



Electronic Mail 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 by suitably privileged users 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

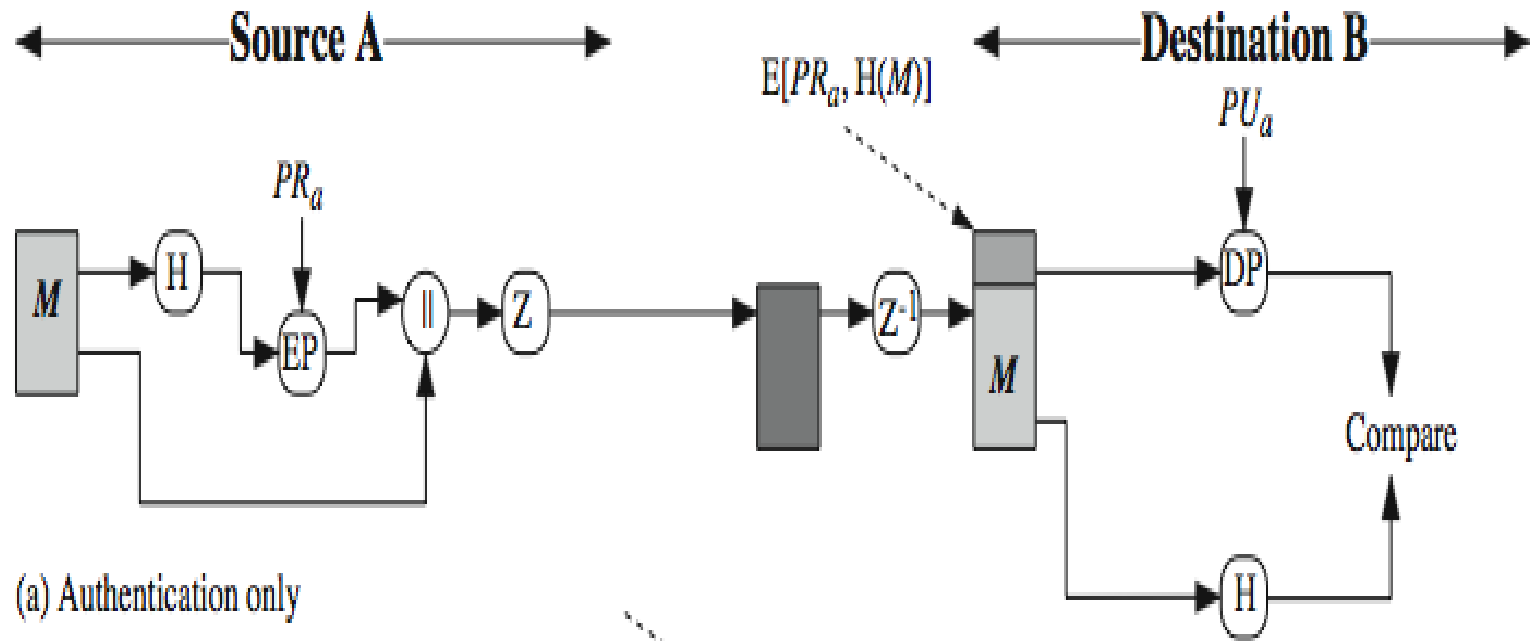
Pretty Good Privacy (PGP)

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

PGP Operation – Authentication

1. sender creates a message
2. SHA-1 used to generate 160-bit hash code of message
3. hash code is encrypted with RSA using the sender's private key, and result is attached to message
4. receiver uses RSA or DSS with sender's public key to decrypt and recover hash code
5. receiver generates new hash code for message and compares with decrypted hash code, if match, message is accepted as authentic

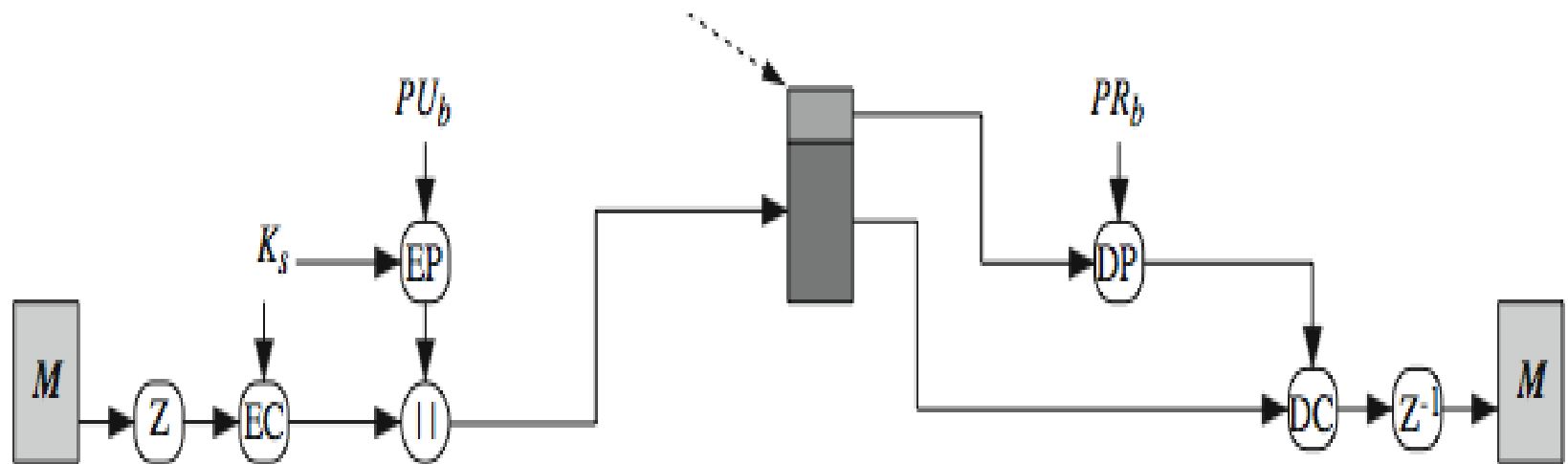
PGP Operation – Authentication



PGP Operation – Confidentiality

1. sender generates message and random 128-bit number to be used as session key for this message only
2. message is encrypted, using CAST-128 / IDEA/3DES with session key
3. session key is encrypted using RSA with recipient's public key, then attached to message
4. receiver uses RSA with its private key to decrypt and recover session key
5. session key is used to decrypt message

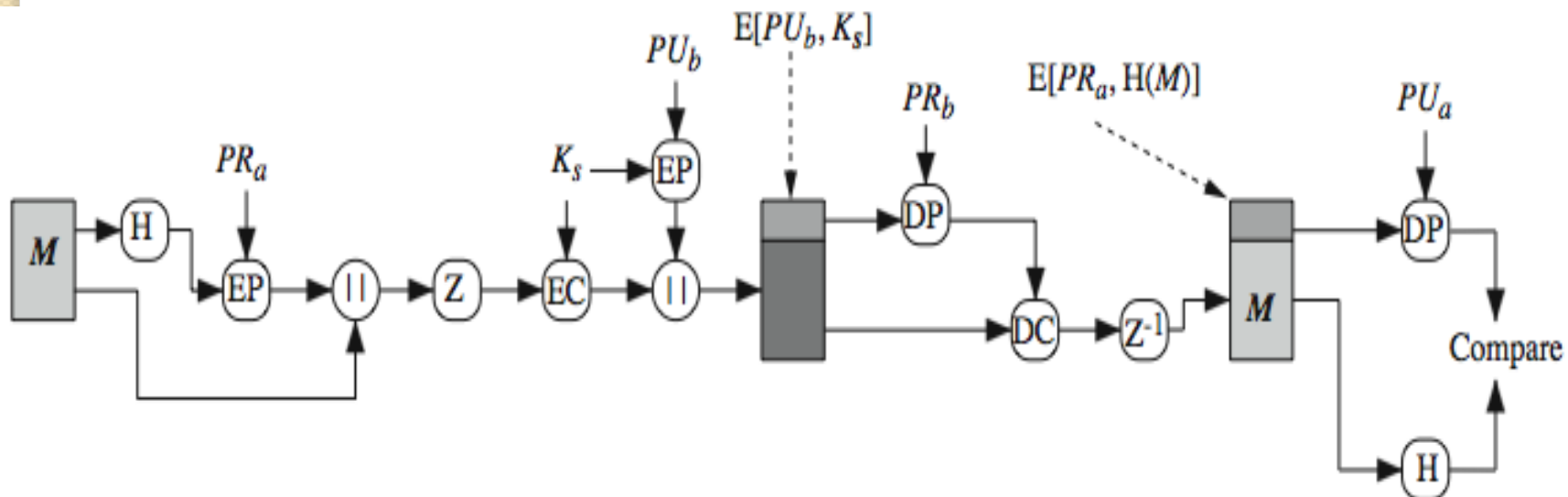
PGP Operation – Confidentiality



(b) Confidentiality only

PGP Operation – Confidentiality & Authentication

- uses both services on same message
 - create signature & attach to message
 - encrypt both message & signature
 - attach RSA encrypted session key



(c) Confidentiality and authentication

PGP Operation – Compression

- by default PGP compresses message after signing but before encrypting
 - so can store uncompressed message & signature for later verification
 - & because compression is non deterministic
- uses ZIP compression algorithm

PGP Operation – Email 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
 - maps 3 bytes to 4 printable chars
 - also appends a CRC
- PGP also segments messages if too big

PGP Services

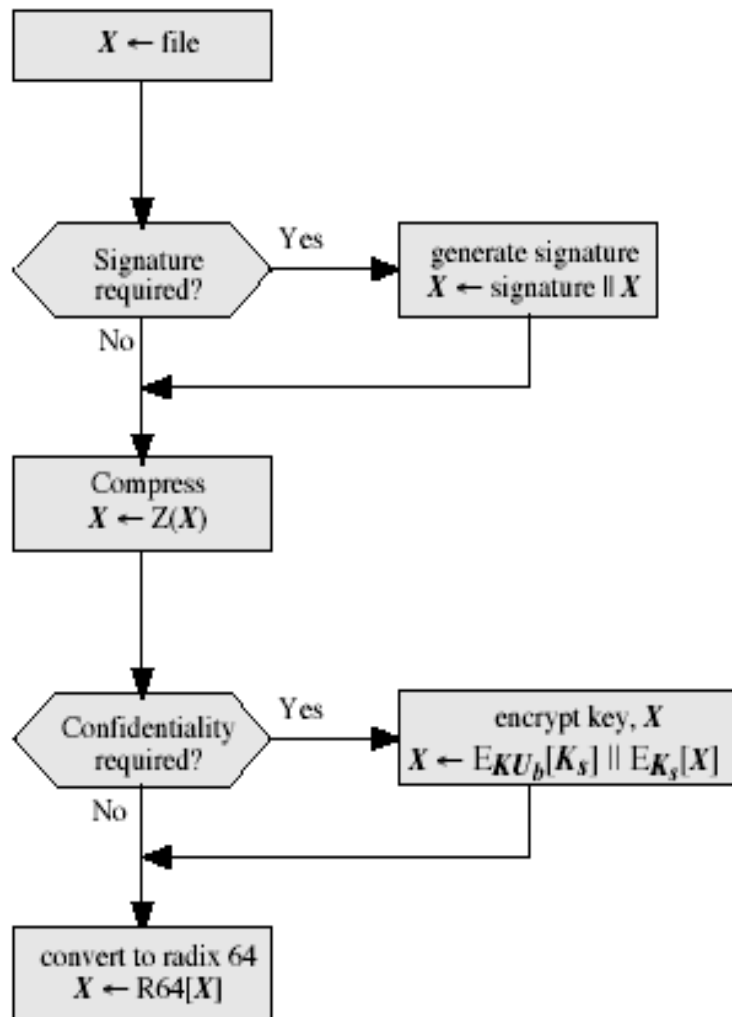
| | | |
|--------------------|---|--|
| Digital signature | DSS/SHA or RSA/SHA | A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message. |
| Message encryption | CAST or IDEA or Three-key Triple DES 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. |

PGP

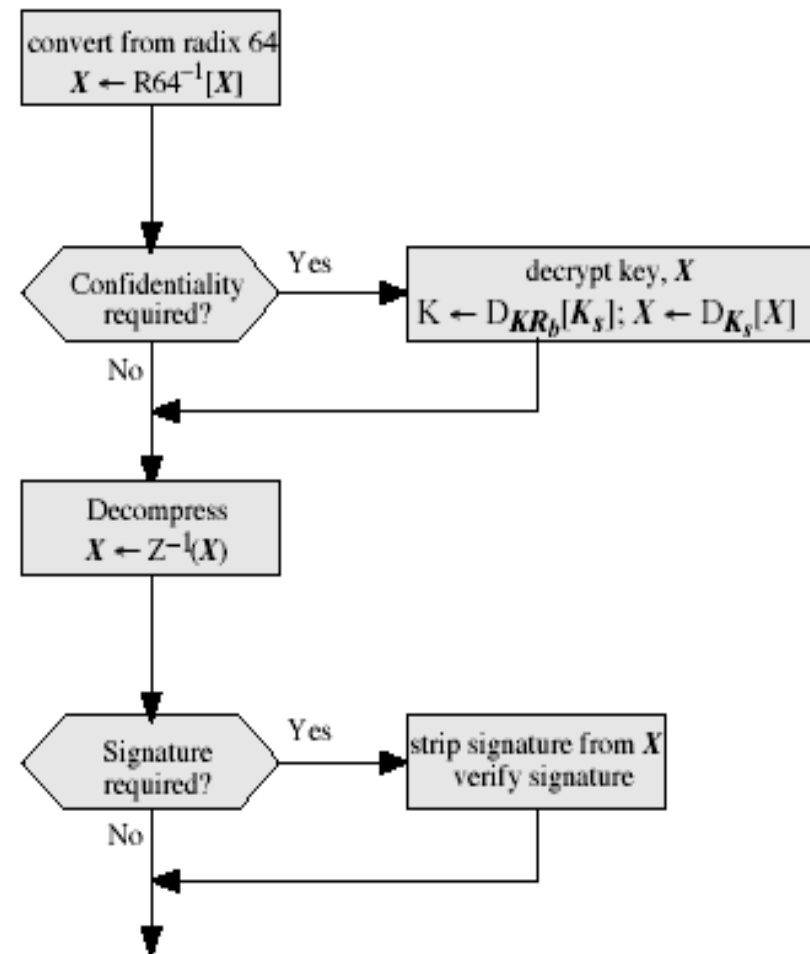
(Pretty Good Privacy)

| | | |
|---------------------|---------------------|---|
| 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. |

PGP Operation – Summary



(a) Generic Transmission Diagram (from A)



(b) Generic Reception Diagram (to B)

PGP Session Keys

- need a session key for each message
 - of varying sizes: 56-bit DES, 128-bit CAST or IDEA, 168-bit Triple-DES
- generated using ANSI X12.17 mode
- uses random inputs taken from previous uses and from keystroke timing of user

PGP Public & Private Keys

- since many public/private keys may be in use, need to identify which is actually used to encrypt session key in a message
 - could send full public-key with every message
 - but this is inefficient
- rather use a key identifier based on key
 - is least significant 64-bits of the key
 - will very likely be unique
- also use key ID in signatures

PGP Key Rings

- each PGP user has a pair of keyrings:
 - public-key ring contains all the public-keys of other PGP users known to this user, indexed by key ID
 - private-key ring contains the public/private key pair(s) for this user, indexed by key ID & encrypted keyed from a hashed passphrase

PGP Key Management

- rather than relying on certificate authorities
- in PGP every user is own CA
 - can sign keys for users they know directly
- forms a “web of trust”
 - trust keys have signed
 - can trust keys others have signed if have a chain of signatures to them
- key ring includes trust indicators
- users can also revoke their keys

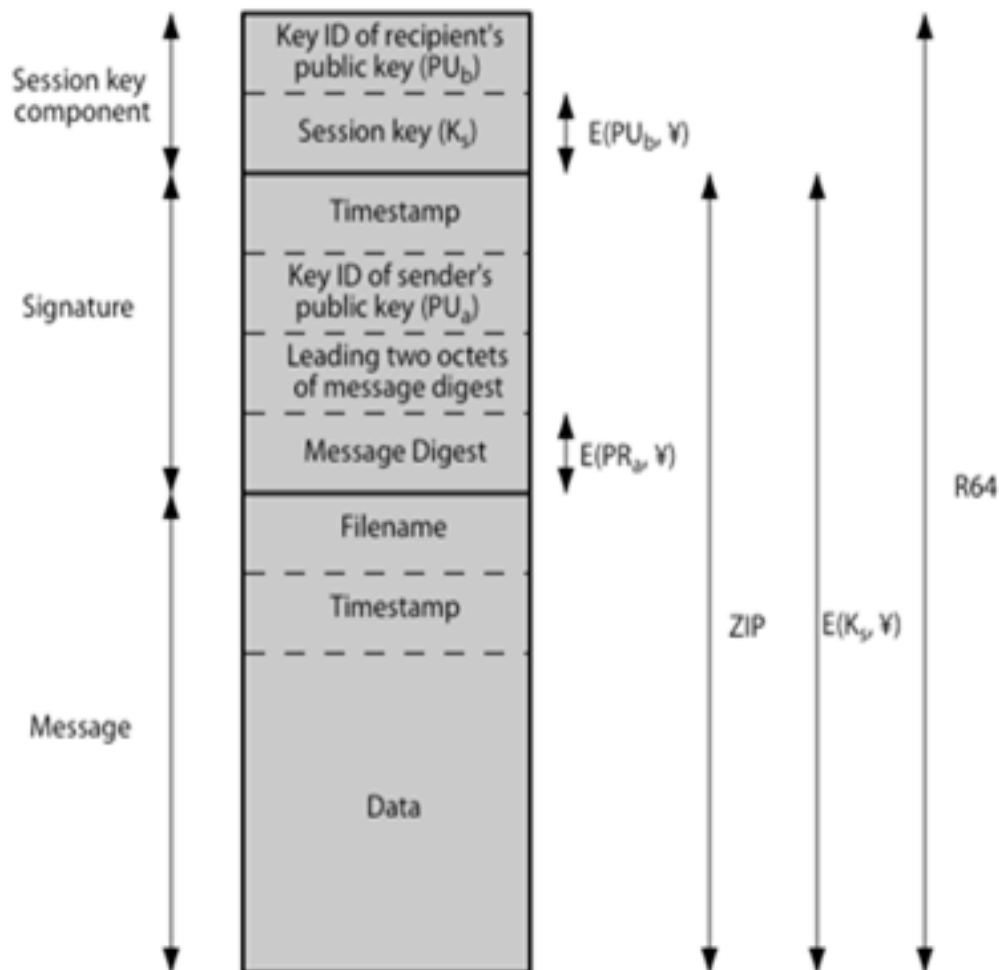
PGP message consist of:

- Message component – the actual data to be transmitted + a filename + a timestamp;
- Signature component – timestamp + hash of message and timestamp + first part of message (so user can check that they are decrypting correctly) + Key ID of sender's public key
- Session Key component – session key + key ID of recipient's public key

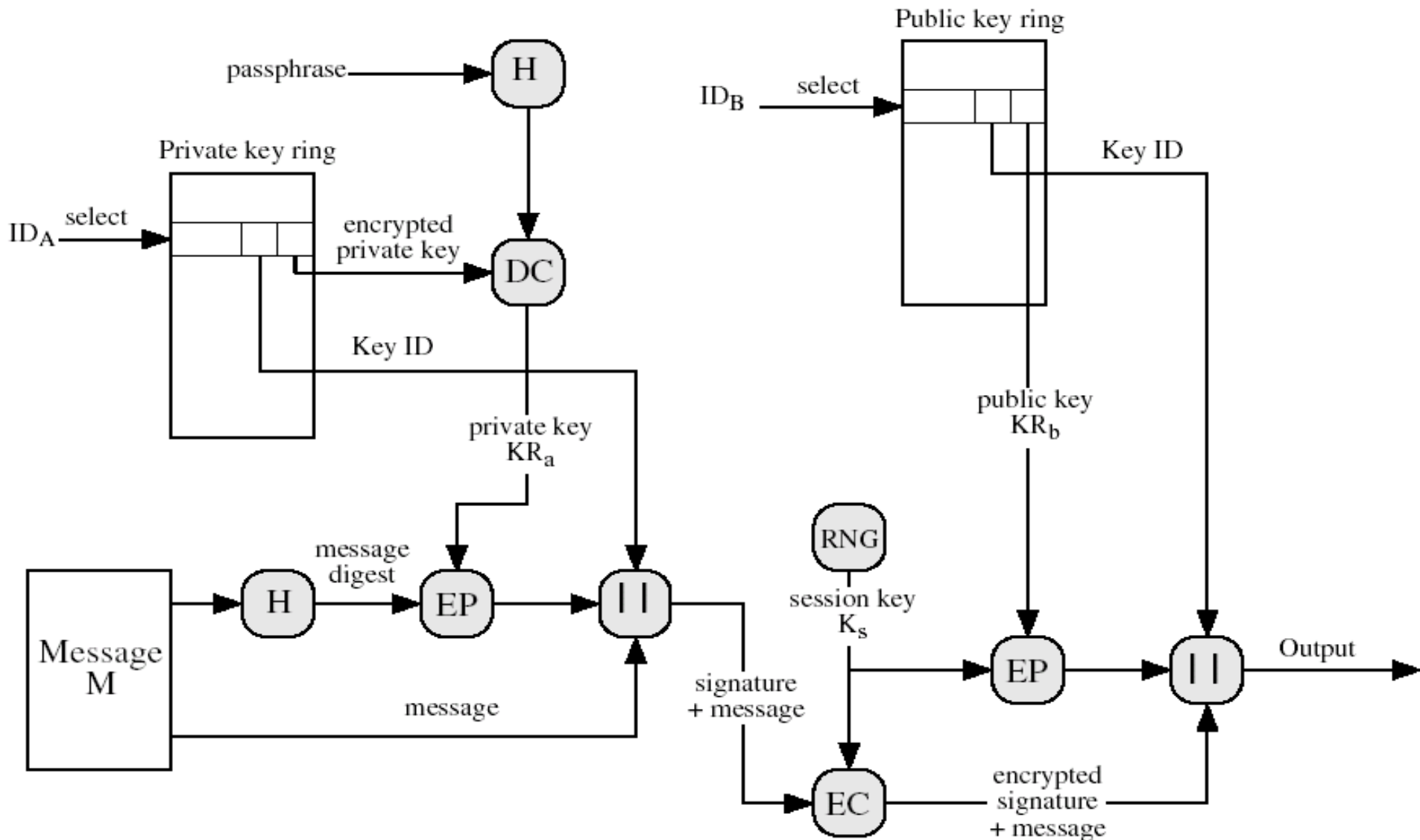
* PGP Message Format

Content

Operation



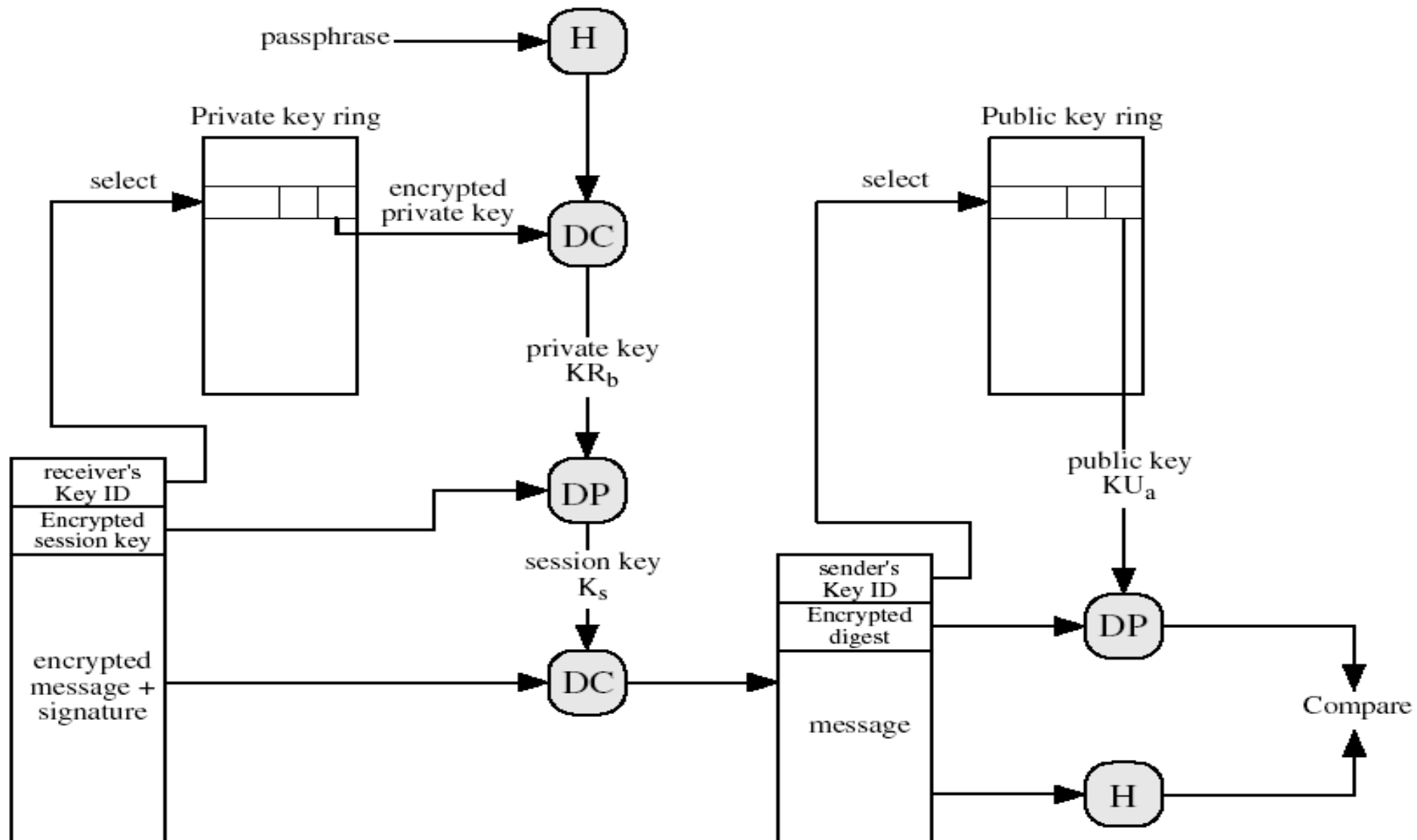
PGP Message Generation



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.