Privacy Enhanced Mail (PEM)

Privacy Enhanced Mail (PEM) is an email security standard to provide secure electronic mail communication over the internet. Security of email messages has become extremely important nowadays. In order to deal with the security issues of emails the internet architecture board has adopted it.

The PEM mainly provides the following services –

1. **Confidentiality** –
   Confidentiality refers to the act of preventing unauthorized access to the information hence protecting it. The confidentiality is obtained in PEM by encrypting the messages by using various standard algorithms such as Data Encryption Standard DES in cipher block chaining mode is being currently used by PEM.
2. **Integrity** –
   Data integrity refers to the consistency of data through out its life cycle. This is obtained by using a unique concept called as message digest where message digest is a hash function which converts the message into an image called digest on taking the message as input. PEM uses RSA encryption, MD2 and MD5 hash functions to generate the digests. An octet value is generated from the hash functions which is then encrypted which is then run against the message digest by the receiver assured of the integrity of the message that is transmitted.

Working of PEM :
The PEM works basically in 4 main steps.

- **Canonical Conversion**

  This step involves the conversion of the message into a standard format that is independent of the computer architecture and the operation system of the sender and the receiver. If the sender and receiver has different computer architecture or operating system. It may lead to generation of different message digest due to difference in their interpretation because of syntactical difference from one operating system to an other.

- **Digital Signature –**
  In this step, the digital signature is generated by encrypting the message digest of an email message with the sender's private key.

Encryption -

- The encrypted message is generated by encrypting the original message and digital signature together along with the symmetric key as shown in the figure below.This step is very crucial in order to obtain the confidentiality.

- **Base-64 Encoding**

  This is the last step where the binary output is transformed into character output. The binary output which is 24 bits is divided into 4 equal sets and mapped with the 8 bit character output generating a decimal code. Now PEM uses a separate map table and each number from the code generated is mapped with its corresponding value from the mapping table and binary equivalent corresponding to the 8 bit ASCII of the character is written.

Pretty Good Privacy (PGP)

With the explosively growing reliance on electronic mail for every conceivable purpose, there grows a demand for authentication and confidentiality services. Two schemesstand out as approaches that enjoy widespread use: Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extension (S/MIME). The latter is a security en- hancement to the MIME Internet e-mail format standard, based on technology from RSA Data Security. Although both PGP and S/MIME are on an IETF standards track, it appears likely that S/MIME will emerge as the industry standard for commercial andorganisational use, while PGP will remain the choice for personal e-mail security for many users. In this course we will only be looking at PGP. S/MIME is discussed in detail in the recommended text.

PGP consists of the following five services:

- Authentication
- Confidentiality
- Compression
- E-mail compatibility
- Segmentation

1. Authentication

Figure 12.2a illustrates the digital signature service provided by PGP. The figure is similar to ones we have looked at earlier. The hash function used is SHA-1 which creates a 160 bit message digest. EP (DP) represents public encryption (decryption) and the algorithm used can be RSA or DSS (recall that the DSS can only be used for the digital signature function and unlike RSA cannot be used for encryption or key exchange). The message may be compressed using and algorithm called **ZIP**. This is represented by "Z" in the figure.

The combination of SHA-1 and RSA provides an effective digital signature scheme. Due to the strength of RSA the recipient is assured that only the possessor of the

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed, for storage or transmission, using ZIP.
Email compatibility	Radix 64 conversion	To provide transparency for email applications, an encrypted message may be converted to an ASCII string using radix 64 conversion.
Segmentation	—	To accommodate maximum message size limitations, PGP performs segmentation and reassembly.

Figure 12.1: Summary of PGP services.

matching private key can generate the signature. Because of the strength of SHA-1 the recipient is assured that no one else could generate a new message that matches the hash code and hence, the signature of the original message.

2. Confidentiality

Another basic service provided by PGP is confidentiality which is provided by encrypting messages to be transmitted or to be stored locally as files. In both cases, the user has a choice of CAST-128, IDEA or 3DES in 64 bit cipher feedback (CFB) mode. The symmetric key is used only once and is created as a random number with the required number of bits. It is transmitted along with the message and is encrypted using the recipients public key. Figure 12.2c illustrates the sequence:

1. The sender generates a message and a random number to be used as a sessionkey for this message only.
2. The message is encrypted using CAST-128, IDEA or 3DES with the session key.
3. The session key is encrypted with RSA (or another algorithm known as ElGamal) using the recipients public key and is prepended to the message.

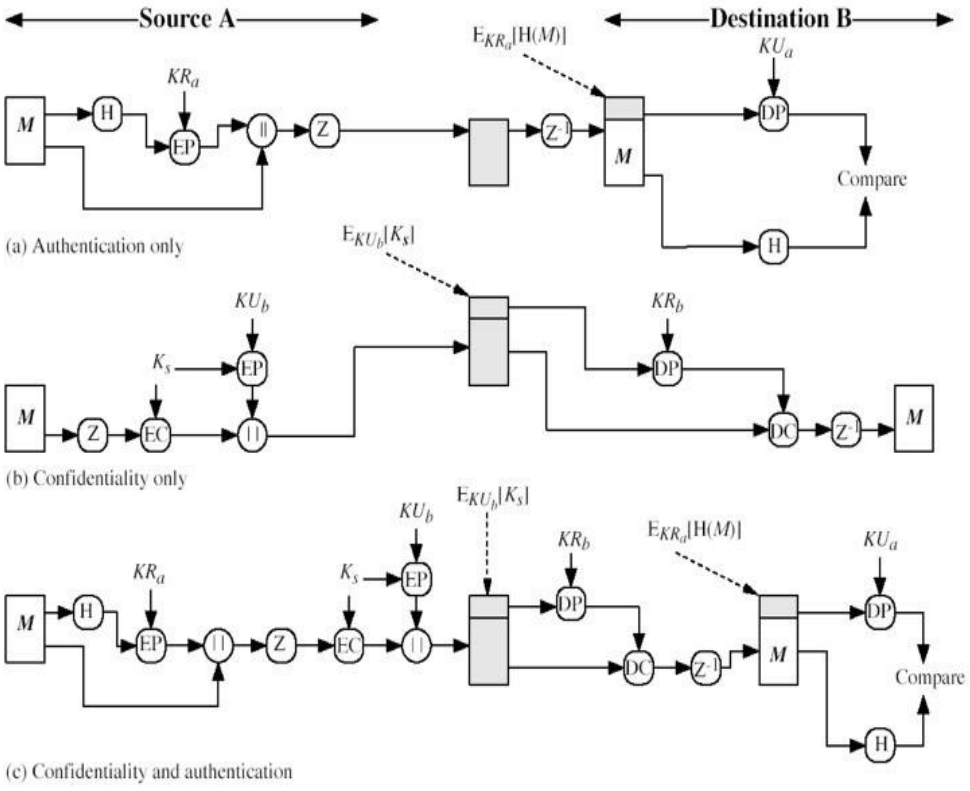


Figure 12.2: PGP cryptographic functions

4. The receiver uses RSA with its private key to decrypt and recover the sessionkey.

5. The session key is used to decrypt the message.

As mentioned before, public key encryption is a lot more computationally intensive than symmetric encryption. For this reason both forms are used as public key encryption solves the key distribution problem. However as will be noticed, the message itself (which is the largest part of the transmission) is encrypted using symmetric key cryptography whereas only the key is encrypted using the public key algorithm.

3. Confidentiality and Authentication

As figure 12.2c illustrates, both services may be used for the same message. First, a signature is generated for the plaintext message and prepended to the message. Then the plaintext message plus signature is encrypted using CAST-128 (or IDEA or 3DES), and the session key is encrypted using RSA (or ElGamal). This sequence is preferable
to the opposite: encrypting the message and then generating a signature of the encrypted message. It is generally more convenient to store a signature with a plaintext

version of a message. Furthermore, for purposes of third party verification, if the signature is performed first, a third party need not be concerned with the symmetric key when verifying the signature.

4. Compression

As a default, PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space both for e-mail transmission and for file storage.

The placement of the compression algorithm, indicated by Z for compression and Z^{-1} for decompression in figure 12.2 is critical:

6. The signature is generated before compression for two reasons:
   (a) It is preferable to sign an uncompressed message so it is free of the need for a compression algorithm for later verification.
   (b) Different version of PGP produce different compressed forms. Applying the hash function and signature after compression would constrain all PGP implementation to the same version of the compression algorithm.
7. Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the originalplaintext, cryptanalysis is more difficult.

The compression algorithm used is called **ZIP** which is described in the recommended text.

5. E-mail compatibility

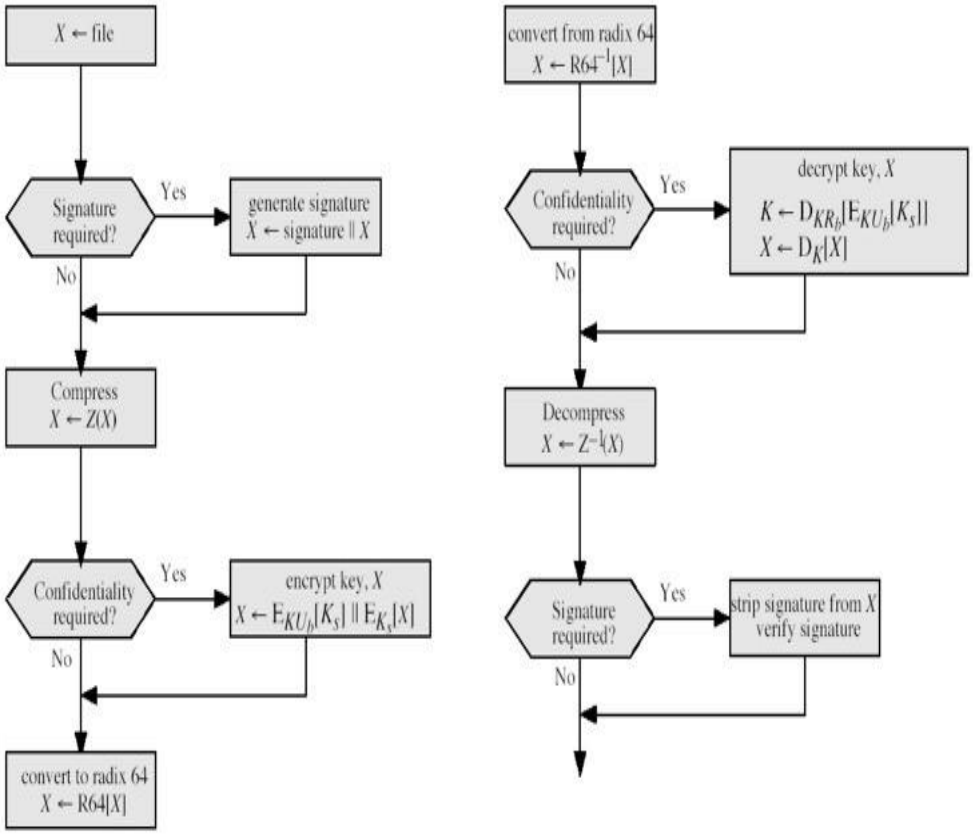
Many electronic mail systems only permit the use of blocks consisting of ASCII text. When PGP is used, at least part of the block to be transmitted is encrypted. This basically produces a sequence of arbitrary binary words which some mail systems won't accept. To accommodate this restriction PGP uses and algorithm known as **radix64** which maps 6 bits of a binary data into and 8 bit ASCII character. Unfortunately this expands the message by 33% however, with the compression algorithm the overall compression will be about one third (in general).

6. Segmentation

E-mail facilities are often restricted to a maximum message length. For example, many of the facilities accessible throughout the Internet impose a maximum length of 50,000 octets. Any message longer than that must be broken up into smaller segments, each of which is mailed separately.

To accommodate this restriction, PGP automatically subdivides a message that is too large into segments that are small enough to sent via e-mail. The segmentation is done after all the other processing, including the radix-64 conversion. Thus the session key

component and signature component appear only once, at the beginning of the first segment. At the receiving end, PGP must strip off all e-mail headers and reassemble the entire original block before performing the steps illustrated in figure 12.3



(a) Generic Transmission Diagram (from A)

(b) Generic Reception Diagram (to B)

Figure 12.3: Transmission and Reception of PGP messages.