Experiment No:\_\_

**Aim:** Case study of Pretty Good Privacy (PGP)

**Theory:**

PGP is a remarkable phenomenon. Largely the effort of a single person, Phil Zimmermann, PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications. In essence what Zimmermann has done is the following:

* Selected the best cryptographic mechanisms (algorithms) as building blocks.
* Integrated these algorithms into a general purpose application that is independent of operating system and processor and that is based on a small set of easy to use commands.
* Made the package and its source code freely available via the Internet, bulletin boards, and commercial networks such as America On Line (AOL).
* Entered into an agreement with a company (Viacrypt, now Network Associates) to provide a fully compatible low cost commercial version of PGP.

From its beginnings about 15 years ago, PGP has grown explosively and is now very widely used.

A number of reasons are cited for such growth:

* It is available free worldwide in versions that run on many different platforms, Windows, UNIX, Mac etc. In addition the commercial version satisfies those who want vendor support.
* It is based on algorithms that have survived extensive public review and are considered secure. Specifically, the package includes RSA, DSS and Diffie-Hellman for public-key encryption; CAST-128, IDEA, and 3DES for symmetric encryption; and SHA-1 for hash coding.
* It has a wide range of applicability, from corporations that wish to select and enforce a standardised scheme for encrypting files and messages to individuals who wish to communicate securely with others worldwide over the Internet.
* It was not developed by, nor is it controlled by, any government or standards organisation. For those with an instinctive distrust of “the establishment”, this makes PGP attractive. In the last few years commercial versions have become available.
* PGP is now on an Internet standards track (RFC 3156). Nevertheless, PGP still has an aura of an anti-establishment endeavor.

**Operational Description:**

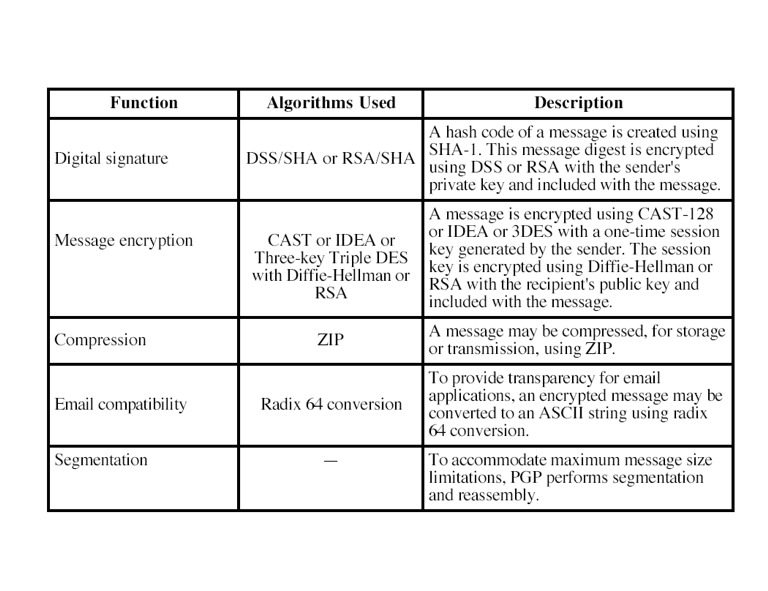
PGP consists of the following five services:

1. Authentication
2. Confidentiality
3. Compression
4. E-mail compatibility
5. Segmentation

**Cryptographic Keys:**

PGP makes use of four types of keys:

1. One-time session symmetric keys
2. Public keys
3. Private keys
4. Passphrase based symmetric keys



**Authentication:**

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 an algorithm called ZIP.

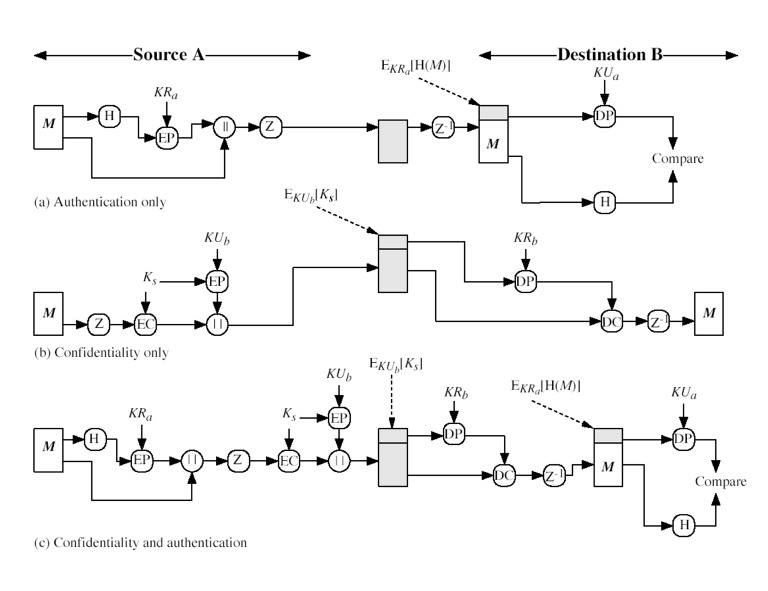
The combination of SHA-1 and RSA provides an effective digital signature scheme.Due to the strength of RSA the recipient is assured that only the possessor of the matching private key can generate the signature. Because of the strength of SHA-1 the recipient is assured that no one else could generate a new message that matches the hash code and hence, the signature of the original message.

**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 recipient's public key.

* The sender generates a message and a random number to be used as a session key for this message only.
* The message is encrypted using CAST-128, IDEA or 3DES with the session key.
* The session key is encrypted with RSA (or another algorithm known as ElGamal) using the recipient's public key and is prepended to the message.
* The receiver uses RSA with its private key to decrypt and recover the session key.
* 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.

**Confidentiality and Authentication:**

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.

**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 is critical:

* The signature is generated before compression for two reasons:
  + It is preferable to sign an uncompressed message so it is free of the need for a compression algorithm for later verification.
  + 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.
* Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult.

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

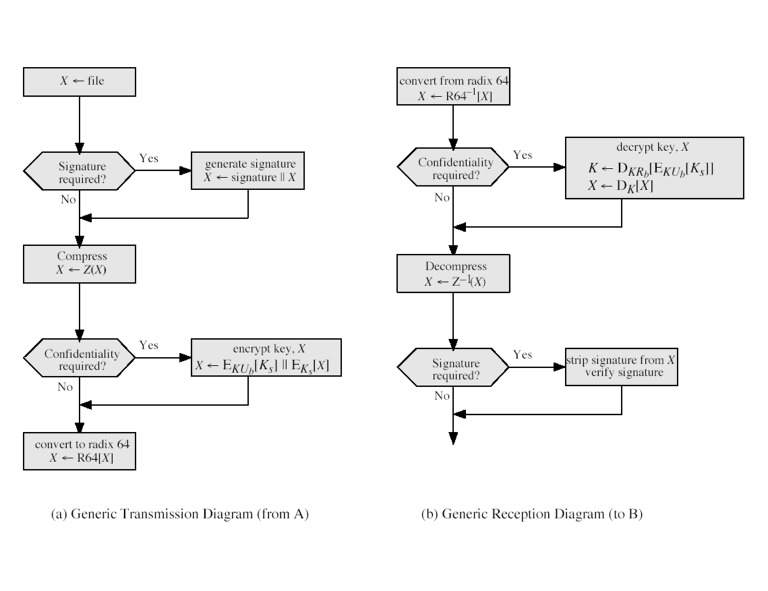
**Email compatibility:**

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 an algorithm known as radix64 which maps 6 bits of a binary data into an 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).

**Segmentation:**

Email 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 email headers and reassemble the entire original block before performing the steps illustrated in figure.



**Conclusion:**

To summarize the state of secure e-mail software, we can say that software exists now to establish trust between two individuals or within a small group so they can exchange text-based e-mail. Software is available for PGP in a similar manner, although it is not nearly as widespread and is mostly available commercially.