Network Security

Asim Rasheed

Where we are …

- Introduction to network security
- Vulnerabilities in IP
- I. CRYPTOGRAPHY
  - Symmetric Encryption and Message Confidentiality
  - Public-Key Cryptography and Message Authentication
- II. NETWORK SECURITY APPLICATIONS
  - Authentication Applications (Kerberos, X.509)
  - Electronic Mail Security (PGP, S/MIME)
  - IP Security (IPSec, AH, ESP, IKE)
  - Web Security (SSL, TLS, SET)
- III. SYSTEM SECURITY
  - Intruders and intrusion detection
  - Malicious Software (viruses)
  - Firewalls and trusted systems

E-mail Security: S/MIME

What is S/MIME?

- Secure / Multipurpose Internet Mail Extension
- A security enhancement to MIME
- Provides similar services to PGP
- Based on technology from RSA Security
- Industry standard for commercial and organizational use
- RFC 2630, 2632, 2633

RFC 822

- Defines a format for text messages to be sent using e-mail
- Structure of RFC 822 compliant messages
  - header lines (e.g., from: …, to: …, cc: …)
  - blank line
  - body (the text to be sent)
- Example
  - Date: Tue, 16 Jan 1998 10:37:17 (EST)
  - From: “Levente Buttyan” <buttyan@hit.bme.hu>
  - Subject: Test
  - To: afriend@otherhost.bme.hu

Problems with RFC 822 and SMTP

- Executable files must be converted into ASCII
  - various schemes exist (e.g., Unix UUencode)
  - a standard is needed
- Text data that includes special characters (e.g., Hungarian text)
- Some servers
  - reject messages over a certain size
  - delete, add, or reorder CR and LF characters
  - truncate or wrap lines longer than 76 characters
  - remove trailing white space (tabs and spaces)
  - pad lines in a message to the same length
  - convert tab characters into multiple spaces

MIME

- Defines new message header fields
- Defines a number of content formats (standardizing representation of multimedia contents)
- Defines transfer encodings that protects the content from alteration by the mail system

MIME - New header fields

- MIME-Version
- Content-Type
  - describes the data contained in the body
  - receiving agent can pick an appropriate method to represent the content
- Content-Transfer-Encoding
  - indicates the type of the transformation that has been used to represent the body of the message
- Content-ID
- Content-Description
  - description of the object in the body of the message
  - useful when content is not readable (e.g., audio data)

MIME – Content types and subtypes

- Text
  - Plain: unformatted text (ASCII)
  - Enriched: greater formatting flexibility
- Multipart type: contain multiple Independent parts
  - Mixed: Different parts transmitted together. Presented to the receiver in the order that they appear message.
  - Parallel: Differs from Mixed. Defined for delivering the parts to the receiver.
  - Alternate: Different parts are alternative versions of the same information.
  - Digest: Similar to Mixed, but the default type/subtype of each part is message/rfc822.

MIME – Content types and subtypes

- Message
  - Rfc822: The body is itself an encapsulated message that conforms to RFC 822.
  - Partial: Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
  - External-body: Contains a pointer to an object that exists elsewhere.

MIME – Content types and subtypes

- Image
  - Jpeg: The image is in JPEG format, JFIF encoding.
  - Gif: The image is in GIF format.
- Video: mpeg
- Audio:
  - Basic: Single-channel 8-bit ISDN µ-law encoding at a sample rate of 8 kHz.
- Application
  - Postscript: Adobe Postscript
  - Octet-stream: General binary data consisting of 8-bit bytes.

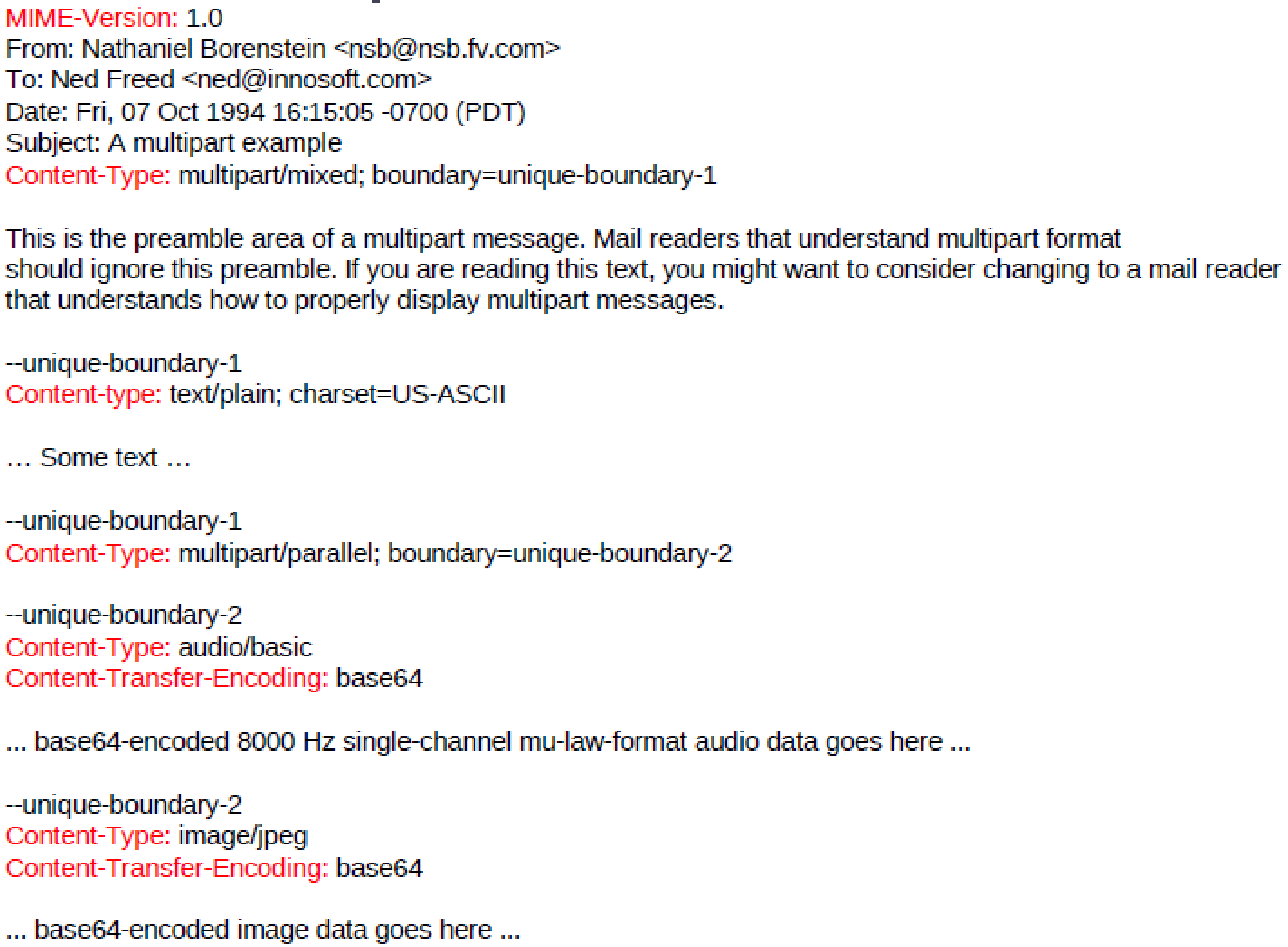
MIME – Transfer encodings

- 7bit
  - short lines of ASCII characters
- 8bit
  - short lines of non-ASCII characters
- Binary
  - non-ASCII characters
  - lines are not necessarily short
- Quoted-printable
  - non-ASCII characters are converted into hexa numbers (e.g., =EF)

MIME – Transfer encodings

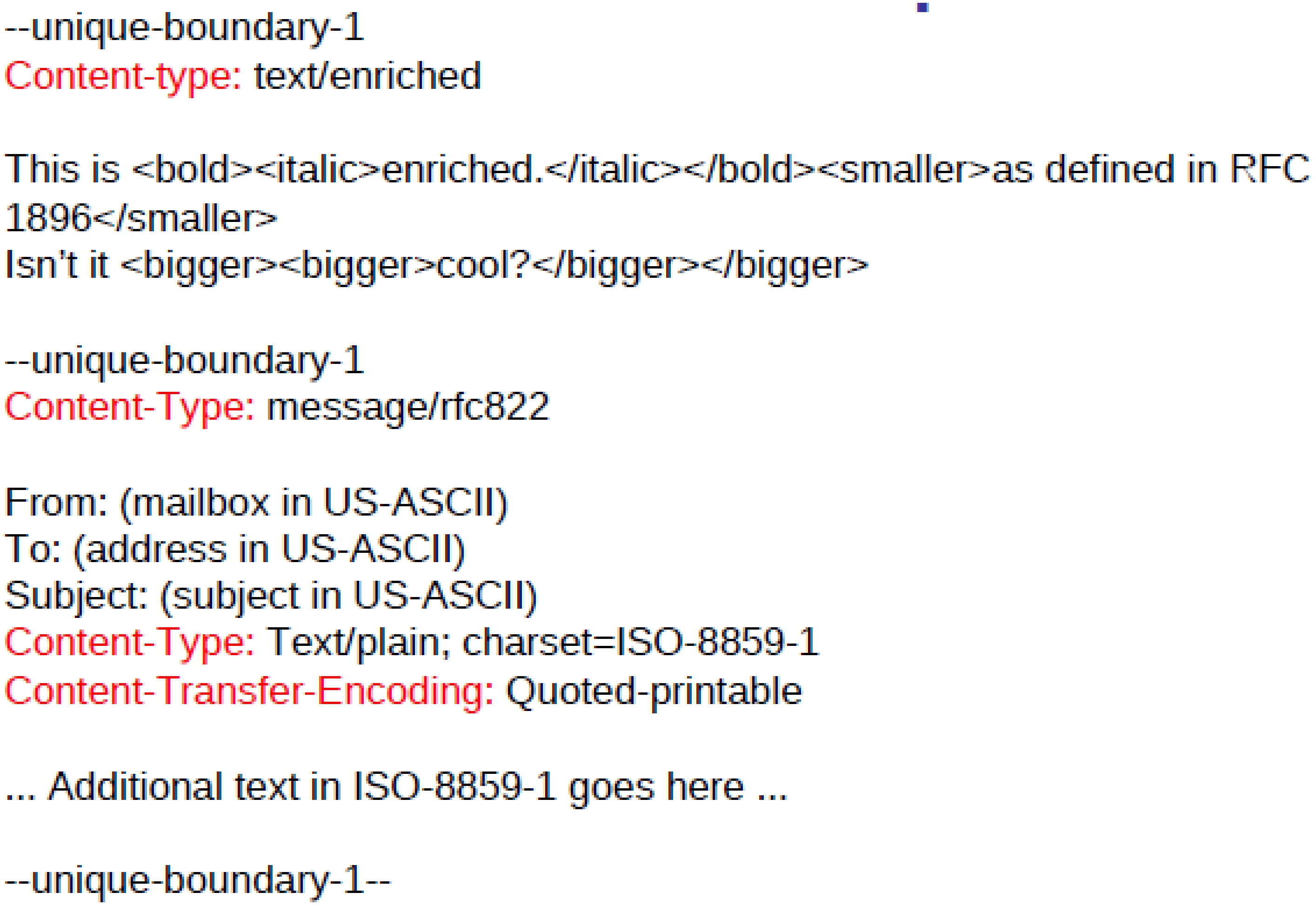
- Base64 (radix 64)
  - 3 8-bit blocks into 4 6-bit blocks
- x-token
  - non-standard encoding

MIME – Example



```
MIME-Version: 1.0
From: Nathaniel Borenstein <nsb@nsb.fv.com>
To: Ned Freed <ned@innosoft.com>
Date: Fri, 07 Oct 1994 16:15:05 -0700 (PDT)
Subject: A multipart example
Content-Type: multipart/mixed; boundary=unique-boundary-1

This is the preamble area of a multipart message. Mail readers that understand multipart format
should ignore this preamble. If you are reading this text, you might want to consider changing to a mail reader
that understands how to properly display multipart messages.

--unique-boundary-1
Content-type: text/plain; charset=US-ASCII

… Some text …

--unique-boundary-1
Content-Type: multipart/parallel; boundary=unique-boundary-2

--unique-boundary-2
Content-Type: audio/basic
Content-Transfer-Encoding: base64

... base64-encoded 8000 Hz single-channel mu-law-format audio data goes here ...

--unique-boundary-2
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

... base64-encoded image data goes here ...
```

MIME – Example



```
--unique-boundary-1
Content-type: text/enriched

This is <bold><italic>enriched.</italic></bold><smaller>as defined in RFC
1896</smaller>
Isn't it <bigger><bigger>cool?</bigger></bigger>

--unique-boundary-1
Content-Type: message/rfc822

From: (mailbox in US-ASCII)
To: (address in US-ASCII)
Subject: (subject in US-ASCII)
Content-Type: Text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: Quoted-printable

... Additional text in ISO-8859-1 goes here ...

--unique-boundary-1--
```

S/MIME services

- Enveloped data (application/pkcs7-mime; smime-type = enveloped-data)
  - standard digital envelop
- Signed data (application/pkcs7-mime; smime-type = signed data)
  - standard digital signature (“hash and sign”)
  - content + signature is encoded using base64 encoding

S/MIME services

- Clear-signed data (multipart/signed)
  - standard digital signature
  - only the signature is encoded using base64
  - recipient without S/MIME capability can read the message but cannot verify the signature
- Signed and enveloped data
  - signed and encrypted entities may be nested in any order

Cryptographic algorithms

- Message digest
  - must: SHA-1
  - should (receiver): MD5 (backward compatibility)
- Digital signature
  - must: DSS
  - should: RSA
- Asymmetric-key encryption
  - must: ElGamal
  - should: RSA

Cryptographic algorithms

- Symmetric-key encryption
  - sender:
    - should: 3DES, RC2/40
  - receiver:
    - must: 3DES
    - should: RC2/40

Securing a MIME entity

- S/MIME secures MIME entity with signature, encryption or both
- MIME entity is prepared according to the normal rules for MIME message preparation
- Prepared MIME entity is processed by S/MIME to produce a PKCS object
- The PKCS object is treated as message content and wrapped in MIME

Enveloped Data

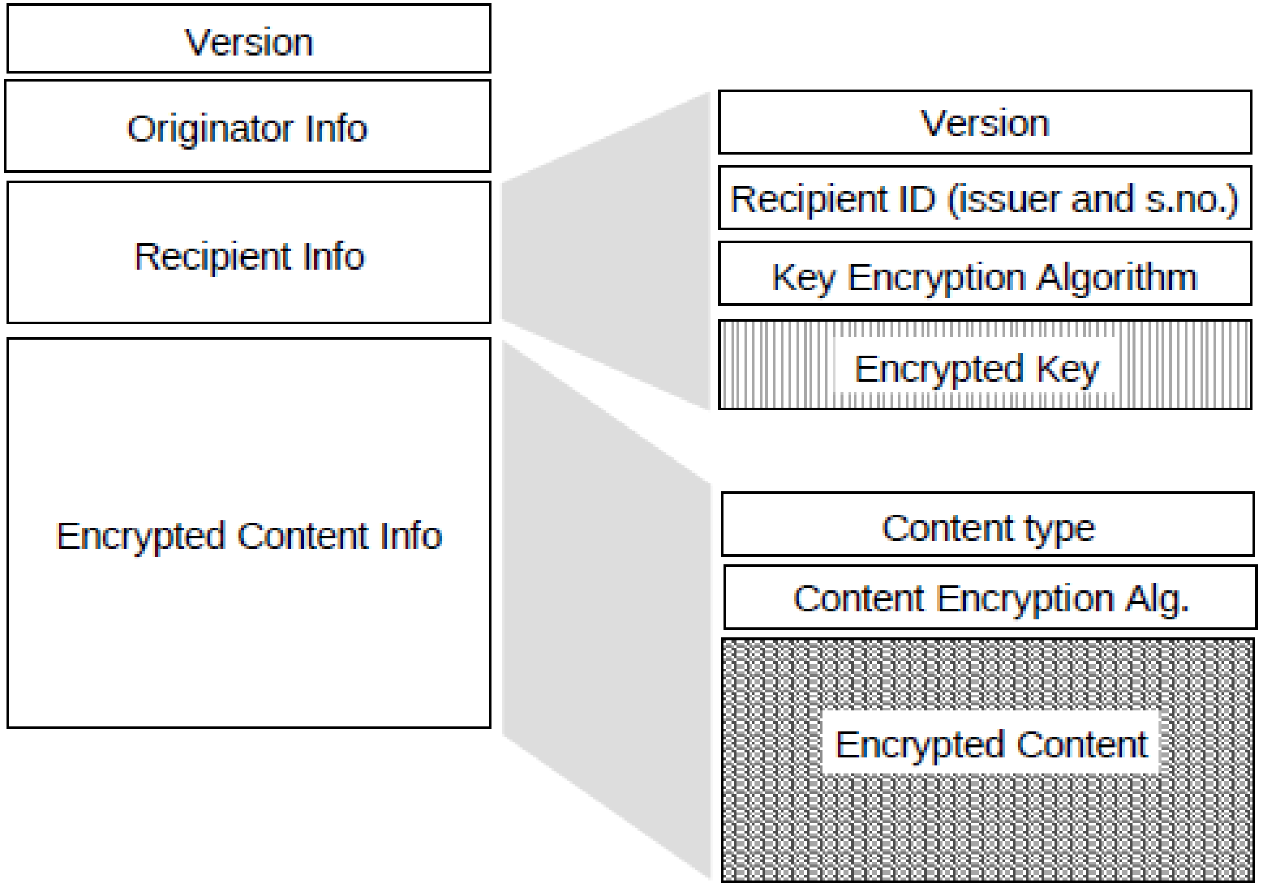
- Create a pseudorandom session key for a symmetric encryption
- For each recipient, encrypt session key with recipient's public key
- For each recipient, prepare a block known as Recipient Info that contains
  - Identifier of recipient's public key certificate,
  - Identifier of algorithm
  - Encrypted session key
- Encrypt the message content with session key

Enveloped data – Example

- Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m
- Content-Transfer-Encoding: base64
- Content-Disposition: attachment; filename=smime.p7m

```
rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG
QpfyF467GhIGfHfYT6
7n8HHGghyHhHUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB
9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfy
F4 0GhIGfHfQbnj756YT64V
```

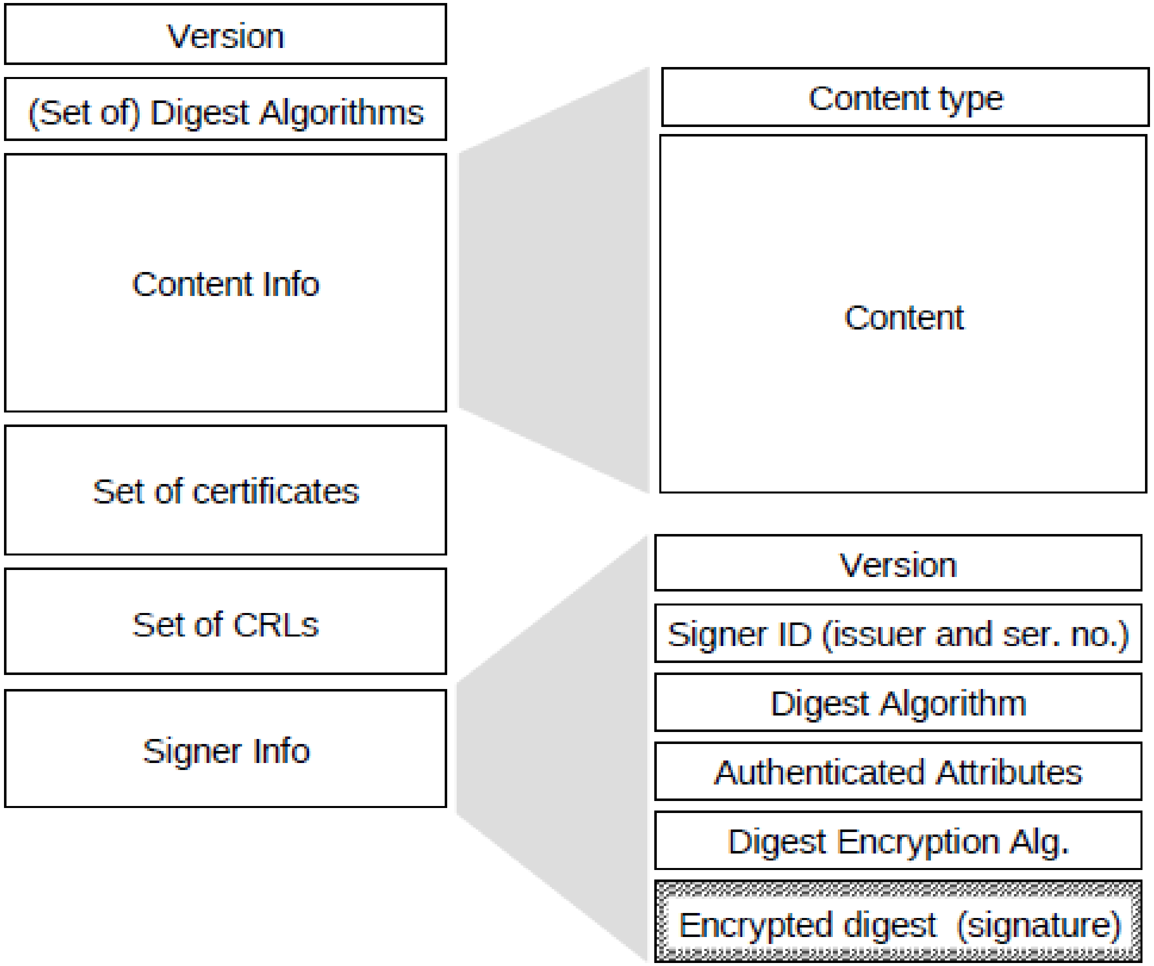
PKCS7 “enveloped data



Signed Data

- Can be used with one or more signers
- Select a message digest algorithm
- Compute the message digest, or hash functions, of the content to be signed
- Encrypt the message digest with signer's private key
- Prepare a block known as SignerInfo that contains signer's public key certificate
  - Identifier of the message digest algorithm
  - Identifier of the algorithm for encryption
  - Encrypted message digest

PKCS7 “signed data”



Clear Signing

- Achieved using multipart content type with a signed subtype
- Two parts
  - MIME type, must be prepared so that not altered during transfer
  - MIME content type of application and subtype of PKCS7-signature

Clear-signed data – Example

```
Content-Type: multipart/signed; protocol="application/pkcs7-signature";
  micalg=sha1; boundary=boundary42

--boundary42
Content-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
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GhIGfHfYT64VQbnj756

--boundary42--
```

Key Management

- Certificates are signed by certification authorities (CA)
- Key authentication is based on chain of certificates
- Users/Managers are responsible to configure their clients with a list of trusted root keys

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 certificates
- And own public/private key pairs & certificates
- Certificates must be signed by trusted CA's

User Agent Role

- S/MIME user has several key management functions
  - Key generation: User must be able to generate separate Diffie-Hellman and DSS key pairs
  - Registration: User's public key must be registered with CA
  - Certificate Storage and Retrieval: User requires access to a local list of certificates for verification of signatures

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

Any question ?