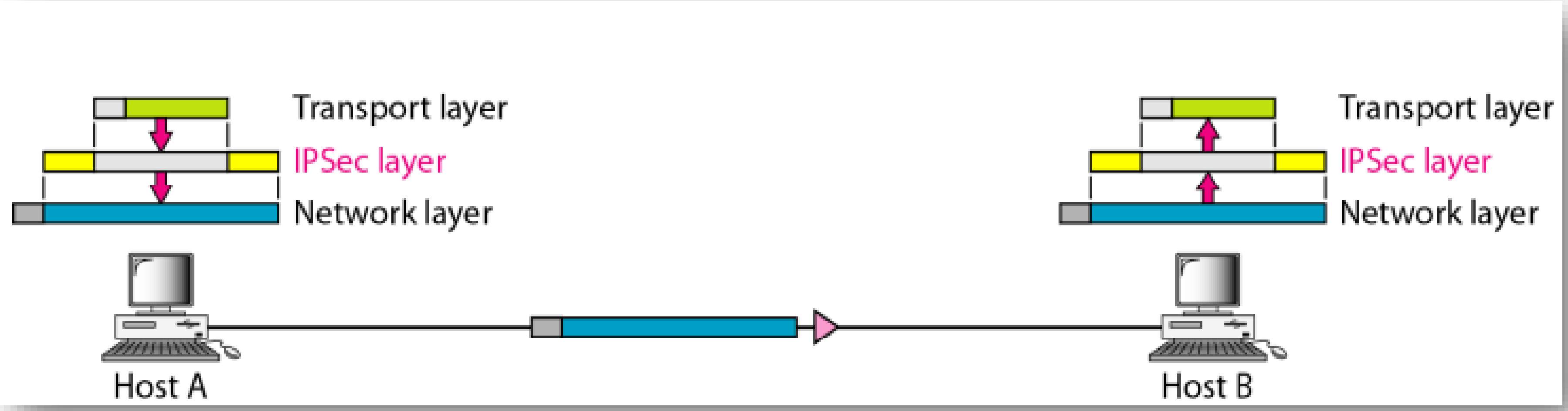


Transport Mode

- In the **Transport mode**, **IPSec protects** what is **delivered from the transport layer** to the **network layer.**
- In other words, the **transport mode protects** the **network layer payload**, the **payload** to be **encapsulated** in the **network layer.**
- Note that the **transport mode does not protect the IP header.**
- In other words, the **transport mode does not protect** the **whole IP packet**; it **protects only** the **packet** from the **transport layer** (the IP layer payload).
- In this mode, the **IPSec header** and **trailer** are **added** to the information coming from the **transport layer.**
- The **IP header** is **added later**.
- The **transport mode** is normally **used** when we need **host-to-host (end-to-end) protection of data.**

Transport Mode

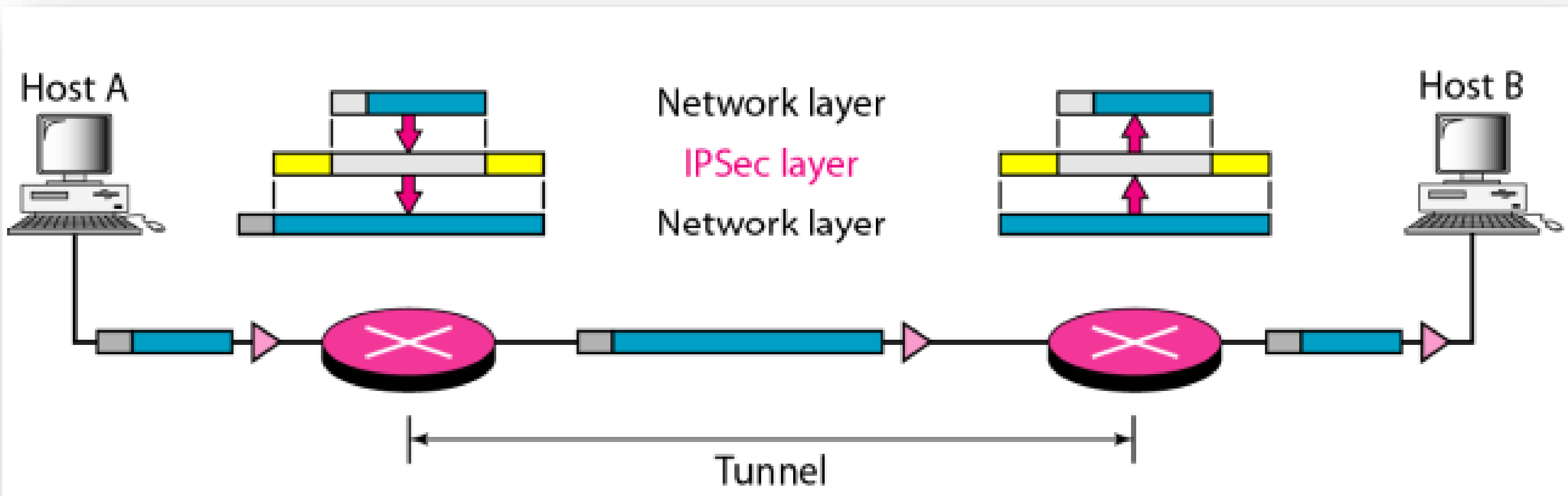
- The **sending host** uses **IPSec** to **authenticate and/or encrypt** the **payload delivered** from the **transport layer.**
- The **receiving host** uses **IPSec** to **check the authentication** and for **decrypt the IP packet** and **deliver** it to the **transport layer.**



Tunnel Mode

- In the **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**.
- The **tunnel mode** is normally **used between two routers**, **between a host and a router,** or **between a router and a host**.
- In other words, we **use the tunnel mode when either the sender or the receiver is not a host.**
- The **entire original packet** is **protected** from **intrusion** between the **sender** and the **receiver.**
- It's as if the **whole packet goes through an imaginary tunnel**.

Tunnel Mode



Host A
Network layer
IPSec layer
Network layer
Host B
Tunnel

Network Layer: Security Protocols

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

1. Authentication Header (AH) Protocol

- The **Authentication Header (AH) Protocol** is designed to **authenticate the source host** and **to ensure the integrity** of the **payload** carried in the **IP packet**.
- The **AH Protocol** provides **source authentication** and **data integrity**, but **not privacy**.
- The **protocol** uses a **hash function** and a **symmetric key** to create a **message digest**; the **digest is inserted** in the **authentication header.**
- The **AH is then placed** in the **appropriate location** based on the **mode (transport or tunnel).**

Network Layer: Security Protocols

2. Encapsulating Security Payload (ESP)

- The **AH Protocol** does **not provide privacy**, only **source authentication** and **data integrity.**
- **IPSec** later defined an **alternative protocol** that provides **source authentication**, **integrity,** and **privacy** called **Encapsulating Security Payload (ESP).**
- **ESP adds** a **header** and **trailer** both**.**
- **ESP's** **authentication data** are **added at the end of the packet.**

Security Protocol at Transport Layer

- A **Transport layer security** provides **end-to-end security** services for **applications** that use a **reliable transport layer protocol** such as **TCP.**
- The **idea** is to provide **security services** for transactions on the **Internet.**
- **Two protocols** are **dominant today** for providing **security** at the **transport layer**:

1. Secure Sockets Layer (SSL) Protocol

2. Transport Layer Security (TLS) Protocol.

- The latter(**TLS**) is actually an **IETF version** of the former(**SSL**).

SSL Services

- **Netscape** developed **SSL in 1994**.
- **Secure Socket Layer (SSL)** is designed to provide **security** and **compression services** to **data** generated from the **application layer.**
- Typically, **SSL** can **receive data** from any **application layer protocol**, but usually the protocol is **HTTP.**
- The **data received** from the **application** are **compressed (optional), signed,** and **encrypted.**
- The **data** are then **passed** to a **reliable transport layer protocol** such as **TCP.**
- **SSL** provides **several services** on **data** received from the **application layer.**

SSL Services

Fragmentation

- First, **SSL divides** the data into **blocks of 2^{14} bytes** or less.

Compression

- Each **fragment** of data is **compressed** by using one of the **lossless compression methods** negotiated between the client and server. This service is **optional.**

Message Integrity

- To **preserve** the **integrity** of **data**, **SSL** uses a **keyed-hash function** to **create a MAC**.

SSL Services

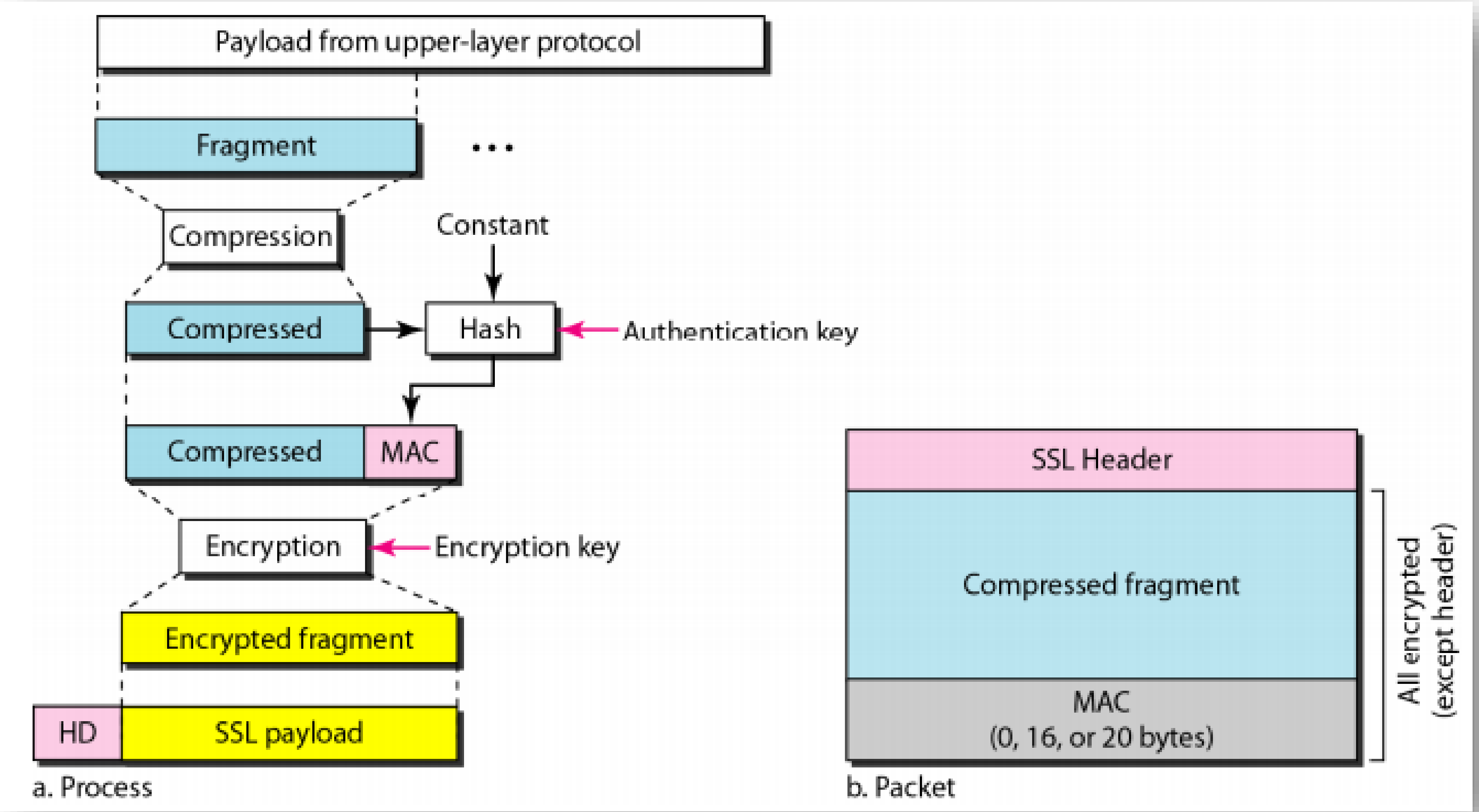
Confidentiality

- To **provide confidentiality**, the **original data** and the **MAC are encrypted using symmetric key cryptography.**

Framing

- A **header** is added to the **encrypted payload**.
- The **payload** is then **passed** to a **reliable transport layer protocol.**

SSL Services



Security Protocol at Application Layer

- 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**.
- **Sending** an **e-mail** is a **one-time activity**.
- In **IPSec** or **SSL**, we assume that the **two parties create a session** between themselves and **exchange data in both directions**.
- **In e-mail**, there is **no session.**
- **Alice** and **Bob cannot create a session.**
- **Alice** sends a **message** to **Bob;** sometime later, **Bob** reads the **message** and may or may not **send a reply.**

PGP

- In **PGP** the **security parameters** need to be **sent along with the message**.
- In **PGP,** the **sender** of the **message** needs to **include** the **identifiers of the algorithms used** in the **message** as well as the **values of the keys**.

PGP Services

- **PGP** can provide **several services** based on the **requirements** of the **user.**
- An **e-mail** can use one or more of these **services.**

Plaintext

- The **simplest case** is to **send** the **e-mail** message in **plaintext (no service).**
- **Alice,** the **sender**, composes a message and **sends** it to **Bob,** the **receiver.**
- The **message** is **stored** in **Bob's mailbox** until it is **retrieved** by him.

PGP

Message Authentication

- Probably the next improvement is to let **Alice sign the message**.
- **Alice creates a digest** of the message and **signs it with her private key.**
- When **Bob** receives the **message**, he **verifies** the **message** by using **Alice's public key.**
- **Two keys** are **needed** for this **scenario.**
- **Alice** needs to know her ***private key***; **Bob** needs to know **Alice's public key**.

Compression

- A further **improvement** is to **compress the message and digest to make the packet more compact.**
- This **improvement** has **no security benefit**, but it eases the **traffic.**

PGP

Confidentiality with One·Time Session Key

- **Confidentiality** in an **e-mail system** can be achieved by using **conventional encryption** with a **one-time session key.**
- **Alice** can create a **session key**, use the **session key** to **encrypt the message** and the **digest,** and **send the key itself with the message.**
- To **protect** the **session key**, **Alice encrypts** it with **Bob's public key**.

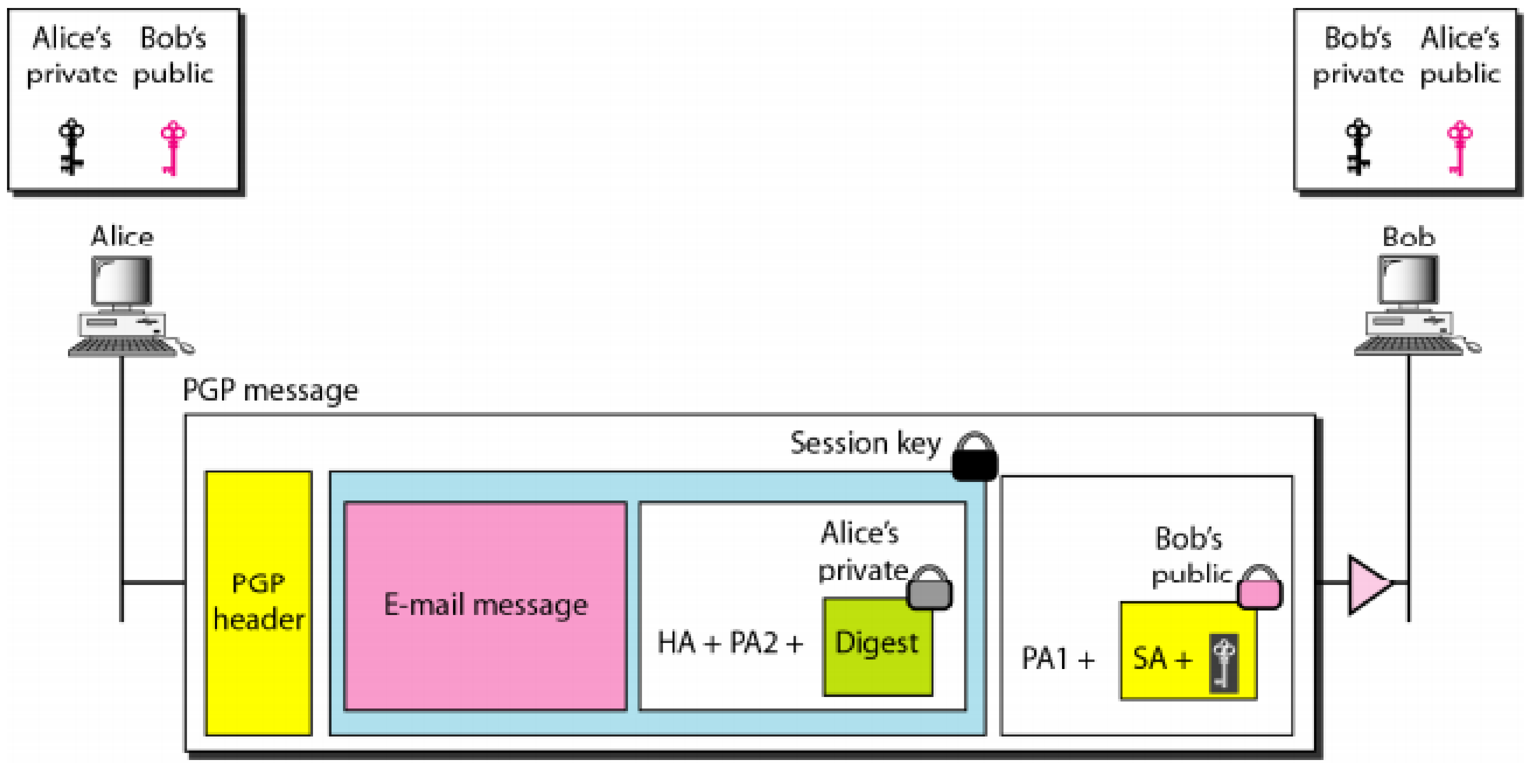
Code Conversion

- Most **e-mail systems** allow the message to consist of **only ASCII characters**.
- To **translate other characters** not in the **ASCII set**, **PGP** uses **Radix 64 conversion**.
- **Each character** to be **sent** (after encryption) is **converted** to **Radix 64 code.**

PGP

- The **whole idea** of **PGP** is based on the **assumption** that a **group of people** who need to exchange **e-mail messages** **trust** one another.
- **Everyone** in the **group** somehow knows (with a **degree of trust**) the **public key** of **any other person** in the **group.**

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)