Electronic Mail Security

Outline

- Email Security
- PGP Overview
- PGP Operational Description
- PGP Key Generation and Key Rings
- PGP Public-Key Management
- MIME (& RFC 822)
- S/MIME

1. Email Security

Email (electronic mail):

- An email message is made up of a 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.
- Moreover, email is the only distributed application that is widely used across all architectures and platforms.
- Email is very popular mainly due to its convenience.

1. Email Security

However, basic 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.

1. Email Security

In this lecture, we are going to discuss email security

- **PGP: Pretty Good Privacy (https://www.openpgp.org/)**
- **S/MIME: Secure/Multipurpose Internet Mail Extensions**

2. PGP Overview

Basically, PGP provides confidentiality and authentication services to enhance the security for email transmission and storage.

- Developed by Philip Zimmermann.

- PGP and OpenPGP operations are specified in a few documents (RFC 2015, 3156, 4880).

2. PGP Overview

Summary of PGP Services

Function	Algorithms Used
Digital Signature (Authentication)	DSS/SHA or RSA/SHA
Message Encryption	CAST, IDEA, 3DES, AES, RSA, ElGamal, etc.
Compression	ZIP
E-mail Compatibility	Radix-64 conversion
Segmentation	-

3. PGP Operational Description

Operational Description

- Authentication
- Confidentiality
- Confidentiality and Authentication
- Email Compatibility
- Segmentation and Reassembly

3. PGP Operational Description

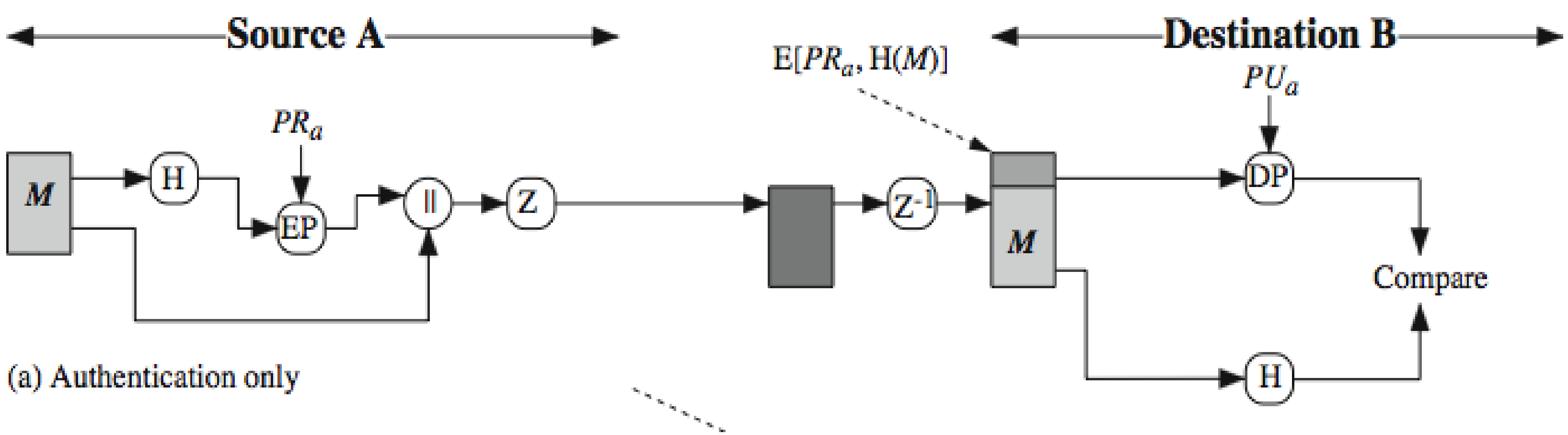
Notations

Ks: one-time session key
PRa: private key of user A
PUa: public key of user A
EP: encrypting with PU or signing with PR
DP: decrypting with PR or verifying with PU
EC: symmetric encryption
DC: symmetric decryption
H: hash function
||: concatenation
Z: compression using ZIP algorithm
R64: conversion to radix 64 ASCII format

3. PGP Operational Description

Authentication only (RSA-SHA1):

1. Sender creates a message and its SHA1 160-bit hash
2. Sender signs the hash with RSA and prepends the signature to the message
3. Receiver hashes the message and verifies the signature

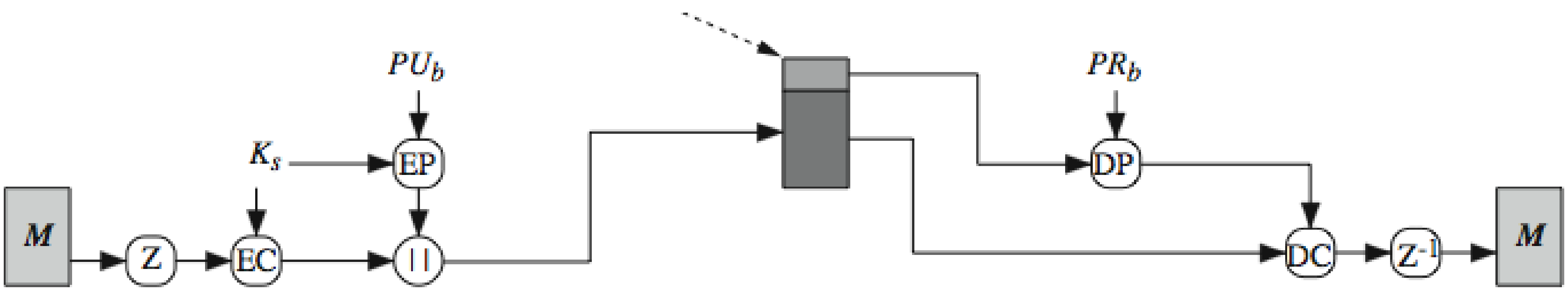


(a) Authentication only

3. PGP Operational Description

Confidentiality only:

1. sender generates a 128-bit random session key
2. encrypts message with session key
3. attaches session key encrypted with RSA/ElGamal
4. receiver decrypts & recovers session key
5. session key is used to decrypt message

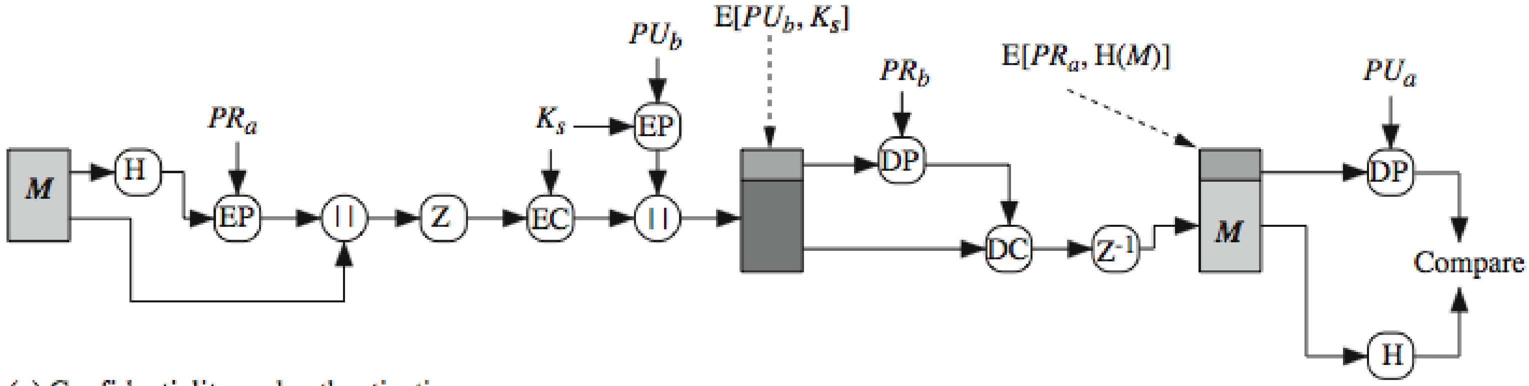


(b) Confidentiality only

3. PGP Operational Description

Confidentiality and Authentication:

- can use both services on same message
  - create signature & attach to message
  - encrypt both message & signature
  - attach RSA/ElGamal encrypted session key



(c) Confidentiality and authentication

3. PGP Operational Description

Compression: Using ZIP.

- The order of operations: sign→compress→encrypt.
- More convenient to store a signature with plain message.

Q: what about encrypt then sign?

3. PGP Operational Description

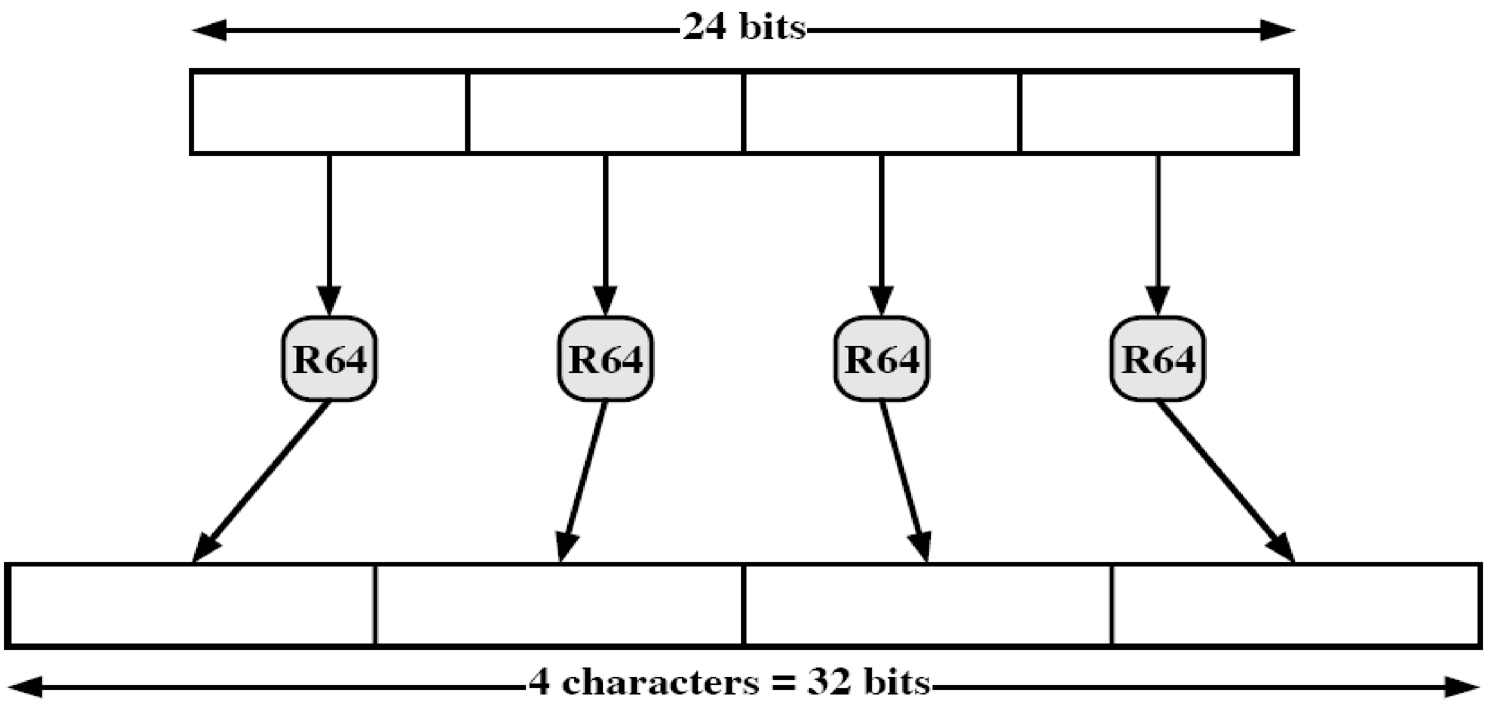
Email Compatibility:

- 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.

3. PGP Operational Description

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



3. PGP Operational Description

Segmentation and Reassembly:

- Email systems often limit the size of a message up to 50,000 octets.
- So, a longer message must be broken up into segments.
- After all other operations, PGP automatically subdivides a long message into small segments.
- Once getting those emails, the receiver first strips of all email headers and reassemble the block, and then perform other processing.

4. Key Generation & Key Rings

Key Generation:

- RSA & RSA
- DSA & ElGamal
- RSA (sign only)
- DSA (sign only)
- Each session key (for encrypting the real email message) is only associated with one message.

4. Key Generation & Key Rings

Key Identifiers (Key IDs):

- One user may use 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 key fingerprint.

4. Key Generation & 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.

Next slide shows the structures of these two key rings.

4. Key Generation & 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

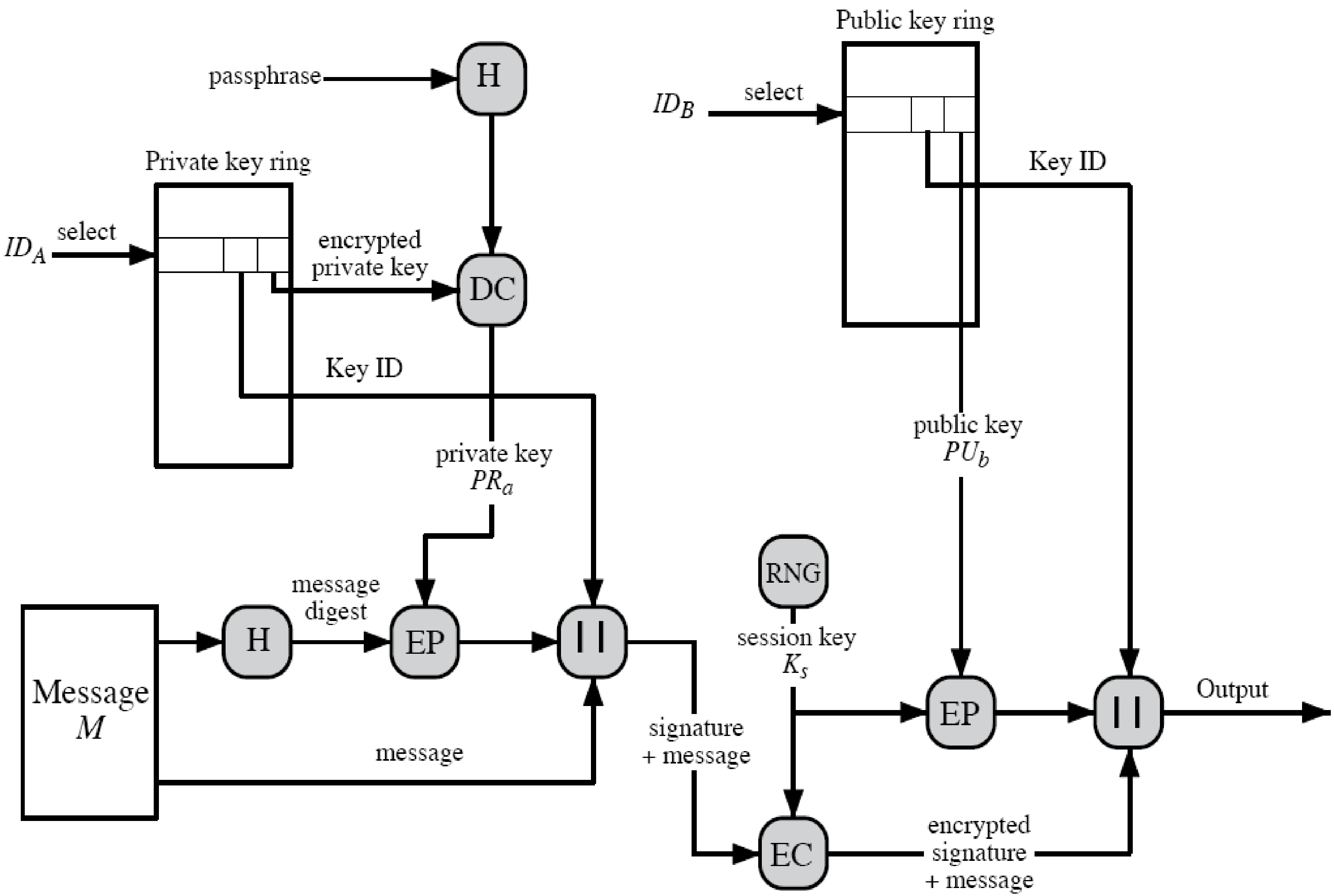
4. Key Generation & Key Rings

In the above diagram, Pi is the user’s password.

- **Security of private keys thus depends on the pass-phrase security**

Next two slides showing:

- **The Procedures to Generate a PGP Message**
- **The Procedures to Receive a PGP Message**



Public key ring
passphrase
H
ID_B
select
Private key ring
Key ID
ID_A
select
encrypted
private key
DC
Key ID
public key
PU_b
private key
PR_a
RNG
message
digest
Message
M
H
EP
| |
session key
K_s
EP
| |
Output
message
signature
+ message
EC
encrypted
signature
+ message

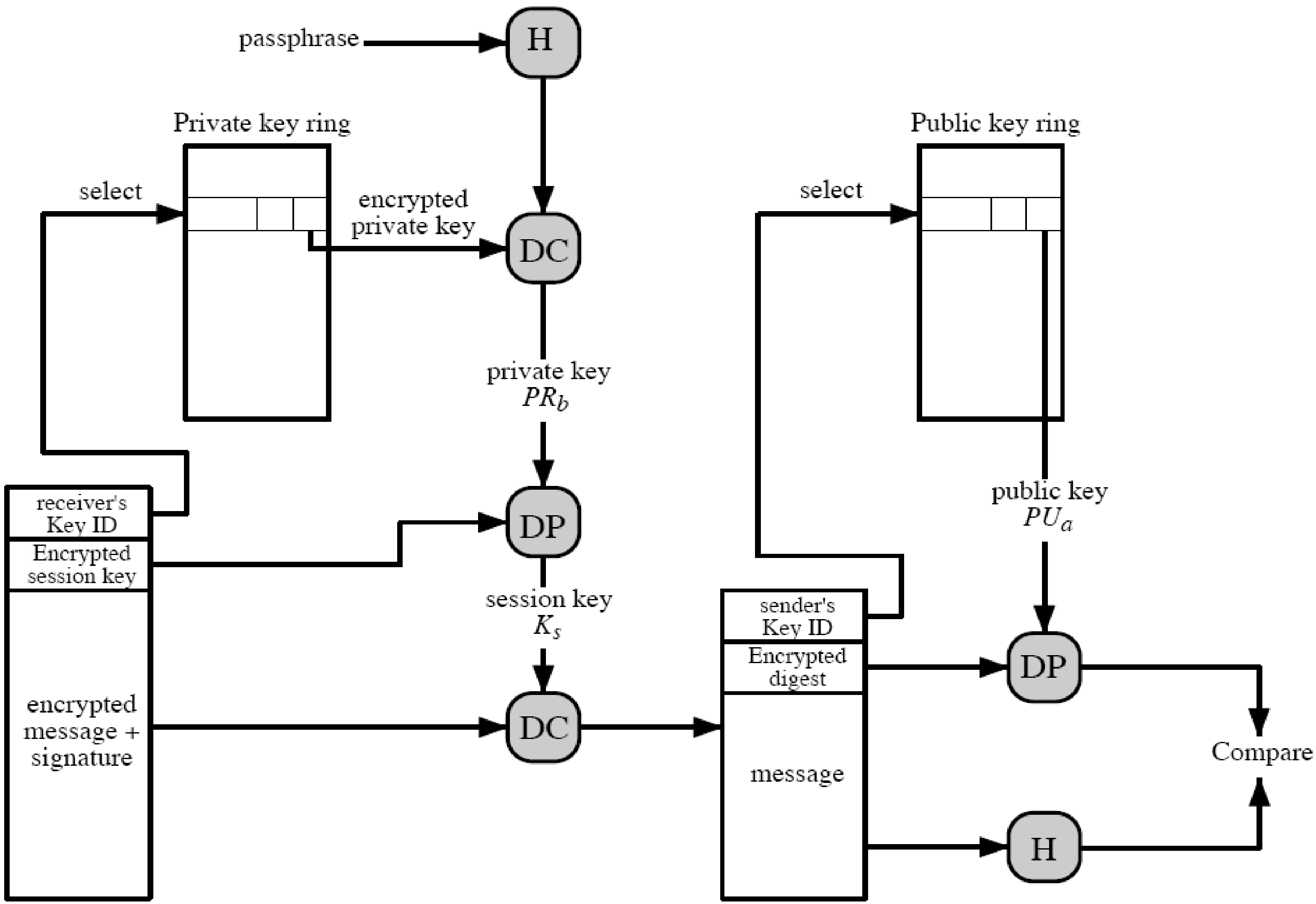


Figure 15.6 PGP Message Reception (from User A to User B; no compression or radix 64 conversion)

5. PGP Public Key Management

- In X.509, public keys are certified by trusted CAs.
- PGP uses a completely different model – **the web of trust**.

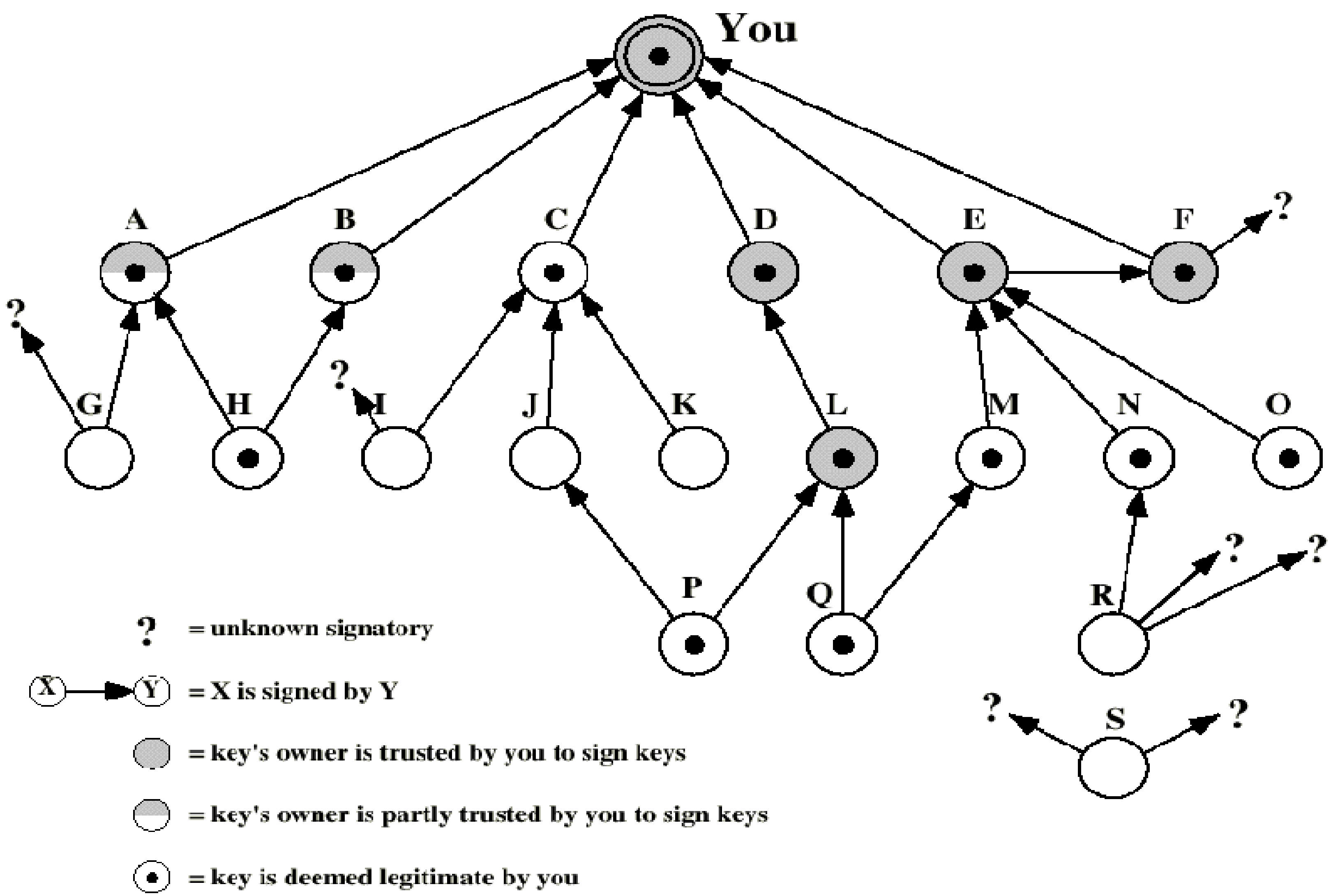
- Each PGP user assigns **a trust level to other users (Owner Trust Field)**.

- Each user can **certify** (i.e., 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.

5. PGP Public Key Management



5. PGP Public Key Management

- **(X)→(Y)** means that X’s public key is signed by Y.
- **A shading circle** shows a user (owner of the key) that is trusted by you. So, you trust all public keys certified by this user.
- **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 is trusted by you.

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.**

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.

6. 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.**

6. 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).

7. 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, Lotus Notes etc.

7. S/MIME

S/MIME

- Functions
- Algorithms
- Processing
- Certificate management

7. S/MIME: Functions

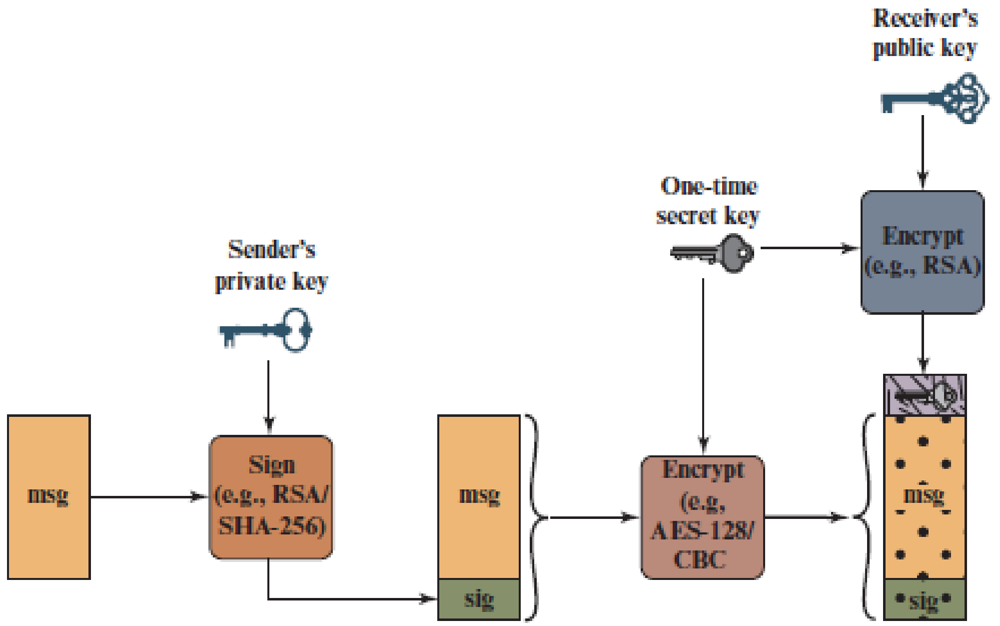
Similar to PGP, S/MIME provides the following functions to secure email:

- **Enveloped Data:** encrypted-only.
- **Signed Data:** signed-only.
- **Signed and Enveloped:** nesting of signed and encrypted entities.

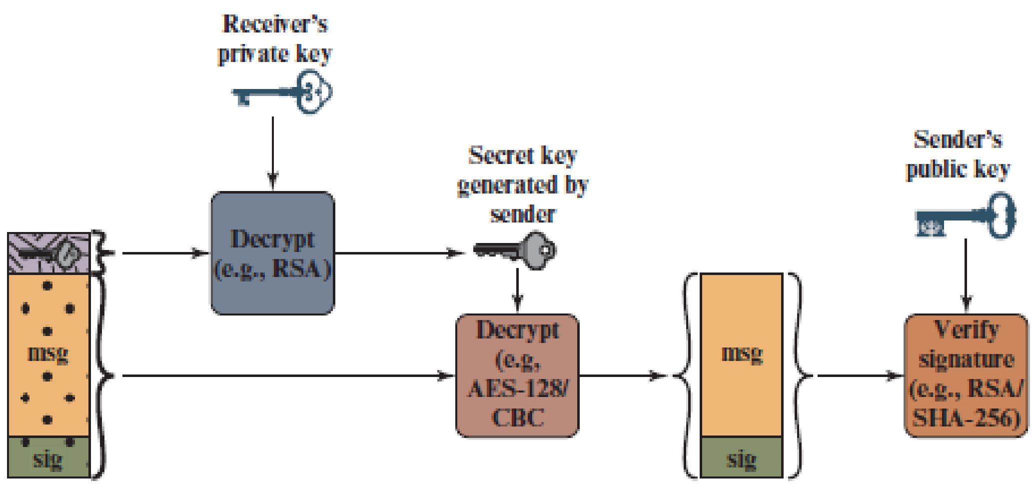
7. S/MIME: Algorithms

S/MIME supports the following algorithms.

- digital signatures: DSS & RSA
- session key encryption: ElGamal & RSA
- message encryption: AES, Triple-DES, and others
- MAC: HMAC with SHA



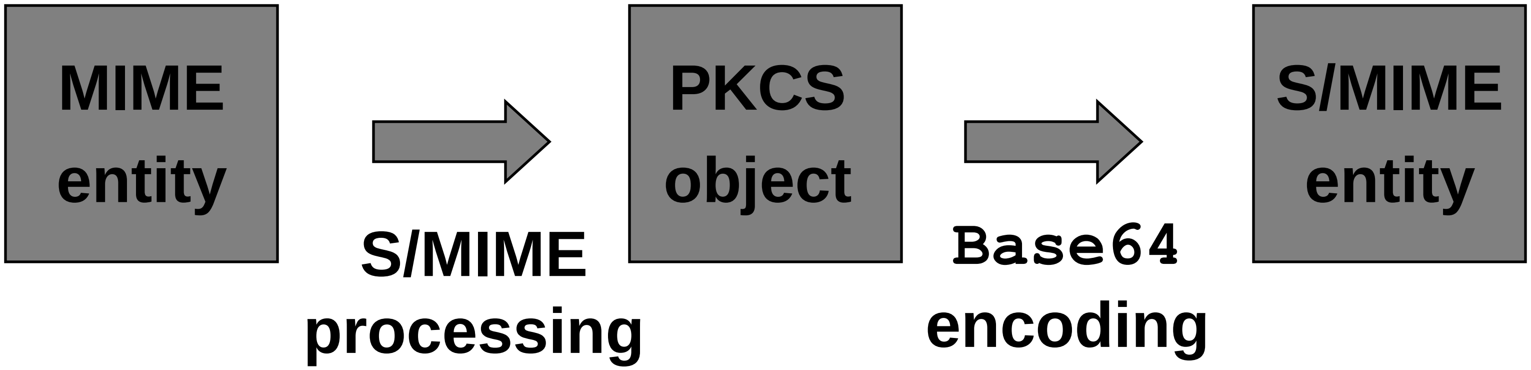
(a) Sender signs, then encrypts message



(b) Receiver decrypts message, then verifies sender's signature

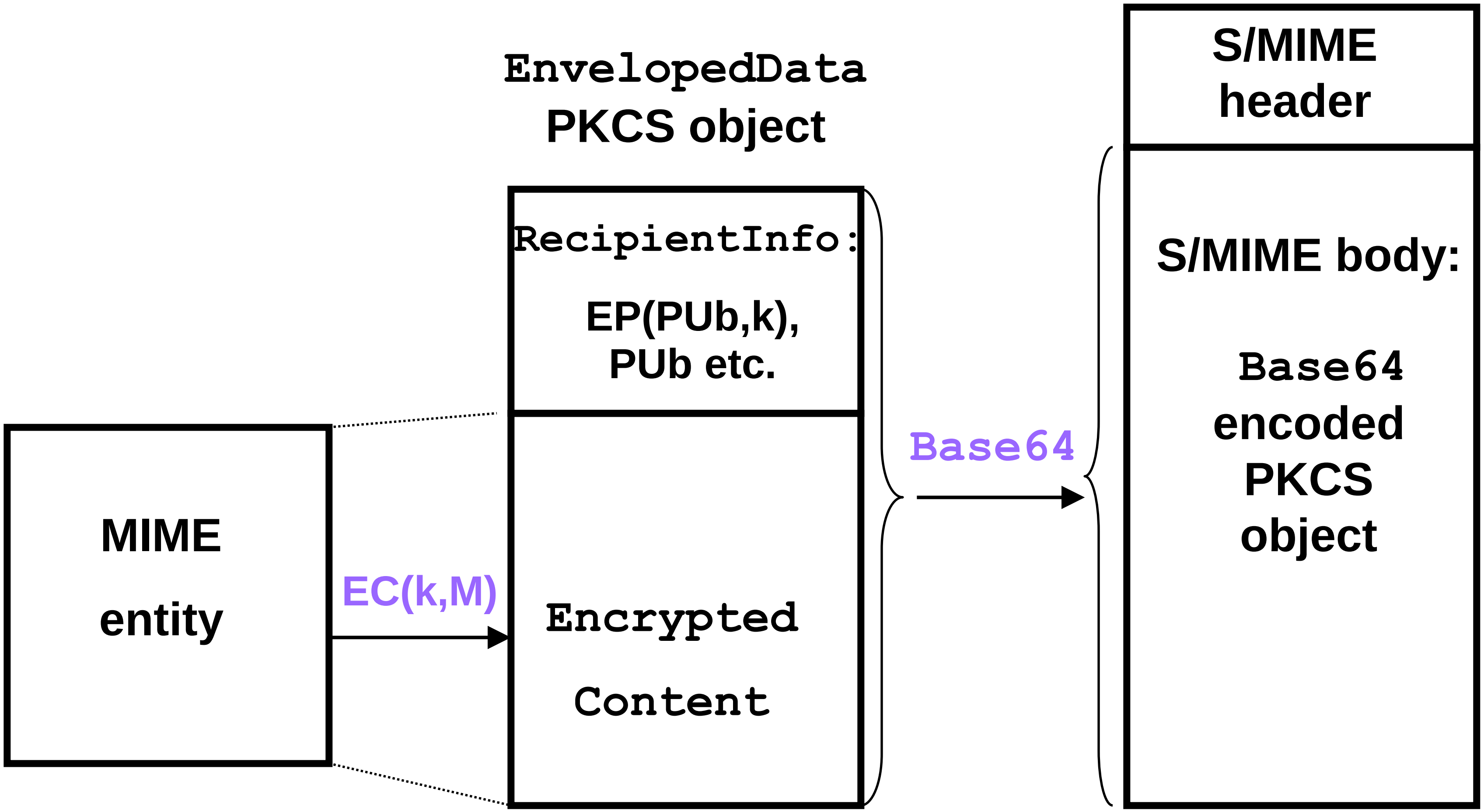
Figure 19.3 Simplified S/MIME Functional Flow

7. S/MIME: Processing

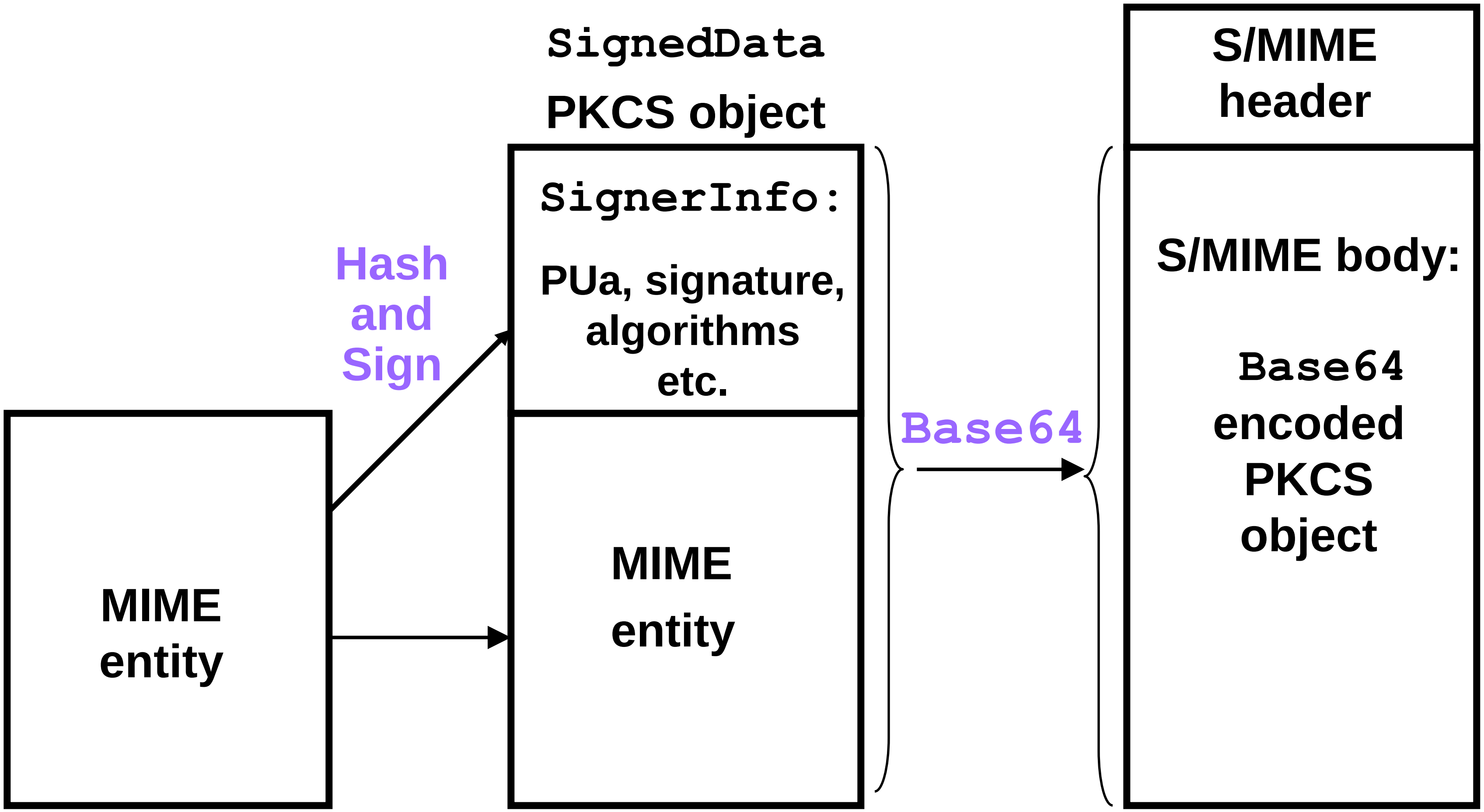


- PKCS: Public Key Cryptography Standard.
- A PKCS object includes the original content plus all information needed for the recipient to perform security processing.

7. S/MIME: EnvelopedData



7. S/MIME: SignedData



7. S/MIME: Certificate Management

- S/MIME uses X.509 v3 certificates
- 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