Email Security

Email Security

- email is one of the most widely used and regarded network services
- currently message contents are not secure
  - may be inspected either in transit
  - or can be disclosed on destination system

Email Security Enhancements

Confidentiality

- protection from disclosure

Authentication

- of sender of message

Message integrity

- protection from modification

Non-repudiation of origin

- protection from denial by sender

Pretty Good Privacy (PGP)

- widely used de facto secure email
- developed by Phil Zimmermann
- selected best available crypto algs to use
- integrated into a single program
- on Unix, PC, Macintosh and other systems
- originally free, now also have commercial versions available

The actual operation of PGP consists of five services:

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

Table 12.1 Summary of PGP Services

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.

Authentication

The steps for authentication are as follows

- The sender creates a message
- SHA-1 is used to generate 160-bit hash code
- Hash code is encrypted with RSA using senders private key
- Receiver uses RSA to decrypt the hash code
- Receiver generates a new hash code and compares with the decrypted one

Confidentiality

The steps to obtain confidentiality are as follows

- The sender generates a message and a random 128-bit number called the session key
- The message is encrypted with CAST-128
- The session key is encrypted with recipients' public key using RSA
- The recipient uses RSA with its private key to decrypt the session key
- The session key decrypts the message

Confidentiality and Authentication

To have both confidentiality and authentication

- The sender first signs the message using it's own private key
- Then encrypts the message with the session key
- Then encrypts the session key with the recipient's private key

Compression

PGP compresses files using a ZIP algorithm

- The signature is generated before compression
  - To store the uncompressed message with the signature
  - Would interfere with compression because of multiple compression algorithms exist.
- Message encryption is after compression
  - To strengthen cryptographic security, as it reduces redundancy

Compatibility

E-mail sends only ASCII characters

- Because of this PGP converts message to ASCII
  - Converts three octets into four ASCII characters
  - Expands message by 33%
  - After compression, there is a net reduction by a third

Segmentation and Reassembly

Some mail providers impose a maximum length of 50,000 octets

- PGP will automatically subdivide any message too large into small enough segments to send via e-mail
  - This is done after all other processing

Cryptographic Keys

PGP uses four types of keys

- Session keys
- Public keys
- Private Keys
- Passphrase keys

Cryptographic Keys

Three requirements for the keys

- Needs a mean of generating unpredictable session keys
- Would like a way to allow each user to have multiple public/private key pairs
- Maintain a file of the public/private key pairs

Session Key Generation

Random 128-bit numbers are generated using CAST-128

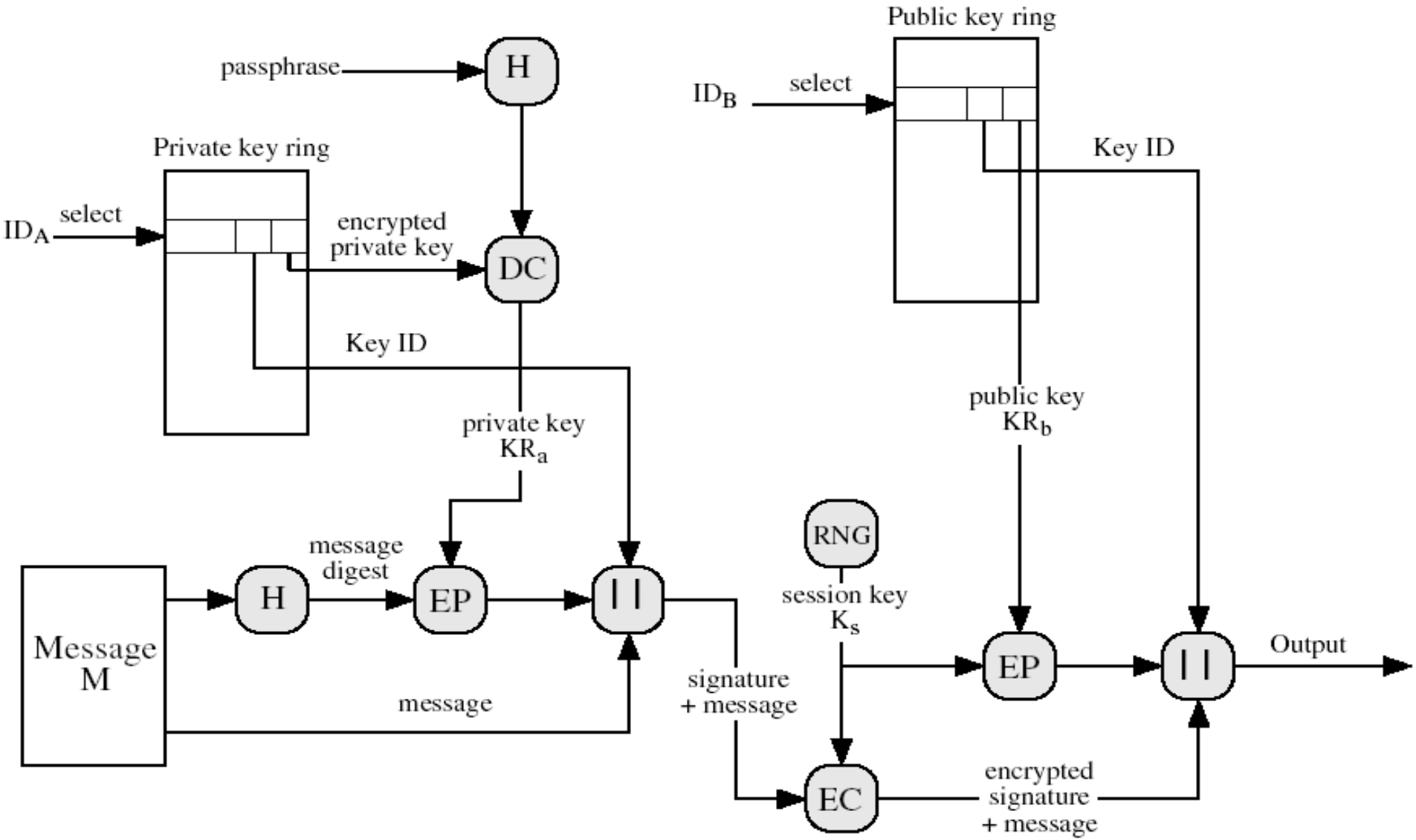
Input to the number generator takes in is a 128-bit key and two 64-bit blocks of plaintext.

Key identifiers

With multiple private/public key pairs, there needs to be a way for the receiver to know which to use

- How this is done is through the combination of a 64 bit key ID, which is unique to a user ID.
  - With this key ID, the receiver can retrieve the correct public key of the sender to decrypt the message.
  - A list of these key ID's are placed in what is called a key ring.

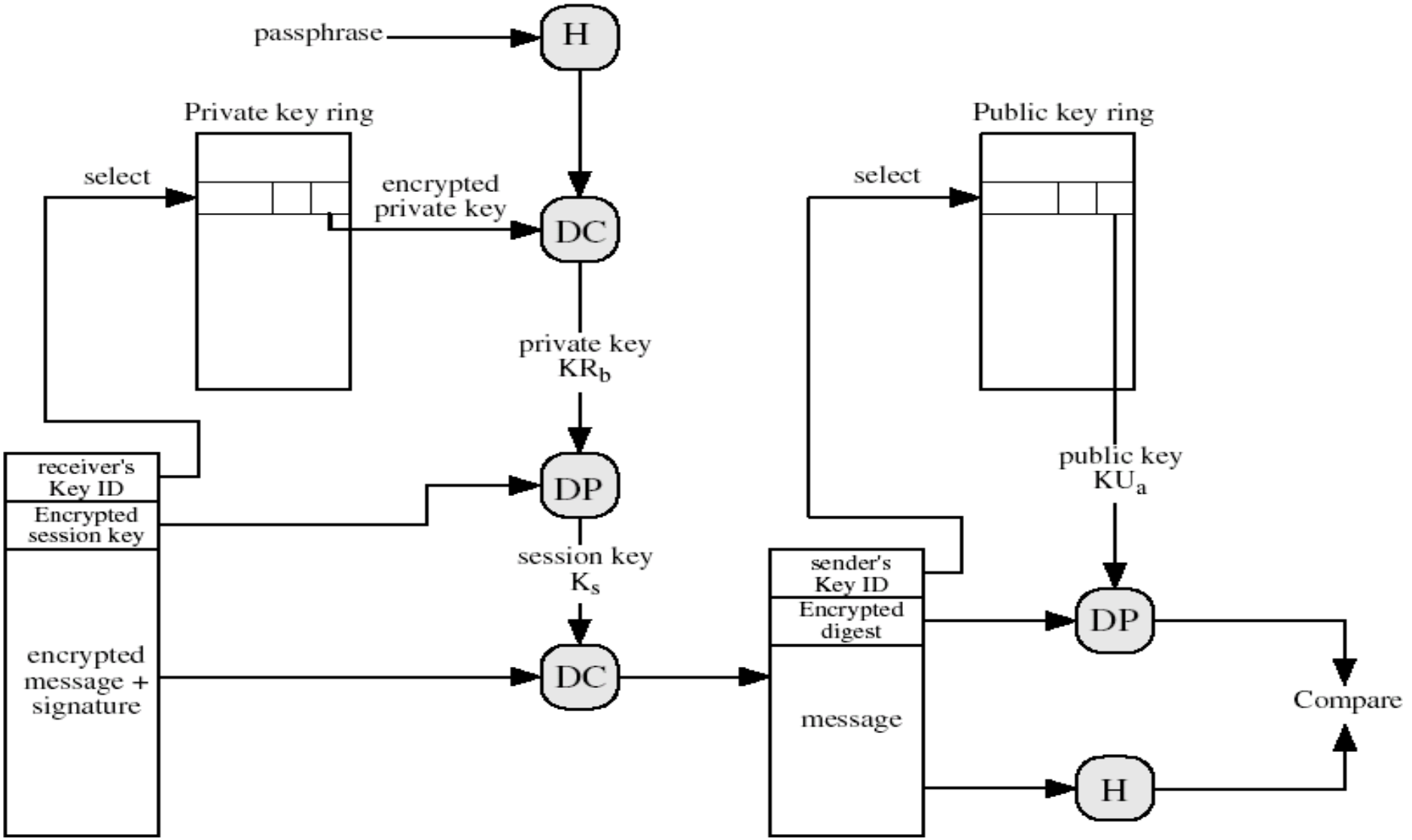
PGP Message Generation



PGP Message Generation

- The sending PGP entity performs the following steps:
  - Signs the message:
    - PGP gets sender's private key from key ring using its user id as an index.
    - PGP prompts user for passphrase to decrypt private key.
    - PGP constructs the signature component of the message.
  - Encrypts the message:
    - PGP generates a session key and encrypts the message.
    - PGP retrieves the receiver public key from the key ring using its user id as an index.
    - PGP constructs session component of message

PGP Message Reception



PGP Message Reception

- The receiving PGP entity performs the following steps:
  - Decrypting the message:
    - PGP get private key from private-key ring using Key ID field in session key component of message as an index.
    - PGP prompts user for passphrase to decrypt private key.
    - PGP recovers the session key and decrypts the message.
  - Authenticating the message:
    - PGP retrieves the sender's public key from the public-key ring using the Key ID field in the signature key component as index.
    - PGP recovers the transmitted message digest.
    - PGP computes the message for the received message and compares it to the transmitted version for authentication.