IT-Security 1

Chapter 11: Email Security

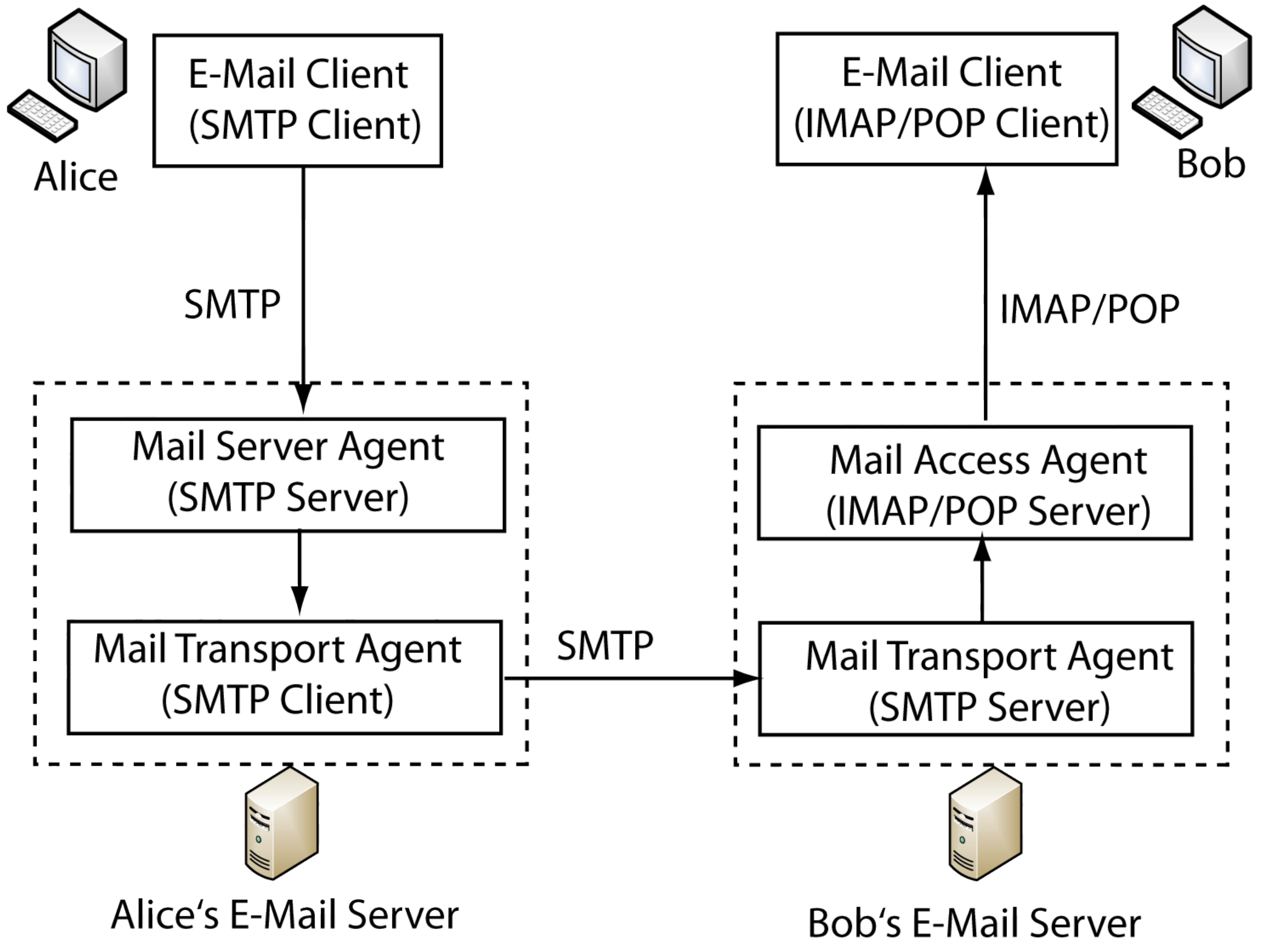
Prof. Dr.-Ing. Ulrike Meyer

WS 15/16

Chapter Overview

- Typical email architecture and protocols
- Email security in general
- Pretty Good Privacy (PGP)
- Secure Multipurpose Internet Mail Extension (S/MIME)

Typical Email Architecture


Alice
E-Mail Client (SMTP Client)
SMTP
Mail Server Agent (SMTP Server)
Mail Transport Agent (SMTP Client)
SMTP
Alice's E-Mail Server
E-Mail Client (IMAP/POP Client)
Bob
IMAP/POP
Mail Access Agent (IMAP/POP Server)
Mail Transport Agent (SMTP Server)
Bob's E-Mail Server

- Alice sends email to Bob
- Note: sending email is asynchronous

Protocols

- Simple Mail Transfer Protocol (SMTP)
  - Used by Alice to push email to pre-configured SMTP server
  - Used by Alice's email server acting as SMTP client to push email to Bob's email server acting as SMTP server
- Internet Message Access Protocol (IMAP) or Post Office Protocol (POP)
  - Used by Bob to pull email from his mailbox at Bob's mail server

Email Security – Two Approaches

- End-to-end protection of emails between sender and receiver
  - E.g. PGP/MIME, S/MIME
  - Protection of emails between sender and receiver
- Hop-by-hop protection of emails by extensions to the protocols transporting them
  - E.g. SMTPs, POP3s, IMAPs use SSL to protect mail transfer
  - Used to provide confidentiality and integrity on the connection to mail server when sending mail or retrieving mail from mailbox
  - Can be used to provide confidentiality and authenticity of connection between mail servers as well

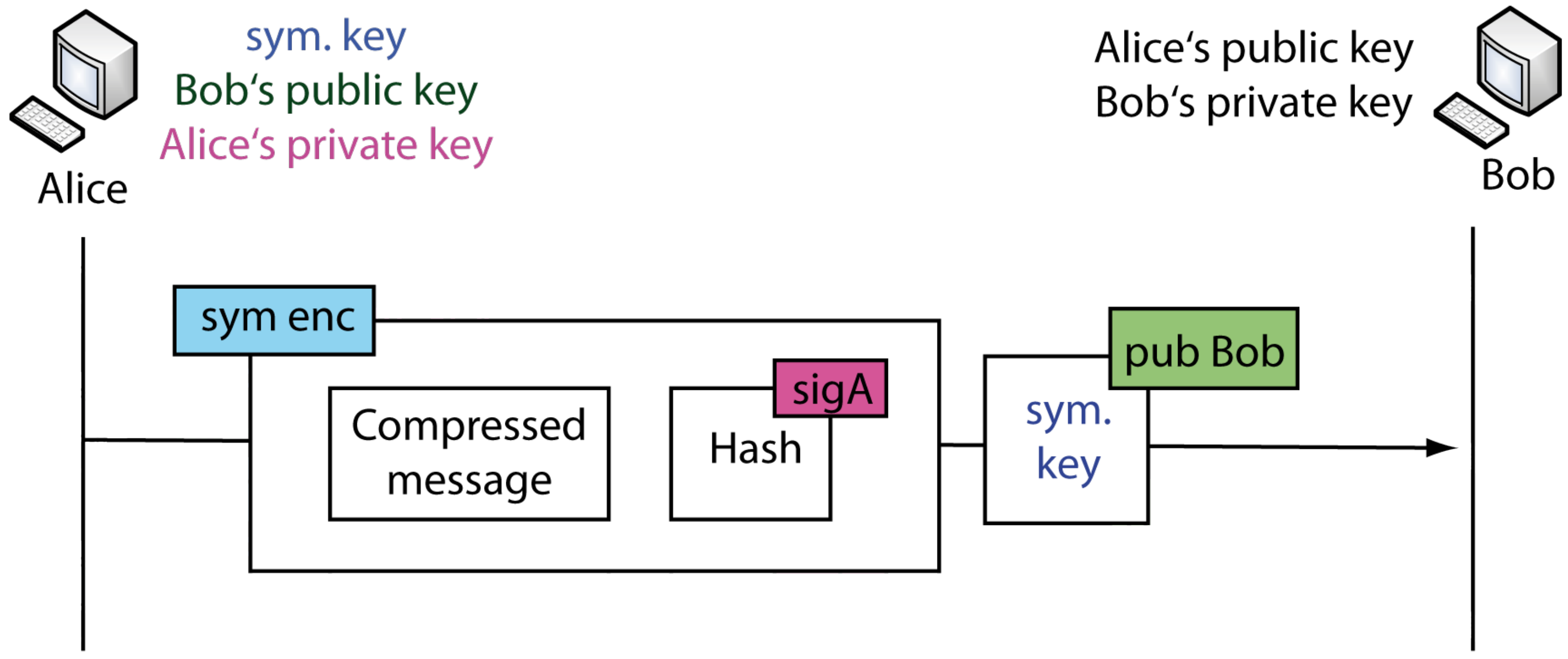
End-to-End Email Security Goals

- Confidentiality
  - Protection of email content from disclosure to anyone but sender and receiver
- Authentication
  - Of sender of message
- Message integrity
  - Protection of message content from modification
- Non-repudiation of origin
  - Sender cannot deny having sent email

End-to-End Email Protection Approach

- Hybrid approach e.g. used by PGP and S/MIME
- Sender
  - Selects symmetric encryption key
  - Encrypts email message with symmetric cipher
  - Encrypts symmetric encryption key with public key of receiver
  - Sends encrypted message, identifiers of cryptographic algorithms used, and encrypted symmetric key to receiver
- Sender has to be able to obtain authentic copy of public key of receiver
  - Solved with the help of certification authorities in S/MIME
  - Solved with the help of “a web of trust” in PGP

PGP Overview



- If Alice sends an email to Bob she
  - Hashes message and signs hash with her private key
  - Compresses the message
  - Encrypts compressed message and signed hash with a symmetric key
  - Encrypts symmetric key with Bob's public key and appends it to the encrypted message

PGP Key Rings

- In PGP each user keeps two key rings
  - A public ring and
  - A private ring
- The public ring
  - Contains public keys of potential receivers
- The private ring
  - Contains pairs of private/public keys of the sender
    - Sender may want to keep more than one private/public key pair to use them for different purposes
      - E.g. an employee of a company may share his business private/public key pair with his surrogate and keep an other private/public key pair for private emails

PGP Public Key Algorithms

- Used to sign digests or encrypt symmetric keys

ID	Description
1	RSA (encryption or signing)
2	RSA (encryption only)
3	RSA (signature only)
16	ElGamal (encryption only)
17	DSS
18	Reserved for elliptic curve
19	Reserved for ECDSA
20	ElGamal (encryption or signing)
100 – 110	Private algorithms

PGP Symmetric Encryption Algorithms

- Used to encrypt the compressed message and the signed hash of the message

ID	Description
0	No encryption
1	IDEA
2	Triple DES
3	CAST-128
4	Blowfish
5	SAFER-SK 128
6	DES/SK
7	AES-128
8	AES-192
9	AES-256
100 – 110	Private algorithms

PGP Hash Functions

- Used to hash the message before signing it

ID	Description
1	MD5
2	SHA-1
3	RIPE-MD/160
4	Reserved for double-width SHA
5	MD2
6	Tiger/192
100 - 110	Private Algorithms

PGP Compression Algorithms

- Used to compress messages

ID	Description
0	Uncompressed
1	ZIP
2	ZLIB
100 - 110	Private Algorithms

PGP’s Web of Trust

- No certification authorities required
  - Use of CAs also not excluded
- Public keys can be signed by any PGP user
- If a certificate is signed by a PGP user that user is called “introducer”
- There can be multiple paths in the line of trust from an introducer to a certificate
- Multiple introducers can issue certificates for the same user

Introducer Trust Level

- Trust level Alice assigns to each potential introducer
- Implementations typically by default support four trust levels
  - Don’t know
    - Alice assigns “don’t know” to Bob as introducer if she doesn’t know anything about his behavior
  - No trust
    - Alice assigns “no trust” to Bob if she knows that he doesn’t really check to care identities before signing public keys of others
  - Partial trust
    - Alice assigns “partial trust” to Bob if she knows that Bob does some checking but is not extremely careful before signing
  - Full trust
    - Alice assigns “full trust” to Bob if she knows he’s a security expert and would never sign anyone’s key without thoroughly checking the identity first

Certificate Trust Level

- The certificate trust level is the trust level Alice assigns to each certificate she receives
- The trust level of a certificate corresponds to the trust level assigned to the introducer (= the signer of the certificate)
  - E.g. if Alice fully trusts the introducer Bob and Alice receives a certificate signed by Bob for John then Alice assigns full trust to this certificate of John
  - If Alice partially trusts Bob, she will trust John’s certificate only partially
- Note that Alice may receive multiple certificates for the same public key from several different introducers
  - The trust level she assigns to these certificates add up to the “key legitimacy” (see next slide)

Key Legitimacy in General

- Determined for each public key of a potential receiver
- Computed as weighted sum of the trust level of the certificates received for this public key
- I.e. Alice assigns weights of
  - 1/x to certificates issued by fully trusted introducers
  - 1/y to certificates issued by partially trusted introducers
  - 0 to certificates issued by untrusted and “don’t know” introducers
- If Alice has
  - N_x certificates issued by partially trusted introducers for John,
  - N_y certificates issued by fully trusted introducers for John
- Then the key legitimacy is computed by:

$$\text{Key legitimacy} = \begin{cases} 1 \text{ if } N_x / x + N_x / y >= 1 \\ 0 \text{ if } N_x / x + N_x / y < 1 \end{cases}$$

Example (x=1, y=2)

User	Introducer Trust	Certificates	Certificates Trust Level	Key Legitimacy
Bob	Fully	$\mathrm{Cert}_{\mathrm{Clare}}$(Bob) $\mathrm{Cert}_{\mathrm{Ian}}$(Bob)	1, 0	1
Clare	Fully	$\mathrm{Cert}_{\mathrm{ian}}$(Clare)	0	0
Dave	Partially	$\mathrm{Cert}_{\mathrm{Han}}$(Dave), $\mathrm{Cert}_{\mathrm{John}}$(Dave)	½ , ½	1
Han	Partially	$\mathrm{Cert}_{\mathrm{John}}$(Han), $\mathrm{Cert}_{\mathrm{Ian}}$(Han)	½ , 0	0
Ian	Untrusted	$\mathrm{Cert}_{\mathrm{Bob}}$(Ian), $\mathrm{Cert}_{\mathrm{Clare}}$(Ian)	1, 1	1
John	Partially	$\mathrm{Cert}_{\mathrm{Clare}}$(John), $\mathrm{Cert}_{\mathrm{Han}}$(John)	1, ½	1

- For each other user, table shows
  - How Alice defined the introducer trust level for them
  - Which certificates Alice received by whom for this user
  - What certificate trust levels are associated with the certificates
  - What the current key legitimacy is

Example (x=2, y=4)

User	Introducer Trust	Certificates	Certificates Trust Level	Key Legitimacy
Bob	Fully	$\text{Cert}_{\text{Clare}}$(Bob) Cert_{Ian}(Bob)	1/2, 0	0
Clare	Fully	Cert_{ian}(Clare)	0	0
Dave	Partially	Cert_{Han}(Dave), $\text{Cert}_{\text{John}}$(Dave)	1/4 , 1/4	0
Han	Partially	$\text{Cert}_{\text{John}}$(Han), Cert_{Ian}(Han)	1/4 , 0	0
Ian	Untrusted	Cert_{Bob}(Ian), $\text{Cert}_{\text{Clare}}$(Ian)	1/2, 1/2	1
John	Partially	$\text{Cert}_{\text{Clare}}$(John), Cert_{Han}(John)	1/2, 1/4	0

- For each other user, table shows
  - How Alice defined the introducer trust level for them
  - Which certificates Alice received by whom for this user
  - What certificate trust levels are associated with the certificates
  - What the current key legitimacy is

Private Key Ring Table

- PGP defines the following structure for the private key ring table each user stores

User ID	Key ID	Public key	Encrypted private key	Timestamp
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*

Explanation

- User ID – email address of user
- Key ID – uniquely identifies the public key among the user’s public keys (64 LSBs of public key)
- Public key – stores the public key
- Encrypted private key – private key encrypted with a passphrase
- Timestamp – date and time of key pair creation, can be used to decide when to purge old key pairs and create new ones

Public Key Ring Table

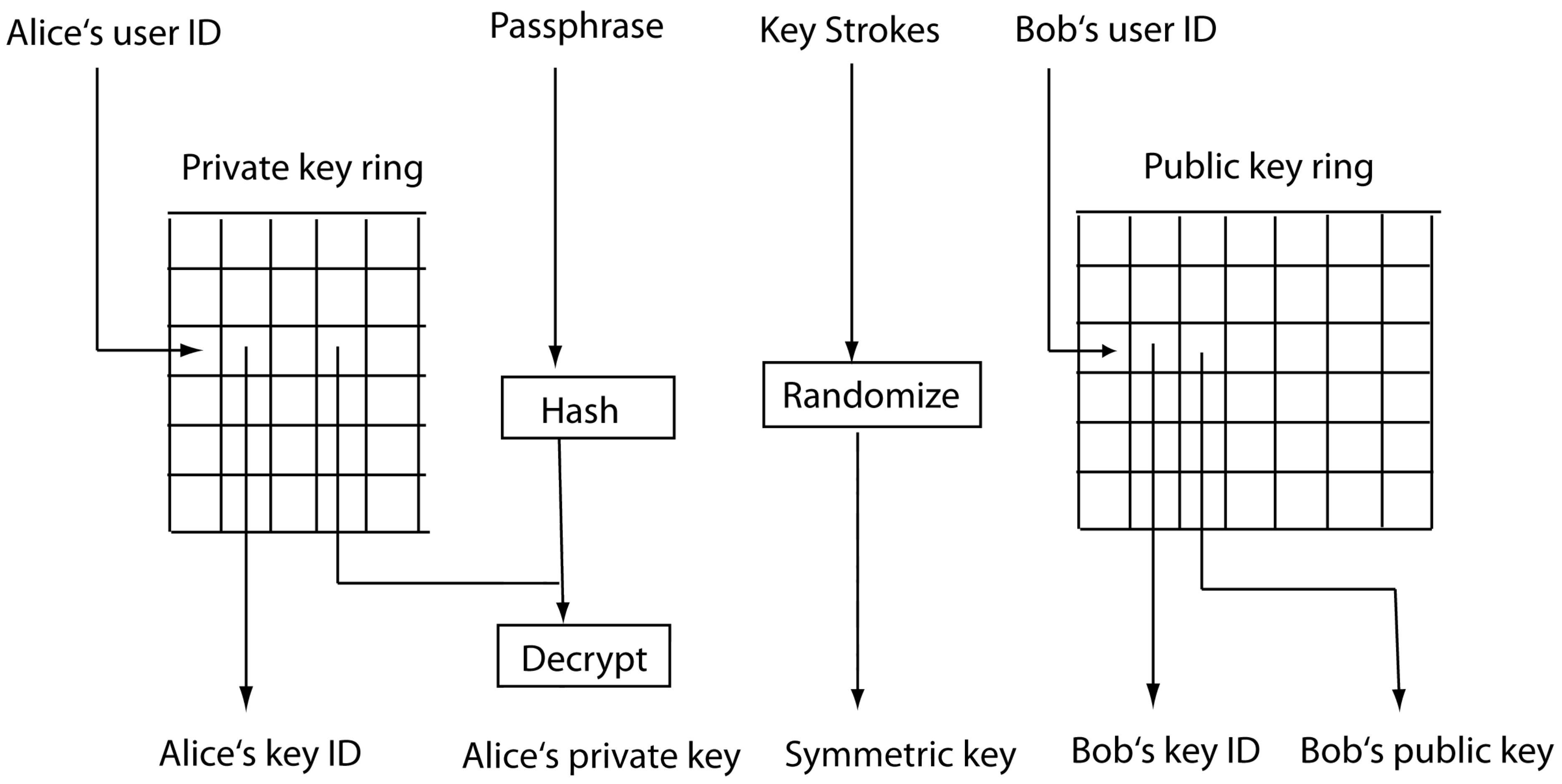
- PGP defines the following structure for the public key ring table each user stores

User ID	Key ID	Public key	Intro-ducer trust	Certi-ficates	Certi-ficates trusts	Key Legi-timacy	Time-stamp
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*

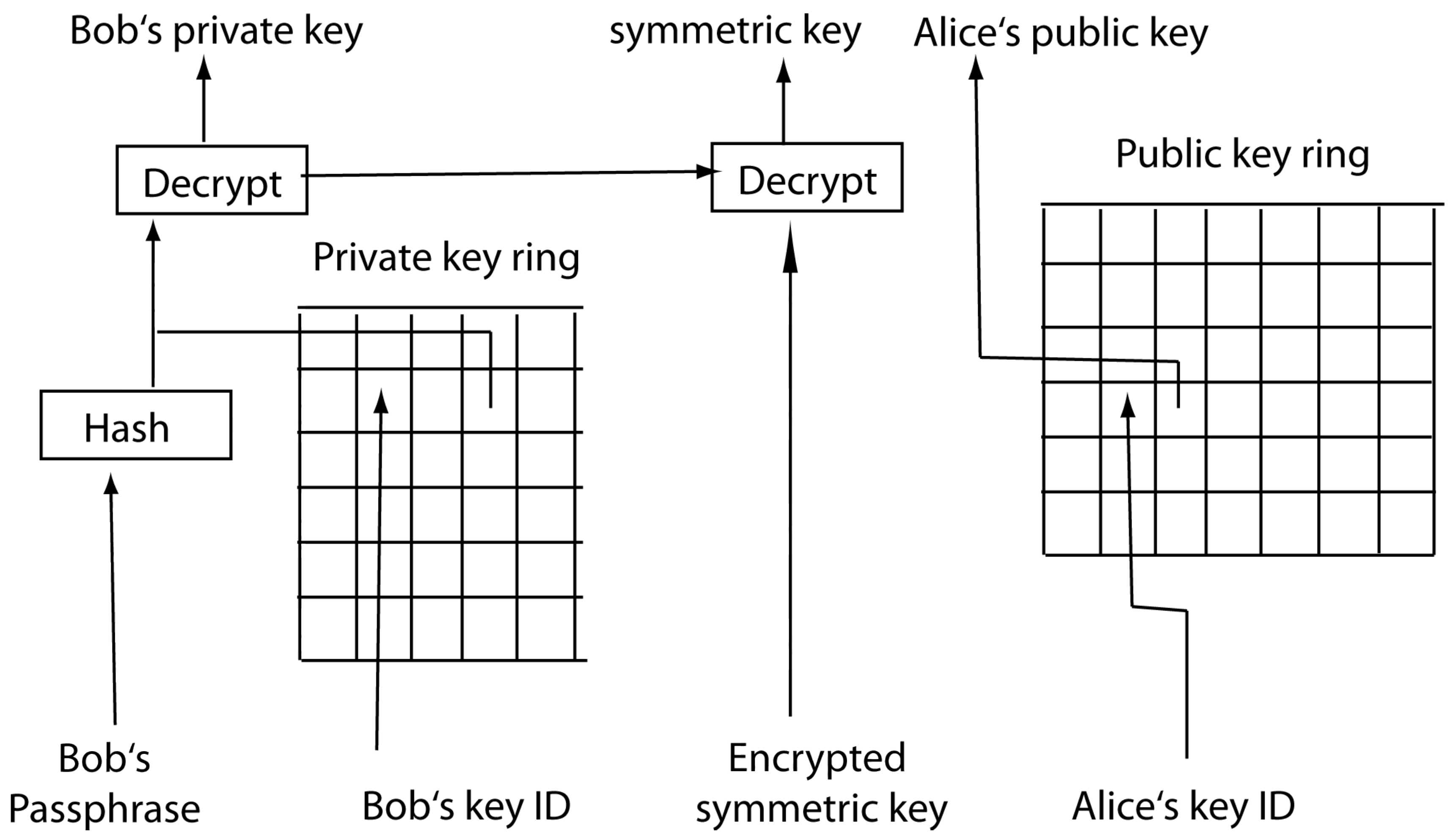
Explanation

- User ID – email address
- Key ID – 64 LSB of the public key
- Public key – public key
- Introducer trust – trust level of user if acting as introducer
- Certificates – list of certificates for the public key
- Certificates trust – list of trust levels corresponding to the certificates
- Key Legitimacy – weighted sum of the certificate trust levels
- Timestamp – data and time of row entry creation

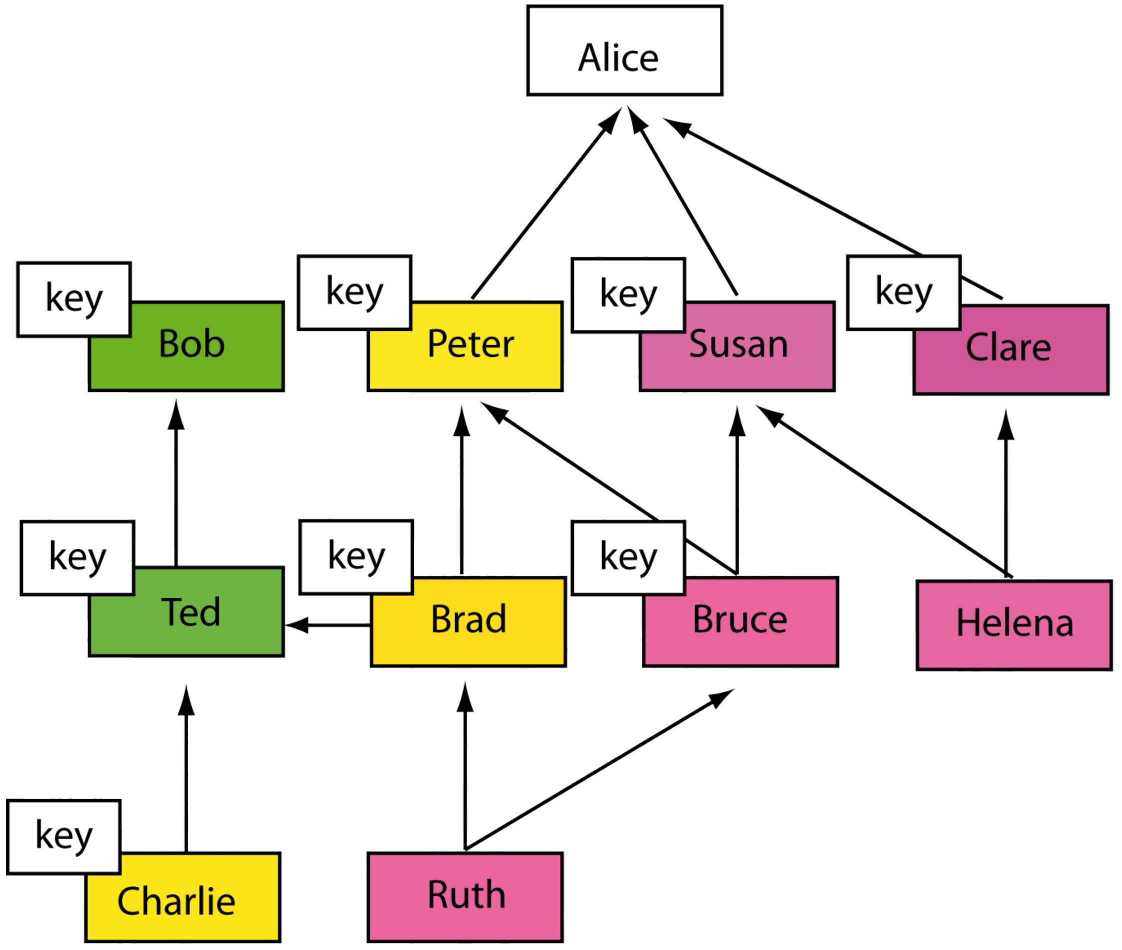
PGP Processing on Sender Side



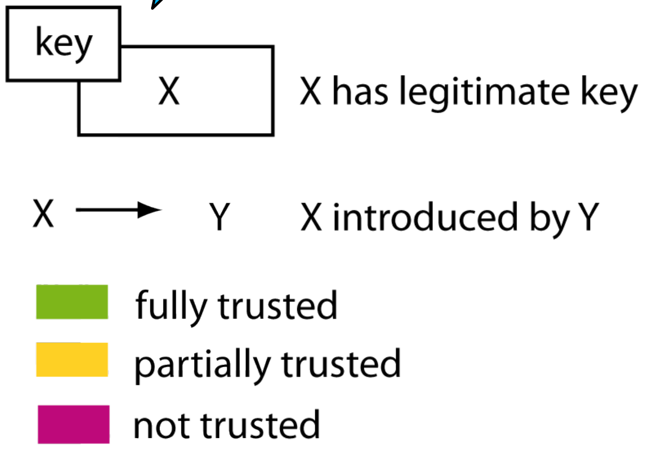
PGP Processing on Receiver Side



Trust Model for Alice in PGP



Assumes that two partially trusted and one fully trusted certificates are required for key legitimacy



Key Revocation

- To revoke a key the owner disseminates a revocation certificate signed by the key that is to be revoked
- The revocation certificate needs to be distributed to everyone that uses the key
  - How???
  - E.g. with the help of key servers that also help with distributing public keys in the first place
- What if the private key corresponding to the public key is lost or stolen and no copy of it is available any more at its owner?
- Solution: sign revocation certificate right after key generation and store it safely
  - Anyone can publish the revocation certificate

PGP History and Application

- Originally designed by Phil Zimmermann in 1991
- Source code published in 1995
- Bought by McAfee in 1997
- Resold to Phil Zimmermann and colleagues in 2002
- Further development of PGP by the IETF
  - Latest RFC: RFC 4880 November 2007
- Supported by many email clients

S/MIME

- S/MIME is a security service designed for email
- Stands for Secure/MIME
- Is an enhancement of the MIME protocol
- Adds new content types to MIME that can carry the original MIME content types
- Supports the following security services
  - Origin authentication of messages
  - Message integrity
  - Non-repudiation of origin
  - Data confidentiality

Email Format and MIME

- Originally, electronic mail only supported 7-bit ASCII format (RFC 2822) as SMTP only supports 7-bit ASCII
  - Emails formatted in two parts: header and body
- As a consequence languages that are not supported by the 7-bit ASCII format were not supported
  - Including German, Chinese, Hebrew, Japanese, Arabic
- Multipurpose Internet Mail Extension (MIME) allows for non-ASCII data to be sent through SMTP
- MIME
  - Transforms non-ASCII data into ASCII data
  - Delivers it to the email client to sent through the Internet via SMTP

MIME Header

- MIME defines a new header that is added to the original email header
- The MIME header contains
  - MIME-Version – currently 1.1
  - Content-Type – type/subtype
    - See next slide
  - Content-Transfer-Encoding – encoding type (7bit, 8bit ASCII, binary, radix-64, quoted-printable)
  - Content-ID – uniquely identifies the message
  - Content Description – image, audio or video

MIME Content Types

Type	Subtype	Description
Text	Plain	Unformatted
	HTML	HTML format
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above but unordered
	Digest	Similar to mixed but default is message/RFC822
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message

MIME Content Types (cont'd)

Type	Subtype	Description
Image	JPEG	Image in JPEG format
	GIF	Image in GIF format
Video	MPEG	Video in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data

New Content Types in S/MIME

- Data content type (arbitrary string)
- Signed-data content type
- Enveloped-data content type
- Digested-data content type
- Encrypted-data content type
- Authenticated-data content type

S/MIME Signed-data Content Type

- Used to apply a signature to a message to provide
  - Authentication, message integrity, and non-repudiation of origin
- Contains:
  - Any other content type
  - One or more signatures on hashes of the content
  - Certificates for the public verification keys corresponding to the signature generation keys used
  - The signature algorithms used for each signature

S/MIME Enveloped-data Content Type

- Used to apply data confidentiality to a message
- Requires sender to have access to a public key for each recipient of the message to be sent
- Contains:
  - Any other content type symmetrically encrypted
  - The symmetric encryption algorithm used
  - For each recipient
    - The recipient identifier
    - The public key certificate identifier used for each recipient
    - The symmetric key used to encrypt the content type encrypted with the public key of each recipient

S/MIME Digested-data Content Type

- Typically used as the content of enveloped-data type
- Adds a hash and the hash algorithm used to the content type

S/MIME Encrypted-data Content Type

- Used to locally store content, not to transmit content to recipient
  - E.g. store email encrypted on the local hard drive
- Content encrypted with secret key e.g. derived from passphrase

S/MIME Authenticated-data Content Type

- Adds one or more message authentication codes to the content type
- For each recipient
  - Adds the public key certificate identifier
  - Adds the symmetric message authentication code key encrypted with the public key of the recipient
  - Adds the MAC and MAC algorithm used

S/MIME Algorithms in V.3.1

Algorithm	Sender must support	Receiver must support	Sender should support	Receiver should support
Content-encryption	3DES	3DES	AES	AES RC2/40
Session-key encryption	RSA	RSA	ElGamal	ElGamal
Hash algorithm	SHA-1	SHA-1		MD5
Signature algorithm	DSS	DSS	RSA	RSA
MAC	HMAC with SHA-1	HMAC with SHA-1		

S/MIME Certificate Processing

- S/MIME uses X.509 v3 certificates
- Each client has a list of trusted CA's certs
- Each client has its own public/private key pairs and certificates
- Certificates must be signed by trusted CA's
- Each sender has to be able to retrieve the certificate of a potential receiver e.g. via X.500 directory servers or with the help of incoming messages

Recommended Reading

- Book chapters:
- Forouzan Chapter 16
- Stallings Chapter 15

- Original specifications:
- PGP: RFC 4880 OpenPGP Message Format
  - http://www.ietf.org/rfc/rfc4880.txt
- S/MIME:
  - RFC 3851 S/MIME 3.1 Message Specification
  - http://www.ietf.org/rfc/rfc3851.txt
- RFC 3369 Cryptographic Message Syntax (CMS)
  - http://www.ietf.org/rfc/rfc3369.txt
- RFC 3850 S/MIME 3.1 Certificate Handling
  - http://www.ietf.org/rfc/rfc3850.txt