Store and Forward Processing

UA : User Agent
MTA : Message Transfer Agent
Real World

- Private e-mail to friends
- Private e-mail to business associates
- Private and authenticated e-mail to business partners
- Electronic Commerce
- etc.
Security Req’t of E-mail

- Privacy
- Authentication
- Integrity
- Non-repudiation: third-party authentication
- Proof-of-submission: certified mail
- Proof-of-delivery
- Message flow confidentiality: C can’t know the fact A and B communicates
- Anonymity: Not revealing sender’s ID information
- Containment: security labeling
- Audit: logging specific day’s mailing facts
- Accounting: extract statistics
- Self destruct: self destruct after receiving
- Message sequence integrity: sequential delivery of messages
Implementation Example

Sender

Receiver

Plain text

Session Key

Receiver’s Public key

Asym.

H(n)

Sym.

Digital Sig.

Session Key

Asym.

Sender’s Private key

Plain text

Digital Sig.

Yes / No

Receiver’s Private key

Session Key

Sender’s Public key

H(m)
Non-repudiation

- **(Definition in OSI)**
  - security service that counters repudiation where repudiation is defined as “denial by one of the entities involved in a communication of having participated in all or part of the communication”
  - anti-repudiation is better choice

- **(Definition in ABA)**
  - Strong and substantial evidence of the identity of the signer of a message and of message integrity, sufficient to prevent a party from successfully denying the origin, submission of delivery of the message and the integrity of its contents.
Non-repudiation

- **Non-repudiation of Origin (NRO)**
  - prevents or resolves disagreements as to whether a particular party originated a particular item.

- **Non-repudiation of Receipt (NRR)**
  - prevents or resolves disagreements whether a particular party received a particular data item, the time the delivery occurred.
Implementing Non-repudiation

- **Direct Method**
  - Secret exchange protocol
  - Oblivious Transfer protocol
  - Fairness Problem

- **Indirect Method**
  - TTP (Ex: Post Office)
  - DA (Delivery Agent)

- **TimeStamping**
How NRO happens

- A recipient claims to have received
  - a message, but the party identified as sender claims not to have sent any message.
  - a message different from that which the sender claims to have sent.
  - a particular message originated on a specific date and time, but the party identified as sender claims not to have sent that particular message at that specific time and date.
Measures against NRO

- Adequately associate, or link together, various pieces of information including at least
  - The identity of the originator and
  - The content of the message,
  optionally
  - The date and time at which origination occurred.
  - The identity of the intended recipients and
  - The identity of any TTP involved in generating evidence
Way of NRO

(1) Originator’s Digital Signature

Originator

Sign

Message
Signature
Certificate

TTP

Verify

Store

Recipient

(c)ICU Kwangjo KIm
Way of NRO(II)

(2) Digital Signature of TTP

TTP

Originator

Sign

Message

Signature

Verify

Store

Recipient

(c)ICU Kwangjo KI'm
Why NRR happens

- A sender claims to have sent
  - a message, but the party identified as recipient claims not to have sent any message.
  - a message different from that which the recipient claims to have received.
  - a particular message originated on a specific date and time, but the party identified as recipient claims not to have received that particular message at a time and on a date consistent with the claimed time and date of sending.
Measure against NRR

- Adequately associate, or link together, various pieces of information including at least
  - The identity of the recipient and
  - The content of the message,

  optionally
  - The date and time at which delivery of the message occurred.
  - The identity of the originator and
  - The identity of any TTP involved in generating evidence
Way of NRR

(1) Recipient’s Signature

Originator

Verify

Store

TTP

Recipient

Sign

Message

Signature

Certificate

(c)ICU Kwangjo KIm
History of e-mail

- **Early 1980**: Secure/32, Charli Merritt, using PKC
- **1986**: Mail Safe, RSADSI, DOS
- **1990**: PEM (Privacy Enhanced Mail)
  - RIPEM (Riordan’s Internet PEM)
  - TIS/PEM
- PGP (Pretty Good Privacy)
- S/MIME: Multimedia e-mail
Document of PEM

- RFC 1421, Part I: Message Encryption and Authentication Procedure
- RFC 1422, Part II: Certificate-based Key Management
- RFC 1423, Part III: Algorithms, Modes, and Identifiers
- RFC 1424, Part IV: Key Certification and Related Services
Design Environments of PEM

- Work with existing e-mail system in Internet
- Not restricted to particular host or OS
- Compatible with normal, non secure e-mail
- Performed on PC as well as on large system
- Compatible with a variety of key-management approach including manual distribution, centralized key distribution
Security Services of PEM

- Confidentiality
- Data origin authentication
- Message Integrity
- Non-repudiation of origin
- Key Management
Cryptographic Algorithm

- Data Encryption: DES in CBC
- Key Management: DES in ECB, CBC and RSA
- MIC: RSA+MD2, RSA+MD5
- Digital Signature: RSA+MD2, RSA+MD5
Style of message

- Ordinary, unsecured data
- MIC-Clear: integrity and authentication, but no confidentiality (integrity-protected unmodified data)
- MIC-Only: MIC-Clear + encoding (integrity-protected encoded data)
- ENCRYPTED: MIC-Only + confidentiality (encoded encrypted integrity-protected data)
# PEM Message

<table>
<thead>
<tr>
<th>BEGIN-PRIVACY-ENHANCED-MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing Type</td>
</tr>
<tr>
<td>Content Domain</td>
</tr>
<tr>
<td>Message text encryption algorithm</td>
</tr>
<tr>
<td>Issuing authority</td>
</tr>
<tr>
<td>Version/expiration</td>
</tr>
<tr>
<td>Origination certificate</td>
</tr>
<tr>
<td>Originator key information</td>
</tr>
<tr>
<td>Issuer certificate</td>
</tr>
<tr>
<td>MIC information</td>
</tr>
<tr>
<td>Issuing authority</td>
</tr>
<tr>
<td>Version/expiration</td>
</tr>
<tr>
<td>Encrypted DEK</td>
</tr>
<tr>
<td>User Text</td>
</tr>
</tbody>
</table>

| END-PRIVACY-ENHANCED-MESSAGE |

(c)ICU Kwangjo Klm
Processing steps of PEM Message

- **Sending**
  - Canonicalization
  - Message Integrity and originator authentication
  - Encryption (optional)
  - Transmission encoding (optional)

- **Receiving**
  - Decoding (optional)
  - Decrypting (optional)
  - Verifying message integrity and authenticity
  - Translation
Certification Hierarchy

HA : High Assurance, DA: Discretionary Assurance, NA: No Assurance

(c) ICU Kwangjo Klm
PGP

- Program for confidentiality and authentication service
- Select best available algorithm
  - Integrate algorithms into general-purpose
  - Made the package and its document, including source code, freely available via Internet
  - Low-cost commercial version by Viacrypt and Public-domain version
Background of PGP

- Available in various platforms
- Use algorithm survived extensively public review like RSA, DSS, DH, CAST-128, IDEA and 3DES, SHA-1
- Wide range of applicability from cooperation to individual
- Not developed by, nor controlled by, any government and standards organization
History of PGP(I)

- Designed by Phil Zimmerman
  - High security
  - public domain S/W
  - popular for personal use

- PGP Classic: Can’t handle Internet Mail
  - PGP v.1.0: ‘91.6
  - PGP v.2.0: ‘92.9
  - PGP v.2.3a: ‘93.7 (last version of PGP didn’t use RSAREF)
  - PGP v.2.4: original ViaCrypt PGP
  - PGP v.2.5: Interim release of PGP with RSAREF
  - PGP v.2.6: Freeware version of PGP
  - PGP v.2.7: Commercial version by ViaCrypt
History of PGP(II)

- 4 versions
  - PGP Classic: non-commercial use
  - PGP 5.0: Improve security but don’t adapt RSA
  - PGP/MIME
    - MIME-based
    - Use special certificate
    - Handle Internet Mail
  - OpenPGP
    - Use DH, DSA, SHA-1
  - Interoperability with S/MIME
### Features of PGP

<table>
<thead>
<tr>
<th>Function</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Signature</td>
<td>DSS/SHA or RSA/SHA</td>
</tr>
<tr>
<td>Message Encryption</td>
<td>CAST-128 or IDEA or 3DES (64bCFB) w/ DH or RSA</td>
</tr>
<tr>
<td>Compression</td>
<td>ZIP</td>
</tr>
<tr>
<td>E-mail compatibility</td>
<td>Radix 64</td>
</tr>
<tr>
<td>Segmentation</td>
<td></td>
</tr>
</tbody>
</table>

(Note) Signing before compression
Encryption after compression
Notation

- $K_s$: session key for conventional algorithm
- $KR_a$: Private key of user A for PKC
- $KU_a$: Public key of user A for PKC
- $EP$: PK encryption
- $DP$: PK decryption
- $EC$: conventional encryption
- $DC$: conventional decryption
- $H$: Hash function
- $||$: concatenation
- $Z$: compression
- $R64$: conversion to radix 64 ASCII format
Security Service in PGP(I)

(a) Authentication only

(b) Confidentiality only

(c) ICU Kwangjo Klm
Security Service in PGP(II)

Source A  Destination B

(c) Confidentiality and Authentication

(c) ICU Kwangjo Klm
Steps of sending a message

1. \( X \leftarrow \text{file} \)
2. \( \text{Signature Req'd?} \)
   - \( \text{Y} \rightarrow \text{Generate signature} \)
     \( X \leftarrow \text{signature} \parallel X \)
   - \( \text{N} \rightarrow \text{Compress} \)
     \( X \leftarrow Z(X) \)
3. \( \text{Confidentiality Req'd} \)
   - \( \text{Y} \rightarrow \text{Encrypt} \)
     \( \{\text{key, } X\} \leftarrow E_{Kub}[K_s] \parallel E_{ks}[X] \)
   - \( \text{N} \rightarrow \text{Convert to radix 64} \)
     \( X \leftarrow R64[X] \)
Steps of receiving a message

Convert from radix 64
\( X \leftarrow R_{64}^{-1}[X] \)

Confidentiality Req’d

Decompress \( X \leftarrow Z^{-1}(X) \)

Signature Req’d?

Yes

Decrypt \( \{\text{key}, X\} \)
\( K_s \leftarrow D_{KRB}[K_s]; X \leftarrow D_{ks}[X] \)

Strip signature from \( X \)
Verify signature

No

(c)ICU Kwangjo KIm
**Message format (A->B)**

- **Session Key Component**
  - Session Key (Ks)
  - Timestamp
  - Key ID of recipient’s public key (KUb)
  - Key ID of sender’s public key (KUa)
  - Leading 2 octets of message digest
  - Message digest
  - File Name
  - Timestamp
  - Data

- **Signature**
  - Session (Ks)
  - Timestamp
  - Key ID of sender’s public key (KUa)
  - Leading 2 octets of message digest
  - Message digest
  - File Name
  - Timestamp
  - Data

- **Encryption**
  - EKUb: encryption with user b’s public key
  - EKRa: encryption with user b’s private key
  - EKs: encryption with session key

- **Compressions**
  - ZIP: ZIP compression functions
  - R64: Radix-94 conversion function

---

(c)ICU Kwangjo KIm
Key Management

- One-time session key: 128bit for CAST or IDEA, 168 bit for 3DES
- Public Key
- Private Key
- Passphrase-based conventional key
### Key Rings of PGP

**Private Key ring:** store his own public and private keys

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>KeyID</th>
<th>Public Key</th>
<th>Encrypted Private key</th>
<th>UserID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ti</td>
<td>KU&lt;sub&gt;i&lt;/sub&gt; mod 2&lt;sup&gt;64&lt;/sup&gt;</td>
<td>KUi</td>
<td>E&lt;sub&gt;H(pi)&lt;/sub&gt;[KRi]</td>
<td>User i</td>
</tr>
</tbody>
</table>

**Public Key ring:** store all known entities’ public key

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>KeyID</th>
<th>Public Key</th>
<th>Owner Trust</th>
<th>User ID</th>
<th>Key Legitimacy</th>
<th>Signature(s)</th>
<th>Signature Trust(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ti</td>
<td>KU&lt;sub&gt;i&lt;/sub&gt; mod 2&lt;sup&gt;64&lt;/sup&gt;</td>
<td>KUi</td>
<td>trust_flagi</td>
<td>User i</td>
<td>trust_flagi</td>
<td>ERj(H([KUi]))</td>
<td>ERk(H([KUi]))</td>
</tr>
</tbody>
</table>
Use of private key

- Using IDEA, store encrypted key
  - User selects passphrase
  - When generating private/public key pairs, use passphrase
  - Passphrase is inputted to hash ft. MD5 (SHA-1), Use 128 (160)-bit hash value as key of IDEA

- After use, delete it from system
Message sending (Detailed)

Private key ring

Passphrase

Private key, KRa

Select

Message M

Message digest

Encrypted Private key

DC

H

Key ID

Public key ring

IDb

Select

Encrypted signature + message

Public key

KUb

RNG

Session key, KS

Key ID

Encrypted signature + message

Signature + message

(c) ICU Kwangjo KIm
Message receiving (Detailed)

Select

Passphrase → H

Private key ring

Public Key VIN

Encrypted Private key

DC

Receiver's Key ID

Encrypted Session Key

Encrypted message + signature

Select

Sender's Key ID

Encrypted Digest

Message

Compare

Public key ring

Select

Public Key VIN

Encrypted Private key

DC

Session Key KS

(c)ICU Kwangjo Klm
Distribution of Public Key

- Direct delivery (floppy disk, mail,..)
- Sending e-mail and confirm by telephone
  - B → Key (by e-mail) → A → (128-bit MD5 digest) → fingerprint (by telephone) → B (matching)

- TTP
- CA
PGP’s Trust Model

- 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
- Unknown signatory

You

(c)ICU Kwangjo KIm
Revocation of Public key

- Issue public key revocation signature
  - Similar form of usual Signature Certificate
  - Signature using secret key of public key to be revoked
  - Propagate as many as possible

- All public keys signed by revoked key
  - Make Owner_trust and key_legitimacy to untrust