# Topic 10: A Security System

Exploring real-life security solutions:
PGP (Pretty Good Privacy)

---

*Sources:*
*1. https://en.wikipedia.org/wiki/Pretty_Good_Privacy*
*2. http://www.facweb.iitkgp.ac.in/~sourav/PGP.pdf*

# Overview

- Part 1
  - PGP overview
- Part 2
  - PGP services (PGP cryptographic functions)
  - PGP message format
- Part 3
  - PGP key rings and trust model
- Part 4
  - PGP message transmission and reception
  - Conclusions

## PGP Overview – What it is for

- PGP is a general purpose application to protect files – files in storage and emails.
- There are two most notable security packages which can be used to protect emails:
  - PGP (Pretty Good Privacy)
  - S/MIME (Secure/Multipurpose Internet Mail Extension)
- The two packages essentially provide **the same security functionality**, with two major differences
  - PGP uses bottom-up approach to public key management (web-of-trust model), whereas S/MIME uses a top-down approach (PKI X.509).
  - PGP does not include the sender's public key with each message, rather it includes a key ID.

## PGP Overview – Why I choose PGP

- S/MIME is recommended by NIST, while PGP widely used for personal e-mail and file protections.
- Based on well-known crypto algorithms.
- Documentation & source code are freely available.
- Can be used not just for email protections but also for protecting file storage.
- Uses a different trust model (from X.509), which has benefits but also problems.

# PGP Overview – Email security problems

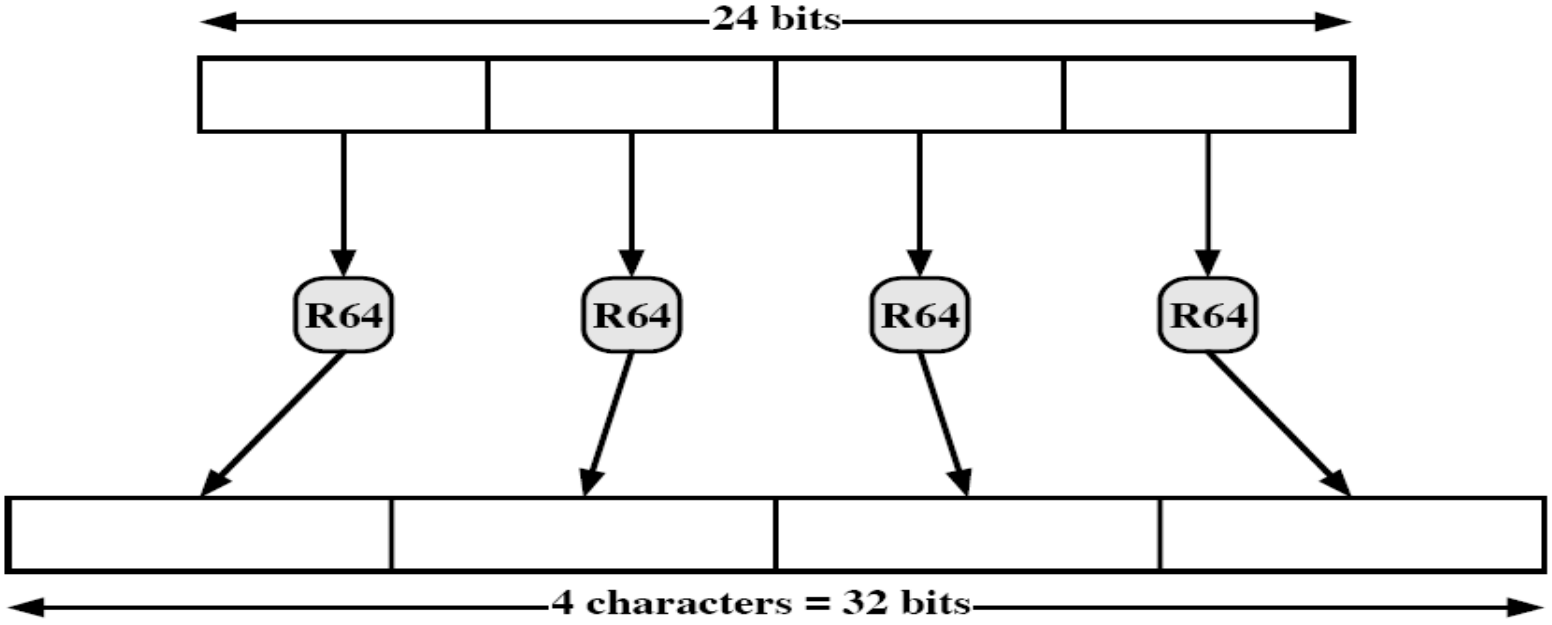
- Message interception
- Message interception and subsequent replay
- Message content modification
- Message origin modification
- Message content forgery (by an outsider or by a recipient)
- Message origin forgery (by an outsider or by a recipient)
- Denial of message transmission/origin (repudiation of transmission/origin)
- Denial of message reception (repudiation of receipt)

## PGP Overview – Capabilities/Services

- Message confidentiality service (disclosure protection): hybrid encryption.
- Message integrity: digital signature.
- Origin authentication & non-repudiation of origin (if public keys certified): web-of-trust.
- E-mail compatibility (encoded character set is universally representable at all sites): radix-64 conversion
- Segmentation: segment/reassemble a long message
- Compression: compresses a message after applying the signature but before encryption
- Trust model (web-of-trust) for key management

# PGP Overview - Compatibility

- Encrypted message may contain some arbitrary octets.
- Many email systems only allow the use of ASCII text.
- Radix-64 conversion is used to convert (table-map) each group of three octets of binary data into four printable ASCII characters.



# PGP Services

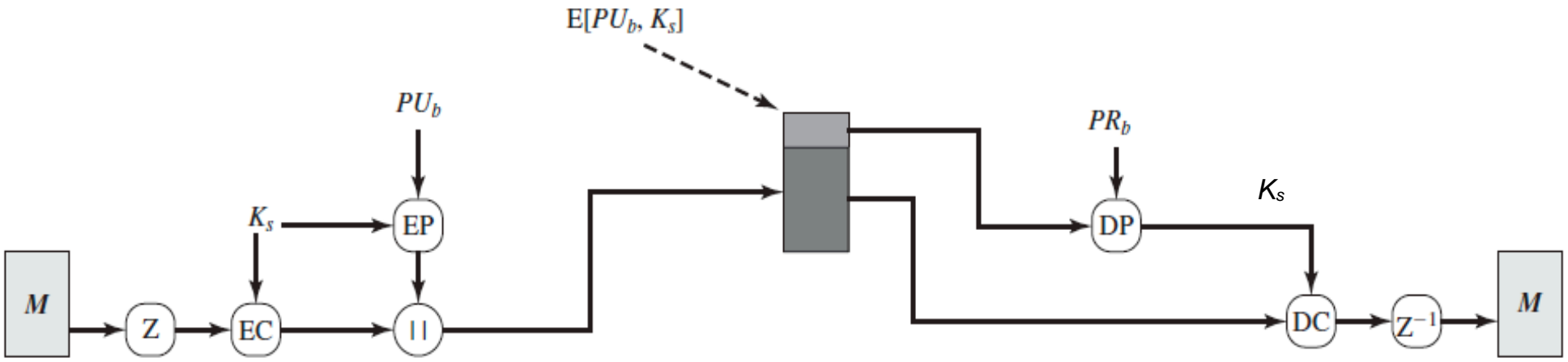
| Functions/services | Algorithms Used | Description |
|---|---|---|
| Digital signature/ Message authentication | DSS/SHA, RSA/SHA (MD5, …) | Message (M) is hashed, timestamped and signed using DSS or RSA |
| Message encryption/ confidentiality | AES, CAST, IDEA, 3DES, …, with Diffie-Hellman, and RSA | M is encrypted using a one-time session key; session key is established using D-H or encrypted using RSA with recipient's public key. |
| Compression | ZIP | |
| E-Mail compatibility | Radix-64 conversion | Local form (e.g. 8-bit binary stream) ←-→ printable ASCII characters |
| Segmentation | - | Performs segmentation/reassembly to accommodate maximum message size limitations |

# Part 2 Overview

- PGP services (PGP cryptographic functions)
- PGP message format

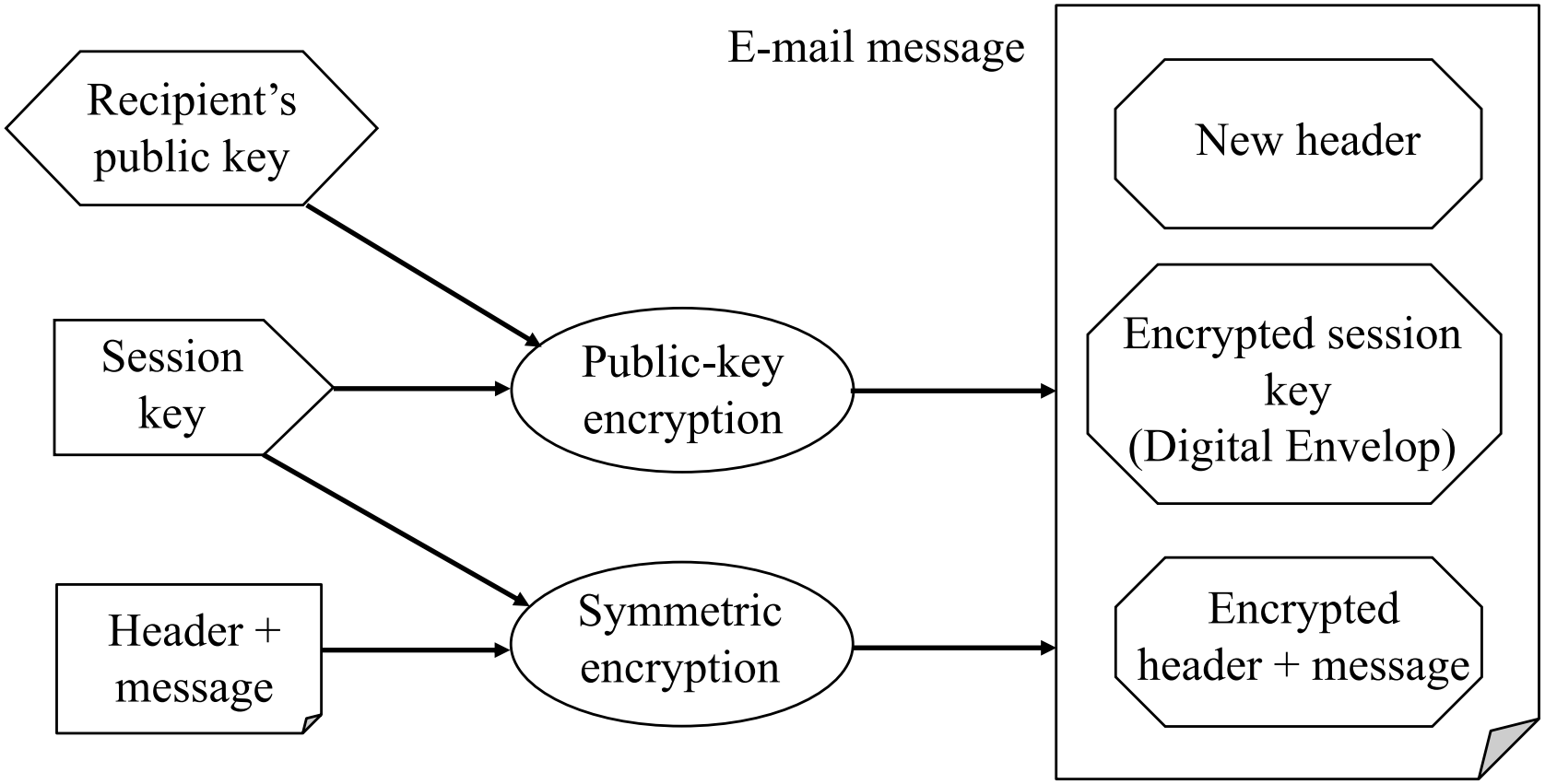
## PGP Services – Confidentiality Only

- Message content is encrypted with a symmetric session key.
- The session key is encrypted with the recipient's public key.
- Notation: $Z/Z^{-1}$ = compression/decompression, EC/DC=symmetric-key encryption/decryption, EP/DP=public-key encryption/decryption, '||'=concatenation



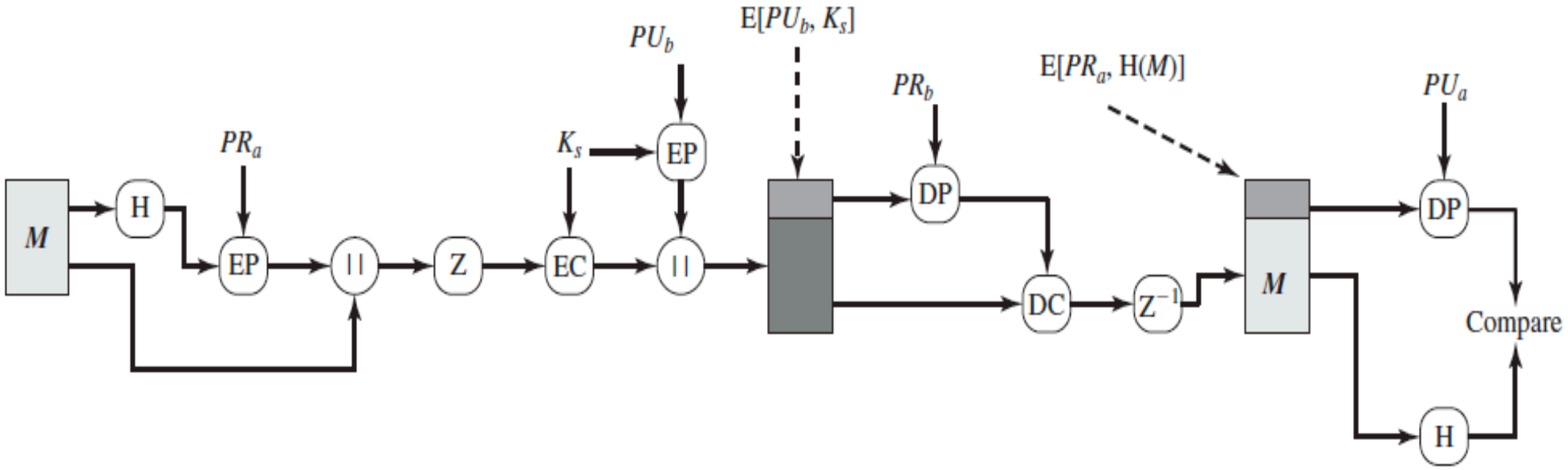
# PGP Services – Confidentiality Only

- Message Confidentiality

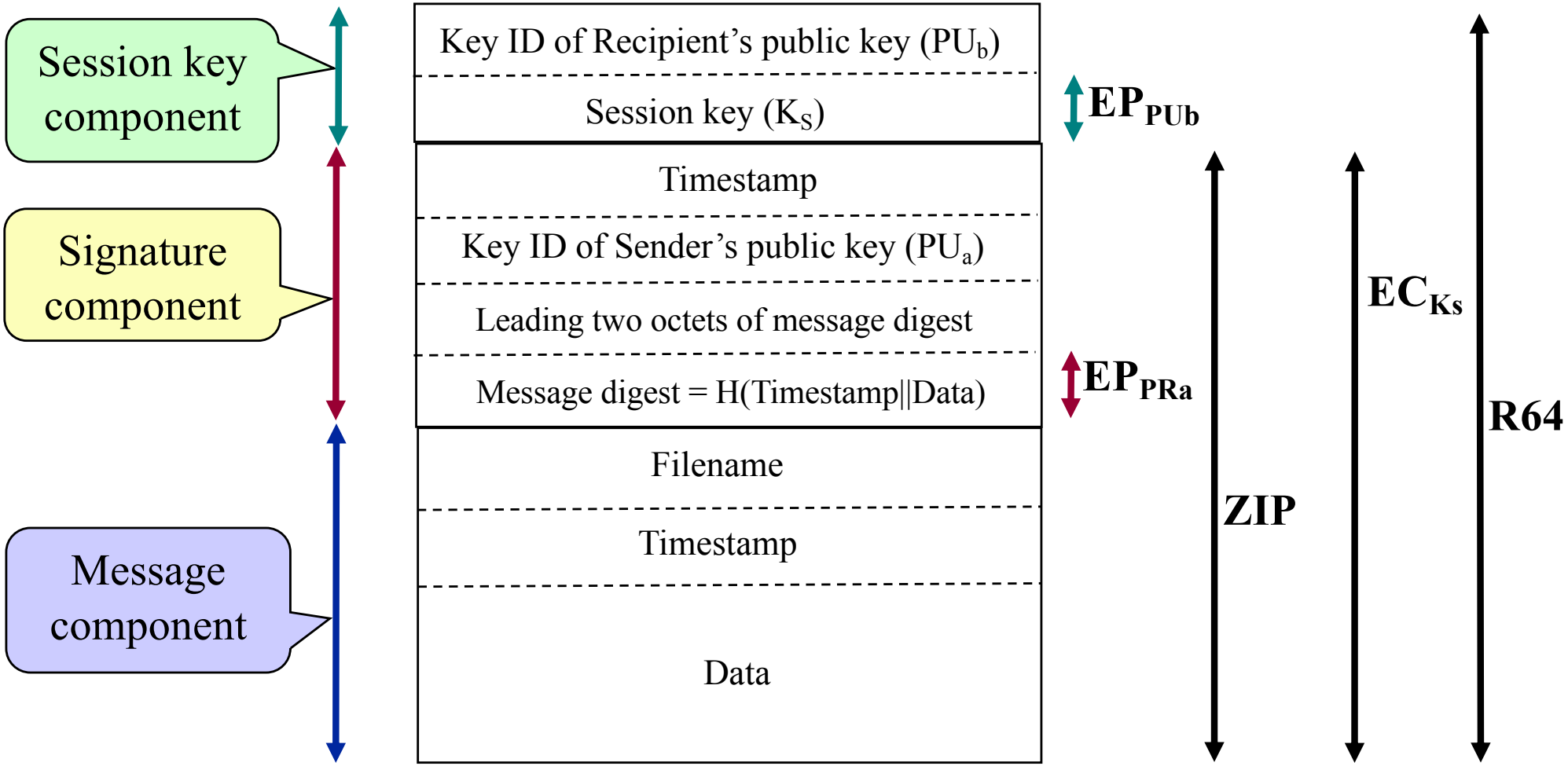


# PGP Services – Confidentiality and Authentication

- The message is digitally signed for the sender authentication
- The message content is encrypted for its confidentiality



# PGP Message Format



## PGP Message Format

- Session key component: include the session key and the identifier of the recipient's public key that was used by the sender to encrypt the session key.
- Message component: include the actual data to be transmitted, as well as a file name and a timestamp that specifies the time of the message creation.

## PGP Message Format

- Signature component:
  - Timestamp – when the signature was generated.
  - Message digest,160-bit SHA-1 digest, encrypted with the sender's private key. This signature is calculated over the signature timestamp concatenated with the data portion of the message component.
  - Leading two bytes of message digest is to enable the recipient to determine if the correct public key was used to decrypt the message digest for authentication by comparing this plaintext copy of the first two bytes with the first two bytes of the decrypted digest.

# Part 3 Overview

- PGP key rings
- PGP trust model

## Key Rings

- Key Identifiers (Key IDs)
  - Each user can have multiple public/private key pairs, so they need identifiers (IDs).
  - How could a recipient identify which public key should be used for an incoming message?
    - Send the public key with the message, or
    - Use a key identifier to identify a particular key in a key server.
  - PGP uses the latter approach, i.e. use a **user ID** plus a **key ID** to identify a public key.
  - Key ID = $(KU_a \bmod 2^{64})$, i.e. the least significant 64 bits of the public key.

## Key Rings

- Each user maintains a pair of data structures (i.e. a pair of key rings) in his/her system:
  - A private-key ring to store the private/public key pairs owned by this user.
  - A public-key ring to store the public keys of other users known to this user.
- Both key rings can be indexed by either User ID or Key ID.
- The user's private key is encrypted using the symmetric algorithm, CAST-128 (128-bit key), and the key-encryption key is the first 128 bits of the hash code of a passphrase, *Pi, chosen by the user. That is,* the encrypted private key = $EC_{H(Pi)}[PR_i]$, where $H$ is a hash function.

## Private-Key Ring

- Used to store the public key & private key pairs owned by this user.
- A table of rows; each entry for one key pair containing:
  - Timestamp: when the key pair was generated.
  - Key ID: identifier of this key (index).
  - Public key: the public key for the entry.
  - Encrypted private key: the private key is encrypted using the hash value of a passphrase.
  - User ID: usually the user's email address; may be different for different key pairs (index).

## Public-Key Ring

- Used to store public keys of other users.
- A table of rows; each entry for one key containing:
  - Timestamp: when the entry was generated.
  - Key ID: identifier of this key.
  - Public Key: the public key for this entry.
  - Owner Trust
  - User ID: identifier for the owner of this key.
  - Key Legitimacy
  - Signature: the signature signed on this certificate (i.e. the binding between this public key and its owner identified by 'User ID').
  - Signature Trust

# General Structures of Private-/Public-Key Rings

| Timestamp | Key ID* | Public Key | Encrypted Private Key | User ID* |
|---|---|---|---|---|
| … | … | … | … | … |
| $T_i$ | $PU_i \bmod 2^{64}$ | $PU_i$ | $EC_{H(Pi)}[PR_i]$ | User $i$ |
| … | … | … | … | … |

| Timestamp | Key ID* | Public Key | Owner Trust | User ID* | Key Legitimacy | Signature | Signature Trust |
|---|---|---|---|---|---|---|---|
| … | … | … | … | … | … | … | … |
| $T_i$ | $PU_i \bmod 2^{64}$ | $PU_i$ | $\text{Trust\_flag}_i$ | $\text{User}_i$ | $\text{Trust\_flag}_i$ | | |
| … | … | … | … | … | … | … | … |

* Field is used to index the tables.

# PGP Trust Model - Web-of-Trust

- Trust is captured by the three attributes
  - **Owner Trust**: the degree to which this PGP user trusts the key's owner to sign public-key certificates for other users; the value is assigned by this user.
  - **Signature Trust**: the trust the signer (of the certificate) has on the identity of the user identified in the certificate; the value is assigned by the PGP system.
  - **Key Legitimacy**: the degree to which this PGP user trusts that this public key belongs to this owner. The higher the value, the stronger the binding between this 'User ID' and this 'Public Key'; the value is calculated by the PGP system based on some heuristics defined by the user based on the trust (signature trust) on the certificate or chain of certificates.

# PGP Trust Model

- **Owner Trust**
  - Assigned by this PGP user (YOU).
  - Indicates the degree YOU trusts the key's owner to sign other public-key certificates.
  - For example, possible values:
    - *unknown user*
    - *partially trusted user*
    - *fully trusted user*

# PGP Trust Model

## Signature Trust

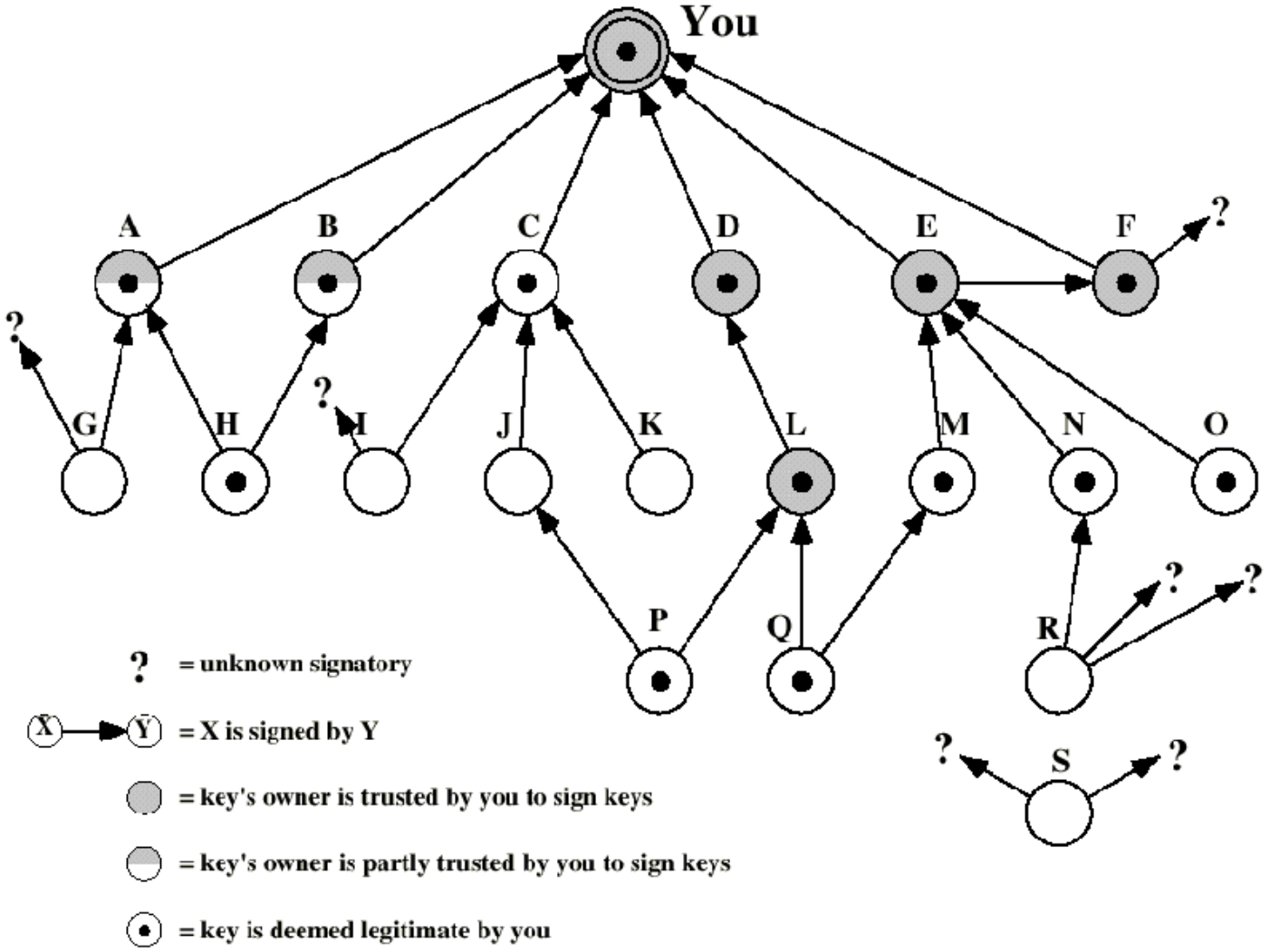
- This value is public on the key server along with the certificate signed.
- It reflects the trust the signer (of the certificate) has on the identity of the user identified in the certificate. If Alice signs (i.e. certifies) Bob's key, this is the value she declares she puts trust in Bob’s identity.

# PGP Trust Model

## Key Legitimacy

- Computed by the PGP system.
- For example, to set full Key Legitimacy (Key Legitimacy = 1) for a pubic key, the public key should
  - be certified by at least one certificate with the value of Signature Trust being ‘fully trusted’,
  - OR be certified by at least two certificates with the value of Signature Trust being ‘partially trusted’.

# PGP Trust Model



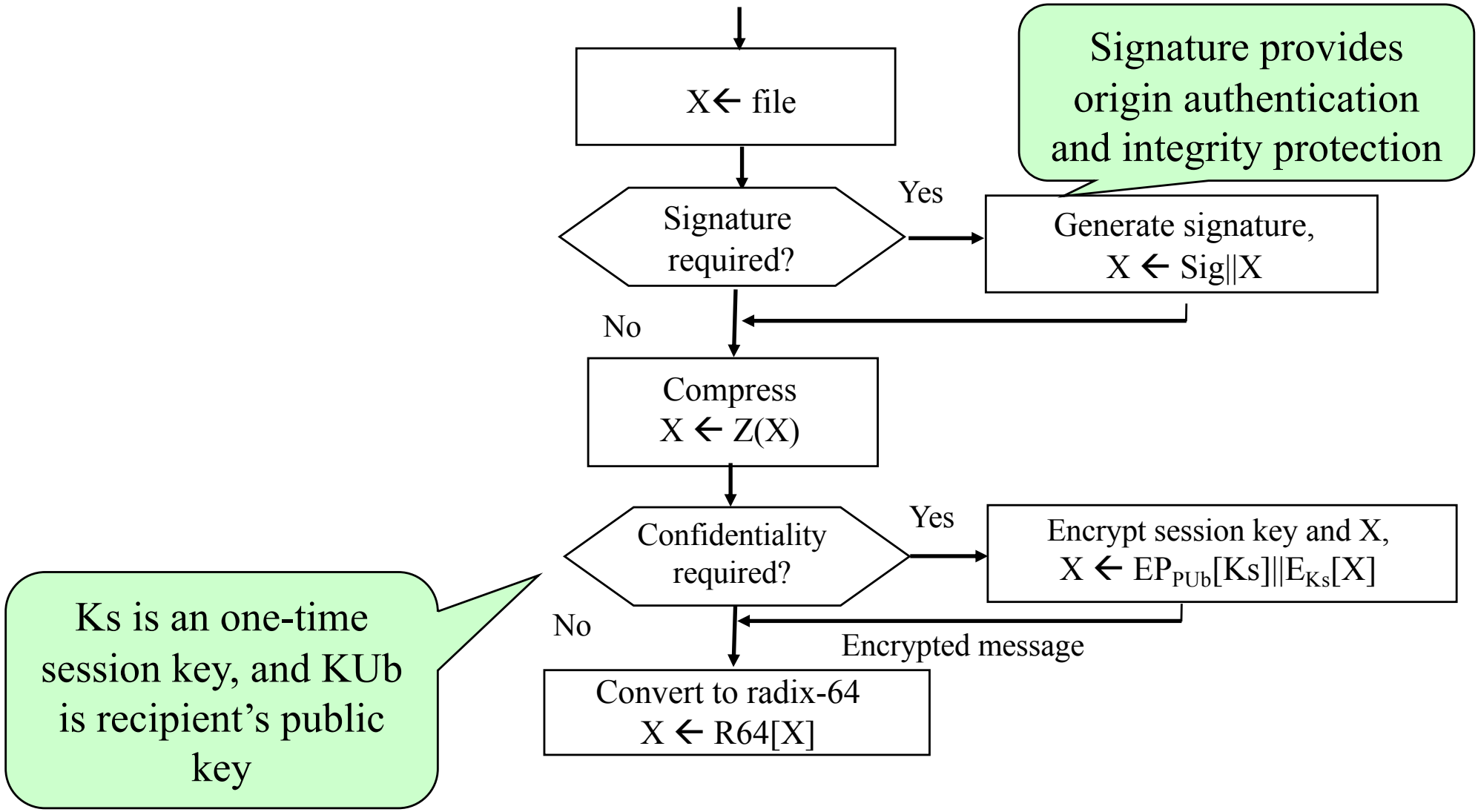
# PGP Trust Model – Public key revocation

- A public key should be revoked, if
  - the private key corresponding to the public key is suspected to have been compromised, or
  - the owner no longer needs the certificate.
- This is done by the owner generating and disseminating a revocation certificate containing the public key to be revoked, which is signed with the corresponding private key.

# Part 4 Overview

- PGP message transmission and reception
- Conclusions

# PGP – Transmission of a PGP message



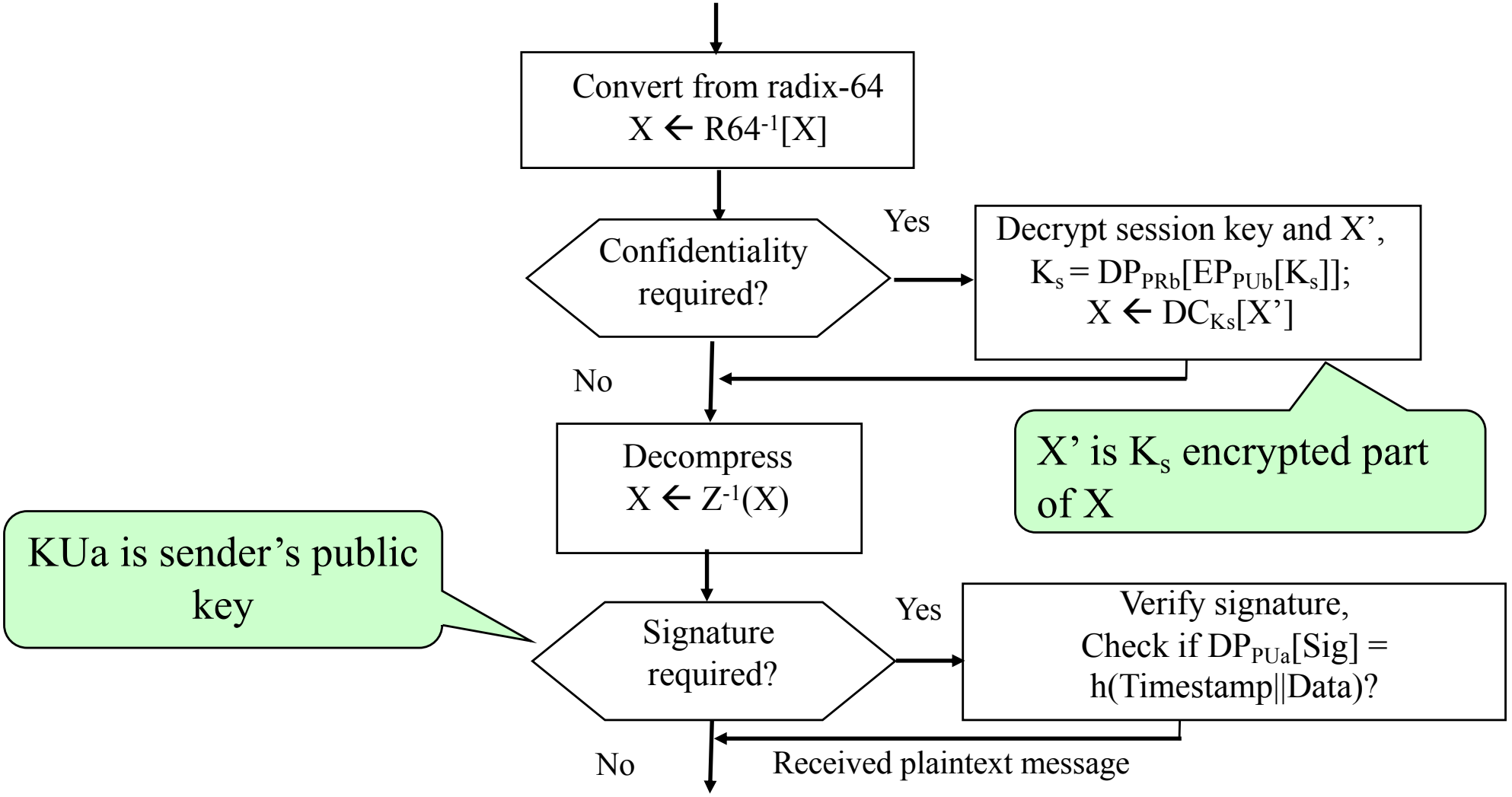
## PGP – Generation of the signature component

- Signing a Message
- PGP retrieves the sender's private key from the private-key ring using the User ID as an index. If no User ID was supplied, retrieve the first private key.
- PGP prompts the user for a passphrase to recover the encrypted private key.
- The signature component of the message is constructed.

## PGP – Generation of the session key component

- PGP generates a session key and encrypts the message using the session key.
- PGP retrieves the recipient's public key from the public-key ring using her User ID as an index.
- The session key component of the message is constructed.

# PGP – Reception of a PGP message



## PGP – Decryption of the incoming message

- PGP retrieves receiver's private key from the private-key ring, using the Key ID field in the session key component of the message as the index.
- PGP prompts the user for the passphrase to recover the private key.
- PGP recovers the session key and use it to decrypt the message.

# PGP – Authentication verification of the incoming message

- PGP retrieves sender's public key from the public-key ring, using the Key ID field in the signature key component of the message as the index.
- PGP decrypts the signature to recover the message digest received.
- PGP computes a fresh message digest based on the message received and compares the two message digests.

## Exercise Question – E10.1

In secure email systems, such as PGP (pretty good privacy), the order of message protection operations performed by a sending entity is to first sign the outgoing/email message, then compress the signed message, and finally encrypt the compressed message. Answer the following questions:

(i) What are the justifications (or benefits) for using this order of operations, and

(ii) Would this order of operations be suitable for all cases of applications?

(iii) Assuming that Alice sends Bob an email, M. Express the whole message that is sent by Alice using protocol notation format.

## Exercise Question – E10.2

Contrast the two trust models in terms of their respective merits and weaknesses:

- PKI X.509
- PGP web-of-trust

## Conclusions

- Security issues affect many aspects of our lives.
- There are many interesting security solutions out there for you to explore.
- Hope, through these exemplar security solutions, you could get a flavour of security problems, and methods/approaches to address the problems.