School of Electrical Engineering and Computing

SENG2250/6250
SYSTEM AND NETWORK SECURITY
(S2, 2020)

Email Security

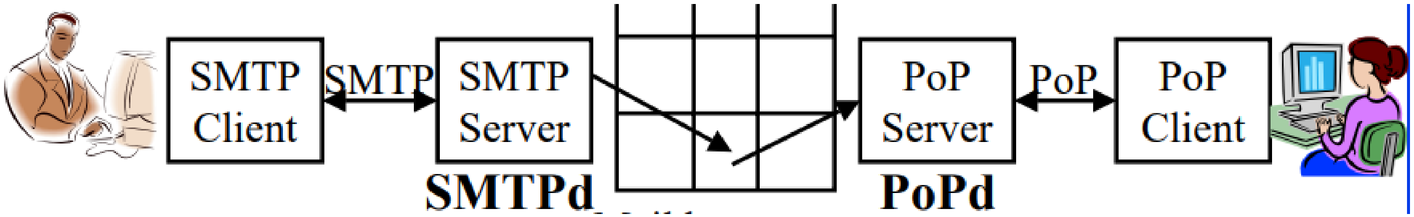
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Outline

- SMTP
- Email Security
- Pretty Good Privacy (PGP)
- S/MIME

Email Overview

- Simple Mail Transfer Protocol (SMTP)
  - *Transfer email from one user to anther user's **mailbox.***
- Post Office Protocol (PoP)
  - *Retrieve email from mailbox*
  - *Authenticates user*
- Internet Mail Access Protocol (IMAP)
- Multimedia Internet Mail Encoding (MIME)
  - *To encode non-text messages*



SMTP

- Defined in RFC 2821 and RFC 2822
- Clients connect to port 25 of SMTP server
- It is a push protocol and does not allow to pull
- Extended SMTP (ESMTP) is defined in RFC 2821
- ESMTP uses EHLO in stead of HELO
- ESMTP allows finding the maximum message size
- SMTP-AUTH is an authentication extension to SMTP (RFC 4954)
- Allows only authorized users to send email

SMTP

- SMTP defines a mechanism for electronic mail based on TCP/IP. It supports
  - *Sending a single message to one or more recipients identified by email address.*
  - *Sending messages that include text, voice, video, ore graphics. Sending message outside the Internet.*
- SMTP Mechanism
  - *A human user uses a user agent (UA) to prepare the message contains header and body*
  - *Creating the envelope containing the sender's address, receiver's address, and other information*
  - *The Message Transfer Agent (MTA) transfers the mail across the Internet, from MTA client to MTA server.*
  - *The user agent periodically checks the mailbox.*

Email Security

- An email message is made up of string of ASCII characters in a format specified by RFC 822.
- Then, such a message travels to the recipient via Internet.
- Email is a widely used network-based application.
- Email is very popular mainly due to its convenience.

Email Security

However, email has very weak security

- Lack of confidentiality
  - *Sent in clear over open networks.*
  - *Stored on potentially insecure clients and servers.*
- Lack of integrity
  - *Both the header and content can be modified.*
- Lack of authentication
  - *The sender of an email is also forgeable.*
- Lack of non-repudiation
  - *The sender can later deny having sent an email.*
  - *The recipient can later deny having received the message.*

PGP Overview

- PGP - Pretty Good Privacy
- Provides confidentiality and authentication services to exchange the security for email transmission and storage.
- A widely used de facto standard for secure email.
- Developed by Philip Zimmermann.
- Strong crypto algorithms are integrated into a single application, which is independent of OS platforms.
- Originally free software, though commercial versions are also available.
- PGP is on an Internet standard track, RFC3156.

Summary of PGP Services

Function	Algorithms (examples)
Digital Signature	DSS/SHA-1 or RSA/SHA-1
Message Encryption	CAST, IDEA, 3DES, RSA, ElGamal
Compression	ZIP
Email Compatibility	Radix-64 conversion
Segmentation	-

PGP Operational Description

- Operational Description
  - *Authentication*
  - *Confidentiality*
  - *Confidentiality and Authentication*
  - *Email Compatibility*
  - *Segmentation and Reassembly*

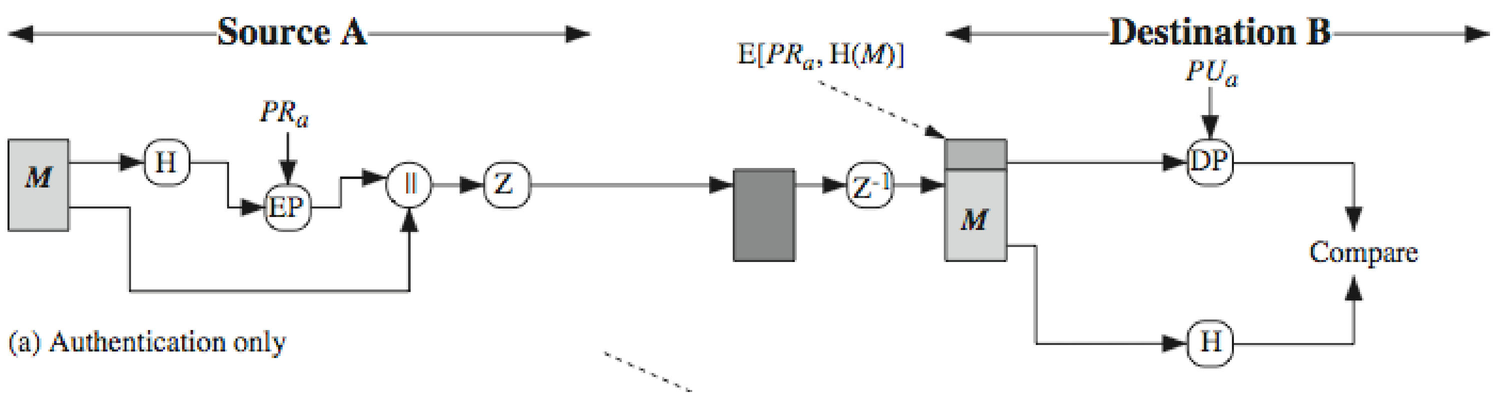
PGP Operational Description

Notations	
Ks:	One-time session key
PRa:	Private key of user A
PUa:	Public key of user A
EP:	Public key encryption
DP:	Public key decryption
EC:	Symmetric key encryption
DC:	Symmetric key decryption
H:	Hash function
||:	Concatenation
Z:	Compression using ZIP algorithm
R64:	Conversion to radix 64 ASCII format

PGP Operational Description

- Authentication only:

1. Sender creates message
2. Make SHA-1 160-bit hash of message
3. Attached RSA signed hash to message
4. Receiver decrypts & recovers hash code
5. Receiver verifies received message hash

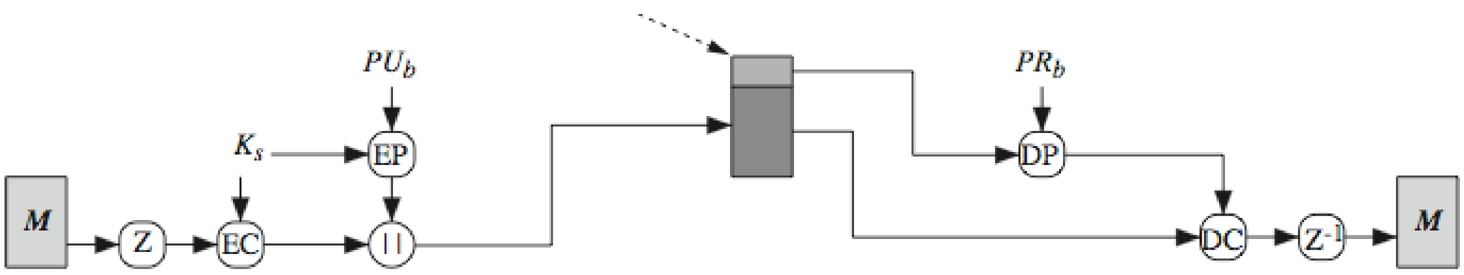


(a) Authentication only

PGP Operational Description

- Confidentiality only:

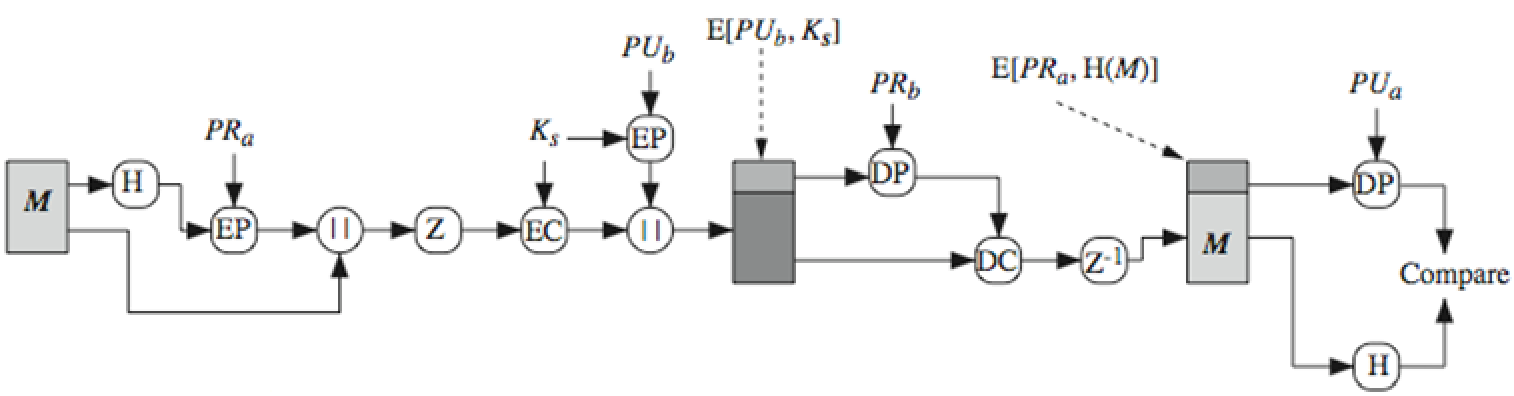
1. Sender forms 128-bit random session key
2. Encrypts message with session key
3. Attaches session key encrypted with RSA
4. Receiver decrypts & recovers session key
5. Session key is used to decrypt message



(b) Confidentiality only

PGP Operational Description

- Confidentiality and Authentication:
- Two services on the same message
  - *Create signature and attach to message*
  - *Encrypt both message and signature*
  - *Attach RSA/ElGamal encrypted session key.*



PGP Operational Description

- Compression: ZIP
- The order of operations:

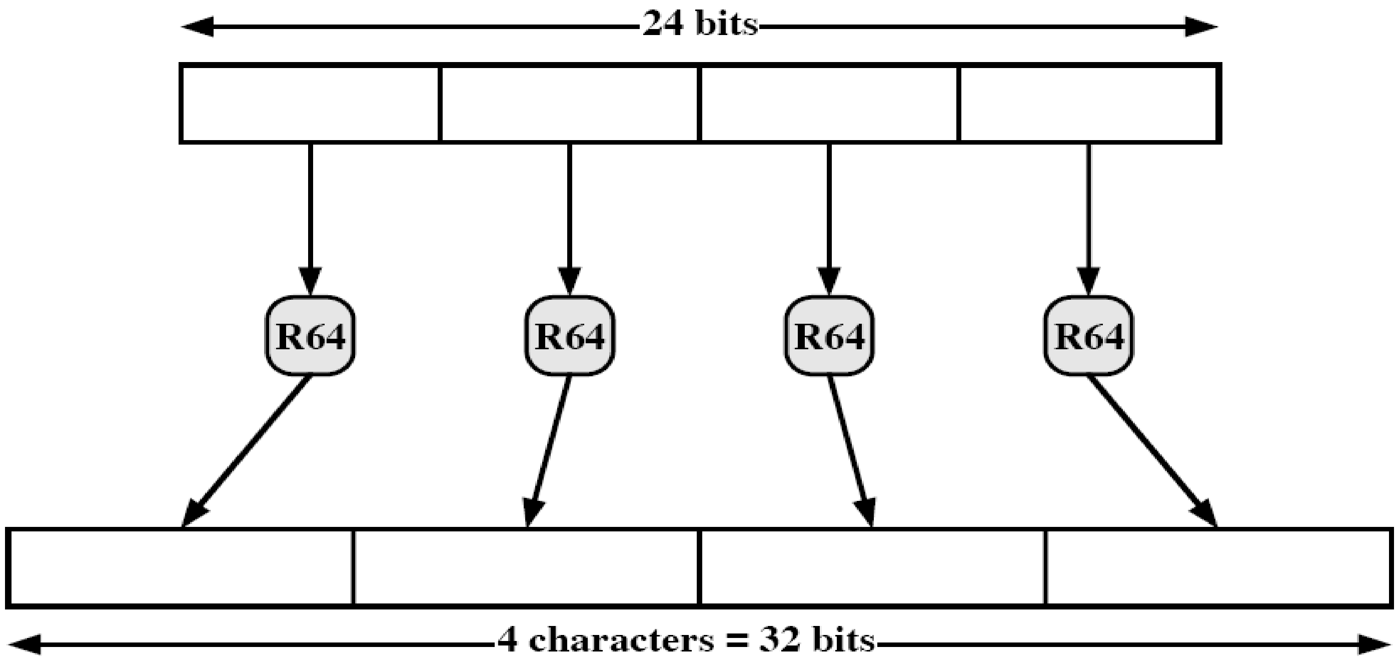
  sign → compress → encrypt
- Why PGP follows this order?
  - *More convenient to store a signature with plain message.*
  - *Otherwise, we need to store the session key and/or run compression algorithm before validating a signature.*

PGP Operational Description

- Email Compatibility:
  - *After the above security operations, the resulting message will contain some arbitrary octets.*
  - *PGP needs to convert the raw 8-bit binary stream into a stream of printable ASCII characters.*

PGP Operational Description

- Therefore, the radix-64 conversion is used.
- This operation expands the message by 33%.



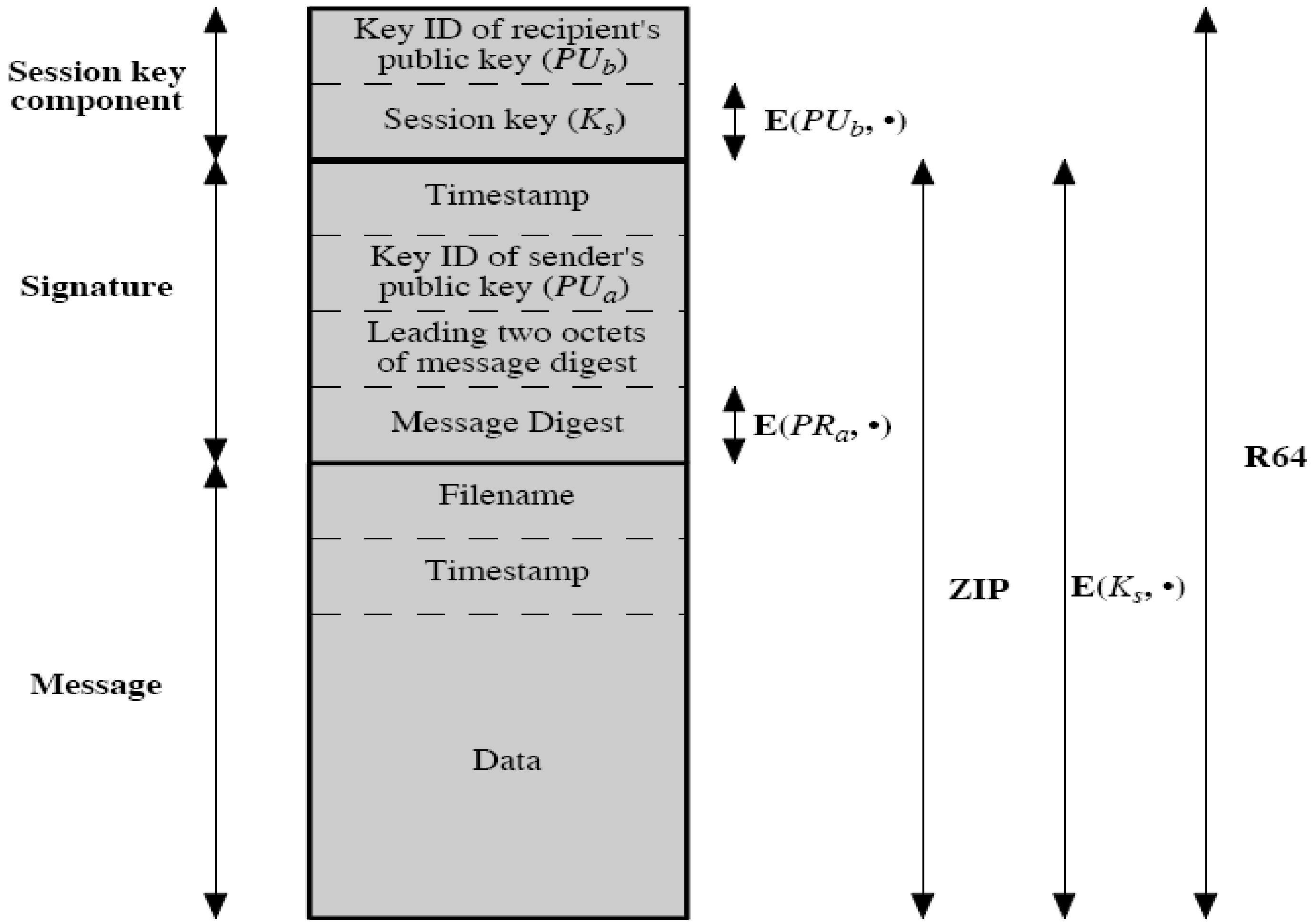
Key Generation and Key Rings

- Session Key Generation:
  - *Each session key is only associated with one message.*
  - *Randomness is generated based on keystroke input from the user, where both the keystroke timing and the actual keys struck are used to generate a randomized stream of numbers.*

Key Generation and Key Rings

- Key Identifiers (Key IDs):
  - *One user usually needs multiple public/private key pairs.*
  - *How to let the receiver know which key pair is used?*
  - *Trivial approach*
    - Receiver tries each possible public key
  - *PGP uses the Key ID to identify a public key*
    - **Key ID** = (PUa mod 2^{64}), i.e. the least significant 64 bits of the public key.

Key Generation and Key Rings



Key Generation and Key Rings

- Key Rings:
  - *Each user maintains two key rings in his/her system.*
  - *A private-key ring stores the private/public key pairs owned by the user.*
  - *A public-key ring stores the public keys of other users.*
  - *Both rings can be indexed by either User ID or Key ID.*

Key Generation and Key Rings

Private Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
• • •	• • •	• • •	• • •	• • •
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
• • •	• • •	• • •	• • •	• • •

Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
T_i	$PU_i \bmod 2^{64}$	PU_i	trust_flag_i	User i	trust_flag_i		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

* = field used to index table

Figure 15.4 General Structure of Private and Public Key Rings

Key Generation and Key Rings

- In the above diagram, Pi is the user's password.
- Security of private keys depends on the pass-phrase security.

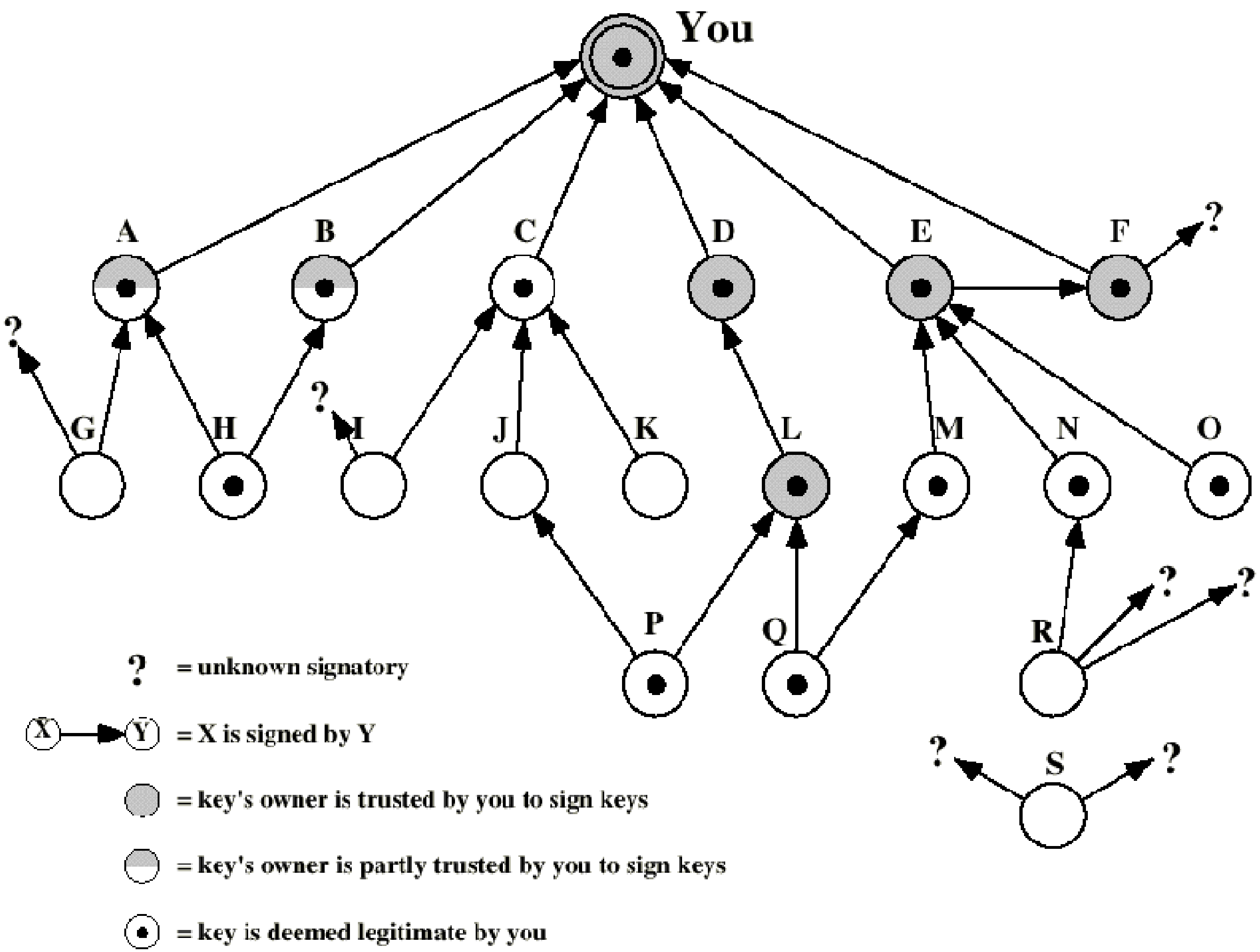
PGP Public Key Management

- Traditionally, public keys are certified by trusted CAs, using PKI.
- PGP uses a completely different trust model – the web of trust.
- Each PGP user assigns a trust level to other users (Owner Trust Field)
- Each user can certify (sign) the public keys of users he/she knows.
- In the public key ring, each entry stores a number of signatures that certify this public key.
- PGP automatically computes a trust level for each public key (Key Legitimacy Field) in the key ring.

PGP Public Key Management

- Trust levels
  - *Undefined*
  - *Unknown*
  - *Partially trusted*
  - *Always trusted*
  - *Ultimately trusted (for own keys)*

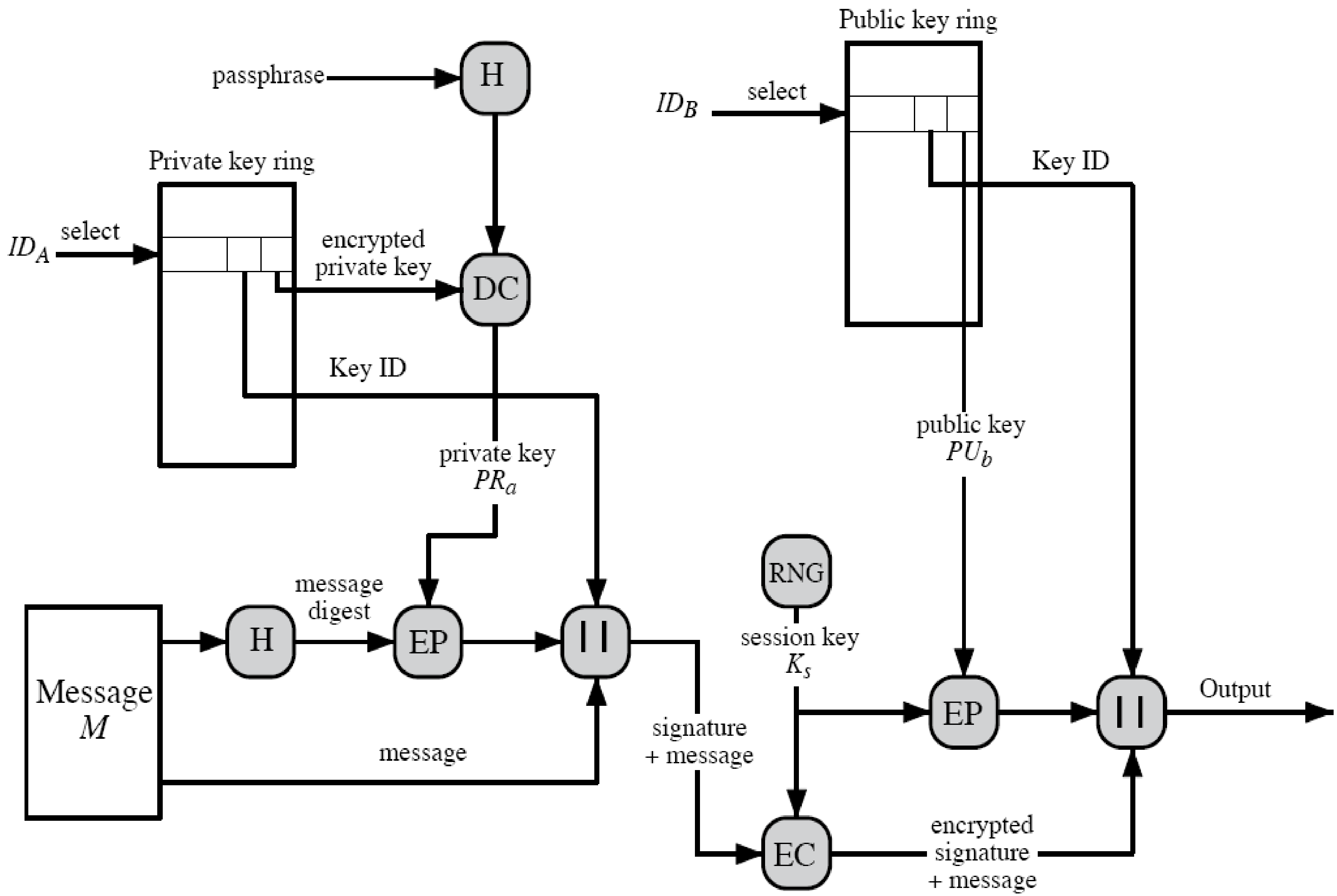
PGP Public Key Management



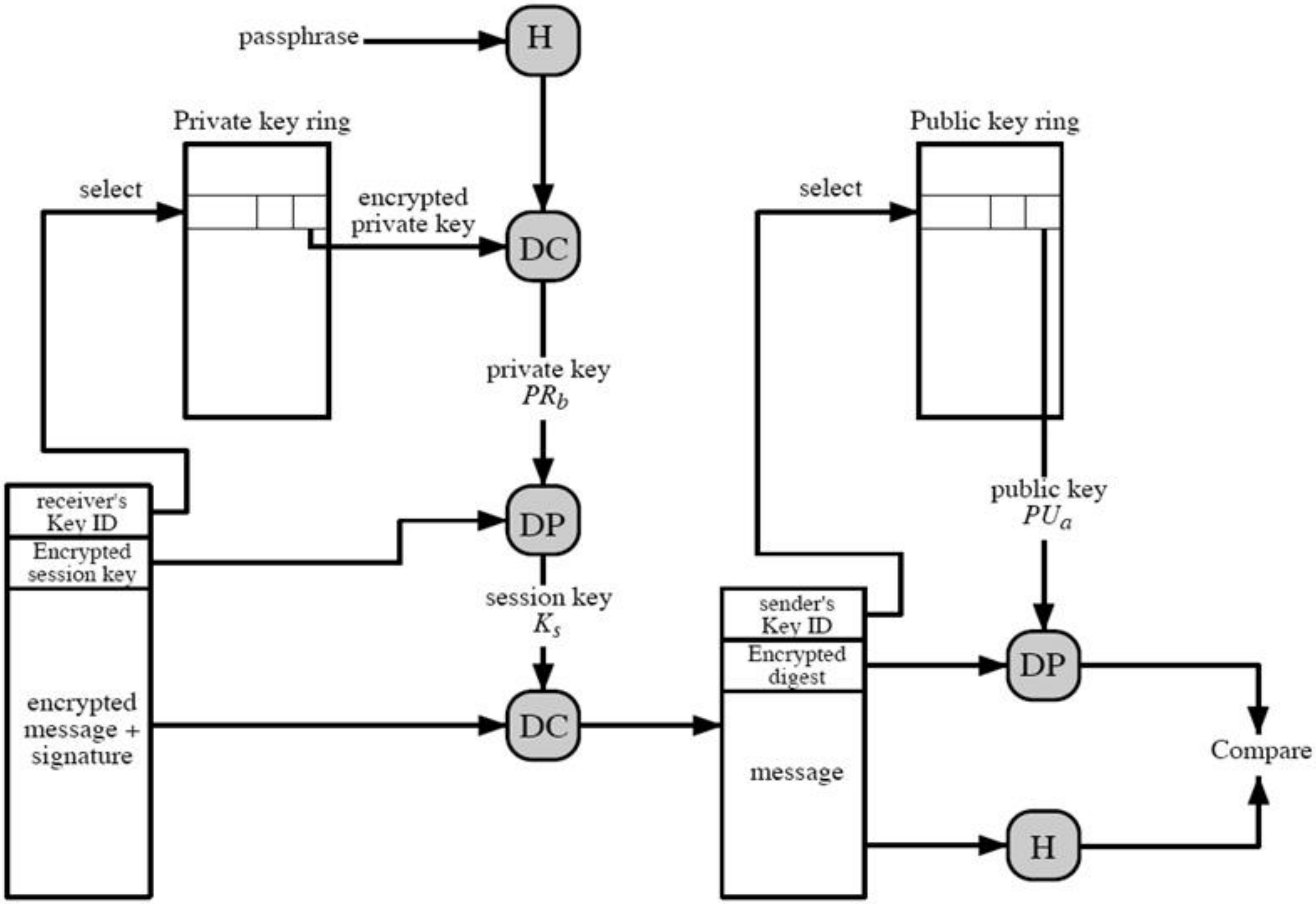
PGP Public Key Management

- Comments on the above example:
  - *(X) → (Y) means that X's public key is signed by Y.*
  - *A shading circle shows a user that is trusted by you.*
  - *A half shading circle shows a user is partially trusted by you. A public key is also trusted if it has been certified by at least two partially trusted users.*
  - *A solid dot shows that the public key for this user is trusted by you.*

PGP Message Generation



PGP Message Reception



No compression or radix-64 conversion.

MIME RFC 822

- **S/MIME** (Secure/Multipurpose Internet Mail Extensions)
  - *A security enhancement to MIME email*
  - *based on technology from RSA Data Security (Now, the Security Division of EMC Corporation).*
  - *specified by RFCs 3369, 3370, 3850 and 3851.*
- **To understand S/MIME, we need first to know MIME.**

MIME RFC 822

- RFC 822 defines a format for Internet-based text mail message.
- In RFC 822, each email is viewed as having an envelope and content.
- The envelope contains all information needed for email transmission and delivery.
- RFC 822 applies only to the contents.
- The content has two parts, separated by a blank line:
  - ***The header:*** *Date, From,To, Subject, …*
  - ***The body:*** *containing the actual message.*

MIME RFC 822

```
Date: Fri, 5 August 2011 13:58:44
From: guilin@uow.edu.au
Subject: RFC 822 example
To: alice123@hotmail.com
Cc: guilin@uow.edu.au

This is just a test message to
illustrate RFC 822. It’s not very
long and it’s not very exciting …
```

MIME

MIME is intended to avoid a number limitations in RFC 822:

- Extends the capabilities of RFC 822 to allow email to carry messages with non-textual content and non-ASCII character sets.
- Supports long message transfer.
- Introduces new header fields in RFC 822 email to specify the format and content of extensions.
- Supports a number of content types together with a number of encoding schemes.
- Specified in RFCs 2045-2049.

MIME

Five new fields are defined in MIME:

- **MIME-Version:** version number
- **Content-Type:** Describes the data contained in the message body.
- **Content-Transfer-Encoding:** Indicates which of encoding schemes is used to represent the body data.
- **Content-ID (optional):** Identifies a message uniquely.
- **Content-Description (optional):** A text description of the object with the body (useful if the object is not readable).

MIME Content Type

MIME defined 7 major content types with 15 subtypes:

- **Text:** Plain / Enriched
- **Multipart:**
  - *Mixed: Ordered independent parts.*
  - *Parallel: Unordered independent parts (e.g., a picture accompanied by a voice).*
  - *Alternative: Different versions of the same message.*
- **Message:** rfc822 / Partial / External-body
- **Image:** jpeg / gif
- **Video:** mpeg
- **Audio:** Basic
- **Application:** PostScript / octet-stream

MIME Content-Transfer-Encoding

- RFC 822 emails can contain only ASCII characters.
- MIME messages are intended to transport arbitrary data.
- The Content-Transfer-Encoding field indicates how data was encoded from raw data to ASCII.
- Base64 (i.e Radix-64) is a common encoding:
  - *24 data bits (3 bytes) are encoded into 4 ASCII characters (4 bytes).*

S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extensions):

- A security enhancement to MIME email.
- Specified by RFCs 3369, 3370, 3850 and 3851.
- Widely supported in many email agents:
  - *MS Outlook, Mozilla, Mac Mail, Netscape Messenger, Lotus Notes etc.*

S/MIME

- Functions
- Algorithms
- Processing
- Certificate management

S/MIME Functions

Similar to PGP, S/MIME provides the following functions to secure email:

- **Enveloped Data:** encrypted message and session key.
- **Signed Data:** encoded message plus signature.
- **Clear-Signed Data:** clear message + encoded signature.
- **Signed and Enveloped:** nesting of signed and encrypted entities.

S/MIME Algorithms

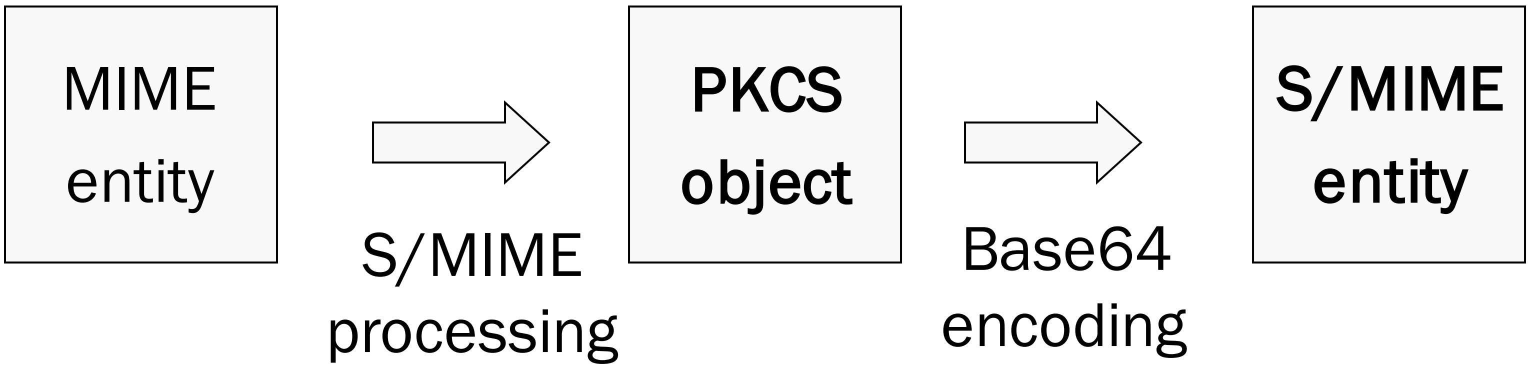
S/MIME supports the following algorithms.

- Digital signatures: DSS & RSA
- Hash functions: SHA-1 & MD5
- Session key encryption: ElGamal & RSA
- Message encryption: AES, Triple-DES, RC2/40 and others
- MAC: HMAC with SHA-1

S/MIME Processing

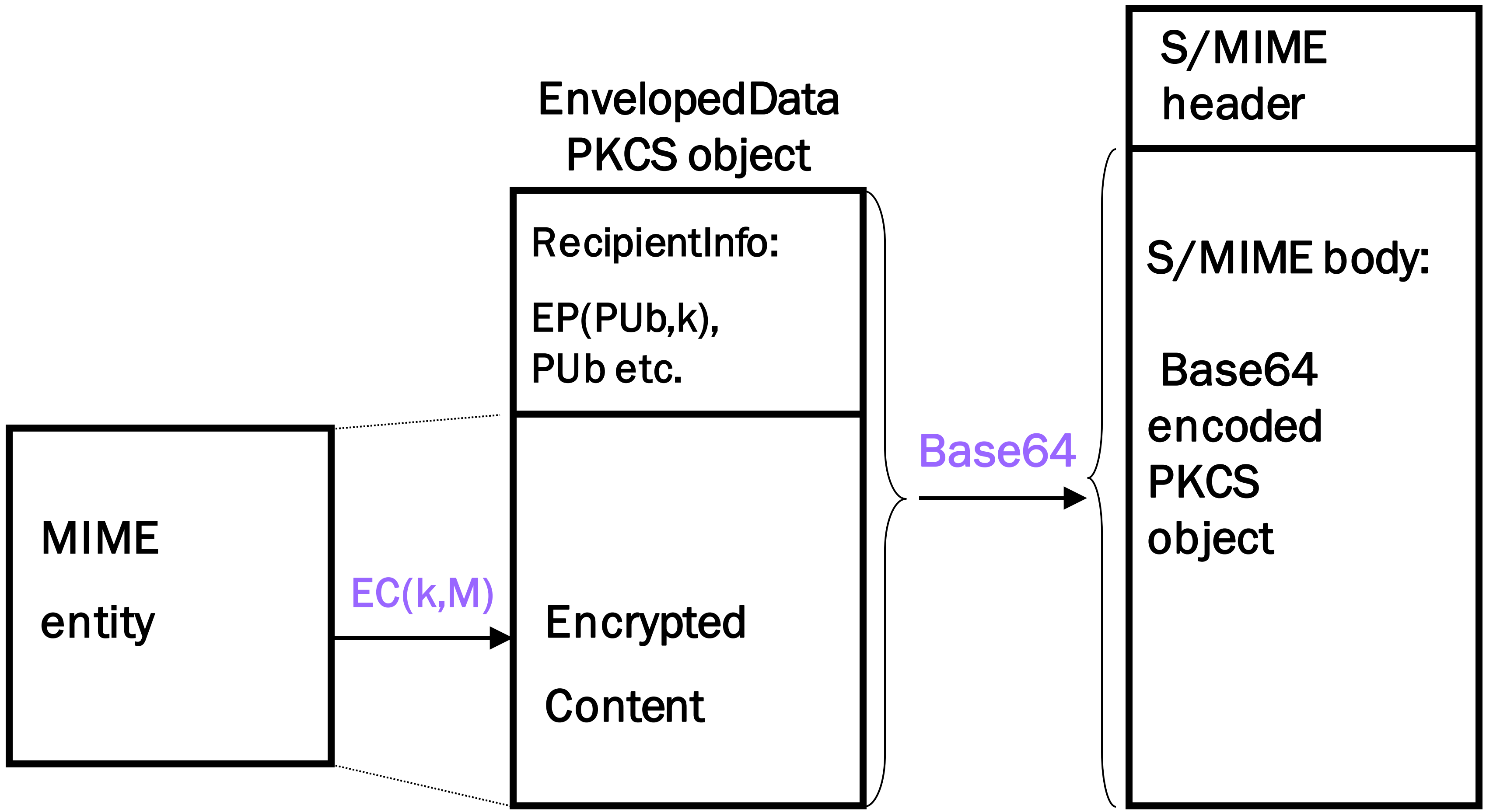
- The MIME entity is prepared normally by MIME rules.
- Then, MIME entity plus some security related data are processed by S/MIME to produce a PKCS object.
- Finally, a PKCS object is treated as message content and wrapped into <u>an MIME message</u>.

S/MIME Processing

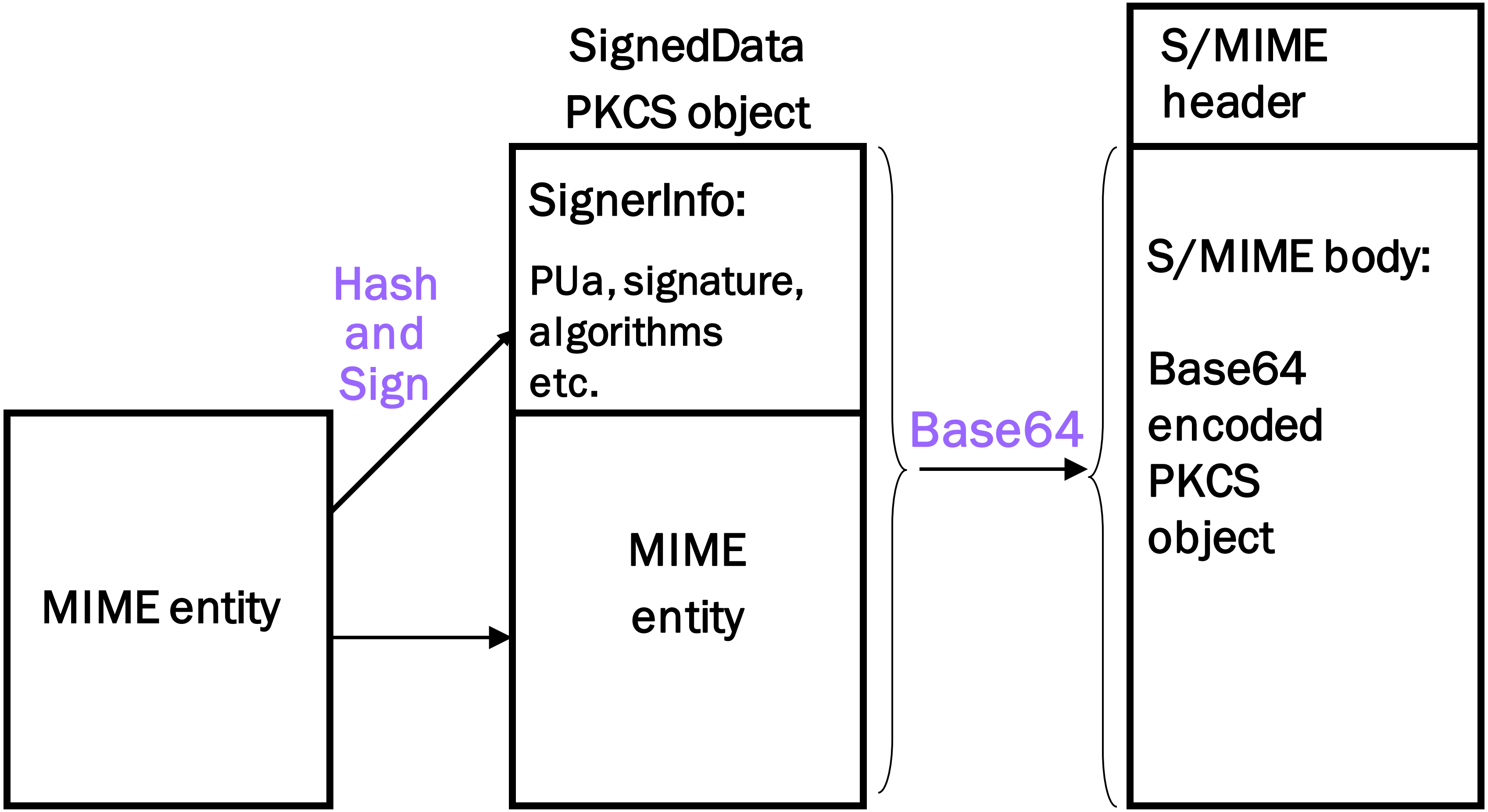


- PKCS: Public Key Cryptography Standard.
- A PKCS object includes the original content plus all information needed for the recipient to perform security processing.

S/MIME EnvelopedData



S/MIME SignedData



S/MIME Certificate Management

- S/MIME uses X.509 v3 certificates
- have several well-known CA's
- Verisign one of most widely used
- Verisign issues several types of Digital IDs
- 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