CHAPTER 15

Pretty Good Privacy

Pretty Good Privacy

- PGP (Pretty Good Privacy) provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.
- It is the effort of a single person Phil Zimmerman.
- PGP has grown explosively and is now widely used because:
  1. It is available free worldwide in versions that run on a variety of platforms including: Windows, UNIX, Macintosh and many more.
  2. It is based on algorithms that are extremely secure. Specially; 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.
  3. It has a wide range of applicability from corporations to individuals.
  4. It was not developed by, nor is it controlled by any governmental or standards organization.
  5. PGP is now on an Internet Standards track (RFC 3156).

Notation

- The following symbols are used:
- K_S = Session key used in Symmetric encryption scheme.
- KR_a = Private key of user A, used in public-key encryption scheme.
- KU_a = Public key of user A, used in public-key encryption scheme.
- EP = Public-key encryption.
- DP = Public-key decryption.
- EC = Symmetric encryption.
- DC = Symmetric decryption.
- H = Hash function.
- || = Concatenation.
- Z = Compression using ZIP algorithm.
- R64 = Conversion to radix 64 ASCII format.

Operational Description

- The actual operation of PGP consists of five services:
  1. Authentication
  2. Confidentiality
  3. Compression
  4. E-mail Compatibility and
  5. Segmentation.

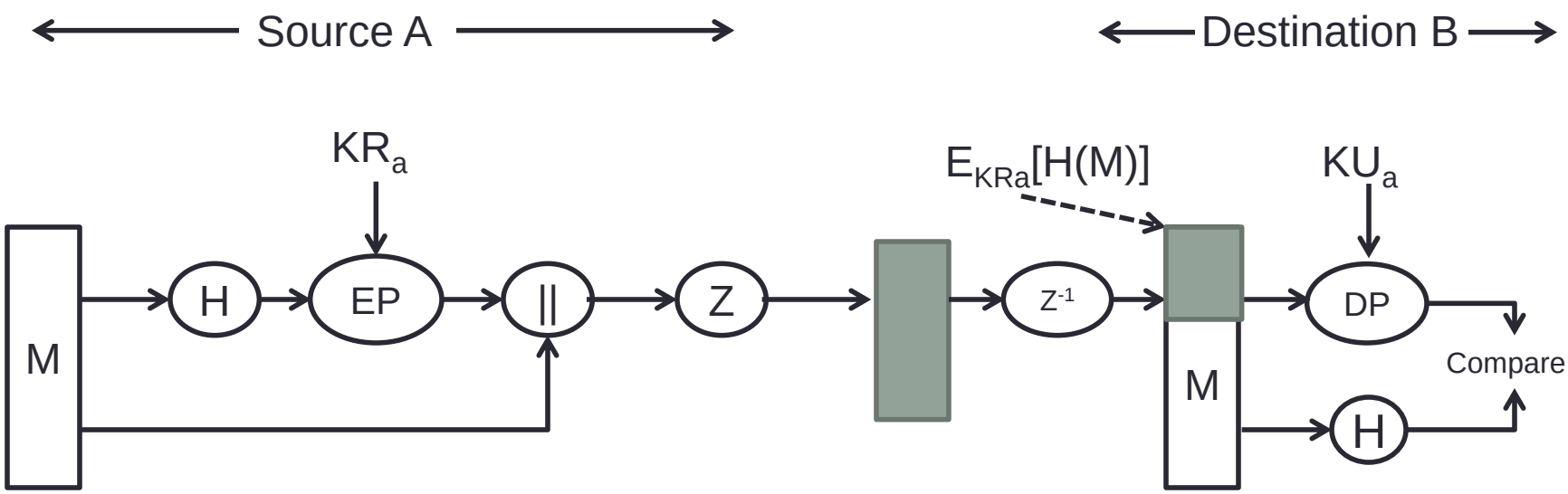
- Authentication:

Figure on the next slide illustrates the digital signature service provided by PGP. The sequence is as follows:

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

Authentication

- The combination of SHA-1 and RSA provides an effective digital signature scheme.
- As an alternative, signatures can be generated using DSS/SHA-1.



Confidentiality

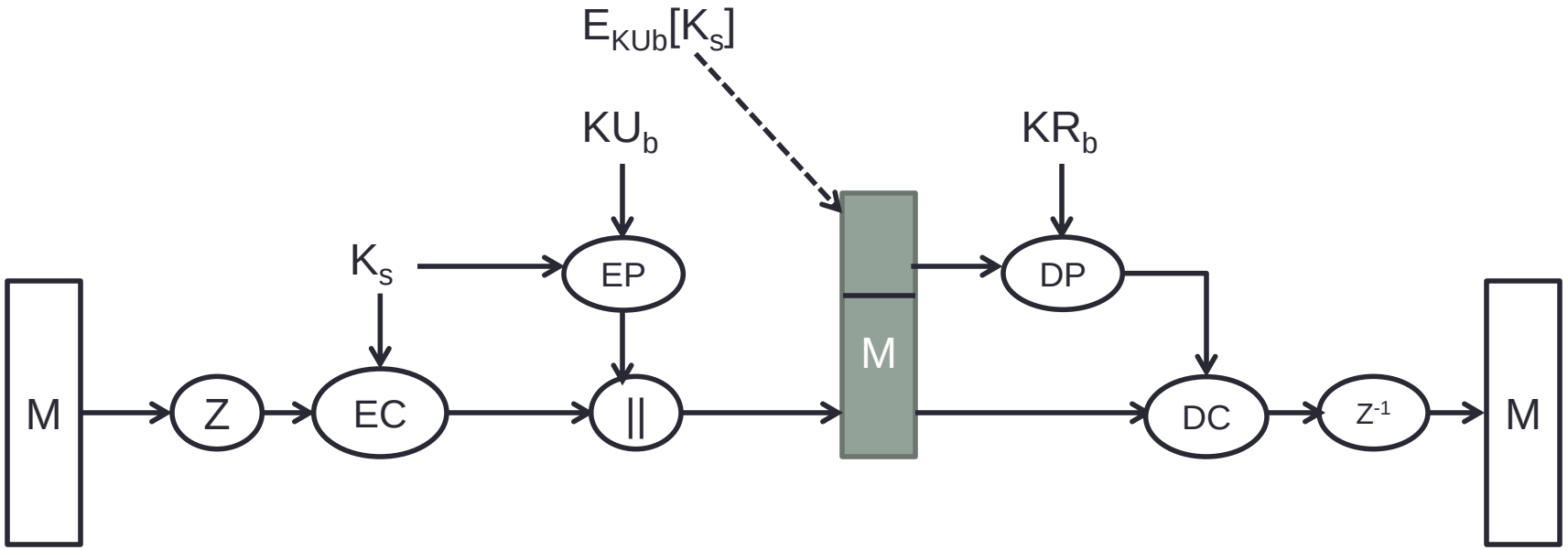
- Confidentiality is provided by encrypting messages to be transmitted or to be stored locally as files.
- In both cases the symmetric algorithm CAST-128 may be used.
- Alternatively, IDEA or 3DES may be used.
- In PGP each symmetric key is used only once. That is a new key is generated as a random 128-bit number for each message.
- The session key is bound to the message and transmitted with it.
- To protect the key, it is encrypted with the receiver’s public key.

Confidentiality Continue…

- The sequence can be described as follows:
  1. The sender generates a message and a random 128-bit number to used as a session key for this message only.
  2. The message is encrypted using CAST-128 (or IDEA or 3DES) with the session key.
  3. The session key is encrypted with RSA, using the recipient's public key and is prepended to the message.
  4. The receiver uses RSA with its private key to decrypt and recover the session key.
  5. The session key is used to decrypt the message.

Confidentiality Continue…

- Confidentiality only:

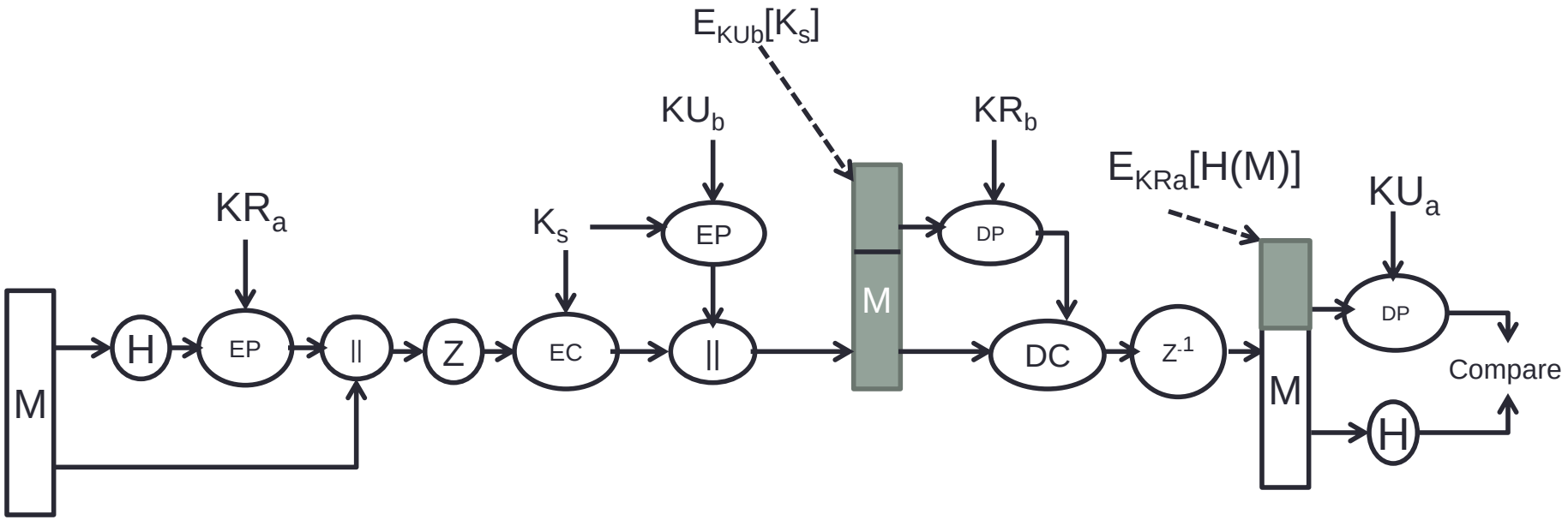


Confidentiality and Authentication

- Both Authentication and Confidentiality can be used for the same message.
- First, a signature is generated for the plaintext message and prepended to the message.
- The plaintext message plus signature is encrypted using CAST-128 (or IDEA or 3DES) and the session key is encrypted using RSA (or ElGamal).
- In summary, when both services are used, the sender first signs the message with its own private key, then encrypts the message with a session key, and then encrypts the session key with the recipient's public key.

Confidentiality and Authentication Continue…

Confidentiality and Authentication:



Compression

- As a default, PGP compresses 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 signature is generated before compression for two reasons:
  a. It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification.
  b. Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty.
- 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.

E-mail Compatibility

- When PGP is used, at least part of the block to be transmitted is encrypted.
- If only the signature service is used, then the message digest is encrypted.
- If the confidentiality service is used, the message plus signature (if present) are encrypted.

Segmentation and Reassembly

- E-mail facilities often are restricted to a maximum message length.
- Any message longer than the allowed maximum(typically 50,000 octets) must be broken up into smaller segments, each of which is mailed separately.
- PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail.
- The segmentation is done after all of the other processing, including the radix-64 conversion
- Thus the session key components and signature component appear only once, at the beginning of the first segment.