其他内容可访问博客：either fight | or die (yst-10.github.io)

# Block4

## 一. Web Security TLS/SSL

### 1.WEB SECURITY CONSIDERATIONS

### (1)The Web

**<1>定义：Is a client/server application running over the Internet using TCP/IP**

是一个使用 TCP/IP 在互联网上运行的客户端/服务器应用程序

<2> the following characteristics of Web usage suggest the need for tailored security tools

Web 使用的以下特征表明，需要量身定制的安全工具

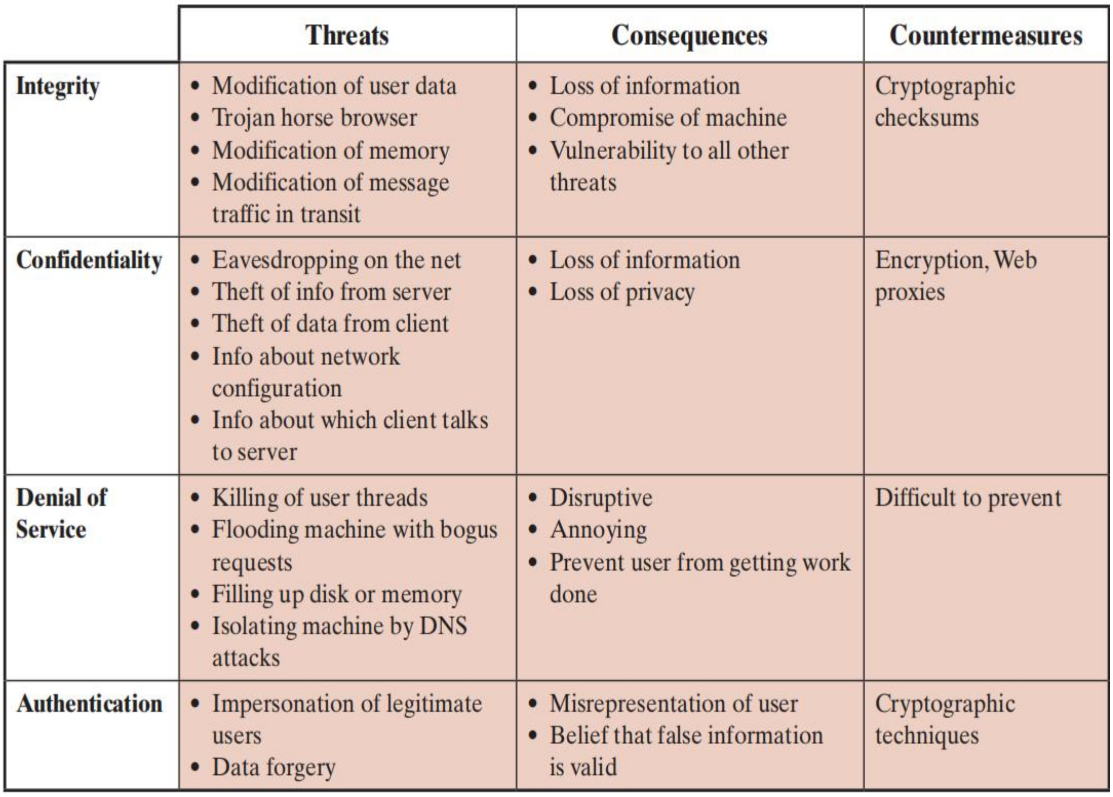
– Web browsers are easy to use and configure, but the underlying software is complex. This complexity may hide potential security flaws.

Web 浏览器易于使用和配置，但底层软件很复杂。这种复杂性可能会隐藏潜在的安全缺陷。

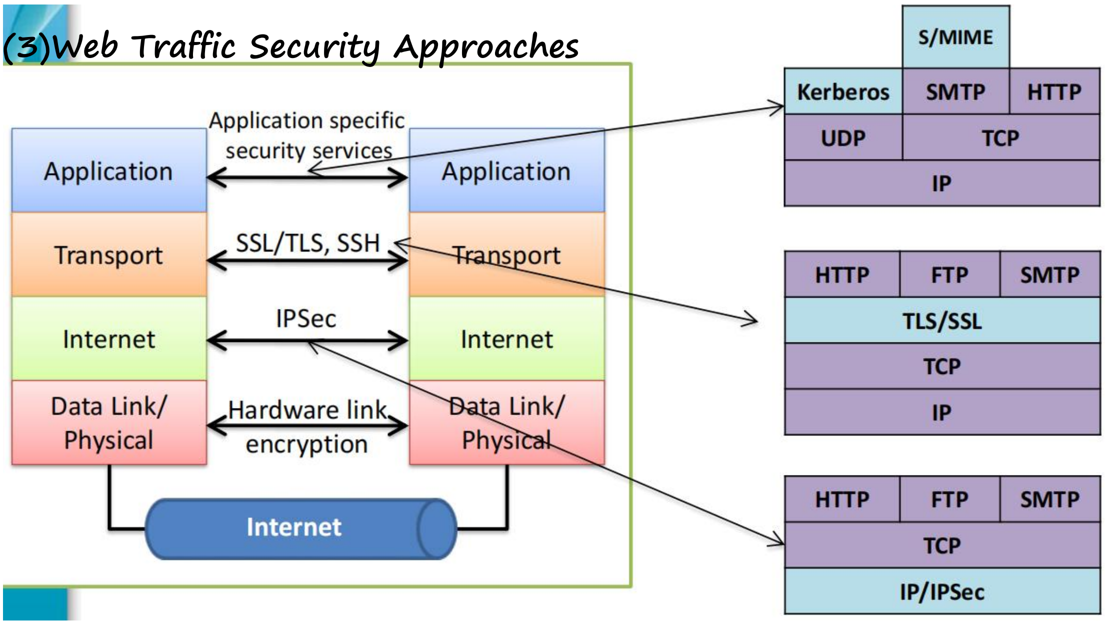
– A Web server can be exploited as a launching pad into the corporation's entire computer system. Web 服务器可以作为公司整个计算机系统的发射台

– In general users are not aware of security risk or do not have the know-how to take effective countermeasures.一般来说，用户不知道安全风险或无权采取有效的应对措施。

### (2)Web Security Threats



| | Threats | Consequences | Countermeasures |
|---|---|---|---|
| **Integrity** | • Modification of user data<br>• Trojan horse browser<br>• Modification of memory<br>• Modification of message traffic in transit | • Loss of information<br>• Compromise of machine<br>• Vulnerability to all other threats | Cryptographic checksums |
| **Confidentiality** | • Eavesdropping on the net<br>• Theft of info from server<br>• Theft of data from client<br>• Info about network configuration<br>• Info about which client talks to server | • Loss of information<br>• Loss of privacy | Encryption, Web proxies |
| **Denial of Service** | • Killing of user threads<br>• Flooding machine with bogus requests<br>• Filling up disk or memory<br>• Isolating machine by DNS attacks | • Disruptive<br>• Annoying<br>• Prevent user from getting work done | Difficult to prevent |
| **Authentication** | • Impersonation of legitimate users<br>• Data forgery | • Misrepresentation of user<br>• Belief that false information is valid | Cryptographic techniques |



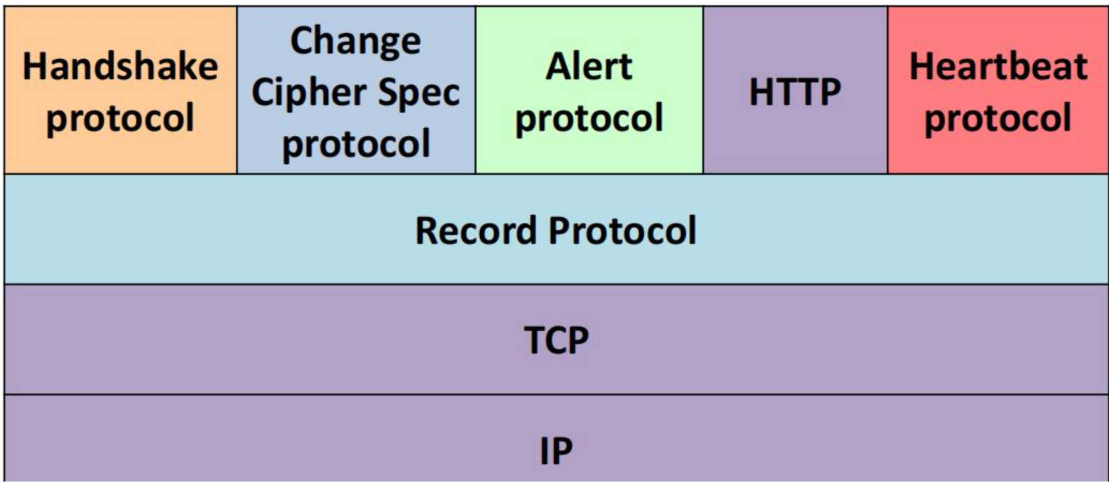
### (3)Web Traffic Security Approaches

## 2.Transport Layer Security (TLS)--derived from Secure Sockets Layer (SSL)

TLS is designed to make use of TCP to provide a reliable end-to-end secure service.
TLS 旨在利用 TCP 提供可靠的端到端安全服务。

### (1)TLS Architecture

TLS is not a single protocol but rather two layers of protocols



The TLS Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of TLS. Three higher-layer protocols are defined as part of TLS: the Handshake Protocol; the Change Cipher Spec Protocol; and the Alert Protocol

## (2)TLS Connection

<1>定义：A **connection** in TLS is a transport that provides suitable type of service (they are peer-to-peer and transient). Every connection is associated with one session. TLS 中的连接是一种提供合适的服务类型的传输（它们是点对点的和瞬态的）。每个连接都与一个会话关联。

<2>A connection state is defined by the following parameters: 连接状态由以下参数定义

– Server and client random (i.e. nonce) 由服务器和客户机为每个连接选择的字节序列

– Server write MAC secret： The secret key used in MAC operations on data sent by the server.

– Client write MAC secret：对客户端发送的数据进行 MAC 操作中使用的对称密钥。

– Server write key：The symmetric encryption key for data encrypted by the server and decrypted by the client

– Client write key: 由客户端加密并由服务器解密的数据的对称加密密钥

– Initialisation vectors (i.e. IVs for CBC encryption) 当使用 CBC 模式下的块密码时，为每个键保持一个初始化向量（IV）。此字段首先由 TLS 握手协议初始化。此后，每个记录的最后密文块被保留，用作以下记录的 IV。

– Sequence number

## (3)TLS Session

<1>定义：A **session** in TLS is an association between a client and a server.

- Sessions are created by the Handshake Protocol.
- Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. 会话定义了一组加密安全参数，它们可以在多个连接中共享
- Sessions are used to avoid expensive renegotiation of security parameters for each connection. 会话用于避免对每个连接进行昂贵的安全参数的重新协商

<2>A TLS Session state is defined by: ( Once a session is established, there is a current operating state for both read and write (i.e., receive and send))

- Session identifier
- Peer certificate (X509.v3) – authentication; to create trust（可以为空）
- Compression method 在加密之前压缩数据的算法
- Cipher Spec (null, DES, MD5, SHA-1, …) 指定算法，定义属性
- Master secret – to authenticate (& relate) the connection to a session 主密钥
- Is resumable – a flag indicating whether the session can be used to initiate new connections 指示该会话是否可以用于启动新的连接的标志

## (4)Handshake protocol

**<1>**This protocol performs all the tasks requiring agreement between the two entities before

they set up the secure TLS channel: 在两个实体设置安全的 TLS 通道之前，该协议将执行它们之间需要达成协议的所有任务：

– agree on the cipher suite to be used to establish the secure channel;
 同意使用用于建立安全通道的密码套件

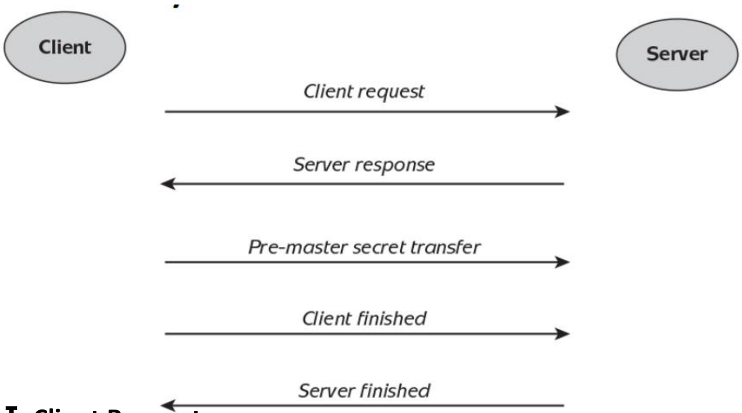
– allows the server and client to authenticate each other; and 同意彼此身份认证

– establish the keys needed to secure the channel. 建立保护通道所需的密钥

注：Cipher suite: a list that contains the combinations of cryptographic algorithms.

密码套件：一个包含加密算法组合的列表。

## <2>Handshake protocol (A simple version for TLS 1.2 and earlier versions)



**Ⅰ.Client Request:**

– a session ID: a unique identifier for the session;

– a pseudorandom number (nonce) $r_c$: for the provision of freshness; and

一个伪随机数（一次性）rc：为了提供新鲜度

– a list of cipher suites the client supports (including key exchange method)客户端支持的密码套件列表（包括密钥交换方法）

补充：**Supported Key exchange methods**

**• RSA**

**• Fixed Diffie-Hellman**: Diffie-Hellman public parameters contained in server's certificate, signed by CA. 服务器证书中包含的 Diffie-Hellman 公共参数，由 CA 签名。

**• Ephemeral Diffie-Hellman**: 短暂

– Sender generates a fresh set of parameters, and sends the public values alongside a digital signature on the chosen parameters.

发送方生成一组新的参数，并将公共值与所选参数上的数字签名一起发送。

– This creates ephemeral(temporary, one-time) secret keys, and considered the most secure DH option. 这将创建短暂（临时、一次性）密钥，并被认为是最安全的 DH 选项。

– Offers perfect forward secrecy. 提供完美的前瞻性保密。

**• Anonymous Diffie-Hellman**: Basic Diffie-Hellman used without authentication.

匿名 D-H：没有认证的基本的 D-H

**Ⅱ.Server Response:**

• the session ID;

• Server's nonce $r_s$ ; 服务器的随机数

• the particular cipher suite the server has decided to use;

服务器已经决定使用的特定密码套件

• a copy of the server's public-key certificate; and 服务器的公钥证书的副本

• if the Ephemeral Diffie–Hellman is chosen, then the server also generates a fresh set of parameters, and sends the public values alongside a digital signature on the chosen parameters.

如果选择了短暂的 D-H，那么服务器还会生成一组新的参数，并将公共值与所选参数上的数字签名一起发送。

注意：After receiving server response message, client need to

- check the server’s public-key certificate is valid  检查服务器的公钥证书是否有效
- If the Ephemeral Diffie–Hellman protocol is being used, then the client should verify the digital signature on the Diffie– Hellman parameters.  如果使用了短暂的 D-H，那么客户端应该验证 D-H 参数上的数字签名。

### Ⅲ. Pre-master Secret Transfer:

The client and server now need to agree on a shared secret $K_P$ (the pre-master secret).
客户端和服务器现在需要就共享秘密 KP（预主秘密）达成一致。
– RSA: the client generates $K_P$ , encrypted using the server’s public key and sends to the server;
客户端生成 KP，使用服务器的公钥进行加密，并发送到服务器
– Ephemeral Diffie–Hellman: the client generates a fresh temporary Diffie–Hellman key pair and sends the public value to the server, after which both client and server compute the shared secret $K_P$ .  短暂的 D-H：客户端生成一个新的临时的 D-H 密钥对，并将公共值发送到服务器，之后客户端和服务器都计算共享的秘密 KP

补充：The client and server can now derive the keys required to secure the TLS session:
客户端和服务器现在可以派生出保护 TLS 会话所需的密钥
– compute the master secret $K_M$ using a key derivation function, taking $K_P$ , $r_C$ and $r_S$ as part of inputs.  使用密钥推导函数计算主密钥 KM，以 KP、rC 和 rS 作为输入的一部分。
– derive MAC and encryption keys from KM. From this point on, all exchanged messages are cryptographically protected.
从 KM 派生 MAC 和加密密钥。从那时起，所有交换的消息都受到加密保护

### Ⅳ.Client Finished.

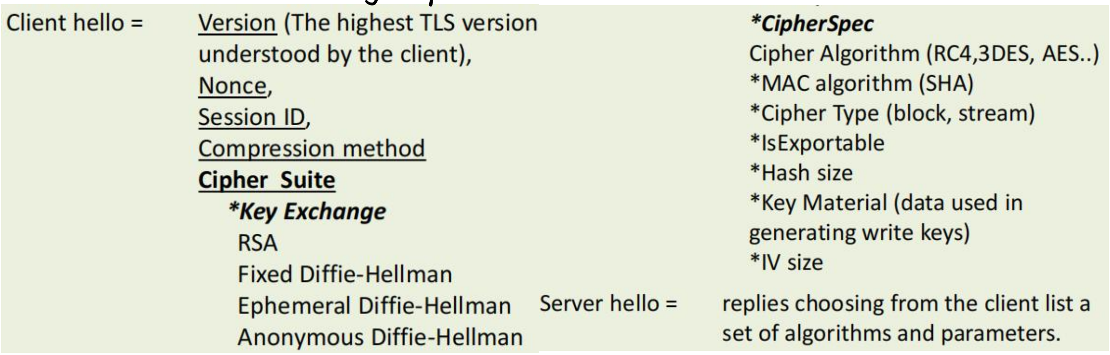
– The client computes a MAC on the hash of all the messages sent thus far.
客户端根据迄今为止发送的所有消息的哈希值计算一个 MAC
– This MAC is then encrypted and sent to the server.  然后将此 MAC 加密并发送到服务器

### Ⅴ.Server Finished.

– The server checks the MAC received from the client.  服务器将检查从客户端接收到的 MAC。
– The server then computes a MAC on the hash of all the messages sent thus far.  然后，服务器根据迄今为止发送的所有消息的哈希值计算一个 MAC
– This MAC is then encrypted and sent to the client.  然后将此 MAC 加密并发送到客户端。

### <3>Handshake Protocol:

### Phase 1: Establish Security Capabilities 建立安全能力

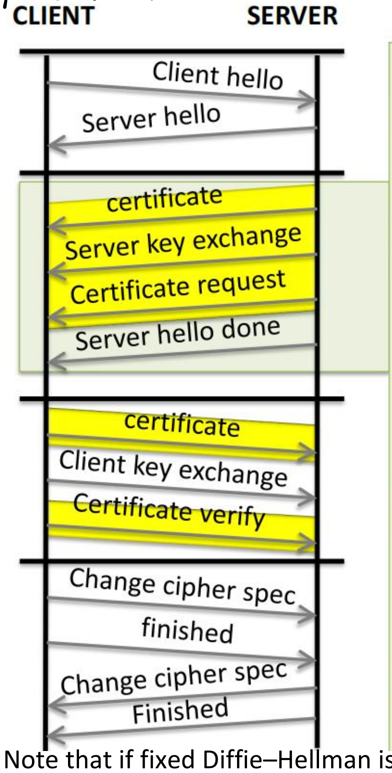


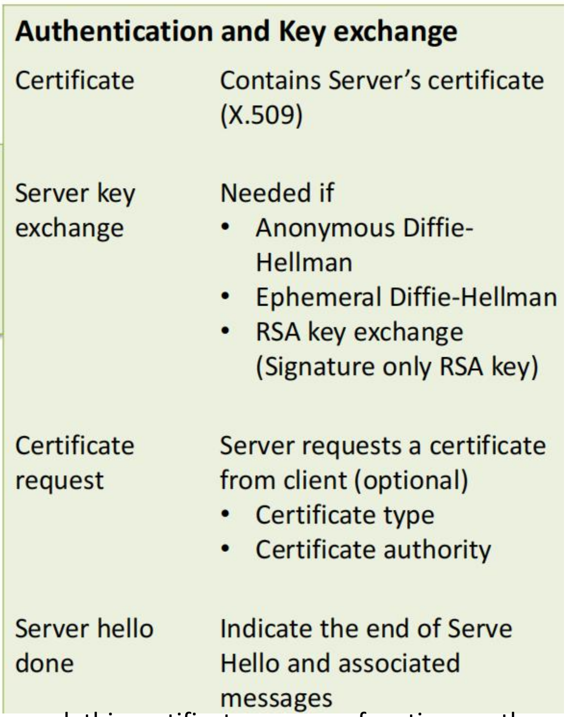
Client hello = Version (The highest TLS version understood by the client),
Nonce,
Session ID,
Compression method
**Cipher Suite**
***Key Exchange***
RSA
Fixed Diffie-Hellman
Ephemeral Diffie-Hellman
Anonymous Diffie-Hellman

***CipherSpec***
Cipher Algorithm (RC4,3DES, AES..)
*MAC algorithm (SHA)
*Cipher Type (block, stream)
*IsExportable
*Hash size
*Key Material (data used in generating write keys)
*IV size

Server hello = replies choosing from the client list a set of algorithms and parameters.

## phase 2: Authentication and Key exchange





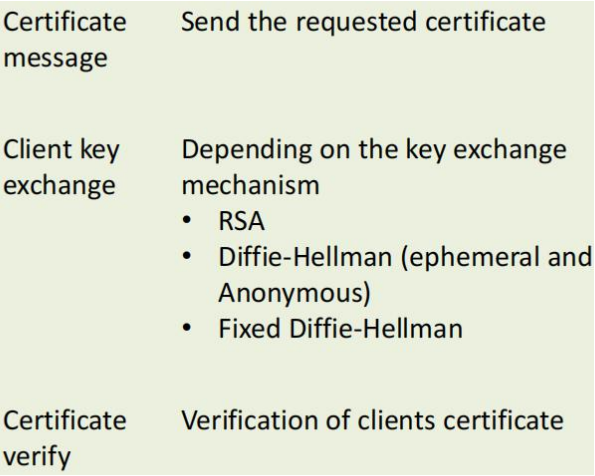
**Authentication and Key exchange**

| | |
|---|---|
| Certificate | Contains Server's certificate (X.509) |
| Server key exchange | Needed if<br>• Anonymous Diffie-Hellman<br>• Ephemeral Diffie-Hellman<br>• RSA key exchange (Signature only RSA key) |
| Certificate request | Server requests a certificate from client (optional)<br>• Certificate type<br>• Certificate authority |
| Server hello done | Indicate the end of Serve Hello and associated messages |

Note that if fixed Diffie–Hellman is used, this certificate message functions as the server's key exchange message because it contains the server's public Diffie–Hellman parameters. 请注意，如果使用 fixed D-H，此证书消息将作为服务器的密钥交换消息，因为它包含服务器的公共 D-H 参数。

Next, a server_key_exchange message may be sent if it is required. It is not required in two instances: (1) The server has sent a certificate with fixed Diffie– Hellman parameters; or (2) RSA key exchange is to be used. 接下来，如果需要，可以发送一个服务器密钥交换消息。在两种情况下不需要：(1)服务器发送了具有 fixed D-H 参数的证书；或(2)使用 RSA 密钥交换。

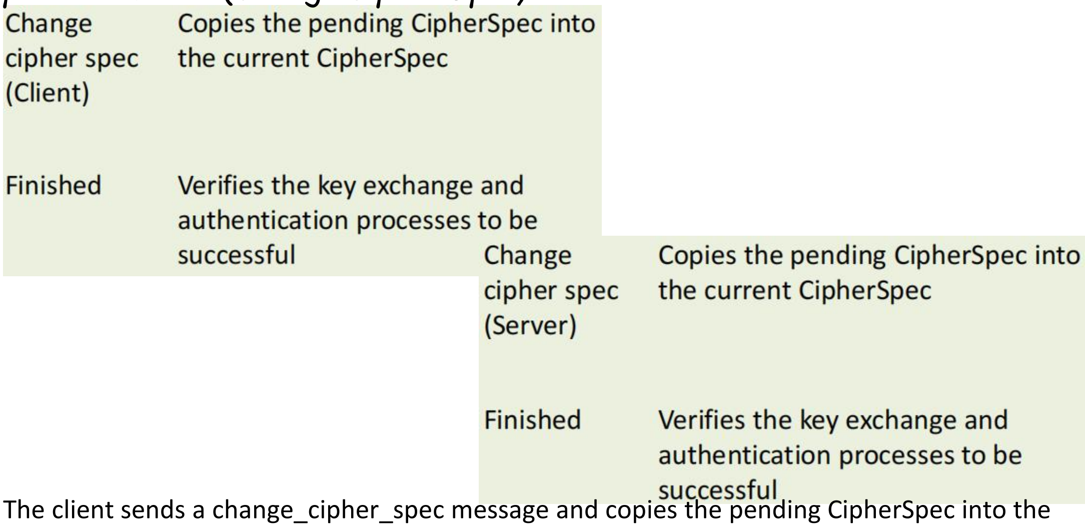
## phase 3: Client Authentication and Key exchange



| | |
|---|---|
| Certificate message | Send the requested certificate |
| Client key exchange | Depending on the key exchange mechanism<br>• RSA<br>• Diffie-Hellman (ephemeral and Anonymous)<br>• Fixed Diffie-Hellman |
| Certificate verify | Verification of clients certificate |

Fixed Diffie–Hellman: The client's public Diffie–Hellman parameters were sent in a certificate message, so the content of this message is null.

### phase 4: Finish (Change Cipher Spec)



| | |
|---|---|
| Change cipher spec (Client) | Copies the pending CipherSpec into the current CipherSpec |
| Finished | Verifies the key exchange and authentication processes to be successful |

| | |
|---|---|
| Change cipher spec (Server) | Copies the pending CipherSpec into the current CipherSpec |
| Finished | Verifies the key exchange and authentication processes to be successful |

The client sends a change_cipher_spec message and copies the pending CipherSpec into the current CipherSpec. 客户端发送一个更改_cipher_spec 消息，并将挂起的密码规范复制到当前密码规范。Note that this message is not considered part of the Handshake Protocol but is sent using the Change Cipher Spec Protocol.请注意，此消息不被认为是握手协议的一部分，而是使用 Change Cipher Spec Protocol 发送的

## (5)Change Cipher Spec Protocol

<1>This is essentially the last phase of Handshake protocol.基本上是握手协议的最后一个阶段。

<2>Change of cipher suites 改变密码套件

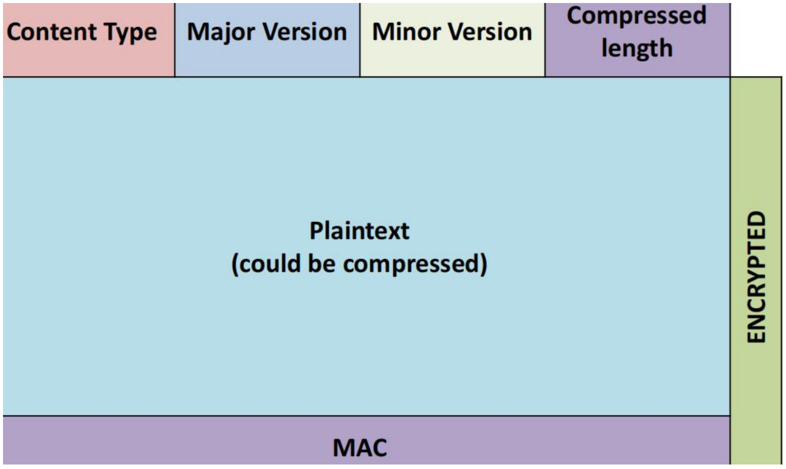
– Sends one message which updates the cipher suite to be used on this connection. The message is a single byte with value 1. 发送一条消息，其中将更新要在此连接上使用的密码套件。该消息是一个值为 1 的单个字节

补充：The sole purpose of this message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection.此消息的唯一目的是使挂起状态复制到当前状态中，该状态将更新要在此连接上使用的密码套件

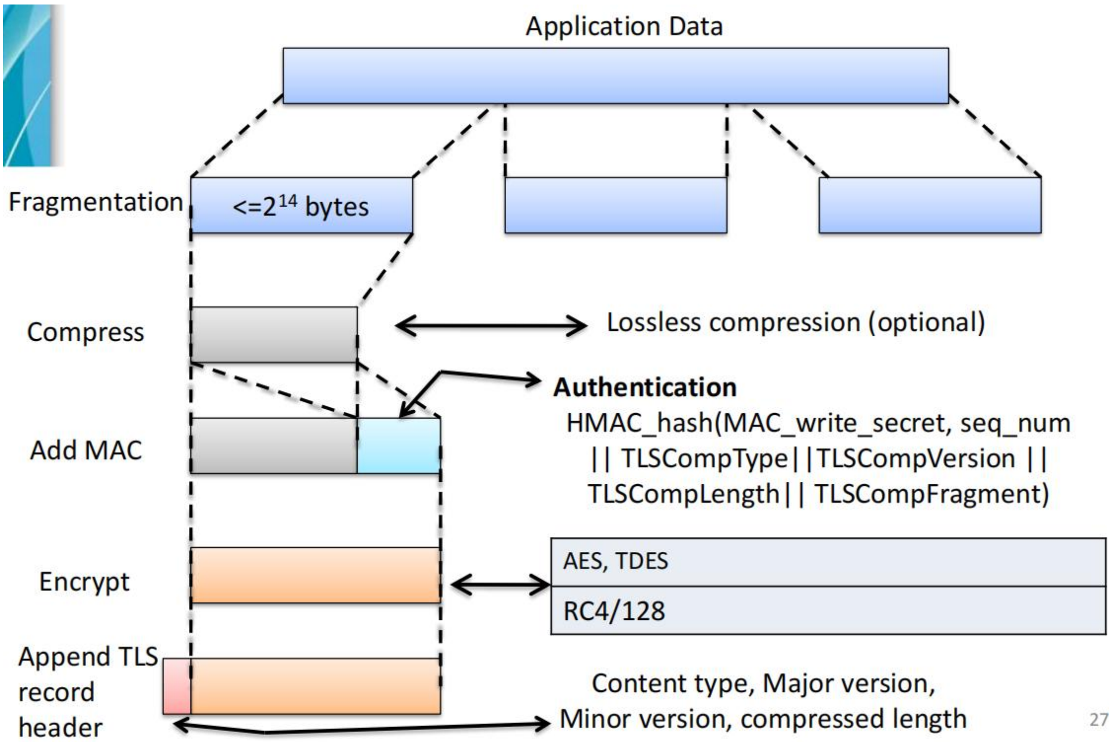
## (6)TLS Record Protocol

- TLS Record protocol provides:
  - – Confidentiality – Message Integrity

### <1>TLS Record Format



## <2>overall operation of the TLS Record Protocol



Application Data

Fragmentation <=$2^{14}$ bytes

Compress

Lossless compression (optional)

**Authentication**

HMAC_hash(MAC_write_secret, seq_num || TLSCompType||TLSCompVersion || TLSCompLength|| TLSCompFragment)

Add MAC

Encrypt

AES, TDES

RC4/128

Append TLS record header

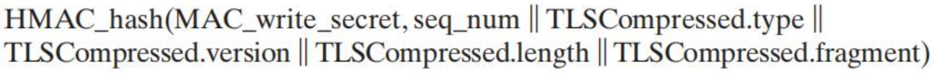
Content type, Major version, Minor version, compressed length

27

Ⅰ.Fragmentation. Each upper-layer message is fragmented into blocks of 2^14 bytes (16,384 bytes) or less.

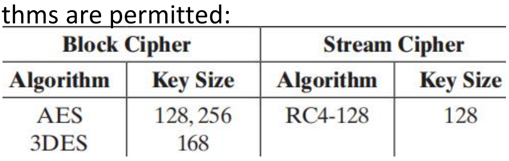
Ⅱ.Compression is optionally applied. Compression must be lossless and may not increase the content length by more than 1024 bytes. In TLSv2, no compression algorithm is specified, so the default compression algorithm is null. 压缩必须是无损的，并且内容长度不能超过 1024 字节。在 TLSv2 中，没有指定压缩算法，因此默认的压缩算法为空

Ⅲ.Add MAC.



HMAC_hash(MAC_write_secret, seq_num || TLSCompressed.type || TLSCompressed.version || TLSCompressed.length || TLSCompressed.fragment)

Ⅳ.Encrypted using symmetric encryption.Encryption may not increase the content length by more than 1024 bytes, so that the total length may not exceed 2^14 + 2048. The following encryption algorithms are permitted:



| Block Cipher | | Stream Cipher | |
|---|---|---|---|
| Algorithm | Key Size | Algorithm | Key Size |
| AES<br>3DES | 128, 256<br>168 | RC4-128 | 128 |

For stream encryption 压缩后的消息加上 MAC 都是加密的。请注意，MAC 是在加密发生之前计算的，然后 MAC 与明文或压缩明文一起加密

For block encryption,可以在加密之前在 MAC 之后添加填充。填充的形式是多个填充字节，后面跟着一个字节，表示填充的长度。填充可以是任何结果的密码块长度的倍数，最多 255 字节。例如，如果密码块长度为 16 字节（例如，AES)，如果明文本（或使用压缩文本）加

MAC 加填充长度字节为 79 字节，则填充长度（以字节计）可以为 1、17、33 等等，最多可达 161。当填充长度为 161 时，总长度为 79 + 161 = 240。

Ⅴ.Append TLS record header.Content type(8bits 用于处理封闭的碎片的高级协议 TLS 结构的前三个＋应用数据 HTTP 等), Major version(8 TLSv2=3), Minor version(8 TLSv2=1), compressed length(16 最大 2^14 + 2048)

## (7)Alert Protocol

<1>Used to convey TLS-related alerts to the peer entity. 用于向对等实体传递 TLS 相关的警报与其他使用 TLS 的应用程序一样，警报消息按照当前状态指定进行压缩和加密
<2>Consist of two bytes
The first byte flags a Warning or Fatal 第一个字节标记警告或致命的
The second byte contains the code that indicates the specific alert,
第二个字节包含指示特定警报的代码
e.g. bad_record_mac/ handshake_failure /decryption_failed /bad_certificate; etc

## (8)Heartbeat protocol

<1>A heartbeat is a periodic signal generated to indicate normal operation or to synchronise. 心跳是一个为指示正常操作或同步而产生的周期性信号
<2>A heartbeat protocol is typically used to monitor the availability of a protocol entity. (In the specific case of TLS, a heartbeat protocol was defined in 2012 in RFC6250.) 心跳协议通常用于监视协议实体的可用性。
<3>Two purposes:
1. assures the sender that the recipient is still alive, even though there may not be any activity for a while. 确保发件人的收件人仍然活着，即使在一段时间内没有任何活动
2. avoid closure by a firewall that does not tolerate idle connections.避免被不允许空闲连接的防火墙关闭

## (9)TLS attacks

<1>Attacks on the handshake protocol
<2>Attacks on the record and application data protocols
– chosen-plaintext attack, session hijacking 被选择的明文攻击，会话劫持
<3>Attacks on the PKI
<4>Other attacks – Heartbleed (buffer over-read) 心脏出血（缓冲区过分读取）

## (10)TLS 1.3

<1>Various attacks on earlier versions of TLS resulted in a series of fixes having to be proposed, which is not desirable.对早期版本的 TLS 的各种攻击导致了必须提出一系列的修复方案，但这是不可取的
<2>The Handshake Protocol is somewhat inefficient. 握手协议有些低效。
<3>Major revision of TLS resulted in TLS 1.3 published in August 2018 (RFC8446).
<4>What is new in TLS 1.3:

– Perfect Forward Secrecy. removing support for key establishment based on RSA and mandating the use of Ephemeral Diffie–Hellman.完美的向前保密。取消对基于 RSA 的密钥建立的支持，并强制使用短暂的 D-H

– New Handshake Protocol. only requiring one full round trip between client and server. More of the data exchanged in the new Handshake Protocol is encrypted.新的握手协议。只需要在客户端和服务器之间进行一次完整的往返旅行。在新的握手协议中交换的更多数据被加密了。

– Authenticated encryption modes. Encryption in TLS 1.3 must be conducted using an authenticated-encryption mode of a block cipher. 经过身份验证的加密模式。TLS 1.3 中的加密必须使用块密码的身份验证加密模式进行。

## 二. Email Security

• Basic requirements

– Confidentiality – Authentication – Integrity

• Other requirements

– Non-repudiation 不可拒绝 – Proof of submission 提交证明

– Proof of delivery 递送证明 – Anonymity 匿名

– Revocability 可撤回 – Resistance to traffic analysis 交通阻力分析

### 1.Pretty Good Privacy (PGP)

#### (1)The services provided by PGP

Actual operations of PGP consist of **five services**:

• **Authentication** - DSS/SHA or RSA/SHA

• **Confidentiality** – CAST5 or IDEA or RSA or 3DES

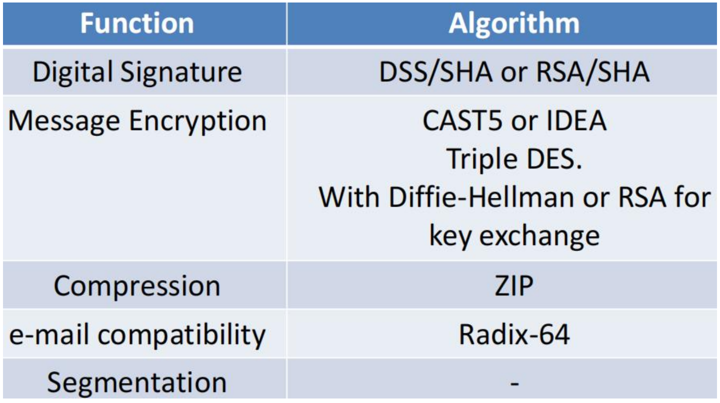
• **Compression**: A message may be compressed, for storage or transmission using ZIP

• **E-mail compatibility**: To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII using Radix-64

为了为电子邮件应用程序提供透明度，可以使用 Radiz-64 将加密的消息转换为 ASCII

• **Segmentation**: To accommodate maximum message size limitations, PGP performs segmentation and reassembly. 为了适应最大的消息大小限制，PGP 执行分割和重组

• The latest PGP services:



| Function | Algorithm |
|---|---|
| Digital Signature | DSS/SHA or RSA/SHA |
| Message Encryption | CAST5 or IDEA<br>Triple DES.<br>With Diffie-Hellman or RSA for key exchange |
| Compression | ZIP |
| e-mail compatibility | Radix-64 |
| Segmentation | - |

##### <1>Authentication

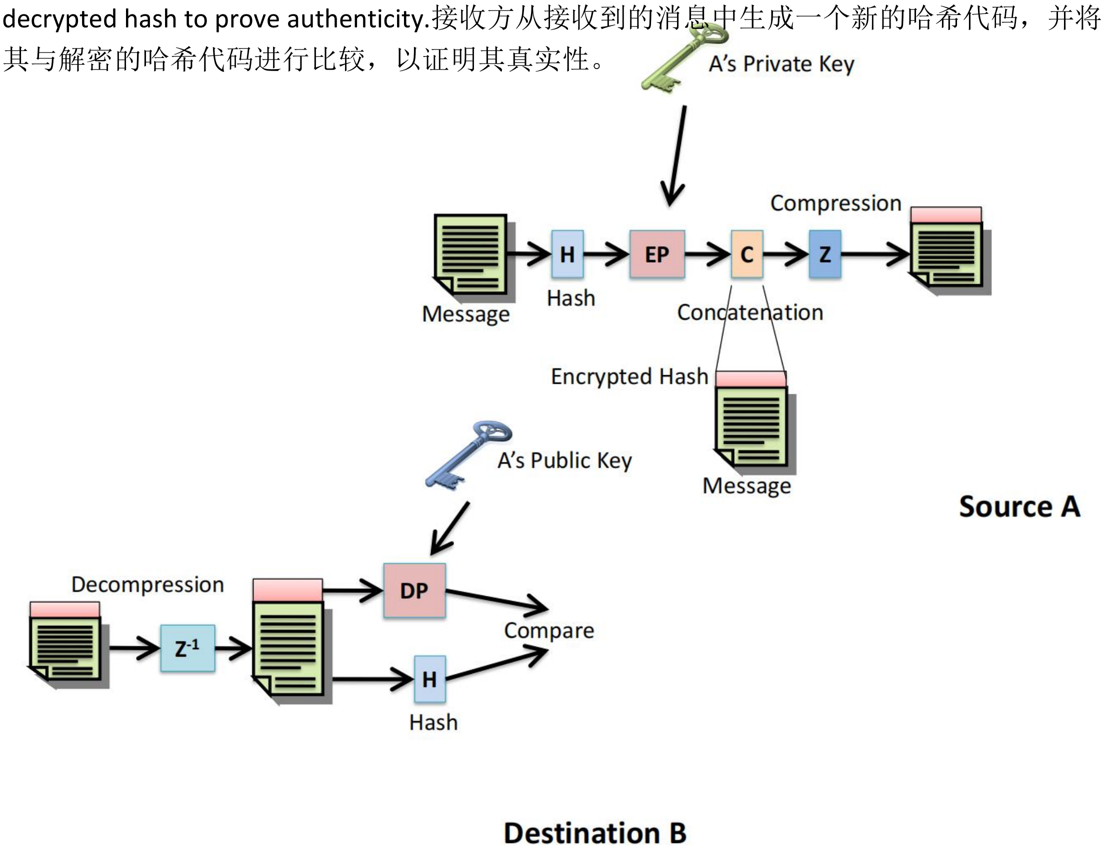
• Create a message.

• SHA used to generate 160-bit hash code.

• Hash encrypted with RSA using sender’s private key, the result is prepended to the message.

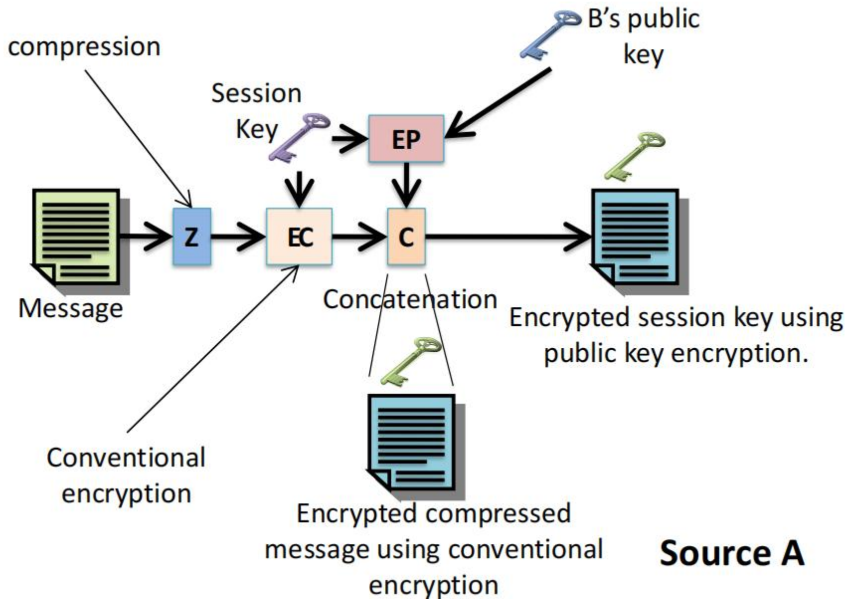
使用发件人的私钥用RSA加密哈希，结果被放在消息的前面

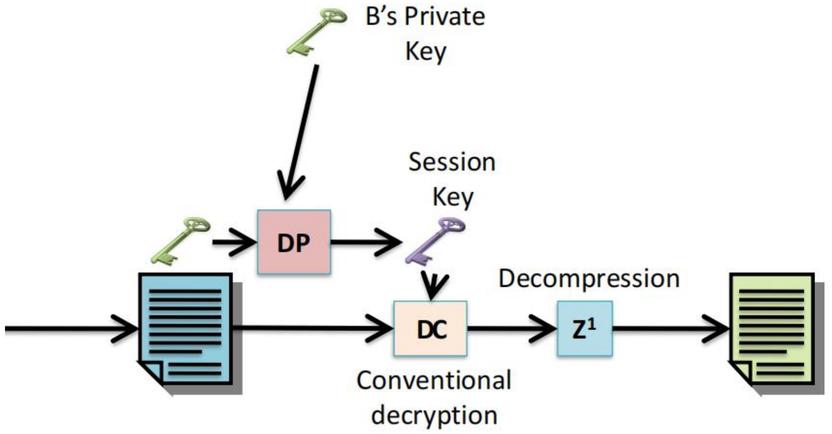
- The receiver uses RSA with sender’s public key.接收方使用带有发送方公钥的RSA
- The receiver decrypts the hash with the sender’s public key.
- The receiver generates a new hash code from the received message and compares it with the decrypted hash to prove authenticity.接收方从接收到的消息中生成一个新的哈希代码，并将其与解密的哈希代码进行比较，以证明其真实性。



## <2>Confidentiality

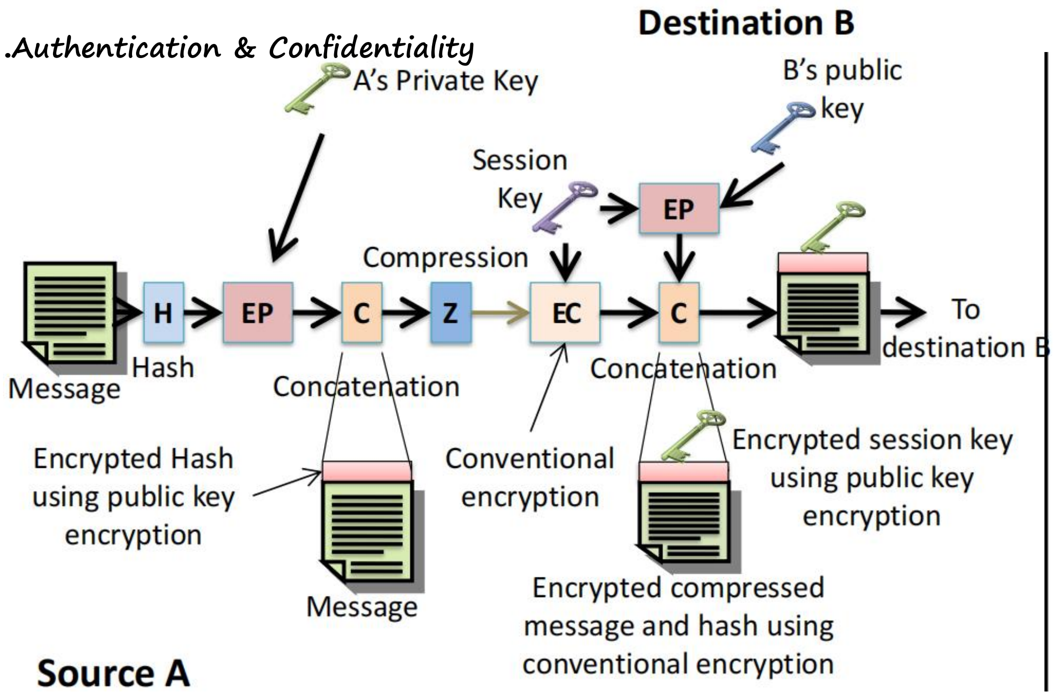
- Sender generates a message.
- Sender generates a session key (128-bits long).
- Message is encrypted using CAST5 (or IDEA, 3DES) using the session key.
- The session key is encrypted using RSA and the recipient’s public key. The encrypted session key is prepend to the message.
- The receiver uses RSA and its’ private key to recover the session key.
- Using the session key, the receiver decrypts the message.


compression
B’s public key
Session Key
EP
Z
EC
C
Message
Concatenation
Encrypted session key using public key encryption.
Conventional encryption
Encrypted compressed message using conventional encryption
Source A

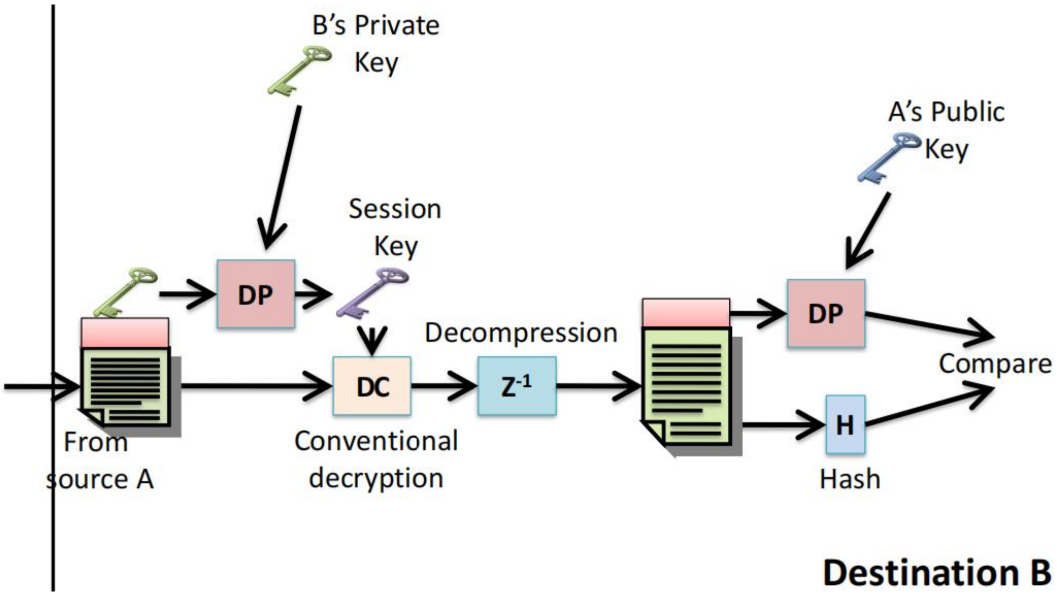


注意合起来：

Ⅰ.*Authentication & Confidentiality*



Ⅱ.*Confidentiality & Authentication*



### <3>Compression

Ⅰ.To save space both for e-mail and storage as a default PGP compresses the message after applying the signature but before encryption.
要将电子邮件和存储空间保存为默认 PGP，请在应用签名之后但在加密之前压缩消息。

Ⅱ.If the message was first compressed and then signed then for future verification

如果消息首先被压缩，然后签名，然后将来验证
– A compressed version of the document has to be stored, or 必须存储该文档的压缩版本
– Re-compress the message when verification is required.或者在需要验证时重新压缩消息

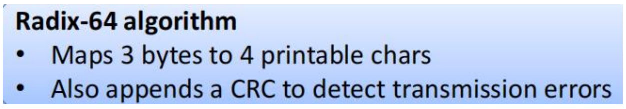
Ⅲ.But more relevantly, a compression algorithm is not deterministic.

– That is the same message when compressed can produce different compressed forms (this depends on running speed vs compression ratio).
当压缩可以产生不同的压缩形式时，这是相同的消息（这取决于运行速度与压缩比）
– If sender and receiver use different settings for the compression algorithm they obtain different forms. This makes authentication difficult. 如果发送方和接收方对压缩算法使用不同的设置，它们会获得不同的形式。这使得身份验证很困难

Ⅳ.The compressed message has less redundancy so is more difficult to cryptanalysis. 压缩消息的冗余度较少，因此很难进行密码分析。

### <4>E-mail compatibility 兼容性

• When PGP transmits a message at least part of the message is encrypted. The encrypted part (can be the whole document) consist of a stream of 8-bit octets. 当 PGP 传输一条消息时，至少有一部分消息被加密。加密部分（可以是整个文档）由一个 8 位八进制流组成。
• Many electronic mail systems only permit the use of blocks consisting of ASCII text.
许多电子邮件系统只允许使用由 ASCII 文本组成的块
• PGP converts the raw 8-bit octets stream to a stream of printable ASCII characters using radix-64 expansion.
PGP 使用 radix-64 扩展将原始的 8 位八位制流转换为可打印的 ASCII 字符流

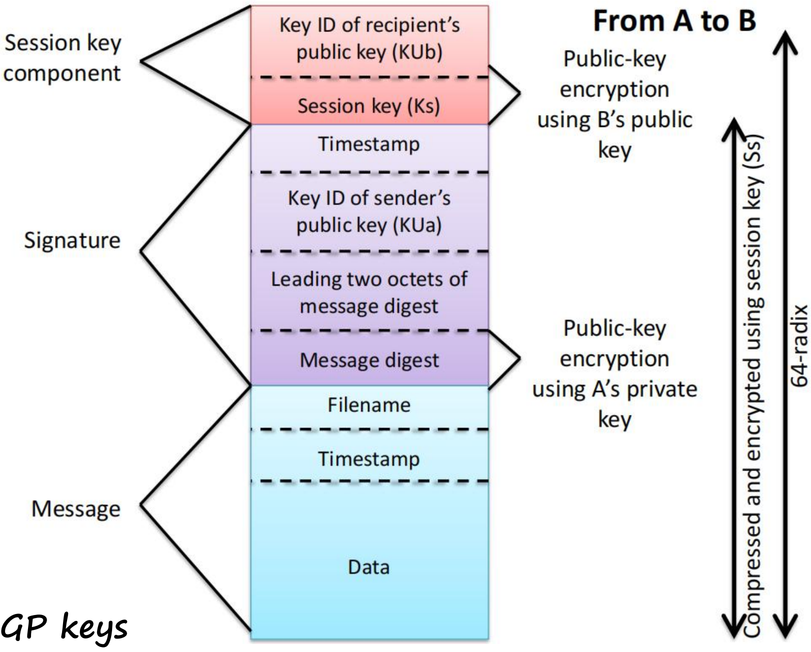


**Radix-64 algorithm**
- Maps 3 bytes to 4 printable chars
- Also appends a CRC to detect transmission errors

• PGP also segments messages if too big. 如果信息太大，PGP 也会分割信息

### <5>General Format: Compressed and encrypted using session key (Ss) 64-radix

**一般格式：使用会话密钥（Ss）进行压缩和加密**


Session key component
Signature
Message
Key ID of recipient's public key (KUb)
Session key (Ks)
Timestamp
Key ID of sender's public key (KUa)
Leading two octets of message digest
Message digest
Filename
Timestamp
Data
From A to B
Public-key encryption using B's public key
Public-key encryption using A's private key
Compressed and encrypted using session key (Ss)
64-radix

## (2)PGP keys

- There are four types of keys

– Pass-phrase key 通过短语密钥

– Session keys (random keys generated)会话密钥（生成的随机密钥）

– Public-key

– Private-key

- PGP allows to have more than one set of private-public keys per user.

  PGP 允许每个用户拥有超过一组 private-public keys

### <1>Key management

- Because of the possibility of multiple public–private keys per user, the recipient of the message needs to know which of his/hers public key was used for encryption.

由于每个用户可能有多个公私钥，消息的接收者需要知道哪一个公钥用于加密

- One possibility is that the sender of the message includes the public key of the recipient, but it is unnecessarily wasteful of space.

一种可能性是，消息的发送者包含了收件人的公钥，但这是不必要的空间浪费

- PGP send an identifier of the recipients public key.
- PGP assigns an ID to each public key by using the least significant 64 bits of the key. (ID → KUA mod(2^64)) PGP 通过使用最小的 64 位为每个公钥分配一个 ID

### <2>Key rings

- One or more keys stored together constitute a key-ring.

一个或多个钥匙存储在一起，构成了一个钥匙环

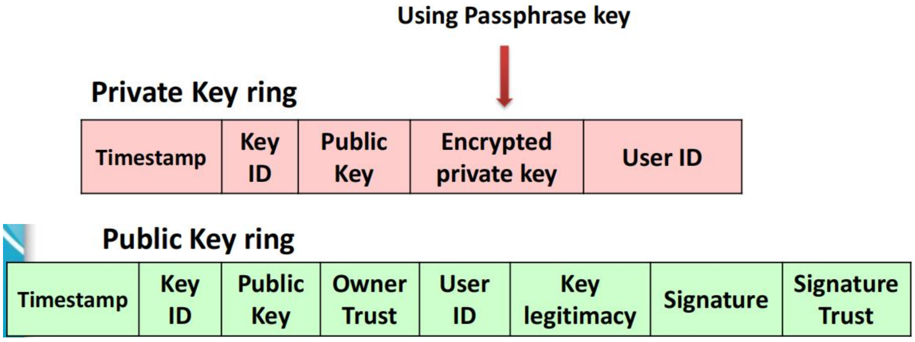
- There are two classes:

**– Private-key ring**: Stores the private/public key pairs owned at this node.

存储在此节点上拥有的私钥/公钥对

**– Public-key ring**: Stores the public keys of other users known at this node.

存储在此节点上已知的其他用户的公钥



Using Passphrase key

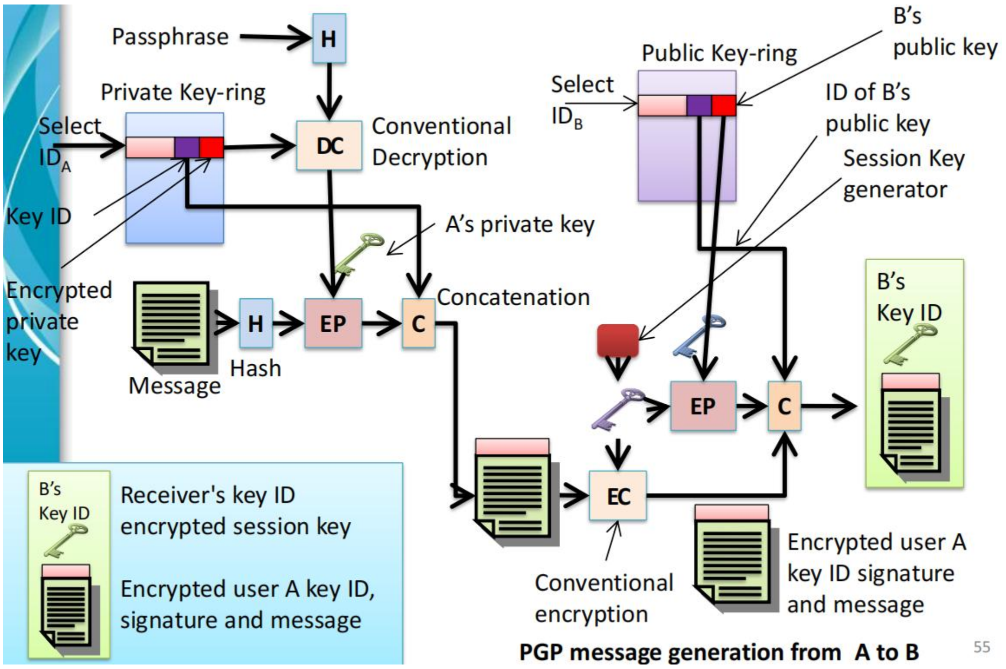
Private Key ring

| Timestamp | Key ID | Public Key | Encrypted private key | User ID |
|---|---|---|---|---|

Public Key ring

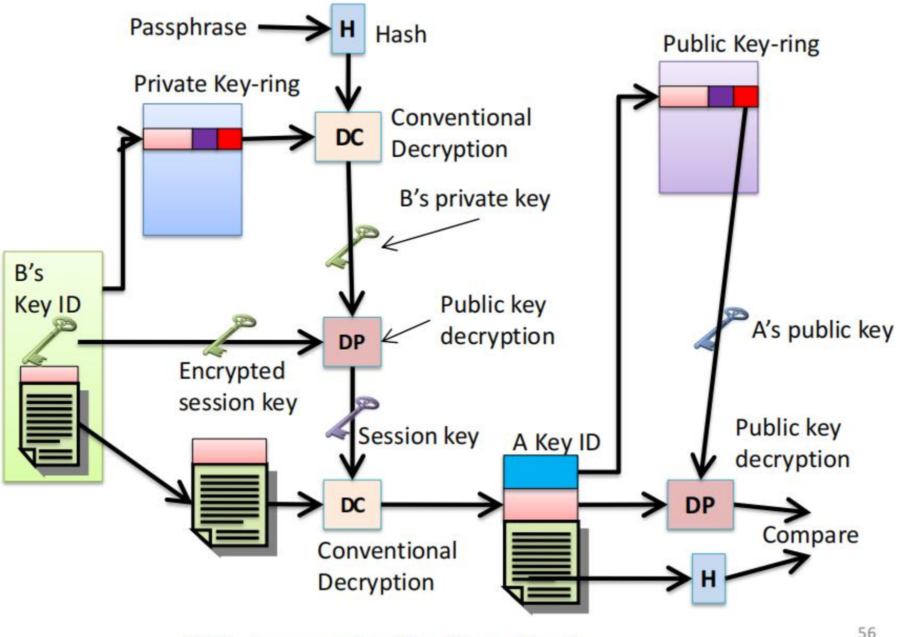
| Timestamp | Key ID | Public Key | Owner Trust | User ID | Key legitimacy | Signature | Signature Trust |
|---|---|---|---|---|---|---|---|

## <3>PGP message generation PGP 消息生成



PGP message generation from A to B

## <4>PGP message reception PGP 消息接收



PGP message reception from A to B

## <5>PGP Public-key management

• The problem: A’s public-key ring contains a public key attributed to B, but how can A be sure the public key is from B, not someone else?

A 的公钥环包含一个属于 B 的公钥，但是 A 如何确定公钥来自 B，而不是其他人呢？

• The solution:

– PGP does not include any specification for establishing certifying authorities

PGP 不包括任何关于建立认证机构的规范

– adopts a different trust model – the “web of trust”

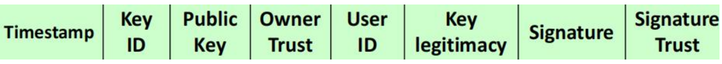
采用了一种不同的信任模式——“信任之网”

– Individuals sign one another’s public keys (the “signature field” in the public-key ring) and create an interconnected community of public-key users. 个人对彼此的公钥（公钥环中的“签名字段”）进行签名，并创建一个相互关联的公钥用户社区

– PGP computes a trust level for each public key in key ring

PGP 为密钥环中的每个公钥计算一个信任级别

## <6>Trust (public-key ring)



| Timestamp | Key ID | Public Key | Owner Trust | User ID | Key legitimacy | Signature | Signature Trust |
|---|---|---|---|---|---|---|---|

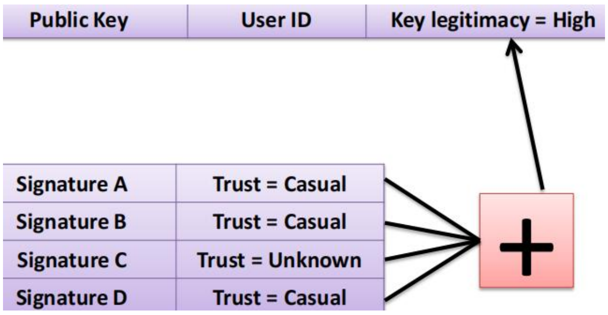
• **Key legitimacy** → indicates the extent to which PGP will trust that this is a valid public key for this user; the higher the level of trust, the stronger is the binding of this user ID to this key. This field is computed by PGP. 密钥合法性→指示 PGP 对该用户的有效公钥的信任程度；信任级别越高，该用户 ID 与此密钥的绑定就越强。这个字段是由 PGP 计算出来的

• **Owner Trust** → indicates the degree to which this public key is trusted to sign other public-key certificates; this level of trust is assigned by the user 所有者信任→表示对此公钥签署其他公钥证书的信任程度；此信任级别由用户分配

• **Signature Trust** → indicates the degree to which this PGP user trusts the signer to certify public keys. The key legitimacy field is derived from the collection of signature trust fields in the entry. 签名信任→表示此 PGP 用户信任签名者认证公钥的程度。密钥合法性字段来源于条目中的签名信任字段的集合

• Example UserID = politician – Owner Trust = low – Key legitimacy = high

## <7>Example of how PGP compute key legitimacy



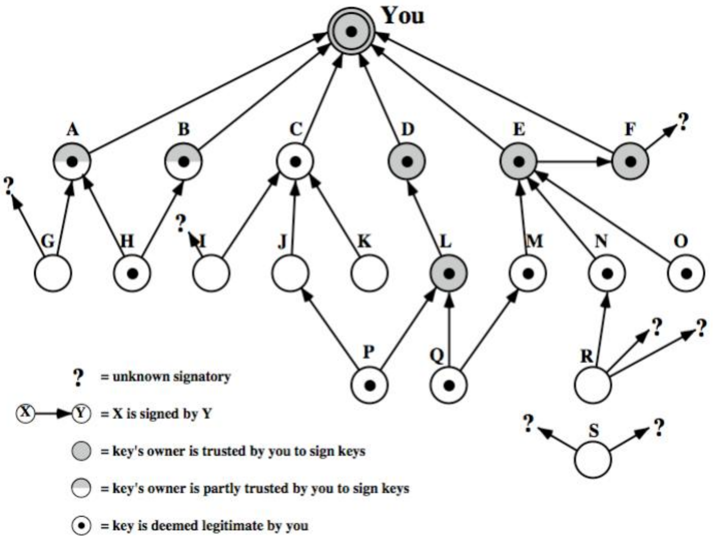
| Public Key | User ID | Key legitimacy = High |
|---|---|---|

| | |
|---|---|
| Signature A | Trust = Casual |
| Signature B | Trust = Casual |
| Signature C | Trust = Unknown |
| Signature D | Trust = Casual |

## <8>PGP Trust Model Example

The structure of a public-key ring where the user has acquired a number of public keys, some directly from their owners and some from a third party such as a key server.

公钥环的结构，用户获得了许多公钥，一些直接从他们的所有者，另一些从第三方，如密钥服务器


You
A
B
C
D
E
F
?
?
G
H
?
I
J
K
L
M
N
O
P
Q
R
?
?
?
S
?
? = unknown signatory
X → Y = X is signed by Y
= key's owner is trusted by you to sign keys
= key's owner is partly trusted by you to sign keys
= key is deemed legitimate by you

整体结构提出了去中心化环境中信任传播的概念，其中对公钥的信任是通过网络中其他人的签名建立的。网络中的距离越近，您就越有可能信任该密钥，因为通往该密钥的信任路径越多. 问号表示未知签名者，这意味着有您不直接信任或不认可的人的认可

## 2.S/MIME Secure/Multipurpose Internet Mail Extension

### (1)介绍

• S/MIME is a security enhancement for the MIME Internet e–mail format standard based on technology from RSA Data Security. S/MIME 是一种基于 RSA 数据安全技术的 MIME 互联网电子邮件格式标准的安全增强。

• MIME provides a convenient mechanism for transferring composite data.
MIME 为传输复合数据提供了一种方便的机制

• Binary data handled via base64 encoding. 通过 base64 编码处理的二进制数据。

• S/MIME: security related information sent as sections of a multipart message
S/MIME：作为多部分信息的部分发送的安全相关信息

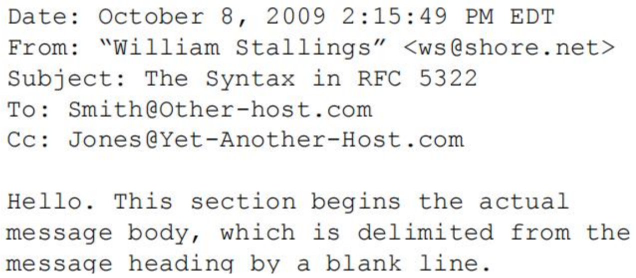
– multipart/signed – multipart/encrypted

### (2)RFC5322

• Defines a format for text messages that are sent using electronic mail
定义了使用电子邮件发送的短信的格式

• A message consists of header lines (the header) followed by unrestricted text (the body). 邮件由标题行（标题）和不受限制的文本（正文）组成。

• This is an example message:



```
Date: October 8, 2009 2:15:49 PM EDT
From: “William Stallings” <ws@shore.net>
Subject: The Syntax in RFC 5322
To: Smith@Other-host.com
Cc: Jones@Yet-Another-Host.com

Hello. This section begins the actual
message body, which is delimited from the
message heading by a blank line.
```

## (3)Multipurpose Internet Mail Extensions (MIME)

<1>MIME is an extension to the RFC5322 framework

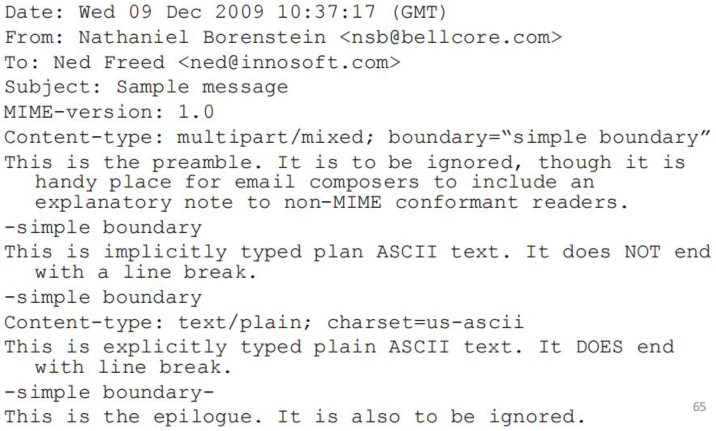
<2>MIME addresses some problems and limitations of the use of SMTP(Simple Mail Transfer Protocol ) and RFC5322, e.g., cannot transmit executable files or other binary objects.
MIME 解决了使用 SMTP（简单邮件传输协议）和 RFC5322 的一些问题和限制，例如，不能传输可执行文件或其他二进制对象

<3>The MIME specification includes the following elements: 规范包括以下要素

– Five new message header fields are defined (The **Content-Type** header field is used for secure communications) 定义了五个新的邮件头字段（Content-Type 头字段用于安全通信）

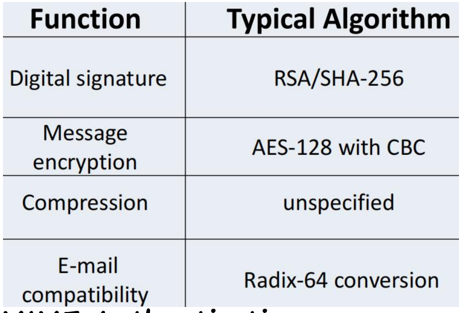
– A number of content formats are defined, which support multimedia electronic mail
定义了许多支持多媒体电子邮件的内容格式

<4>This is an example of multipart type email. The headers of a simple email in MIME looks like this (Taken from RFC 2046):



```
Date: Wed 09 Dec 2009 10:37:17 (GMT)
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Sample message
MIME-version: 1.0
Content-type: multipart/mixed; boundary=“simple boundary”
This is the preamble. It is to be ignored, though it is
   handy place for email composers to include an
   explanatory note to non-MIME conformant readers.
-simple boundary
This is implicitly typed plan ASCII text. It does NOT end
   with a line break.
-simple boundary
Content-type: text/plain; charset=us-ascii
This is explicitly typed plain ASCII text. It DOES end
   with line break.
-simple boundary-
This is the epilogue. It is also to be ignored.
```

## (4)S/MIME services



| Function | Typical Algorithm |
|---|---|
| Digital signature | RSA/SHA-256 |
| Message encryption | AES-128 with CBC |
| Compression | unspecified |
| E-mail compatibility | Radix-64 conversion |

### <1>S/MIME Authentication

1. The sender creates a message
2. SHA-256 is used to generate a 256-bit message digest of the message
3. The message digest is encrypted with RSA using the sender’s private key, and the result is appended to the message. Also appended is identifying information for the signer, which will

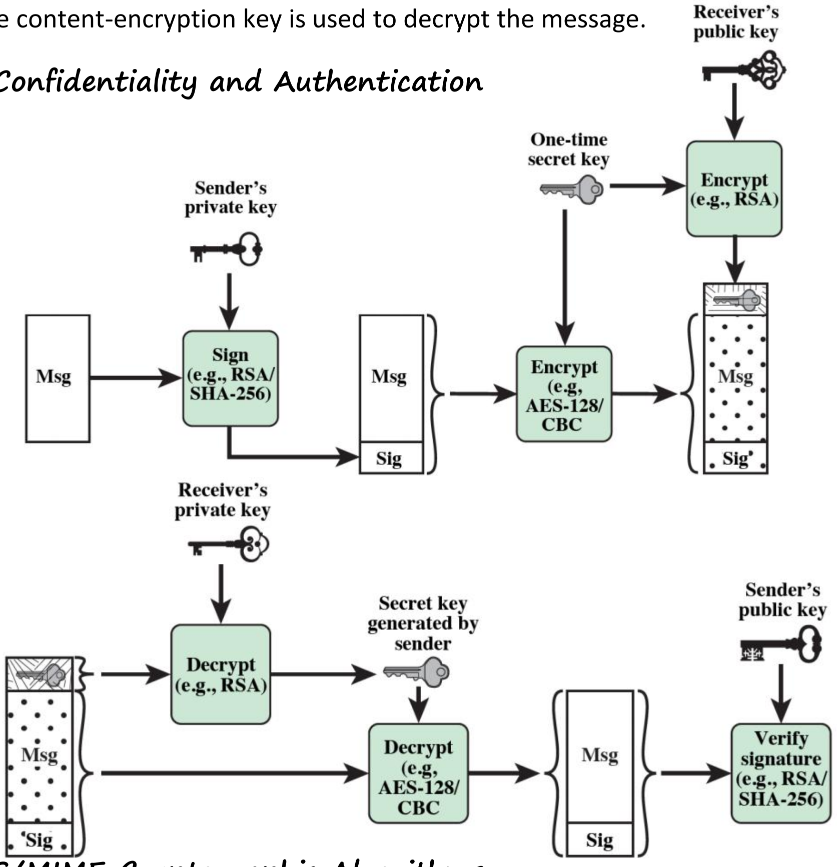
enable the receiver to retrieve the signer’s public key 消息摘要使用发件人的私钥用 RSA 加密，结果将附加到消息中。还附加了签名者的标识信息，这将使接收者能够检索签名者的公钥

4. The receiver then verifies the message digest.

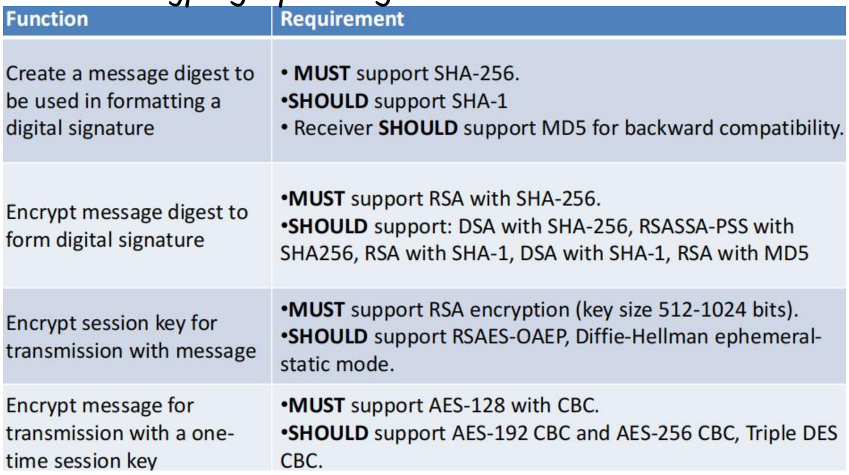
## <2>S/MIME Confidentiality

1. The sender generates a message and a random 128- bit number to be used as a content-encryption key for this message only. 发送方生成一个消息和一个随机的 128 位数字，仅用于此消息的内容加密密钥。
2. The message is encrypted using the content-encryption key.
3. The content-encryption key is encrypted with RSA using the recipient's public key and is attached to the message. 内容加密密钥使用收件人的公钥用 RSA 进行加密，并附加到邮件中。
4. The receiver uses RSA with its private key to decrypt and recover the content-encryption key.
5. The content-encryption key is used to decrypt the message.

## <3>Confidentiality and Authentication



## <4>S/MIME Cryptographic Algorithms



| Function | Requirement |
|---|---|
| Create a message digest to be used in formatting a digital signature | • **MUST** support SHA-256.<br>•**SHOULD** support SHA-1<br>• Receiver **SHOULD** support MD5 for backward compatibility. |
| Encrypt message digest to form digital signature | •**MUST** support RSA with SHA-256.<br>•**SHOULD** support: DSA with SHA-256, RSASSA-PSS with SHA256, RSA with SHA-1, DSA with SHA-1, RSA with MD5 |
| Encrypt session key for transmission with message | •**MUST** support RSA encryption (key size 512-1024 bits).<br>•**SHOULD** support RSAES-OAEP, Diffie-Hellman ephemeral-static mode. |
| Encrypt message for transmission with a one-time session key | •**MUST** support AES-128 with CBC.<br>•**SHOULD** support AES-192 CBC and AES-256 CBC, Triple DES CBC. |

### <5>S/MIME Certificates processing

- Uses public-key certificates that conform to X.509 v3. 使用符合 X.509 v3 标准的公钥证书
- Key management is hybrid between X.509 and PGP's web of trust.

密钥管理是 X.509 和 PGP 的信任网络之间的混合体

- Each client has a list of trusted CA's certificates and own public/private key pairs & certificates.

每个客户端都有一个受信任的 CA 证书列表，以及自己的公钥/私钥对和证书

- Certificates must be signed by trusted CAs (e.g. VeriSign)

### (5)S/MIME Problems

- Earlier versions used mostly crippled crypto. 早期版本大多使用瘫痪加密
- S/MIME cracking screen saver released 1997.
- Original S/MIME based on patented RSA and proprietary RC2 (rejected as a standard).
- S/MIME v3 uses strong crypto and nonpatented, non-proprietary technology.

# 三. Threats to Security

## 1.Malicious Software 恶意软件

(1)Introduction

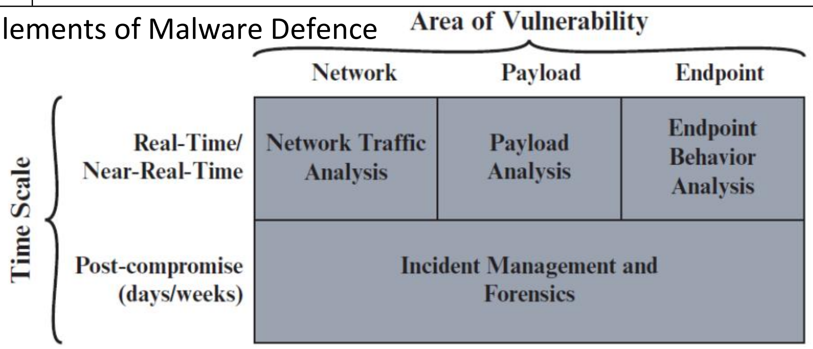
- Commonly called malware, is perhaps the most significant security threat to organizations
- Definition: "a program that is covertly inserted into another program with the intent to destroy data, run destructive or intrusive programs, or otherwise compromise the confidentiality, integrity, or availability of the victim's data, applications, or operating system"秘密插入另一个程序，意图破坏数据、运行破坏性或侵入性程序，或以其他方式损害受害者数据、应用程序或操作系统的机密性、完整性或可用性的程序
- Malware can pose a threat to application programs
- Malware can also be used on compromised or malicious Web sites and servers, or in spam emails or other messages, which aim to trick users into revealing sensitive personal information 恶意软件也可以用于被泄露或恶意的网站和服务器，或垃圾邮件或其他信息，旨在欺骗用户泄露敏感的个人信息

(2)Types of Malicious Software

| Term | Description |
|---|---|
| Virus | Malware that, when executed, tries to replicate itself into other executable code; when it succeeds the code is said to be infected. When the infected code is executed, the virus also executes.恶意软件，当执行时，试图复制自己到其他可执行代码；当它成功时，代码被感染。当执行被感染的代码时，病毒也会被执行 |
| Worm | A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network. 一种可以独立运行并可以将自己的完整工作版本传播到网络上的其他主机上的计算机程序<br>The main differences between viruses and worms is that the worms can self-replicate and propagate without human interaction and that the worm does not integrate into existing code. 病毒和蠕虫之间的主要区别是，蠕虫可以在没有人类互动的情况下进行自我复制和传播，而且蠕虫没有整合到现有的代码中。 |
| Trojan | A computer program that appears to have a useful function, but also has a hidden and |

| | |
|---|---|
| horse | potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program.一种计算机程序，似乎有一个有用的功能，但也有一个隐藏的和潜在的恶意功能，逃避安全机制，有时通过利用调用特洛伊木马程序的系统实体的合法授权。 |
| Spyware | Software that is secretly installed into an information system to gather information on individuals or organizations without their knowledge 秘密安装到信息系统中的软件，用来在个人或组织不知情的情况下收集其信息的软件 |
| Rootkit | A set of tools used by an attacker after gaining root-level access to a host, to conceal the attacker’s activities on the host and permit the attacker to maintain root-level access.攻击者在获得对主机的根级访问权限后使用的一组工具，用来隐藏攻击者在主机上的活动，并允许攻击者维护根级访问权限 |
| Backdoor | Usually installed by the attackers or by a malware program, a backdoor is a program that has the ability to bypass a system’s security control, allowing an attacker to access the system stealthily 后门通常由攻击者或恶意软件程序安装，后门是一种能够绕过系统安全控制的程序，允许攻击者秘密访问系统 |
| Bot (Zombie) | Program that is installed on a system to launch attacks on other machines. A collection of bots that act in concert is referred to as a botnet 安装在系统上的用来对其他机器发动攻击的程序。一个协同行动的机器人的集合被称为僵尸网络 |

### (3)Five Elements of Malware Defence


Area of Vulnerability
Network
Payload
Endpoint
Time Scale
Real-Time/ Near-Real-Time
Network Traffic Analysis
Payload Analysis
Endpoint Behavior Analysis
Post-compromise (days/weeks)
Incident Management and Forensics

### (4)Malware Defence

- Network Traffic Analysis: involves monitoring traffic flows to detect potentially malicious activity, involves misuse detection or anomaly detection. 网络流量分析：包括监控流量流以检测潜在的恶意活动，涉及误用检测或异常检测。
- Payload Analysis: involves looking for known malicious payloads or looking for payload patterns that are anomalous 有效负载分析：包括寻找已知的恶意有效负载或寻找异常的有效负载模式
- Endpoint Behavior Analysis: involves a wide variety of tools and approaches implemented at the endpoint, i.e. antivirus software, application whitelisting.端点行为分析：涉及到在端点上实现的各种工具和方法，即杀病毒软件、应用程序白名单

## 2.Distributed DoS Attacks

### (1)Distributed Denial of Service Attacks

- A denial of service (DoS) attack is an attempt to prevent legitimate users of a service from using that service. When this attack comes from a single host or network node, then it is simply referred to as a DoS attack. DoS 攻击是指试图阻止服务的合法用户使用该服务。当此攻击来自单个主机或网络节点时，它就被简单地称为 DoS 攻击
- DDoS Attack make computer systems inaccessible by flooding servers, networks, or even end-user systems with useless traffic so that legitimate users can no longer gain access to those

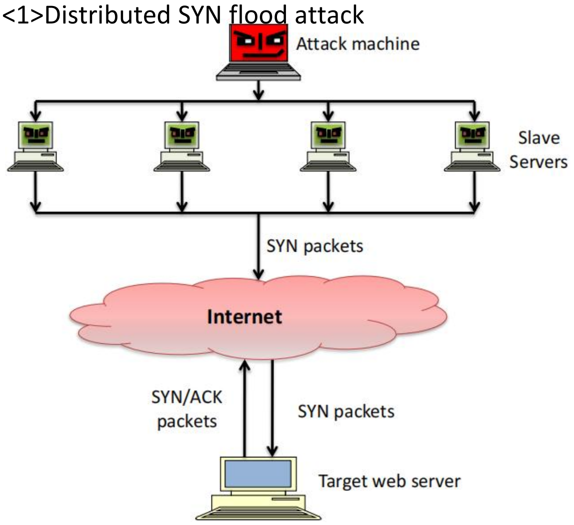
resources DDoS 攻击使计算机系统无法被服务器、网络、甚至是流量无用的终端用户系统访问，从而使合法用户无法再访问这些资源

- In a typical DDoS attack, a large number of compromised hosts are amassed to send useless packets 在典型的 DDoS 攻击中，大量损坏的主机发送无用的数据包
- DDoS attack examples

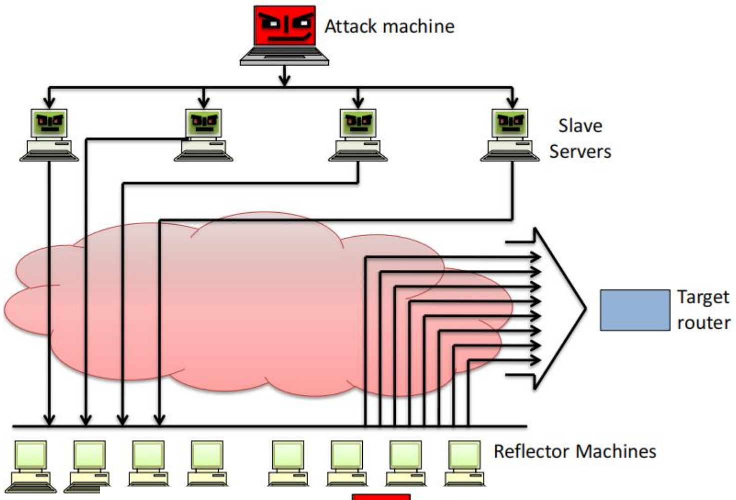
– Internal Resource Attack (SYN or ICMP)

– Direct or Reflector flooding attack

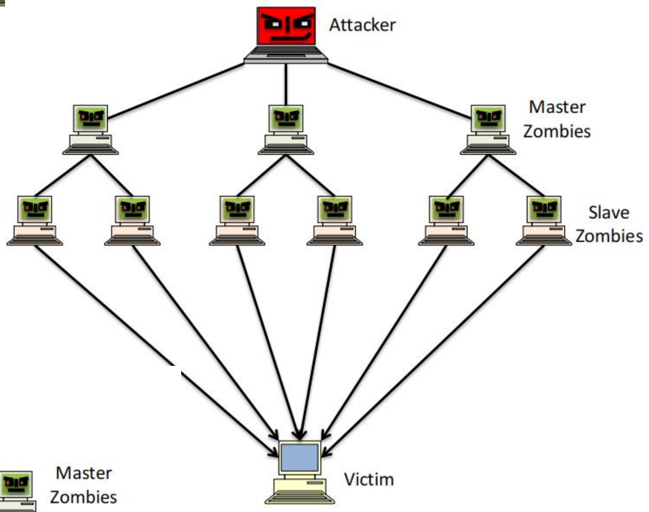
<1>Distributed SYN flood attack

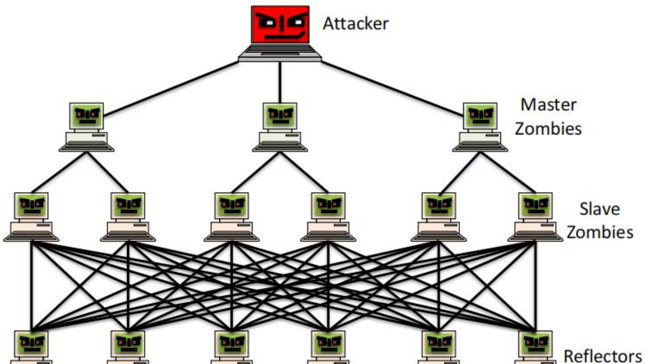


<2>Distributed ICMP attack



<3>Direct DDoS attack (flooding based)





<4>Reflector DDoS Attack (flooding based)

(2)DDoS Countermeasures 对策

• **Attack prevention and pre-emption** (before the attack) Techniques include enforcing policies for resource consumption and providing backup resources available on demand

攻击预防和抢占（攻击前）技术包括强制执行资源消耗策略和按需提供可用的备份资源

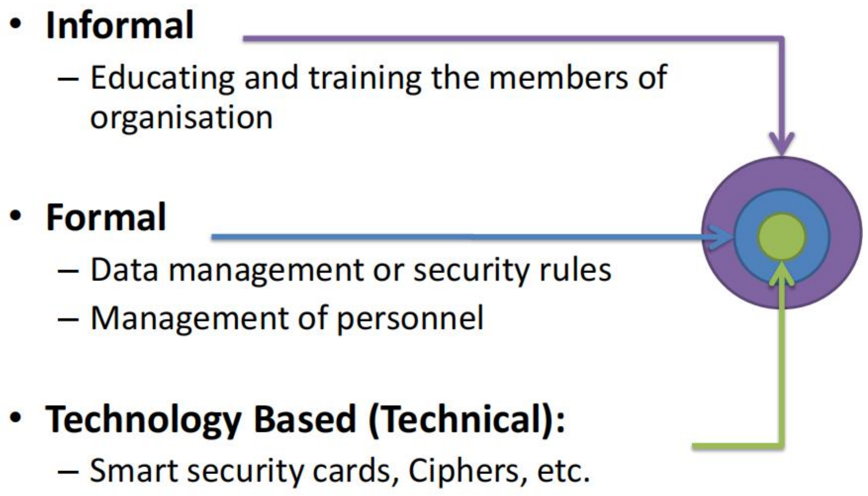
• **Attack detection and filtering** (during the attack) These mechanisms attempt to detect the attack as it begins and respond immediately

攻击检测和过滤（在攻击期间）这些机制试图在攻击开始时检测到攻击并立即作出响应

• **Attack source trace-back and identification** (during and after the attack) This is an attempt to identify the source of the attack as a first step in preventing future attacks.

攻击源追溯和识别（在攻击期间和之后）这是一种试图识别攻击源，作为防止未来攻击的第一步

## 四．*Information Systems Security*



- **Informal**
  - Educating and training the members of organisation
- **Formal**
  - Data management or security rules
  - Management of personnel
- **Technology Based (Technical):**
  - Smart security cards, Ciphers, etc.