V	Network security: Electronic Mail Security: Pretty good privacy- S/MIME. IP Security: Architecture- authentication Header- Encapsulating Security payload- Combining Security associations- Key management.	7	20 %

Electronic Mail Security

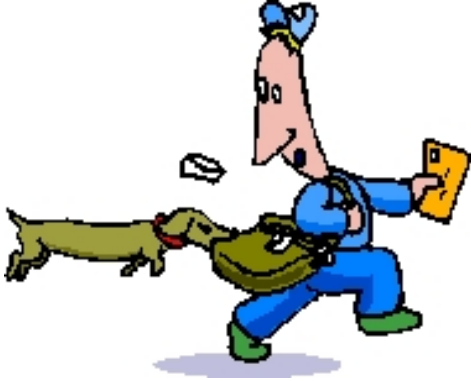
Electronic Mail Security

- Two schemes
  - Pretty Good Privacy (PGP)
  - Secure/Multipurpose Internet Mail Extension (S/MIME)

Why Study E-mail Security?

- After web browsing, e-mail is the most widely used network-reliant application.
- Mail servers, after web servers, are the most often attacked Internet hosts.
- Basic e-mail offers little security, counter to public perception.
- Good technical solutions are available, but not widely used.
  - If we understand why this is so, we might understand something about why security is ‘hard’.

Threats to E-mail



- Loss of confidentiality.
  - *E-mails are sent in clear over open networks.*
  - *E-mails stored on potentially insecure clients and mail servers.*
- Loss of integrity.
  - *No integrity protection on e-mails; anybody be altered in transit or on mail server.*

E-mail security

- What are the Options?
  - Secure the server to client connections (easy thing first)
    - https access to webmail
    - Protection against insecure wireless access
  - Secure the end-to-end email delivery
    - The PGPs of the world
    - Practical in an enterprise intra-network environment

Pretty Good Privacy (PGP)

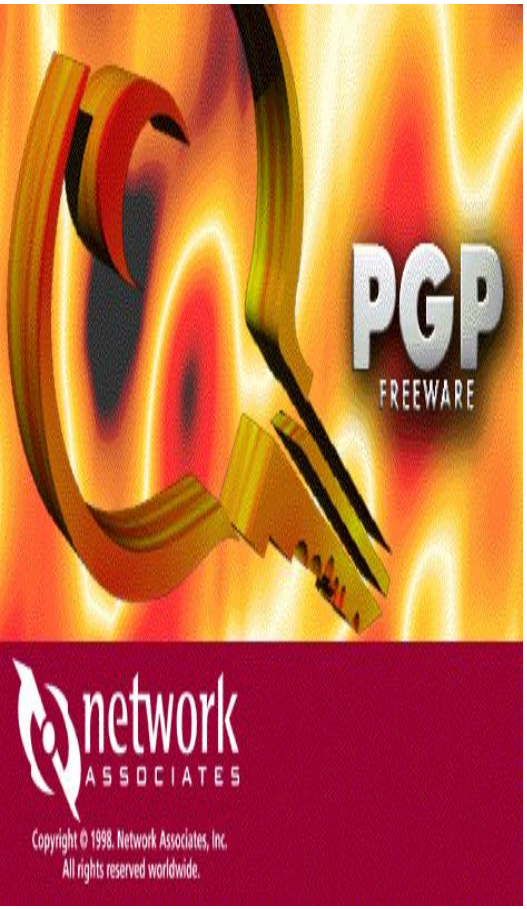
Pretty Good Privacy (PGP)

- Developed by Phil Zimmermann
- Selected best available cryptographic algorithms to use
- Integrated into a single program
- Originally free, now also have commercial versions available

PGP(Pretty Good Privacy)

- PGP is an e-mail security program written by Phil Zimmermann, based on the IDEA algorithm for encryption of plaintext and uses the RSA Public Key algorithm for encryption of the private key.
- PGP incorporates tools for developing a public-key trust model and public-key certificate management.

PGP(Pretty Good Privacy)



- PGP is an open-source freely available software package for e-mail security. It provides authentication; confidentiality; compression; e-mail compatibility; and segmentation and reassembly.

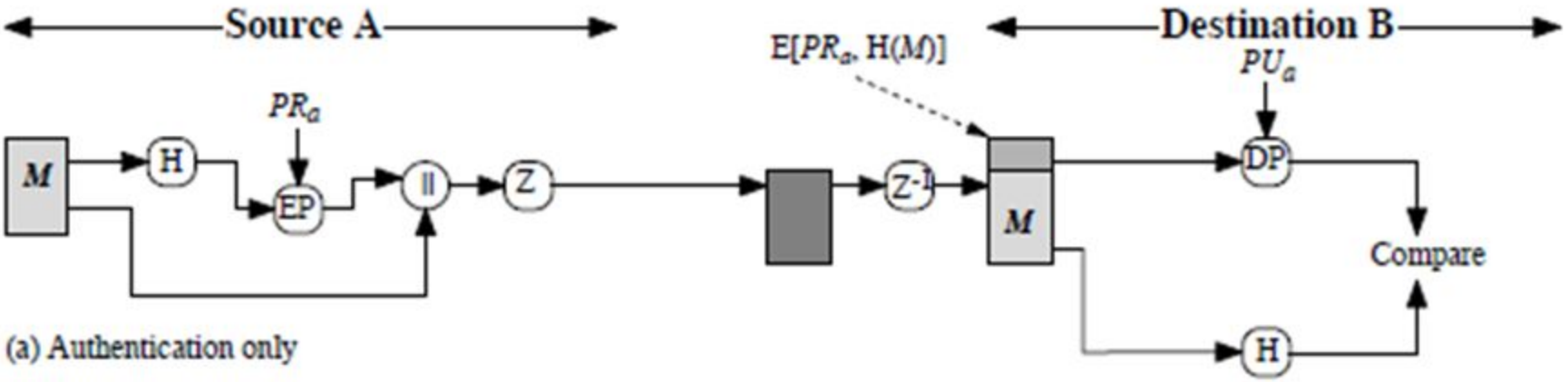
PGP Services

- Authentication
- Confidentiality
- Compression
- e-mail Compatibility
- Segmentation

Authentication

1. Sender creates message
2. Use SHA-1 to generate 160-bit hash of message
3. Signed hash with RSA using sender's private key, and is attached to message
4. Receiver uses RSA with sender's public key to decrypt and recover hash code
5. Receiver verifies received message using hash of it and compares with decrypted hash code

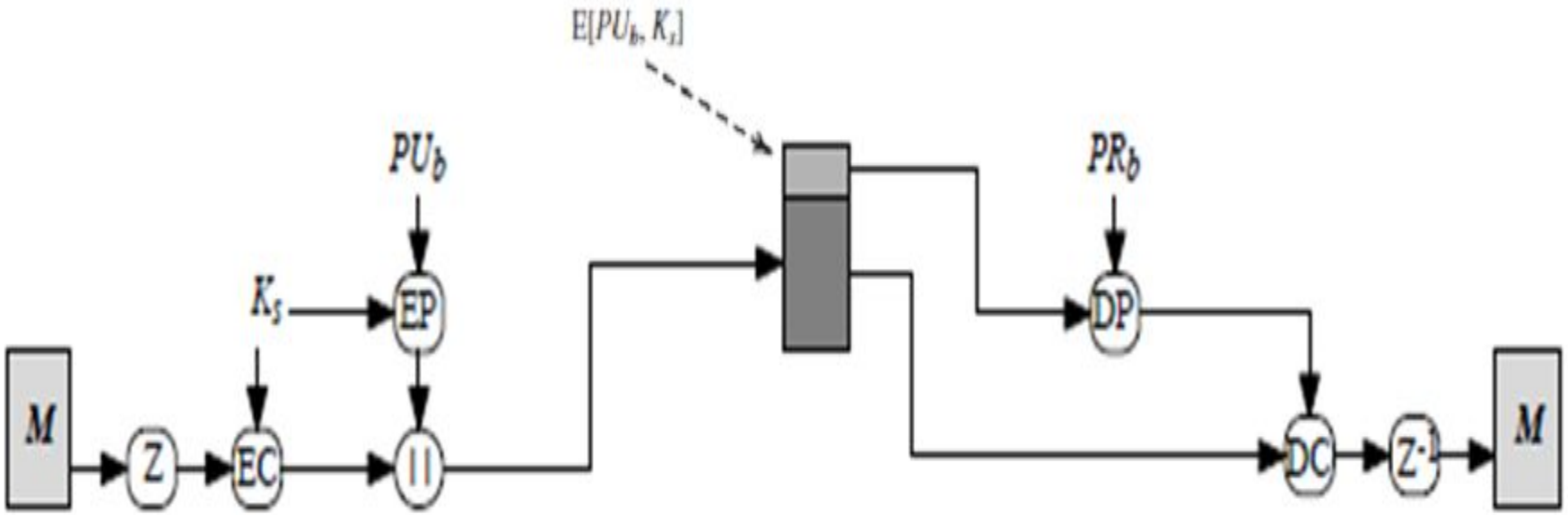
Authentication



Confidentiality

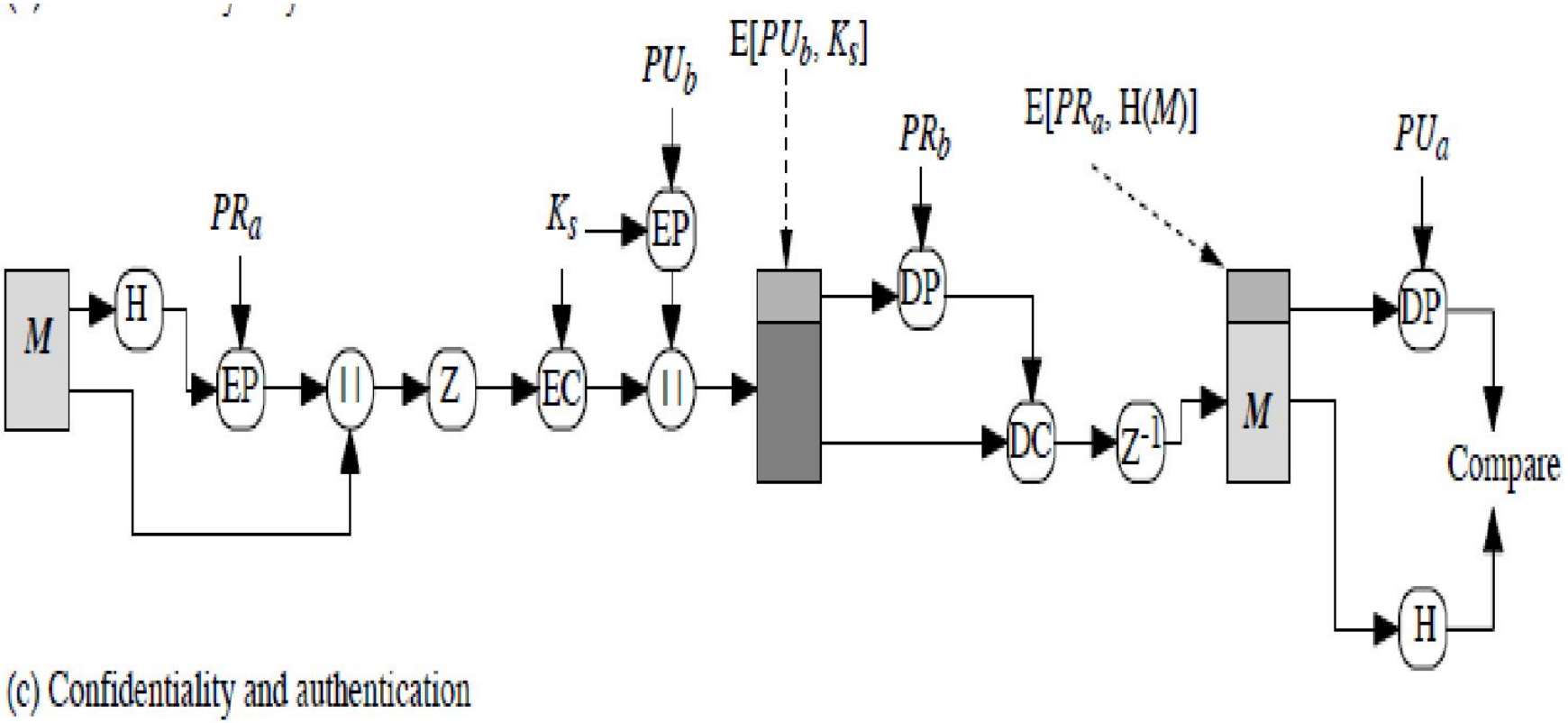
1. Sender generates message and 128-bit random number as session key for it
2. Encrypt message using CAST-128 / IDEA / 3DES with session key
3. Session key encrypted using RSA with recipient's public key, & attached to msg
4. Receiver uses RSA with private key to decrypt and recover session key
5. Session key is used to decrypt message

Confidentiality



(b) Confidentiality only

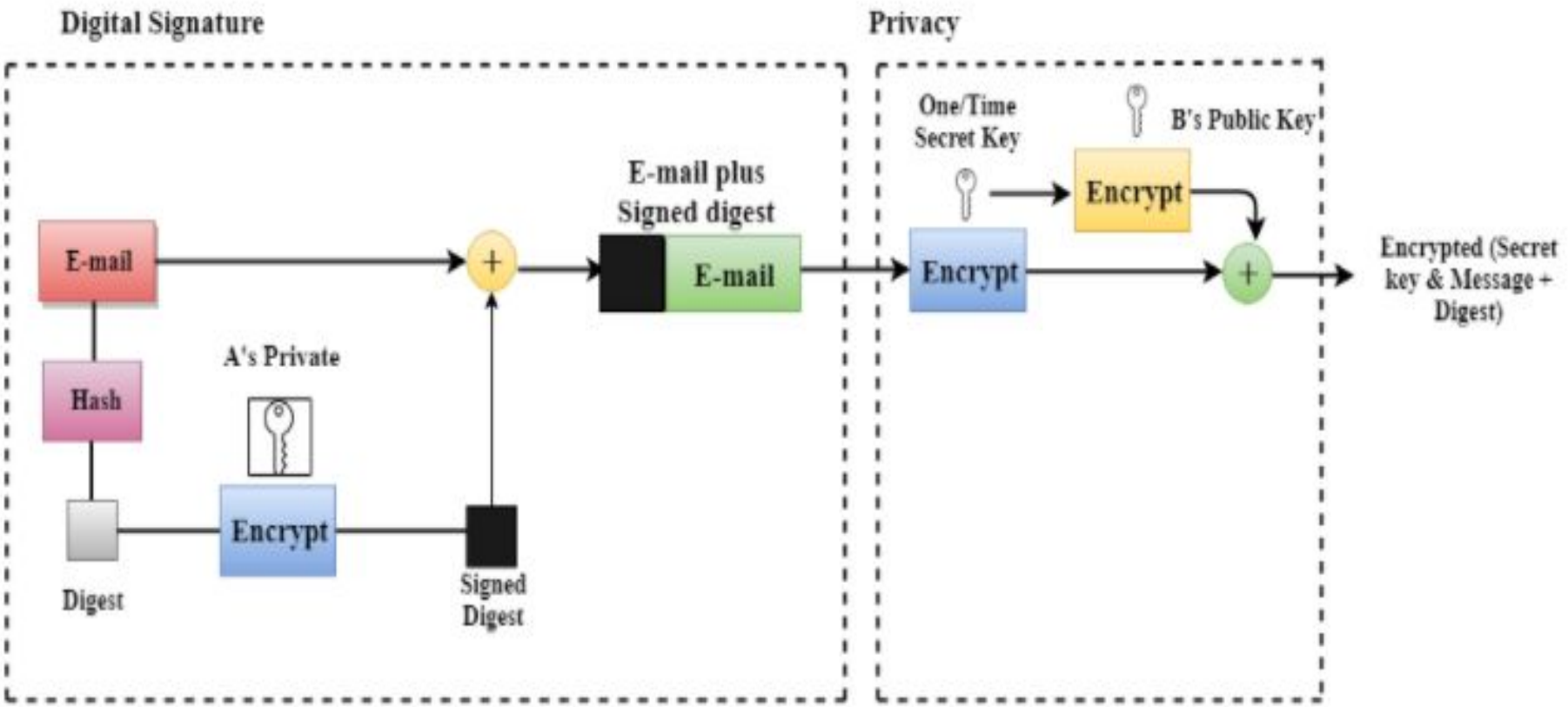
Confidentiality and Authentication



(c) Confidentiality and authentication

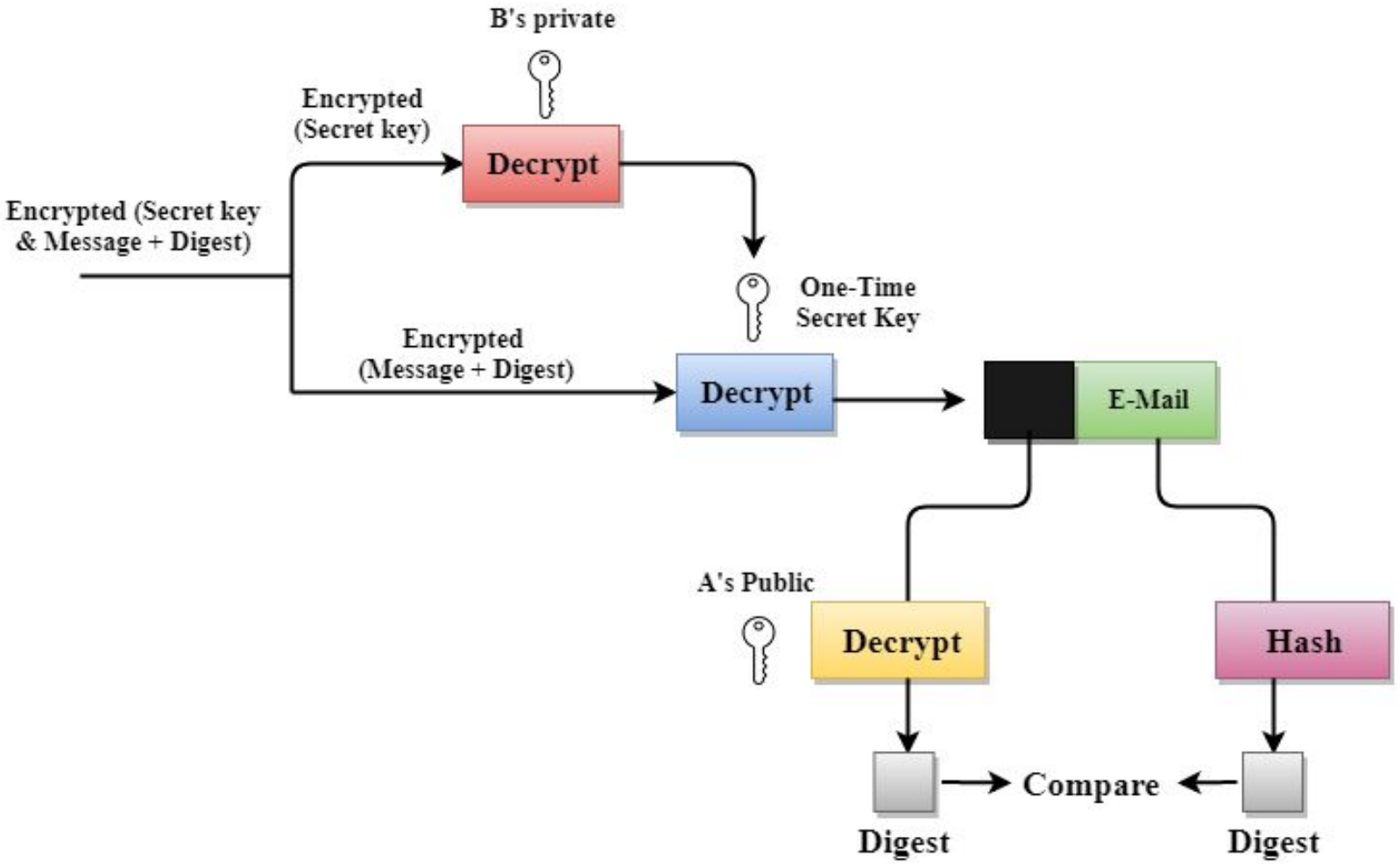
- **Following are the steps taken by PGP to create secure e-mail at the sender site**:
- The e-mail message is hashed by using a hashing function to create a digest.
- The digest is then encrypted to form a signed digest by using the sender's private key, and then signed digest is added to the original email message.
- The original message and signed digest are encrypted by using a one-time secret key created by the sender.
- The secret key is encrypted by using a receiver's public key.
- Both the encrypted secret key and the encrypted combination of message and digest are sent together.

PGP at the Sender site (A)



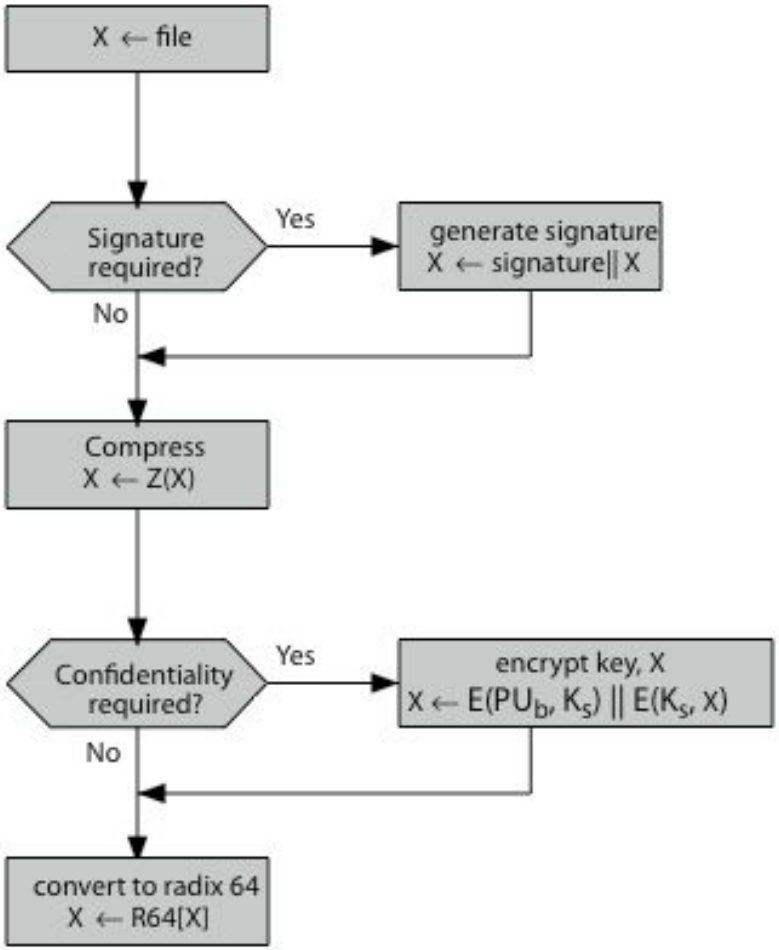
- The receiver receives the combination of encrypted secret key and message digest is received.
- The encrypted secret key is decrypted by using the receiver's private key to get the one-time secret key.
- The secret key is then used to decrypt the combination of message and digest.
- The digest is decrypted by using the sender's public key, and the original message is hashed by using a hash function to create a digest.
- Both the digests are compared if both of them are equal means that all the aspects of security are preserved.

PGP at the Receiver site (B)

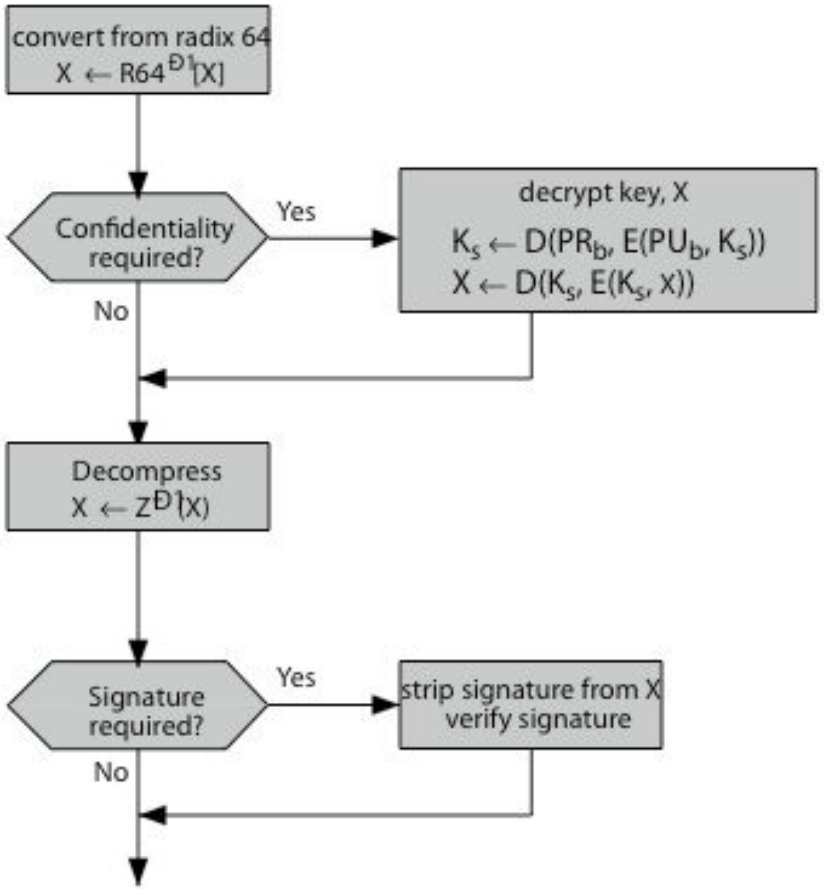


e-mail Compatibility

- When using PGP will have binary data to send (encrypted message etc)
- However email was designed only for text
- Hence PGP must encode raw binary data into printable ASCII characters
- Uses radix-64 algorithm
- PGP also segments messages if too big
  - It is reassembled by receiver before doing any operations

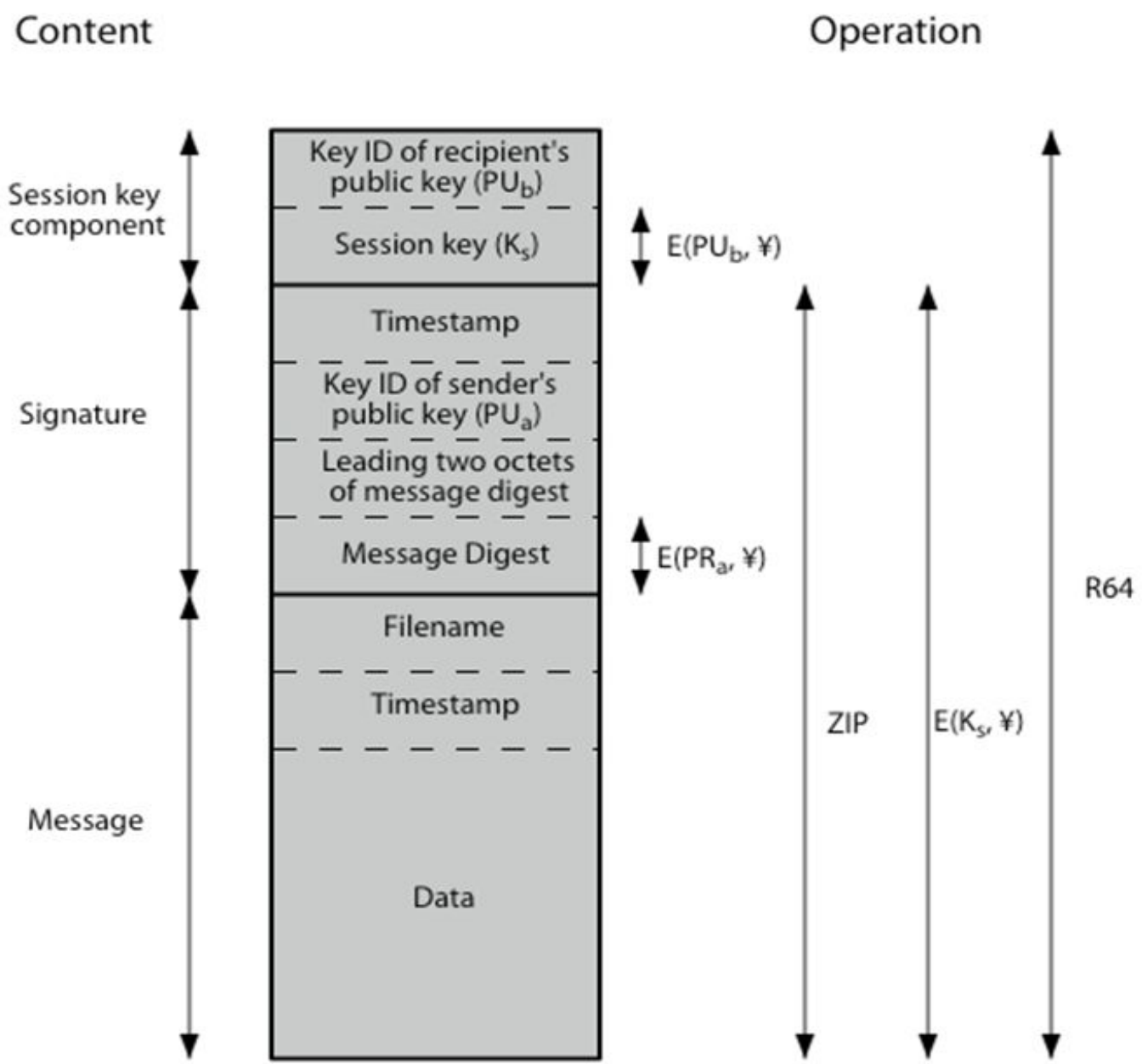


(a) Generic Transmission Diagram (from A)



(b) Generic Reception Diagram (to B)

PGP Operation – Summary



PGP Message Format

Secure/Multipurpose Internet Mail Extension (S/MIME)

Secure/Multipurpose Internet Mail Extension (S/MIME)

- Security enhancement to MIME email
  - Original internet RFC822 email was text only
  - MIME provided support for varying content types and multi-part messages
  - With encoding of binary data to textual form
  - S/MIME added security enhancements

RFC 822

- Defines format for text messages sent using e-mail
- Messages have :
  - Envelop
  - Contents
- RFC 822 applies only to contents
- Contents consists of
  - header field,
  - followed by a blank line and
  - the body

RFC 822: Sample

```
Date: October 8, 2009 2:15:49 PM EDT
From: "William Stallings" <ws@shore.net>
Subject: The Syntax in RFC 5322
To: Smith@Other-host.com
Cc: Jones@Yet-Another-Host.com

Hello. This section begins the actual
message body, which is delimited from the
message heading by a blank line.
```

Limitations of RF822/SMTP

- Can’t transmit exe files or binary objects
- National language characters not supported
- Mails messages limited to a certain size
- Doesn’t support non textual data

MIME

- MIME includes
  1. Five New header fields
  2. A number of content formats
  3. Transfer encoding for protection against alteration

MIME: Header fields

1. MIME - version
2. Content - type
3. Content-Transfer-Encoding
4. Content-ID
5. Content-Description

MIME Content Types

Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
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.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript format.
	octet-stream	General binary data consisting of 8-bit bytes.

MIME Transfer Encoding

- To provide reliable delivery
- Two methods of encoding data:
  1. Quoted –printable
  2. Base64
- Another encoding is x-token
  - Application specific encoding

S/MIME Functions

- Enveloped data
  - Encrypted content and associated keys
- Signed data
  - Encoded message + signed digest
- Clear-signed data
  - Clear text message + encoded signed digest
- Signed & enveloped data
  - Nesting of signed & encrypted entities

S/MIME Cryptographic Algorithms

- Digital signatures: DSS & RSA
- Hash functions: SHA-1 & MD5
- Session key encryption: elgamal & RSA
- Message encryption: AES, triple-DES, RC2/40 and others
- Have process to decide which algorithms to use

S/MIME Message Preparation

- S/MIME secures a MIME entity with a signature, encryption, or both
  - MIME entity is prepared.
  - Add security related data to produce a PKCS object.
  - This is again wrapped in MIME.

S/MIME content-types:

i. Enveloped data
ii. Signed data
iii. Clear-signed data
iv. Registration request
v. Certificate only message

Enveloped data

1. Generate a pseudorandom session key for a particular symmetric encryption algorithm.
2. For each recipient, encrypt the session key with the recipients public RSA key.
3. For each recipient, prepare RecipientInfo block containing recipient's public-key certificate, id of algo used to encrypt the session key, encrypted session key.
4. Encrypt the message content with the session key.

Signed data

1. Select a message digest algorithm
2. Compute message digest.
3. Encrypt the message digest with the signer's private key.
4. Prepare SignerInfo block that contains signer's public-key certificate, an id of the message digest algorithm, id of the algorithm used to encrypt the message digest and the encrypted message digest.

Clear Signing

- Achieved using the multipart content type with a signed subtype
- Message content ‘is clear’
  - Messages are not encoded
  - Signature part is encoded

Registration Request

- An application or user will apply to a certification authority for a public-key certificate

Certificate-Only Messages

- A message containing only certificates or a certificate revocation list (CRL) can be sent in response to a registration request.