

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

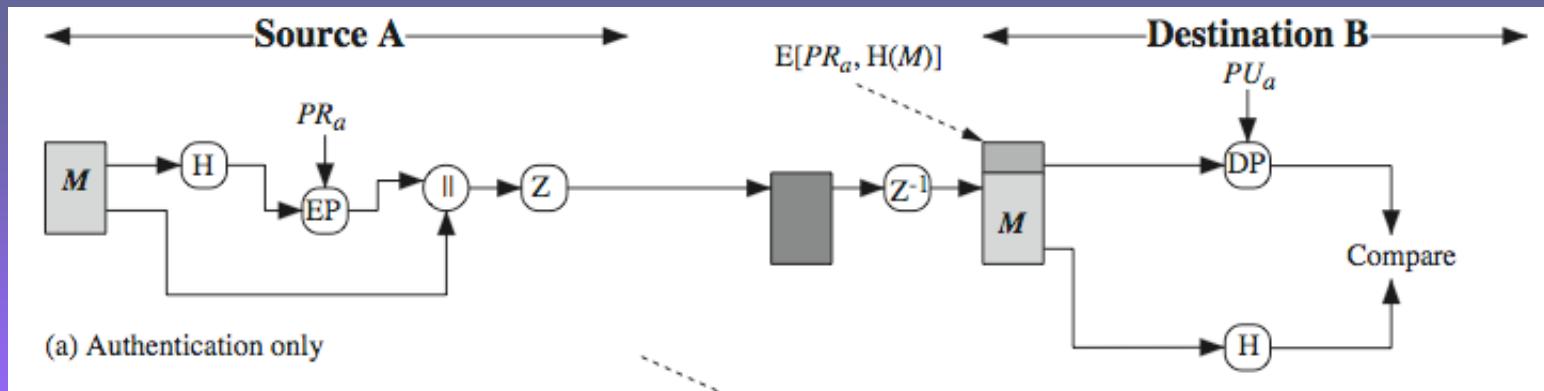
- confidentiality
 - protection from disclosure
- authentication
 - of sender of message
- message integrity
 - protection from modification
- non-repudiation of origin
 - protection from denial by sender

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
- on Unix, PC, Macintosh and other systems
- originally free, now also have commercial versions available

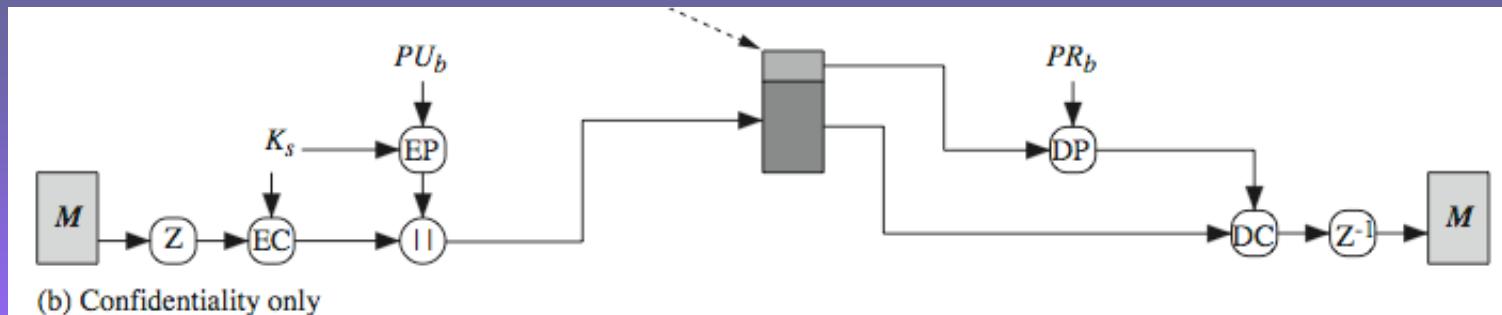
PGP Operation – Authentication

1. sender creates message
2. make SHA-1160-bit hash of message
3. attached RSA signed hash to message
4. receiver decrypts & recovers hash code
5. receiver verifies received message hash



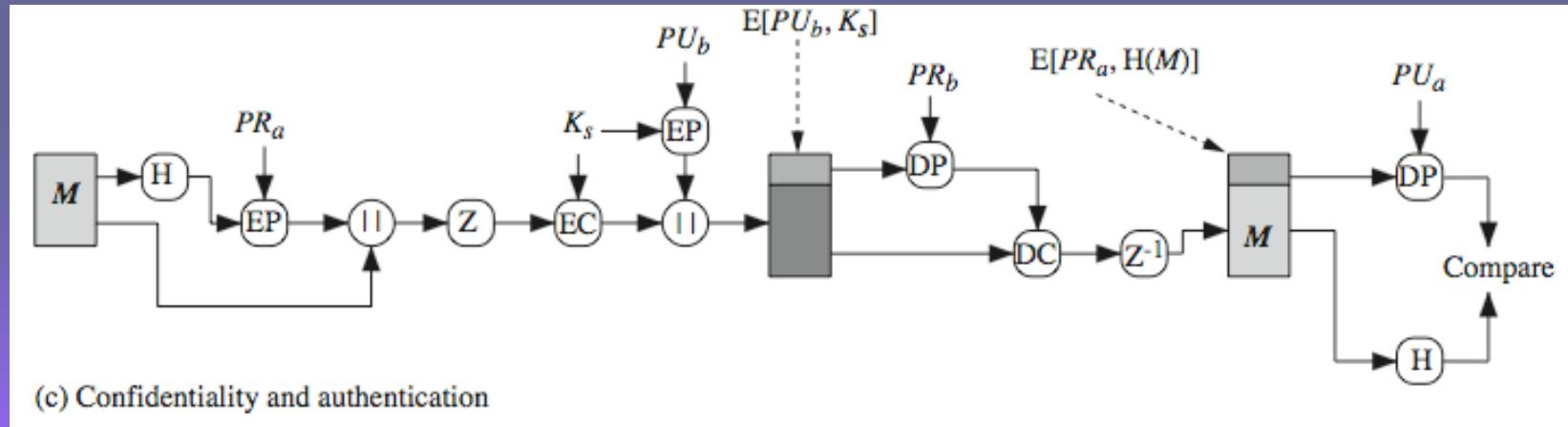
PGP Operation – Confidentiality

1. sender forms 128-bit random session key
2. encrypts message with session key
3. attaches session key encrypted with RSA
4. receiver decrypts & recovers session key
5. session key is used to decrypt message



PGP Operation – Confidentiality & Authentication

- can use both services on same message
 - create signature & attach to message
 - encrypt both message & signature
 - attach RSA/EIGamal encrypted session key



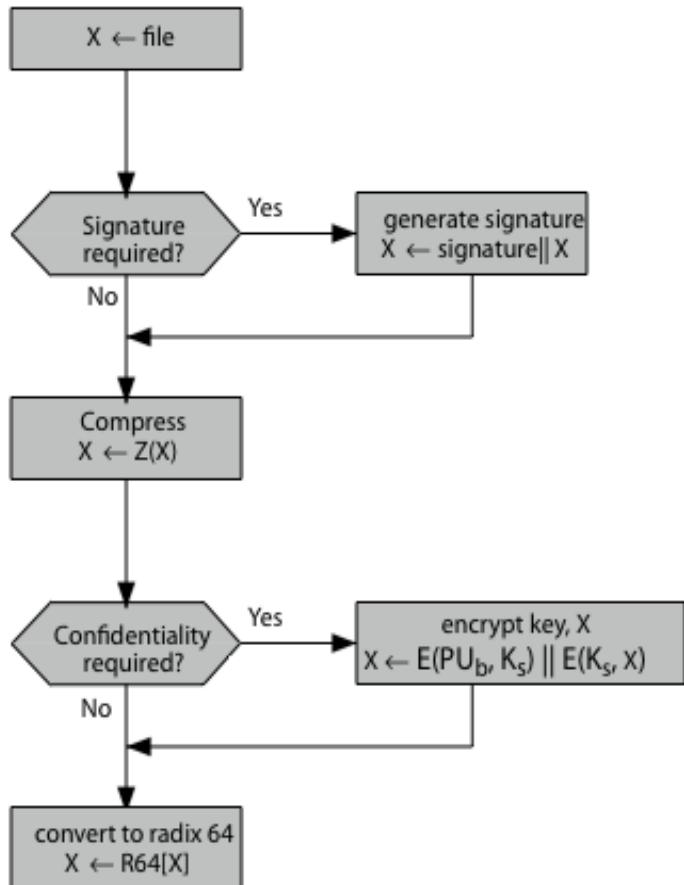
PGP Operation – Compression

- by default PGP compresses message after signing but before encrypting
 - so can store uncompressed message & signature for
 - uses ZIP compression algorithm

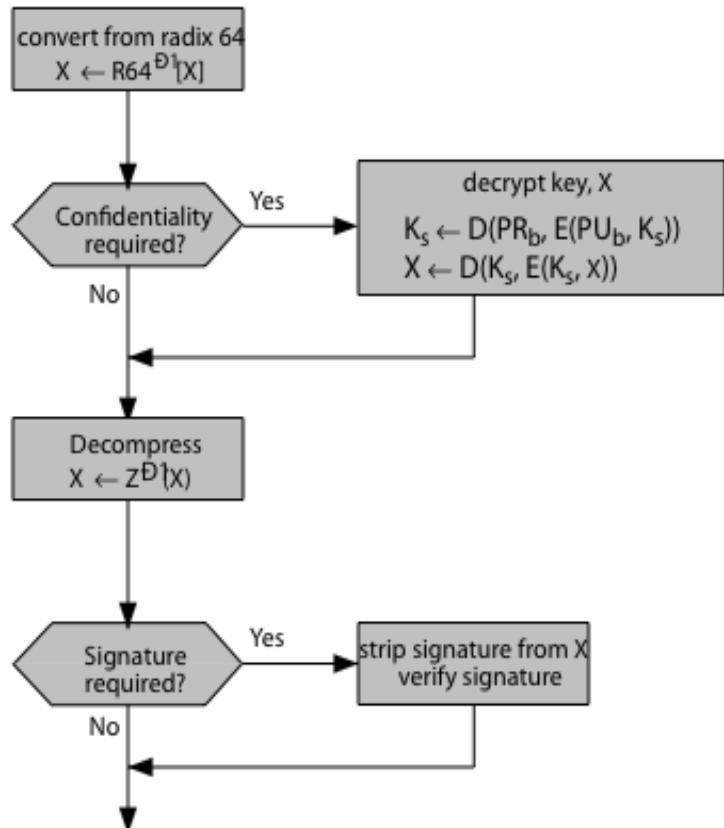
PGP Operation – Email Compatibility

- 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
- Radix-64 divides the binary data into 24 bit blocks.
- uses radix-64 algorithm
 - maps 3 bytes to 4 printable chars
 - also appends a CRC
- PGP also segments messages if too big

PGP Operation – Summary



(a) Generic Transmission Diagram (from A)



(b) Generic Reception Diagram (to B)

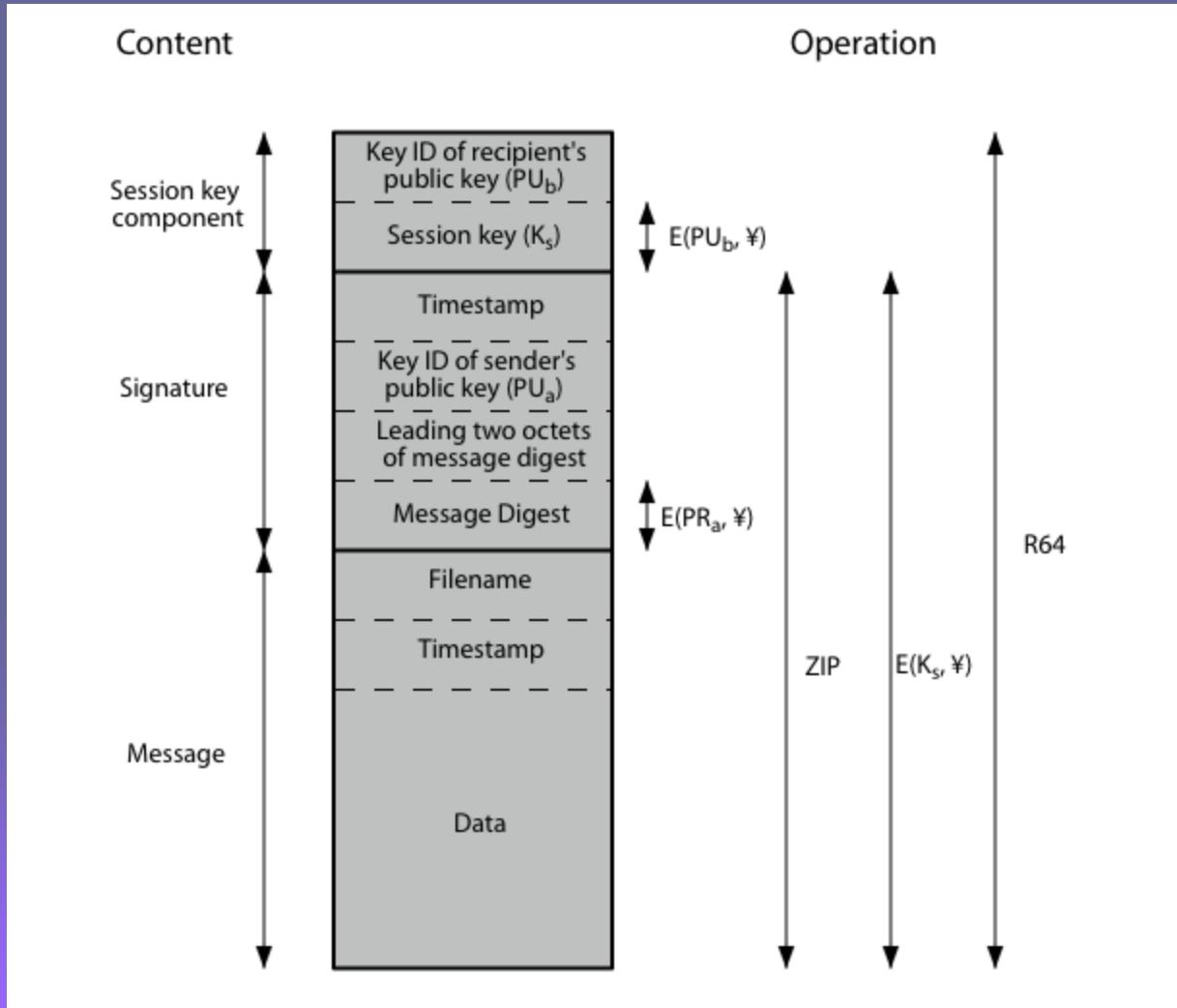
PGP Session Keys

- 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 Message Format



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
- security of private keys thus depends on the pass-phrase security

PGP Key Rings

Private Key Ring

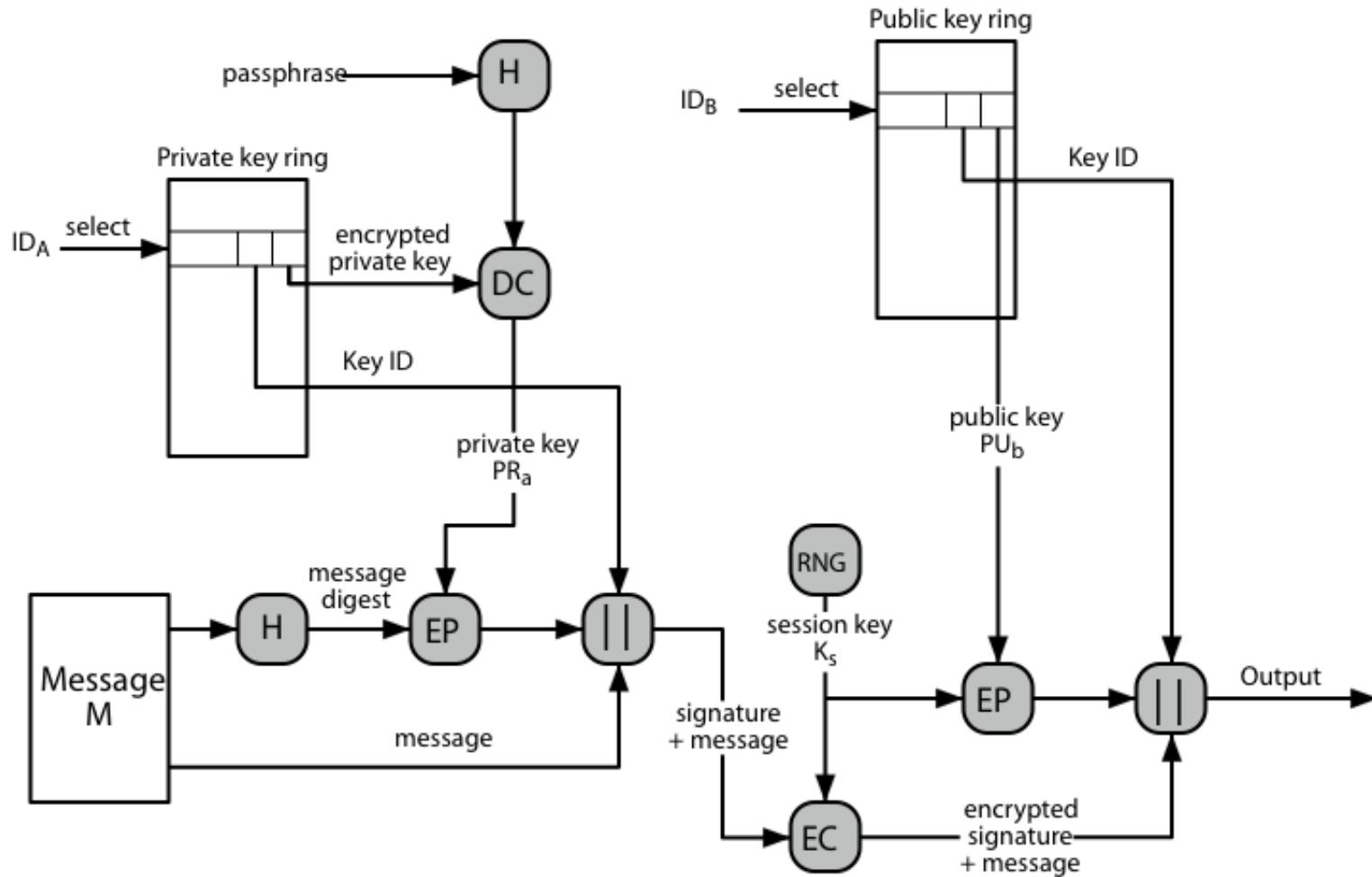
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(PU_i), PR_i)$	User i
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮

Public Key Ring

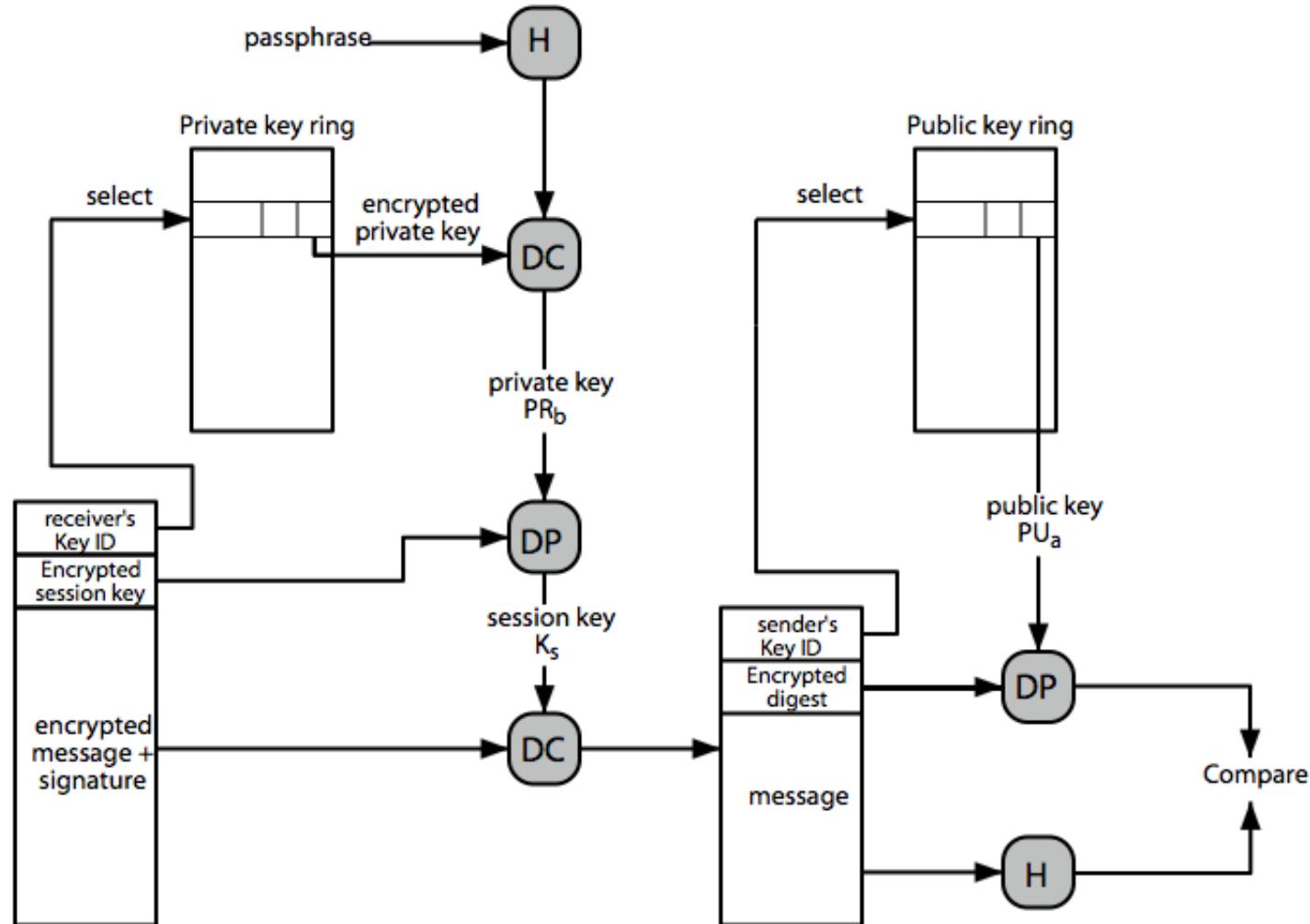
Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_i	$PU_i \bmod 2^{64}$	PU_i	$trust_flag_i$	User i	$trust_flag_i$		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

* = field used to index table

PGP Message Generation



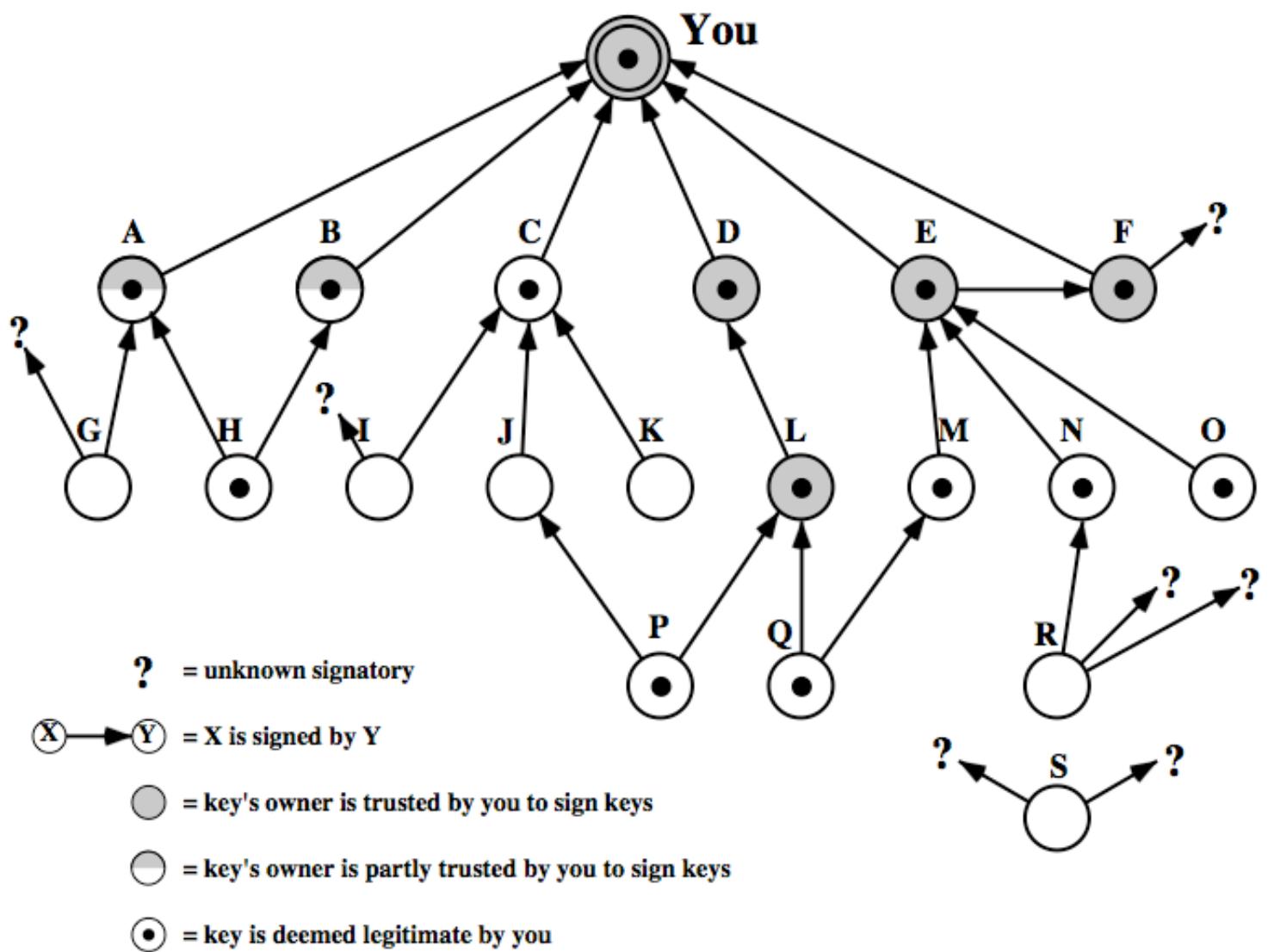
PGP Message Reception



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 Trust Model Example



S/MIME (Secure/Multipurpose Internet Mail Extensions)

- security enhancement to MIME email
 - original Internet RFC822 email was text only
 - MIME provided support for varying content types and multi-part messages
 - with encoding of binary data to textual form
 - S/MIME added security enhancements
- have S/MIME support in many mail agents
 - eg MS Outlook, Mozilla, Mac Mail etc

S/MIME Functions

- enveloped data
 - encrypted content and associated keys
- signed data
 - encoded message + signed digest
- clear-signed data
 - cleartext message + encoded signed digest
- signed & enveloped data
 - nesting of signed & encrypted entities

S/MIME Cryptographic Algorithms

- digital signatures: DSS & RSA
- hash functions: SHA-1 & MD5
- session key encryption: ElGamal & RSA
- message encryption: AES, Triple-DES, RC2/40 and others
- MAC: HMAC with SHA-1
- have process to decide which algs to use

S/MIME Messages

- S/MIME secures a MIME entity with a signature, encryption, or both
- forming a MIME wrapped PKCS object
- have a range of content-types:
 - enveloped data
 - signed data
 - clear-signed data
 - registration request
 - certificate only message

MIME Header

- The five header fields defined in MIME are as follows:
- **MIME-Version: Must have the parameter value 1.0. This field indicates that the message** conforms to RFCs 2045 and 2046.
- **Content-Type: Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner.**
- **Content-Transfer-Encoding: Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.**
- **Content-ID: Used to identify MIME entities uniquely in multiple contexts.**
- **Content-Description: A text description of the object with the body; this is useful when the object is not readable (e.g., audio data).**

MIME Content Types

Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
Message	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
	rfc822	The body is itself an encapsulated message that conforms to RFC 822.
	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
Image	External-body	Contains a pointer to an object that exists elsewhere.
	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript.
	octet-stream	General binary data consisting of 8-bit bytes.

S/MIME Certificate Processing

- S/MIME uses X.509 v3 certificates
- managed using a hybrid of a strict X.509 CA hierarchy & PGP's web of trust
- each client has a list of trusted CA's certs
- and own public/private key pairs & certs
- certificates must be signed by trusted CA's

Certificate Authorities

- have several well-known CA's
- Verisign one of most widely used
- Verisign issues several types of Digital IDs
- increasing levels of checks & hence trust

Class	Identity Checks	Usage
1	name/email check	web browsing/email
2	+ enroll/addr check	email, subs, s/w validate
3	+ ID documents	e-banking/service access

S/MIME Enhanced Security Services

- 3 proposed enhanced security services:
 - signed receipts
 - security labels
 - secure mailing lists

Summary

- have considered:
 - secure email
 - PGP
 - S/MIME
 - domain-keys identified email