

E-mail Security policy

Security Services for E-mail

- ▶ privacy
 - ▶ authentication
 - ▶ integrity
 - ▶ non-repudiation
 - ▶ anonymity
 - ▶ proof of submission
 - ▶ proof of delivery
 - ▶ message flow confidentiality, etc.
- 

Key Management

- ▶ A per-message symmetric key is used for message encryption,
- ▶ which is conveyed in the mail, encrypted under a long-term key (typically a public key)
- ▶ Long-term keys can be established,
 - offline
 - online, with help from a trusted third party
 - online, through a webpage (for public keys)

Multiple Recipients

- ▶ Message key will be encrypted under each recipients long term key in the message header.
 - Bob's ID, $K_{\text{Bob}}\{S\}$
 - Carol's ID, $K_{\text{Carol}}\{S\}$
 - Ted's ID, $K_{\text{Ted}}\{S\}$
 - $S\{m\}$

- ▶ E.g.:

To: Bob, Carol, Ted

From: Alice

Key-info: Bob-4276724736874376

Key-info: Carol-78657438676783457

Key-info: Ted-12873486743009

Msg-info: UHGuiy77t65fhj87oi.....

Text Format Issues

- ▶ Mail gateways / forwarders may modify the format of the message (wrapping long lines, end-of-line character, high order bits, etc.), causing the integrity check to fail
- ▶ Encode messages in a format supported by all mailers. 6-bit representation, no long lines, etc. (similar to uuencode)

Text Format Issues (cont'd)

- ▶ Problem: Non-supportive clients should be able to read authenticated (but not encrypted) messages, which they no longer can.
- ▶ Two options:
 - MAC without encoding
(subject to corruption by mail routers)
 - Encode & MAC/encrypt
(may not be readable at the other end)

Providing Different Services

- ▶ confidentiality: by encryption
 - ▶ auth./integrity: by signature or MAC
 - ▶ non-repudiation: by signature
 - ▶ some eccentric services,
 - anonymity
 - message flow confidentiality
 - non-repudiation with secret keys
- can be provided by TTP support.

PEM & S/MIME

- ▶ Privacy Enhanced Mail (PEM)
 - Developed by IETF, to add encryption, source authentication & integrity protection to e-mail
 - Allows both public & secret long-term keys
Message key is always symmetric
 - Specifies a detailed certification hierarchy
- ▶ Secure/MIME (S/MIME)
 - PEM never took off; CA hierarchy difficult to realize
 - S/MIME: PEM design incorporated into MIME

PEM Key Exchange & Encryption

- ▶ “Interchange keys”: Users’ long-term PEM keys
 - public (a detailed PKI is defined)
 - secret (pre-shared symmetric keys)
- ▶ Encryption
 - A symmetric per-message key is sent encrypted under the interchange key.
 - The message is encrypted under the per-message key (typically with DES in CBC mode)
- ▶ Authentication
 - Message is authenticated by a “MIC”
(Q: Any authentication for the per-message key?)

PEM Certificate Hierarchy

- ▶ The root CA: “Internet Policy Registration Authority” (IPRA)
- ▶ “Policy Certification Authorities”: Second-level, CA-certifying CAs, each with a different policy:
 - High Assurance (HA): super-secure
 - implemented on secure platforms
 - regulates that the child CAs (also HACAs) enforce the same rules
 - Discretionary Assurance (DA): secure
 - requires that the child CAs own their names
 - No Assurance (NA): no constraints
 - can be used to certify Internet personas (pseudonyms)
- ▶ Lower-level CAs, certifying individuals or other CAs

S/MIME vs. PEM

- ▶ Incorporated into MIME; no other encoding
 - ▶ Any sequence of sign & encrypt is supported (each as a recursive MIME encapsulation)
 - ▶ Has more options than PEM
 - ▶ ASN.1 header encoding
 - ▶ No prescribed certification hierarchy
 - ▶ Has a good prospect of deployment for commercial & organizational usage
- 

Pretty Good Privacy (PGP)

- ▶ Popular mail & file encryption tool
 - ▶ Developed by Phil Zimmermann, 1991
 - ▶ Based on RSA, IDEA, MD5 (later DSS, ElGamal (DH), 3DES, SHA1)
 - ▶ Many different versions have emerged (from PGP, from GNU (GPG), from IETF (Open PGP))
- 

Publishing and Notification Security Policy

- ▶ It is a commonly used pattern for inter-object communication.
- ▶ Notification: may have specifications that define a standard web services approach to notification using a topic based publish /subscribe pattern.