Introduction into Cyber Security

Chapter 11: E-Mail Security

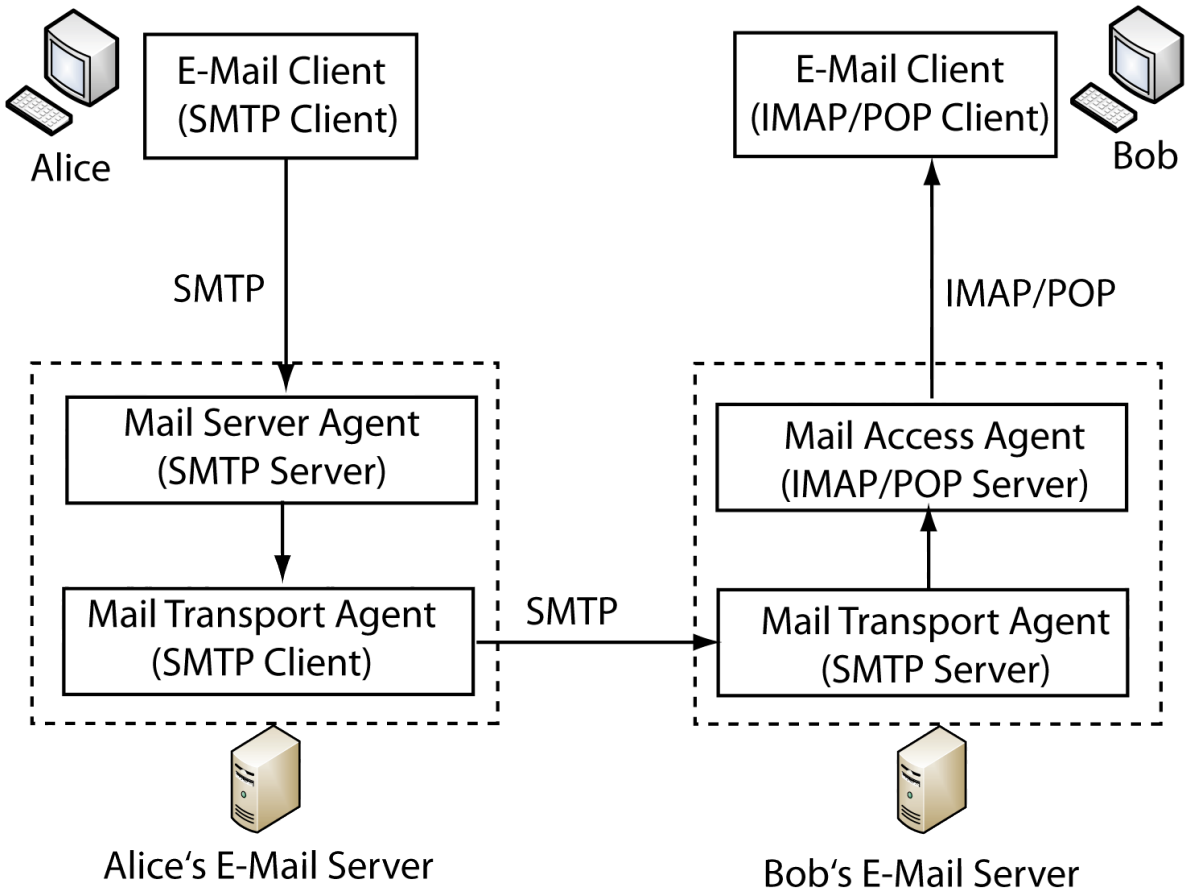
WiSe 18/19

Chair of IT Security

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 sends email to Bob
- Note: sending email is asynchronous

Protocols

- Simple Mail Transfer Protocol (SMTP)
  - Used by Alice to sent 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
  - Confidentiality and authenticity 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 protect connection to mail server when sending mail or retrieving mail from mailbox

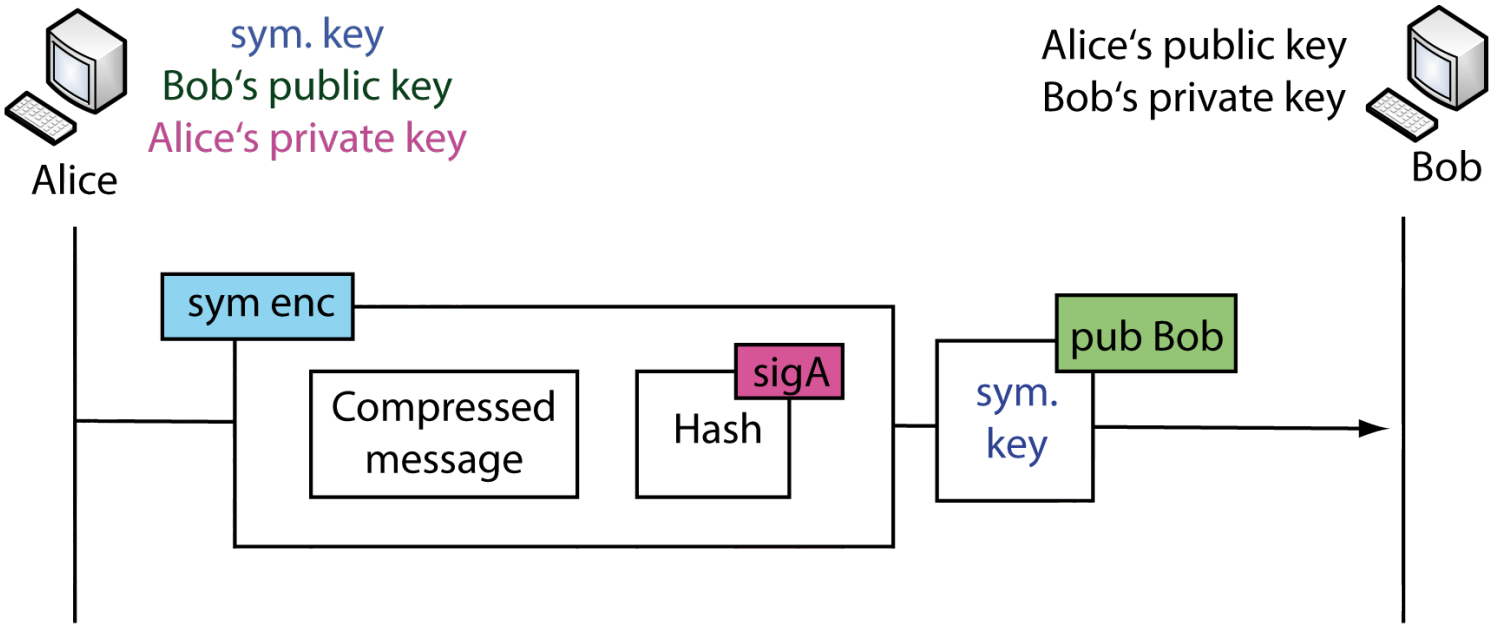
End-to-End Email Security Goals

- 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
  - Protection from denial of sender having sent email

End-to-End Email Protection Approach

- Encryption of email with symmetric encryption algorithm
- Symmetric encryption key selected by sender
- Encryption of symmetric encryption key with public key of receiver
- Sender includes in the message sent to receiver:
  - Identifiers of cryptographic algorithms used
  - Encrypted symmetric key
- Sender has to be able to obtain authentic copy of public key of receiver
- E.g. used by PGP and S/MIME

PGP Overview



- If Alice sends an email to Bob she
  - Hashes the message and signs the hash with her private key
  - Compresses the message
  - Encrypts the compressed message and the signed hash with a symmetric key
  - Encrypts the 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/pubic 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 Certificates

- 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 users 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 three trust levels
  - Full trust
  - Partial trust
  - No trust

Certificate Trust Level

- Trust level Alice assigns to each certificate she receives
- Corresponds to the trust level of the introducer
  - I.e. if Alice fully trusts the introducer Bob and Alice receives a certificate issued by Bob for John then Alice assigns full trust to Johns certificate
- Note that Alice may receive multiple certificates for the same public key from several different introducers

Key Legitimacy

- Determined as sum of weights on the trust level of certificates
- A public key can be used if it's legitimacy is one
- E.g. Alice may assign a weight of
  - 1 to fully trusted certificates
  - ½ to partially trusted certificates
  - 0 to untrusted certificates
- Then if Alice has two partially trusted certificates for the public key of John, she can use that public key as its legitimacy will be one
- Is this a reasonable model to compute key legitimacy?

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

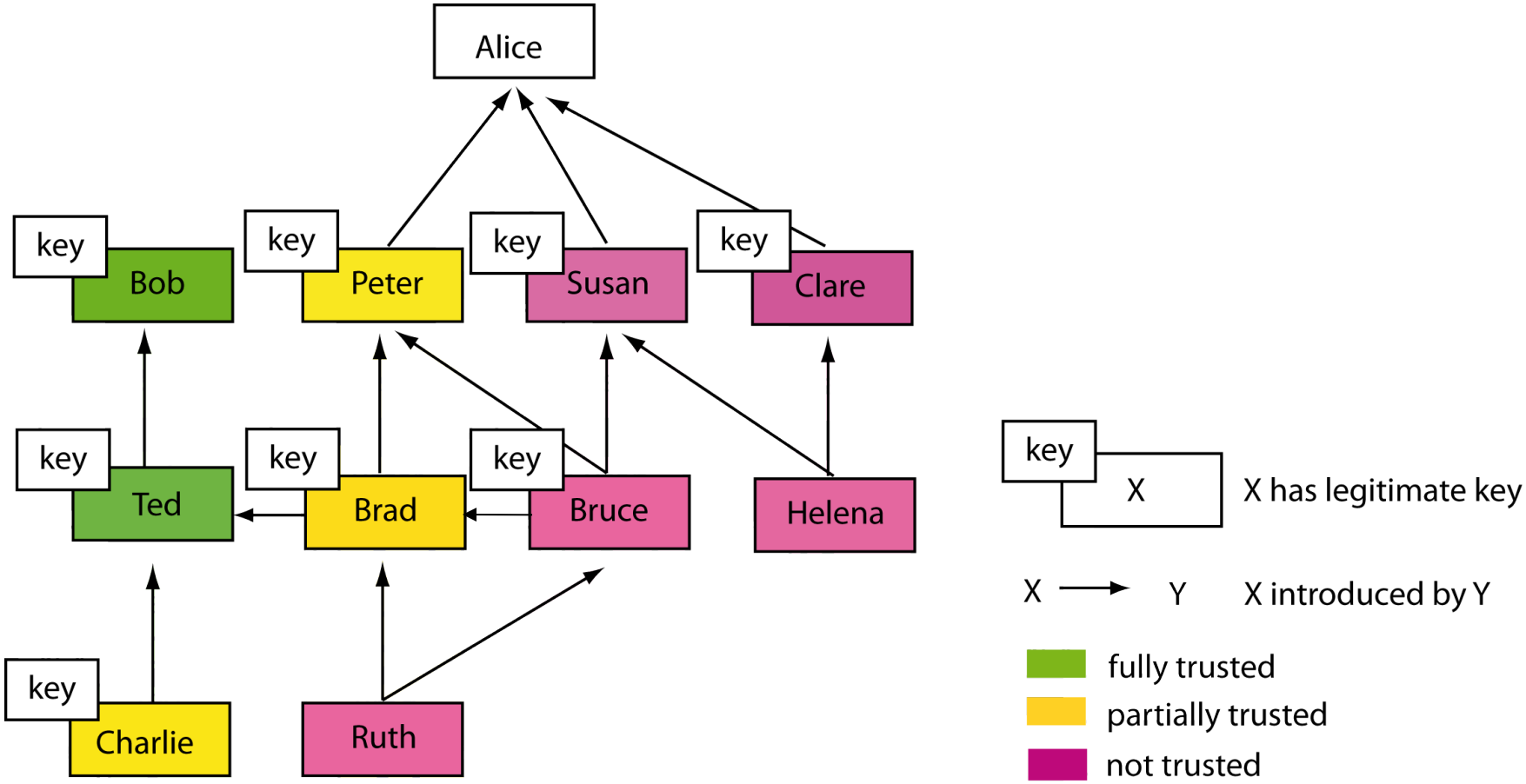
- PGP defines the following structure for the public key ring table each user stores

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 – computed from the certificate trust levels
- Timestamp – data and time of row entry creation

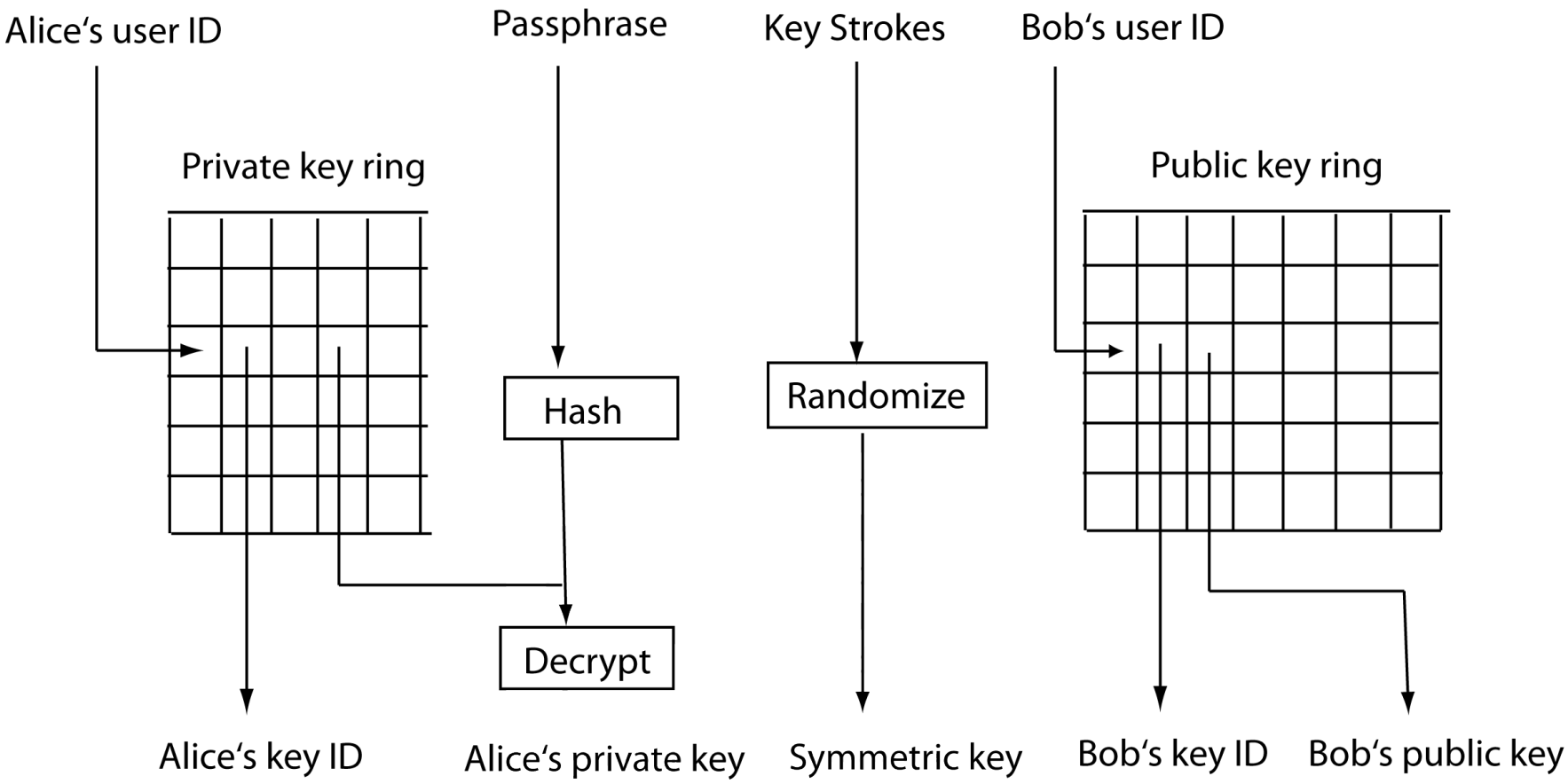
Trust Model for Alice in PGP



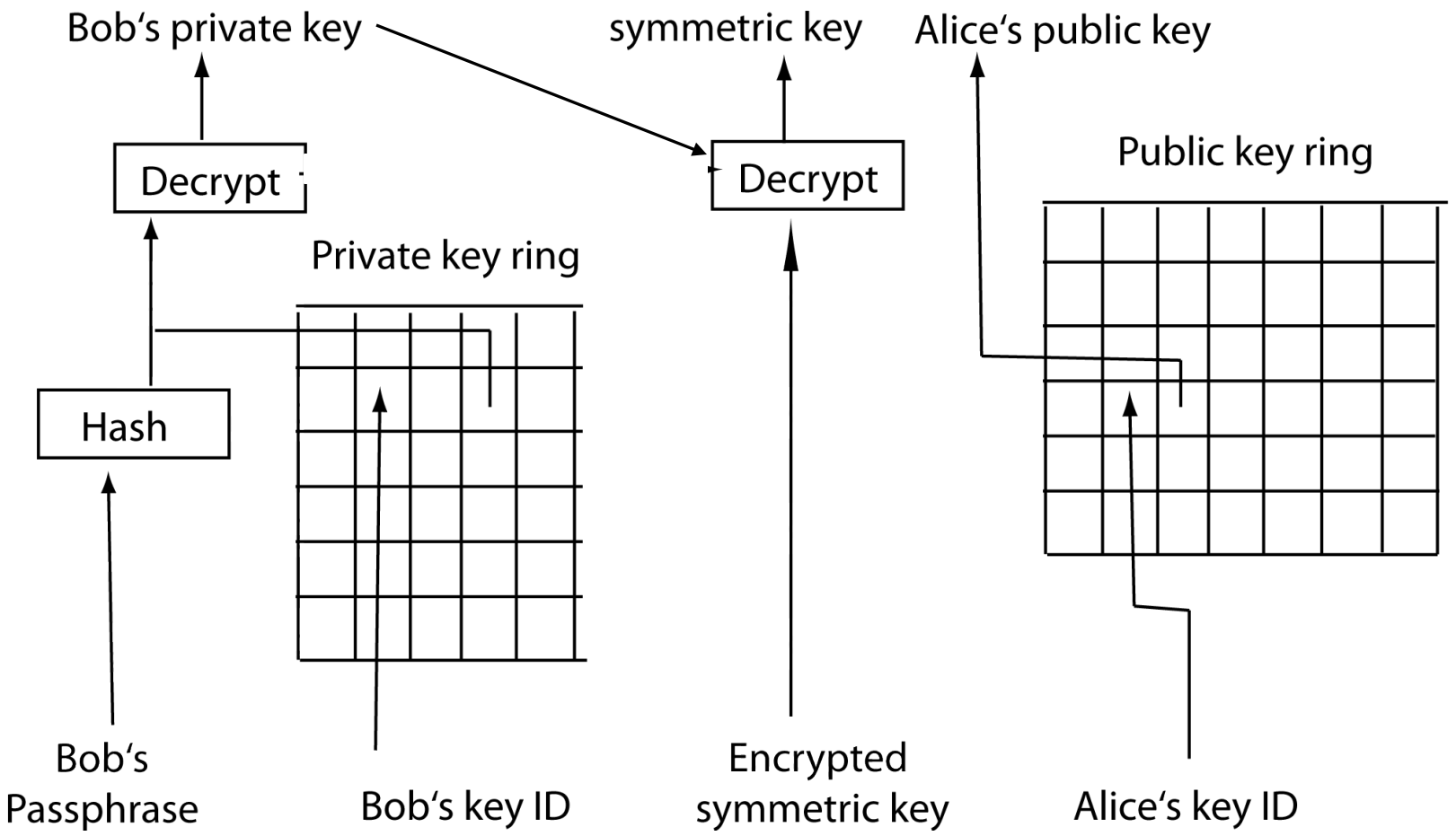
Key Revocation

- To revoke a key the owner disseminates a revocation certificate signed by the old key
- The revocation certificate needs to be disseminated 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?

PGP Processing on Sender Side



PGP Processing on Receiver Side



PGP Messages and Packets

- A PGP message consists of one or more packets
  - E.g. a signed messages is made up of a literal packet that carries the data to be signed and a signature packet that carries the signature and indicates the algorithms used
- Each packet starts with a generic header
- In the latest PGP version the header has two fields
  - A 1 byte tag field that indicates the packet type
  - A 1, 2, or 5 byte length field
    - Length of length field depends on the packet type and can be calculated from the first byte of the length field

Commonly Used Packet Types

Tag Value	Packet Type
1	Session key packet encrypted with a public key
2	Signature packet
5	Private-key packet*
6	Public-key packet
8	Compressed data packet
9	Data packet encrypted with a secret key
11	Literal data packet
13	User ID packet

* Can be used to delegate a private/public key pair to someone else, the private key is than symmetrically encrypted

PGP Literal Data Packet

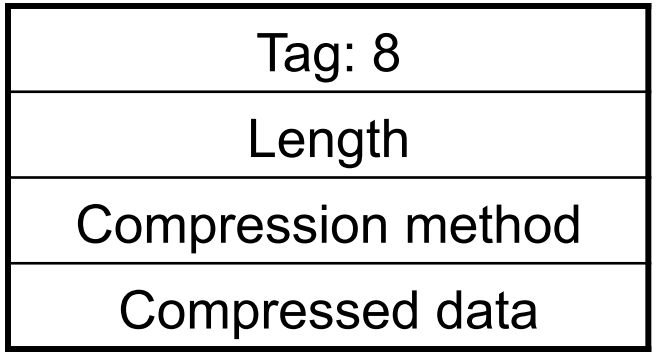
- Packet that holds the actual data that is being transmitted
- Most elementary packet type, cannot include any other packet

Tag: 11
Length
Mode (binary or text)
Length of next field
File name
Timestamp
Literal data

- Mode: binary, text or any other locally defined value (1 byte)
- Length: 1 byte
- File name: ASCII string of variable length (variable)
- Timestamp: four-byte field, time of last modification of packet
- Literal data: message or file

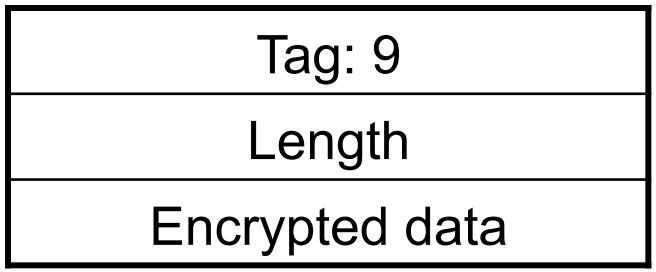
PGP Compressed Data Packet

- Carries compressed data packets



PGP Encrypted Data Packet

- Carries one or more packets that have been encrypted using a symmetric encryption algorithm
- Must be pre-pended by a packet carrying the symmetric key



PGP Signature Packet

- Carries the signature of a packet

Tag: 2
Length
Version
Length
Signature type
Timestamp
Key ID
Public-key algorithm
Hash algorithm
First two bytes of hash
Signature

- Signature type: indicates type of data signed, e.g. binary document, certificate, certificate revocation, text document,…
- Timestamp: four-byte field, time when signature was calculated
- Public key algorithm: indicates signature algorithm
- Hash algorithm: indicates hash algorithm
- First two bytes of hash: used as check sum, ensure that receiver is using the right key ID to decrypt the signed hash

PGP Session-Key Packet

- Used to send the symmetric key encrypted with the receivers public key

Tag: 1
Length
Version
Key ID
Public-key algorithm
Encrypted session key

- Key ID: 64 LSBs of public key
- Public key algorithm: indicates algorithm used to encrypt session key
- Encrypted session key: symmetric key, symmetric key algorithm identifier and checksum encrypted with public key of receiver

PGP Public Key Packet

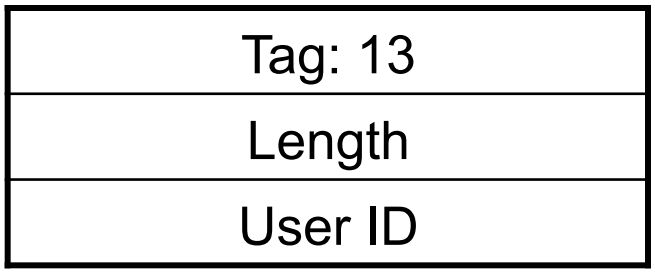
- Contains the public key of the sender

Tag: 6
Length
Timestamp
Validity
Public-key algorithm
Public key

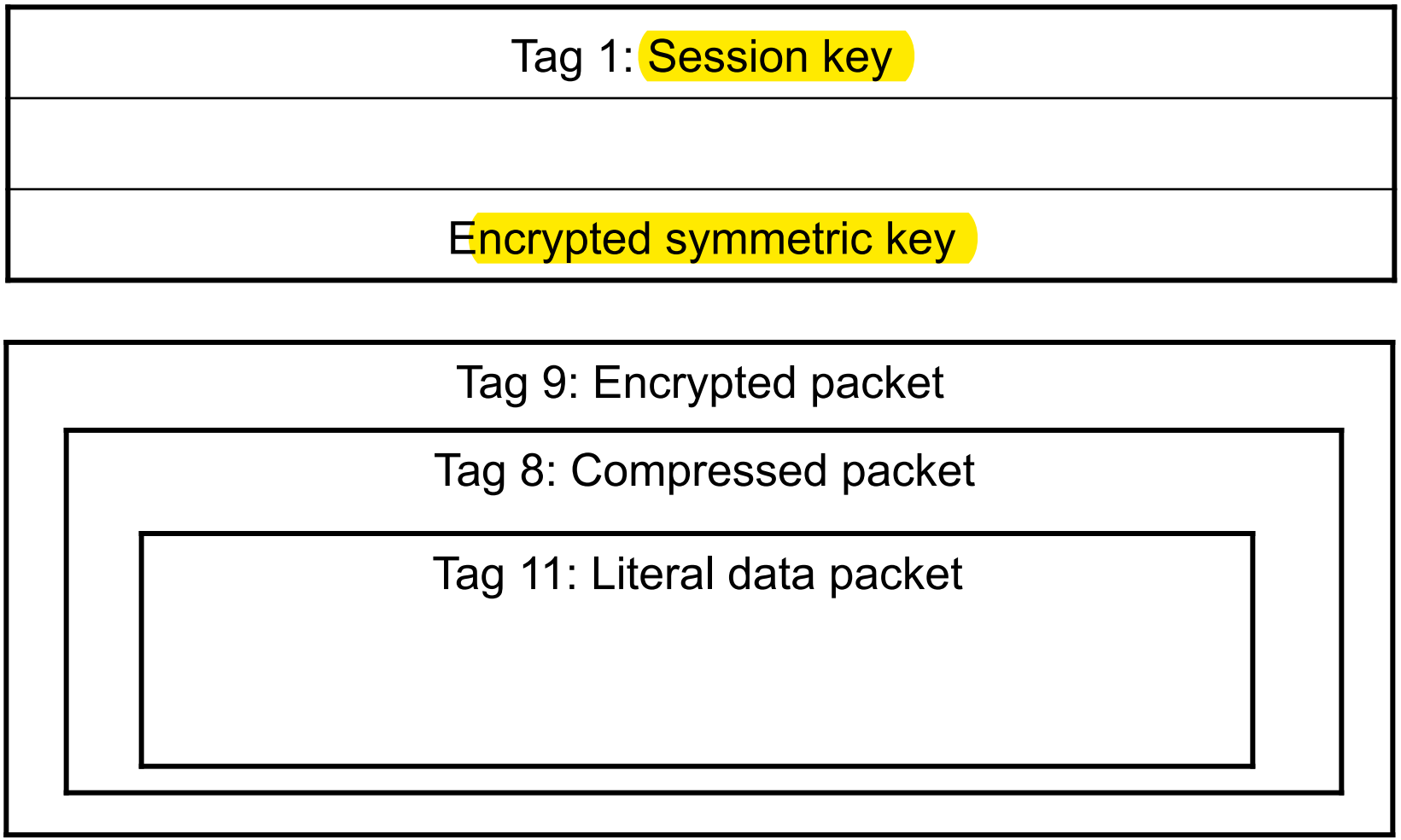
- Timestamp: indicates time the packet was created
- Validity: Number of days the public key is valid
- Public key algorithm: indicates algorithm for which the public key can be used
- Public key: public key

User ID Packet

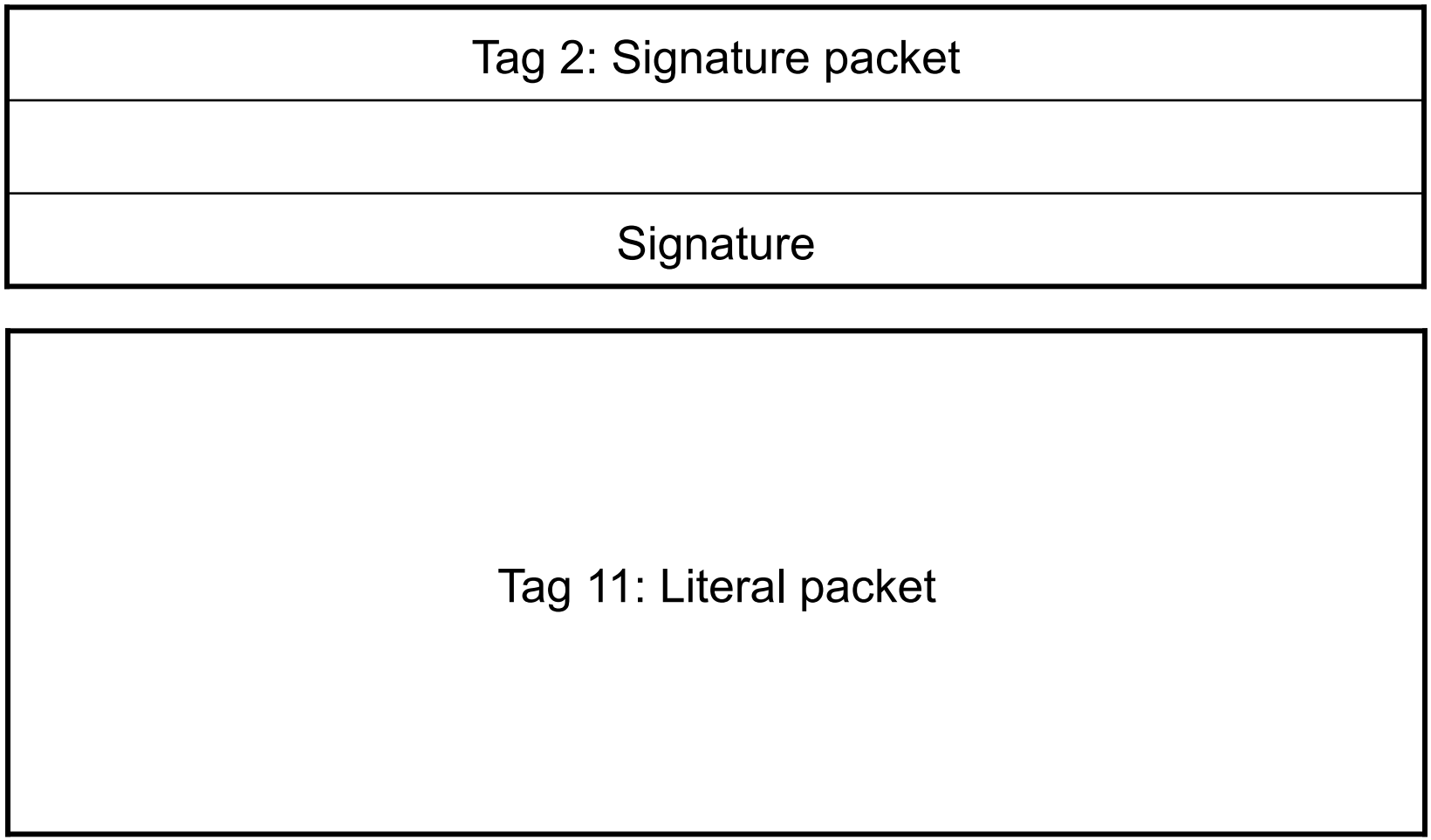
- Identifies a user
- Typically contains the name of the user and its email address



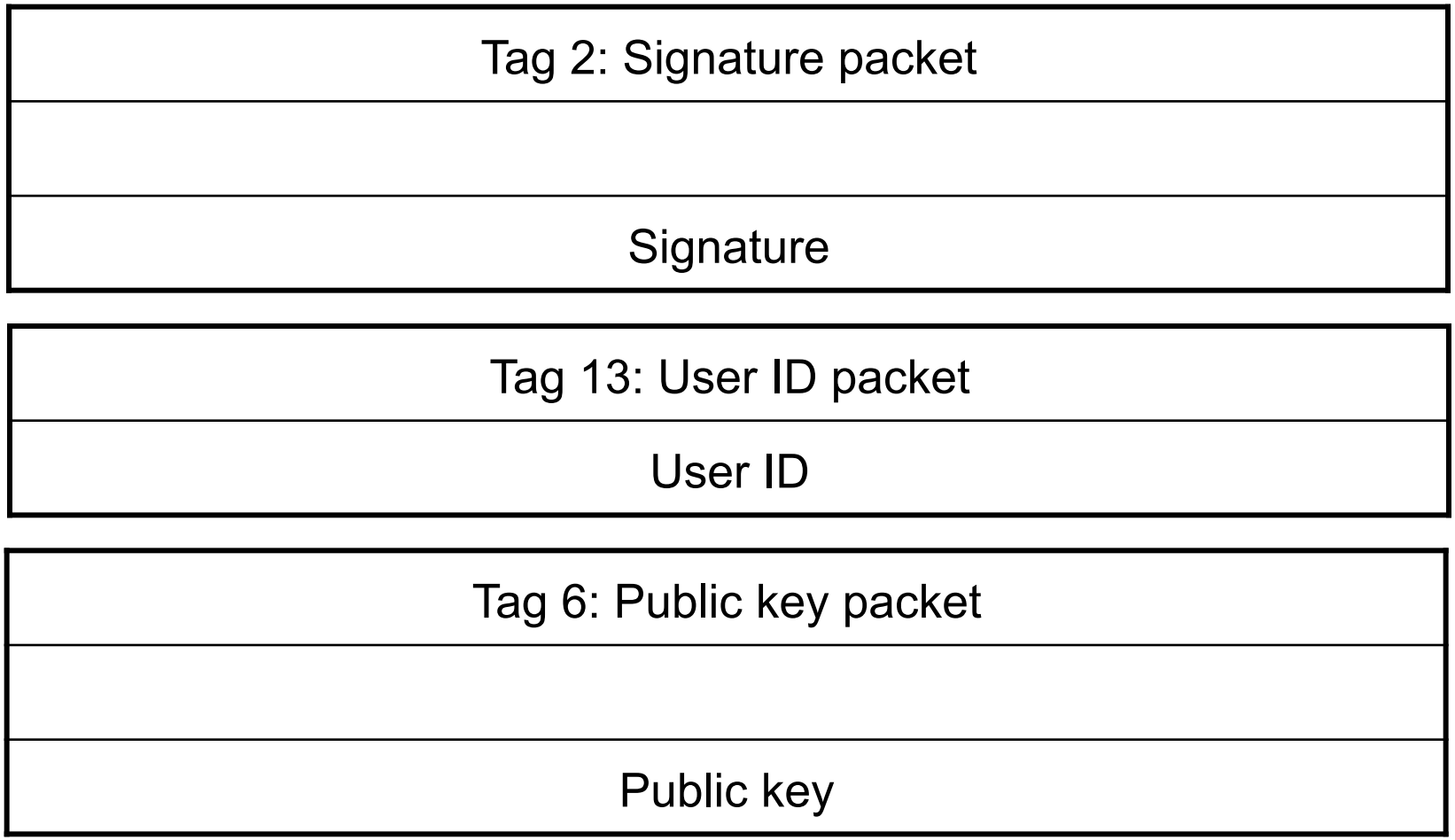
Example: PGP Encrypted Message



PGP Signed Message



PGP Certificate Message



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
- PGP/MIME 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
  - MIME = Multipurpose Internet Mail Extension
- 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 identifiers
    - 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 encryption algorithms 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		

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