MONASH University
Information Technology

FIT3031 INFORMATION & NETWORK SECURITY

MONASH University
Information Technology

FIT3031 INFORMATION & NETWORK SECURITY

Lecture 7:
Electronic Mail Security

Unit Structure: Lecture Topics

- ✓ OSI security architecture
  - **common security standards and protocols for network security applications**
  - **common information risks and requirements**
- ✓ operation of private key encryption techniques
- ✓ operation of public encryption techniques
- ✓ concepts and techniques for digital signatures, authentication and non-repudiation
- ✓ security threats of web servers, and their possible countermeasures
- ✓ Wireless Network Security Issues
- ✓ security threats of email systems and their possible countermeasures
- IP security
- intrusion detection techniques for security purpose
- risk of malicious software, virus and worm threats, and countermeasures
- firewall deployment and configuration to enhance protection of information assets
- network management protocol for security purpose

Review of Last Lecture

- **have considered:**
  - IEEE 802.11 Wireless LANs
    - > protocol overview and wireless security mechanisms
  - Wireless Application Protocol (WAP)
    - > protocol overview
  - Wireless Transport Layer Security (WTLS)
  - end-to-end security between wireless clients and network servers

Lecture 7: Objectives

- **Understand the security issues associated with <u>email security</u>**
- **Be familiar with secure email standards**
- **Understand the operation of Pretty Good Privacy (PGP)**
- **Understand how cryptographic techniques are applied to secure email communications**
- **Understand the operation of Secure MIME**
- **Understand the basic operation of DKIM**

Lecture 7: Outline

- **Electronic Mail Security**
- **Pretty Good Privacy (PGP)**
  - PGP Services
  - PGP Message Format
  - PGP Message Generation and Reception
- **Secure MIME**
- **DomainKeys Identified Mail (DKIM)**

Electronic Mail

- **Perhaps the most widely used network-based application**
  - vital for business operation as well as home users
  - organizations use e-mail for internal official communication and also for communicating with external customers
- **Currently all message contents are not secure**
  - may be inspected either in transit
  - or by suitably privileged users on destination system
  - abuse of e-mail system is increasing
  - a small change in financial data or invoice, can have disastrous consequences
  - email can be forged easily

Electronic Mail

- **The basis for email over the Internet**
  - Simple Mail Transfer **Protocol** (SMTP specified in RFC-821 standard)
  - Message **syntax** (specified in RFC-822 standard)
  - Multipurpose Internet Mail **Extension** (MIME specified in RFC 2045-2049)
- ***Neither SMTP nor the message syntax supports security services***
- **So we need to look at securing MIME**

Electronic Mail Security

- **Required security properties:**
  - **confidentiality**
    - > protection from disclosure
  - **authentication**
    - > of sender of message
  - **message integrity**
    - > protection from modification
  - **non-repudiation of origin**
    - > protection from denial by sender
- **Two main schemes for email security**
  - **PGP and**
  - **S/MIME**

Pretty Good Privacy (PGP)

- **PGP was developed by Philip R. Zimmerman**
- **PGP provides confidentiality and authentication services that can be used for electronic mail and file storage applications**
- **Availiable free on a variety of platforms**
  - *a commercial version is also available*
- **Based on well known algorithms**
- **Wide range of applicability**
- ***Not developed or controlled by governmental or standards organizations***

PGP Services

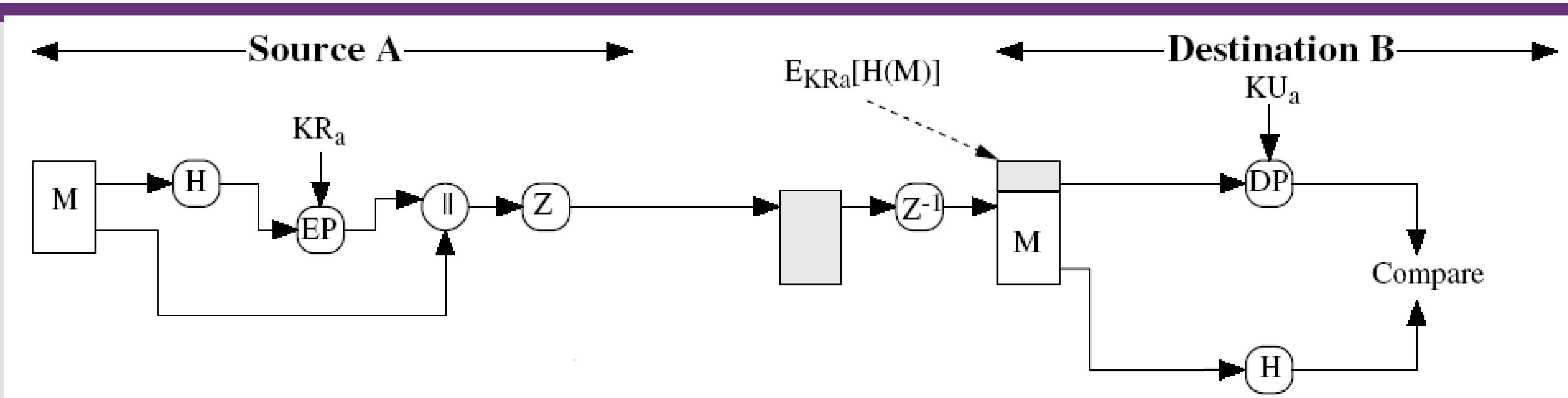
PGP offers five services: Authentication, Confidentiality, Compression, E-mail compatibility & Segmentation

	Function	Algorithms Used	Description
Authentication + Integrity	Digital signature	DSS/SHA 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.
Confidentiality	Message encryption	CAST, IDEA, TDES + DH 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.
Compression	Compression	ZIP	A message may be compressed, for storage or transmission, using ZIP.
E-Mail Compatibility	Email compatibility	RADIX 64	To provide transparency for email applications, an encrypted message may be converted to an ASCII string using radix 64 conversion.
Segmentation to support message size limitations	Segmentation	Segmentation standardized	To accommodate maximum message size limitations, PGP performs segmentation and reassembly.

PGP Operation - **Authentication**

- **Sender creates a message**
- **SHA-1 is used to generate 160-bit hash code of message**
- **Hash code is encrypted with RSA using the sender's private key, and the result *precedes* the message**
- **Receiver uses RSA or DSS (Digital Signature Standard) with sender's public key to decrypt and recover the hash code**
- **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**

PGP Operation – **Authentication**…

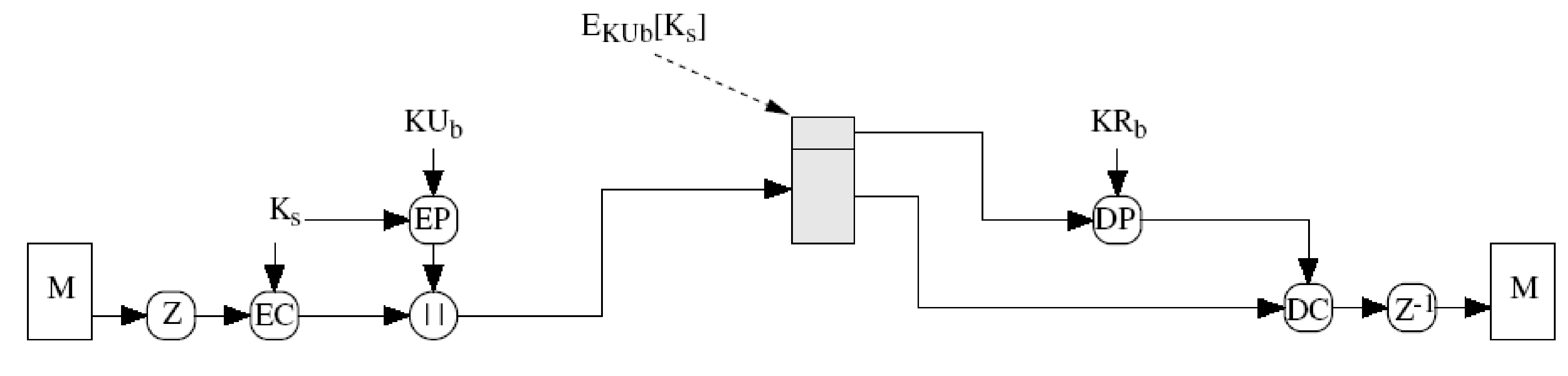


- (KR_a and KU_a) are private-public key pair of user A
- H ➔ SHA-1 (160 bit) hash algorithm
- EP, DP, Z ➔ encryption, decryption and compression operation respectively
- Detached signatures are also supported
  - detached signature of an executable program can detect virus infection
  - is useful when more than one party must sign a document, e.g., legal contract

PGP - Confidentiality

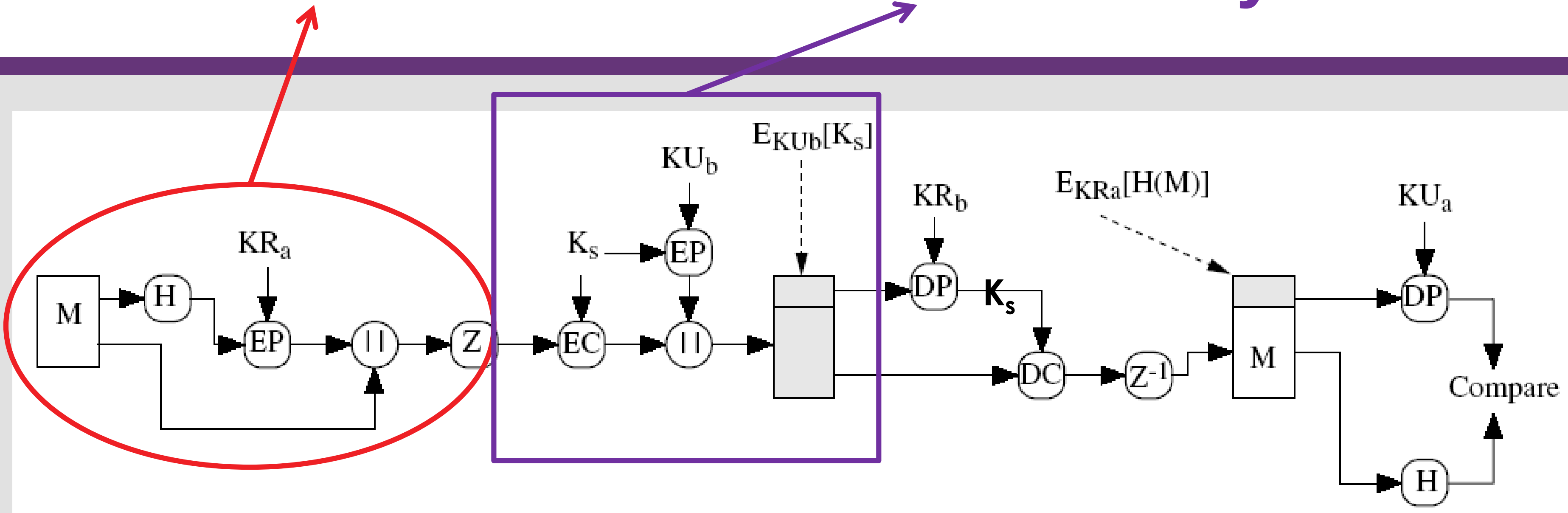
- Sender generates a message and random 128-bit number is used as session key
  - > session key is valid for this message only
- Message is encrypted, using CAST-128 / IDEA /3DES with the session key
- The session key is encrypted using RSA with recipient's public key, which then precedes the encrypted message
- Receiver uses RSA with its private key to decrypt and recover the session key
- Session key is used to decrypt the message

PGP Operation – Confidentiality…



- EC, DC are symmetric encryption and decryption respectively, while EP, DP Asymmetric algorithms
- PGP offers an option to use a variant of Diffie-Hellman known as El-Gamal.
- The session key is distributed with asymmetric encryption
  - in practice it is an one-time key for each message
  - uses random inputs is taken from previous uses and from keystroke timing of user
- The message is encrypted with symmetric encryption to reduce encryption/decryption time
- The encrypted session key is bound to the message
  - no need for a session key exchange protocol
  - arrives with the message

PGP-Authentication & Confidentiality



- **Uses both services on the same message:**
  - The sender first signs the message with its own private key
  - Encrypts zipped (message + signature) with the session key
  - Encrypts the session key with the recipient's public key
  - and the encrypted key precedes the rest

PGP - Compression

- **Uses ZIP compression algorithm**
- **Placement of compression is important**
- **Compresses message after applying the signature but before encryption:**
  - The sender needs to store only the uncompressed message and the signature for future verification
    - Otherwise, compressed message needs to be stored as well
  - There are different compression algorithms and different versions of the same algorithm
    - If compression is done after encryption, all PGP implementation must use the same version of the same algorithm
  - It strengthens the security; as cryptanalysis on compressed message is more difficult

PGP – Email Compatibility

- **When using PGP, binary data will be sent (encrypted message etc.)**
- ***<u>However many email systems only permit ASCII text</u>***
- **Hence PGP must encode raw binary data into printable ASCII characters**
- **PGP overcomes this problem by using radix-64 algorithm**
  - maps 3 bytes (24bit) to 4 printable chars (32bit)
  - **expands the message by 33% but it is compensated by compression to offset**
  - also appends a CRC

PGP – Email Compatibility

Table:
Base64 or
Radix-64 encoding or
ASCII armour

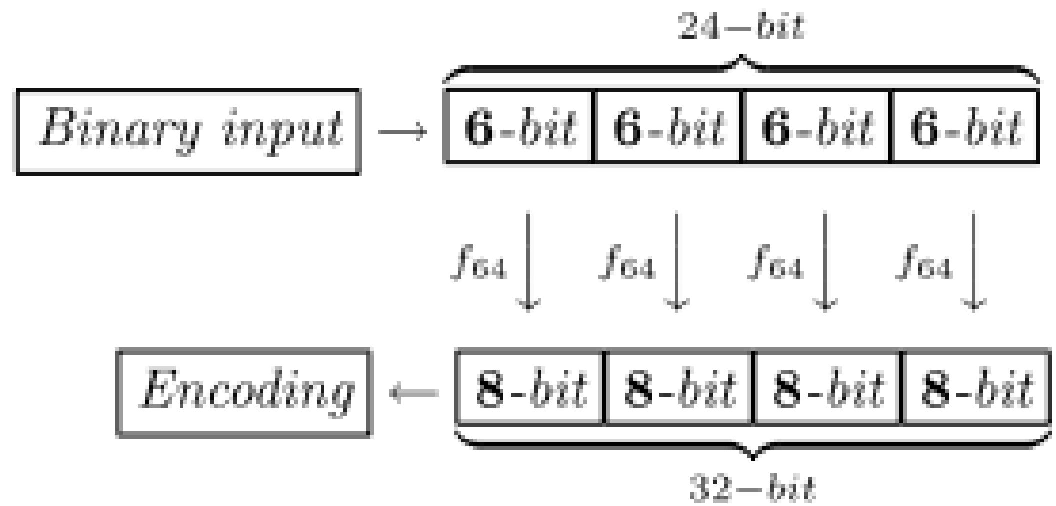
Binary	ASCII	Binary	ASCII	Binary	ASCII	Binary	ASCII
000000	A	010000	Q	100000	g	110000	w
000001	B	010001	R	100001	h	110001	x
000010	C	010010	S	100010	i	110010	y
000011	D	010011	T	100011	j	110011	z
000100	E	010100	U	100100	k	110100	0
000101	F	010101	V	100101	l	110101	1
000110	G	010110	W	100110	m	110110	2
000111	H	010111	X	100111	n	110111	3
001000	I	011000	Y	101000	o	111000	4
001001	J	011001	Z	101001	p	111001	5
001010	K	011010	a	101010	q	111010	6
001011	L	011011	b	101011	r	111011	7
001100	M	011100	c	101100	s	111100	8
001101	N	011101	d	101101	t	111101	9
001110	O	011110	e	101110	u	111110	+
001111	P	011111	f	101111	v	111111	/

Radix-64 Conversion

Table G.1

6-bit Input	0	1	2	3	4	5	6	7	8	9	10
Encoding	A	B	C	D	E	F	G	H	I	J	K
6-bit Input	11	12	13	14	15	16	17	18	19	20	21
Encoding	L	M	N	O	P	Q	R	S	T	U	V
6-bit Input	22	23	24	25	26	27	28	29	30	31	32
Encoding	W	X	Y	Z	a	b	c	d	e	f	g
6-bit Input	33	34	35	36	37	38	39	40	41	42	43
Encoding	h	i	j	k	l	m	n	o	p	q	r
6-bit Input	44	45	46	47	48	49	50	51	52	53	54
Encoding	s	t	u	v	w	x	y	z	0	1	2
6-bit Input	55	56	57	58	59	60	61	62	63	PAD	
Encoding	3	4	5	6	7	8	9	+	/	=	

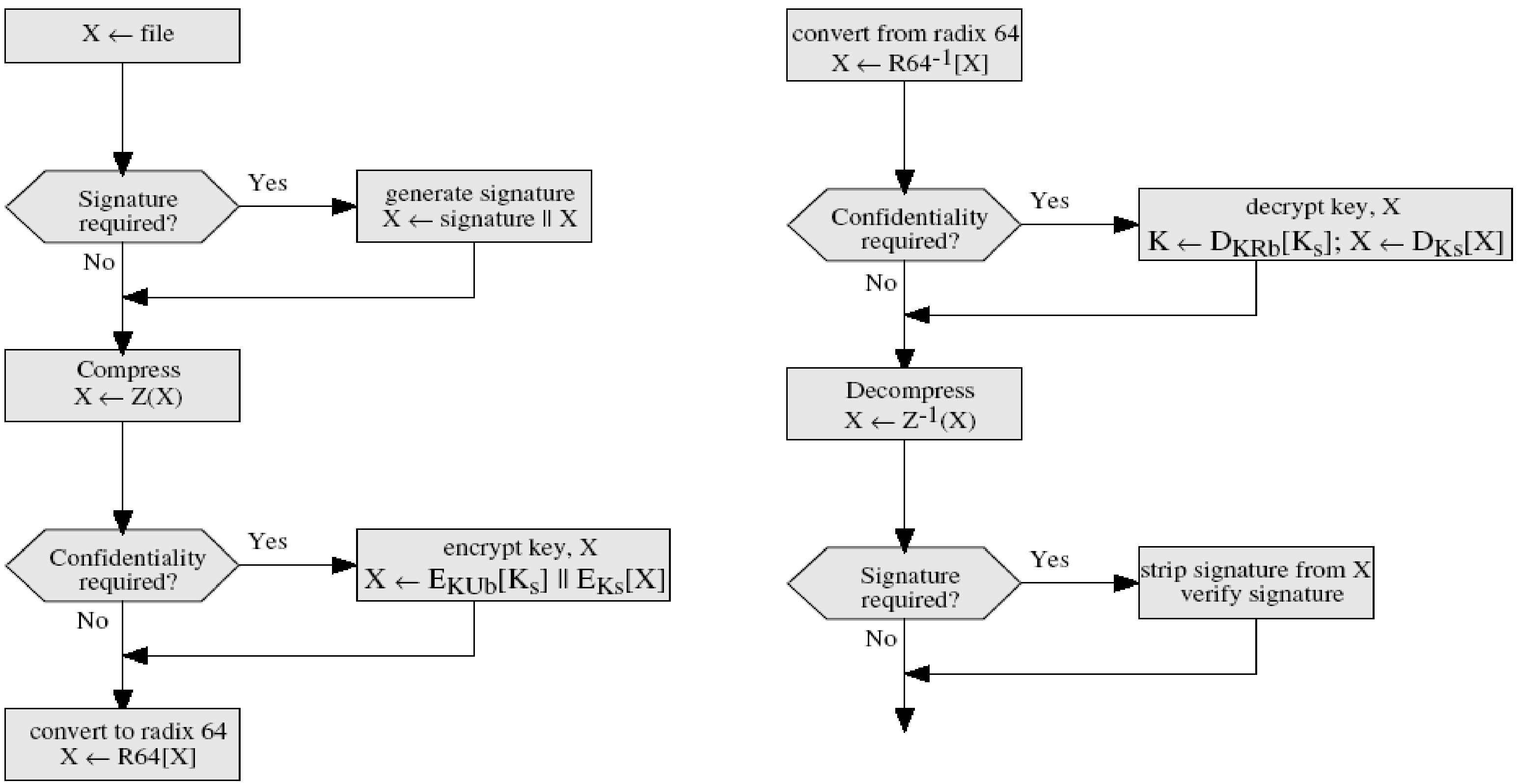
Diagram G.1 Radix-64 Encoding Illustration



PGP - Segmentation & Reassembly

- **Most email systems are restricted to a maximum message length**
- **To handle a large message, PGP**
  - breaks it down into smaller segments
  - then reassembles those segments at the receiving end
- **Segmentation is done after all other processing including radix-64 conversion**
- **Only the first segment carries the session key and signature component**
  - all other segments carry the email header
  - at the receiving end, PGP strips off all headers and reassembles the segments

PGP Summary



(a) Generic Transmission Diagram (from A)

(b) Generic Reception Diagram (to B)

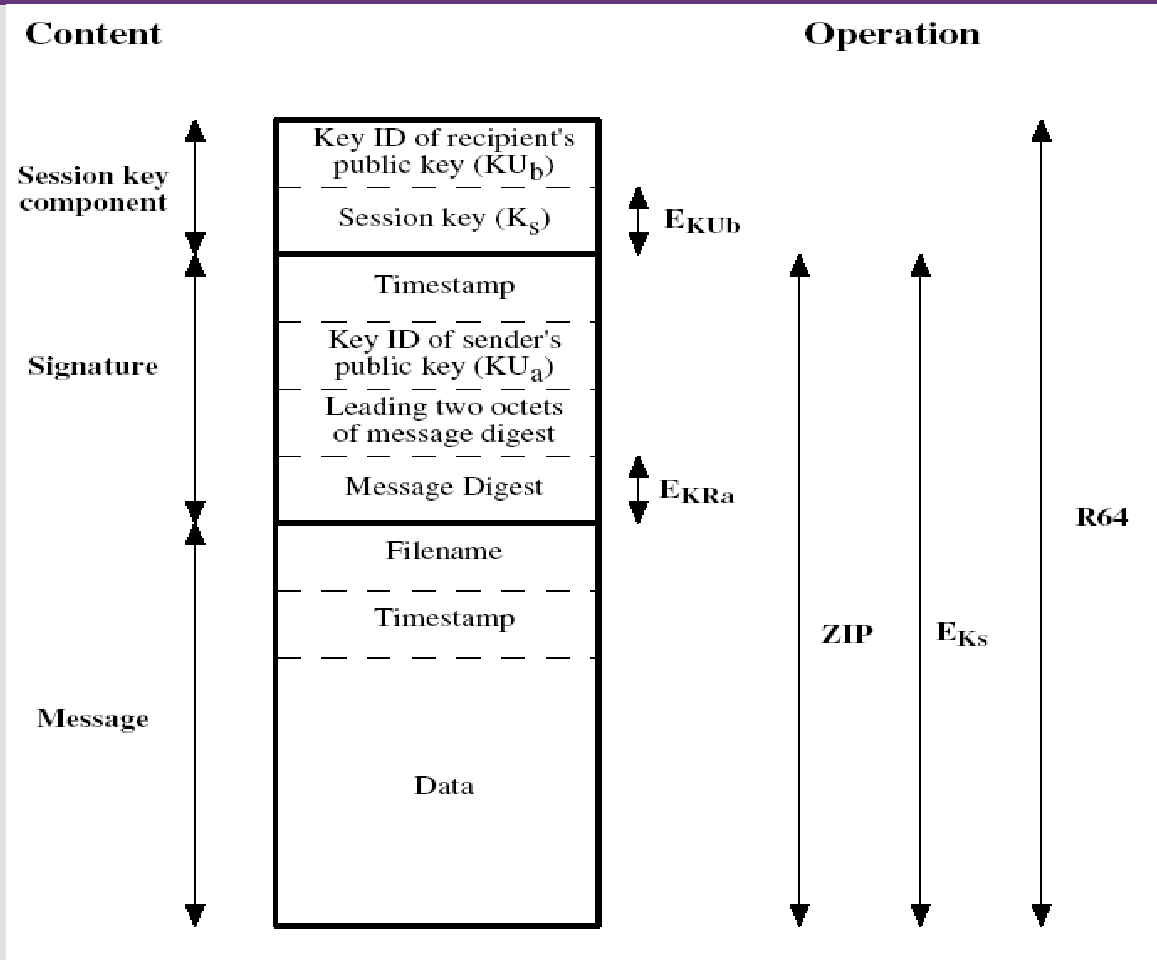
PGP Session Keys

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

PGP Public & Private Keys (Asymmetric pairs)

- **A user may have multiple key pairs**
  - may change key from time to time
  - different keys to interact with different people
- **The recipient needs to identify which one is actually being used to encrypt the 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
  - Key ID of (public key, KU_a) = $KU_a \bmod 2^{64}$
- **also use key ID in PGP digital signatures**

PGP Message Format



PGP Key Rings (1)

- **Each PGP user has a pair of key rings:**
  - **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 with key from a hashed passphrase
    - while generating public/private key pairs PGP asks for a passphrase to the user
    - a 160-bit hash code is generated using SHA-1 of the passphrase; then the passphrase is discarded
    - Encrypts the private key with 128 bit of the hash code using CAST-128 and hash code is discarded
    - User must supply passphrase to retrieve his/her private key

PGP Key Rings (2)

- PGP includes a facility for assigning a level of trust to individual signers and to keys.
- PGP computes a **"key legitimacy field"** for each public key certificate in the key ring. The higher the trust level, the higher the confidence the certificate is authentic.

PGP uses two trust fields to maintain key legitimacy:
[1] **Signature trust field**:- indicates the degree to which the PGP user trusts the signer to certify public keys
[2] **Owner trust field:-** indicates the degree to which this public key is trusted to sign other public key certificates

Private Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
⋮	⋮	⋮	⋮	⋮
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{H(P_i)}[KR_i]$	User i
⋮	⋮	⋮	⋮	⋮

Public Key Ring

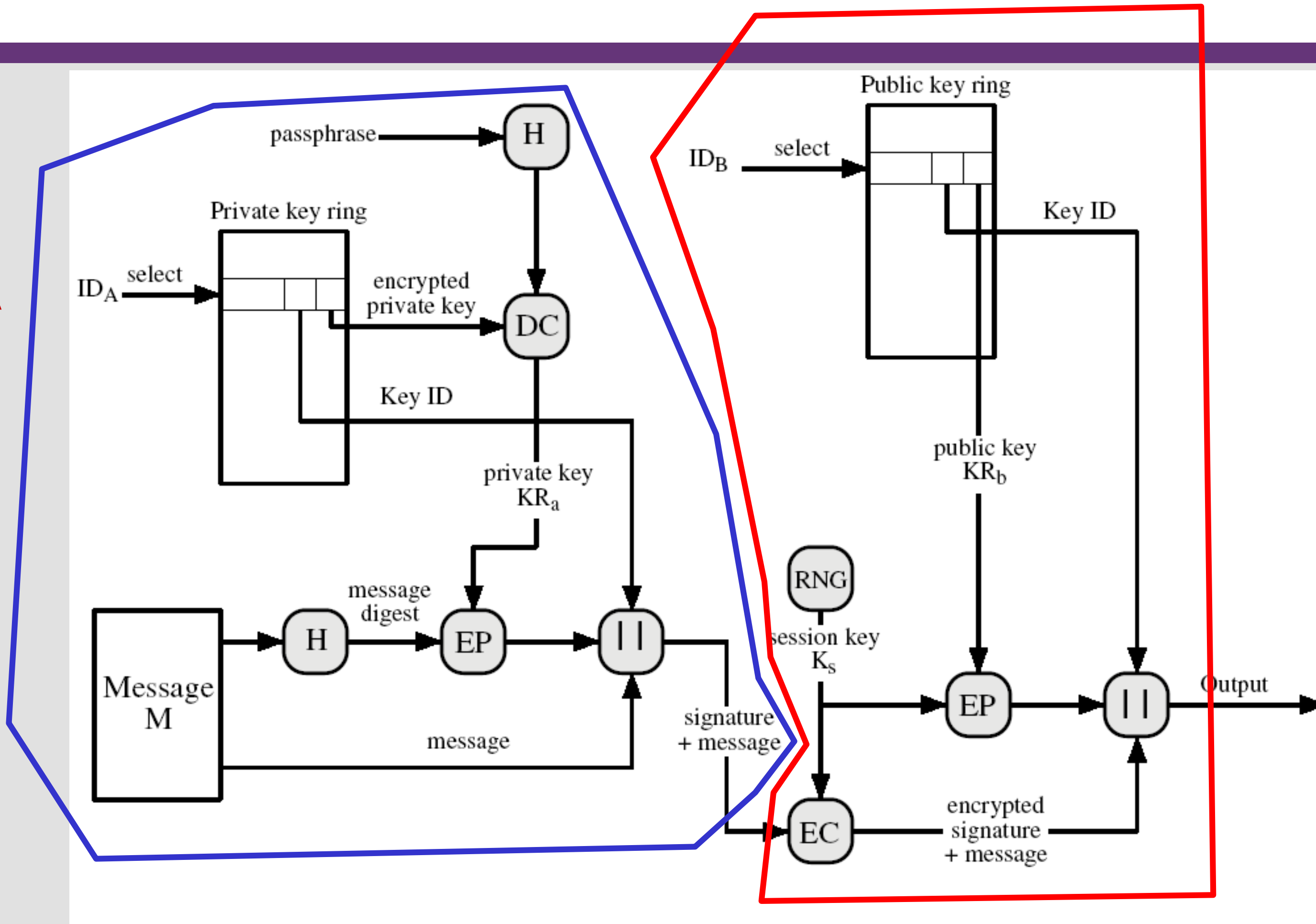
Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_i	$KU_i \bmod 2^{64}$	KU_i	$trust\_flag_i$	User i	$trust\_flag_i$		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

PGP Message Generation (1)

PGP Message Generated by User A to User B: using key rings

For simplicity sake!

No compression
And
No radix-64



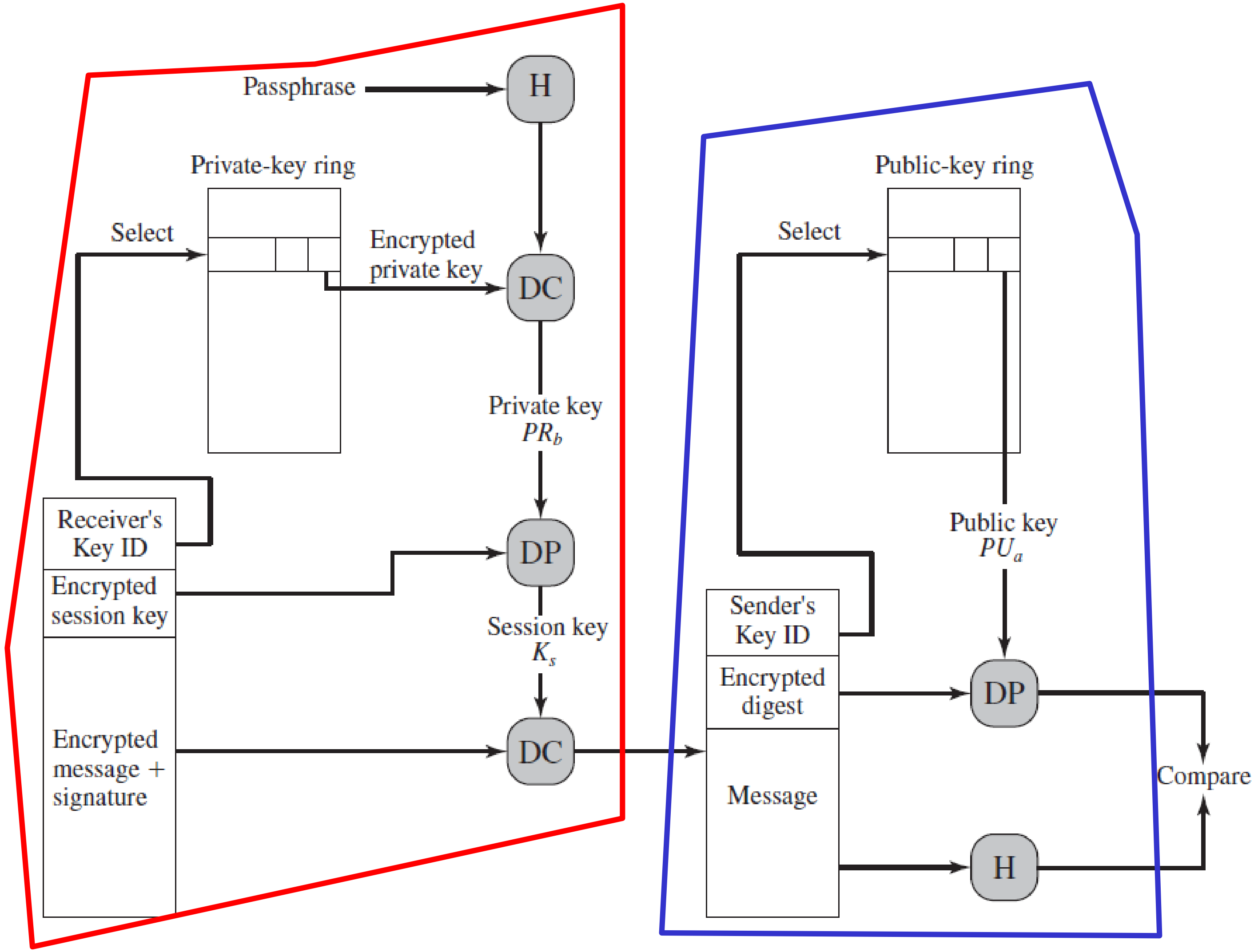
PGP Message Generation (from User A to User B; no compression or radix 64 conversion)

PGP Message Reception

PGP Message Reception by User B from User B: using key rings

For simplicity sake!

No compression
And
No radix-64

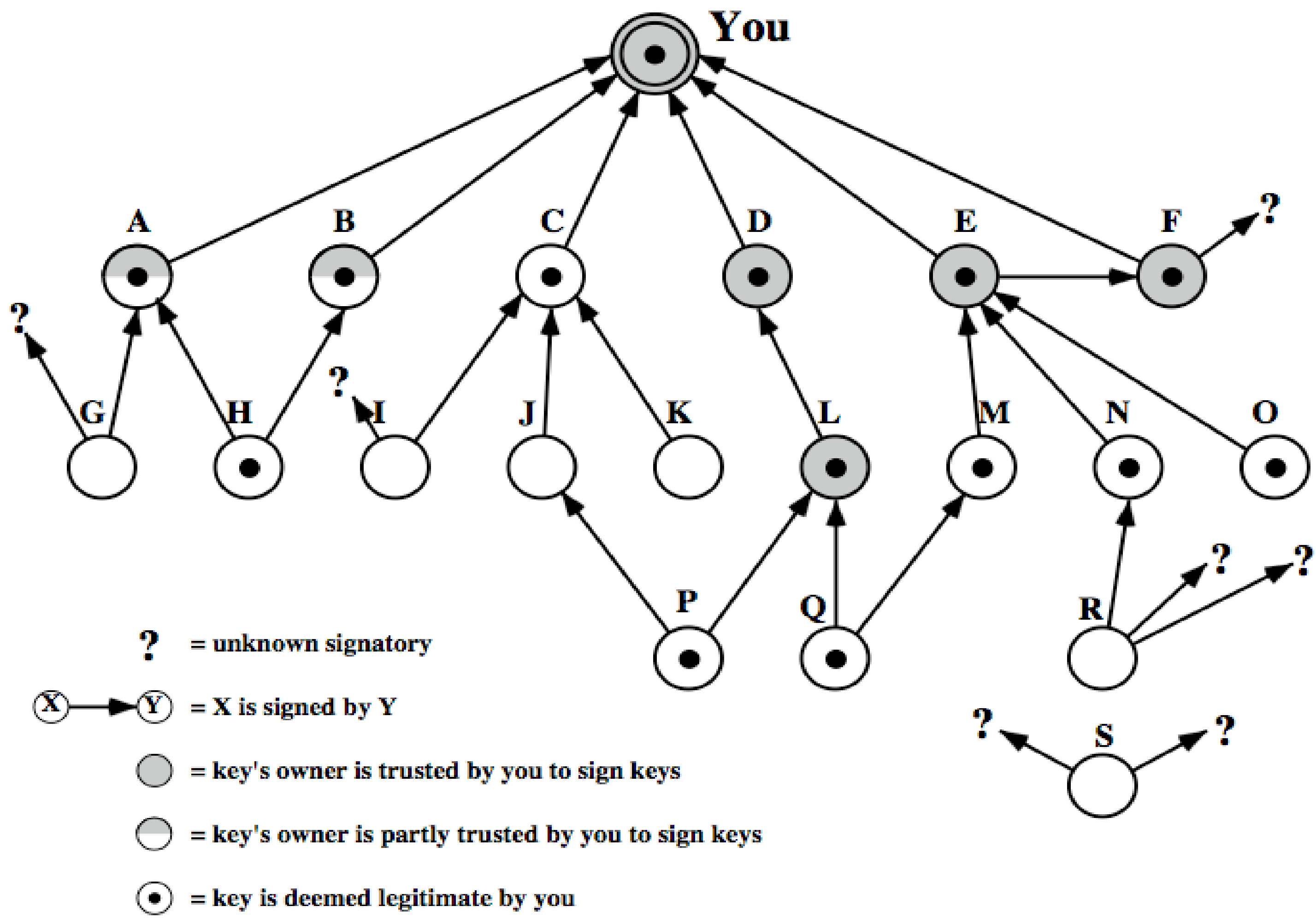


PGP Public Key Management

- **Rather than relying on certificate authorities, in PGP every user is one's own CA**
  – can sign keys for users they know directly
- **Forms a "web of trust"**
  – trusted keys will be signed
  – can trust keys others have signed if there is chain of signatures to them
- **Key ring includes trust indicators**
- **Users can also revoke their keys**

PGP Trust Model Example

The figure shows the structure of a public-key ring



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**
  - e.g. MS Outlook, Mozilla, Mac Mail etc.

MIME

- **SMTP has limitations - cannot transmit, or has a problem with:**
  - executable files, or other binary files (jpeg image)
  - "national language" characters (non-ASCII)
  - messages over a certain size
  - ASCII to EBCDIC translation problems
  - lines longer than a certain length (72 to 254 characters)
- **MIME provided support for varying content types and multi-part messages**
- **MIME encodes binary data to textual data**
  - multimedia applications generates data stream of arbitrary binary 8-bit pattern
  - conversion needed, 8-bit → printable characters and then re-conversion needed, printable characters → 8-bit

MIME Header Fields

- **MIME-Version:** **Must be "1.0" -> RFC 2045, RFC 2046**
- **Content-Type:** **More types being added by developers (application/word)**
- **Content-Transfer-Encoding:** **How message has been encoded (radix-64)**
- **Content-ID:** **identifies MIME entities uniquely in multiple contexts**
- **Content Description:** **Needed when content is not a readable text (e.g., mpeg)**

S/MIME Function

- **S/MIME is very similar to PGP. It also offers the ability to sign and/or encrypt messages with the following functions**
- **enveloped data**
  - encrypted content and associated keys
- **signed data**
  - encoded (message + signed digest)
- **clear-signed data**
  - clear-text message + encoded signed digest
  - recipient with MIME capability but not S/MIME capability can read the message
- **signed & enveloped data**
  - nesting of signed & encrypted entities, i.e., various orderings for encrypting and signing

S/MIME Cryptographic Algorithms

- **digital signatures: DSS & RSA**
- **hash functions: SHA-1 & MD5**
- **session key encryption: El-Gamal & RSA**
- **message encryption: AES, Triple-DES, RC2/40 and others**
- **MAC: HMAC with SHA-1**
- **have process(s) to decide which algorithms 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

S/MIME Certificate Processing

- **S/MIME uses X.509 v3 certificates**
- **managed by 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**
- **<u>certificates must be signed by trusted CA's</u>**

Certificate Authorities

- **have several well-known CA's**
- **Verisign is 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

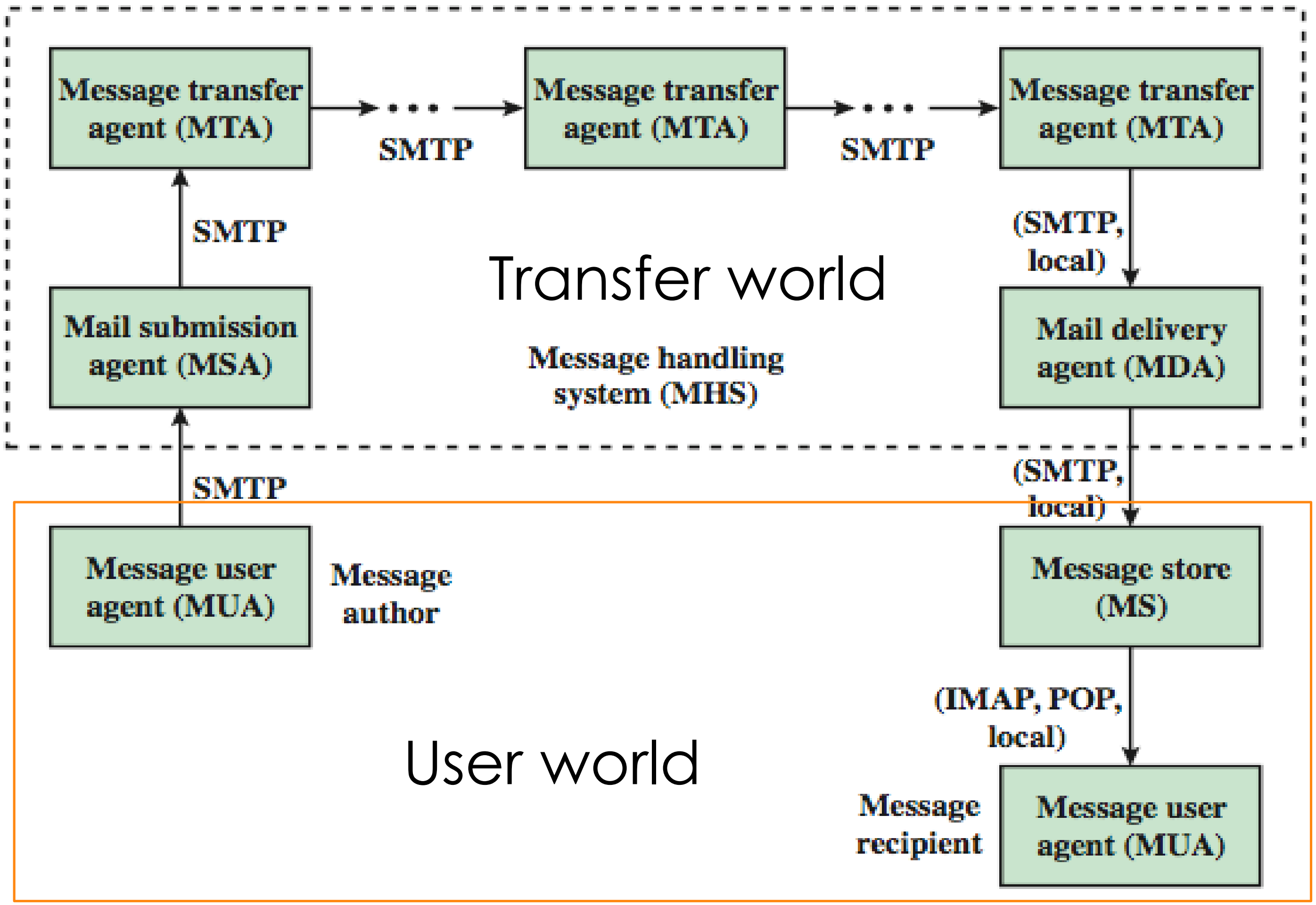
- **Three proposed enhanced security services:**
  - **signed receipts**
    - provide proof of delivery to the originator
  - **security labels**
    - Attribute regarding sensitivity of the content for access control
  - **secure mailing lists**
    - S/MIME Mail List Agent (MLA) performs each recipient specific encryption

Domain Keys Identified Mail (DKIM)

- a specification for cryptographically signing email messages
- so signing domain claims responsibility
- recipients / agents can verify signature
- proposed Internet Standard RFC 4871
- has been widely adopted

Internet Mail Architecture

Function Modules and Standardized Protocols for the Internet



MONASH University
Information Technology

Concepts

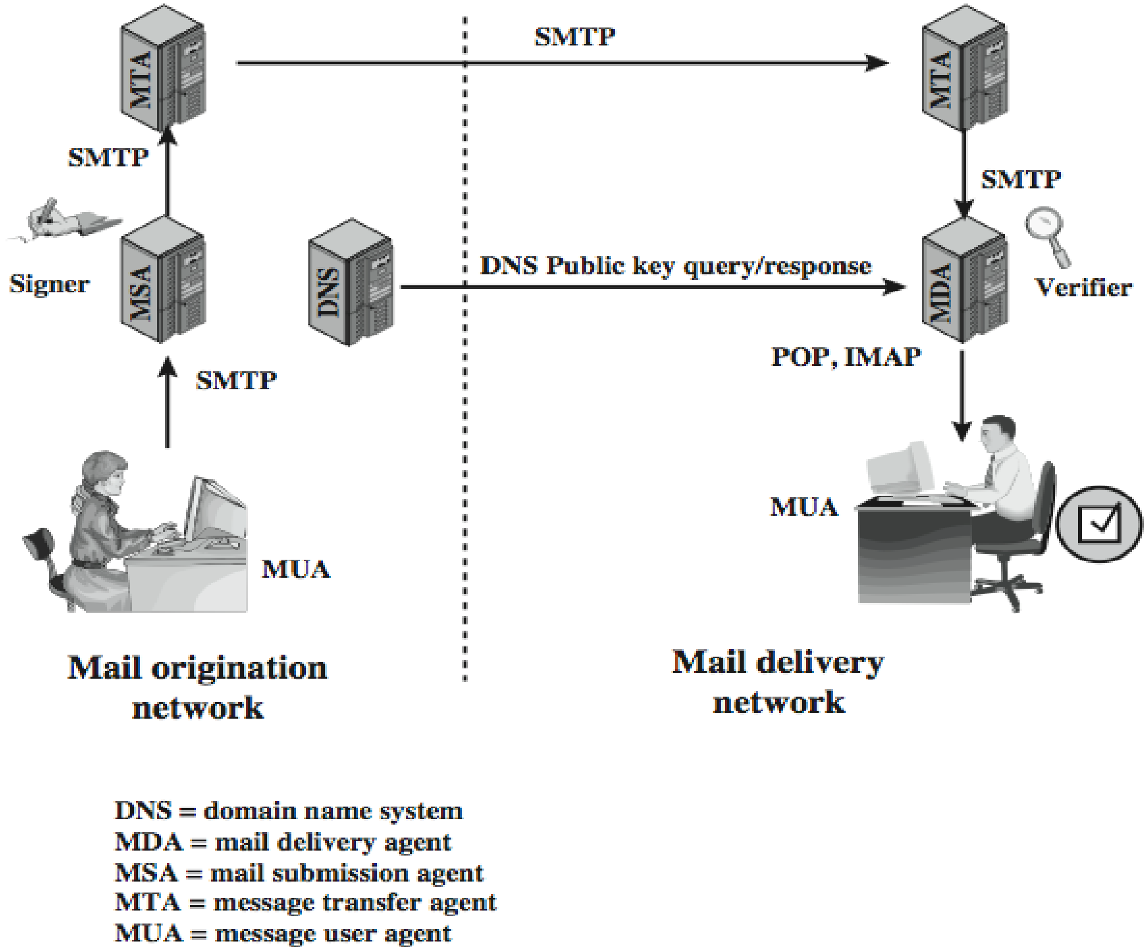
- **ADMD: Administrative Management Domain**
  - Internet email provider
    - > A department operating a local mail relay (MTA)
- **DNS: Domain Name System**
  - Directory lookup service for mapping between host name on the internet and its numerical address

Email Threats

- **see RFC 4684 - Analysis of Threats Motivating Domain Keys Identified Mail(DKIM)**
- **describes the problem space in terms of:**
  - range: low end, spammers, fraudsters
  - Capabilities: in terms of where submitted, signed, volume, routing naming etc.
  - Outside: located attackers

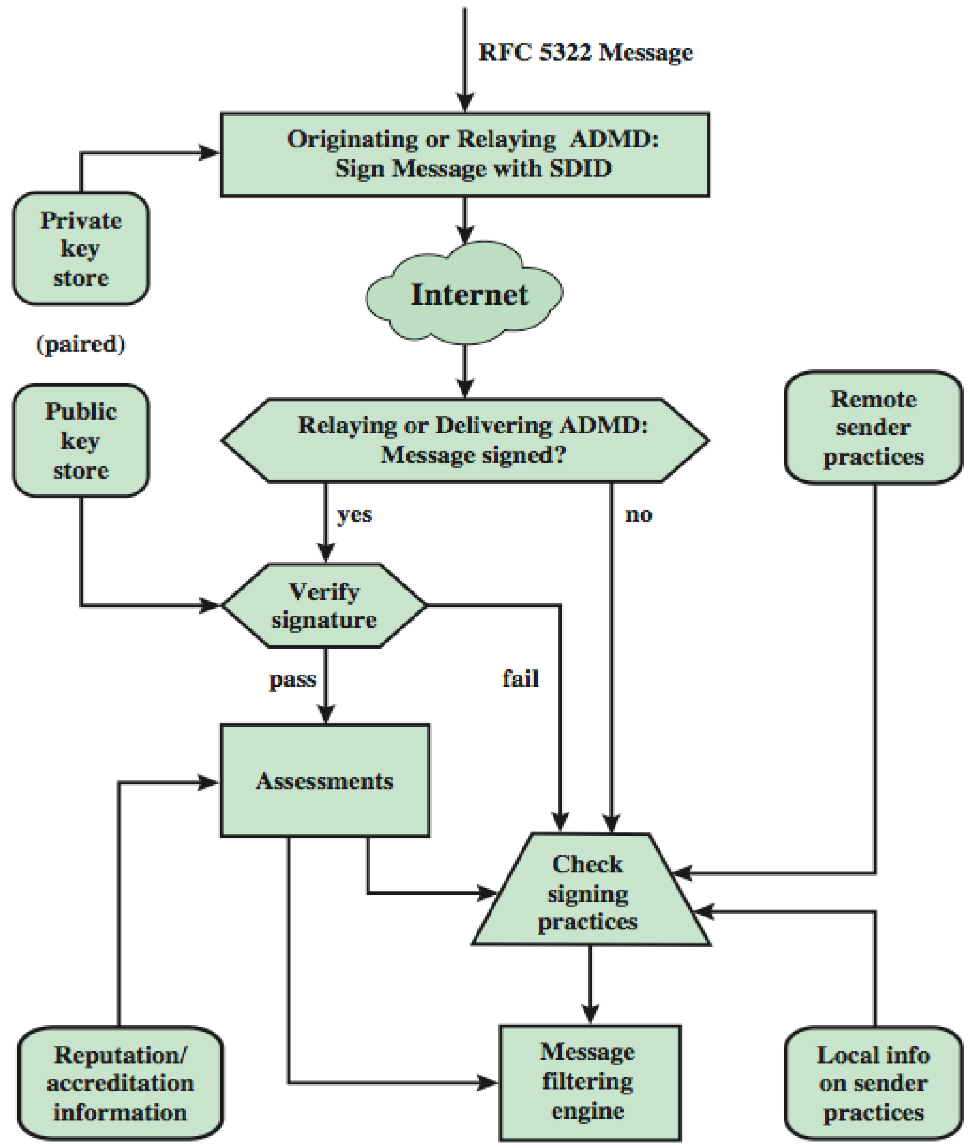
DKIM Strategy

- **transparent to user**
  - MSA(Mail Submission Agent) **sign**
  - MDA(Mail Delivery Agent) **verify**
- **for pragmatic reasons**



MONASH University
Information Technology

DKIM Functional Flow



Summary

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

Further Reading

- **Study Guide 7**
- **Chapter 7 of the textbook: Network Security Essentials- Application & Standards" by William Stallings 5th Edition, Prentice Hall, 2013**
- **Additional resources for this week**

- **Acknowledgement: part of the materials presented in the slides was developed with the help of Instructor's Manual and other resources made available by the author of the textbook.**