SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

19ECT301-COMMUNICATION NETWORKS III YEAR/ V SEMESTER

UNIT 4- NETWORK & DATA SECURITY

TOPIC –Email Security

Why Study E-mail Security?

- After web browsing, e-mail is the most widely used network-reliant application.
- Mail servers, after web servers, are the most often attacked Internet hosts.
- Basic e-mail offers little security, counter to public perception.
- Good technical solutions are available, but not widely used.
  - If we understand why this is so, we might understand something about why security is ‘hard’.

Threats to E-mail

- Loss of confidentiality.
  - *E-mails are sent in clear over open networks.*
  - *E-mails stored on potentially insecure clients and mail servers.*
- Loss of integrity.
  - *No integrity protection on e-mails; anybody be altered in transit or on mail server.*

Threats to E-mail

- Lack of data origin authentication.
  - *Is this e-mail really from the person named in the* From:*field?*
- Lack of non-repudiation.
  - *Can I rely and act on the content? (integrity)*
  - *If so, can the sender later deny having sent it? Who is liable if I have acted?*

Threats to E-mail

- Lack of notification of receipt.
  - *Has the intended recipient received my e-mail and acted on it?*
  - *A message locally marked as ‘sent’ may not have been delivered.*

E-mail security

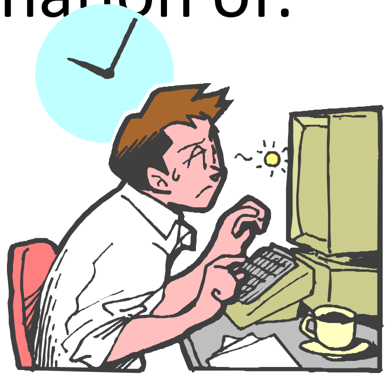
- What are the Options?
  - Secure the server to client connections (easy thing first)
    - https access to webmail
    - Protection against insecure wireless access
  - Secure the end-to-end email delivery
    - The PGPs of the world
    - Practical in an enterprise intra-network environment

E-mail security

- Email based Attacks
  - Active content attack
    - Clean up at the server
  - Buffer over-flow attack
    - Fix the code
  - Trojan Horse Attack
  - Web bugs (for tracking)
    - Mangle the image at the mail server

E-mail security

- Software for encrypting email messages has been widely available for more than 15 years, but the email-using public has failed to adopt secure messaging. This failure can be explained through a combination of:
  - – technical,
  - – community,
  - – and usability factors



E-mail security

- **Why Don’t People Use Email Security?**
  – I don’t because I don’t care.
  – I doubt any of my usual recipients would understand
  – the significance of the signature.
  – Never had the need to send these kinds of emails.
  – I don’t think it’s necessary to encrypt my email.
  – it’s just another step & something else I don’t have time

E-mail security

- Secure E-mail Standards and Products
  - Other now defunct standards: PEM (privacy enhanced mail), X.400. S/MIME.
  - We focus on PGP

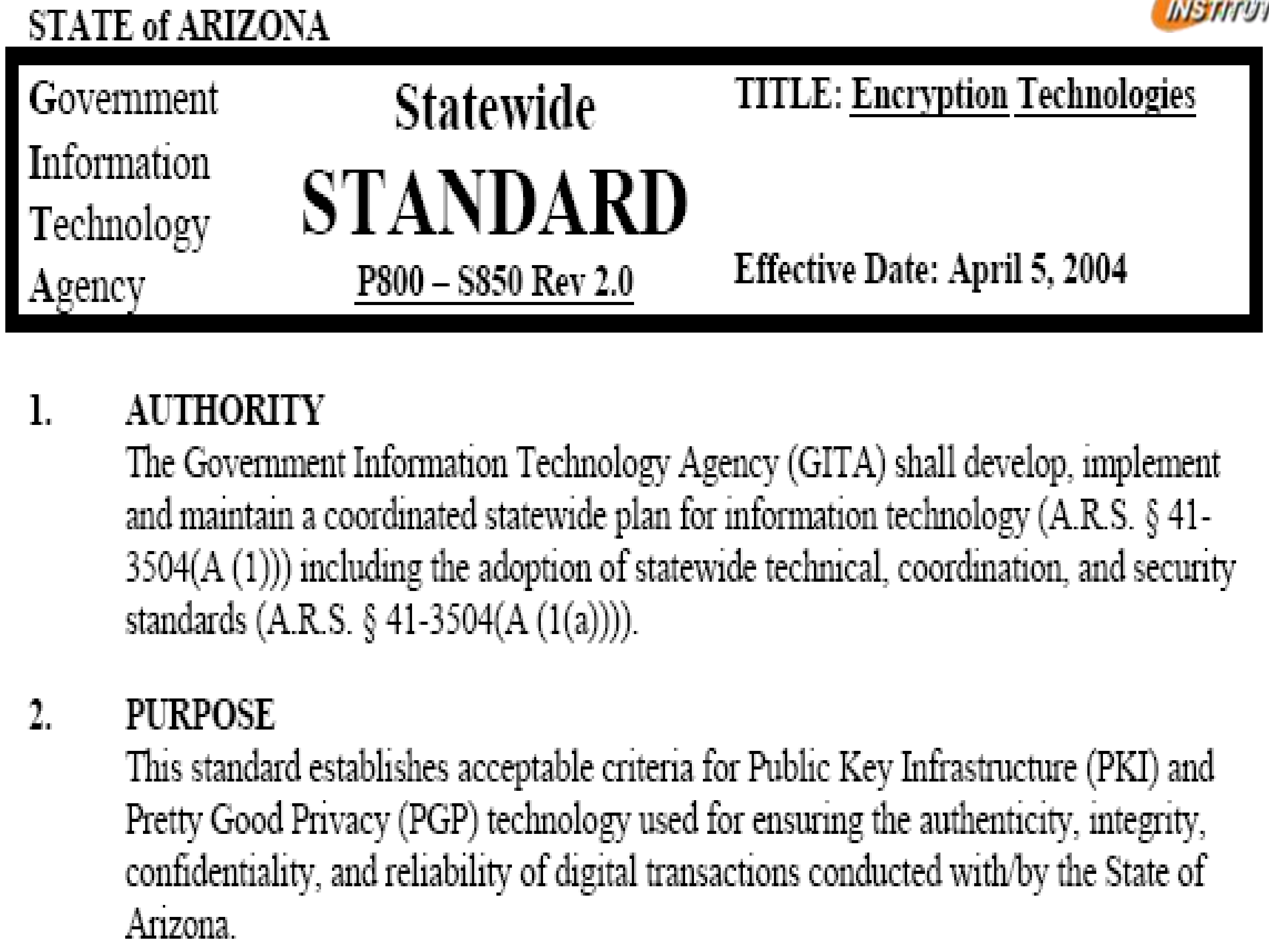
S/MIME

(Secure/Multipurpose Internet Mail Extension)

- Originated from RSA Data Security Inc. in 1995.
- Further development by IETF S/MIME working group at: www.ietf.org/html.charters/smime-charter.html.
- Version 3 specified in RFCs2630-2634.
- Allows flexible client-client security through encryption and signatures.
- Widely supported, e.g. in Microsoft Outlook, Netscape Messenger, Lotus Notes.

PGP
(Pretty Good Privacy)

- Freeware: Open PGP and variants:
  - www.openpgp.org, www.gnupg.org
- Open PGP specified in RFC 2440 and defined by IETF Open PGP working group.
  - www.ietf.org/html.charters/openpgp-charter.html
- Available as plug-in for popular e-m clients, can also be used as stand-alo software.



STATE of ARIZONA

Government Information Technology Agency	Statewide **STANDARD** <u>P800 – S850 Rev 2.0</u>	TITLE: <u>Encryption Technologies</u> Effective Date: April 5, 2004

1. **AUTHORITY**
   The Government Information Technology Agency (GITA) shall develop, implement and maintain a coordinated statewide plan for information technology (A.R.S. § 41-3504(A (1))) including the adoption of statewide technical, coordination, and security standards (A.R.S. § 41-3504(A (1(a))).

2. **PURPOSE**
   This standard establishes acceptable criteria for Public Key Infrastructure (PKI) and Pretty Good Privacy (PGP) technology used for ensuring the authenticity, integrity, confidentiality, and reliability of digital transactions conducted with/by the State of Arizona.

PGP
(Pretty Good Privacy)

- *“If all the personal computers in the world—260 million—were put to work on a single PGP encrypted message, it would still take an estimated 12 million times the age of the universe, on average, to break a single message.”*

PGP
(Pretty Good Privacy)

- PGP is an e-mail security program written by Phil Zimmermann, based on the IDEA algorithm for encryption of plaintext and uses the RSA Public Key algorithm for encryption of the private key.
- PGP incorporates tools for developing a public-key trust model and public-key certificate management.

PGP
(Pretty Good Privacy)

- PGP is an open-source freely available software package for e-mail security. It provides authentication; confidentiality; compression; e-mail compatibility; and segmentation and reassembly.

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
(Pretty Good Privacy)

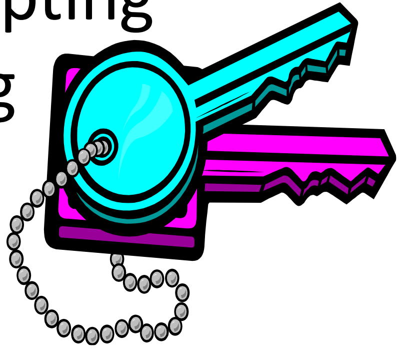
- **Fake PGP:** Since it’s all open source, there are fake versions of the famous software floating about the net. Unless you’re sure that your copy of the program is from a trusted source, it wouldn’t be surprising to realize one day that your pass phrase was sent to an attacker via email the moment you went online! Once he has your pass phrase, he has your private key.

PGP (Pretty Good Privacy)

- **PGP Algorithms**
  - **Symmetric encryption:**
    - DES, 3DES, AES and others.
  - **Public key encryption of session keys:**
    - RSA or ElGamal.
  - **Hashing:**
    - SHA-1, MD-5 and others.
  - **Signature:**
    - RSA, DSS, ECDSA and others.

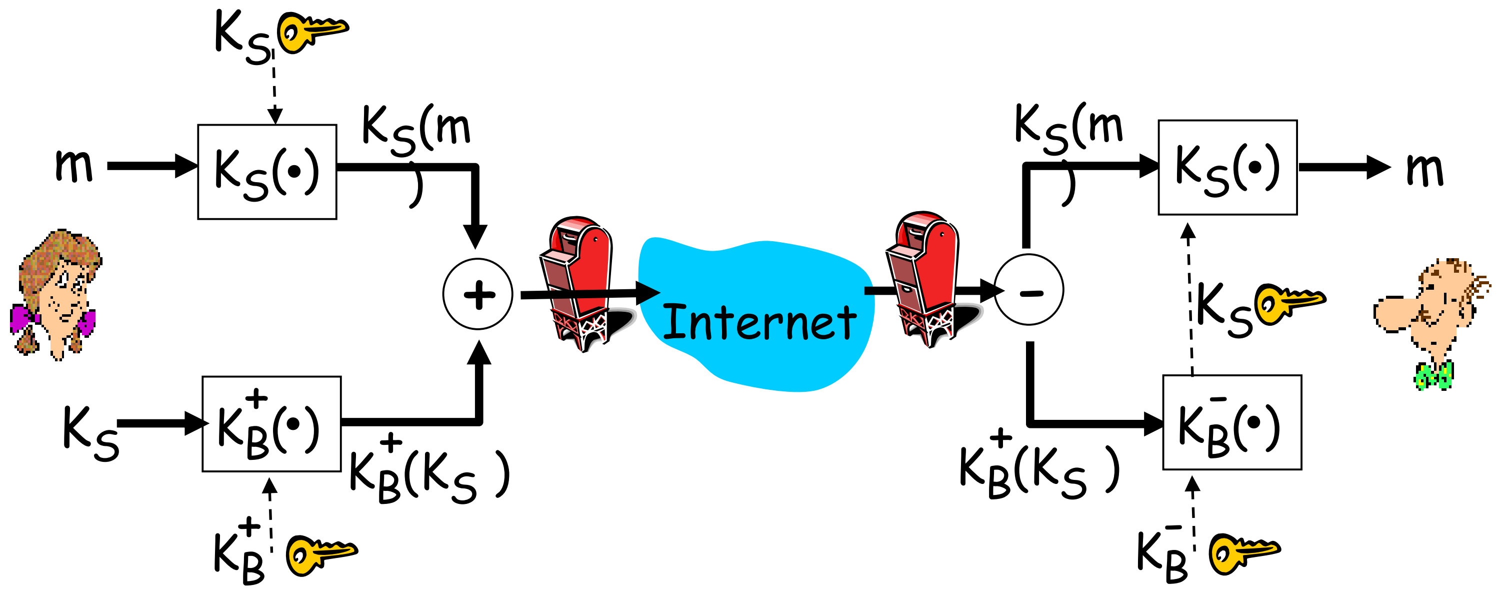
PGP
(Pretty Good Privacy)

– PGP use:
  - **public keys** for encrypting session keys / verifying signatures.
  - **private keys** for decrypting session keys / creating signatures.



PGP

- Alice wants to send confidential e-mail, m, to Bob.

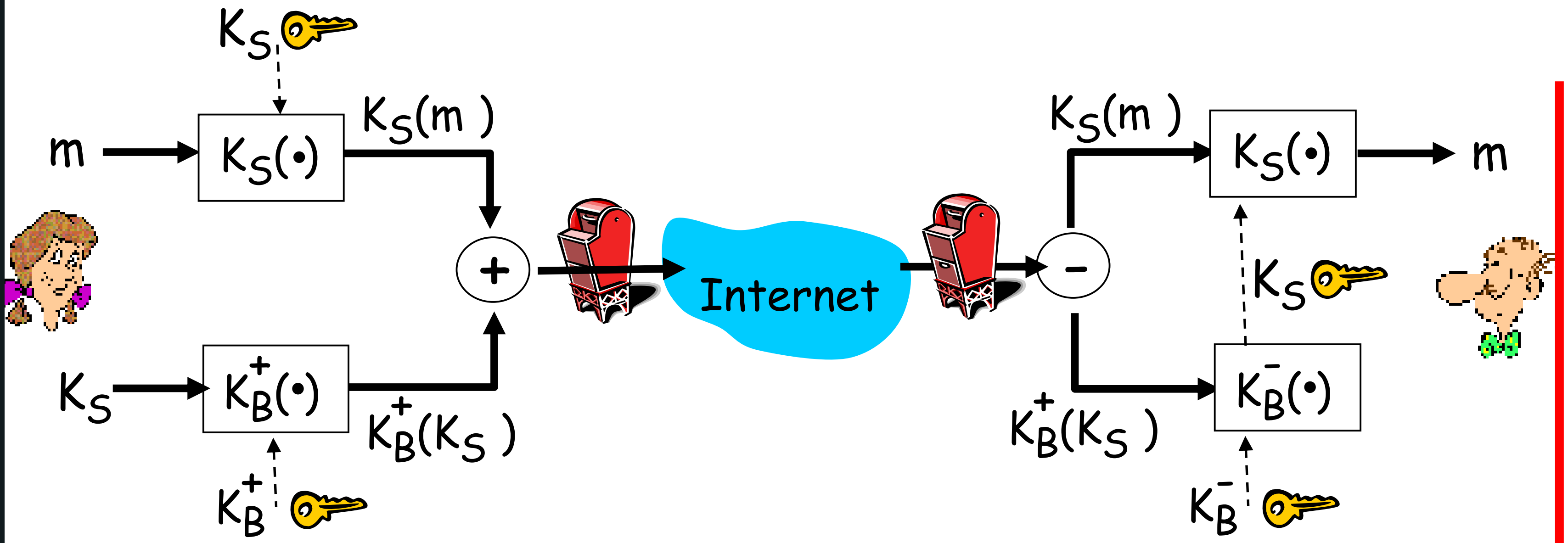


Alice:

- generates random *symmetric* private key, K_S.
- encrypts message with K_S (for efficiency)
- also encrypts K_S with Bob's public key.
- sends both $K_S(m)$ and $K_B(K_S)$ to Bob.

PGP

- Alice wants to send confidential e-mail, m, to Bob.

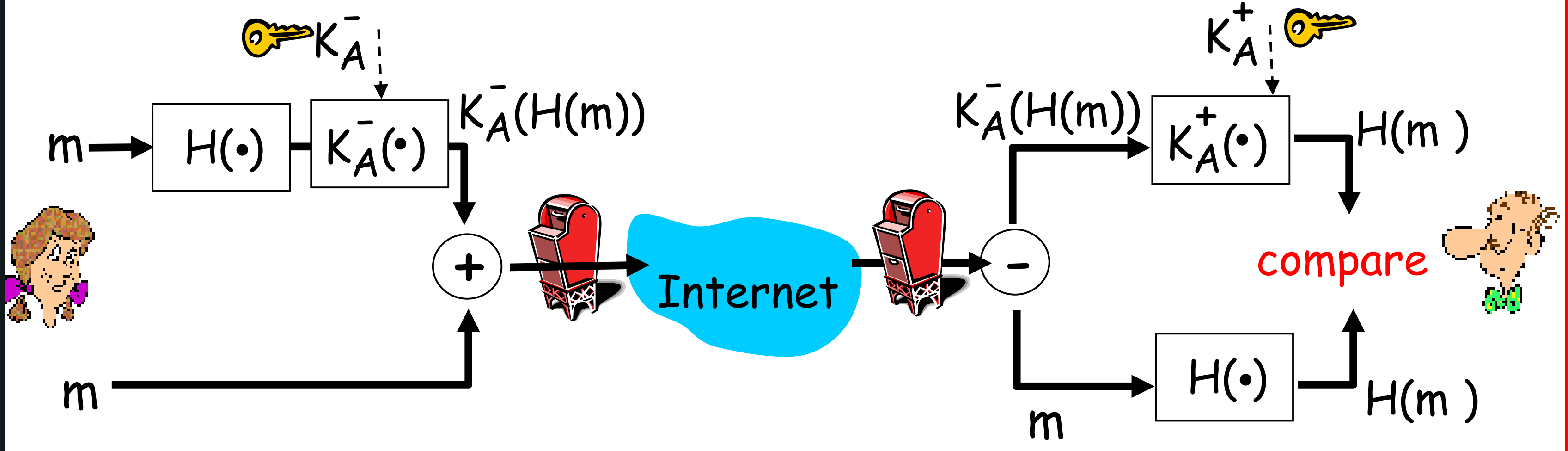


Bob:

- uses his private key to decrypt and recover K_S
- uses K_S to decrypt $K_S(m)$ to recover m

PGP

- Alice wants to provide sender authentication message integrity.

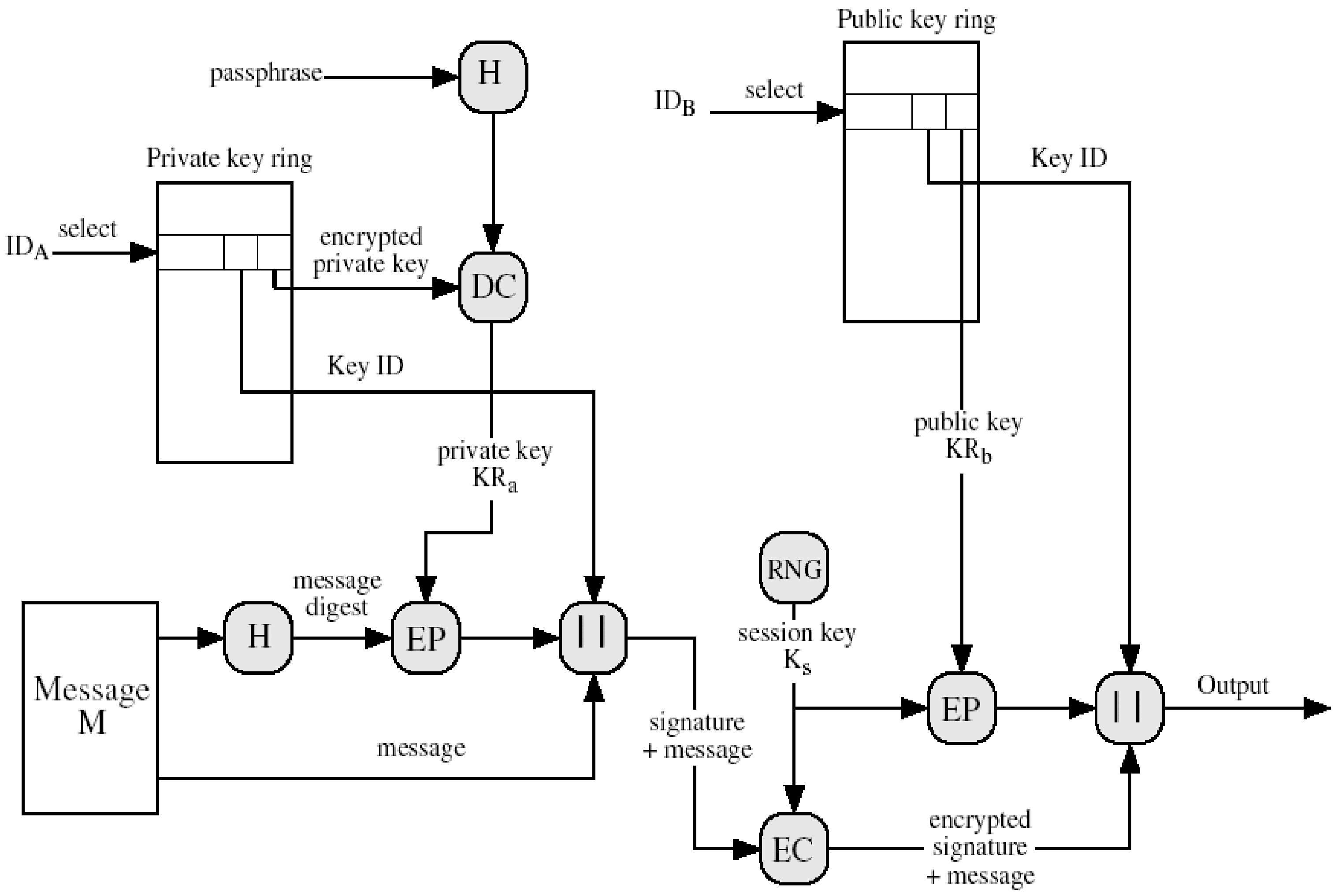


- Alice digitally signs message.
- sends both message (in the clear) and digital signature.

PGP
(Pretty Good Privacy)

- PGP Key Rings
  - PGP supports multiple public/private keys pairs per sender/recipient.
  - Keys stored locally in a *PGP Key Ring* – essentially a database of keys.
  - Private keys stored in encrypted form; decryption key determined by user-entered pass-phrase.

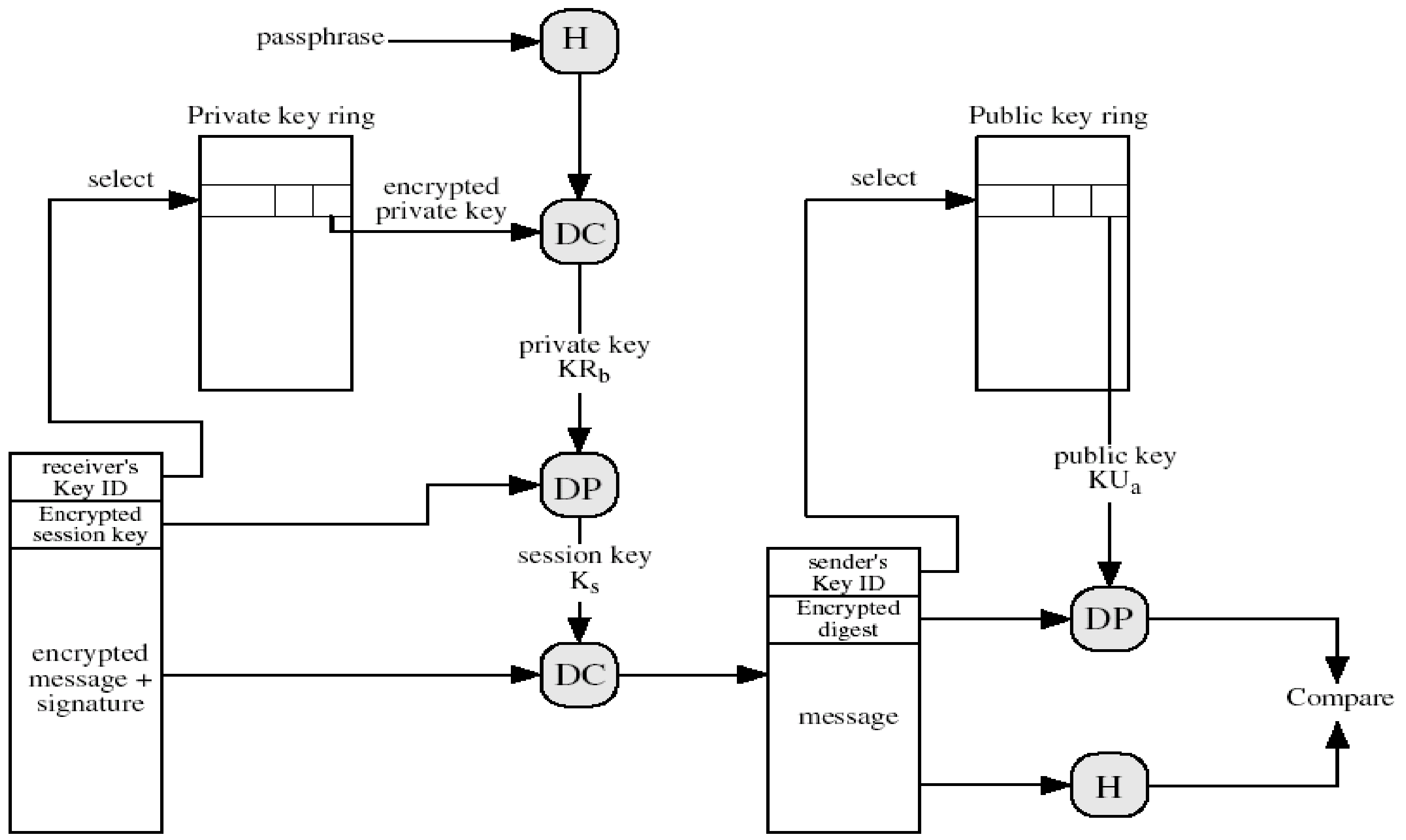
PGP Message Generation



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

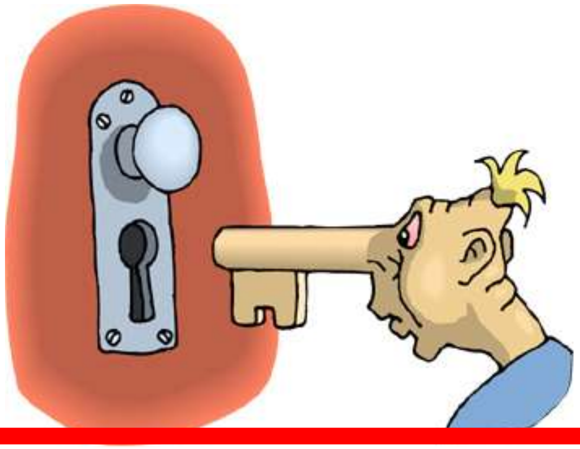


PGP Message Reception

- 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.

PGP
(Pretty Good Privacy)

- Key Management for PGP
  - Public keys for encrypting session keys / verifying signatures.
  - Private keys for decrypting session keys / creating signatures.
  - Where do these keys come from and on what basis can they be trusted?



PGP (Pretty Good Privacy)

- PGP adopts a trust model called the *web of trust.*
- No centralised authority
- Individuals sign one another’s public keys, these “certificates” are stored along with keys in key rings.
- PGP computes a *trust level* for each public key in key ring.
- Users interpret trust level for themselves.

PGP
(Pretty Good Privacy)

- Trust levels for public keys dependent on:
  - Number of signatures on the key;
  - Trust level assigned to each of those signatures.
- Trust levels recomputed from time to time.

PGP
(Pretty Good Privacy)

- **Security of PGP**
- There are many known attacks against PGP.
- Attacks against cryptoalgorithms are not the main threat
- IDEA is considered strong, and while cryptoanalysis advances, it should be strong still for some time.
- RSA may or may not be strong. There are recent rumors of possible fast factorization algorithms..
- The main threats are much more simple.

PGP (Pretty Good Privacy)

- An attacker may socially engineer himself into a web of trust, or some trustable person may change. Then he could falsify public keys. This breaks most of the security.
- PGP binaries can be corrupted when they are obtained.
- The PGP binaries can be modified in the computer.
- The passphrase can be obtained by a Trojan. Weak passphrases can be cracked.
- On multiuser system, access to the secret key can be obtained.

Thank You!