Network Security

Course notes

Version 2013.1

Contents

1 Preliminaries **1**
- 1.1 Motivation and definition . . . 1
- 1.2 How to understand security . . . 2
- 1.3 Critical characteristics of information . . . 3
- 1.4 Relating security concepts . . . 6
  - 1.4.1 Threats and attacks . . . 7
  - 1.4.2 Security controls . . . 7
- 1.5 Computer security strategy . . . 8
  - 1.5.1 Course structure . . . 8

2 Network Security **11**
- 2.1 Introduction . . . 11

3 VPN **13**

4 IPsec & SSL/TLS **15**
- 4.1 Intro: Layered security . . . 15
- 4.2 IPsec . . . 16
  - 4.2.1 IPsec overview . . . 17
  - 4.2.2 Communication modes: transport mode and tunnel mode . . . 17
  - 4.2.3 Security modes: AH and ESP . . . 17
  - 4.2.4 Combining modes . . . 18
- 4.3 SSL/TLS . . . 19
  - 4.3.1 Known vulnerabilities . . . 19
  - 4.3.2 Setting up a TLS/SSL connection . . . 22
  - 4.3.3 SSL/TLS data exchange . . . 24

Chapter 1

Preliminaries

1.1 Motivation and definition

We live in the *information age* and the society depends upon computers and information processing.
Observation: IT evolves fast (HW/SW) but security does not keep pace.
Why?

Developments	*Consequences*
Role of computers increases	
- invoices → strategic decisions	Value of information increases
- work → social interaction	Attack surface increases
Increased connectedness (Internet)	Increased vulnerability by increased access
Processes become more and more automated	Irregular behaviour noticed later
- computers talking more and more to computers	
Democratization of information technology	
- computer center → work place → phones	Less control, less expertise

Security is never a goal by itself. Security exists within a context, and that leads to various trade offs and balances. These trade offs can be captured by *security dilemmas*.

Definition 1 (Dilemma) *a situation requiring a choice between equally undesirable alternatives.*

For example:

Security Dilemma 1 *Security unaware users have specific security requirements but usually no security expertise.*

Example 1.1.1 (Security Dilemma 1.) *Users want security, but lack expertise.*

1. *See Figure 1.1.*
2. *Another example is the "Dancing Pigs" problem. In the words of Bruce Schneier (*Secrets and Lies*, pg. 262):*

> *If J. Random Websurfer clicks on a button that promises dancing pigs on his computer monitor, and instead gets a hortatory message describing the potential dangers of the applet – he's going to choose dancing pigs over computer security any day. If the computer prompts him with a warning screen like: "The applet DANCING PIGS could contain malicious code that might do permanent damage to your computer, steal your life's savings, and impair your ability to have children," he'll click OK without even reading it. Thirty seconds later he won't even remember that the warning screen even existed.*

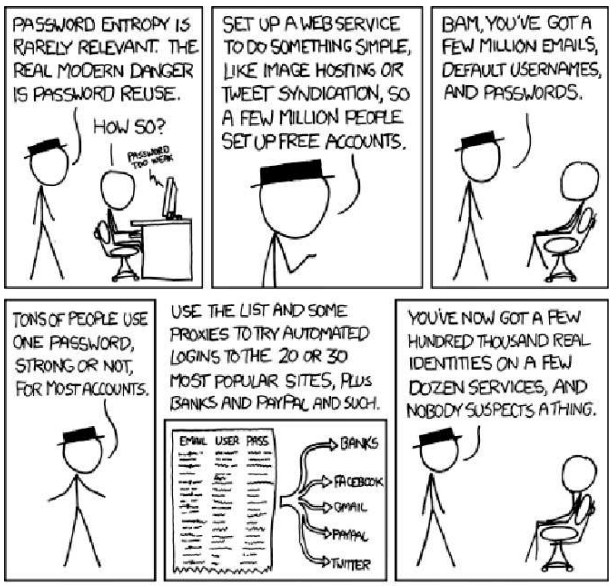


Figure 1.1: Security Dilemma 1: Users want security, but are not experts (by Randall Munroe, `http://xkcd.com/792/`)

The evolution of the context of security follows a similar path, from lack of understanding to expertise:

non-existent & not necessary	→	many levels & essential
(physical access control)		(physical, logical, organizational)

Many definitions of information security, but my preferred is: (by NSTISSC (CNSS = Committee on National Security Systems)) definition of information security

Definition 2 *Information security is the protection of information and the systems and hardware that use, store and transmit that information.*

Definition by NIST:

Definition 3 *Computer security: The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).*

1.2 How to understand security

There are three distinct categories to consider when trying to understand security:

1. What is "the system"?
2. What is the security requirement?
3. Who is trying to attack the system?

Note that the last point – who is the attacker – is often left unspecified (also in these lecture notes), and assumed clear from context. However, this is often a major course of security issues – underestimating the attacker.

Example 1.2.1 (Securing the lecture room.) *Consider the goal of securing access to a lecture room. The* system *consists of the physical properties of the room (doors, locks, windows, etc.) and the access controls it has (keys). The* security requirement *is that only teachers can allow access to the room.*

In this example, the attacker is left unspecified. From the context, one might assume that it concerns a student.

This constitutes a risk!

Examples of problems with this: A teacher may decide to leave exams in the room, secure in the knowledge that only other teachers can unlock the door and they will keep an eye on things, keeping students honest. However:

- An attacker can steal the key,
- An attacker could ask the cleaning staff to open the door,
- An attacker could come late at night with a ladder and break in via the window,
- An attacker could bribe the security guard,
- An attacker could trigger the fire department into breaking open the door,
- An attacker can break the wall (or floor, or ceiling),
- ...

We cannot say "a system is secure". We *can* say "this system satisfies this particular security requirement against an attacker who can do all of these actions."

Moreover, note that attackers think out of the box. To judge the security, we should learn to do so too!

1.3 Critical characteristics of information

Confidentiality, Integrity, Availability: C.I.A.

- Confidentiality
  Prevent disclosure to unauthorized persons or systems.
  E.g. nobody is allowed to read my e-mail. Possible security breaches:
  - inadvertently print e-mail;
  - administrator;
  - unintentional forward/group reply;
  - hacking.

  Two subclasses:
  - Data confidentiality: assures that confidential information is not made available or disclosed to unauthorized individuals.
  - Privacy: assures that individuals control what information related to them may be collected and to whom that information may be disclosed.
- Integrity
  Information is complete and uncorrupted (in store, in transmission).
  E.g. tampering with file containing students' marks. Possible security breaches:

  - system crash;
  - type `\rm -rf` in wrong directory;
  - virus deleting a file;
  - break into my account and change one of the grades.

  Two subclasses:

  - Data integrity: assures that information and programs are changed only in a specified and authorized manner;
  - System integrity: assures that a system performs its intended function in an unimpaired manner.

- Availability
  assures that systems work promptly and service is not denied to authorized users.
  E.g. I want to be able to read my e-mail. Possible security breaches:

  - Spam flooding;
  - Server crash.

  Other example: a website is accessible with Firefox v8 or above, Opera v8 or above, Safari v12 or above, Lynx v1.5 or above.

  An attack on availability is often called a *denial-of-service* (DoS) attack. If the attacker uses a number of different computers for attacking the system, it is called *distributed denial-of-service* (DDoS) attack.

The following requirements have been suggested in addition to the traditional CIA requirements:

- Authenticity
  The property of being genuine and being able to be verified and trusted.
  E.g. If I send you an e-mail concerning your grades, you want to be sure that I sent the e-mail. Possible security breaches:

  - Sender spoofing;
  - Somebody accessed my computer while I was away.

  Often considered as part of the integrity requirement.

- Accountability
  Actions affecting security can be traced to the responsible party.
  E.g. Someone broke into my computer, which provisions are in place?

  - Inspect system/network log files.

  Subclasses

  - Non-repudiation
    A party cannot falsely deny having send or received a message.
    E.g. Deny having ordered a book over the Internet.

- Accuracy
  Suitability of information for the envisioned purpose.
  E.g. Timestamping: how detailed is it?

  - Friday 13 April, 2007 vs.

  - Friday 13 April, 2007, 20:08 vs.
  - Friday 13 April, 2007, 20:08:00 CEST.

Note that confidentiality refers to *information*, not to *data*.

Definition 4 (Data vs Information) *Distinction between data and information:*

Data *is a physical phenomenon chosen by convention to represent certain aspects of our conceptual and real world.*

Information *is the meaning we assign to data.*

Example 1.3.1 (Data vs. information: morse code) Data: · · · — — — · · ·

Information: *distress call ("Save Our Souls")*
Note: There are multiple levels of encoding (data) here: the beeps and blips, but also "SOS", which is code for a distress call.

It is possible to *infer* information, without knowing how to decode the data, e.g.

- Traffic analysis (using data *about* data: meta-data):
  In WWII Pacific theatre: until 1942 traffic analysis was the only source of intelligence upon strategic positioning of the Japanese Navy.
- Amount of Wikipedia edits predicted the running mates of the two presidential candidates in 2008:
  As an example of a traffic analysis attack, consider the fact that before Sarah Palin became a running mate, this could have been concluded from observing the Wikipedia. There was a significant increase in the number of updates on her Wikipedia page right before the announcement. Furthermore, those who were making updates were also updated McCain's page.
- Database inference
  See chapter on Database Security.

Security Dilemma 2 *Security vs. ease of use.*

Ease of use is important for acceptance. Examples:

- requiring passwords to consist of at least 10 random numbers (more security)
  → user must write it down to remember (easier to use).
- Sever closing FTP connections idle which are for over 2 minutes
  → user writes a script to poke the communication every minute.

Security Dilemma 3 *Cost of security vs. gain.*

Security carries a cost:

- extra computational resources;
- operational overhead for end-user;
- extra cost for development, maintenance, education, management, etc.

On the other hand: security offers no direct gain, only prevention of loss. Sometimes, achieving a security requirement just isn't worth the cost of doing so.

Example: The Chinese 1 Fen banknote (all but disappeared). 1 Fen $\approx$ 0.1 eurocent. The banknote (Fig. 1.2) has few security provisions. The cost of increasing security provisions quickly outweighs possible gains.

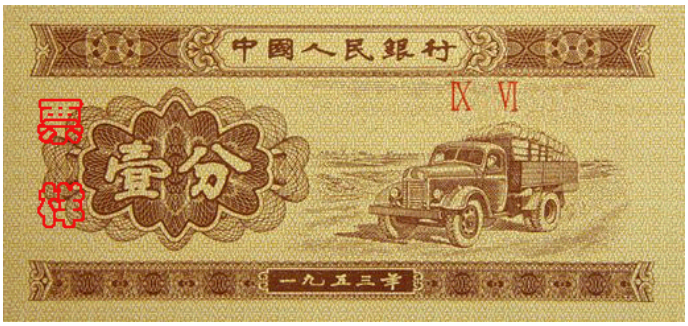


Figure 1.2: Chinese banknote of 1 fen (actual size, from WikiMedia)

Security Dilemma 4 *Cost of prevention vs. cost of failure and/or recovery.*

A failure due to a security incident may cost less than the cheapest security measure.
Example: buying a $10,000 safe to safeguard 2 documents worth $1,000 each.

Security Dilemma 5 *Theoretical vs. practical security.*

Example:

- (theory) If conditions are met abuse is (provably) impossible.
  E.g. encryption with shared key.

- (practice) increase friction for abuse above a certain threshold.
  E.g. Credit card payments often require the 3 or 4 extra digits printed in ink on the back, so this is not visible on slip, which is contact copied.

Security Dilemma 6 *Security vs. complexity*

Adding more security $\Longrightarrow$ increased complexity $\Longrightarrow$ more security holes/risks.

Example 1.3.2 (Security vs. complexity) *Airplane security. Very complex & strict, yet easy to breach (e.g., passengers accidentally taking weapons through security without being caught).*

1.4 Relating security concepts

Information security considers the protection of *assets*. An asset is anything that has a value for the organization or owner and that is central in the achievement of its (business) goals. Assets include:

- Hardware (computer systems, storage media, communication devices);

- Software (programs, operating systems);

- Data (files, databases, password files, network structure, configuration files);

- Network infrastructure (links, bridges, routers).

(these categories do not include assets such as buildings, cars, etc.).

1.4.1 Threats and attacks

Classification:

	Insider	Outsider	
Passive			
Active			

- Passive attacks are hard to detect, so aim to prevent.
  (release of message contents, traffic analysis)
- Active attacks are hard to prevent, so aim to detect.
  (masquerade, replay, modification, denial of service)

Example 1.4.1 (Classifying WWII traffic analysis.) *The traffic analysis mentioned above was done by an* outsider *(the US navy) and was a* passive *attack (eavesdropping).*

Example 1.4.2 (Classifying attacks from movies.)

- *The gang in "Ocean's 13": outsider, active.*
- *The character Phoenix (Wesley Snipes) in "Demolition Man": insider, active.*
- *The Mole in "Austin Powers: Goldmember": insider, passive.*

1.4.2 Security controls

A measure to "aid" security.
Aid:

- Prevent a security breach
- Detect a security breach
- Correct/Respond a security breach

Note:

- Goal is to minimize risk given constraints
- Security controls may result in new vulnerabilities (cf. Security Dilemma 6).
- Will have residual vulnerability.

Classification:

	Physical	Logical/Technical	Administrative/Policies
Preventive			
Detective			
Corrective/Responsive			

Example 1.4.3 (fire) *Security controls wrt. fire in an office building:*

	Physical	*Logical/Technical*	*Administrative*
Preventive			*"no smoking" policy*
Detective		*smoke detectors*	
Corrective/Responsive	*sprinklers*		

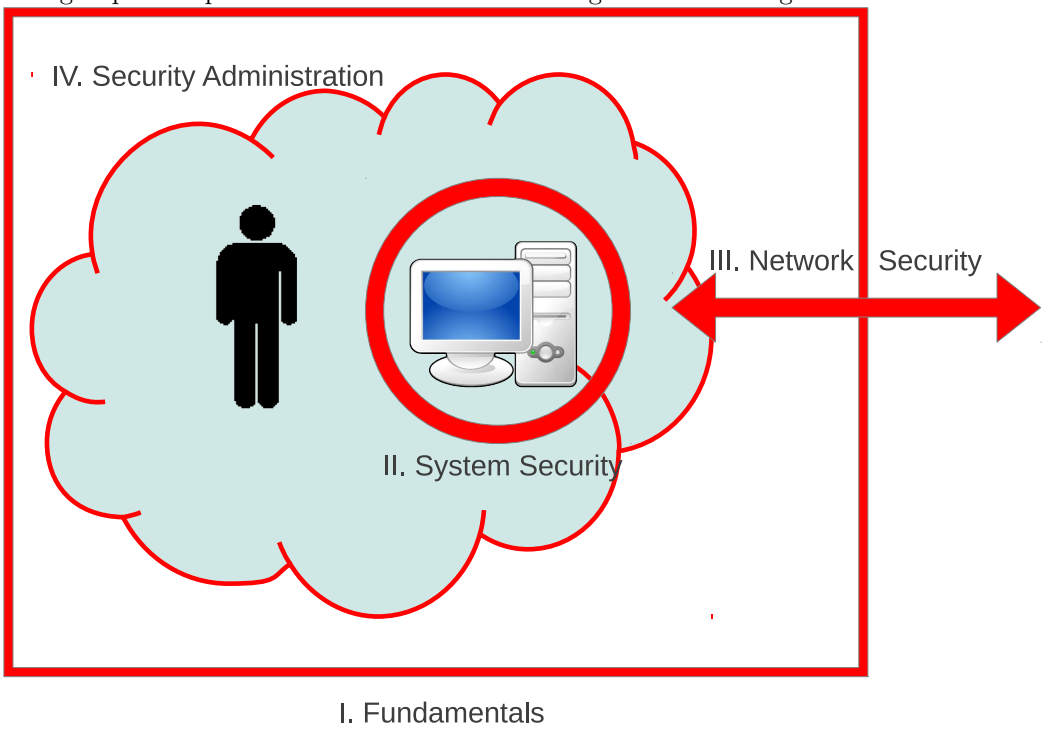
1.5 Computer security strategy

(cf. Chapter **??**.)
A strategy for providing computer security involves:

- Specification/policy
  - What is the security scheme supposed to protect?
  - From whom?
  - Codify in policy and procedures.
- Implementation/mechanisms
  - How does it work?
  - Prevention, detection, response, recovery.
- Correctness/assurance
  - Does it really work?
  - Assurance, evaluation.

1.5.1 Course structure

We will group the topics of the next lectures according to the following domains.



I. Fundamentals
Provide basic knowledge needed for information security.
Terminology of basic concepts, cryptography, ...

II. System security
Program security, viruses, operating system security, access control, database security, ...

III. Network security
Internet security, firewalls, intrusion detection, communication protocols, ...

IV. Administrative security
Security analysis, policies, planning, auditing, risk assessment, ...

Chapter 2

Network Security

2.1 Introduction

Network characteristics imply vulnerabilities.

Characteristic		*Vulnerability*
• Anonymity of attacker	$\Longrightarrow$	no penalty in attempting to attack.
• Many points of attack	$\Longrightarrow$	principle of weakest link
• Sharing of resources between users	$\Longrightarrow$	complicated access control.
• Complexity of system (heterogeneous, large)	$\Longrightarrow$	hard to control.
• Unknown perimeter (a node can belong to several networks, new nodes can be easily added)	$\Longrightarrow$	hard to control.
• Unknown path of packets	$\Longrightarrow$	hard to control.
• New challenges: wireless/roaming/mobile users	$\Longrightarrow$	complex protocols, new ways of attack. e.g. theft of service.

Some security controls:

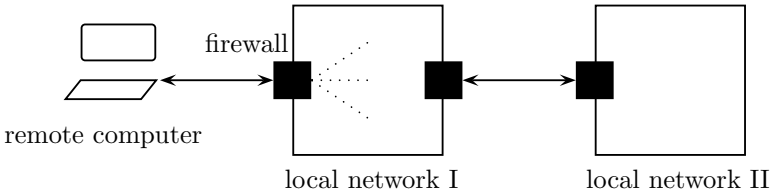
- Proper design and implementation.
- Architectural security controls:
  - segmentation (separated services e.g. company applications/web server/database connected via firewalls)
  - redundancy (no single point of failure) (duplicate web servers, distributed database)
  - security protocols/encryption
    - link encryption: encrypt data in between two intermediate nodes; message in plain text in each intermediate node (at physical or data link layer).
    - end-to-end encryption: encrypt data at one end of the communication, decrypt at the other end (at application or presentation layer).

Chapter 3

VPN

A virtual private network (VPN) is a computer network that is implemented in an additional logical layer (overlay) on top of an existing network. It has the purpose of creating a private scope of computer communications or providing a secure extension of a private network into an insecure network such as the Internet.

Briefly, secure remote access to a local network, as if you are part of this local network. Can be used either to connect a client to the network, or to connect two separate networks together (e.g. when a company has sites in different countries).



Phases:

1. Authentication (of client and server).
2. Key establishment to create an encrypted "tunnel".
   Tunnel = a new channel over an existing channel (e.g. SMTP over an HTTP channel)
   Tunneling operates at the transport layer.
3. Secure communication between client and firewall through tunnel; firewall forwards the communication to the nodes in the local network.

Often, a distinction is made between

- **Trusted VPN**: provides a limited view of the network. Often used for traffic segmentation on large core networks.
- **Secure VPN**: limited view + protection for traffic remote $\leftrightarrow$ firewall (authentication, confidentiality, ...).

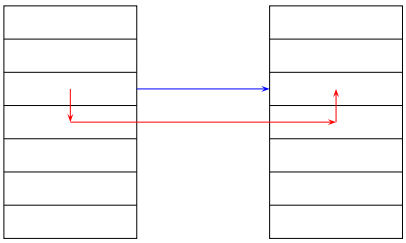
VPNs can occur at different layers of the OSI/OSI stack: layer 2 (but not secure), layer 3, and layer 4.

Chapter 4

IPsec & SSL/TLS

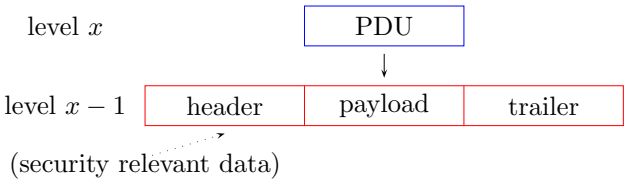
4.1 Intro: Layered security

Recall the ISO/OSI layered network model



7.	application	software (e.g. HTTP)
6.	presentation	encryption (e.g. SSL)
5.	session	channel management (e.g. NetBIOS)
4.	transport	reliable data transfer (e.g. TCP)
3.	network	point-to-point connections (e.g. IP)
2.	data link	connect within network (e.g. PPP)
1.	physical	signal en-/decoding (e.g. Bluetooth)

A Protocol Data Unit (PDU) at layer x is considered payload of one or more packets at layer $x-1$.



Note: A connection at level x may seem a direct controlled link between entities, but this may not be the case at level $x-1$ (e.g. due to bridges/gateways/routers).

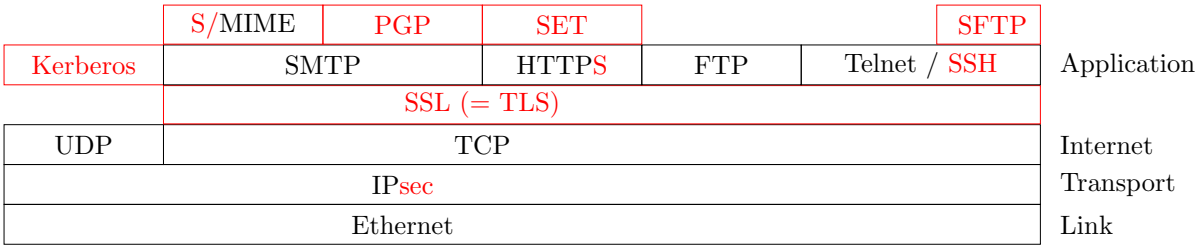
Security Dilemma 7 *At which level (layer) to take security measures?*

- High level: specific, tailored to application, different applications need their own security protocol, cryptographic keys shared between users, users are aware of security.
- Low level: general, may not meet the applications' requirements, cryptographic keys shared between nodes, transparent for user.

Thus, there are security concerns at every level.

Note: If a level is compromised, then every higher level without additional security measures is compromised too.

Examples of security provisions in the Internet stack (which does *not* correspond directly to the ISO/OSI layered model):



- S/MIME = Secure / Multipurpose Internet Mail Extensions
- PGP = Pretty Good Privacy
- SET = Secure Electronic Transactions
- Kerberos = authentication protocol
- SMTP = Simple Mail Transfer Protocol
- HTTPS = HyperText Transfer Protocol over Secure Sockets Layer
- SFTP = SSH File Transfer Protocol (SSH = Secure Shell)
- SSL = Secure Sockets Layer
- TLS = Transport Layer Security
- UDP = User Datagram Protocol
- TCP = Transmission Control Protocol
- IPsec = Internet Protocol Security

4.2 IPsec

Example threats:

- Sniffing (read sensitive data)
- Spoofing (forged source address, forged contents), although IPv4 has a CRC checksum (Cyclic Redundancy Check, to verify integrity of received message), this does not protect against intentional modifications (a cryptographic hash is needed for that).
- Deny transaction (will be ignored)
- Traffic analysis (who is sending a message to whom?)

IP is connectionless and stateless (each datagram is an independent entity), there is no guarantee of delivery.

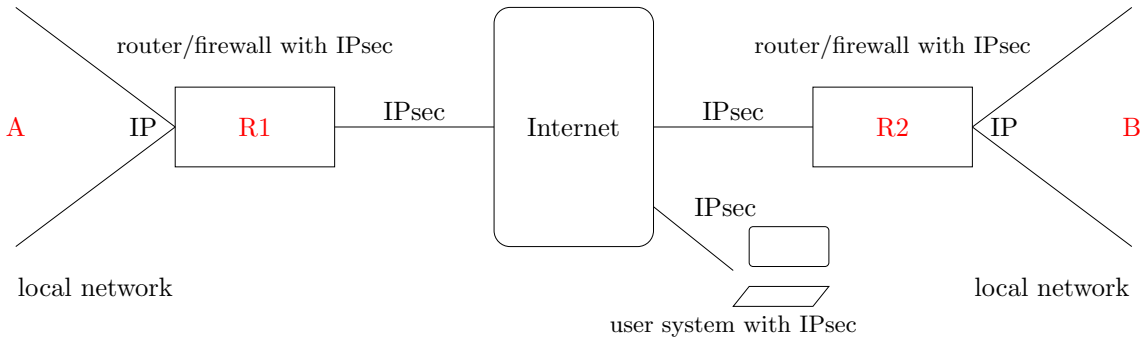
- IPv4 (1981) no security provisions needed/included.
- IPv6 (1996) to solve e.g. limited address space, and IP address spoofing/packet sniffing.
- IPsec is an optional extension for IPv4. It has been a mandatory part of IPv6 until December 2011. Then it has been downgraded to optional by RFC 6434.

IP header:

- Source IP address: gives no protection against address spoofing.
- Checksum (CRC, not cryptographic): no protection against *intentional* modifications.

4.2.1 IPsec overview

Typical configuration:



4.2.2 Communication modes: transport mode and tunnel mode

IPsec has two "communication modes": *transport mode* and *tunnel mode*.

Let's denote a message P from A to B by (A,B,M) and let's denote a "secured" message (encrypted and/or authenticated, depending on the security mode) by [M].

If IPsec in the previous picture operates in transport mode, then only the payload is secured. In tunnel mode, the whole message (A,B,M) is secured and is used as the payload for a newly constructed message between the routers. Schematically:

- Transport mode: $(A,B,M) \xrightarrow{package} (A,B,[M]) \xrightarrow{depackage} (A,B,M)$
  - for end-to-end communication (e.g. two workstations)
  - only payload is encrypted/authenticated, not the headers
- Tunnel mode: $(A,B,P) \xrightarrow{package} (R1,R2,[(A,B,P)]) \xrightarrow{depackage} (A,B,P)$
  - used when both ends are gateways
  - protect entire IP packet

4.2.3 Security modes: AH and ESP

What "secured" message means, depends on the security mode of IPsec. There are two security modes: *Authentication Header* (AH) and *Encapsulating Security Payload* (ESP).

Authentication Header (AH)

AH provides:

- Data integrity
  (modifications to data can be detected).
- Authentication of IP packets
  (at IP address level only; user/application authentication has to be provided by the end system/network device).
- Anti-replay service
  (replaying the same IP packet can be detected).

Data integrity and authentication are established by calculating a MAC (Message Authentication Code, e.g. MD5/SHA-1 with shared key) over some of the fields of the packet. This requires that a cryptographic key is shared between sender and recipient. The MAC is added to the header of the packet.

Replay prevention. Packet replay is prevented by extending the header of an IP packet with a *sequence number*. There are two considerations for this sequence number.

1. This number consists of 32 bits, so after $2^{32} - 1$ messages the sequence number is back to 0. This could still allow for a replay attack every 2^{32} messages.
   The solution is to "reset" the connection after 2^{32} messages by establishing a new *Security Association* between the sender and receiver.

2. Packets may get lost and may arrive in different order.
   The solution is that the receiver keeps a "window" for the last 64 packets, marking for every packet in the window whether it has been received. Given a window of accepted packet numbers from *low* to *high* (inclusive), and given a boolean array *Rcv* that keeps track of received packeds ($Rcv[i] = true$ meaning that the packet was received), the receiver acts as follows on arrival of a packet with sequence number i.

   - IF $i \in AcceptedPacketNumbers$ AND $Rcv[i] = false$, and the MAC is ok
     $\rightarrow$ set $Rcv[i] := true$.
   - IF $i > high$ (package is newer than window) AND MAC is ok
     $\rightarrow$ slide window to right by $high := i; low := low + (i - high)$, set $Rcv[i] := true$.
   - IF $i < low$ (package too old) OR MAC fails OR $Rcv[i] = true$ (packet already received)
     $\rightarrow$ discard packet, log fault event.

Encapsulating Security Payload (ESP)

Encapsulating Security Payload provides:

- Confidentiality of message contents
  (through encryption of the payload).
- (Limited) traffic flow confidentiality
  (through encryption of payload plus original source and destination, an outsider cannot determine source and destination).
- optional: same authentication as in AH.
- anti-replay service (as in AH).

Confidentiality is established by encrypting the payload.

4.2.4 Combining modes

The different modes can be combined as follows.

	Transport mode	**Tunnel mode**
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts inner IP packet.
ESP + authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts inner IP packet. Authenticates inner IP packet.

It is possible to define a "tunnel within a tunnel" (iterated tunneling).

4.3 SSL/TLS

- SSL = Secure Socket Layer;
  Developed by Netscape (1994), last version 3 (1996)
- TLS = Transport Layer Security;
  first version 1.0 = SSL 3.1 (1999); current version 1.2 = SSL 3.3.

TLS is an Internet Standard (IETF).
Purpose: placed between reliable connection-oriented network layer protocol and application protocol layer.
Provides secure communication between between the client and the server by supporting:

- mutual authentication;
- digital signatures for integrity;
- encryption for confidentiality.

Example: used for secure communication with web site (https = http over SSL).

Based on Public Key Infrastructure (=PKI) with certificates.

Certificates are signed by a Certification Authority (CA), e.g. VeriSign, DigiNotar. Web browsers are shipped with lists of CA's that will implicitly trust to identify hosts. In addition it is possible to load a new CA into your browser.

Problem: it is possible to sign your own certificates. In this case, the user will be warned that the certificate is not signed by a trusted CA. The average user will not understand this warning and simply click the warning window away and continue interaction with the web site.

Problem 2: root CAs are not always diligent about security (see incidents with Comodo, DigiNotar).

A certificate typically contains the following information:

- Subject: Name, Public Key;
- Issuer: name, Signature;
- Period of validity: Not before date, Not after date;
- Administrative information: Version number.

Certificate chains are allowed.
SSL operation consists of three main phases:

1. Negotiation between peers to determine supported algorithms;
2. Session key exchange (using public key encryption) and authentication (using certificates);
3. Data exchange (encrypted with symmetric session key).

4.3.1 Known vulnerabilities

Known issues with SSL 3.0 / TLS 1.0-1.1:

- client renegotiation attack (discovered in 2009)
  Roughly: server accepts unauthenticated requests during handshake, and can merge them into the current session.
- traffic analysis may still work (in specialized cases).
  Example: detecting what someone is watching on Google Maps:
  `http://blog.ioactive.com/2012/02/ssl-traffic-analysis-on-google-maps.html`

- BEAST: Browser Exploit Against Ssl 3.0 / Tls 1.0
  A JavaScript suite that automates (and makes practical) a theoretical weakness in one cryptomode of TLS. The weakness required loads and loads of ciphertexts to exploit; the JavaScript part ensures this happens.
  `http://www.educatedguesswork.org/2011/09/security_impact_of_the_rizzodu.html`
- CRIME: Compression Ration Information-leak Made Easy.
  Attacker injects guess of partial cookie. Correct guess $\implies$ detectably better compression.
  `http://en.wikipedia.org/wiki/CRIME`
- TIME: Timing Information-leak Made Easy.
  CRIME-like attack that (on a bad guess) shifts data outside the TCP sliding window $\implies$ big timing difference in responses between good guesses and bad guesses.
- BREACH: Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext.
  Similar to CRIME, except CRIME uses TLS-compression and BREACH works on HTTP-compression (`Accept-encoding: gzip, deflate`).
  `http://breachattack.com/`
- Lucky 13 (TLS 1.2): TLS 1.2 was designed to stop the *Padding Oracle Attack*, which relied on timing variations. The time difference between handling correct and false (injected) traffic was reduced.
  There still is a timing difference. Lucky 13 uses that to recover plaintext from a TLS CBC-mode connection.
  `http://en.wikipedia.org/wiki/Lucky_Thirteen_attack`
- HeartBleed (OpenSSL implementation): there is a "heartbeat" option in TLS, that keeps the connection alive (by sending small communications) when the connection is otherwise idle. Unfortunately, in the OpenSSL implementation of this, the client-side could set the length of the response... which was not matched to anything. This allows the client to extract data from the server memory (see Fig. 4.1).
  `http://heartbleed.com/`

Some examples of weaknesses of how SSL/TLS is used. These attacks and weaknesses are old (early 2000s), but are often not sufficiently addressed. (Mostly from Moxie Marlinspike's DefCon 17 talk, `https://www.youtube.com/watch?v=ibF36Yyeehw`)

- SSLSniff (SSL certificates): most SSL certificates allow you to sign a certificate (they shouldn't). Moreover, many certificate checks didn't (2011) check if the certificate is *allowed* to sign. So use your certificate for "mywebsite.com" to sign a certificate for "paypal.com".
- SSLStrip: most people don't type in "https://", but click a link somewhere on an HTTP page. So: MITM attack on HTTP connection, changing any HTTPS into HTTP link. Keep client on HTTP while talking to server on HTTPS.
- Revocation checking (certificates):
  - To check whether a certificate has not been revoked, a browser is supposed to contact the signing authority (CA) and check that they did not revoke the certificate (for each certificate in the certificate chain). However, if the connection to the signing CA fails, the default is to accept the certificate...
  - The protocol to check whether the certificate is revoked or not does not authenticate its entire response, in particular: the response status. So: change that to "tryLater" – accepted (in 2011) by many SSL implementations as "good enough".
- Certificate validation: You can get a certificate for any subdomain on your site – and also wildcard subdomains (`*.yourdomain.com`). The name you submit may contain a `\0` character, which in C code terminates a string. Most certificate implementations are written in C code, and incorrectly terminate at the `\0` character (e.g. `*\0.yourdomain.com`).

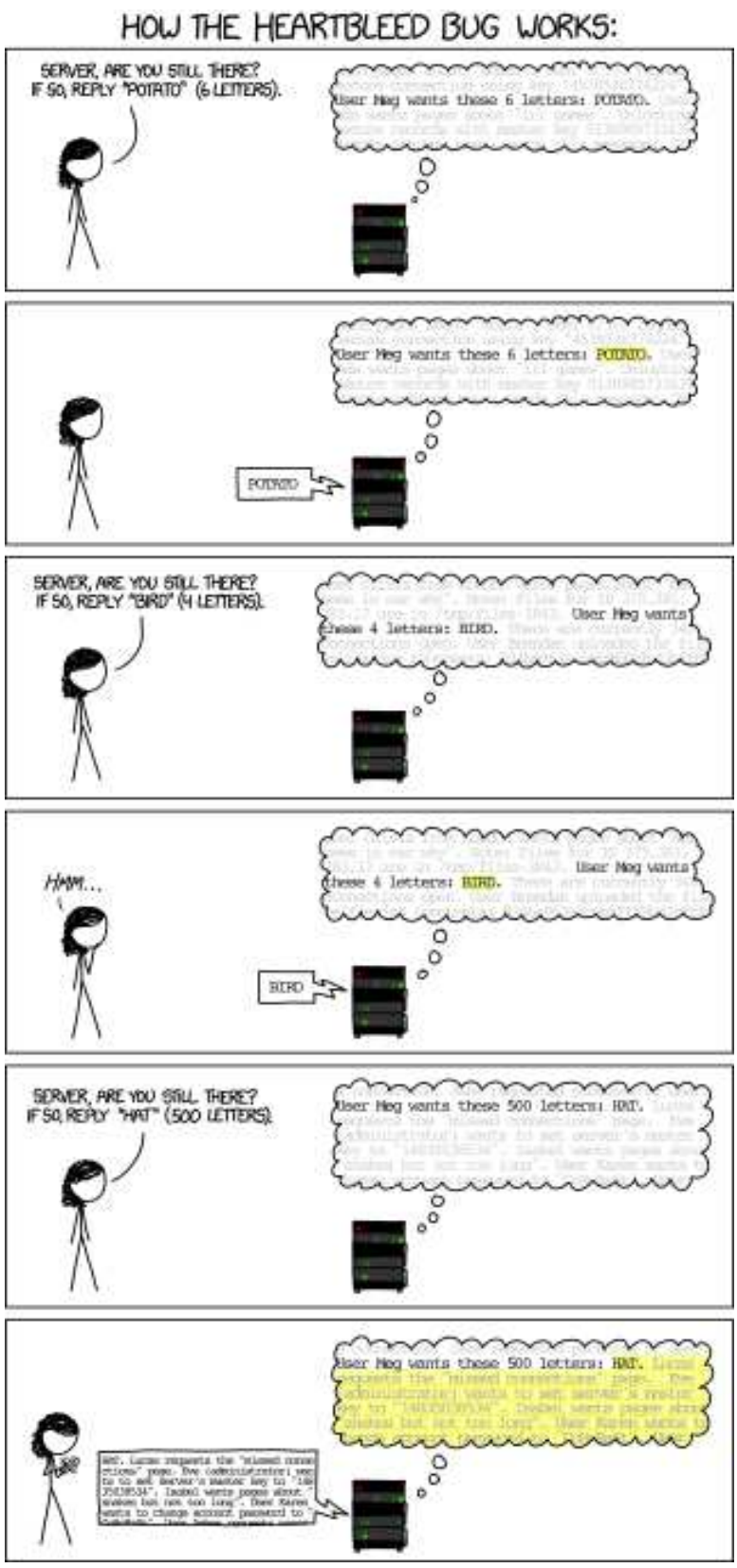


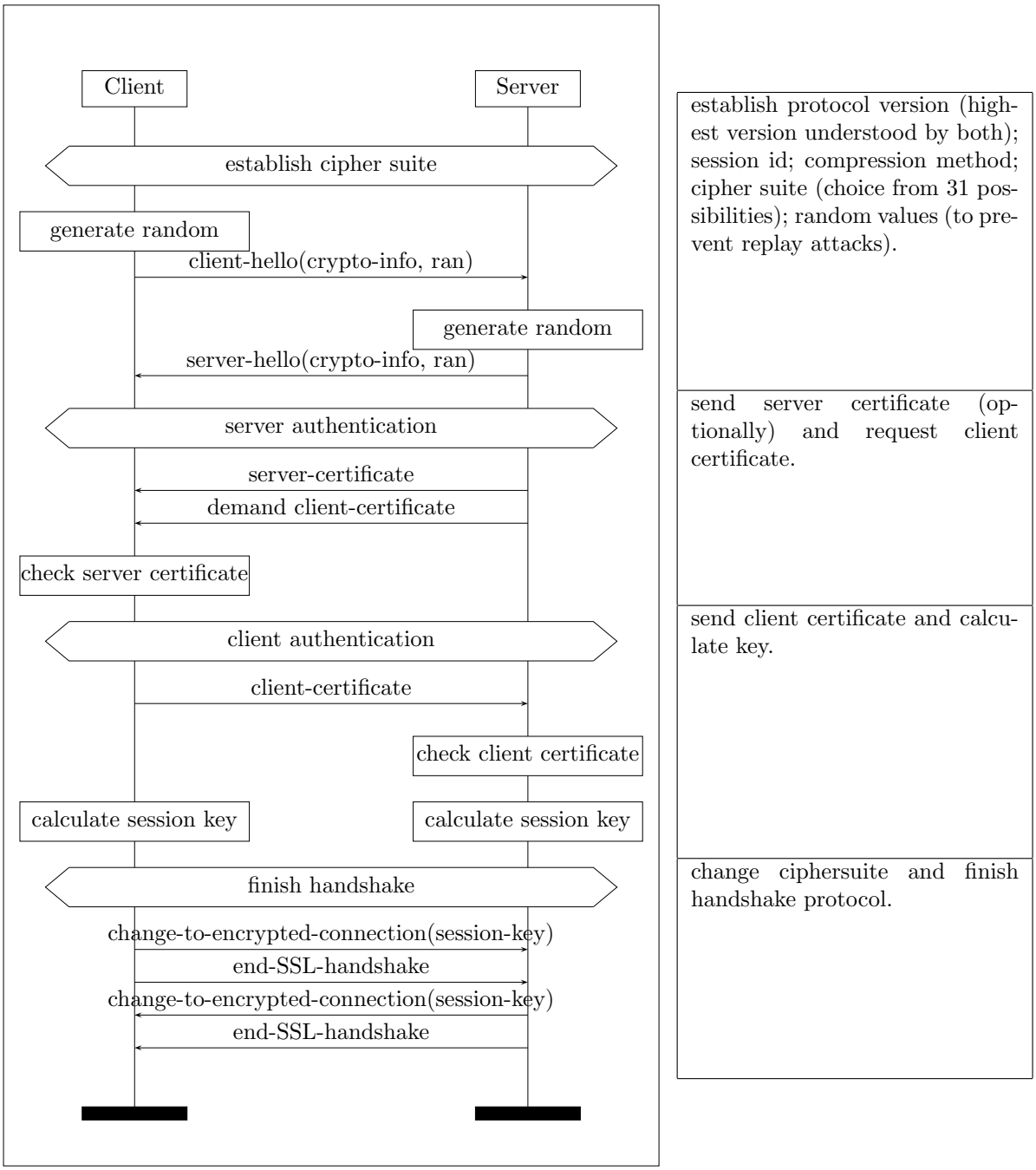
Figure 4.1: HeartBleed bug (XKCD 1354 © Randall Munroe)

4.3.2 Setting up a TLS/SSL connection

Session establishment through the *SSL handshake protocol*. This works in 4 phases:

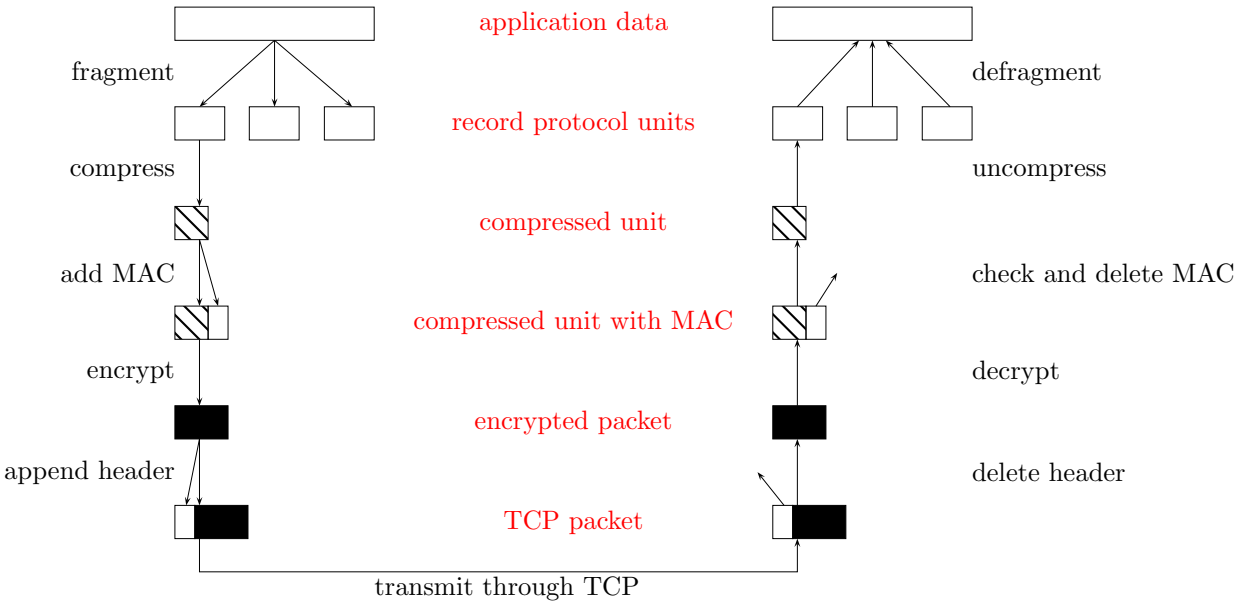
- paremeter agreement (protocol version, supported ciphers, etc),
- server authentication,
- (optionally) client authentication,
- switch to an encrypted connection.

Schematically:



4.3.3 SSL/TLS data exchange

Data exchange through the *SSL record protocol*:



Bibliography

[1] S. Even and Y. Yacobi. Relations among public key signature systems. Technical Report 175, Computer Science Department, Technion Haifa, Israel, Mar 1980.

[2] Adi Shamir. On the power of commutativity in cryptography. In *ICALP*, pages 582–595, 1980.

[3] Jianying Zhou, Robert H. Deng, and Feng Bao. Some remarks on a fair exchange protocol. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 46–57. Springer, 2000.

[4] Jianying Zhou and Dieter Gollmann. A fair non-repudiation protocol. In *IEEE Symposium on Security and Privacy*, pages 55–61. IEEE Computer Society, 1996.

[5] Jianying Zhou and Dieter Gollmann. Observations on non-repudiation. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT*, volume 1163 of *Lecture Notes in Computer Science*, pages 133–144. Springer, 1996.