Unit 3
Electronic Mail Security

1

電子郵件弱點

- 電子郵件發展歷史
  - 電子郵件翻譯自英文的email或e-mail
  - 早在網際網路流行以前，電子郵件就已經存在了
  - 現在已經演變成為一個更加複雜並豐富得多的系統
  - 網際網路擴展了其應用的範圍
  - 使用在支援TCP/IP協定或具有SMTP和POP的網路

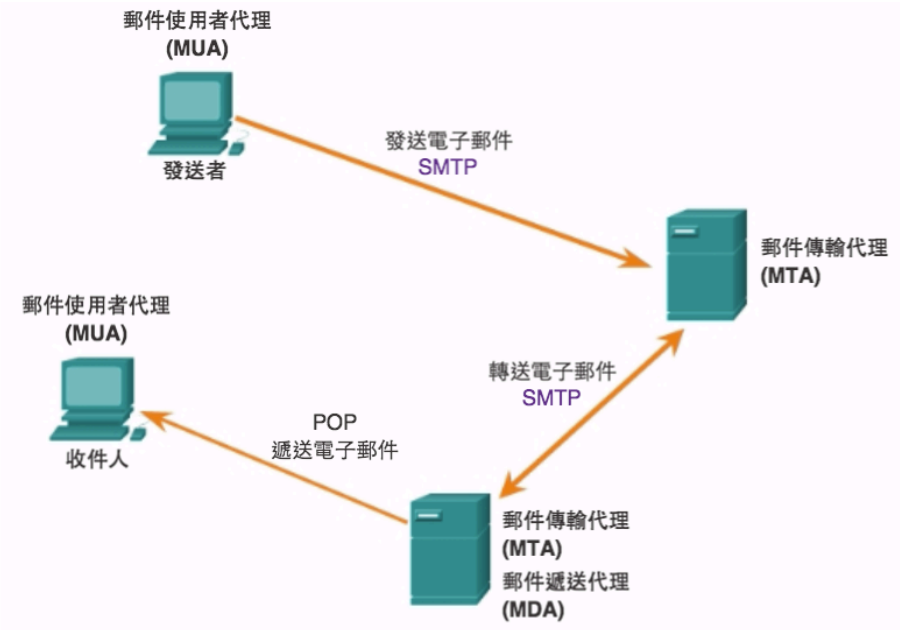
電子郵件的誕生

- 電子郵件的發明人雷.湯姆林森(Ray Tomlinson)研製出一套新程式，改善以往傳遞資訊的缺點
  - 可輕易透過電腦網路發送和接收資訊
  - 為了易識別的電子郵箱位址，決定採用@符號，符號前面加用戶名，後面加用戶郵箱所在的地址

電子郵件的30年發展歷程

- 電子郵件是在70年代發明的，它卻是在80年才得以興起
  - 70年代的沉寂主要是網路人口太少，網路的速度太慢
  - 80年代中期，電子郵件開始在電腦迷以及大學生中廣泛傳播開來
  - 90年代中期，全球上網人數激增，電子郵件被廣為使用

電子郵件的傳送 (1/5)

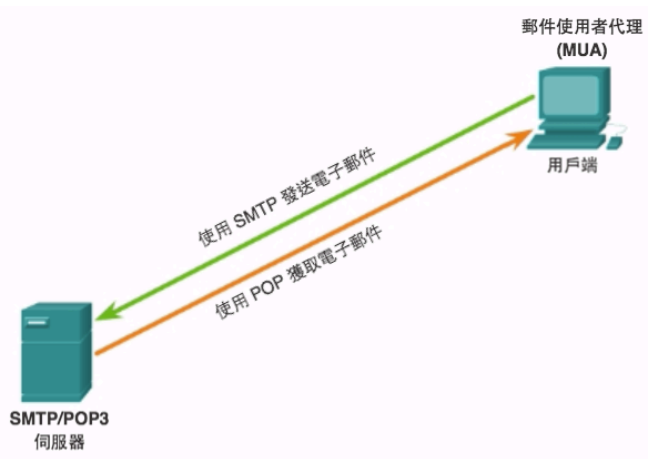


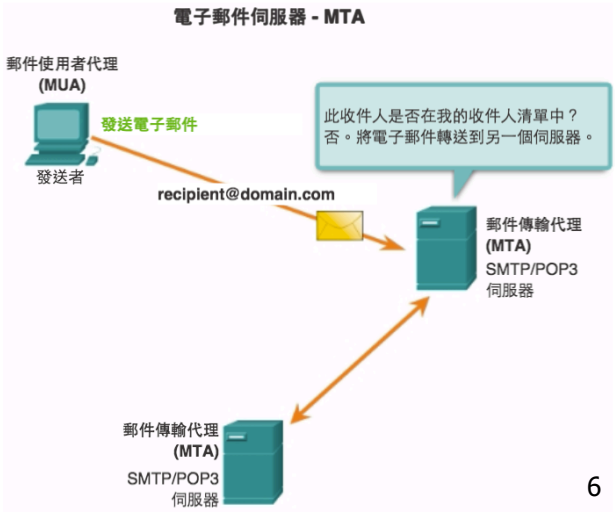
SMTP 用於將電子郵件從用戶端發送到伺服器和在電子郵件伺服器之間轉送電子郵件。

POP 用於遞送電子郵件。

電子郵件的傳送 (2/5)

- MUA (Mail User Agent)：MUA 就是『郵件使用者代理人』
  - 例子：Windows 裡面的 OutLook，Netscape 裡面的 mail 功能與 KDE 裡面的 Kmail 都是 MUA





電子郵件的傳送 (3/5)

- MTA (Mail Transfer Agent) ： MTA 就是郵件伺服器，『郵件傳送代理人』的意思。主要功能有：
  - 收受外部主機寄來的信件
  - 幫使用者傳送 (寄出) 信件
  - 讓使用者自己的信可以收回去

電子郵件的傳送 (4/5)

- MDA (Mail Delivery Agent)
  - 將 MTA 所收受的信件，依照信件的流向 (送到哪裡去) 來將該信件放置到本機帳戶下的郵件檔案中 (Mailbox) !
  - 如果信件的流向是到本機當中時，這個郵件代理人的功能還具有郵件分析 (filtering) 與其他相關的功能。

電子郵件的傳送 (5/5)

- Mailbox
  - 『郵件信箱』就是在主機上面的一個目錄下某個人『專用』的信件收受檔案。
  - 以 UNIX 來說，系統管理員 root ，有個信箱在 /var/spool/mail/root 。
  - 當 MTA 收到 root 的信時，就會將該封信件存到 /var/spool/mail/root 這個檔案中。

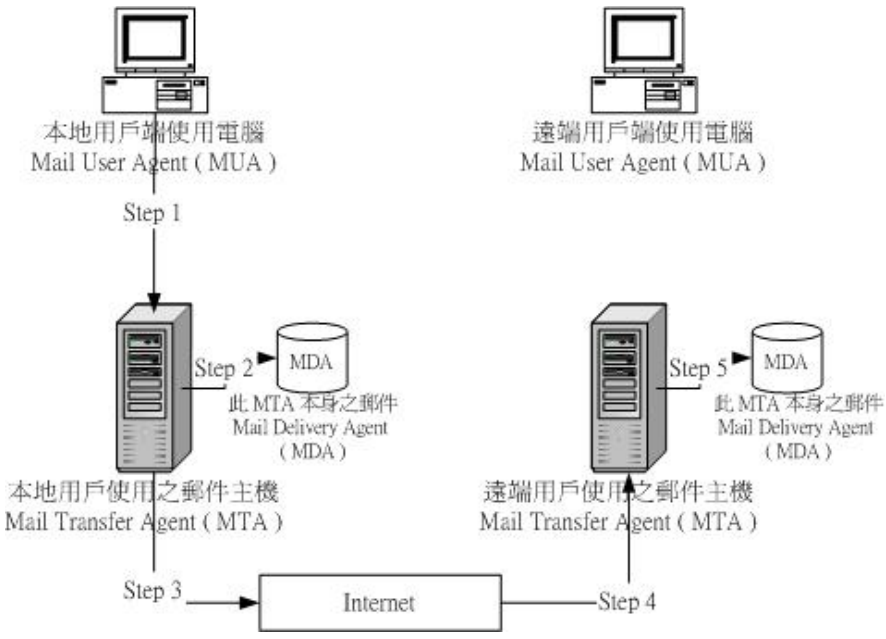
使用的協定– SMTP

- 郵件主機使用 SMTP (Simple Mail Transfer Protocol) 這個協定，port number 為 25 。
- 寄信時，MUA 主動連接 smtp 協定 (port 25) 而送出去。
- 郵件主機 MTA 在轉遞的時，也是經下一部 MTA 的 smtp 協定 (port 25)來將信送出去。

使用的協定– POP3

- 收信是 MUA 經由 POP (Post Office Protocol) 協定來連接 MTA 的使用者 Mailbox
- 目前常用的 POP 協定為 POP3 (Post Office Protocol version 3 ，port number 為 110)
- MUA 經由 MTA 的 port 110 將信件由 MTA 的 mailbox 收到本地端的 MUA 上

如何將信寄出去

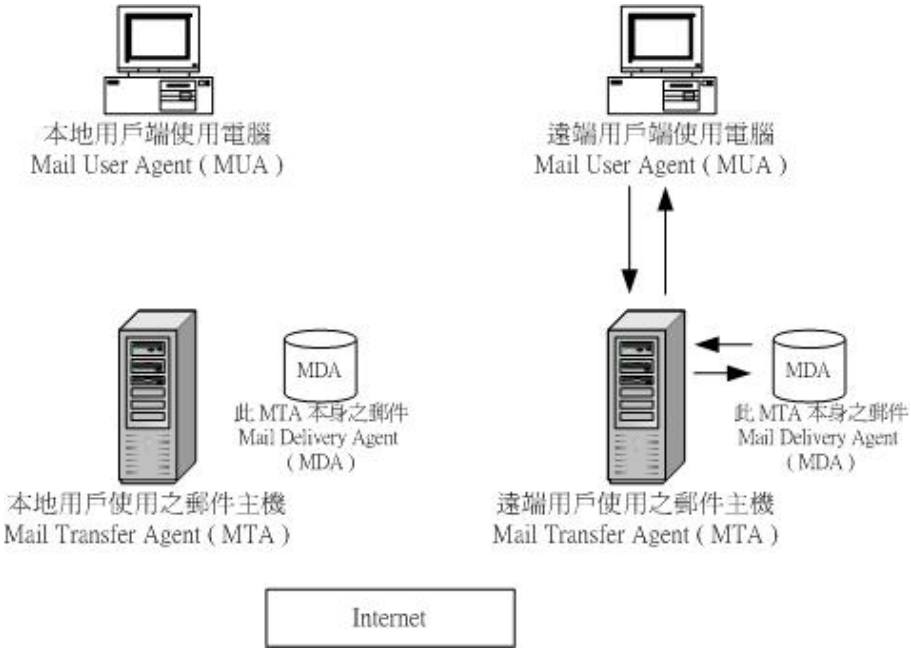


電子郵件以郵件主機寄送信件示意圖

如何將信寄出去

- Step 1： 使用者利用 MUA 寄信到 MTA 上
- Step 2：MTA 收到自己的信件，交由 MDA 發送到該帳號的 MailBox
- Step 3 ：MTA 將信再轉送出去
- Step 4 ：遠端 MTA 收受本地的 MTA 所發出的郵件
- Step 5 ：信件會存放在遠端的 MTA 上面

收信的動作 (1/2)



用戶端收受郵件主機的電子郵件示意圖

收信的動作 (2/2)

- 遠端用戶使用的電腦直接連接到MTA。
- MTA 透過 MDA 檢查信件。
- 同時，根據 MUA 的不同設定， MTA 會選擇將該 mailbox 清除掉，或者繼續保留。

垃圾郵件的來源

- 在網際網路開始時就有垃圾郵件。
- 垃圾郵件也被稱作是“未經收信人許可的商業郵件” 或“未經收信人許可的大量郵件” 。

電子郵件的安全 (1/5)

- 你知道你的電子郵件只比你從牙買加郵寄的風景明信片稍微安全一點嗎？
- 即使確信並非人人能輕易攔截並且讀你的電子郵件時，這個危險仍然存在的
- 你或許透過網際網路傳送許多祕密和合法敏感檔案，其中的危險會立即使你感到害怕

電子郵件的安全 (2/5)

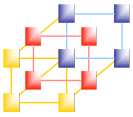
- 電子郵件的弱點
  - Sniffer
  - Security Key

電子郵件的安全 (3/5)

- Sniffer
  - 電子郵件最常被侵害的是電子竊聽（electronic eavesdropping），或者是被稱為網路監聽（sniffing）
  - 不要認為你的密碼非常的長而且有非常複雜的保護，那和電子郵件的傳送和儲存模式是有關的

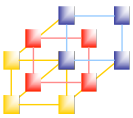
電子郵件的安全 (4/5)

- Security Key
  - 防止電子郵件被未授權的讀取最常用的方法是使用軟體加密
  - 任何沒有解碼器（decoder）或者鑰匙（key）的人無法讀取它



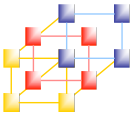
電子郵件的安全 (5/5)

- 有兩個主要的商業加密標準：PGP 和S/MIME
- PGP是最廣泛接受的工具
- 讀取PGP 加密的訊息，需要兩把鑰匙
  - 私鑰
  - 公鑰



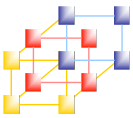
Outline

- Pretty good privacy (PGP)
- S/MIME
- DomainKeys Identified Mail (DKIM)



Pretty good privacy (PGP)

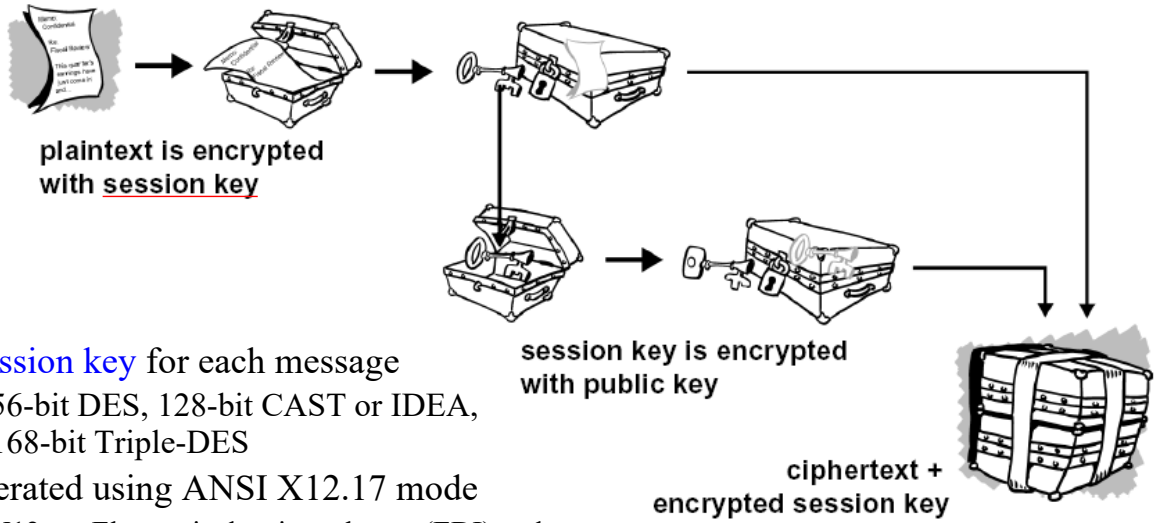
- PGP has grown explosively and is now widely used
- Developed by Phil Zimmermann
- Selected best available crypto algorithms to use
- Integrated into a single program
- Available on Windows, Unix, Macintosh …
- Originally free, now have commercial versions available also



PGP operations

- Authentication
- Confidentiality
- Compression
- E-mail compatibility
- Segmentation

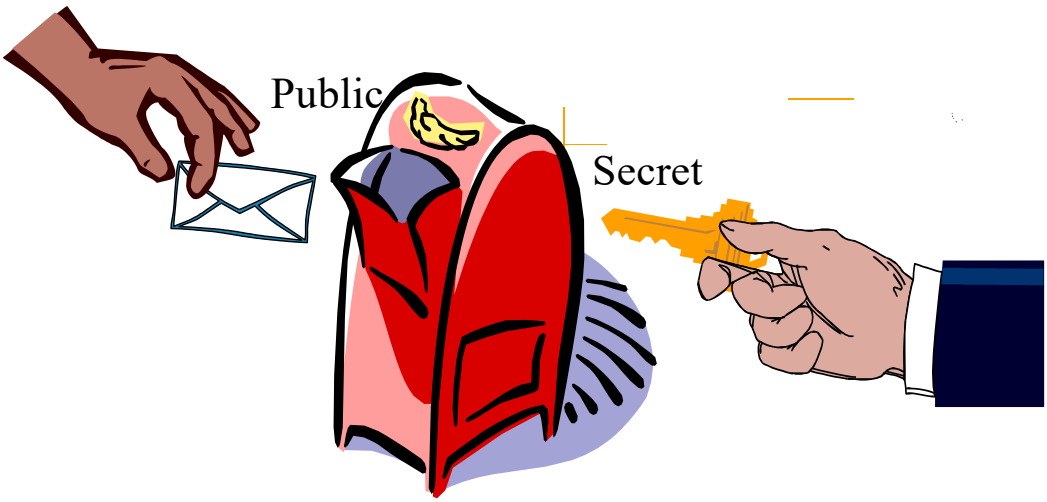
PGP session keys



- A session key for each message
  - 56-bit DES, 128-bit CAST or IDEA, 168-bit Triple-DES
- Generated using ANSI X12.17 mode
  - X12: an Electronic data interchange (EDI) and Context Inspired Component Architecture (CICA) standards along with XML schemas which drive business processes globally
- Uses random inputs taken from previous uses and from keystroke timing of user

Conventional cryptosystem

Public Key cryptosystem

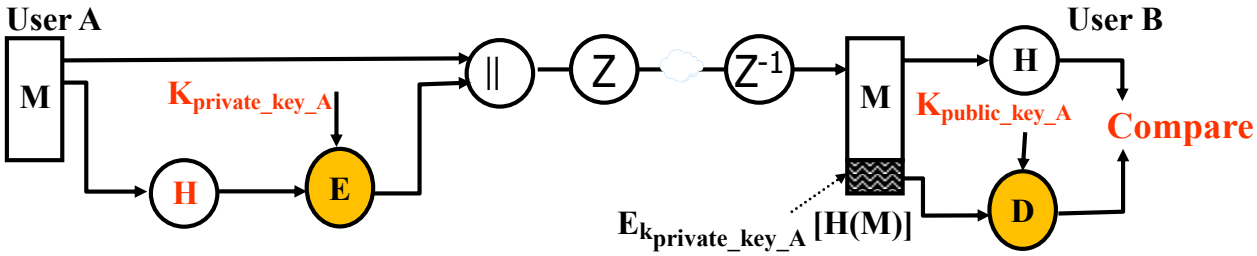


Conventional and public-key

- Advantages of public key cryptosystem
  - Increase security and convenience
  - Provide digital signature
- Disadvantage of public key cryptosystem
  - Speed
  - Vulnerable impersonation
- Public key cryptography is not mean to replace conventional cryptography, but rather to supplement it, to make it more secure
  - Digital envelope
    - `Public key system + conventional system`

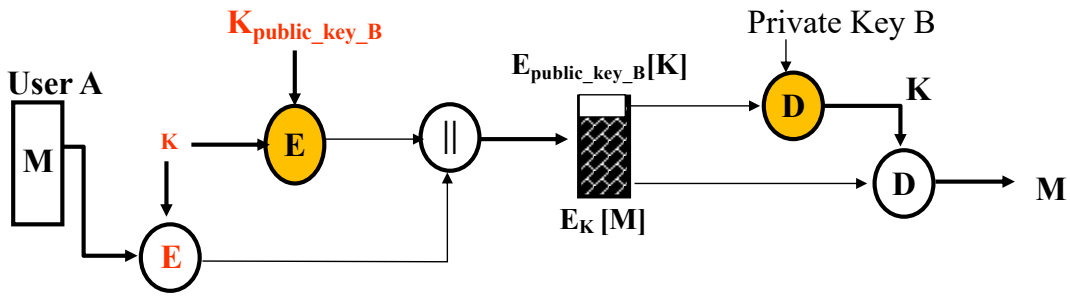
PGP operation -- authentication

The assurance that the communicating entity is the one that it claims to be.

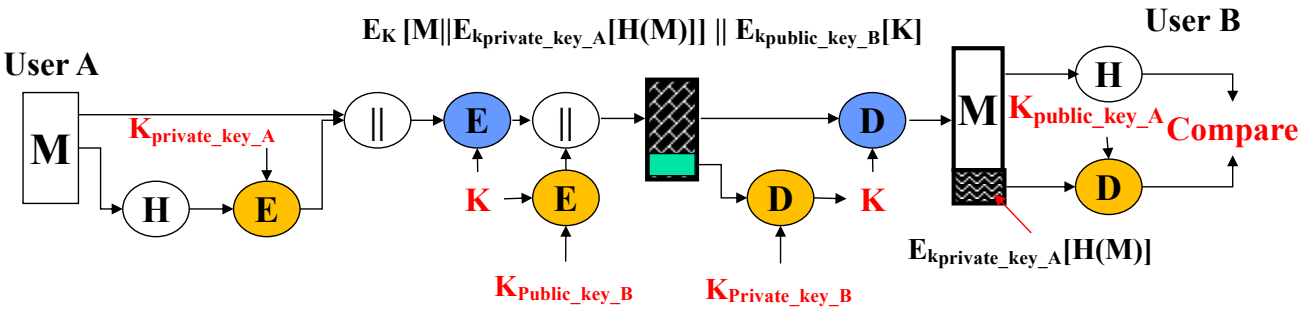


PGP operation -- confidentiality

The protection of data from unauthorized disclosure.



PGP operation – confidentiality and authentication



先產生數位簽章，再進行訊息加密。

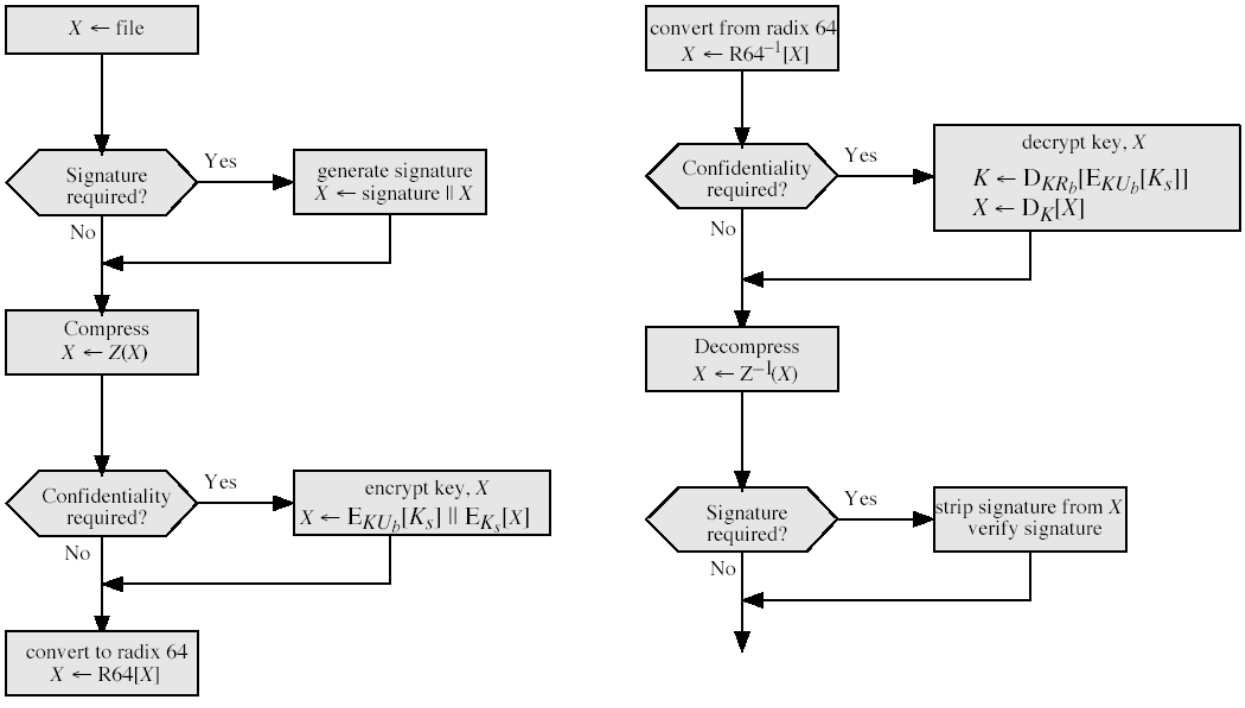
PGP operation – compression

- PGP compresses message after signing but before encrypting
  - Compress -> encryption v.s. encryption -> compression?
- Uses ZIP compression algorithm

PGP operation – e-mail compatibility

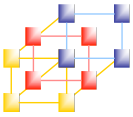
- When using PGP will have binary data to send
  - some email system was designed only for text
    - encode raw binary data into printable ASCII characters
    - Uses radix-64 algorithm
- PGP also segments messages if too big

PGP operation – summary



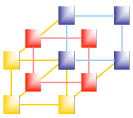
(a) Generic Transmission Diagram (from A)

(b) Generic Reception Diagram (to B)



Key identifier

- A unique identifying number associated with each key.
  - This identification number is useful for distinguishing between two keys that share the same user name and email address.
- The key ID associated with each public key consists of it least significant 64 bits
  - The Key ID of public key KUa is (KUa mod 2^{64})
  - The probability of duplicate key ID is very small
- A key ID is also required for the PGP digital signature



PGP key rings

- Each PGP user has a pair of keyrings
  - Public-key ring contains all the public-keys of other PGP users known to this user, indexed by key ID
  - Private-key ring contains the public/private key pair(s) for this user, indexed by key ID & encrypted keyed from a hashed passphrase

General structure of private/public key ring

Private Key Ring

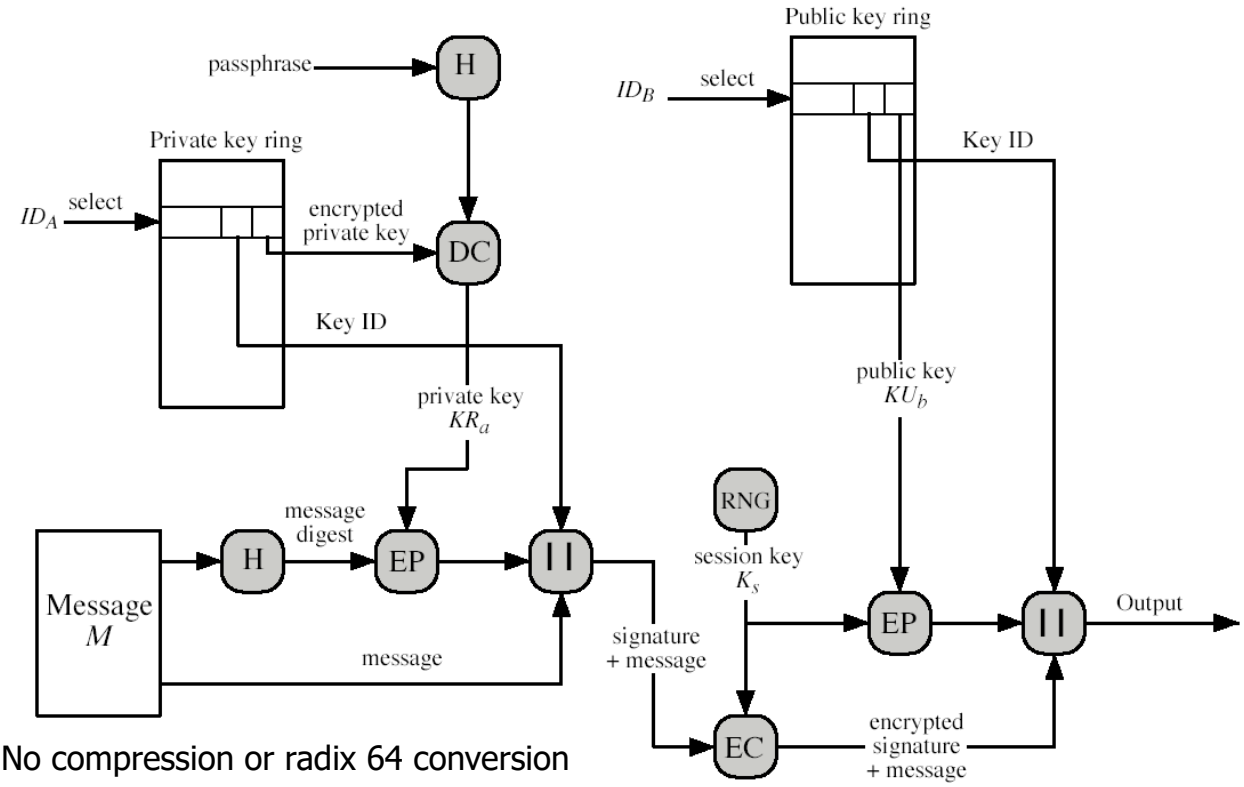
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
• • •	• • •	• • •	• • •	• • •
T_i	KU_i mod 2^{64}	KU_i	$E_{H(Pi)}[KR_i]$	User i
• • •	• • •	• • •	• • •	• • •

Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
T_i	KU_i mod 2^{64}	Ku_i	trust_flag$_i$	User i	trust_flag$_i$		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

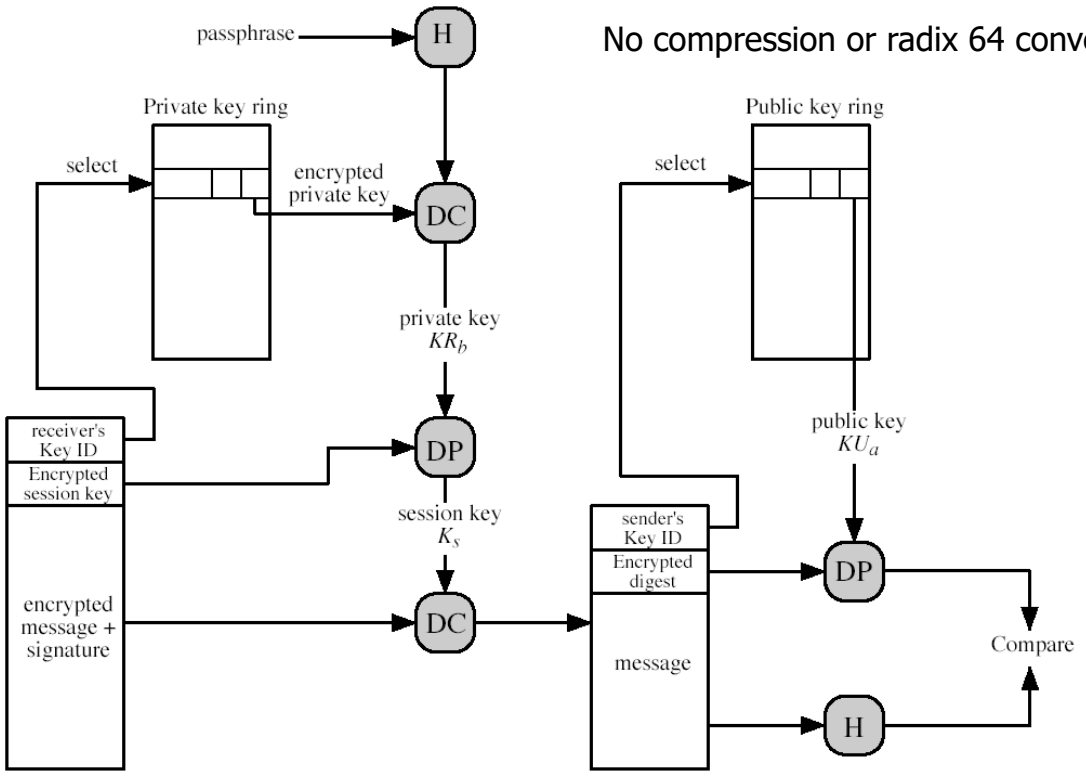
* = field used to index table

PGP message generation



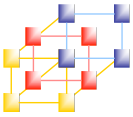
No compression or radix 64 conversion

PGP reception



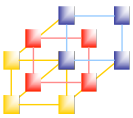
PGP Key Management

- Rather than relying on certificate authorities, in PGP every user is own CA
  - Can sign keys for users they know directly
- Forms a "web of trust"
  - Trust keys have signed
  - Can trust keys others have signed if have a chain of signatures to them
- Key ring includes trust indicators
- Users can also revoke their keys



Outline

- Pretty good privacy (PGP)
- S/MIME
- DomainKeys Identified Mail (DKIM)



Secure/Multipurpose Internet Mail Extension

- Security enhancement to MIME email
  - The original Internet email (RFC822) was text only
  - MIME provided support for varying content types and multi-part messages
  - With encoding of binary data to textual form
  - S/MIME added security enhancements
- S/MIME support in various modern mail agents
  - MS Outlook, Netscape, etc.

MIME 郵件標準 (1/3)

- RFC 822 封裝格式
  - 標頭 (Header)
  - 主體 (Body)

From: 志明 <bob@cc.cma.edu.tw>
To:春嬌 <alice@pchome.com.tw>
Subject: See you tomorrow
Date: Fri. 26 Dec 2003 10:12:37 – 0400

Please come to meet me at tomorrow.
<LF>&<CR>

MIME 郵件標準 (2/3)

- MIME 封裝格式
- MIME 標頭列
  - MIME-Version
  - Content-Type
  - Content-Transfer-Encoding
  - Content-ID
  - Content-Description
- MIME 內文型態
  - Text
  - Multipart
    - Multipart/Mixed
    - Multipart/Parallel
    - Multipart/Alternative
    - Multipart/Digest
  - Message
    - Message/rfc822
    - Message/partial
    - Message/external-body
    - Application/Octet-stream
  - Image
    - Image/Jpeg
    - Image/Gif
  - Audio
  - Video
  - Application
    - Application/Octet-stream
  - Application/PostScript
- MIME 內容轉換編碼
  - 7 bit
  - 8 bit
  - binary
  - quoted-printable
  - base64
  - x-token

MIME 郵件標準 (3/3)

- 範例：Multipart/mixed

```
From: Nathaniel Borenstein
To: Ned Freed
Date: Sun, 21 Mar 1993 23:56:48 -0800 (PST)
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 a handy place for composition agents to include an
    explanatory note to non-MIME conformant readers.
--simple boundary
    This is implicitly typed plain US-ASCII text.
   It does NOT end with a linebreak.
--simple boundary
Content-type: text/plain; charset=us-ascii
    This is explicitly typed plain US-ASCII text.
    It DOES end with a linebreak.
--simple boundary—
    This is the epilogue.  It is also to be ignored.
```


S/MIME 安全郵件 (1/4)

- S/MIME 安全郵件 (Secure/Multipurpose Internet Mail Extension)
  - 訊息摘要：MD5, SHA-1, SHA-256, SHA-512, …
  - 數位簽章：RSA / DSA 演算法
  - 訊息加密：RC2/40 , Triple DES, AES 密碼系統
  - 會議鑰匙加密： ElGamal 演算法
- 安全郵件型態 (1)
  - Multipart/Signed 型態
    - MIME 型態名稱：`Multipart/Signed`
    - 參數：`boundary, protocol, micalog`

```
Content-Type: multipart/signed; protocol="TYPE/STYPE";
        micalg="MICALG"; boundary="Signed Boundary“
--Signed Boundary
Content-Type: text/plain; charset="us-ascii“
   This is some text to be signed although it could be
   any type of data, labeled accordingly, of course.
--Signed Boundary
Content-Type: TYPE/STYPE
  CONTROL INFORMATION for protocol "TYPE/STYPE" would be here
--Signed Boundary--
```

S/MIME 安全郵件 (2/4)

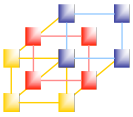
- 安全郵件型態 (2)
  - Application/pkcs-7-mime
    - 信件包裝成 CMS (Cryptographic Message Syntax)
    - 數位信封格式
    - PCKS #7 安全套件

```
EnvelopedData ::= SEQUENCE {
    version Version,
    recipientInfos RecipientInfos,
    encryptedContentInfo EncryptedContentInfo }
RecipientInfos ::= SET OF RecipientInfo
EncryptedContentInfo ::= SEQUENCE {
    contentType ContentType,
    contentEncryptionAlgorithm
    ContentEncryptionAlgorithmIdentifier,
    encryptedContent
      [0] IMPLICIT EncryptedContent OPTIONAL }
EncryptedContent ::= OCTET STRING
```


S/MIME 安全郵件 (3/4)

- 僅信封包裝格式
  - 包裝成『數位信封』
  - 可加密或明文封送

```
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
        name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
    rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6
    7n8HHGghyHhHUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
    f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
    0GhIGfHfQbnj756YT64V
```

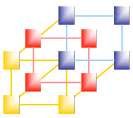


S/MIME 安全郵件 (4/4)

- 僅簽署郵件
  - 採用 Application 型態
  - 採用 Multipart 型態

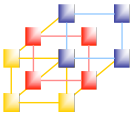
```
Content-Type: multipart/signed;
        protocol="application/pkcs7-signature"; micalg=sha1; boundary=boundary42
--boundary42Content-Type: text/plain
     This is a clear-signed message.
--boundary42
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s
     ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
     4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbn
     n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
     7GhIGfHfYT64VQbnj756
-boundary42--
```

  - 簽署並加密郵件
  - 利用 signed-only 與 encrypted-only 交替處理
  - 一般皆先簽署再加密



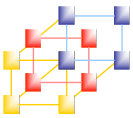
S/MIME functions

- Enveloped data
  - Encrypted content and associated keys
- Signed data
  - Encoded message + signed digest
- Clear-signed data
  - Cleartext message + encoded signed digest
- Signed & enveloped data
  - Nesting of signed & encrypted entities



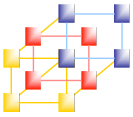
Cryptographic algorithm used in S/MIME

- Create a message digest
  - SHA-1 (MUST), MD5 (SHOULD)
- Encrypt message digest to form digital signature
  - DSS (MUST), RSA (SHOULD; key size 512 ~ 1024)
- Encrypt session key
  - ElGamal (MUST; a variant of Diffie Hellman)
  - RSA (SHOULD)
- Encrypt message
  - 3DES (MUST; recommended), 40-bits RC2 (MUST), AES



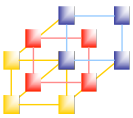
S/MIME messages

Type	Subtype	smime Parameter	Description
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the signature.
Application	pkcs7-mime	signedData	A signed S/MIME entity.
	pkcs7-mime	envelopedData	An encrypted S/MIME entity.
	pkcs7-mime	degenerate signedData	An entity containing only public-key certificates.
	pkcs7-signature	—	The content type of the signature subpart of a multipart/signed message.
	pkcs10-mime	—	A certificate registration request message.



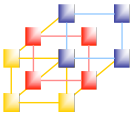
Registration request

- An application or user will apply to a certification authority for a public-key certificate
- The application/pkcs10 S/MIME entity is used to transfer a certificate request
- The certificate request
  - certificationRequestInfo block
    - A name of the certificate subject
    - A bit-stream representation of the user's public key
  - An identifier of the public-key encryption algorithm
  - The signature of the certificationRequestInfo block



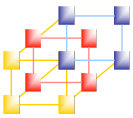
Certificates only message

- Contain only certificate revocation list (CRL)
- The message is an application/pkcs7-mime type/subtype with an smime-type parameter of degenerate



S/MIME certificate processing

- S/MIME uses X.509 v3 certificates
- Managed using a hybrid of a strict X.509 CA hierarchy & PGP's web of trust
- Each client has a list of trusted CA's certs and own public/private key pairs & certificates
- Certificates must be signed by trusted CA's
- Key management functions on S/MIME user
  - Key generation
  - Registration
  - Certificate storage and retrieval



Certificate authorities

- Have several well-known CA's
- Verisign one of most widely used
  - Verisign issues several types of Digital IDs
  - with increasing levels of checks & hence trust

Class	Identity Checks	Usage
1	name/email check	web browsing/email
2+	enroll/addr check	email, subs, s/w validate
3+	ID documents	e-banking/service access

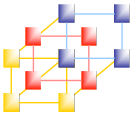
VeriSign public-key certificate classes

	Summary of Confirmation of Identity	IA Private Key Protection	Certificate Applicant and Subscriber Private Key Protection	Applications implemented or contemplated by Users
Class 1	Automated unambiguous name and E-mail address search	PCA: trustworthy hardware; CA: trust-worthy software or trustworthy hardware	Encryption software (PIN protected) recommended but not required	Web-browsing and certain e-mail usage
Class 2	Same as Class 1, plus automated enrollment information check plus automated address check	PCA and CA: trustworthy hardware	Encryption software (PIN protected) required	Individual and intra- and inter-company E-mail, online subscriptions, password replacement, and software validation
Class 3	Same as Class 1, plus personal presence and ID documents plus Class 2 automated ID check for individuals; business records (or filings) for organizations	PCA and CA: trustworthy hardware	Encryption software (PIN protected) required; hardware token recommended but not required	E-banking, corp. database access, personal banking, membership-based online services, content integrity services, e-commerce server, software validation; authentication of LRAAs; and strong encryption for certain servers

IA Issuing Authority
CA Certification Authority
PCA VeriSign public primary certification authority
PIN Personal Identification Number
LRAA Local Registration Authority Administrator

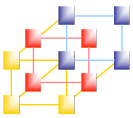
S/MIME V3和OpenPGP的比較 (1/2)

- OpenPGP 規範
  - 『非常好的隱密』(Pretty Good Privacy, PGP) 由 Phil Zimmermann 教授獨立發展
  - RFC 1991, PGP Message Exchange Formats
  - RFC 2015, MIME Security with Pretty Good Privacy
  - RFC 2440, OpenPGP Message Format
  - RFC 3156, MIME Security with Pretty Good Privacy
- S/MIME 規範
  - Secure/MIME 由 RSA Data Security Inc. 發行
  - RFC 2311, S/MIME Version 2 Message Specification
  - RFC 2312, S/MIME Version 2 Certification Handling
  - RFC 2313, PKCS #1: RSA Encryption Version 1.5
  - RFC 2314, PKCS #10: Certification Request Syntax Version 1.5
  - RFC 2315, PKCS #7: Cryptographic Message Syntax Version 1.5
  - RFC 2268, Description of the RC2 Encryption Algorithm



S/MIME V3和OpenPGP的比較 (2/2)

制定規範	S/MIME v3	OpenPGP
訊息格式		Binary, based on PGP
憑證格式	Binary, based on X.509v3	Binary, based on PGP
秘密鑰匙系統	Triple DES	Triple DES
簽章演算法		ELGamal DSS
雜湊演算法		SHA-1
MIME 簽署封裝	multipart/signed 或 CMS	multipart/signed
MIME 加密封裝	application/pkcs7-mime	multipart/encrypted



Outline

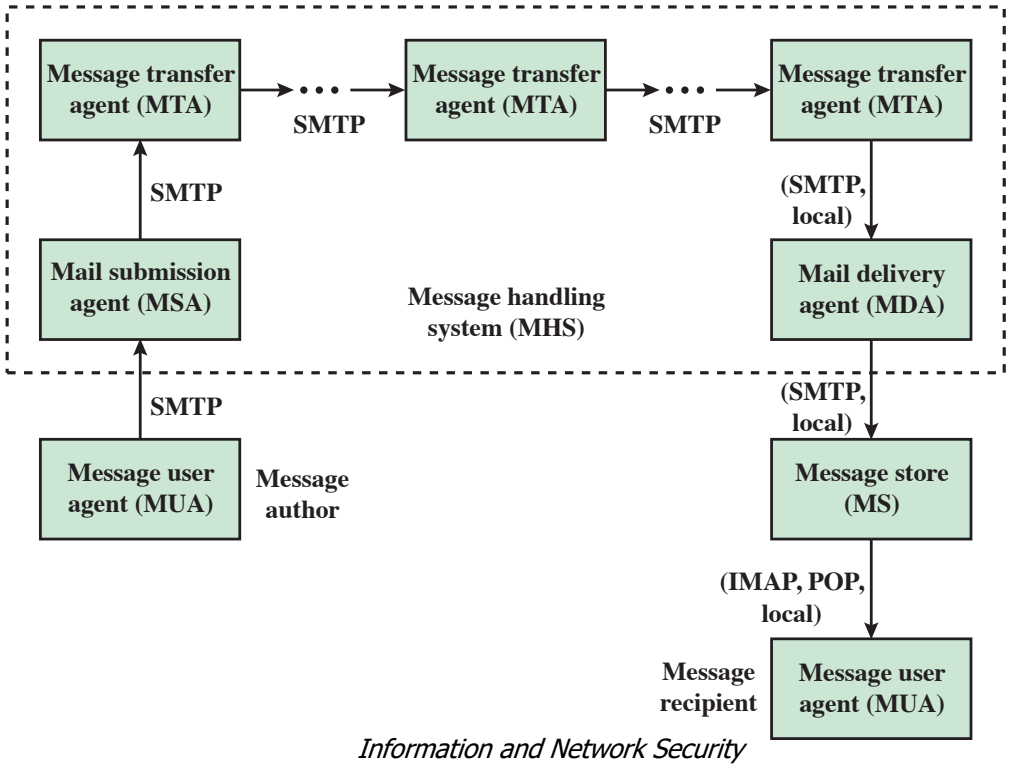
- Pretty good privacy (PGP)
- S/MIME
- DomainKeys Identified Mail (DKIM)

DomainKeys Identified Mail (DKIM)

- A specification for cryptographically signing e-mail messages, permitting a signing domain to claim responsibility for a message in the mail stream
- Message recipients can verify the signature by querying the signer's domain directly to retrieve the appropriate public key and can thereby confirm that the message was attested to by a party in possession of the private key for the signing domain
- Proposed Internet Standard RFC 4871
- Has been widely adopted by a range of e-mail providers and Internet Service Providers (ISPs)

DKIM 會在所有外寄郵件的標頭加上加密簽名，收到簽名郵件的電子郵件伺服器則會使用 DKIM 來解密郵件標頭，驗證郵件寄出後並未遭人竄改。

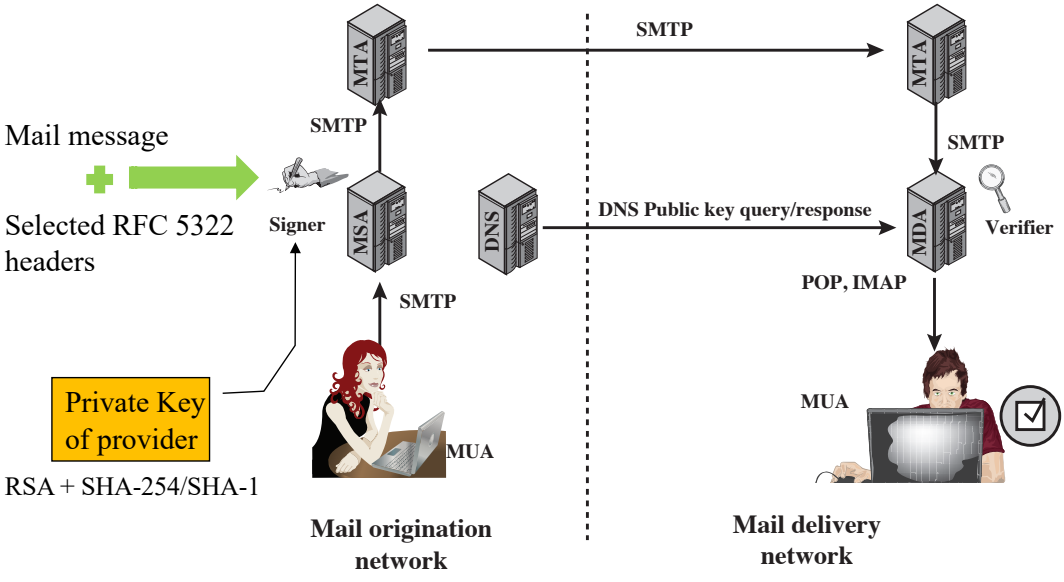
Internet Mail Architecture



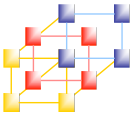
E-mail Threats

- RFC 4684 (Analysis of Threats Motivating DomainKeys Identified Mail) describes the threats being addressed by DKIM in terms of the characteristics, capabilities, and location of potential attackers
  - At the low end are attackers who simply want to send e-mail that a recipient does not want to receive (廣告、垃圾信)
  - The next level are professional senders of bulk spam mail and often operate as commercial enterprises and send messages on behalf of third parties (非法廣告提供者)
  - The most sophisticated and financially motivated senders of messages are those who stand to receive substantial financial benefit, such as from an e-mail based fraud scheme (詐騙)

DKIM Deployment



DNS = domain name system
MDA = mail delivery agent
MSA = mail submission agent
MTA = message transfer agent



DKIM Functional Flow

ADMA: Administrative Management Domain
SDID: Signing Domain IDentifier