

IT security



Monday 1st April
Course day #8

Theme D (i)
Internet security

Case: Are websites protected against attacks on TLS?

Niels Christian Juul (ncjuul@ruc.dk)
Niels Jørgensen (nielsj@ruc.dk)

Theme D: Network security

A. Computer security technology and principles (Part One in Stallings & Brown)	11 th Feb 18 th Feb 25 th Feb	
B. Software and system security (Part Two)	11 th March (#5)	
C. Management issues (Part Three)	4 th March (#4) 18 th March 25 th March	
D. Network security (Part Five)	1 st April 8 th April	 Internet security (Ch. 22)  Wireless security (Ch. 24)

Literature and exam questions for today

Stallings & Brown:

- 22.1: Secure email
- 22.3: Secure sockets
- 22.4: Secure web browsing

Additional mandatory literature:

- Empirical analysis of SSL/TLS weaknesses in real websites: Who cares? (12 pages)
- TCP/IP Attacks, Defenses and Security Tools (7 pages)

Exam questions

- "TCP SYN attacks - and how to defend against them"
- "TLS POODLE attacks - and how to defend against them"

Plan for today



Secure email (S/MIME)

Internet basics (TCP/IP)

- Student presentation (Daniel):
TCP/IP attacks

Secure sockets (SSL/TLS)

- Student presentation (Lucas):
SSL/TLS weaknesses in real websites

Practical exercise with nmap

Internet security

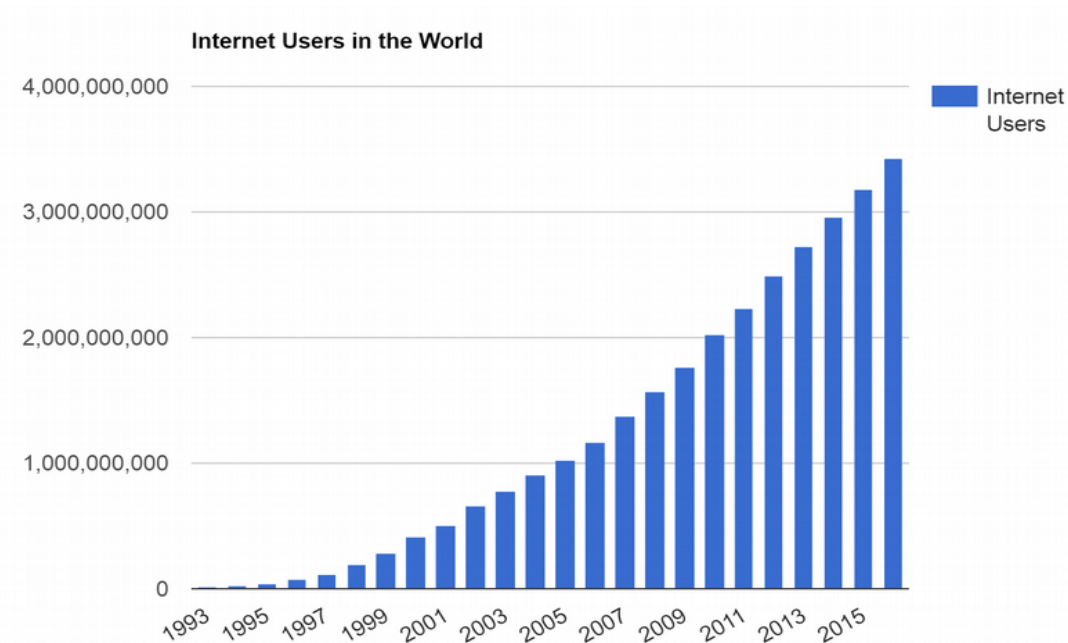
“This chapter looks at some of the most widely used and important security protocols and standards.” (Stallings & Brown, p 683)

We have already covered aspects of the Internet

- confidentiality
 - encryption of transmitted data is important because data is transmitted on *open* networks
- authentication
 - secure logon is important because many services are available on the internet

Now we will focus specifically on some protocols and other standards

- public protocols and standards are essential for communication



S/MIME

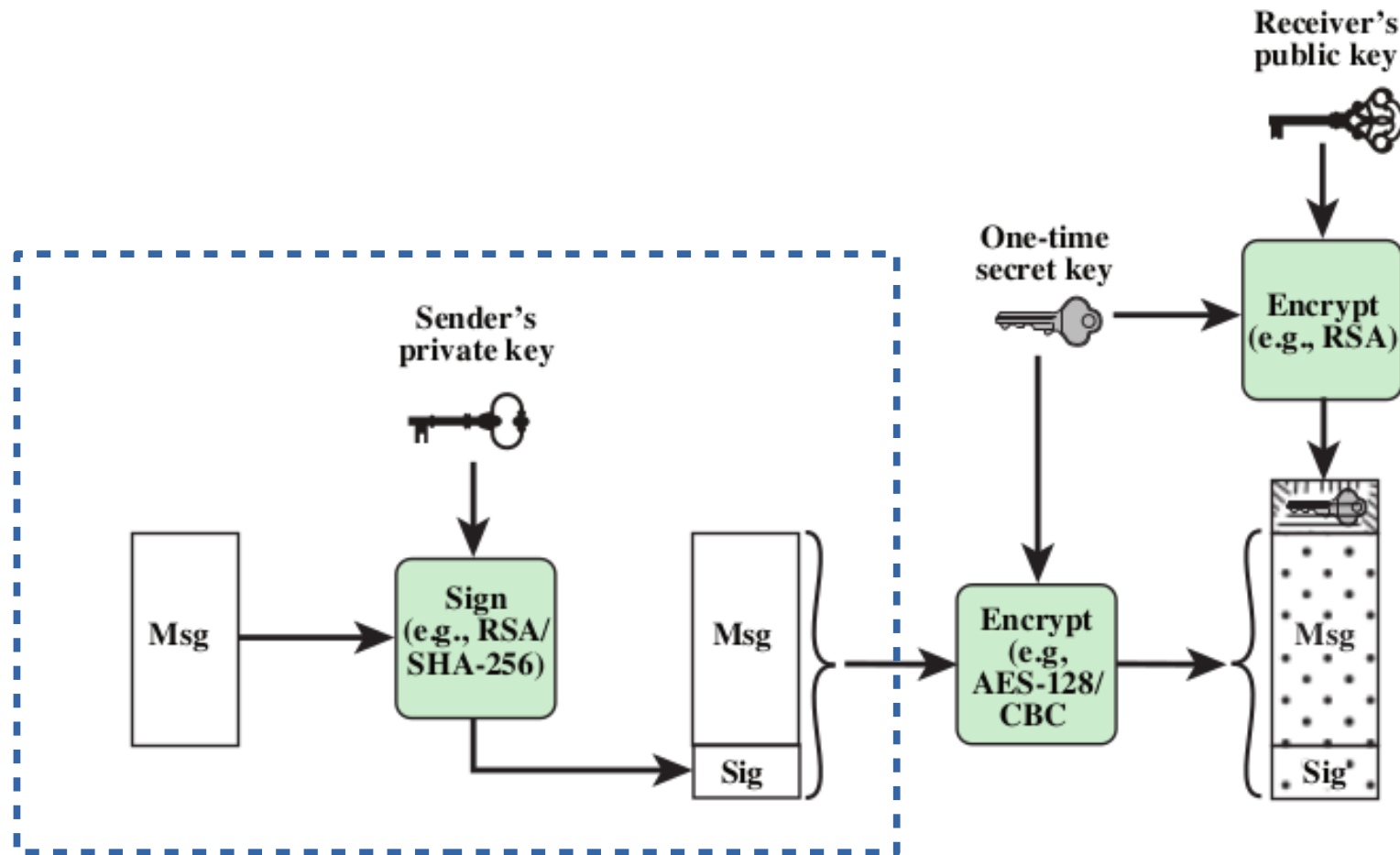
S/MIME is a standard for signing and encrypting email

- similar to, but not the same as PGP

Learning goals

- understand how secure email uses symmetric and asymmetric encryption
- understand how S/MIME is defined
 - by RFC 5751 and other RFCs
 - a standard (specifying message formats)
 - but not a protocol (in the sense of specifying actions, such as TCP/IP)

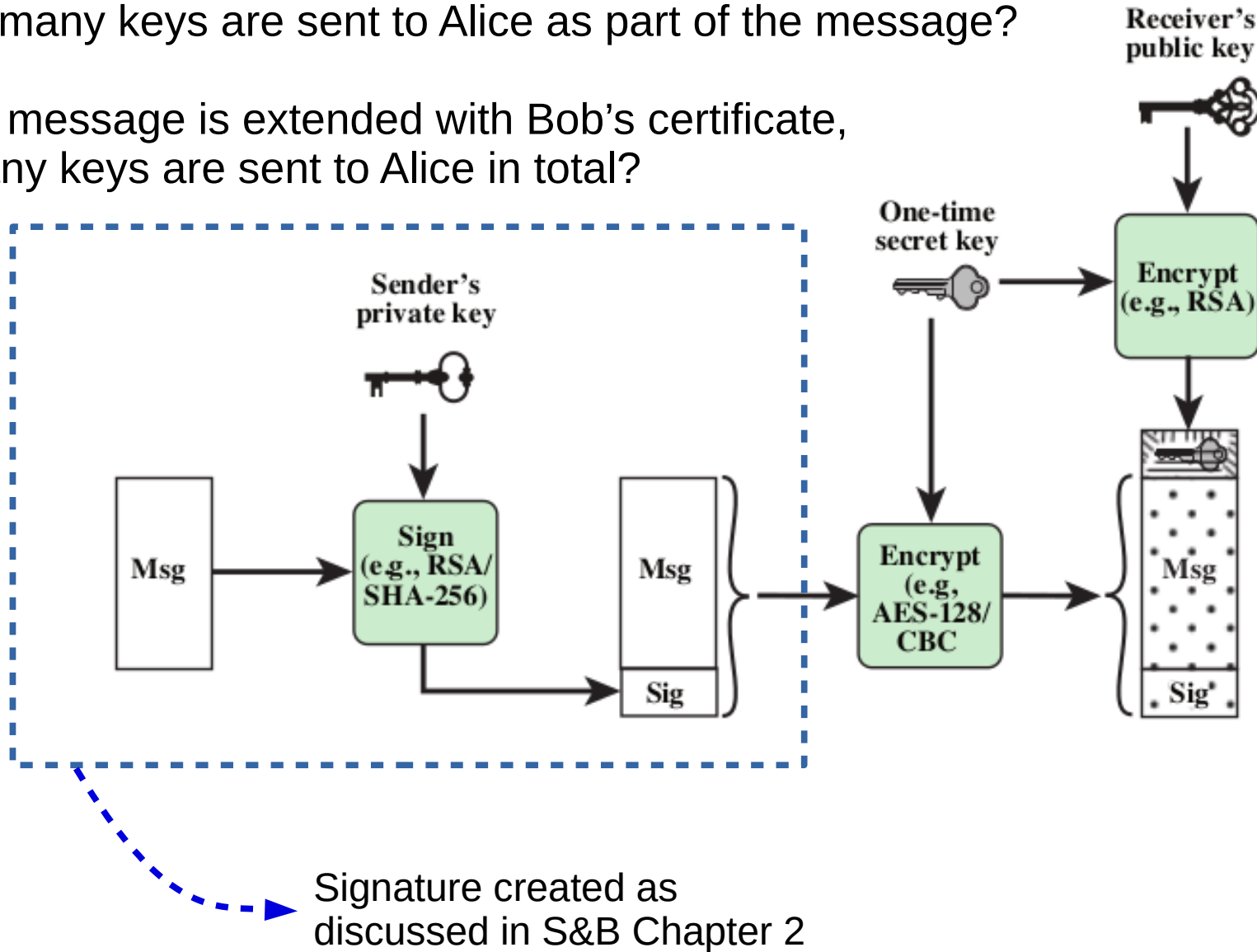
Bob (sender) --> Alice (receiver): a signed and encrypted mail message



Signature created as
discussed in S&B Chapter 2

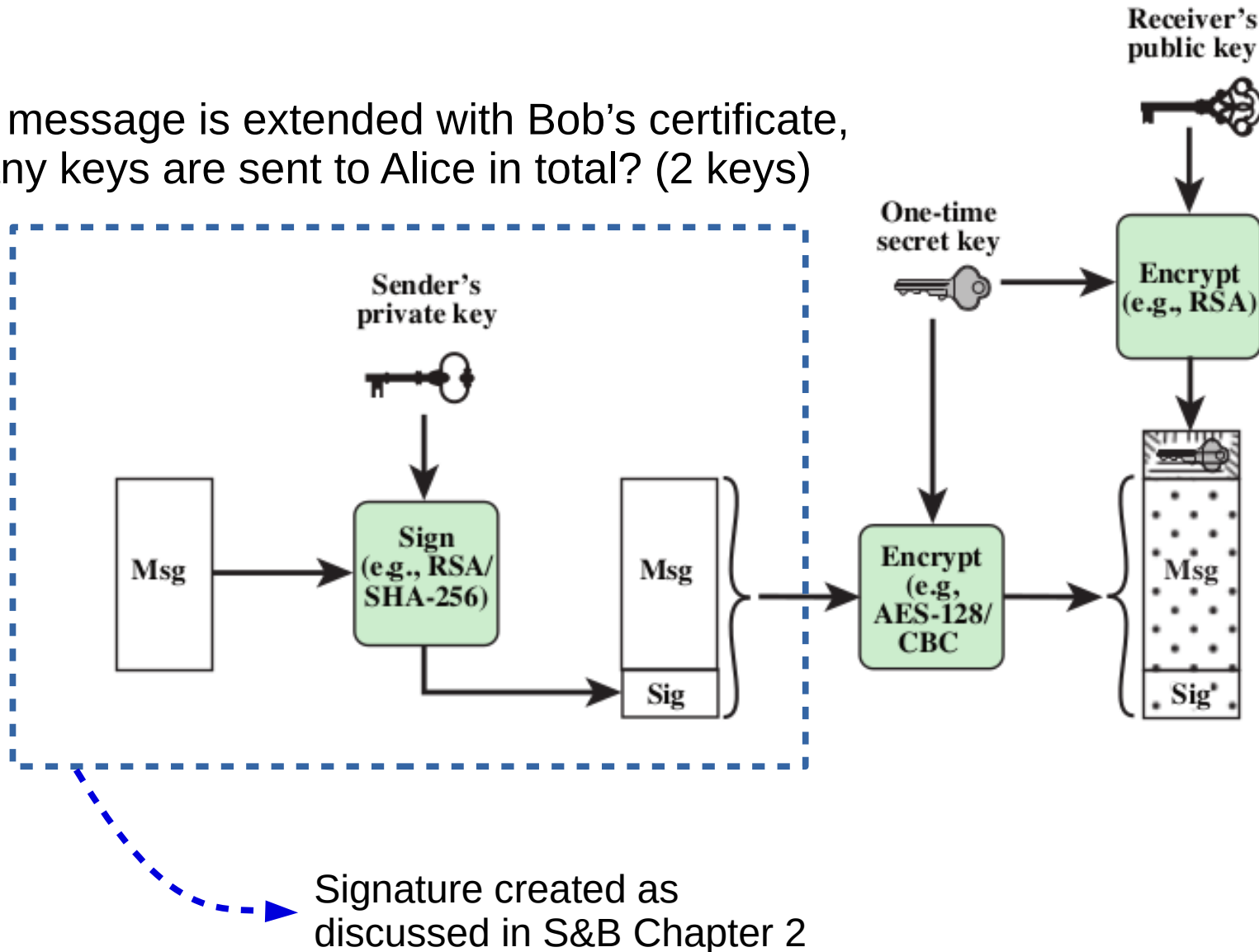
Exercise

- 1) How many keys were used by Bob?
(to create the message, using Bob's software)
- 2) How many keys are sent to Alice as part of the message?
- 3) If the message is extended with Bob's certificate, how many keys are sent to Alice in total?

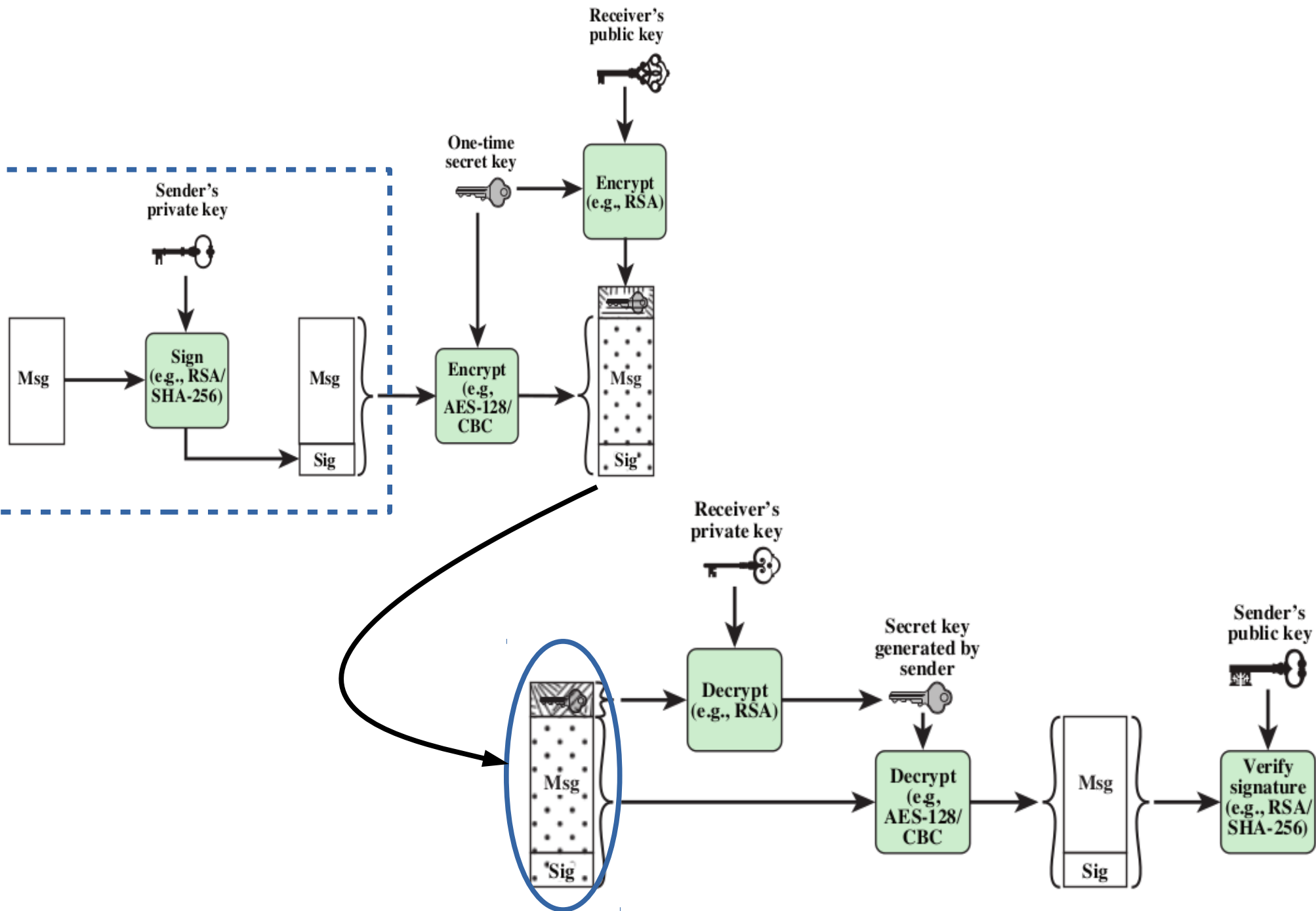


Exercise solution

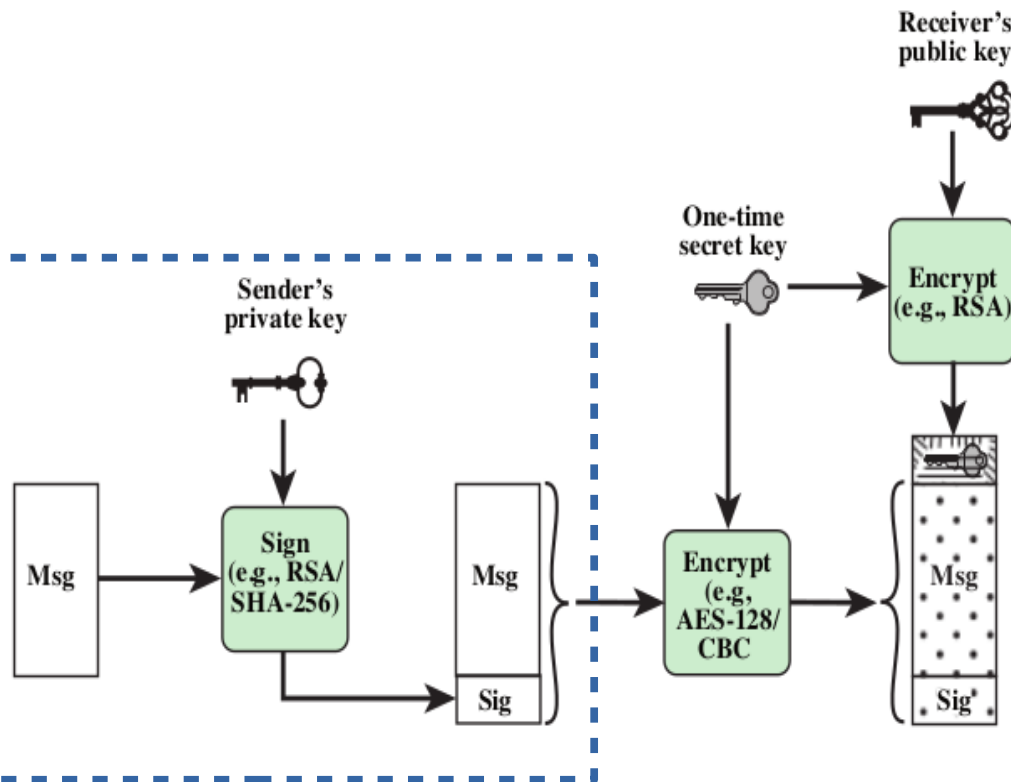
- 1) How many keys were used by Bob? (3 keys)
- 2) How many keys are sent to Alice as part of the message? (1 key)
- 3) If the message is extended with Bob's certificate, how many keys are sent to Alice in total? (2 keys)



How can we describe the final message?



How can we describe the final message?



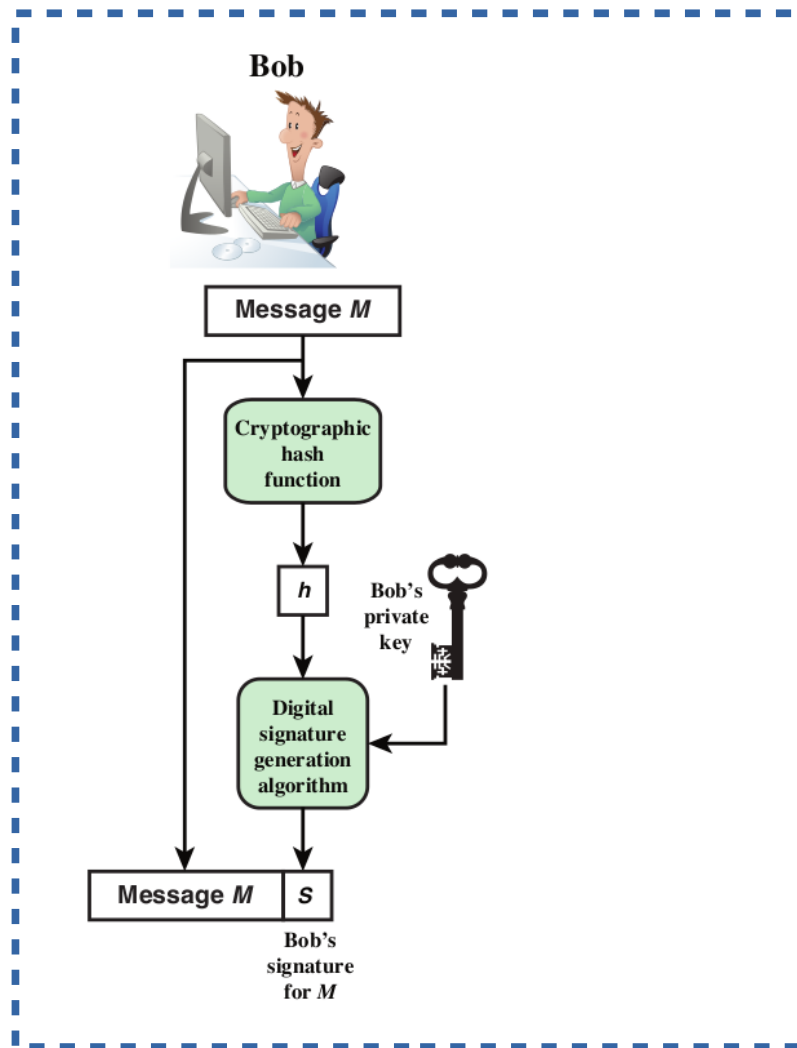
Keys

- Bob's keys
 - public key: PU_b
 - private key: PR_b
- Alice's keys
 - public key: PU_a
 - private key: PR_a
- The symmetric key: k

Functions

- Hash algorithm: $h(..)$
- Asymmetric encryption: $E2(.., ..)$
- Asymmetric decryption: $D2(.., ..)$
(2 for "two key")
- Symmetric encryption: $E1(.., ..)$
- Asymmetric decryption: $D1(.., ..)$
(1 for "single key")

Describing the signed message

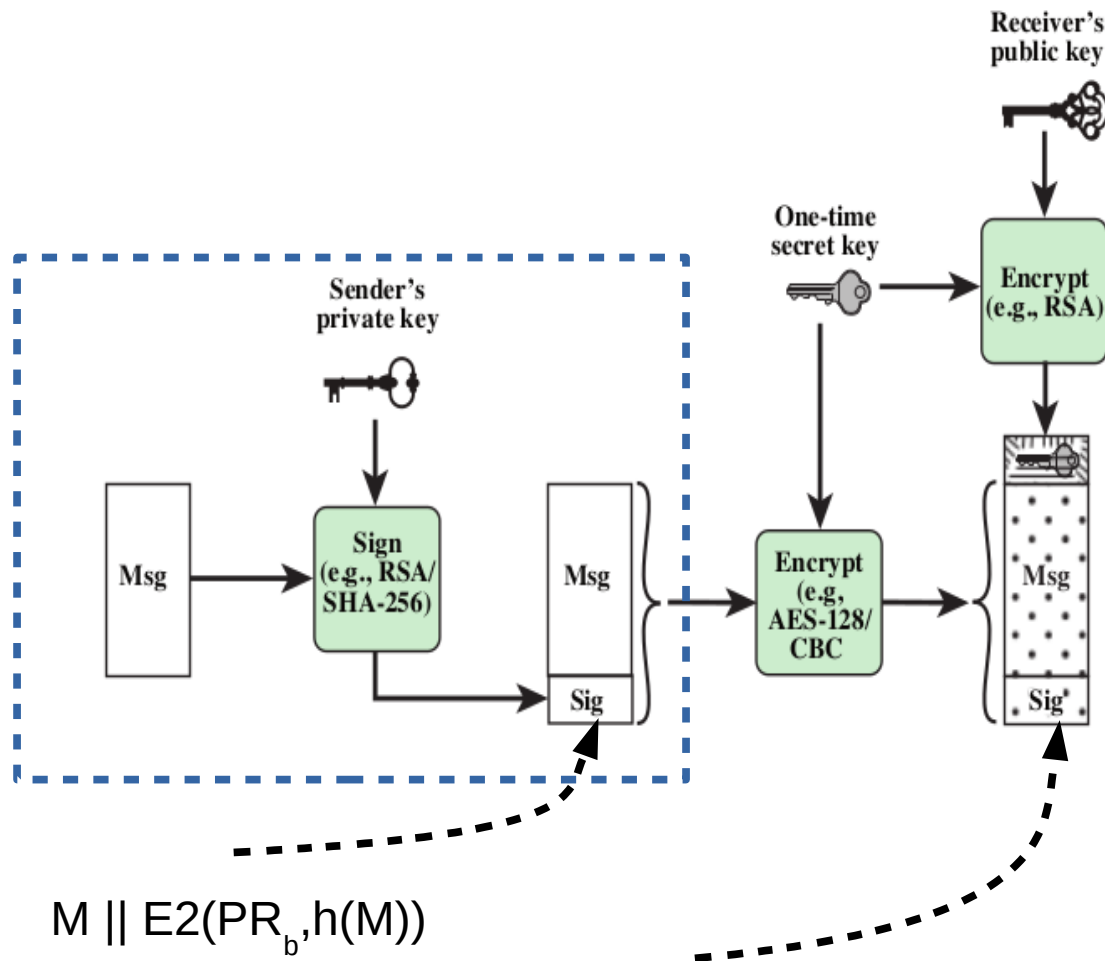


Signed message:

$$M \parallel S$$
$$= M \parallel E2(PR_b, h(M))$$

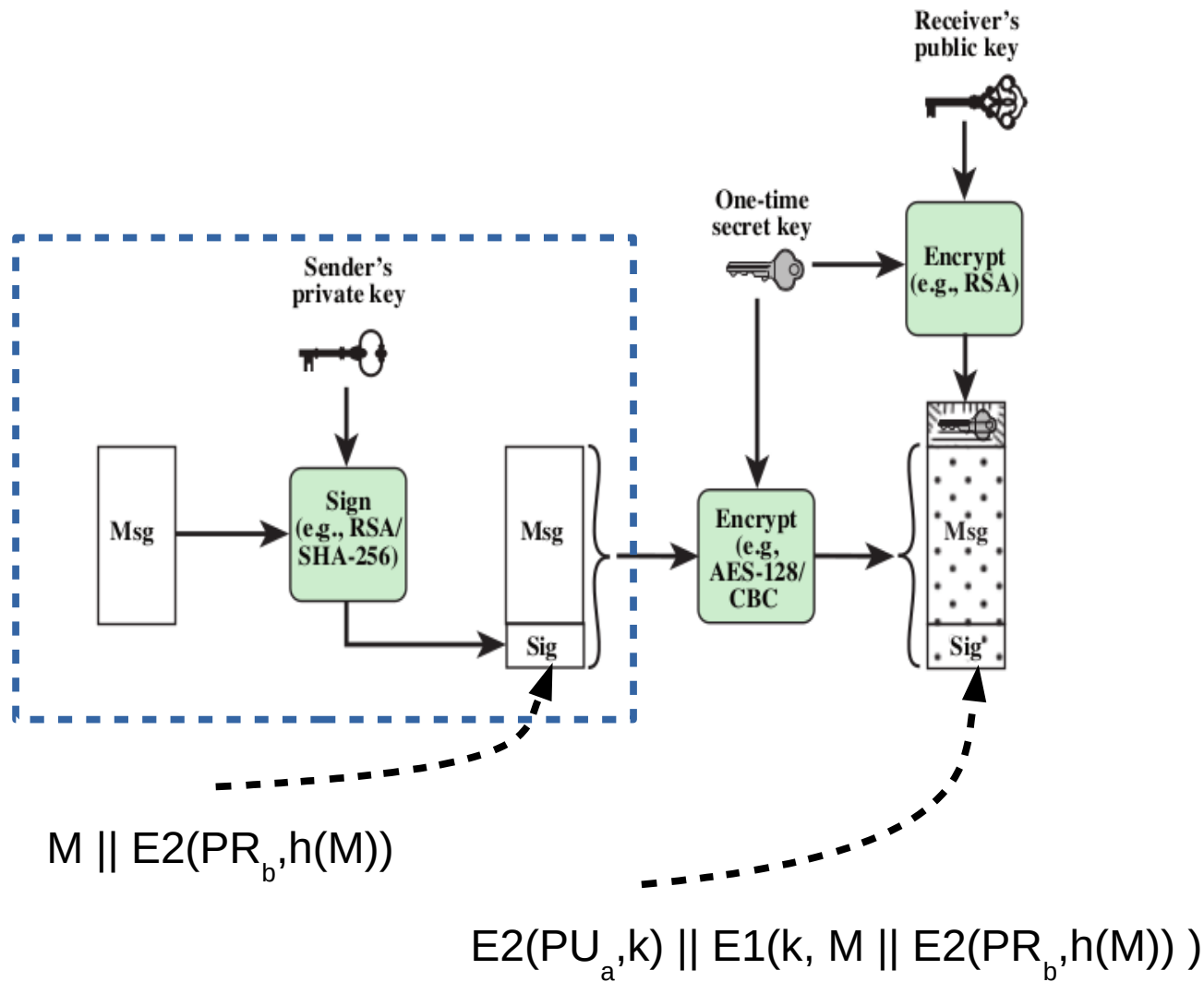
Signature created as
discussed in S&B Chapter 2

Exercise: mail message (symbols)

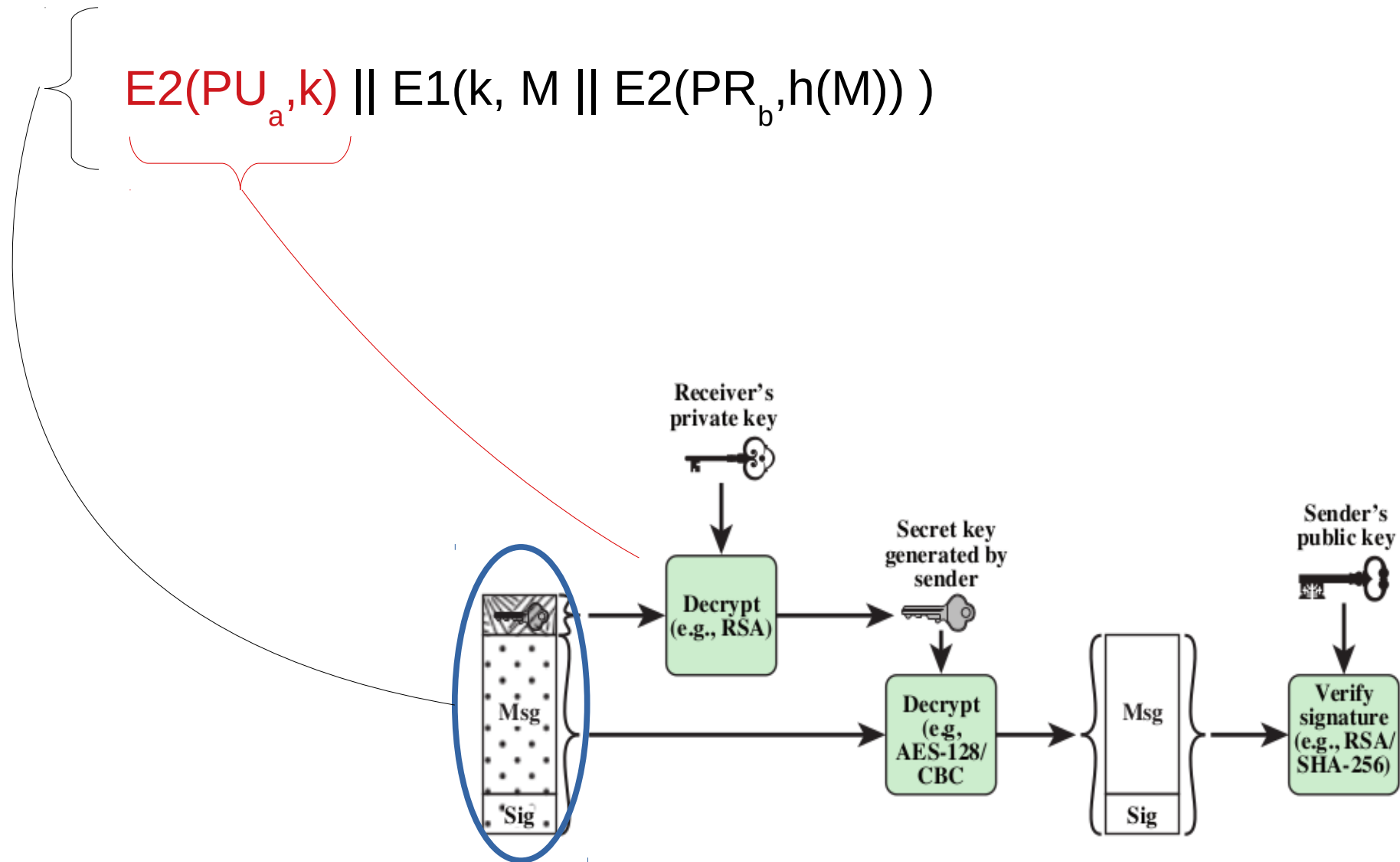


*Exercise:
describe the mail message using symbols*

Exercise solution



Now that we understand the message,
we can explain how Alice must treat it



For Alice to process the mail message, Alice must be able to identify its parts



Therefore, the S/MIME standard must specify a syntax for the parts of a message

- keys, encrypted keys
- messages
- signatures

and specify how to identify the parts

Example signed mail message

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

FreeBSD-SA-17:07.wpa
Security Advisory The FreeBSD Project

I. Background

....
The latest revision of this advisory is available at
URL:[https://security.FreeBSD.org/advisories/
FreeBSD-SA-17:07.wpa.asc](https://security.FreeBSD.org/advisories/FreeBSD-SA-17:07.wpa.asc)>
-----BEGIN PGP SIGNATURE-----

iQKTBAEBCgB9FiEEHPf/b631yp+G4yy7Wfs1l3Pa
....
ZhMb/V4WmWV+4WnLKPwCQZ9fimKA==aNWn

-----END PGP SIGNATURE-----

freebsd-announce@freebsd.org mailing list
<https://lists.freebsd.org/mailman/listinfo/freebsd-announce>
To unsubscribe, send any mail to
"freebsd-announce-unsubscribe@freebsd.org"

The cleartext signed message consists of:

- The cleartext header '-----BEGIN PGP SIGNED MESSAGE-----' on a single line,
- One or more "Hash" Armor Headers,
- Exactly one empty line not included into the message digest,
- The dash-escaped cleartext that is included into the message digest,
- The ASCII armored signature(s) including the '-----BEGIN PGP SIGNATURE-----' Armor Header and Armor Tail Lines.

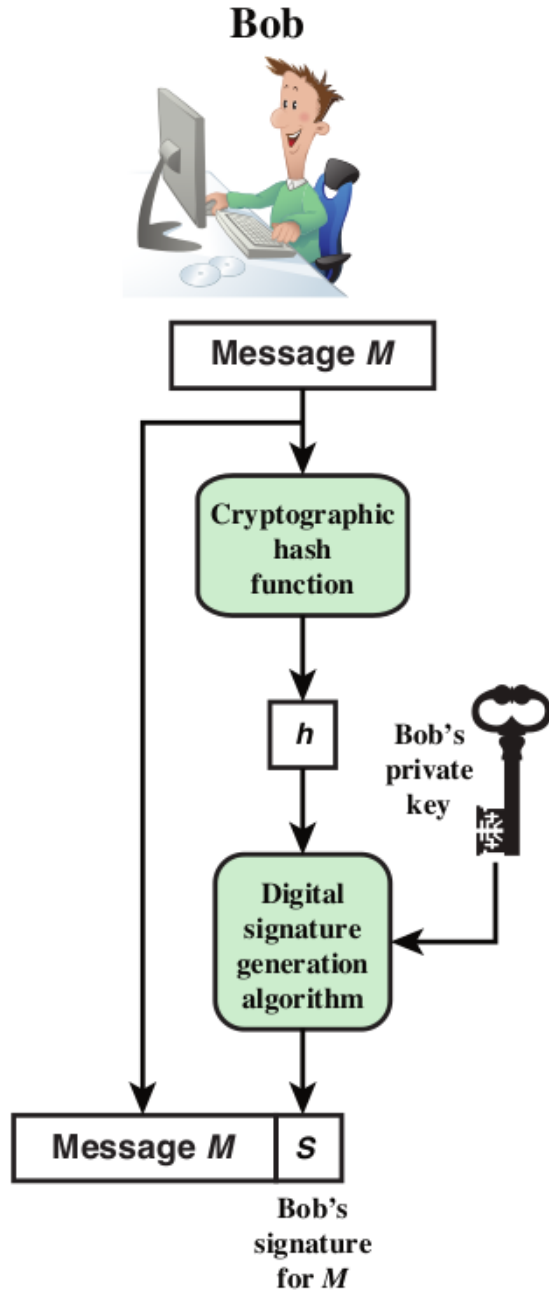
Message
part

Signature
part

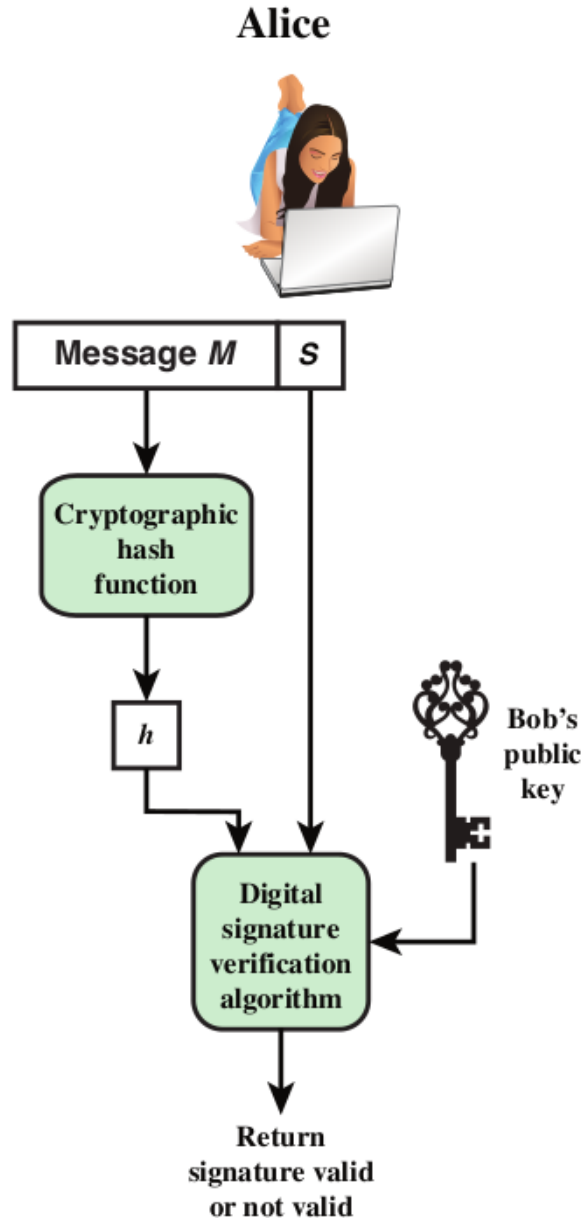
This format is for PGP
as defined by RFC 4880 etc.

- different from S/MIME
- but similar

Digital signatures



(a) Bob signs a message



(b) Alice verifies the signature

You may recall an exercise on course day #3

In the exercise, I claimed that the order of hashing and asymmetric encryption could be reversed

- if the order on the receiving side was reversed as well
- but that is not true

Plan for today

Secure email (S/MIME)



Internet basics (TCP/IP)

- Student presentation (Daniel):
TCP/IP attacks

Secure sockets (SSL/TLS)

- Student presentation (Lucas):
SSL/TLS weaknesses in real websites

Announcement

(from RUC's newsletter Friday 29th March)

Evan Selinger gives a lecture from 9.30 to 11.30 in building 41.1-14
"Biografen":

9.30 – 10:30: Talk, followed by discussion.

Date: Monday 15th April

Don't Re-Engineer Humanity With Facial Recognition Technology

In this talk, I'll propose [...] that it should be impermissible for liberal democracies to legalize facial recognition technology. [...] Like nearly all technologies, facial recognition systems are dual-use, capable of furthering good and bad ends alike.

TLS and SSL

TLS = “Transport Layer Security”

A standard for encryption, integrity and authentication used widely on the internet, including by webservers/browsers

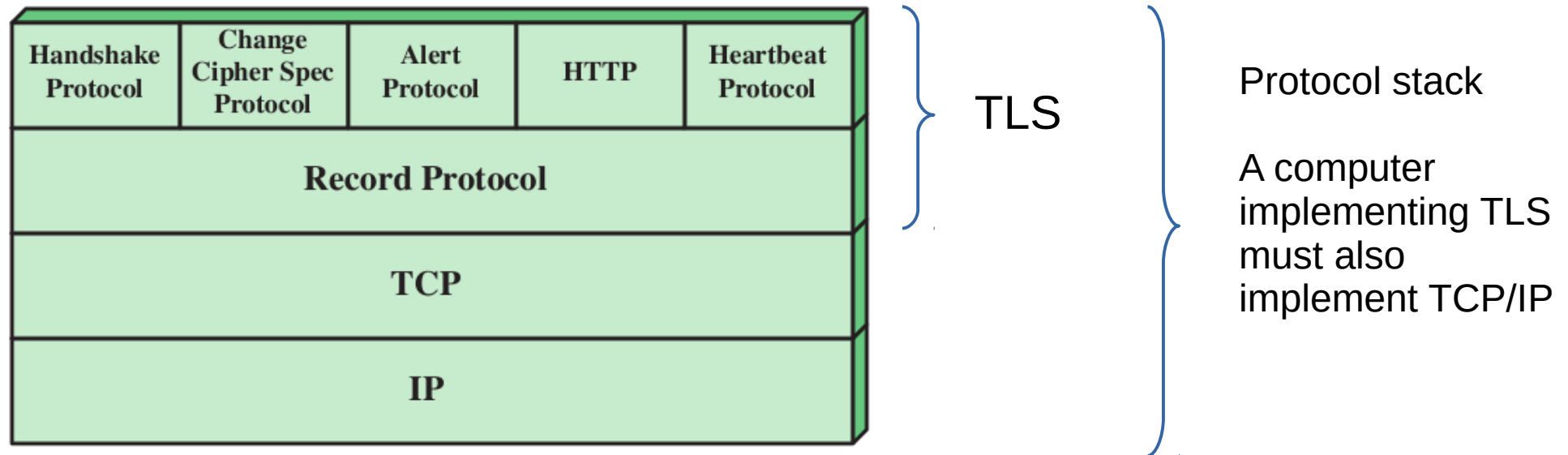
TLS versions:

- Most recent: 1.3, defined in RFC 8446 (august 2018)
- Previous versions are 1.0, 1.1, 1.2

TLS is the successor to SSL

- SSL = “Secure Socket Layer”
- versions 1.0, 2.0 and 3.0 (1994-1996)
- developed by Netscape

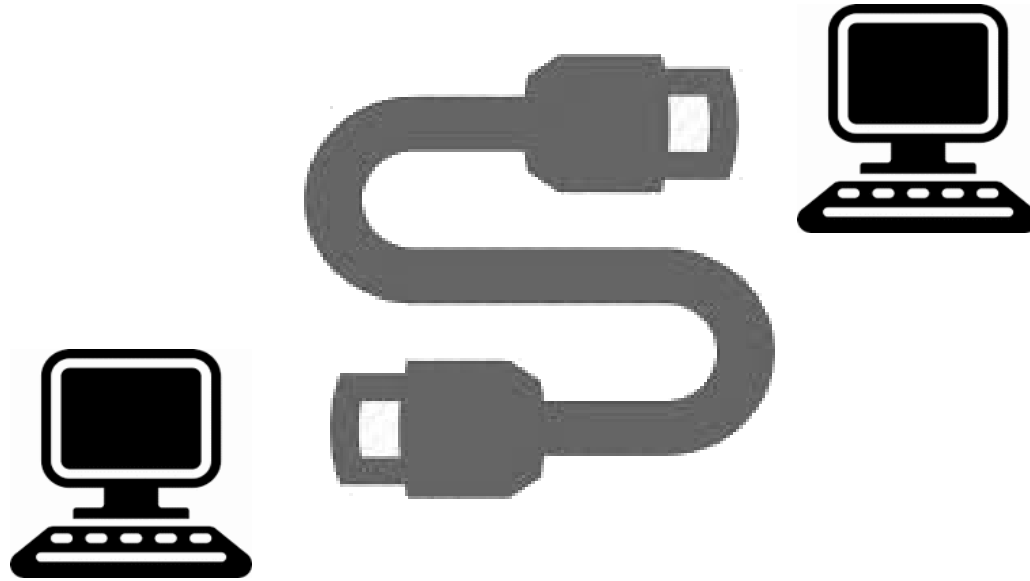
TLS builds on TCP



TLS (secure sockets) builds on TCP (ordinary sockets)

- TLS provides confidentiality and other security goals/services
- TLS then uses TCP and IP to transmit data (encrypted by TLS)
- TCP/IP are the fundamental building blocks of the Internet

So what are ordinary TCP-sockets?



A socket-connection is a connection between two hosts

- a host is a computer connected to the internet

A socket-connection is “reliable”

- checks are made to ensure everything is received
- checks are made to ensure integrity (to some degree)
- data is sent in small packets, and assembled in correct order

There is a socket API with commands for

- opening, sending, receiving, closing
- data can be transmitted in both directions

Setting up a TCP-connection: 3-way handshake



Suppose a computer with a webbrowser (left) wants to send a http-request to a webserver (right)



Then the computer begin by asking for a TCP-connection

I want to setup a TCP connection

SYN

I confirm receipt of SYN and I also want to setup a TCP connection

SYN ACK

I confirm receipt of SYN ACK

ACK

RFC 973 is a protocol defining TCP

- defines syntax of messages
 - eg., "SYN"
- defines actions
 - eg. send "SYN ACK" upon "SYN"

TCP handshake

(Figure 1 in Alqahtani et al.)

No.	Time	Source	Destination	Proto	Length	Info
17	2.31109300	192.168.2.109	173.194.69.120	TCP	66	54010 > http [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
18	2.41225400	173.194.47.120	192.168.2.109	TCP	62	http > 54006 [SYN, ACK] Seq=0 Ack=1 win=4260 Len=0 MSS=1420 SACK_PERM=1
19	2.41241600	192.168.2.109	173.194.47.120	TCP	54	54006 > http [ACK] Seq=1 Ack=1 win=17040 Len=0
20	2.41256800	173.194.69.120	192.168.2.109	TCP	62	http > 54002 [SYN, ACK] Seq=0 Ack=1 win=4260 Len=0 MSS=1420 SACK_PERM=1

Client at 192.168.2.109
sends SYN

Server at 173.194.47.120
sends SYN ACK

Client at 192.168.2.109
sends ACK

Defending against a simple SYN flood attack

No.	Time	Source	Destination	Proto	Length	Info
17	2.31109300	192.168.2.109	173.194.69.120	TCP	66	54010 > http [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
18	2.41225400	173.194.47.120	192.168.2.109	TCP	62	http > 54006 [SYN, ACK] Seq=0 Ack=1 win=4260 Len=0 MSS=1420 SACK_PERM=1
19	2.41241600	192.168.2.109	173.194.47.120	TCP	54	54006 > http [ACK] Seq=1 Ack=1 win=17040 Len=0
20	2.41256800	173.194.69.120	192.168.2.109	TCP	62	http > 54002 [SYN, ACK] Seq=0 Ack=1 win=4260 Len=0 MSS=1420 SACK_PERM=1

Client at 192.168.2.109
sends SYN

- client does not follow up
(does not send ACK or anything else)
- client does not hide its IP-address

A firewall at the server side
can be set up to reject
all SYN-requests from the
clients address

- SYN requests will be deleted
- SYN ACKs will not be sent

Plan for today

Secure email (S/MIME)

Internet basics (TCP/IP)

- Student presentation (Daniel):
TCP/IP attacks



Secure sockets (SSL/TLS)

- Student presentation (Lucas):
SSL/TLS weaknesses in real websites

Secure web browsing (HTTPS)

TLS

TLS provides a secure socket connection

- confidentiality, authentication (checks certificates) and integrity

TLS provides confidentiality in the same way as S/MIME and PGP

- data is encrypted using symmetric encryption
- the key for symmetric encryption is exchanged using asymmetric encryption

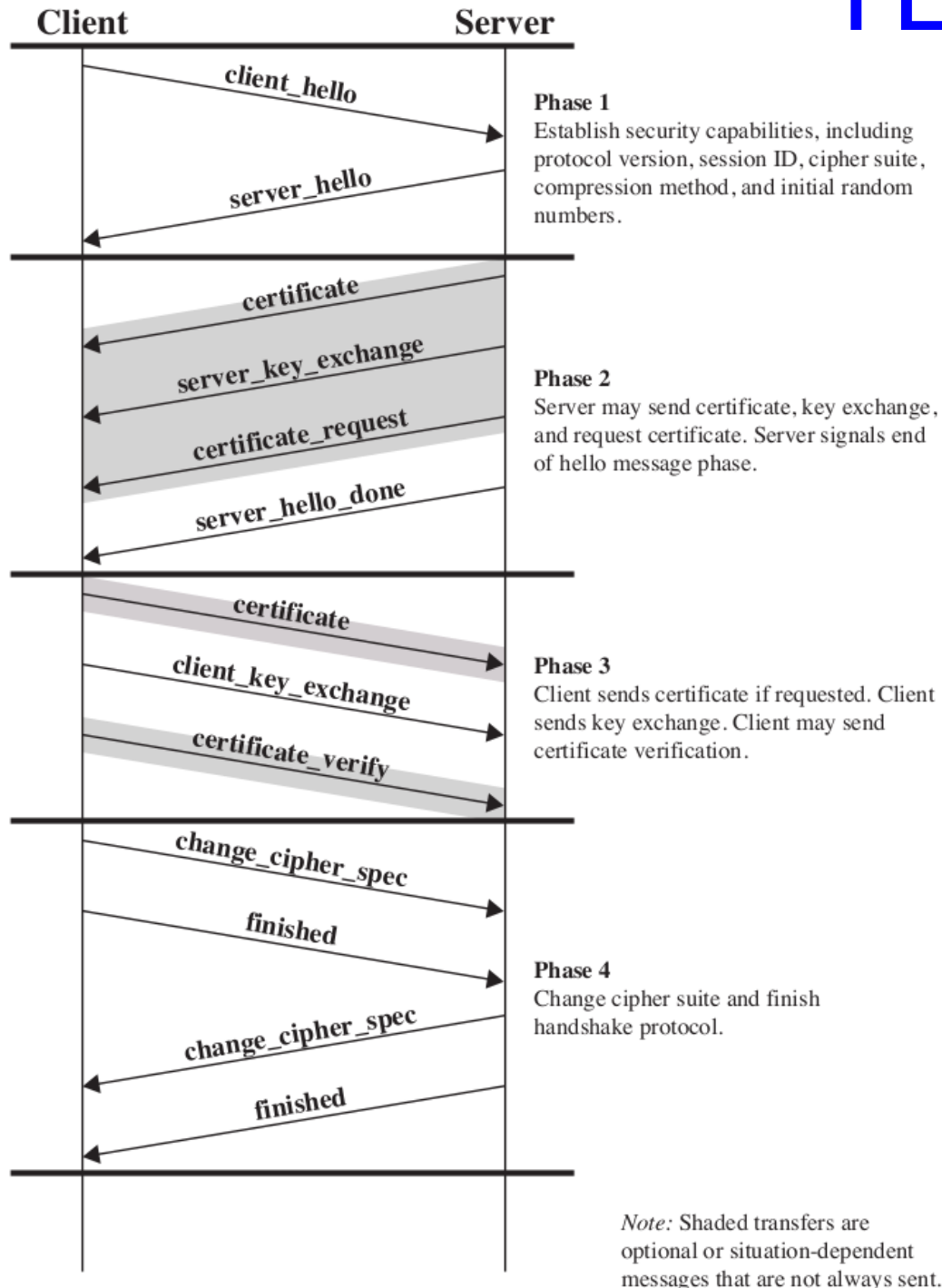
TLS is “transparent”

- the user sees only the plaintext
- not the ciphertext, keys, ..

TLS is “general”

- can be used by other applications, eg. HTTPS
- it is a big advantage for an application to not have to “reinvent” all the details of encryption etc.

TLS handshake



The main purpose of phase 1 is to agree on

(1) a symmetric encryption algorithm

- for encrypting data

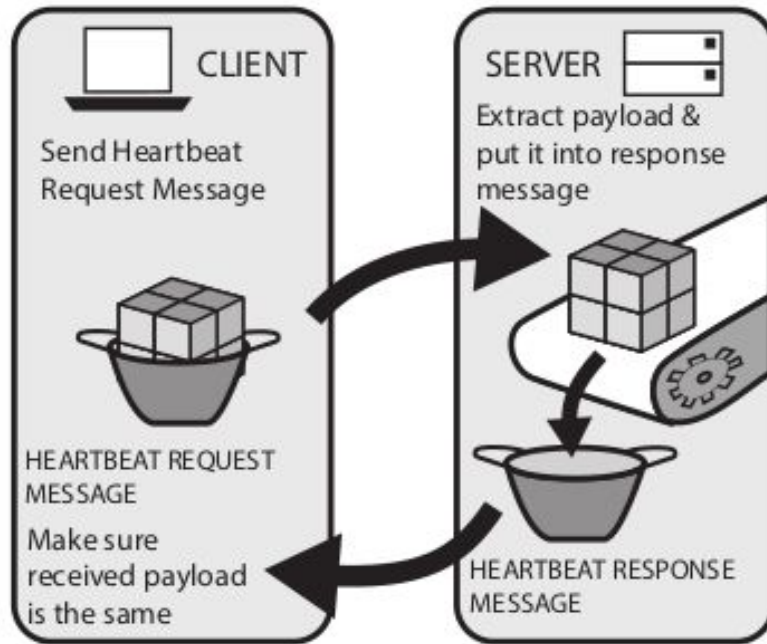
(2) an asymmetric encryption algorithm

- for encrypting the symmetric key

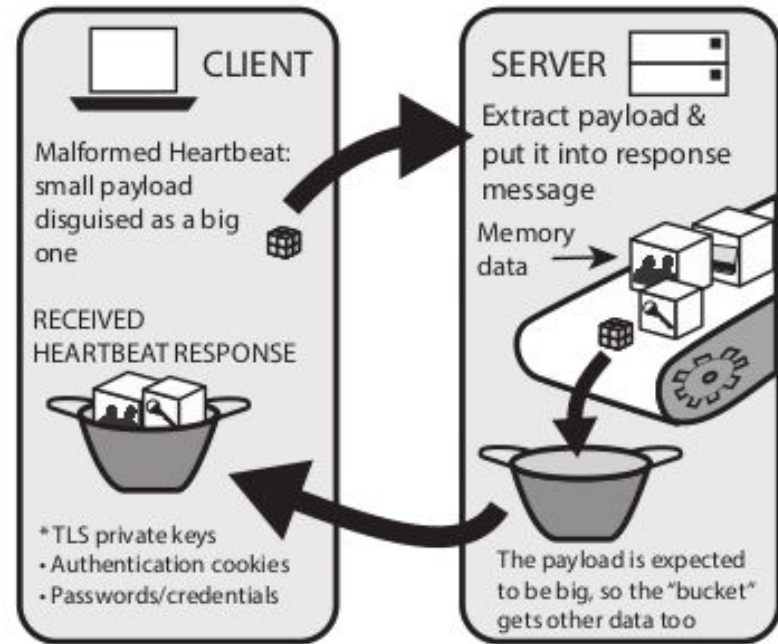
By contrast, in S/MIME and PGP

- no agreement phase
- sender just uses one of a set of recognized algorithms
- and hopes receiver can decrypt

TLS attacks: the heartbleed attack



(a) How TLS Heartbeat Protocol works



(b) How TLS Heartbleed exploit works

Heartbeat protocol

- lets client know that server is "alive"
- prevents connection from being closed
- payload lets client know which request was answered

Heartbleed exploit

- sends too much data back
- if actual payload is shorter the claimed

Remedy:

- make sure only actual payload is returned
- may require new version of OpenSSL

TLS attacks: the POODLE attack

POODLE = “Padding Oracle On Downgraded Legacy Encryption”

- POODLE tricks the two sides of an TLS connection into “downgrading” to older versions of the protocol

In phase 1 of the TLS handshake, the two sides

- agree on encryption methods
- and determine the highest TLS version supported by both

Suppose the server supports

- SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2

and the client supports

- SSL 3.0, TLS 1.0, TLS 1.1

Then they will agree on TLS 1.1 (the most recent version supported by both)

A “man-in-the-middle” may trick both sides

- into believing both sides support only SSL 3.0
- which they will then select
- the attacker may then exploit weaknesses in SSL 3.0

Remedy:

- you should delete all old versions of SSL/TLS (including SSL 3.0)

Some findings of Oh et al.

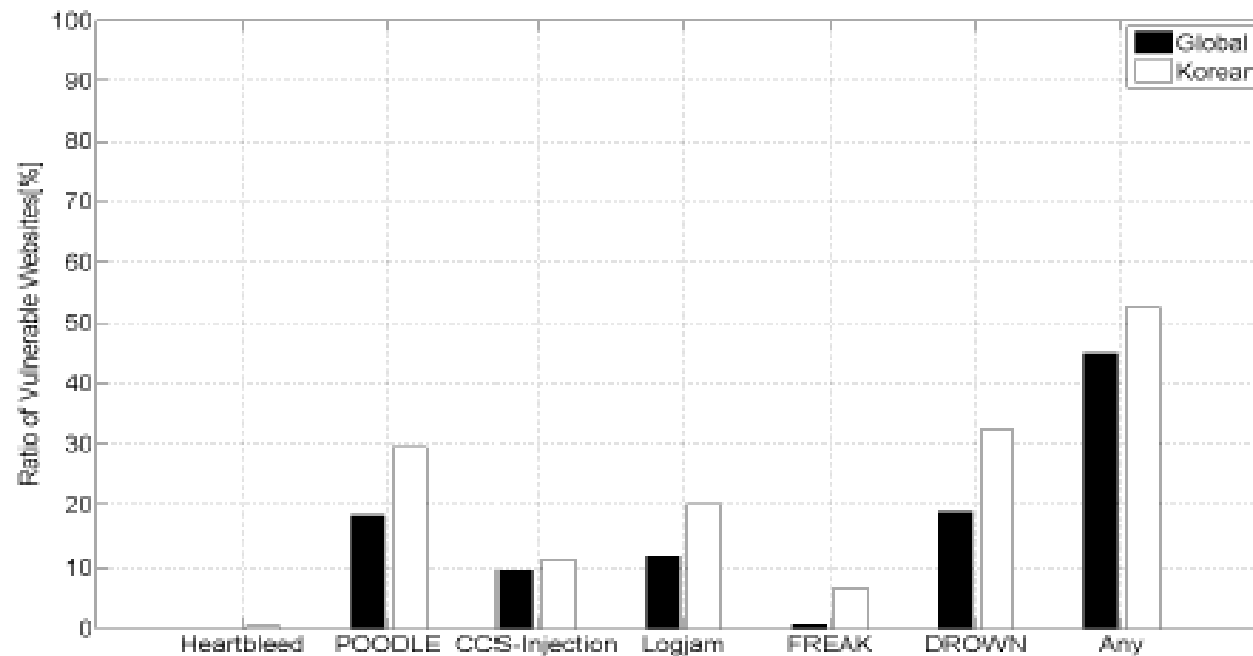


Fig. 1: Ratio of vulnerable websites for each of SSL/TLS attacks.

Table 6: Changes in number of vulnerable global websites tested over four weeks.

	Heartbleed	POODLE	Logjam	CCS injection	FREAK	DROWN
Week 1	0	91	60	48	3	100
Week 2	0	90	61	41	3	80
Week 3	0	91	61	44	4	80
Week 4	0	85	53	42	4	82

Exercise: nmap

nmap is a portscanner

- developed and released from 1997 onwards

nmap scans

- a host
- that is, a computer connected to the internet, or a local network

nmap works by

- sending requests to services on the host
- and studying the responses
- eg., SYN request (to TCP)

nmap - a hacker tool

nmap may be used by intruders

- identifying weaknesses on a host
- and thus help the intruder exploit the weaknesses

nmap may also be used in ethical and lawful ways

- by a normal user, detecting a printer
- by a system administrator, identifying and removing weaknesses

Using nmap may be illegal

- depending on the law in a given country
- certainly if you use the information to exploit the weaknesses
- similar to “walking up to a house and testing if the door is locked”

Using nmap may be unethical even without criminal intent

- by forcing a host to use computing power and network capacity to respond to thousands of requests

Download and install nmap

Download from nmap.org

- we do not need the graphical interface

Mac and Linux

- may have it already, otherwise download
- check in a terminal

```
$nmap --version
```

```
Nmap version 7.70 ( https://nmap.org )
```

Windows:

- when download + installation complete,
- start DOS-prompt and type

```
>nmap --version
```

```
(same response)
```

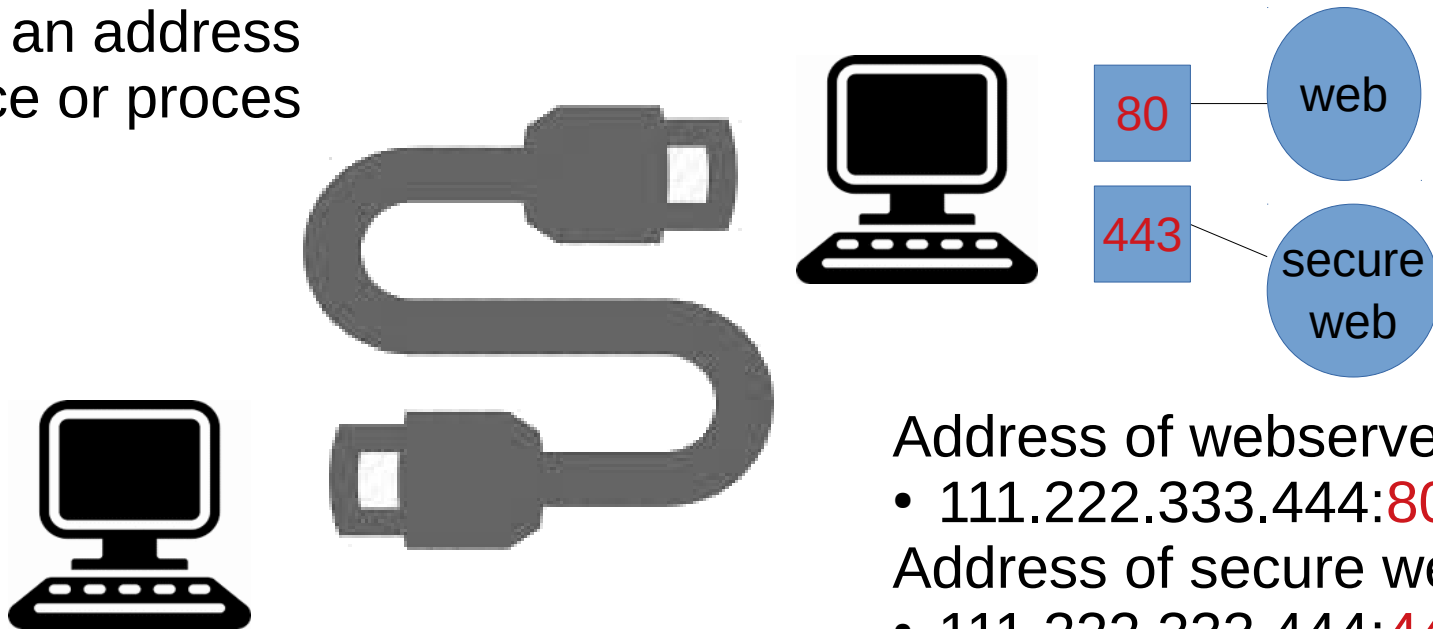
nmap is a port scanner

nmap

- scans one port at a time, on a host
 - commands may specify that nmap scans a specific port, on a host
- ```
> nmap -p 443 <niels' local IP address> (scan port 443)
> nmap -p 80 <111.222.333.444> (scan port 80)
```

A port is a specific address  
on a host computer

- the port is an address  
of a service or proces



Address of webserver:

- 111.222.333.444:80

Address of secure webserver:

- 111.222.333.444:443

# Ports and IP-addresses

Ports are a supplement to IP-addresses.

IP-addresses are the numeric addresses on the internet  
(as opposed to the symbolic addresses)

- The symbolic address `ruc.dk`  
corresponds to the numeric (IP) address `130.225.221.44`

Any computer connected to an internet-based network has 65535 ports (64K).

- ports 0-1023 are intended for “system” use
- ports 1024 and above are for applications
- ports are opened and closed by the programs that use them

Any message sent to a host, using eg. TCP, must specify:

- an IP number
  - plus a port number
- 
- See a list of port numbers  
[wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](http://wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers)

# Exercise

What are ports 20 and 43 typically used for?

# Exercise solution

What are ports 20 and 43 typically used for?

20: ftp

43: the whois service

whois is a service reporting about a domain name:

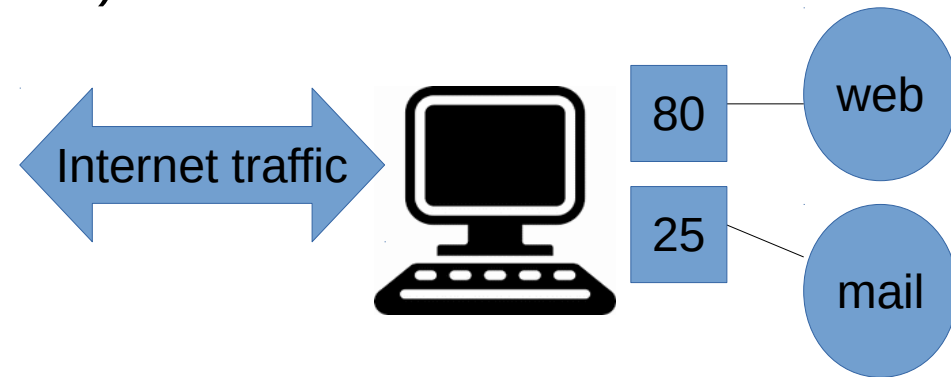
>whois ruc.dk

The reply says that ruc.dk is registered by Roskilde University  
(plus many details)

# What are ports used for?

Ports are a mechanism for filtering messages to a host

- ports supports that a message to a host
- are delivered to a specific service on the host
- eg., delivered to web (port 80), mail (port 25)
- ..



On a given computer,  
most of the 64K ports will be closed,  
but many may be open, say 10, 20 or 100.

Computers may be preconfigured with open ports

- eg. Code Red and the MS IIS on Windows 2000

To open/close a port from 0 to 1023

- programs should have administrator privileges
- for instance, a webserver normally runs with administrator privileges



# Exercises

Only do the suggested exercises  
plus possibly some scans that you know are not harmful  
(check with me)

1) What printing service is attached to port 9100 on print.ruc.dk  
> `nmap -p 9100 print.ruc.dk`

2) Find out what operating system my host/computer is using  
> `nmap -O 10.60.10.42`

3) Find out whether my host is vulnerable to the heartbleed attack  
> `nmap -p 443 --script ssl-heartbleed 10.60.10.42`  
(This uses script [svn.nmap.org/nmap/scripts/ssl-heartbleed.nse](http://svn.nmap.org/nmap/scripts/ssl-heartbleed.nse) )

4) Give your local IP address to a fellow student  
and ask her/him to check whether your host is vulnerable to heartbleed

# Monday 8<sup>st</sup> April: wireless security

Case: The wired equivalent privacy (WEP) protocol and the credit card theft at TJ Maxx

Stallings & Brown:

- Chapter 24. (Mandatory vs. not mandatory TBD)

Additional mandatory literature:

- *T.J. Maxx Data Theft Likely Due to Wireless 'Wardriving'*. (3-4 pages)
- *Choosing the Right Wireless LAN Security Protocol for the Home and Business User*. (Maple et al, 2006). (8 pages)

Student presentation:

- Presentation of the T.J Maxx data theft article.

