Chapter 5

Electronic mail security

Outline

- Pretty good privacy
- S/MIME (Secure/ Multipurpose Internet Mail Extension)
- Recommended web sites

Pretty Good Privacy

- Philip R. Zimmerman is the creator of PGP.
- PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.

About Zimmerman

Philip Zimmermann

Creator of PGP

Background

Philip R. Zimmermann is the creator of Pretty Good Privacy, an email encryption software package. Originally designed as a human rights tool, PGP was published for free on the Internet in 1991. This made Zimmermann the target of a three-year criminal investigation, because the government held that US export restrictions for cryptographic software were violated when PGP spread worldwide. Despite the lack of funding, the lack of any paid staff, the lack of a company to stand behind it, and despite government persecution, PGP nonetheless became the most widely used email encryption software in the world. After the government dropped its case in early 1996, Zimmermann founded PGP Inc. That company was acquired by Network Associates Inc (NAI) in December 1997, where he stayed on for three years as Senior Fellow. In August 2002 PGP was acquired from NAI by a new company called PGP Corporation, where Zimmermann now serves as special advisor and consultant. Zimmermann currently is consulting for a number of companies and industry organizations on matters cryptographic, and is also a Fellow at the Stanford Law School's Center for Internet and Society. He was a principal designer of the cryptographic key agreement protocol for the Wireless USB standard. His latest project is Zfone, which provides secure telephony for the Internet.

Before founding PGP Inc, Zimmermann was a software engineer with more than 20 years of experience, specializing in cryptography and data security, data communications, and real-time embedded systems. His interest in the political side of cryptography grew out of his background in military policy issues.

Zimmermann has received numerous technical and humanitarian awards for his pioneering work in cryptography. In 2003 he was included on the Heinz Nixdorf MuseumsForum Wall of Fame, and in 2001 he was inducted into the CRN Industry Hall of Fame. In 2000 InfoWorld named him one of the Top 10 Innovators in E-business. In 1999 he received the Louis Brandeis Award from Privacy International, in 1998 a Lifetime Achievement Award from Secure Computing Magazine, and in 1996 the Norbert Wiener Award from Computer Professionals for Social Responsibility for promoting the responsible use of technology. He also received the 1995 Chrysler Award for Innovation in Design, the 1995 Pioneer Award from the Electronic Frontier Foundation, the 1996 PC Week IT Excellence Award, and the 1996 Network Computing Well-Connected Award for "Best Security Product." In 1995 Newsweek named Zimmermann one of the "Net 50", the 50 most influential people on the Internet. In 2006 eWeek ranked PGP 9th in the 25 Most Influential and Innovative Products introduced since the invention of the PC in 1981.

In addition to the awards for versions of PGP developed before Zimmermann started a company, subsequent versions of PGP as refined by the company's engineering team continue to be recognized each year with many more industry awards.

Zimmermann received his bachelor's degree in computer science from Florida Atlantic University in 1978. He is a member of the International Association of Cryptologic Research, the Association for Computing Machinery, and the League for Programming Freedom. He served on the Roundtable on Scientific Communication and National Security, a collaborative project of The National Academies and The Center for Strategic and International Studies. He is Chairman of the OpenPGP Alliance, serves on the Board of Directors for Computer Professionals for Social Responsibility, and is on the Advisory Boards for Santa Clara University's Computer Engineering Department, Anonymizer.com, Hush Communications, Debix, and Qualys.

Zimmermann can be reached by email at prz at mit dot edu, or by phone by clicking here.

In 2012, Zimmermann was inducted into the Internet Hall of Fame by the Internet Society.

Why Is PGP Popular?

- It is available free on a variety of platforms.
- Based on well known algorithms.
- Wide range of applicability
- Not developed or controlled by governmental or standards organizations

Summary of PGP Services

Function	Algorithms Used	Description
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.
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.

Terms

- K_S : Session Key
- KR_a : Private key of A
- KU_a : Public key of A
- ER : Encryption using RSA
- DR : Decryption using RSA
- EI : IDEA encryption
- DI : IDEA decryption
- H : hashing (using SHA-1)
- || : concatenation
- Z : ZIP compression
- R64 : radix 64 ASCII Format

IDEA: International Data Encryption Algorithm

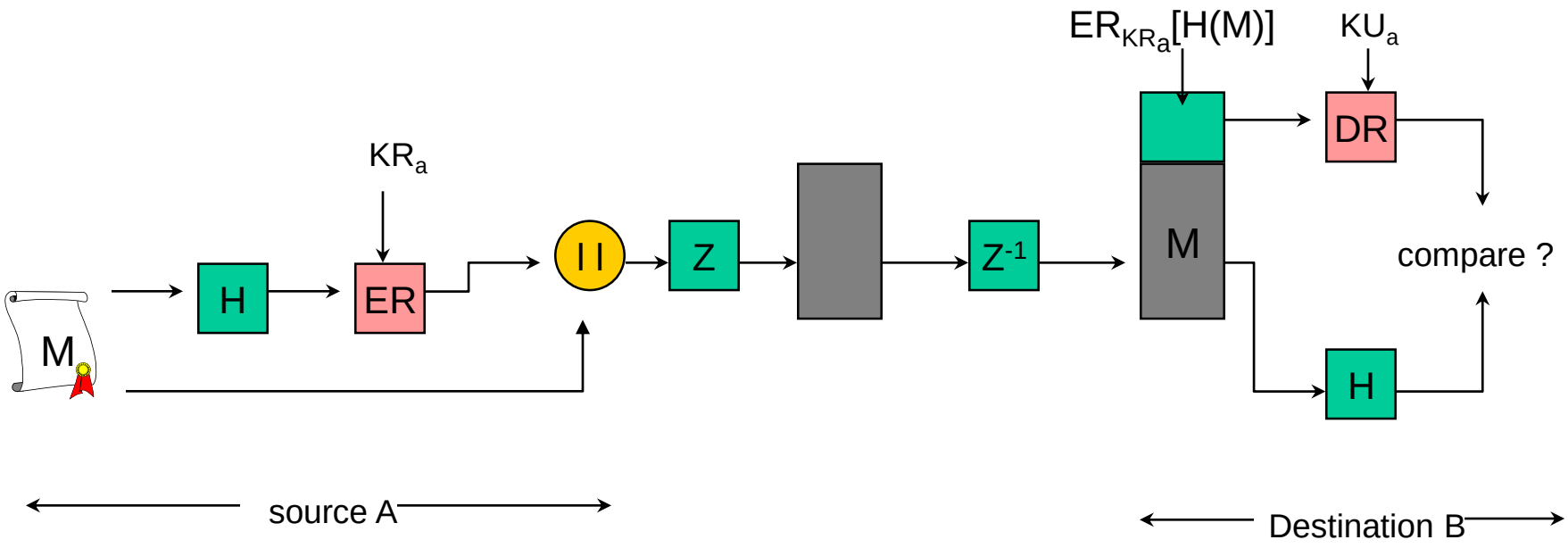
PGP Structure

- Authentication & Digital Signature

1. Creates a message (sender)
2. Generate a 160-bit hash code using SHA-1
3. Encrypt the hash code using RSA with sender's KR
4. Decrypt the hash code (receiver)
5. Generate a new hash code for the message and compare it with the decrypted hash code
6. If matched, then the message is authentic

PGP Structure

- Authentication & Digital Signature
  - Using RSA

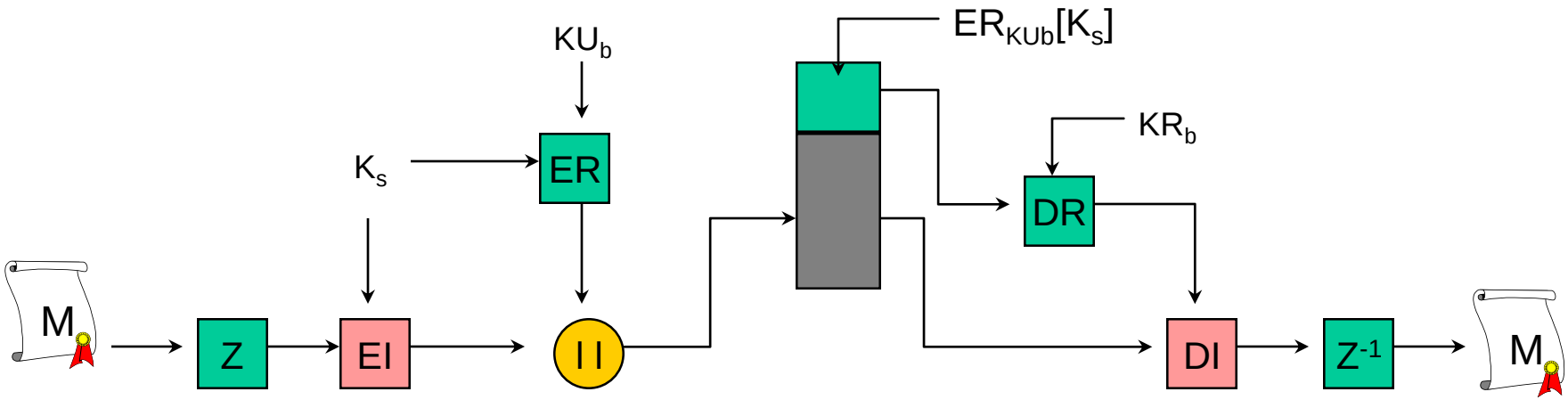


PGP Structure

- Confidentiality
  - Protecting message using session key
    - Source
      - Generating session key and 128 bits random number
      - Encryption of message using IDEA by session key
      - Encryption of session key using destination's RSA public key
    - Destination
      - Getting session key using destination's RSA private key
      - Decryption of message using session key

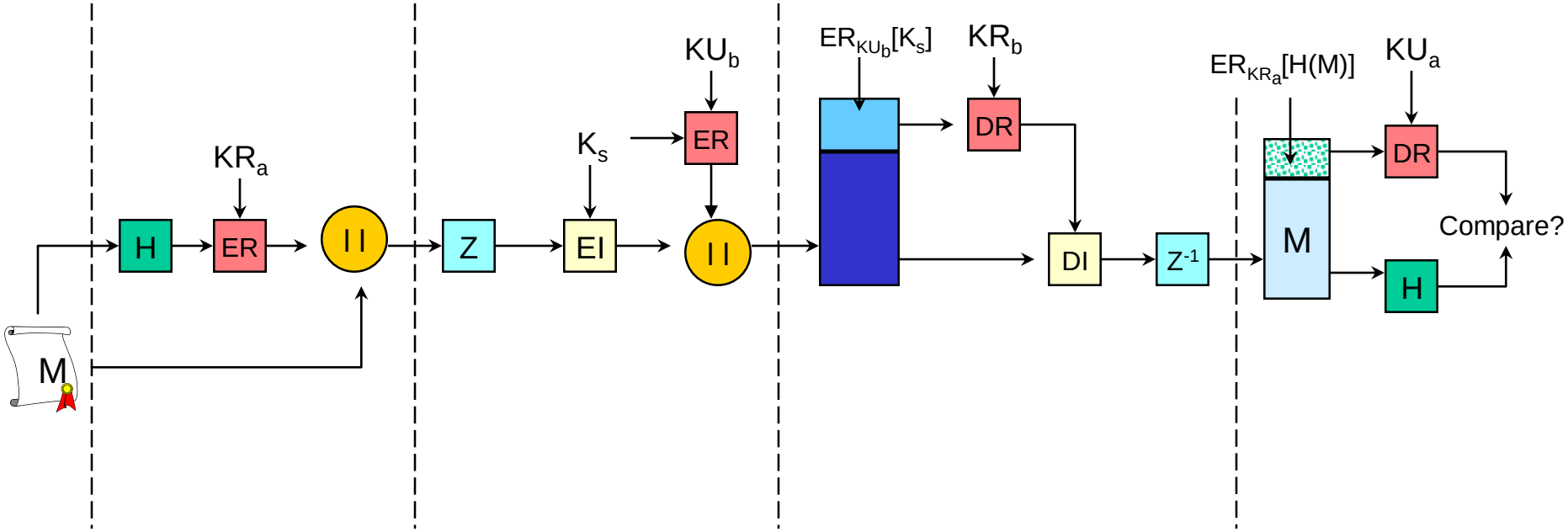
PGP Structure

- Confidentiality
  - Using RSA and IDEA



PGP Structure

- Confidentiality & Authentication
  - The first, authentication,
  - Then, processing for confidentiality



PGP Structure

- Compress
  - Use of ZIP algorithm
  - Done after signature before encryption, as a default
- PGP compresses the message
  - **after applying the signature**
    - can store only the uncompressed message together with the signature for future verification.
    - generate dynamically a recompressed message for verification.
  - **but before encryption to strengthen cryptographic security**
    - the compressed message has less redundancy than the original plaintext
    - cryptanalysis is more difficult

Email Compatibility

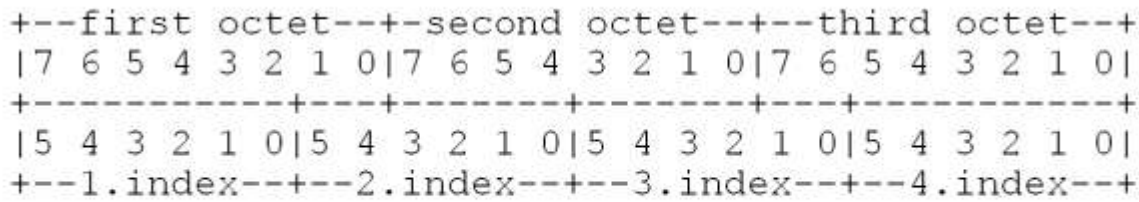
- Part or all of the resulting block consists of a stream of arbitrary 8-bit octets
- Many E-mail systems only permit the use of blocks consisting of ASCII text
- Use radix-64 conversion
  - 3 octets of binary data -> 4 ASCII characters + CRC

Input Data				
Binary representation	00100011	01011100	10010001	
Radix-64 Encoding of Input Data				
Character representation	I1yR			
ASCII code(8bit,zero parity)	01001001	00110001	01111001	01010010
Hexadecimal representation	493179052			

Segmentation and Reassembly

- E-mail facilities often are restricted to a maximum message length
- 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
  - the session key component and signature component appear only once, at the beginning of the first segment

About Radix-64

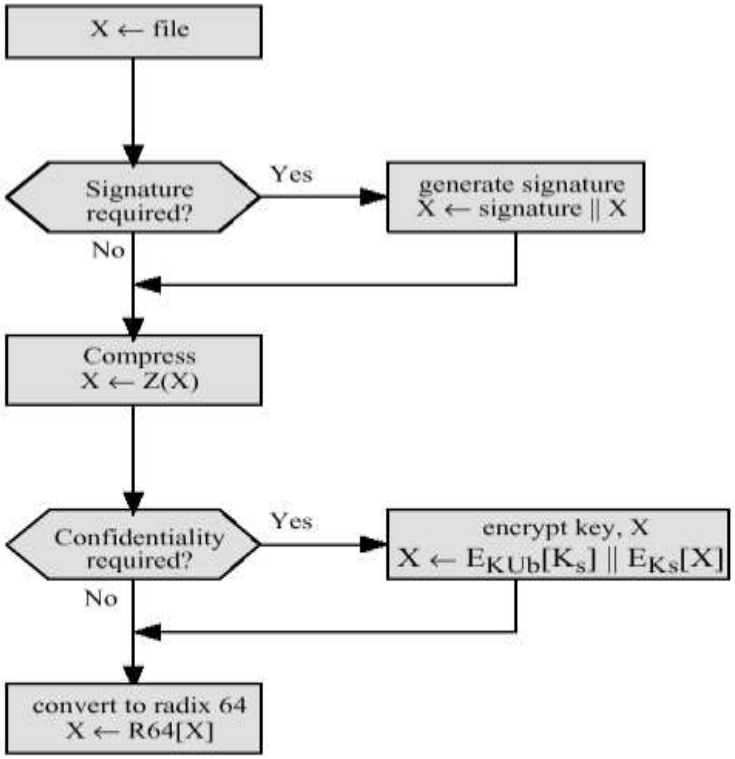


Each 6-bit group is used as an index into an array of 64 printable characters from the table below. The character referenced by the index is placed in the output string.

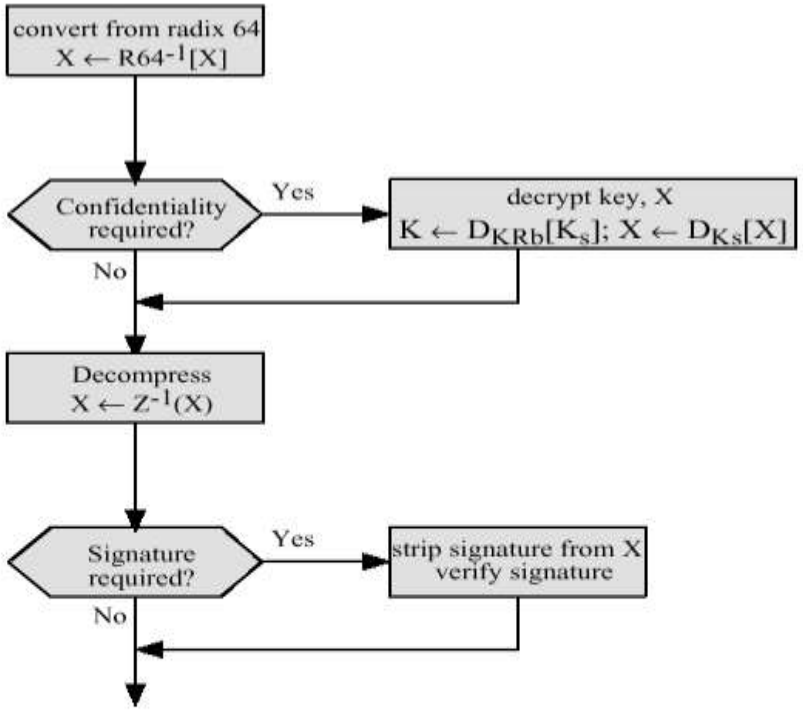
Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46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

PGP Operation – Summary

- Transmission and reception of PGP message



(a) Generic Transmission Diagram (from A)



(b) Generic Reception Diagram (to B)

Cryptographic Keys and Key Rings

- **Requirements for Key Management**

  1. A means of generating unpredictable session keys
  2. Allowance for use of multiple public/private key-pairs
  3. Maintaining a file of its own public/private key-pairs as well as a file of public keys of correspondents

Session Key Generation

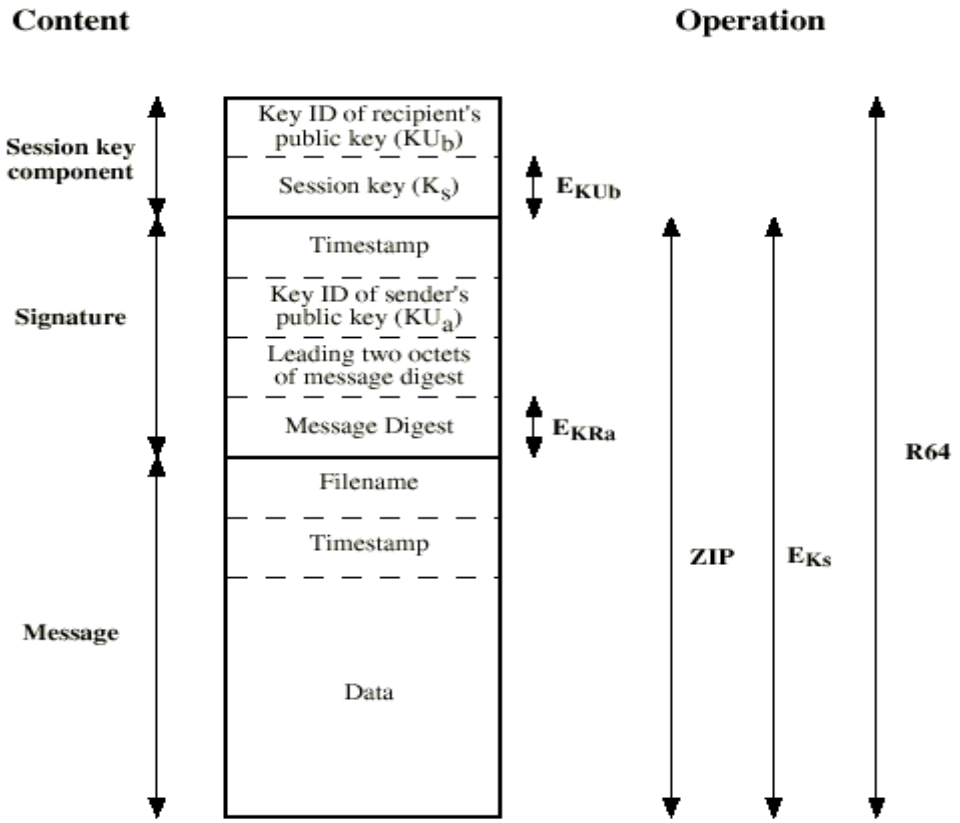
- Based on the algorithm specified in ANSI X12.17
- Random 128-bit numbers are generated using CAST-128
  - symmetric-key block cipher
- Using cipher feedback mode, the CAST-128 encrypter produces two 64-bit cipher text blocks , which are concatenated to form the 128-bit session key

CAST : Carlisle Adams and Stafford Tavares

Key Identifiers

- Any given user may have multiple public/private key pairs
- The key ID of KU_a = KU_a mod 2^{64}
- Very high probability, unique within a user ID
- Signature component includes Key ID of senders' public key
- Session key component includes Key ID of recipients' public key.

General Format of PGP Message



Notation:

E_{KUb} = encryption with user b's **public** key
E_{KRa} = encryption with user a's **private key**
E_{Ks} = encryption with session key
ZIP = Zip compression function
R64 = Radix-64 conversion function

Key Rings (1/4)

- The private-key ring stores the public/private key pairs
  - secring.pkr
- The public-key ring stores the public keys of other users known
  - pubring.pkr
- The private key is encrypted using CAST-128(or IDEA or 3DES)
  - $E_{H(Pi)}[KR_i]$ (Pi : passphrase)

Key Rings (2/4)

Private-Key Ring

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

* = field used to index table

General Structure of Private- and Public-Key Rings

Key Rings (3/4)

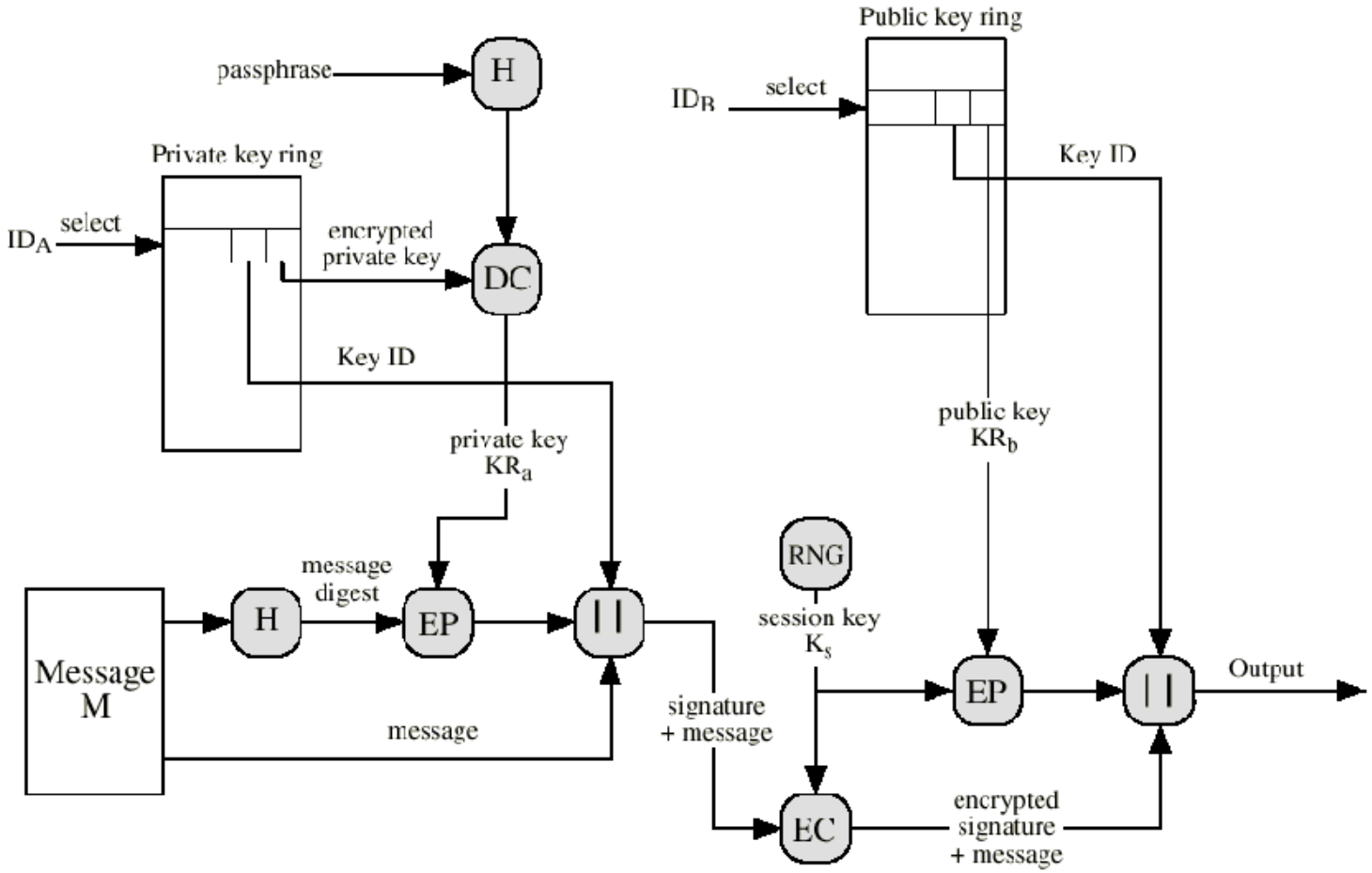


Figure 5.5 PGP Message Generation (from User A to User B; no compression or radix 64 conversion)

Key Rings(4/4)

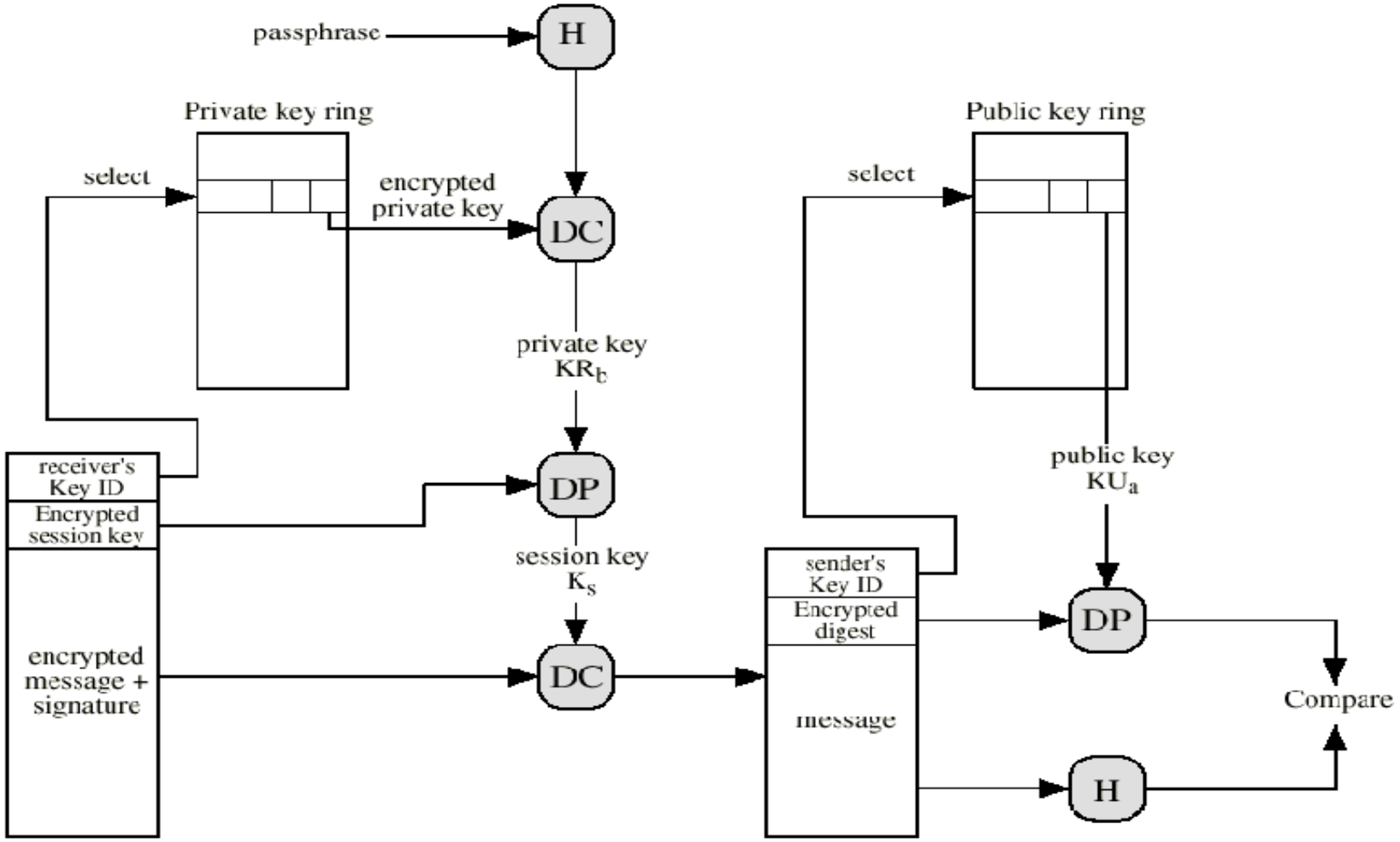


Figure 5.6 PGP Message Reception (from User A to User B; no compression or radix 64 conversion)

Approaches to Public-Key Management

- The essence of the problem
  - Public-Key Management
  - User A must build up a public-key ring containing the public keys of other users to interoperate with them using PGP
  - If A's key ring contains a public key attributed to B but that the key is owned by C
    - C can forge B's signature
    - C can read encrypted message from A to B

Approaches to Public-Key Management

- **Some approaches that could be used**

  1. **Physically get the key from B**
  2. **Verify a key or a digest of the key by telephone**
  3. **Obtain B's public key from a mutual trusted individual D**
  4. **Obtain B's public key from a trusted certifying authority**

The Use of Trust

- A owner trust field
  - the degree to which this public key is trusted to sign other public-key certificates
- A signature trust field
  - the degree to which this PGP user trusts the signer to certify public keys
- A key legitimacy field
  - the extent to which PGP will trust that this is a valid public key for this user

The Use of Trust

Table 12.2 Contents of Trust Flag Byte

(a) Trust Assigned to Public-Key Owner (appears after key packet; user-defined)	(b) Trust Assigned to Public Key/User ID Pair (appears after User ID packet; computed by PGP)	(c) Trust Assigned to Signature (appears after signature packet; cached copy of OWNERTRUST for this signator)
OWNERTRUST Field —undefined trust —unknown user —usually not trusted to sign other keys —usually trusted to sign other keys —always trusted to sign other keys —this key is present in secret key ring (ultimate trust) BUCKSTOP bit —set if this key appears in secret key ring	KEYLEGIT Field —unknown or undefined trust —key ownership not trusted —marginal trust in key ownership —complete trust in key ownership WARNONLY bit —set if user wants only to be warned when key that is not fully validated is used for encryption	SIGTRUST Field —undefined trust —unknown user —usually not trusted to sign other keys —usually trusted to sign other keys —always trusted to sign other keys —this key is present in secret key ring (ultimate trust) CONTIG bit —set if signature leads up a contiguous trusted certification path back to the ultimately trusted keyring owner

Revoking Public Key

- Issues a key revocation certificate, signed by the owner

- Note that an opponent who has compromised the private key of an owner can also issue such a certificate.

PGP install

- International PGP Home Page
  - http://www.pgpi.org/
- PGP Version
  - PGP 2.3a
  - PGP 2.6ui
  - MIT PGP 2.6.2
  - PGP 2.6.3i
  - PGP 3.0
  - PGP 7.0 and 8.0
  - Latest version : PGP Desktop 10.1

S/MIME

- **Secure/Multipurpose Internet Mail Extension**
  - A security enhancement to the MIME e-mail format standard
- **S/MIME will probably emerge as the industry standard.**
  - PGP will remain the choice for personal e-mail security
- **RFC822 → MIME → S/MIME**

※ PGP: RFC4800 (OpenPGP Message Format)

RFC 822

- **RFC 822 defines a format for text messages sent using e-mail**
- **A RFC822 message is viewed as having an envelope and contents**
  - The envelope : the information needed for delivery
  - The contents : the object to be delivered to the recipient
- **The RFC822 standard applies only to the contents**
  - A message consists of the header followed by the body
  - The header : used by the mail system to create the envelope
  - The body : unrestricted ASCII text

An Example Message of RFC822

Date: Tue, 16 Jan 1998 10:37:17(EST)

From: “William Stallings” <ws@shore.net>

Subject: The Syntax in RFC 822

To: Smith@other-host.com

Cc: Jones@yet-Another-Host.com

Hello. This section begins the actual message body, which is delimited from the message heading by a blank line.

MIME

- **MIME is an extension to the limitations and problems of the SMTP/RFC822 scheme**
- **SMTP Limitations**
  - no executable files or other binary objects (JPEG image) are transmitted.
  - no “national language” characters (non-ASCII) are transmitted :7-bit ASCII only
  - no delivery of mail messages over a certain size
  - no consistent mapping between ASCII and EBCDIC :Translation problem of SMTP gateway
  - no textual data included in X.400 messages to be not handled by SMTP gateways
  - Problems due to some implementations do not conformed to SMTP standards

Overview of MIME

- **Five new message header fields are defined**
  - Provision of information about the body of the message
- **A number of content formats are defined**
  - Support of multimedia e-mail
- **Transfer encoding are defined**
  - Conversion of any content format into a format for protection from alteration by the mail system

Header Fields in MIME

- **MIME-Version :** Must be “1.0”
- **Content-Type :** Describing the data in the body for adequate treatment by the recipient
- **Content-Transfer-Encoding :** The type of encoding the body of message(RADIX-64)
- **Content-ID :** Identifying MIME entities uniquely in multiple contexts
- **Content-Description :** A text description of the object with the body (useful for non-readable object: mpeg, audio data)

MIME Content Types

- **The need to support a multimedia environment with a wide variety of information representations**
- **Text : no special software required**
  - Plain : ASCII or ISO 8859 characters
  - Enriched : greater formatting flexibility
- **Multipart : Multiple/independent parts in the body**
  - Mixed/Parallel/Alternative/Digest
- **Message : A number of different capabilities in MIME**
  - RFC822/Partial/External-body

MIME Transfer Encodings

- **The purpose is to provide reliable delivery across the largest range of environments**
- **Quoted-printable : the human readable transfer technique**
  - Useful for the data consisting of largely ASCII characters
- **Base-64(RADIX-64) : representation for all types of data**
  - Common for encoding arbitrary binary data

MIME Example (Multipart Message)

MIME-Version: 1.0

From: Nathaniel Borenstein

Subject: A multipart example

Content-Type: multipart/mixed;

boundary=unique-boundary-1

This is the preamble area of a multipart message. Mail readers that understand multipart format should ignore this preamble. If you are reading this text, you might want to consider changing to a mail reader that understands how to properly display multipart messages.

--unique-boundary-1

Some text appears here... [Note that the preceding blank line means no header fields were given and this is text, with charset US ASCII. It could have been done with explicit typing as in the next part.]

--unique-boundary-1

Content-type: text/plain; charset=US-ASCII

This could have been part of the previous part, but illustrates explicit versus implicit typing of body parts.

--unique-boundary-1

Content-Type: multipart/parallel; boundary=unique-boundary-2

--unique-boundary-2

Content-Type: audio/basic

Content-Transfer-Encoding: base64

... base64-encoded 8000 Hz single-channel u-law-format audio data goes here ...

--unique-boundary-2

Content-Type: image/gif

Content-Transfer-Encoding: Base64

... base64-encoded image data goes here ...

MIME Example (Multipart Message)

- Date: Tue, 3 Sep 1996 09:25:52 -0700 (PDT)
- From: Judith Grobe Sachs <judygs@uic.edu>
- To: Judith Grobe Sachs <judygs@uic.edu>
- Subject: A MIME Example
- Message-ID: <.\Pine.PCW.3.95.96090316.63B110@judys.cc.uic.edu> X-X-Sender: judygs@tigger.cc.uic.edu
- **MIME-Version: 1.0**
- **Content-Type: MULTIPART/MIXED; BOUNDARY="5494-19501-841=:9866"**
- **--5494-19501-841=:9866**
- **Content-Type: TEXT/PLAIN; charset=US-ASCII**
- This is the regular text body of a sample message with MIME. *... the rest of the plain text body ...*
- **--5494-19501-841=:9866**
- **Content-Type: VIDEO/x-msvideo; name="MACAW.AVI"**
- **Content-Transfer-Encoding: BASE64**
- **Content-ID: <Pine.PCW.3.95.96090351.63C@judys.cc.uic.edu>**
- **Content-Description: This is a MS movie.**
  UklGRrC3AQBBVkkgTElTVNQHAABoZHJsYXZpaDgAAACFRQEAUccAAAAAAAAQ

S/MIME Functions

- **Enveloped data :** Encrypted content and encrypted session keys for recipient
- **Signed data :** Digital signature and content encoded by base64
- **Clear-signed data :** Only digital signature encoded by base64
- **Signed and enveloped data :** Signed-only and encrypted-only entities may be nested

Cryptographic Algorithms Used

- **Message Digesting :** SHA-1 and MD5
- **Digital Signatures :** DSS, RSA (with key sizes 512-to-1024 bits)
- **Encryption of session keys :** RSA and Diffie-Hellman
- **Encryption of messages :** 3DES, AES, and RC2/40
- **Creation of a MAC :** HMAC with SHA-1

Choice of a Encryption Algorithm

- **A sending agent has two decisions to make :**
  - Recipient's capability of decrypting a given encryption algorithm
  - Sender's acceptance if recipient is capable of using only weak algorithm
- **To support this decision process,**
  - A sending agent may announce its decrypting capabilities in order of preference to its message
  - A receiving agent may store that information for future use

S/MIME Certificate Processing

- S/MIME uses Public-key Certificates of version 3 of X.509
- Key management scheme used by S/MIME

  is a hybrid between PGP's web of trust and a strict X.509 certification hierarchy
- The certificates are signed by certification authorities

S/MIME Certificate - User Agent Role -

- S/MIME uses Public-Key Certificates - X.509 version 3 signed by Certification Authority
- Functions:
  - **Key Generation** - Diffie-Hellman, DSS, and RSA key-pairs.
  - **Registration** - Public keys must be registered with X.509 CA.
  - **Certificate Storage** - Local (as in browser application) for different services.
  - **Signed and Enveloped Data** - Various orderings for encrypting and signing.

Certification Authority Role
-VeriSign-

- **VeriSign provides a CA service with issuing X.509 certificates**
- **Three levels or classes of security for certificates**
  - **Class-1** : User's e-mail address is confirmed
  - **Class-2** : The specified postal address is confirmed as well
  - **Class-3** : An individual must prove his identity by providing notarized credentials or applying in person (offline)

Certification Authority Role -VeriSign-

- Digital ID contains:
  - Owner's public key
  - Owner's name and alias
  - Expiration date of the Digital ID
  - Serial Number of the Digital ID
  - Name of the CA that issued the Digital ID
  - Digital Signature of CA that issued the Digital ID
  - And user-supplied information
    - Address
    - E-mail address
    - Basic registration information (country, zip code, age, and gender)

Enhanced Security Services

- **Signed receipts : the proof of delivery**
  - Usable for demonstrating to a third party
  - Signing on the entire message plus sender's signature
- **Security labels : access control**
  - A security label is a set of information regarding the sensitivity of the content that is protected by S/MIME
  - By indicating which users are permitted access to an object
- **Secure mailing list : the services for sending multiple users a mail**
  - Relief from the burden on processing per recipient such as the use of each recipient's KU
  - Use of services of an Mail List Agent to handle those things

Summary

- PGP is an open-source freely available software package for email security
- PGP incorporates tools for developing a public key trust model and public key certificate management
- S/MIME is a Internet standard approach to email security that incorporate the same functionality as PGP