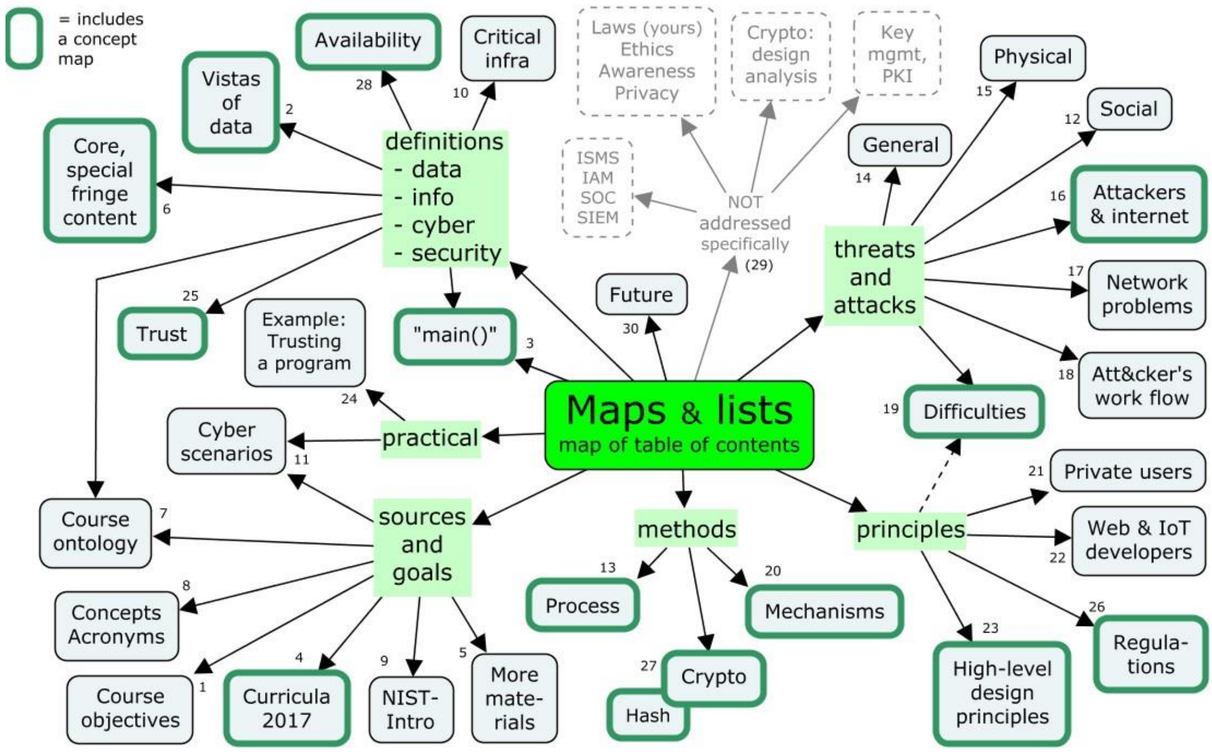
Maps and lists for Cyber Security 1

Contents



This text was written to support learning, but its intention is not to provide direct solutions to problems that appear in the materials of exams or exercises at http://tie-sec.rd.tuni.fi/maso/ and https://tie-sec.rd.tuni.fi/harpo/. The latter requires registration that is available only for students enrolled in this course. And the former shows that already Exam 1 includes more topics than the current document. On the other hand, many details in this document go beyond the course requirements.

1. Course objectives

The concise definition of objectives in the official syllabus is reproduced here in italics. More detailed explanations follow each line.

The aim is to learn basic skills about cyber security, needed by everyone who is studying information technology.
This objective must be understood in a wide sense. The studies themselves impose a need of some cyber security skills, but the main motivation comes from the professions where the students of information technology will end up. In addition, this objective covers skills needed in society and not only as citizens but also as technology-aware counselors of fellow-citizens who are not likely to be able to take sufficient care of their own security. Lastly, the student will also be prepared for further studies in the infosec curriculum.

The student
• *identifies security and privacy threats and responsibilities;*
This is an obvious but not simple starting point of anything you do in the field of information security, and privacy can be considered just a branch of it. You must understand what bad can happen even if everything seems to be fine. In addition, in your work and to some extent in your private life it is not only good but obligatory to act properly in the face of security threats.

• *has a wide knowledge of the concepts, principles and mechanisms of information security;*
This course covers a very large spectrum of them. After passing this course you will not encounter very many completely new things even if you continue your studies in information security. There will always appear new kinds of vulnerabilities and attacks, but this is one of the facts that you also know and most likely you will be able to categorize all new things in the spectrum. It is of course clear that you cannot obtain very deep knowledge over a very wide range of things during a short course. This is why there is the third objective.

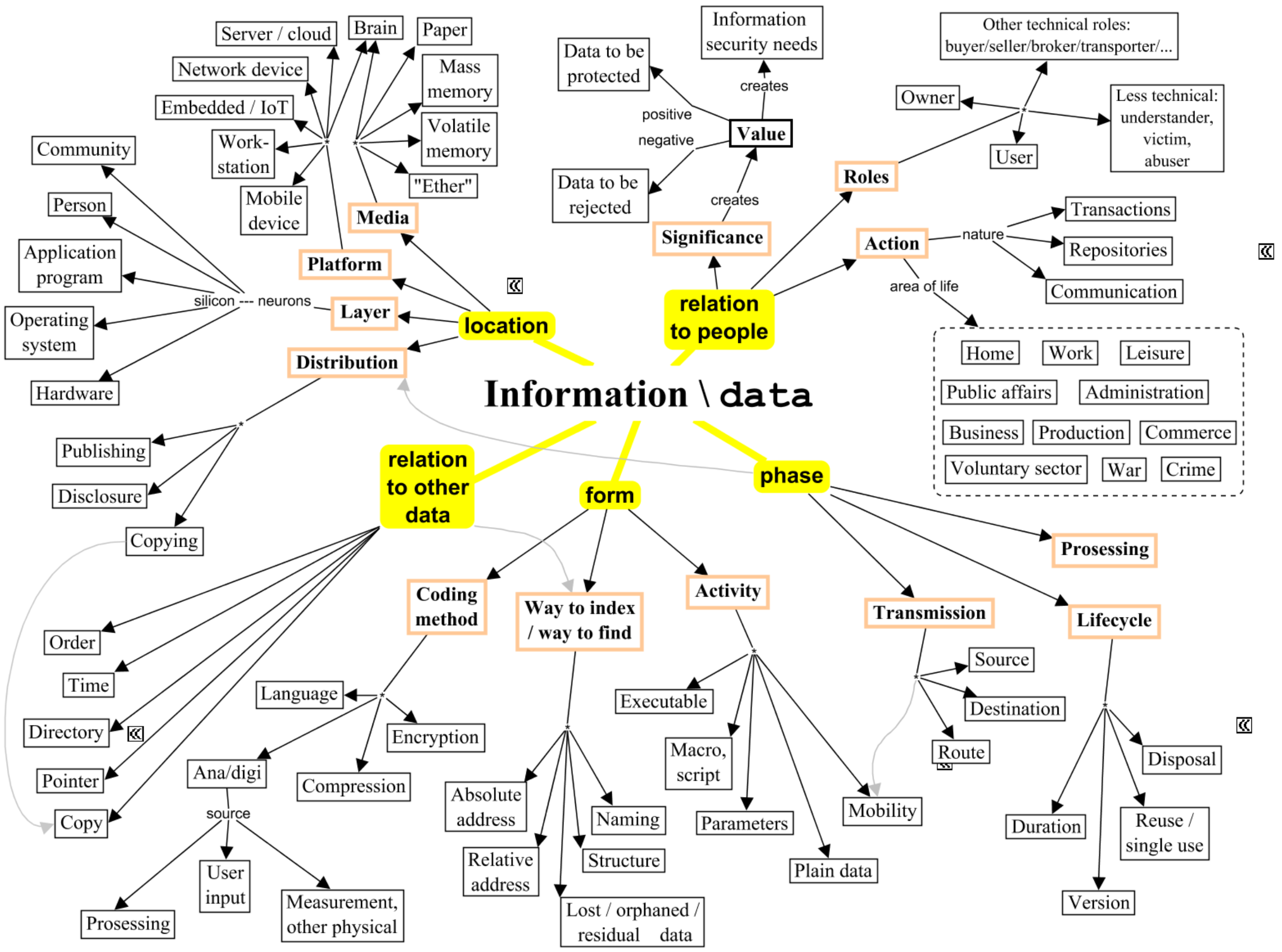
• *knows what kind of additional knowledge and skills are needed to perform various information security tasks in different application areas.*
This is a very important goal of a basic course, but it is difficult to measure. Your way of achieving this will mainly happen by getting familiar with small parts of large documents, web sites that you found and judged yourself, and the constant flow of news during the process of doing an exercise on them. Thereby you are likely to start appreciating these sources of knowledge and understanding the depth of skills that is required in information security professions.

2. From information to information security

The concept map below presents a diversity of aspects and attributes that data can have. You may find more by asking deep questions like how data becomes information and what makes it knowledge. More simply you may create a similar map by asking: what is/are data like, where does it come from, where is it, how is it changed, where does it go, why, who, ... ?. At least 'how is it changed' gives you a good opportunity because the node "Processing" is left without details.

From each item in the concept map – or from your own questions around data – you can find a security perspective by asking, what can *go wrong*, or what *should not* happen. Of course, the question, what *should* happen, also raises security issues. They are usually under the title *availability*, but many of them are often perceived as the overall functioning of information systems.

Security itself is shown only at the node "Value". Very abstractly, protections and rejections are security functions that should happen. It could be illustrative to link the node "Information security needs" to "Owner" by a phrase like "should lead to security policy settings by". This is true in the administrative sense, but it would hide the rest of the security needs. So the security needs are left for you as an exit point of this map to other maps – including the *what-can-go-wrong* map that you draw in your mind on top of this one.

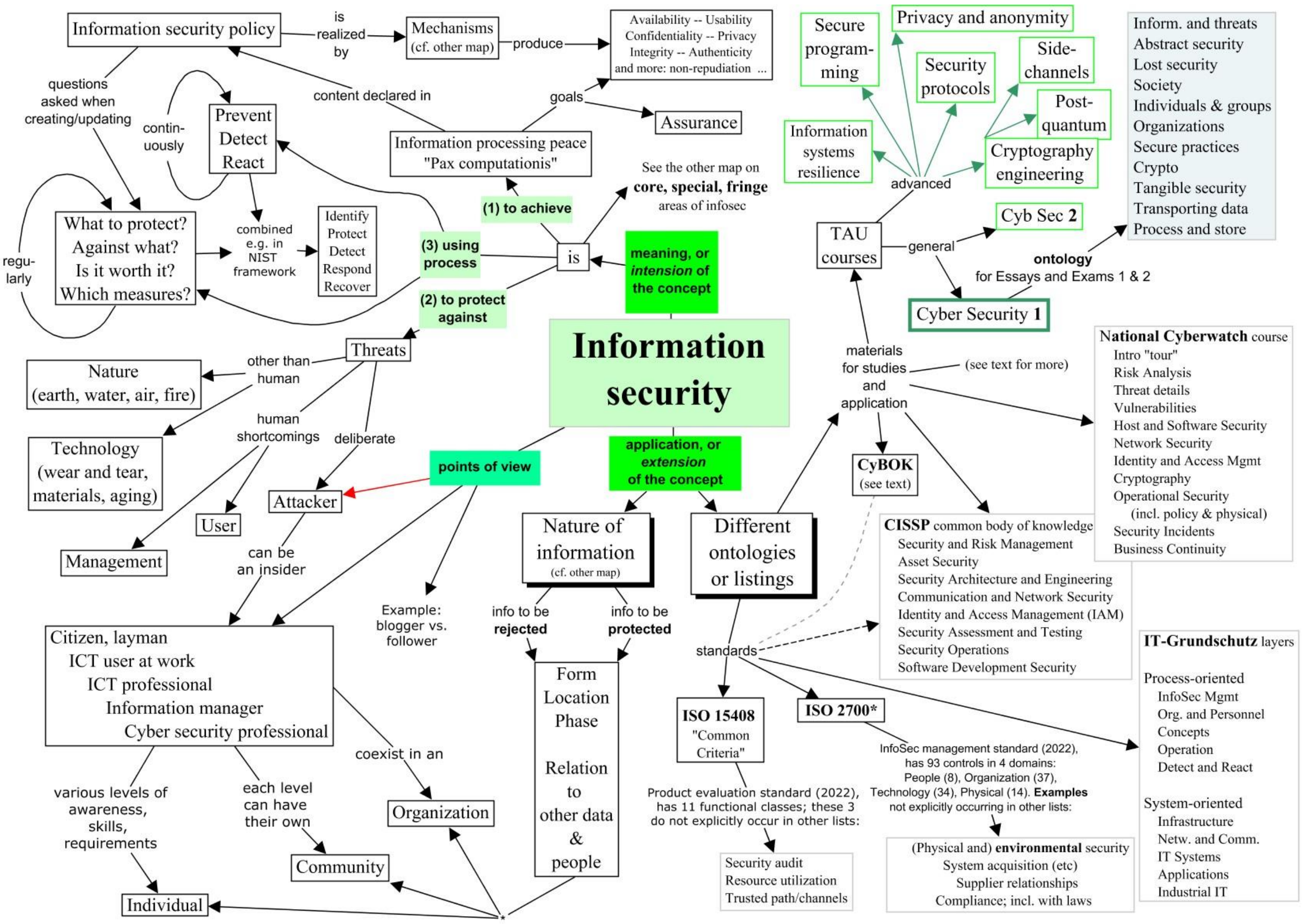


3. A map of information security

Below is a concept map that characterizes information security in terms of what it is and where it is. The *what* is the intension and the *where* is the extension of the concept. The extension makes a reference to the map of section 2 and lists various comprehensive administrative or educational breakdowns of information security. Most of them are briefly described next and one, the CyBOK, is given in more detail. The ontology of Cyber Security 1 can be found in section 7. And in section 9 you will see **ISO 27001** and **ISO 15408** ("Common Criteria"). The other **TAU courses** are not explained but you can find them easily in the study guide.

CISSP stands for Certified information system security professionals. You may have seen some multiple choice questions for that certification in a Harpo exercise. It is so well-known that it is partly linked to "standards" in the map. Another wide initiative in the USA is National CyberWatch Center. It is a government-funded consortium of educational organizations.

IT-Grundschutz is a German standard, also available in English. It has two volumes explaining the management aspects in somewhat similar way as ISO 27001, and volume 3 for risk analysis. The security mechanisms and application areas can be found in the [Compendium](final draft, 2021). It has 787 pages, but you can fairly easily pick the things you are most interested in. For instance, search the word “Introduction” and you can see definition of all sorts of vistas of information processing. If you search for the headings “Basic Requirements” you will have a good supply of protection knowledge for each case. “Standard Requirements” strives for higher security and there are also provisions for “increased needs”. Basic requirements are likely to be enough for your Essays and Exam 3.



[**CyBOK**] stands for Cyber Security Body of Knowledge. Its first version was released in 2019 and version 1.1 is from 2021. It includes a wealth of material for study as mentioned and linked in section 5 below. The content of CyBOK, the 21 knowledge areas are listed here, as an augmentation to the right half of the concept map.[1]

Human, organisational & regulatory aspects	Distributed systems security
Risk management & governance	Formal methods for security
Law & regulation	Authentication, authorisation & accountability
Human factors	Software and platform security
Privacy & online rights	Software security
Attacks & defences	Web & mobile security
Malware & attack technologies	Secure software lifecycle
Adversarial behaviours	Infrastructure security
Security operations & incident management	Applied cryptography
Forensics	Network security
Systems security	Hardware security
Cryptography	Cyber physical systems
Operating systems & virtualisation security	Physical layer and telecommunications security

CyBOK puts a speciality, the *formal methods*, together with ordinary security concepts. Such methods are beyond basic courses (but see a remark in sec. 30). It is a rather modern feature of CyBOK to give emphasis on the *human aspect*, including privacy and the adversarial side. When you study the map

[1] If the mind map here looks confusing you can review the CyBOK map of knowledge area relationships (see page 2 there).

above, you will not find similar in CISSP, IT-Grundschutz, NationalCyberWatch, or in the two ISOs. Identity management (or IAM) is not the same, but of course Personnel (at IT-Gr.) and Supplier relationships (at ISO 2700*) come close, albeit only partly.

The "**NIST framework**" combining two aspects of the security process at top left of the map, is Framework for Improving Critical Infrastructure Cybersecurity (v1.1, 2018). Of course, it would also belong to "standards" in the map. The five functions, or processes as marked in the map, each with 3–6 categories, are clearly expressed on page 23 of the NIST framework, and the following 21 pages describe what should be done, with only one sentence for each of the 107 subcategories. These sentences may give you a quick overview of what it takes to improve cybersecurity. But to be useful in practice many sentences, like "Remote access is managed." (PR.AC-3) need further explanations. For that purpose, the framework relies on detailed references to six standards. They include ISO 27001 but not ISO 15408, which fact is a reminder of the nature of these two, both appearing in the map. Section 13 in this document outlines the two processes of different time scale.

4. Cybersecurity curricula 2017

The standards are of course not intended as learning materials, but they can be useful to get a grasp of the entirety of the study area. There is one standard meant just for that purpose. Actually, it is a joint effort by several professional organizations ACM, IEEE, AIS and IFIP – and not a standard but otherwise influential Curriculum *Guidelines* for Post-Secondary Degree Programs in Cybersecurity. Its shorter name is Cybersecurity curricula 2017 (CSEC 2017). Again, we see a different breakdown of the cybersecurity field, now to 8 "Knowledge Areas":

Data, Software, Component, Connection, System, Human, Organizational and Societal …

security. It is worth noting that unlike in any of the lists of the section 3 map, every item in this division starts from *outside security*. Especially none is "cryptic" or "risky". Risks step in everything of course and CSEC 2017 expresses them as one of the six **crosscutting concepts**. Their full list is:

Confidentiality, Integrity, Availability, Risk, Adversarial Thinking and Systems Thinking.

These build connections between the knowledge areas and are important to students' ability to understand the knowledge area regardless of the discipline they represent. Already within Tampere university the discipline can vary a lot. CSEC 2017 describes the following 5 basic "disciplinary lenses" and an unbounded number of mixtures (like geoinformatics and computational medicine).

- Computer { engineering | science }
- Information { systems | technology }
- Software engineering
- Mixtures: X Informatics, or Computational Y

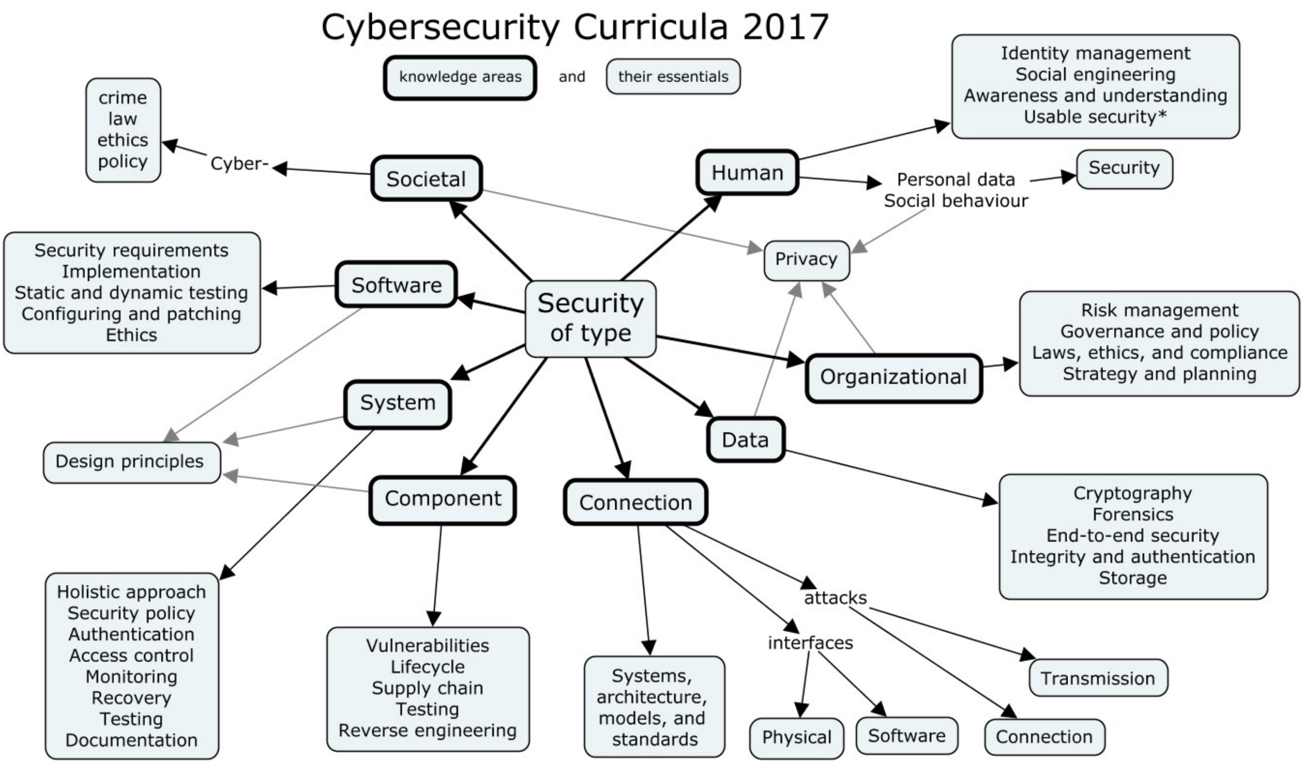
Acknowledging different points of view is a very good property of CSEC 2017. Some disciplines would traditionally not wish to share their core curriculum with others. The infosec concept map in section 3 mentions only ICT professionals (bottom left), but ICT could be replaced by any of the disciplines mentioned here.

The term *component* in CSEC differs from other divisions presented in this document. It contains trusted components (mentioned in secs. 23 and 25), but not only them, and not only hardware. Components are those pieces that can be designed, tested and maintained separately but are integrated into larger systems.

We would like to warn against the standard's notion of *integrity* as "Assurance that the data and information are accurate and trustworthy." This is wider than the ordinary meaning, which is the property of data remaining unaltered, except by authorized parties[2]. It is good to note also that risk

[2] So, both accuracy and trust go beyond ordinary meaning. NICSS glossary defines *data* integrity as "The property that data is complete, intact, [and trusted] and has not been modified [or destroyed] in an unauthorized or accidental manner." If you omit the bracketed words you'll get the ordinary meaning. Destroying data doesn't keep it, so it is availability that is lost. Check the glossary also for "integrity".

usually is related to something bad, but actually the standard defines it well by saying: "Potential for gain or loss." We lose nothing if we include the possibility of gain, too. Being trustworthy or trusted, and the whole matter of trusting is a complicated issue. It has its own section 25.



*) At the node "Human Security" the notion of usable security is mentioned, which CSEC actually does not enlist as an essential (but just a knowledge unit). Having the mindset of considering and improving user experience (a.k.a. usability) is likely to lead to improving also security.

In the map you see the eight knowledge areas of CSEC 2017, and their essentials. Two essential issues that appear in several knowledge areas have been marked, privacy and design principles. There is more overlap than this in knowledge areas, but these are probably the most important ones – but still not crosscutting concepts. You will see CSEC 2017 principles in section 23.

It is worth noting that although "principles" is not a crosscutting concept of all knowledge areas, and only three areas are marked to share some design principles, there is some generic knowledge in all areas. Such abstractions make learning details more effective.

5. Textbooks and other materials for this course

The node "materials for study and application" in the infosec map of section 3 points to several educational resources: the local TAU courses, the commercial CISSP, the course at National CyberWatch Center in the U.S. and the British CyBOK. The CSEC 2017 which is outlined in section 4 is similar to CyBOK in the sense that it describes what should be learned, but it does not include study materials.

CyBOK is different. It has an extensive, and free, text that covers the body of knowledge. You can find it as a 1067-page single text[3] and divided into 22 sections (version 1.1 July 2021). Please note also their page that links to other learning resources.

There are of course many "ordinary" books on Cyber Security, as well. Here are two good ones:

Charles Pfleeger, S. Pfleeger and J. Margulies: Security in Computing, 5th Edition, 2015.

Paul C. van Oorschot: Computer Security and the Internet: Tools and Jewels, 2nd Ed, 2021.

[3] Note that the actual body of CyBOK is just 770 pages while there are lots of supporting material, like a 156-page bibliography. The 12-page glossary is likely to be useful, but for an acronym first check the 21-page list of them!

The first one has, since its earlier editions, been a suggested reading on the two general TAU courses. This is not proof of its high quality but just follows from the fact that the teachers have used it for their work and have found it useful. The second book is a lot smaller, 446 pages vs. 944 of the first book. It omits many details but gives a very good (occasionally even mathematical) treatment of those issues that the author has chosen. It also has a carefully designed pedagogical approach, which includes the property of not including everything – motivating the subtitle "Jewels". Their collection grew with two chapters from the first edition, which was published in 2020, having 387 pages. The link is to a page where the whole book is openly available.

Books will not be what you need to pass this course even at the highest grade level, but they will be useful already now if you intend to study cyber security further. And you certainly need not study any standards for this course. However, another publication by NIST will be recommended and introduced in section 9 ("NIST-Intro"). Similarly, section 11 introduces another recommended text ("Cyber scenarios"). These two texts are also mentioned in the course handbook.

Videos

If you are more interested in listening than reading, you will find many introductions to cybersecurity on YouTube and other video channels.[4] It is often difficult to evaluate their quality without watching them through *and* already having a good insight of the topic. Concerning the length of the courses mentioned below, note that at 160 words per minute[5] you can read the text part of this document in less than two hours.

Iowa state university has a site for security literacy. There is a series of videos for their course. in 2024 those links seem very slow, but the videos are also put together to a single 5-hour video. This material is not intended for professionals, but it gives you a very well-presented introduction to the topic, including explanations of how operating systems and computer networks function. Email, online shopping, WLAN and social media are covered, but cryptography is not. The video mentions for instance hashing passwords and encrypting https-traffic, but it does not tell that https uses a protocol, TLS, that also authenticates the server with a public key and certificate etc. *You* must know this. For the general audience it is indeed better to emphasize, as on the video, that https does not protect the user from malicious websites, which also can use https showing padlock icons etc.

You will see 35 minutes of cryptography in the end of a 5½-hour video that bears the title CompTIA Security+ Full course. CompTIA, Computing Technology Industry Association issues professional certifications like Security+. The video is produced by Pace-IT, and it is very dense in professional detail on the slides. Already the beginning presents the OSI model and at 10 minutes you will get to know NIDS, HIDS and NIPS, which do not appear in this document. The presentation is very clear and not too fast. There are 48 sections individually linked.

A video very different from the two previous ones is Complete IT security course by Google, with length 7h15'. It relies on the very fluent talk by the speaker (reader), and occasional slides. It functions quite well with audio only. Even if it is for beginners, it is a continuation of computing and networking courses, and it contains nearly two hours of crypto related material. The table of contents is very detailed, but it seems to have flaws (duplications), and it does not show on the progress bar of the video.

Two Indian educational companies Simplilearn and Edureka have produced many free videos to promote their courses and certifications. They both have a video by the same name *Cyber Security Full Course in **8 hours***:

Simplilearn 37 sections; Edureka 28 sections

---

[4] An example collections is Computerphile that goes to selected deeper details, too, among various other areas of information technology. If you are on the TAU course you will have the whole TAU curricula to support you. If you are not, or if you feel the need of more preparations, you can check videos A+ and Network+ from the (certification) path recommended by this chat.

[5] "People who read books for radio or podcasts are often asked to speak at 150-160 wpm" (source), but remember that you can increase or decrease the playback speed by multiples of 25%.

Simplilearn seems to have a more helpful division of topics in their listing of sections. Edureka gives a 2½-hour section to "History of Cybersecurity", which actually is a full introduction cybersecurity, albeit with history. That section includes cryptography – even a setup of a new TLS certificate and the TLS handshake. Cryptography, however, has its own sections, including an introduction, later in the video. This seems to mean unnecessary (but perhaps instructive?) repetition and shuffling, but it is not fully avoided by Simplilearn either.

Simplilearn skips the topic of cryptography completely at the level of the listing. It is included in the final two hours that deal with CISSP certification. Several sample MCQ's are explained in detail, which happens during the last hour of the video. Edureka provides in a way a similar coda to their video by a section on interview questions, which means recapitulating essential points of cybersecurity "in front of" a recruiting board.

Some repetition is inevitable while both videos are collections of several sessions by different speakers. Another consequence especially at Edureka is that you have to get accustomed to different speakers of English. Simplilearn has mainly two presenters, and the female voice *reads,* the male voice *lectures.* Edureka's steganography may be interesting, but it is not much needed for this course.

There are also many videos from **live lectures**. MIT Open courseware 6.858 Computer Systems Security, Fall 2014 starts with Intro and threat models, and there are altogether 23 lectures online. Most of them also have the lecture notes available, in case you prefer to glance at them while the speaker writes on the chalk board. These lectures go beyond our Cyber Security 1, but they will be very interesting for further studies. For instance, at about 50 minutes of the first lecture a buffer overflow is studied at assembly language level using a debugger. The lecture ends with a remark on minimizing the trusted computing base, which is one of the design principles in the map in section 23. So, you will probably benefit by following the ideas of these lectures even if the details exceed your current goals. For many of you, lecture 20 on Android security will be useful in future. For some others the economics of spam in lecture 23 might be of independent interest and a stepping stone to understanding the cybercrime ecosystems more generally.

Other

A collection of various cybersecurity materials (and varying quality) can be found at FedVTE in the U.S. Those items that have a time indication contain videos.

This document does not address – and neither does the whole course – much of practical cyber security skills. Some of the courses mentioned above have hands-on exercises and there are several demos shown in the videos. There is, however, an important resource available to you – also for free – in the web, if you want to enhance your skills already at this stage outside the theoretical requirements of this course. Actually, you can (and should) choose to get some practice here, too, in the Harpo exercises, especially for the defensive side. On the offensive side there are many sites that offer far more interesting challenges than the single Harpo hacking exercise. Hands-on Learning Resources for Cybersecurity is a tabular listing of such pages and of also other free resources. Notice that there are more rows *and* columns in the table than shows in the first place.

This section has listed books, videos and learning materials beyond the needs of this course. Sections 3 and 4 do the same on standards, and sections 21 and 22 on guidelines. There are links to various sources in several other sections, too, but still the listing here would be incomplete without at least these additions:

scientific research portals, like IEEE *Xplore*, ACM, ResearchGate, Google Scholar,

cyber security authorities, like ENISA, list of CERTs, CERT/FI, NSA/US, DHS/US.

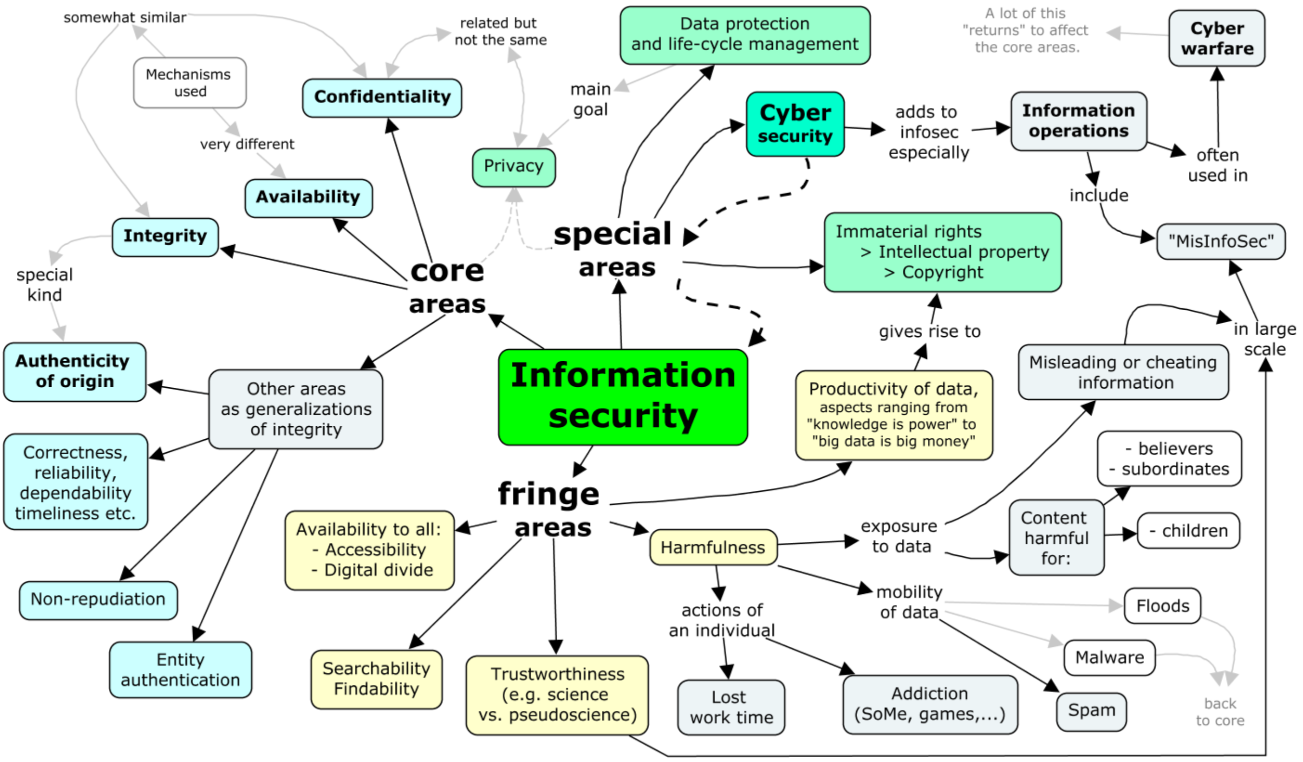
legislative portals in your own country (Finland > Translations, esp. Criminal code)

cybersecurity (-related) organizations, like GIAC, SANS, (ISC)², ISF, ISACA, ISSA.

security "activists" and their sites, like Schneier on Security and Krebs on Security.

6. Content areas of information security

The next concept map shows that there is much more to information security than C, I and A. This is the map that the "main map" of infosec (in section 3) refers to (from the node "intension"). Note that one of the fringe areas "Harmfulness" is more convenient to express with negative terms, and you have to invert them to grasp what information security tries to achieve.



Cyber security is here depicted as a special area (or actually an augmentation) of information security, but the matter can be seen the other way round. There is a very large engineering area "hidden" in the map as a generalization of integrity. It is **dependability**, which could be seen as covering everything here. And security is just part of it. The rest deals mainly with **safety**[6], but Wikipedia notes that these two are just recent additions. English language, your native language and different perspectives will provide a mixture of vocabulary for related concepts. The map above and the whole of this document aim to set the current perspective.

The term "MisInfoSec" means cybersecurity considerations about misinformation, i.e. (widely spread) false information and especially its intentional form, disinformation. The term misinfosec is not common. It stems from the article about weaponized information, which is introduced in section 18. The concept of misinfosec is a generalization of social engineering.

7. Content areas of the course, the ontology

The course contents follow an ontology of information security developed for this purpose. There are 3 major categories subdivided into 11 classes. This division is used to distribute the questions for Exams 1 and 2. The indented paragraphs below describe the classes more widely than the exam questions ask. For instance, p2p, voting systems and diffuse radiation are present here but there is nothing about them in Exams 1 and 2. You may find a topic for your essay exercise out of such fields.

[6] A bibliography of 1992-2014 by Paul and Rioux states: "Safety and security are two risk-driven activities that are traditionally tackled separately. … It is more and more recognised worldwide that both engineering specialties cannot continue to ignore each other." This course, however, continues to focus on security.

(1–3) What can happen if there is a lack of security, what can then be done, and how can everything be understood? (“Insecurity”)

1. Information security defined by looking at information and threats to it.
What needs to be protected and why, what can go wrong, what or who is threatening, how to survey the threats, what is a threat model? (Detailed threats appear in later classes – especially such that are against protections.)

Nature and vistas of information (incl. everything from human memory and hardware tokens to cloud services and internet as a whole). General definitions of information security (content of the concept: data processing peace, "CIA", etc.), threatening natural forces, technical problems, human error, administrative failures, intentional threats (from hacker ethics to malware economy to cyber warfare).

2. General features of security activities.
Most topics in this class can be understood without getting familiar with details in other classes, but a new level of understanding here will follow along with such details – and the other way round, too.

General principles (also “design principles”), perspectives, ontologies, architectural models, organizations, standards, information sources, measurement, testing, assurance, theories, research, security industry, professions, education.

3. Addressing emerging threats
What can be done if a threat is realized? The better prepared you are, the more you can do. However, only in the later classes will the emphasis be proactive in the sense of trying to control what can happen. This class is positioned here to provide a “final frontier” in facing the threats, but many advance preparations are covered here.

Response preparation (planning, procurement etc.), logging, duplication (including backup), detection (including antivirus and IDS), response, recovery, continuity, tracking (forensics) and follow-up.

(4–6) How are people situated in the field of security?

4. Security features of society
Societies, nations, economies and other large networks of people have structures, trends and missions that make some threats more possible than others.

Security of critical infrastructures (also from a corporate perspective), strategies, regulations (including cybercrime, copyrights, data protection), information society, cybercrime ecosystems.

5. Community and individual security aspects
People in their daily life and as members of local or network communities constantly face situations where information security comes to play.

Infosec culture, protection from harmful information and scams, privacy in practice, usability, awareness, ethics and other human factors, plus identity and trust management.

6. How are security measures selected and managed within an organization?
This class covers administrative information security at the level where employees, plans, euros, etc. are discussed, but not usually bits, volts, or TCP ports (which are the topics in classes 11, 9 and 10 respectively).

Strategies, policies, infosec management system, risk management, personnel, auditing.

(7–11) How do technologies help in the field of security?

7. What is the secure way to access and operate information and systems?
This class differs most from the traditional ones by putting together operational security, data security and other practices in information systems. This is the result of considering the title question in the case of normal use of information systems, where users and administrators must apply security-enhancing procedures. Some of them are quite independent security mechanisms, such as passwords or

encryption, and they cover some topics that would otherwise fall into later classes, mainly into cryptology.

> Access control, use of passwords, rules for different stages in the life cycle of information (including when to encrypt / sign e-mail), special systems such as healthcare, governmental systems.

8. Cryptologic methods
This class covers almost all cryptologic theory, but not any more practices like key management (in class 6) or usage of crypto (in class 7).

> Algorithms, implementations, breaking, protocols (including various forms of authentication), special systems like voting.

9. Physical and hardware security aspects
This class encompasses everything that is "tangible" except for people, though they are included in the biological sense.

> Access control, facilities, power supply, anti-tampering, biometrics, security devices, trusted architectures, critical systems and industrial automation, diffuse radiation.

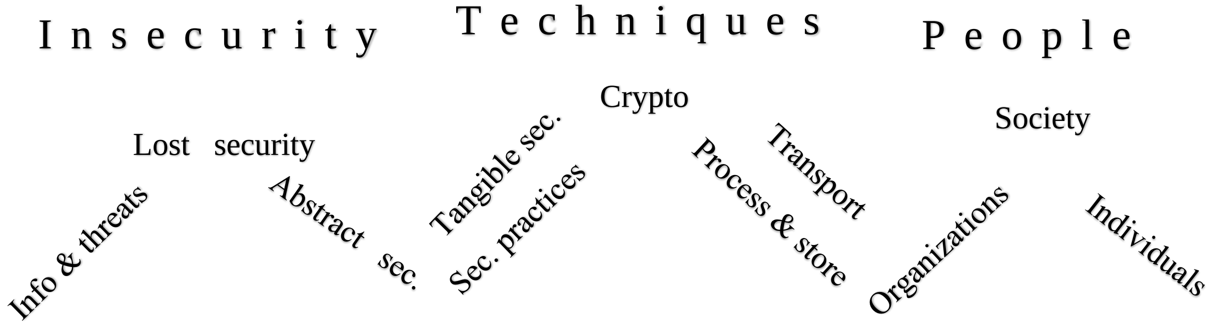
10. Securing the data network and mobile data
The data network is the basis of all kinds of data processing nowadays. This class applies in practice the theoretical knowledge from the classes above, especially from cryptology.

> Secure structure (including hardware) and management of a network that is wired / wireless and of type enterprise / backbone / ad hoc; filtering; security protocols in theory and practice (e.g., IPsec and TLS); special systems like p2p, digital TV, WLAN, Bluetooth, 4G, RFID. Vulnerability testing.

11. Securing "stationary" data processing
Although the movement of data has already been covered in class 10, most phenomena in this class still move data in networks. Here, motion is less essential in implementing security than in class 10.

> Databases, software, operating system, programming, embedded systems.

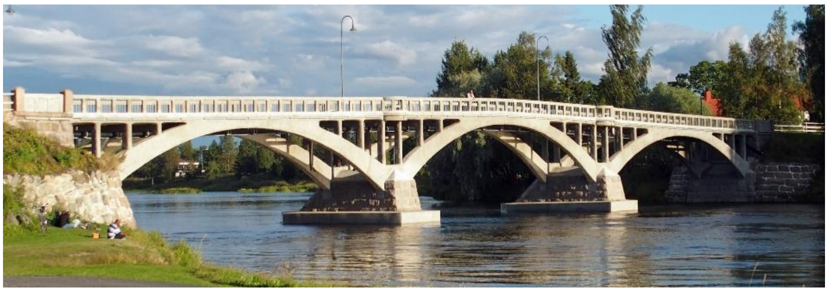
You must cover all 11 classes to pass this course even at the lowest level. You can imagine that the three major categories are arches of a bridge, where the five classes of “techniques” are connecting the arch of “insecurity” – consisting of data & threats, breaches and abstractions – with the arch of “people” consisting of different levels of human activity.



If you do the Exam 1 only, you must still build the whole bridge for yourself and cross it, but your bridge will look like one of these two, a very simple one or something that gives promise to continue to the next exam.

The latter picture may help you to realize that every stone is equally important. However, both bridges are quite narrow.

Mastering Exam 2 will make your bridge more like this:



And Exam 3 will hopefully start your mental process of “reconciling” the various categories into a smooth entirety like below, but especially it will be about giving light to specific spots on – or under – the bridge. This is the purpose of your work in the Essays exercise. But you can see light posts already in the “Exam2 bridge” as a reminder that Harpo exercises will be useful for better grades.

When you continue your cybersecurity studies, you can still build on the same three arches. And you may even forget, where their exact division is, because that is not essential. The firework display in the last picture is to remind that you will also write a sparkling thesis at a specific topic.

You may still doubt whether this bridge analogue will help you to memorize something important while there are already so many charts and maps in this document. You may augment the utility of your kind of bridge by thinking that there is a continuous and constantly changing flow of *adversarial activity* beneath it. The bridge analogue is about your knowledge and skills, though, and its applicability may end here. [7]

[7] The bridges are: (see the links for the sources and the type of CC licences. All pictures here are cropped from the originals)
Harrström bridge in Korsnäs, Finland; Narrow Bridge in Bourton-on-the-Water, UK;
Tulkkila bridge in Kokemäki, Finland; Myllysilta in Turku, Finland; Ponte Vecchio in Florence, Italy

8. List of concepts and acronyms

The MCQs for Exam 1 have been made around these 90 concepts. The words here are links to Wikipedia, usually with some modification to the term. Some remarks are given in footnotes.

2FA	Cryptoprotocol	Hoax	Phishing	Social engineering
Access control	Cryptoalgorithm[8]	Identity theft	PKI	Spam
Accountability	Cryptography	Impersonation	Plaintext	SSH
AES	Cyber attack	Integrity	Policy [10]	Steganography
Anonymity	Data protection	Intellectual property	Privacy	Stream cipher
Attack vector	Defence in depth	IPsec	Pseudonym	TLS
Authentication	Denial of service	KEX	Public – private key	TOR
Availability	DES	Key	Ransomware	Trojan horse
Backdoor	Dictionary attack	Log in	Risk [3]	virus
Backup	Distributed denial of service	Malware	Risk analysis [3]	VPN
Block cipher	DMZ	Man in the middle	Risk management [3]	WPA
Botnet	DRM	Non-disclosure agreement, NDA	Sandbox	X.509 certificate
Buffer overflow	Eavesdropper	Non-repudiation	Script kiddie	Zero-day
CAPTCHA	Encryption	One-time pad	Security control, mechanism, ….	
Certificate authority	Entropy	Overwriting	Separation of duties	
Challenge-response	Exposure[9]	Packet filter	Shared secret	
Checksum	Firewall	Passive attack	Signature	
Copyright	GDPR	Password	Single-sign-on	
Credentials	Hash	Password salt		
Cryptoprimitive		PCI-DSS		

Here is a list of names and acronyms appearing in Exams 1 and 2, partly repeating the above list, but fairly complete in that matter. Cryptographic concepts are given in italics (but not PKI concepts). Gray colour indicates those entries that the exams use in a very restricted context or are otherwise such that you need not memorize them at least on this course.

.cmd	CAPTCHA	DRM	IMEI	MFI	PPP	SQL	USIM
.com	*CBC*	DVV	IMSI	Microsoft	PUK	*SSH*	VLAN
.pif	*CBC-MAC*	*ECB*	IoT	MobilePay	*Rabin*	suid	VLR
.scr	CPU	*ESP*	IP	NFC	RAM	SYN	*VPN*
.tar	CRL	EU	IPR	Nigeria	RBAC	TCB	Wifi
.txt	CSRF	Excel	*IPsec*	NTP	*RC4*	TCP	Wikipedia
A5	ctrl-Z	GDPR	IPv6	OS	RDF	TCP/IP	WLAN
ACID	CVE	HLR	ISO	OSI	ROM	*TLS*	*WPA*
AES	DAC	*HMAC*	Java	Oulu	*RSA*	TOCTOU	X.509
AH	DDoS	HR	JPEG	OWASP	SaaS	TOR	XOR
AI	DEP	HTML	Kerberos	PaaS	SCADA	Traficom	XSS
Amazon	*DES*	HTTP	*Kerckhoffs*	PayPal	sgid	Trojan	
API	*DH*	HVAC	LAN	PC	*SHA...*	Turing	
ARP	DLL	IaaS	Linux	PCI-DSS	SIEM	UDP	
ASCII	DMZ	IAM	MAC	*PGP*	Siirto	UN	
Blowfish	DNS	IEEE	*MAC*	PICS	SIM	Unix	
Bluetooth	*DNSSEC*	*IKE*	*MD5*	PIN	SMS	URL	
CA	DoS	IMAP	*MDC*	PKI	Smurf	USB	

[8] Learn the concept from examples.
[9] This is a link to Wiktionary. Make the distinction to e.g. *disclosure* using ordinary meanings.
[10] Prefer the NIST-Intro (section 9)

It is worth noting that this list does not contain all the acronyms and names that you will encounter and even need to learn during this course. For instance, section 9 already has one in its title, NIST, and refers to another one, ISO 27001. Among some further ones there, at least ISMS is handy to recognize.

9. NIST-Intro

For the student to get a concise and holistic view of the course topics, *NIST Special Publication 800-12, Revision 1*, (2017) is recommended[11]. It bears the title *An Introduction to Information Security*. It suffices to study the main body of the document, pages 7–70 plus section 1.4. There are several topics that will not appear in the exams. The rest of this section gives advice on reading this document, the *NIST-Intro*.

The NIST-Intro is a brief text written in a comprehensive handbook style. The course is not dependent on it or even its vocabulary. For instance, the course rarely uses the terms adversary and adversarial, but instead attacker and malicious. More notably, the course does not include authenticity and non-repudiation under the concept of *integrity* (cf. sections 1.4, 10.20 and Glossary). The term *authorization* is used in the course mainly in the narrow meaning of granting access rights, while the NIST-Intro deploys the term in more complicated managerial contexts (cf. 6.5 and 7.1).

From chapter 1 only section *1.4 Important Terminology* is needed now, but it is good to be aware of the U.S. context given in the rest of chapter 1. Similar legislation and documents exist in other countries. The rest of the NIST-Intro makes frequent referrals to the documents outlined in chapter 1, mainly from the NIST or FIPS[12] publications. They are of course not the topic of this course.

The roles described in chapter 3 are not to be taken too “seriously” (i.e. learned verbatim), but each of them shows something that needs to be done with respect to information security. This also includes the supporting roles in section 3.17.

Note that the insider threat is much more than explained in the example of 4.1.2. It is almost common sense that it is more often related to some gain to the insider than just retaliation through sabotage when being terminated. Section 10.16 Personnel Security helps complement the view.

Chapter 5 Information Security Policy is administrative but very important also for an engineer.

A sidestep to ISO 270**

It is worth noting here that ISO has published a whole set of standards concerning various administrative aspects, together specifying an *information security management system*, ISMS, for an organization. The standard series spans 19 documents in the numbering range 27000–27021 whereby it is called the ISO 27000 series. It is outlined in the freely available introduction ISO 27000 (Overview and vocabulary, 2018)[13]. You must learn from what you are reading just now that **ISO 27001** (2022) is the most important among these, because organizations can (and sometimes must) get their ISMS certified against it. That is, they get audited, pay, and are given a “badge” that says: ‘We are managing our IS well enough.’

This sidestep is worthy also because the NIST-Intro does not mention the ISO 27000 series at all. This little fact is worth learning also as such, because it shows how different standard bodies and different nations can see things differently and do things in their own way. The ISO 27000 document does not, ‘of course’, refer to any NIST publication. As a final detail, the vocabulary in it defines *integrity* simply as “property of accuracy and completeness”, i.e. without requiring authenticity and non-repudiation.

---

[11] NIST=National Institute of Standards and Technology (USA)

[12] Federal Information Processing Standard (USA).

[13] The site ISO27k information security of a consult firm also offers nice listings and personal evaluations of the standard.

Back to NIST-Intro

The brief introduction to the NIST *Risk Management Framework* in chapter 6 is good to read, but as a framework (among many others) it will make more sense on later courses that delve deeper into risk management.[14]

Chapter 7 *Assurance* provides more insight into its topic than is required in the exams. Assurance does not appear on the course in one place but dispersed into various places and usually without being named but just linked to issues that need assurance. Trust is a related concept, but section 25 deals with it from a wider perspective. (Note also that *assurance* does not appear as a term in the ISO 27000 vocabulary.)

The title of chapter 8, *Security Considerations in System Support and Operations* reminds concisely that there are many activities needed to run an information system and most of these activities are not focused on security. And still security needs and opportunities must be considered in everything. The needs are easy to understand for instance in software support or media control (sections 8.2 and 8.5) but spotting opportunities for improvement may require more insight. Examples are shown for instance in sections 8.1 and 8.6 on user support and documentation.

Users are central in several sections in chapter 10 that describes categories of security controls. Section 10.8 *Individual Participation* may seem surprising as a security mechanism or control, but this may not be so if you come from the U.S. or somewhere else where GDPR is not in power; General Data Protection Regulation of the European Union, since 2018, has 99 articles dealing with many aspects of "PII", personally identifying information.

There are considerably more technical details on cryptography on the course, especially in Harpo, than in the NIST-Intro, but chapter 9 gives a very good context for the techniques. Together with a certain number of details such contextual understanding is likely to become of lasting value for most professions in cyber security. And the infosec program at TAU it is definitely useful, because the advanced courses have a strong emphasis in cryptography engineering.

Chapter 10 *Control Families* gives a breakdown of security controls into 20 families. As you become more familiar with information security, and partly on this course already, you may find this division interesting by comparing it to other points of view. One influential viewpoint is the Common Criteria standard that firstly divides security into functions and assurance. The functions (given as *functional requirements* in part 2 of the CC) mainly correspond to what is meant by controls in the NIST-Intro. There are 66 families of functional requirements in the CC standard. So, it is better for you to stick to the NIST-Intro. But you must know the existence of CC and its purpose of facilitating assessment of information security – and its idea of doing this by checking various *assurance* requirements to know that the security *functions* (required by the type of system) are effective. The same idea is of course present in the NIST-Intro.

Pay attention to the cost considerations at the end of various chapters. If you enter the profession you need to cause expenses that your employer might think are not justified because they do not bring any income.

10. Cyber and criticality

It is time for a final word on NIST-Intro, ISO 27000, and CC, the three documents treated above. The word is *cyber*. It does not appear at all in the latter two (except in a name of an institute in CC). In the NIST-Intro it appears 8 times on pages 7–70. Only one of the occurrences is *cyber security* and even that mention comes from a referred document. The other mentions are mainly about cyber attacks and criminals (van Oorschot's book is similarly void of cyber security). The name of the previous versions of this course was still *information security* in 2018. The terminological change is anticipated in the

[14] But you may appreciate the list of methods (incl. DIY!) and tools on the FAQ page of the same firm as in footnote 13. On the simple side you can gain some insight from the "Binary risk analysis", a diagram with just 10 questions.

Glossary of NIST-Intro on page 79: “Note: DoDI 8500.01[15] has transitioned from the term information assurance (IA) to the term cybersecurity. This could potentially impact IA related terms.”

Why is this course called cyber security? It is trendy, one could say, but it is true that no information security professional should any more ignore the security landscape outside his or her own organization. Any attack can also be part of something more serious that affects or is outright targeted at the critical infrastructure. This matter is not emphasized in the course contents which is rather traditional “InfoSec”. You may wish to take up this sort of topic in your Essay exercise.

To mitigate the lack of *cyber* in Exams 1 and 2, a quote follows introducing the “first version” of critical sectors and the way forward with learning “real” cybersecurity, including ENISA’s role in it. ENISA is the European Union Agency for Cybersecurity[16]. This is the executive summary of ENISA’s 2017 study, Stock taking of information security training needs in critical sectors:

> The European Union’s Directive on security of network and information systems (NIS Directive[17]) asserts that “*network and information systems and services play a vital role in society*”, and that the “*magnitude, frequency and impact of security incidents are increasing, and represent a major threat*”. Given that urgency, the NIS Directive goes on to argue that “operators of essential services” need to identify “which services have to be considered as essential for the maintenance of critical societal and economic activities”. This is in fact referring to the operators in the so-called **critical sectors** […]:
>
> - **energy,**
> - **transport,**
> - **banking,**
> - **financial market infrastructures,**
> - **health sector,**
> - **drinking water supply and distribution, and**
> - **digital infrastructure.**
>
> The protection of these seven critical sectors should have the highest priority, because when they are under threat, the functioning of society itself and the well-being of its citizens are at stake. As part of this effort, it is extremely important to increase the competences of cyber security personnel. This requires the availability of high quality trainings across the board, available to all critical sectors.
>
> Within the critical sectors, there are significant differences regarding the maturity level of cyber security. Therefore, some of the critical infrastructure operators will not be as ready as others, to counter the risks resulting from new cyber security threats in a timely and adequate manner.
>
> With the emphasis that the NIS Directive places on the importance of the seven critical sectors, this study aims to identify the current situation in these sectors in regard to the available cyber security trainings, and if there are any training needs specific to each of the sectors, beyond the generic needs for such trainings.
>
> Over the past years, ENISA has developed a wide range of cyber security trainings, and also delivered the training content to several national and governmental CSIRTs (Computer Security Incident Response Teams) as well as their constituents. The next important question that this study set out to answer is if and how the ENISA training portfolio actually is useful for the seven critical sectors – and what could be done to improve the suitability of that portfolio to the existing training needs.
>
> The main general conclusions are:
>
> - the cyber security training field is extensive and diversified, but does not sufficiently address the issue of raising the cyber security resilience of critical infrastructure: CIP-related trainings are still a niche

[15] The DoDI reference is to a Department of Defense Instruction from 2014.
[16] From its founding in 2004 until 2019 it was European Network and Information Security Agency, giving the acronym.
[17] The original version, adopted in 2016.

- there is a shortage of specialised trainings in the field of ICS/SCADA[18] systems cyber security – which is an essential element in countering operational threats (e.g. in the energy sector)
- there are very few trainings specialising in the specific threats encountered in the different (sub)sectors
- cyber security awareness raising trainings are lagging behind
- there is a shortage of trainings in regard to decision making as a result of data leakages or privacy incidents
- there is a pressing need for trainings related to GDPR, since this will affect every sector, and could have an operational impact on the organization.

Here is another quote[19], relating both to the term Cyber and the role of cyber security in organizations:

> It should come as no surprise that CISOs are still clawing their way into these closed-door sessions [=board room meetings]. After all, with other functional disciplines spanning centuries, […], the CISO's role goes back only decades. Back then, we didn't even call it "cybersecurity." These early pioneers claimed their profession as "Information Security," consistent with their field's beginnings in "Information Technology."

You can find an American perspective on Cybersecurity and Critical Infrastructure in a course module at the Federal Virtual Training Environment.

It can be said that identifying the criticality of certain areas in society has contributed to the shift toward "cyber" from mere "information". To emphasize this, the simple briefing above with 7 critical sectors is now enhanced to 12+5 sectors according to the **NIS2 directive of 2023**.

1. Energy (electricity, district heating and cooling, oil, gas, hydrogen)
2. Transport (air, rail, water, road)
3. Banking
4. Financial market infrastructures
5. Health (including labs and research on pharmaceuticals and medical devices)
6. Drinking water [until here the list is the same as the old one]
7. Wastewater [new item, inserted before the next one which is in the old list]
8. Digital infrastructure (internet exchange points, DNS service providers, TLD name registries, cloud computing service providers, data center service providers, content delivery networks, trust service providers, providers of public electronic communications networks and services)
9. ICT service management (managed service providers and managed security service providers) [new items from here till the end]
10. Public administration
11. Space
12. Food

Five *important* entities (not *essential* like the above sectors of high criticality)

13. Postal and courier services
14. Waste management
15. Manufacture, production, and distribution of chemicals
16. Manufacturing
17. Digital providers and research

[18] ICS = Industrial control system, SCADA = Supervisory control and data acquisition.
[19] From chapter 2 of Allison Cerra: The cybersecurity playbook, 2019. CISO = chief information security officer.

Is there anything in a society that remains outside this listing? Many things remain, like entertainment, personal services, tourism, parts of agriculture (cf. 12. Food), retail and wholesale etc. Also, education is not included. However, the quote above emphasized the role of cyber security education, and you will see in section 12 that education in general can be of vital importance for cyber security in a society.

11. Cyber scenarios

This section describes another text for you to study to complement this document and the NIST-Intro. Let's call the text *Cyber scenarios*, while its full name (and direct link) is "Cybersecurity: Exploring core concepts through six scenarios". It is published in 2017 by a research group led by Alan Sherman at UMBC, Maryland. There are sections for educators, but the rest is intended for students, most part of the pages 4–32. The authors actually note that their "primary target readers are students in any first course in cybersecurity."

The scenarios are thoroughly explained and some misconceptions from the authors' experience are described. The scenarios are about

1. E-mail
2. Drone delivery of packages
3. Data base
4. Network boundaries
5. Nuclear arms surveillance
6. A USB stick

While elaborating through these, the article teaches the following topics (the scenario number is given in parentheses):

- Deciding and specifying what level of security you want by setting up a policy (1).
- Realizing and deciding against whom and what you will defend: adversarial model (1, 2).
- Analysing vulnerabilities, threats, risks (2, starting from the definitions).
- Understanding injection attacks and appreciating hints towards prudent secure programming practices (3).
- Familiarizing with the foundation for secure communications (1), which includes:
- elementary cryptographic details (1) and later specific use of public-key cryptography (5).
- Augmenting communication concepts into a real system, which means segmenting network, and isolating processes (sandboxing), and using a speciality: data diode, (4).
- Learning about firewalls and VPNs, which are both discussed at (4) but evaluated not to be sufficient in this case.
- Appreciating physical security and social engineering – kind of two extremes in a single setting (6).

12. Social influence

As a concise introduction, this is a quote from L.D. Freedman's review of the book *The Weaponisation of Everything: A Field Guide to the New Way of War*, by Mark Galeotti, 2022. Although blood shedding reappeared in Europe right after the publication:

> "this is still a valuable and accessible guide to the insidious methods adopted regularly by the Russians and others to wage war by more covert means. The use of
>
> - espionage,
> - propaganda,
> - bribery,
> - counterfeiting, and
> - extortion,
>
> often developed in collaboration with criminals, is hardly new. Galeotti … reflects on how the

- Internet,
- social media, and
- the interconnectedness of modern societies

have provided new opportunities to manipulate all aspects of everyday life."

Weaponized Information: Influence and Deception in the Age of Social Media, is chapter 29 in a book[20] that deals with military matters, but here the authors, Perlman, Breuer and Terp give an abundance of insight into their topic suitable also for less military cyber security. Some concise definitions will be quoted here. Firstly, the authors use the term *social media* to refer to "the collective sphere of socially oriented digital communication", whereas a *social networking service* such as Twitter or Facebook consists according to them of:

- a computer system (i.e., servers, custom software) designed for the primary purpose of facilitating human users sharing human-readable information (e.g., words, pictures) with each other.
- a population of users, including individual humans as well as other account-holding entities.
- the collection of human-scale information hosted on the system (e.g., "posts," "tweets," "memes," "hashtags").
- the metadata stored on the servers, defining the graph relationships of the network, such as who is "friends" with whom and who has "liked" which posts.

It is an example of *sociotechnical* system, and the authors seem to coin the term "misinfosec"[21] to mean the research and practice on sociotechnical security. In simple terms the misinfosec is a generalization of social engineering to massive scale of effects that can be spread in sociotechnical systems rapidly. The attack surface is population instead of individuals.

The table below gives a comparison of these two scales of social influence:

<table>
<tr><th></th><th>Social engineering</th><th>Misinfosec</th></tr>
<tr><td>Target</td><td>Individuals</td><td>Large groups of people, entire populations.</td></tr>
<tr><td rowspan="2">Vulnerabilities</td><td colspan="2">Human cognitive biases[22] and fallacies,</td></tr>
<tr><td>or other human psychological weaknesses, like greed, or virtues, like desire to help, or just curiosity.</td><td>in interaction with biases in algorithms in recommendation engines</td></tr>
<tr><td>Exploits</td><td>techniques of psychological manipulation and deception</td><td>online ad campaigns, click baits, troll farms, click farms, sock puppet[account]s, botnets, control of traditional media.</td></tr>
<tr><td>Attack vectors</td><td>an email, sms, call, or other message targeted at an individual</td><td>memes[23], narratives, hashtags, and other forms of information trends, with topics like nationalism, religion conspiracies, sensations.</td></tr>
<tr><td>Outcome of a successful attack</td><td>generally, credentials or some other form of access to an information system.</td><td>changes in behaviour or beliefs and the real-world effects of those changes, such as political, economic or market movements, e.g. election results, protests, riots, boycotts.</td></tr>
</table>

[20] Strategic latency unleashed: the role of technology in a revisionist global order and the implications for special operations forces. Edited by Davis, Gac, Rager, Reiner, and Snow. Center for Global Security Research, 2021. There are some other articles in the book that might be of interest to you as a cybersecurity student: Two are about the block chain: Ch. 19: *Cryptocurrency: Will the Digitization of Currency Allow Malign Actors to Achieve Strategic Effects?* And Ch. 20: *Blockchain and the Battlefield.* One is about identity: Ch. 11: *As the Helix Turns: How New Biology, Biometrics, and DNA Analysis May Forever Prevent Anonymity*.

[21] Misinformation can be studied even under the term Computational Propaganda, see https://comprop.oii.ox.ac.uk/.

[22] Wikipedia has dozens of these. This infographic has 50 but can be useful for memorizing some.

[23] Wikipedia: Internet_meme, a special case of Wikipedia: Meme.

The book chapter we are referring to in this section continues with some ways of defence, but no strong measures are proposed. Mainly detection is discussed. The best you can do now is to become aware of Misinfosec. One way forward is to read the 9-page appendix of the chapter, where the authors present and analyze a fictive disinformation scenario[24]. This analysis does include a list of defence preparations, but they are geared towards the context of the whole book, namely the defence *sector*. One item is worth repeating here: high-quality education is a national security priority, including education on media literacy and critical thinking, and particularly awareness of cognitive fallacies, and how they feed into journalism.

13. Information security is a process

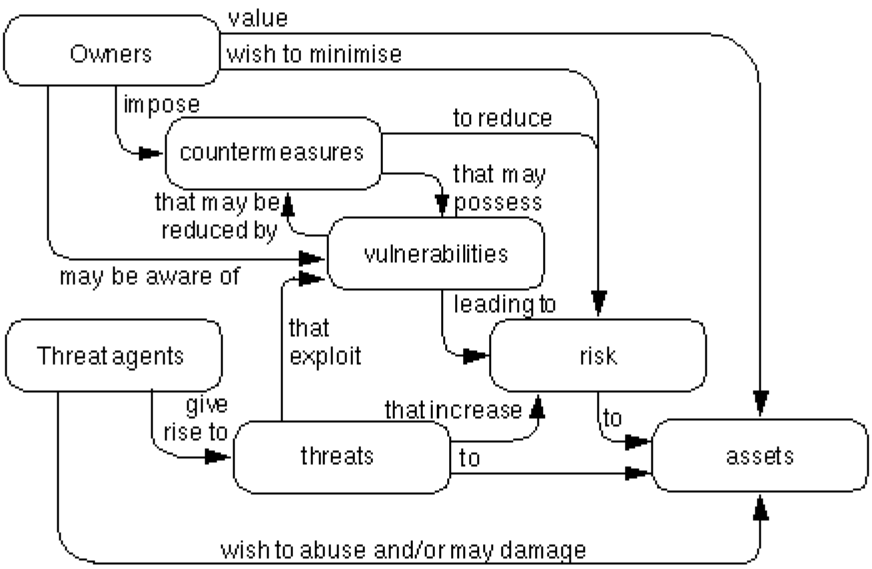
Information security is not the result of an activity in the way a computer or some other product can be. Nor can it be compared to a process in the sense of industrial processes or human metabolism. A person's physical condition is a useful metaphor to explain in what way information security is a process:

- Both good security and physical fitness require constant efforts to build and maintain them. This *ongoing activity* is largely administrative (for humans: awareness, self-discipline, couching etc). It is dominant and guiding in relation to the following:
- Physical exercise and healthy habits are needed for fitness, and they involve *momentary activity*. The analogue in information security is application of different forms of protections. Main types of them are:
  - Prevention (in a wide sense). Most security is related to this, and there is a common subdivision around prevention (in a narrow sense). A human analogue is given in parentheses.
    - avoidance[25], (don’t smoke; don’t go to dangerous places)
    - deterrence, (carry a visible weapon with you; not good for most hazards)
    - prevention, (vaccination, contraception – neither is perfect, unlike avoidance)
    - mitigation, (wear a cycling helmet; use vitamins and other antioxidants)
    - transfer of responsibilities, mainly through an insurance (helps humans, too)
  - Detection. An obvious example is virus scanning. In some cases, this is a process in the sense that a problem has to be monitored for some time before it can be handled (e.g. a suspected attack on a computer network)
  - Response. If the detection mechanisms have been in time, the response begins with
    - stopping. After that (also if detection was too late), the next steps may be
    - recovery (to continue the operation),
    - remediation (damage and / or the cause of the problem) and
    - possibly punishing the culprits.

The constant need to build and maintain security stems from the fact that there is something going on all the time that can change the target of protection or its environment. Early version 2 of the Common Criteria standard summarized the main actors and actions in the diagram below.

[24] with characters like senator Forethought and general Backsight, apparently borrowed from a tactics book of 1904, also promoting critical thinking.

[25] A related method is to *deflect* attacks by providing other targets that look more promising, i.e. not only by making oneself less attractive. The fake targets are often called honeypots. A human analogue may be useful only for V.I.P.s.



For instance, these two sentences can be read from the nonlinear diagram: Owners impose countermeasures to reduce risk to assets. Threat agents give rise to threats that exploit vulnerabilities leading to risk to assets.[26] In version 3 of the standard, the diagram was simplified by removing the vulnerabilities. It makes sense, because there are more vulnerabilities elsewhere than inside (good) countermeasures. Of course, this diagram does not claim otherwise, and the extracted sentences avoid that route. However, studying such a diagram too strictly would give you an equally distorted impression as taking the analogue to human fitness too far. The diagram introduces one further concept to include in the analogue, namely the owner. Data and information systems are like your body: not loose entities but belonging to someone who is supposed to take care of them.

As there were already two different terms used above for the abstract methods that create security (protection and countermeasure), and a third one appears below (measure[27]), here is a list of them all roughly in the order of generality. The word security can be omitted when the context is clear. The last two terms are the least generic. A security provider is already an implementation of services[28], but the term security services can be used in quite an abstract sense.

- safeguard
- security control,
- security mechanism,
- protection,
- security measure, or countermeasure,
- security service,
- security provider

The process of getting any of those “security creators” into use in an information system can be divided into steps, such as the following:

1. Find out what is valuable and worth protecting (usually called assets).
2. Identify threats, vulnerabilities, attacks.
3. Assess the risks.
   If they are low enough, stop and return to the topic again at an appropriate time.
4. Prioritize vulnerabilities.
5. Select and install security measures, return to step 3.

The return instructions in steps 3 and 5 extend this one-time process to continuous. If there have been changes in the system, the return from step 3 must be made to step 1. In fact, it is the changes that give a reason to return to the topic. If no internal changes happened and the “appropriate time” just comes

[26] Such a sentence also appears and is explained in OWASP Secure coding practices quick reference guide v2.0, on page 4.
[27] Security can be also measured (gauged) to some extent. That sort of measure is distinct from this measure.
[28] For instance, a cryptographic service provider, CSP, can be a software library.

from the good practice of regular checking – once per year for instance – then that return may jump to step 2 instead of 1.

This division into iterative steps can be considered as a basic model of security work. In slightly more abstract terms, the same process can be described by the following steps:

(i) evaluate
(ii) plan
(iii) implement
(iv) maintain and monitor.

Step (iv) represents partly step 1 above, partly it is within the countermeasures in step 4.

Costs must be considered. They are unduly increased by both

- very strict security measures: the cost of acquisition and maintenance rises; and
- lack of sufficient security measures: the cost of losses rises.

A compromise has to be found somewhere in the middle, but it is unlikely that there can ever be an exact numerical basis for optimization.

In steps 1–4 of the basic model you need to know your own system, but in step 5 you need to have a lot of information about security mechanisms. The goal of security courses is to tell you about some of them and provide references to others. While many of the mechanisms are in principle simple, the challenge is the diversity, specificity, and detail of these mechanisms, and still a degree of uncertainty, especially when multiple mechanisms are combined.

How do you know which costs are too high or what risks are low enough? In order to answer such questions, you need to know what you want. Someone needs to decide the desired level of security for a system or organization and how much they are willing to pay for it. This decision is called an information security policy and is typically made by the management of the organization, even if the managers would not be the actual (or nominated) data owners. The policy also shows how the responsibilities of the security process are shared.

14. Generally about threats

The need for security is easiest to realize by looking at what bad can happen. On the other hand, information security is defined through some positive features. These qualities, integrity, confidentiality, and availability naturally tell us something about what threats are avoided through them. However, here we take a very abstract look at what bad can happen to information or its processing.

Information can

1. disappear or be destroyed.
2. be transformed or replaced with other data. In such cases the information may seem intact, but its integrity is still lost.
3. be revealed to parties who should not see it. Here the information can be revealed either
   a) as such, like a secret document on paper or as a digital file – even if erased, or
   b) in such a way that an observer makes inferences about sensitive activities, like who communicates with whom, or the interval of keyboard presses (by recording the sound).
4. spread without proper compensation – through copying or theft. These are usually followed by use, which is then unauthorized (see next).
5. to be used without permission, which, in addition to the previous one, also means situations where the information does not (any longer) spread, but its usage exceeds the rights. In place of information here can also be a data processing resource.
6. be initially false, e.g., invented or lied, or just accidentally wrong. Computer programs in particular can be wrong and cause a lot of other annoyances on this list. Fake news is also a very widespread example.

7. be misinterpreted (unintentionally; otherwise see #9)
8. be irrelevant, or its use may otherwise fail. This is about data not making sense, because it is in wrong (or deprecated) context, format or language or the computing resource is unusable. The last point is a very common failure, and it is not only due to denial-of-service attacks. Machines and software can fail for many reasons.
9. be rejected, e.g., when it is not received ("heard but not listened") or it is denied (active misinterpretation). A special case is the denying of information to which one has previously committed in one way or another, e.g., an agreement or activity from which observational information has been accumulated.
10. be abused – that is, used for harmful purposes, such as cheating, extortion, harassment. Such malice with information is, in principle, independent of whose the information is or who uses it, i.e., abuse here originally means other than unauthorized, although it may be that, too, like in identity theft. Originally you may also think of this category as abusing *correct* information. However, a large area of malice can be found when combining #10 and #6. Besides the rather personal harms mentioned here, the combination enters the field of fake news, disinformation, information influencing and information warfare (see sec. 12). And here it may depend on the point of view whether it is abuse or justified use – it *can* be lawful in a country to mislead its own citizens or people in other countries.

The list reveals the very wide spectrum of issues that one must be aware of when identifying risks to information. Many special systems may have ready-made checklists, but your own creative thinking is always an important complement to them. In addition, some active threats, i.e. those caused intentionally by someone, can be structured and even slightly anticipated according to what motivates the attacker.

Wrong kind of information seems to appear in the list in several places: broken integrity, false, misinterpreted, incomprehensible, rejected, abused. Of these six, only the first two are such where the information itself is wrong. In others, it is handled somehow in a wrong way or in a wrong context, and the processing fails or otherwise produces bad results. The difference between information, that lacks integrity, and information, that is initially false, may not be large, when you look at the data, but the breakdown is worthwhile because the control is usually different. Maintaining integrity by fairly technical means is enough, but it does not help against falsehoods.

Any of the above can happen to the information if the person who owns or guards it acts improperly. Even if he tries his best, his own knowledge or perception of what is right can suffer from similar problems. It may happen that there is no guiding information, it is wrong, or he does not understand it. It has already been mentioned above how an attacker can make his victim act wrongly. Directly targeting an individual's thinking and emotional activities can be a *social engineering attack*[29]. It is a personal influence, that results in disturbing the behaviour-controlling knowledge and emotions in the human brain for at least a moment (for instance by shifting the limit values of regulatory mechanisms). Thus, for its immediate effect, social engineering can be categorized at #10 above, but concerning data eventually outside the human brain the social attacker can achieve anything else on the list.

There is a scary visualization, originally from 2013, of large data breaches, especially disclosures of personal data, which are gateways to other data. When looking at threats against information, it must not be forgotten that the impact can extend far beyond any e-commerce server or storage cloud. A more primitive but much more painful example of these was provided by a 14-year-old boy in Łódź in 2008.

15. Physical threats

The scope of *traditional* information security is augmented when *cyber* security also considers, how the physical world is disturbed by information operations. Reversing this thinking means asking, what

[29] To understand social engineering properly, study the 6th scenario in the text introduced of section 11. Compare also with the "population-wide" version of social engineering in section 12.

kinds of physical operations can disturb the world of information. In other words, what are the physical threats to information security? While investigating them one must remember that physics ranges from electrons to tectonic plates, and human attackers, too, can use physical methods. This listing provides four kinds of physical sources of threats against information security. Most of the items are obvious and only contain the name.

Nature

- earthquake
- flood
- wind, felling trees over power lines
- thunder, causing electric surges and outages
- fire, from outside computing
- dust and other particles (volcanic, soil)

Technology

- loss of electricity, by various reasons, the smallest one being worn-out connectors
- electric surge, e.g. from switching on and off of other gadgets like AC compressors
- improper temperature, e.g. due to failure of ventilation
- radiation, from other devices, causing loss of data or disruption of wireless communications
- static electricity, e.g. when humidity is too low, causing harmful voltage spikes
- fire caused by computing, e.g. faulty batteries
- leaking water or too high humidity
- vibrations, e.g. from traffic or construction
- failures of devices, e.g. rotating disks; also solid state disks have a limited life span.

Human errors

- loss of devices
- accidental breaking of devices
- failure to maintain a biometric feature

Attacks

- stealing devices or media
- destroying devices, media or connections
- inserting bait devices or media with malicious content
- tampering, i.e. modifying a device physically, to cause malicious behaviour
- cloning of a device or media, resulting in an undetectable theft
- spying on signals out-of-band. e.g. on power cord, keyboard sounds or touch-screen traces
- spying devices inband. e.g. key logger, wiretapping, Wi-Fi or Bluetooth antennas
- jamming wireless channels with noise
- distorting electric supply, causing an outage or a surge
- EMP and HERF, i.e. weapon effects through electromagnetic pulse and high-energy radio frequency.
- plus "earth, wind, water and fire", i.e. any physical influence that can cause damage in warfare

It may seem from the lists that only intentional attacks cause problems with confidentiality or integrity, and the other sources are just problems of availability. Usually this is the case, but especially natural disasters can expose information systems to opportunistic attackers for other kinds of security breaches. The same can become true under exceptional technological situations, for instance when surveillance cameras fail, or a fire disables other protections.

16. Attacks and attackers examined

Computer security has been around since the 1960’s and a lot of research was done in the 70’s. The Internet is younger, and also a great source for security research, also early on, resulting for instance in

An Analysis of Security Incidents on the Internet 1989-1995. It is a dissertation by John D. Howard. He presented a taxonomy[30] of attacks, which is still useful. It is given in the table below, where the attacker of a computer or network can be classified according to the first column and the goal (motivation) he is aiming for is one of those mentioned in the last column. The goal is achieved through some knowledge-related result. The result is that information can be accessed by some means. Access is divided into four stages.

Attackers		**Tools**		**----------------- Access -----------------**								**Results**		**Objectives**
Hackers		User Command		Implementation Vulnerability		Unauthorized Access				Files		Corruption of Information		Challenge, Status
Spies	==>	Script or Program	==>	Design Vulnerability	=>	Unauthorized Use	=>	Processes	=>	Data in Transit	==>	Disclosure of Information	==>	Political Gain
Terrorists		Autonomous Agent		Configuration Vulnerability								Theft of Service		Financial Gain
Corporate Raiders		Toolkit										Denial-of-service		Damage
Professional Criminals		Distributed Tool												
Vandals		Data Tap												

The idea, then, is that different attacks can be modelled by selecting one (or more) options from each column. Because these are attacks over a computer network, one step is always the "process," that is, the fact that an attacker has launched a program on the target machine.

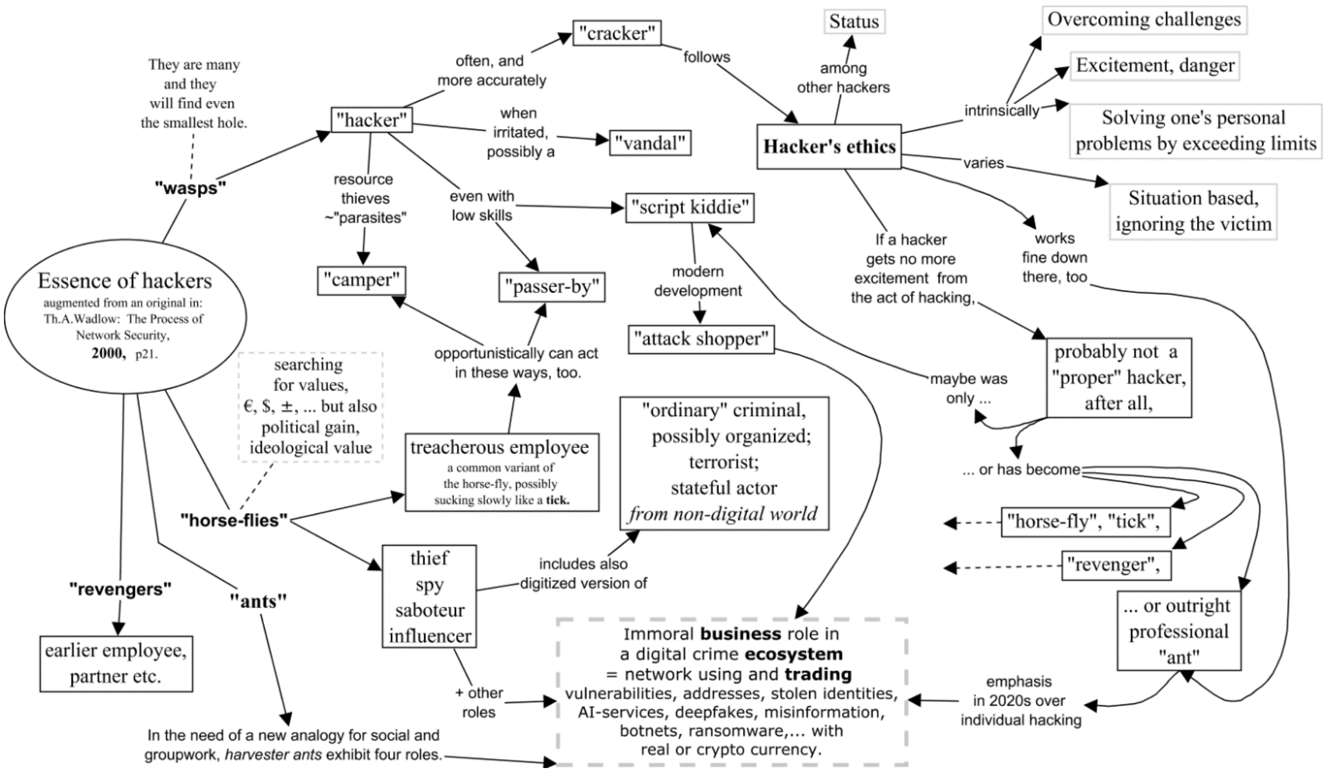
Since this study, the Internet has grown exponentially for a long time, and an increasing proportion of machines have a constantly open high-speed connection. These factors have contributed to the fact that the "Distributed tool" in the Tools column of the table has gained considerable weight with *botnets*. They are programs run on hacked machines and used remotely via a command channel (using traditionally IRC, http (already in 2010 much more commonly than IRC), P2P, and custom protocols) and suitable for all types of attacks mentioned in the Results column. The large quantity of machines has made botnets resemble armies. They have also introduced a new kind of motivation, or rather a new meaning to "Financial gain" in the last column of the table. Namely, a bot army commander can benefit financially by hiring it for various purposes. Bot networks also make it easier than ever to raise money through blackmail based on the threat of an attack. Another way to get victims to pay is to use malware to encrypt their files and demand a ransom for the decryption key: *Ransomware* was invented already earlier but cryptocurrencies made it handy in the early 2010s.

The concept map below parses the individual attackers and their motives in a slightly different way, partly with biological analogies. The term "script kiddie" is a dismissive designation for attackers who just use attack scripts compiled by more advanced ones, without necessarily understanding much about what is going on. The defender should not underestimate such attackers either.

A kind of contrast to script kiddies are the so-called APT attackers. The term APT, *advanced persistent threat*, has become common around 2010. The name is quite illustrative. Persistence is roughly the same as the horse-flies in the concept map in the sense that an APT is motivated specifically for a particular target and does not give up easily. In many cases, there are state actors in the background.

Non-state actors, too, may aim for the "political gain", because it can include all kinds of ideological goals. On the other hand, private hackers can participate in state-motivated cyber-attacks (e.g., from Russia to Estonia in 2007, and much more widely esp. in 2022), or they can "organize" themselves on the basis of other types of ideas into activist groups, such as the infamous Anonymous.

[30] A **taxonomy** divides an entire field into mutually exclusive categories. Such a classification can be used, for instance, to compile statistics on actual security threats, in which case conclusions can be drawn about the probabilities of their occurrence. This information, in turn, can be applied to security planning through risk analysis. Concept maps without cycles are often taxonomies, and those with cycles and explanations between concepts often represent **ontologies**. Actually, Howard's taxonomy is more than its name says. You'll see larger ones (linked) in section 18 where they have the name *matrix*.



There are many different characterizations of hackers. A common one is "black vs. white hat", the latter being a (truly) ethical hacker. This is often augmented with gray and sometimes more colours. Some hackers change their colour, also towards white. Hacker's ethics, or rather psychology, is a subject of scientific study, often through interviews: an example. The recent remark at the bottom of the map emphasizes that the importance of individual hacking has been superseded by ecosystems of business-like actors with various roles while the digital world has become the mainstream venue for profitable criminal activity, also with other means than hacking, e.g., phishing, cheating, influencing.

17. Security problems in a computer network

A computer network is a computing environment with multiple independent computers that are able to communicate with each other. Each machine can have multiple users, but otherwise the size of the machines and the distance between them are irrelevant to the operating principles of the network.

You should have some understanding of these concepts: Host, server, client, local area network (LAN), topology, gateways between networks, router, networks connecting networks (=internets), addresses, names, protocols and their layered structure (OSI model and TCP/IP model).

This is a list of network resources to be considered in the risk analysis, ordered by distance from the user: local nodes and links between them, the local network, its equipment, processes and stored data, e.g. control information; a gateway to an external network, and links from the gateway to the outside, external network routers, and resources such as databases.

The network has many useful features that also cause insecurity:

- Sharing resources is one of the benefits of networking, but it means that (i) there are more potential users (and at the same time abusers) and (ii) everyone has access to more machines than before, which may not always be the justified on the basis of earlier access rights.
- Complexity: the operating system of a single machine already is so large and complex that it is difficult to make it secure, at least so that it can be really trusted. When there are several machines and each has a slightly different configuration, the complexity of the entirety is closer to the product of the complexities of the individual systems than to their sum. This is not only a disadvantage, as different systems are not likely to be vulnerable in the same way. If there happen to be several firewall machines on some path of packets in the same network, it may be

beneficial to run a different operating system on them. However, on workstations of an organization, already differences between versions of the same operating system can do more harm than good.
- Vague boundaries. Network extensibility is one of the advantages of a network, but especially interconnecting networks means that a user may not know what other parties – and of what characteristics – may be connected to it at any given time.
- Multiple attack points. While the effort of handling data (processes and files) is spread across multiple locations (which is a good thing), the number of locations where attacks can occur is increasing, and the user has to trust on access control mechanisms and other security measures of many machines.
- Anonymity: An attack can take place from very far away and through several intermediate stations, making it difficult to catch or even identify the perpetrator. Authentication between machines is also not directly as reliable as between people. It can be made more reliable, but it requires carefully designed protocols. There are also protocols that strengthen anonymity. They can be used both for good and bad: safely reporting problems in real life, or planning crimes.
- Unpredictable paths: The availability of computing is enhanced by the use of alternative servers or traffic routes, but at the same time you may end up using less secure nodes or links than you would like.

The threats to communication in a networked environment are similar to those that threaten information more generally. They can be summarized in four points: exposing a message, altering a message along the way, blocking a message, and appearing of a forged message. As a fifth problem there is the unauthorized use of a remote resource over the network.

Let's take a closer look at what can go wrong in a network and how:

- Eavesdropping, "listening on the wire – or in the air". The question is whether it is worth sending anything without encryption.
  - **Cable**. In a local network (in the same segment) everyone hears everything, but the situation changes between wired switch- and router-based LANs. However, in such networks, eavesdropping on the traffic of others with "packet sniffers" (such as Wireshark) is possible by means of port mirroring or e.g. ARP spoofing / poisoning. In addition, the electromagnetic field around the conductor allows the use of induction, but the difficulty of this operation varies greatly between different types of conductors. Optical cables do not have this problem, and neither can they be wiretapped in the same way as copper wires. Outside LANs, transmissions are multiplexed and therefore more difficult but not impossible to disentangle.
  - **Radio path**. Transmission cannot be completely focused, and it travels anyway through space where someone else can set up a receiver. Satellite transmission has very wide spread, but the problem is not very serious due to multiplexing.
- Wrong identity, appearing as someone else (impersonation) through
  - guessing or eavesdropping on passwords.
  - avoiding authentication due to some error or design weakness, e.g. buffer overflow, getting a fake IP address validated.
  - getting access as trusted from a "neighbour" (`.rhosts` file and `rlogin` command) or as a `guest` from an open door or by a known / standard means (installation password).
- In addition to the above, confidentiality can be broken by
  - incorrect address in a message (due to a technical error or misspelling)
  - exposed data because of temporary decryption on intermediate stations (switches, routers, bridges, intermediate servers) or in working memory areas of programs that handle the message.

- traffic analysis: the existence of a message or the frequency of communication already reveals something, the routing structure may also be revealed, or the names used in a LAN.

- Message integrity may be broken by
modifying a message or part thereof, replacing a message with another, using an old message ('replay'), changing the sender's name, redirecting, deleting a message. There can also be (unintentional) noise, which is controlled quite well by error-correcting codes.
- Code integrity:
Digital services, especially in the era of cloud computing, abound with situations where code is downloaded or used directly over the network. The user is not usually familiar with the code, does not always even realize that this is happening, and often cannot influence the download.
- Denial of Service (DoS):
Networks usually have redundancy in routing and services as well, but failure of critical paths crashes the network. This can happen because of a "flood": an attacker sends a large quantity of messages to someone, maybe authentic-looking ones that get into processing or just rubbish that fills the channel, possibly still duplicated as acknowledgments or error messages. A different type of blocking situation arises when routing becomes cluttered due to incorrect or attacker-modified "road signs".
    - Distributed DoS (DDoS):
    All kinds of "regular" DoSs get very high power when an attacker can directly or through intermediaries use a large number (e.g. thousands, but sometimes millions) of DoS agents against a victim. Such DoS agents, often called bot agents, can be poorly maintained machines on which, for example, a worm may have installed an attack program.

18. "Full" list of threats and attacks

What you saw above in several sections was an introduction from different points of view to attacks against infosec. If you want to see a large, detailed listing of currently known attack techniques you may consult MITRE's ATT&CK at https://attack.mitre.org/. It "is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations." It provides various ways to view the adversarial details against organizations and mobile devices, both divided also on the basis of the computing platform (operating systems or cloud systems). A separate listing describes attacks against Industrial Control Systems.

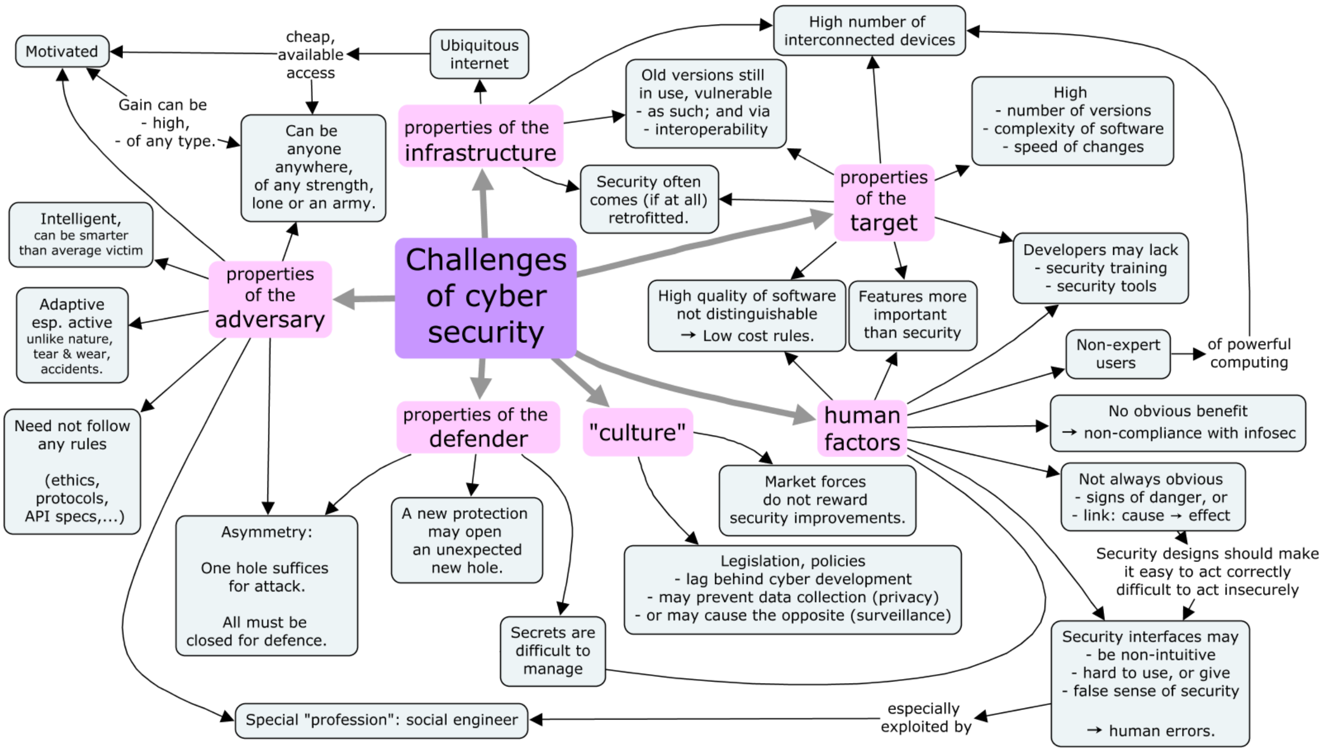
The listings can be useful for checking details while you are learning but you will benefit more when you are already working in this field. It is, however, useful to see how ATT&CK organizes the attackers' workflow. The number in parentheses shows how many techniques are listed in the enterprise case. Note that not all stages are needed for a successful attack.

- Reconnaissance (10): Gather information for use in planning future operations.
- Resource Development (8): Obtain suitable tools and "launch pads", like software, domain names or social media accounts.
- Initial Access (10): Get something more than the policy allows, at least trust.
- Execution (14): Start a process (cf. taxonomy in section **Error! Reference source not found.**).
- Persistence (20): Make it possible to keep the access.
- Privilege Escalation (14): Improve your situation at the target information system.
- Defense Evasion (43): Try to hide (note, that the list is a lot longer than the others)
- Credential Access (17): Try to obtain access similar to what legitimate users have.
- Discovery (32): Learn more about the internal environment at the target,
- Lateral Movement (9): get to more interesting places in there,
- Collection (17): and gather valuable data.
- Command and Control (18): Establish stronger presence than mere persistence.
- Exfiltration (9): Get the stolen data out safely.
- Impact (14): Alter the target system somehow, possibly trying to hide the damage.

19. Why is cyber security difficult?

This section bears a title similar to section 1.8 in van Oorshcot's book (see section 5). That book does not use the phrase cyber security at all. Instead, the section, as the book itself, is titled *computer* security, which as a term can be considered a predecessor of *information* security (and a subconcept of it). Infosec on the other hand has evolved into cyber security, as explained in section 10.

Anyway, van Oorshcot's book is excellent and fully applicable to cyber security, and its section 1.8 starts with a good reminder: "Many of today's fundamental problems in computer security remain from decades ago, despite huge changes in computing hardware, software, applications and environments." The section gives a non-exhaustive list of 20 difficulties of computer security. The following concept map reorganizes them and some others. This is also a high-level conclusion of several preceding sections that deal with threats.



20. Overview of security mechanisms

The means to promote information security can be classified to

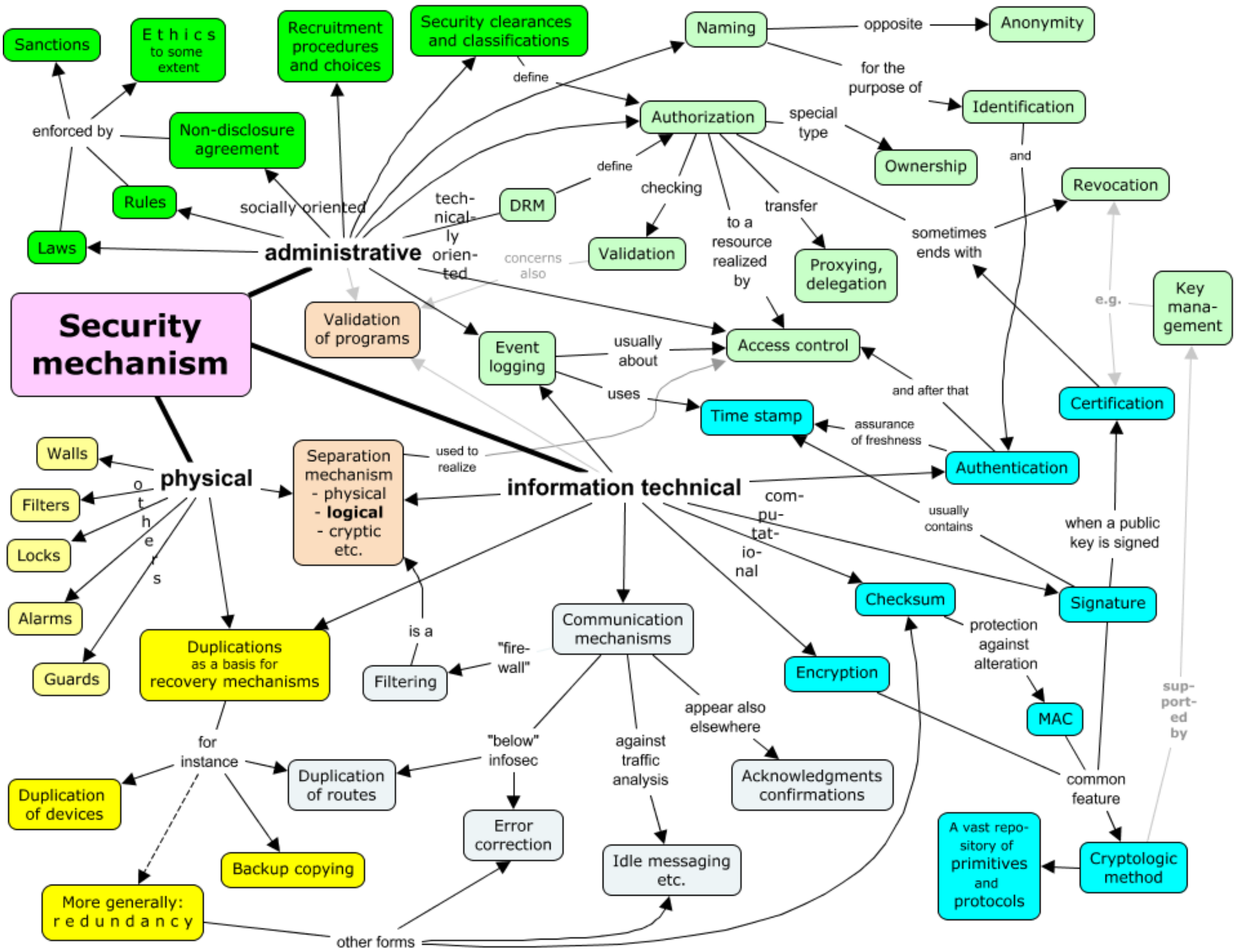
- Physical: includes hardware-specific means. The main means are various forms of separation and detection, as well as duplication, i.e. copies and fallback systems.
- Information technological or logical: realized with some level of programming. This is the most multifaceted category – as befits securing information.
- Administrative: regulatory and maintenance activities. This category has the most difficult challenges because it is closest to people.

A number of security mechanisms have been presented in the concept map below. They are related in many ways, but only a few connections are marked. Here is a brief explanation of the main logical mechanisms, i.e., those in the diagram that are directly linked to the term "information technological."

1. Encryption can be used to achieve confidentiality: the information will only be visible to those entitled to it. At the same time, integrity can usually be achieved quite well. In telecommunications, you have to decide whether to encrypt end to end, link by link, or something in the middle of these.
2. A checksum can be used to check for integrity: if the checksum is correct and *not tampered with*, then the information is likely to be in its original form. The probability of this gets higher the more

complicated the checksum is. A *cryptographic* hash is needed if one wants to be sure. With a shared secret (a key) one can compute a MAC, message authentication code (often from a hash function), that prevents tampering of the check value itself to go undetected. A hash code can also be used to check software and other files for alteration by malware. In addition to hash codes, other integrity mechanisms are used in data communications (like CRC), but their purpose is not to protect against intentional modifications.

3. Signature binds data to the signer. The signing entity can also be a machine. Besides tying together the signatory and the data, a digital signature can support integrity of the data.
4. Authentication: verification of who or what an entity is – after or at the time identification. The entity can be a human or a system, e.g. a server machine.
5. Separation mechanisms implement access control, which is administrative, rights-based reasoning that restricts access to certain resources only for certain entities. There are physical separations, but mostly they are programmatic.
6. Event logging means appending each new event to a log, which can be used for instance as a basis of intrusion detection. Logs are sometimes called audit logs or audit trails, because checking them can also happen by auditors, sometimes external to an organization. Security audit is generally an inspection of whether all mechanisms are set up properly and according to a security policy.
7. The validation of programs includes both administrative and software-based means, like checking code signatures and running anti-virus, to ensure that the programs in a system are good and at least not harmful. Software license management belongs here, or it can be thought of as part of DRM, digital rights management.
8. Communication mechanisms: error correcting codes are a traditional method. Traffic analysis can be thwarted by occasional "idle traffic" or making data packets of equal size etc.
9. Recovery mechanisms: redundancy as the basic method, where data is backed up and alternative services and routes built. (Also "exotic" methods, like cryptographic secret sharing).



The administrative and physical mechanisms are quite obvious. Here is an explanation of some other points on the map.

• Timestamp indicates when the related data were created, at the latest.
• Acknowledgment is a notification of receiving data and confirmation of receiving service. These are usually just parts of other mechanisms.
• Anonymity may not seem to be a mechanism, as special methods are needed to implement it in various contexts. The starting point for such methods, however, is to allow activity without name, and this makes anonymity itself a mechanism.
• Ownership is similarly not so mechanism-like as anonymity. The ownership of data by users and processes and the fact that this is taken into account, for example in access control, is a security mechanism,
• and the same goes for authorization, which is about granting rights.

21. Guidelines for private use of IT

Security of personal or home computing can be promoted a little more simply than following all the above mechanisms. It is likely that your bank or insurance company offers good security guidelines on their websites. A minimal set instructed in 2004 by the Finnish communications authority was just to take care of

- software updates,
- anti-virus and
- firewall.

The first two of these fall into the box “validation of programs” in the concept map of mechanisms (sec. 20), but anti-virus also shares features of filtering and so do firewalls. As a logical separation mechanism, filtering can also do other kind of pruning, discarding for instance spam and adult content.

An ordinary user of a computer, also of the handheld one, the smartphone, does not read instructions and could not care much less about the mechanisms on his or her devices. And even if one cares about one’s own security and the instructions are read, they may remain without effect. This can easily happen to any of the following measures that have been condensed from a newer (2019) version of instructions of the same Finnish authority, nowadays called Traficom (the source page changed again, 2024):

- Whenever you can, protect your account with multifactor authentication.
- Ensure that your devices have appropriate security already when you take them into use.
- Check before regretting, even for a second and third time.
- Install updates automatically.
- Make sure that all of your backup copy is actually saved and that you can recover data from it.

These instructions give a fine-tuning to authentication, program validation and backup procedures in the concept map of mechanisms (sec. 20), and introduce two measures that are not shown in the map: (1) Secure initialization of devices and services, especially of the IoT type, actually means almost everything in the map, but as things should be prepared by the manufacturer, it is again about validation. Proper security audit is too demanding for others than professionals. (2) Checking before regretting, on the other hand, is mainly about using common sense, at least not getting cheated by offers that are too good to be true.

The last remark above is also the last item in the *General online shopping checklist*, that fills page 18 of the document Fraud in cross-border e-commerce, published by the European Consumer Centres Network in 2021.

22. Guidelines for developers

As mentioned in the previous section, the main responsibility of enabling secure IoT should be on manufacturers, developers and service providers. These three groups are the industry audience to whom the Australian government has designed its Code of practice for securing the IoT for consumers, (2020). It has 13 points that are easy to understand and justify, but the fact that they must be reminded also tells that they have been sources of vulnerabilities. (*A few explanations are added to the listing below in italics.*)

1. No duplicated default or weak passwords, *includes secure credential management.*
2. Implement a vulnerability disclosure policy, *i.e. manage reporting and actions on reports.*
3. Keep software securely updated.
4. Securely store credentials.
5. Ensure that personal data is protected.
6. Minimise exposed attack surfaces.
7. Ensure communication security.
8. Ensure software integrity, *i.e. against an unauthorized change.*
9. Make systems resilient to outages.
10. Monitor system telemetry data, *that is, if it is collected, use it for anomaly detection.*
11. Make it easy for consumers to delete personal data.
12. Make installation and maintenance of devices easy.
13. Validate input data.

The last item in the above list corresponds to the first title in the OWASP Secure coding practices quick reference guide v2.0. While the above list only has short explanations in the document, the OWASP guide has altogether 213 points to check under 14 titles. The purpose of the two lists is of course nearly the same, but the vantage point is different. Most of the items in the above list relate to security needs of the application, albeit quite detailed ones, like credentials (#4) and personal data deletion (#11). Item #6 in the above list (attack surface) is a very general principle, and it is served by several of the OWASP list, which is given below. The number in parentheses shows the length of the checklist for each item, and numbers with # are main references to the above list. They do not mean inclusion in either direction, and especially #6 could be related to more items than the three (namely 5, 8 and 11) that are now marked.

1. Input Validation (16)	relates to #13
2. Output Encoding (6)	
3. Authentication and Password Management (35),	relates to #1, #4
4. Session Management (19)	
5. Access Control (24)	relates to #1, #6
6. Cryptographic Practices (6)	
7. Error Handling and Logging (24)	relates to #10
8. Data Protection (12)	relates to #2, #5, #6, #11
9. Communication Security (8)	relates to #7
10. System Configuration (16)	relates to #3, #4, #8
11. Database Security (13)	relates to #6
12. File Management (14)	
13. Memory Management (9)	
14. General Coding Practices (11)	relates to #3

It is worth noting that OWASP is *Open Web Application Security Project*, but the secure coding principles are good for non-web applications as well. From the IoT listing only number #9, resilience, is not served by the OWASP checklists. It is about providing availability, and it is slightly surprising that the OWASP checklists do not have much to say about this. Of course, redundant servers and secure restoration of service are somewhat at a different level than coding. But outages can happen also by attackers. Denial-of-service is mentioned within item 3 of the OWASP list in relation to authentication. Note that item 3 has by far the longest checklist.

23. System security design principles

What was said in section 22 concerned secure IoT and Web development, but those were only the target contexts of the referenced documents. Their applicability is wider, and on a basic course like this they are valuable in opening the eyes to all sorts of details that building secure systems requires.

This section contains a large chart of even more widely applicable principles. Before that several listings are presented to lay ground for the chart and especially to choose the subset of most important principles (page 35). You may skip the sources and start from the subsection *The map* on page 36.

Knowledge unit "Cybersecurity Principles"

The significance of design principles was already reasoned above. The general goal of learning principles is to help students "create systems that are worthy of being trusted". This is said on page 8 of a 2020 document called Knowledge Units[31] That page contains the foundational knowledge unit "Cybersecurity Principles". The general goal is not so surprising, but the interesting part comes in the following two details of learning outcomes: The student should understand

a) how each principle enables the development of security mechanisms that can implement desired security policies.
b) the interaction between security and system usability and the importance for minimizing the effects of security mechanisms.

The remaining outcomes are ordinary expressions of learning goals, but in these two outcomes you actually see two *principles* hidden: (a) Policy-driven design and (b) usability. The latter is also repeated as one of the 15 "Cybersecurity Principles", but the first one is not. Only one item among those 15 does not appear in the lists that we present below. It is encapsulation. Its meaning is not given, but usually it means wrapping data within some protections, most commonly encryption. As such we do not consider that a principle but just a mechanism.

NIST principles

NIST Special Publication 800-160, Systems Security Engineering[32] from 2016/2018 has been rewritten to become Engineering Trustworthy Secure Systems in 2022. It is still a fairly abstract and verbose document and not only about information security. The content is, however, relevant to infosec professionals, too. For the purpose of this section the 32 principles from Appendix F of the *earlier version* are listed here.

Security Architecture and Design: Clear Abstractions Hierarchical Trust Least Common Mechanism * Inverse Modification Threshold Modularity and Layering Hierarchical Protection * Partially Ordered Dependencies Minimized Security Elements Efficiently Mediated Access Least Privilege Minimized Sharing * Predicate Permission	Reduced Complexity * Self-Reliant Trustworthiness Secure Evolvability * Secure Distributed Composition Trusted Components * Trusted Communication Channels *Security Capability and Intrinsic Behaviors:* Continuous Protection Secure Failure and Recovery * Secure Metadata Management Economic Security	* Self-Analysis * Performance Security Accountability and Traceability Human Factored Security Secure Defaults Acceptable Security *Life Cycle Security:* Repeatable and Documented Procedures Secure System Modification * Procedural Rigor Sufficient Documentation

These are not intended for you to memorize. The intention is just to show that even if the subsequent map is large, there are many issues not included in it[33]. And still, this listing does not include for

[31] It lists requirements of CAE-CDE programs in the USA: Centers of Academic Excellence in Cyber Defense Education.
[32] The original 2016/2018-version of this publication appears in one of the Harpo exercises of this course.
[33] The number 32 is small in comparison to security checklists, like the 214 items in the OWASP list (cf. sec. 21). The German handbook (cf. sec. 3) is naturally beyond check-list usage with its 855 pages, but it is still much better than the "internet" where you can continue searches without getting a holistic picture.

instance Redundancy[34], which is in the listing of the new version of SP 800-160. It also has Diversity (to counter common mode failures related to redundancy), but as the new listing has quite a different focus[35] the old one is used here.

An * is marked at such "smaller" principles here that are not included in the map. The others are there, albeit many with different words, and possibly with different emphasis. Note that there are 3 subtitles in this listing.

Principles from Cybersecurity Curricula 2017

Here is one example of a list of security principles that has a little more structure than above. This is from Cybersecurity Curricula 2017 (see section 4). Reference to most of these principles happens in the knowledge areas of Software Security and System Security. Component security borrows some of these and presents some of its special ones, but they will be shown only in the subsequent map.

Among the borrowed ones Component security adds a recommendation to treat *security as an integral part of system design*. This is certainly a very good and generic principle, and it will appear as one of the high-level principles on the map.

Restrictiveness	*Simplicity*	*Methodology*
Least privilege	Economy of mechanism*	Open design
Fail-safe defaults	Minimize common mechanism*	Layering
Complete mediation*	Least astonishment	Abstraction
Separation		Modularity
Minimize trust		Complete linkage
		Design for iteration

All of these will appear on the map but those marked with an * will have a (very) different term. Complete mediation is one of them and there is a separate discussion on that below. Note that complete linkage is something different: it means linkage between design and implementation on one hand and security specifications on the other hand.

Smith's principles

In Rick E. Smith's book *Elementary information security*, the list of security design principles contains only 8 items. See his page[36], also to appreciate that many of the principles in the other lists did not appear for the first time in them. An asterisk marks those that appear in the map under a different term.

Continuous Improvement*	Chain of Control*
Least Privilege	Deny by Default
Defense in Depth	Transitive Trust*
Open Design	Separation of Duty

Are these more profound than the 32 in the NIST list? No, as you will soon see, but you may also think otherwise. This document presents just one map based on several listings. You can create your own set of principles for your work or study. This could be done by abstracting ideas from lists and organizing them in one or more concept maps that reflect your own way of understanding and remembering. But apparently it is not wise to work against any of the established principles without a good reason – perhaps dictated by a contradicting principle.

[34] The map does not include redundancy, either. See the explanation (with "backups") at the very end of this section.
[35] The overlap of the two lists is only about 40%.
[36] Or the original article mentioned therein: Smith, R.E. "A Contemporary Look at Saltzer and Schroeder's 1975 Design Principles," 2012. The title shows the famous origin of security design principles.

Van Oorschot's principles

In the beginning chapter of his book (cf. sec. 4) Paul van Oorschot gives an excellent account on difficulties of information security (cf. sec. 19). Before the difficulties van Oorschot presents 20 principles, and while explaining them he interrelates them and refers to further principles. Here is his (main) list:

Simplicity and necessity	Small trusted bases	Data-type verification
Safe defaults	Time-tested tools	Remnant removal
Open design	Least surprise	Trust-anchor justification
Complete mediation*	User buy-in*	Independent confirmation*
Isolated compartments	Sufficient work-factor*	Request-response integrity*
Least privilege	Defense in depth	Reluctant allocation
Modular design	Evidence production	

This listing does not have a subdivision, but the more important principles are in the beginning and the more detailed ones in the end. Again, we mark an * if we use an essentially different wording in the map.

What are the most important principles among the many?

The Cybersecurity Curricula (sec. 4) gave six *crosscutting principles*. Out of them Confidentiality, Integrity and Availability are too deeply *defining* security as not to be considered even principles in our context. So, we could say here that the curricula document contributes three crosscutting principles

- Risk
- Adversarial Thinking
- Systems Thinking

The NIST Systems Security Engineering (version 2016/2018) gives three *overarching principles:*

- Reference monitor concept
- Isolation
- Defence-in-depth

In addition to the 20 principles van Oorschot presents two *higher level principles*, which deal with the life cycle of the system:

- Security by design, or in other words: build security in. The result is better than retrofitting security. The same principle appeared as *security as an integral part of system design* in the Cybersecurity Curricula, and this is how it appears in the map. This principle is sometimes named DevSecOps (development, security, and operations[37]).
- Design for evolution: updates will be needed, and it is good to have a design that can adjust to them. This is the same as *Design for iteration* in the Cybersecurity Curricula, which is somewhat more general than *Continuous improvement* in Smith's list. The map uses *iteration*. It may not be the most intuitive term, but it covers both aspects: improvement as an intrinsic aim, and evolution which can also come from outside changes.

Additionally van Oorschot presents a maxim "trust but verify" borrowed from a proverb that sounds better in the original form in Russian "*Do-veryay, no pro-veryay*". He relates this to four of his 20 principles, and the most general of them is complete mediation. And so, the maxim is actually "verify before trusting". Choosing this as a principle will give us something easy to memorize and it will replace both complete mediation and the reference monitor concept.

Not using "complete mediation" as a principle as such also avoids Rick Smith's criticism (2012, the same link as above), where he says that it is obsoleted in the distributed and networked environment,

[37] See also: a blog entry, and a RedHat page.

where no single mediator is making the whole decision of access control. We note – as counter-criticism and based on the three other (more recent) sources presenting complete mediation as a principle – that it is still of high importance that each access to resources gets verified. Actually, Smith's own principle Chain of control is a special case of this. So, verifying before trusting replaces also that.

The map

The most important 10 principles – as discussed but not yet counted above – are marked with orange colour in the map. There are 30 other principles, but the whole is divided into seven classes, with the intention of making things easier to digest and remember. Even the largest classes only have 7 principles.

Several of the highest-level principles are collected under the class ***Apply in your mind…***. These *mind-sets* are such that you must have adopted them into your thinking, but it is still good to be able to identify them by names. One of them consists of three points of view that require mutual balancing: risks, costs and policy. Combining this with adversarial thinking leads to the principle of making sure that the protections are stronger than the attacker. This is obvious but must be remembered especially when choosing cryptography.

Organize your work contains two high-level principles dealing with the life cycle of the system to be designed. In a sense they could be labeled as *start* and *re-start*. The risk-cost-policy node, too, can be seen from the administrative point of view.

As a class the *mind-set* is clearly at the highest level and the *administrative* class is close to it together with the *minimizing* class, where the principles are targets to minimize in the design. Actually, the class is named ***Make smaller***, which is a simplified form of "Minimize under the constraints of all other principles". One of the main principles belongs here: *Minimize the attack surface*. All other minimizations can be seen as realizations of this. Even minimal surprise reduces situations where attacks are possible. Minimizing the number of *shared facilities* is originally the well-known principle of "least common mechanism". Its motivation is that such minimization also reduces opportunities for unintended communication or interference between processes or users. The word facility is used because it is more general than mechanism and may more easily bring into mind also variables and files besides code.

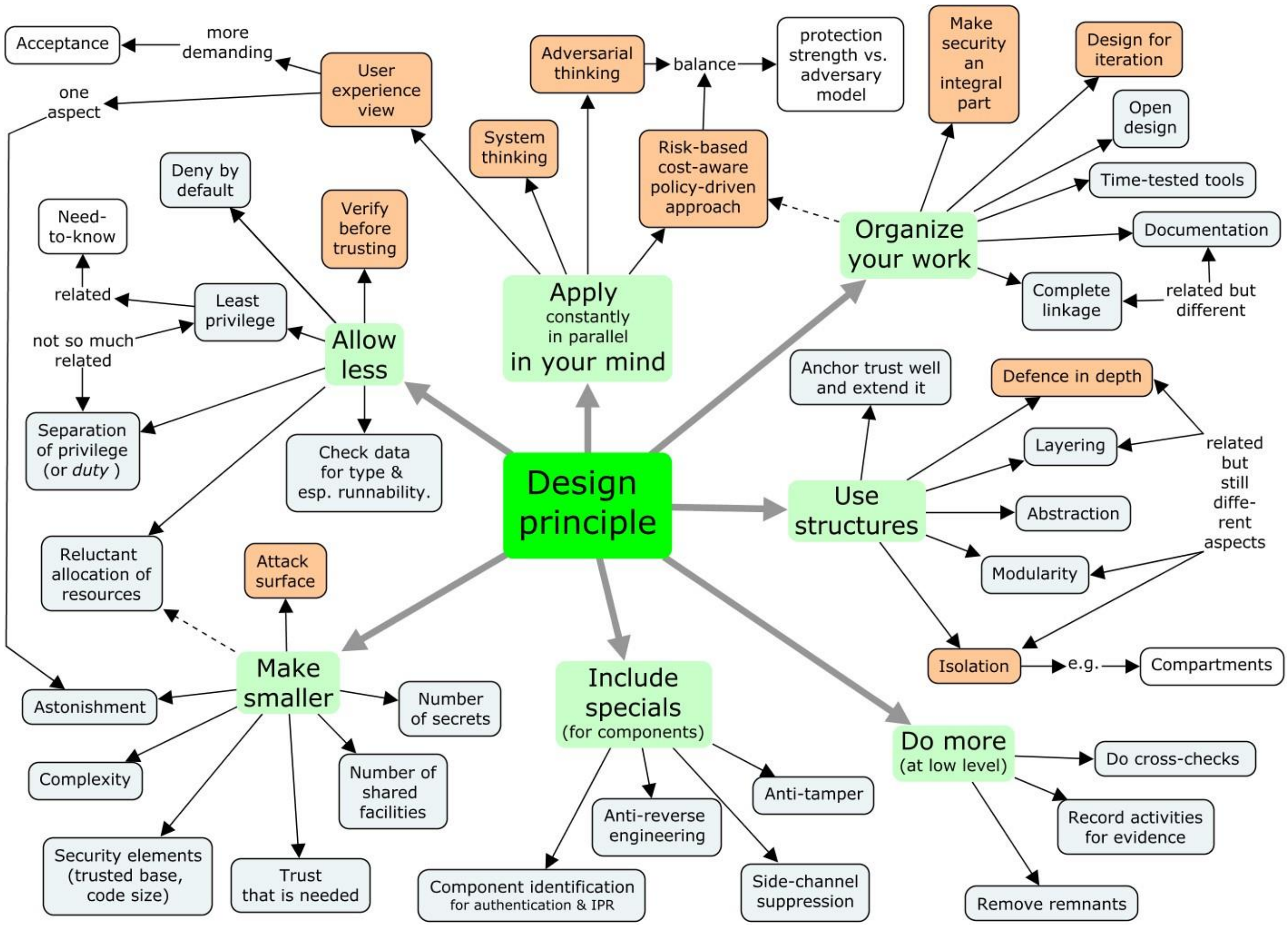
The fourth level is the "architectural" class ***Use structures***, which still has two high-level principles. They could be roughly labeled as *depth* and *width.* The layers have a little more to do with depth and the modules with isolation, but they are separate structural aspects.

The only remaining high-level principle belongs to the class ***Allow less***, which is about restrictions or throttling[38] and is mostly very different from *Make smaller*. The restrictions in *Allow less* are mainly about access to resources within the system under design, and they stem from the high-level principle *Verify before trusting*. This is the principle that has many versions as discussed above. The other principles in this class also have several phrasings. We used *privileges* in two of them to show the similarity in name but to emphasize the difference in meaning. The principle of reluctant allocation is useful in avoiding denial-of-service situations. It could be also viewed as a minimization target, but then it should be rephrased clumsily to something like "Minimize the urge to allocate resources".

There are two classes without any orange nodes: *Do more* and *Include specials*. Both classes deal with adding some properties that serve security and are quite remote from the ordinary functioning of the system. Both classes are based on one of the sources: *Include specials* is from the *Component security* knowledge unit of Cybersecurity Curricula and *Do more* stems from van Oorschot. ***Include specials*** – except *identification* – may complicate the functionality quite a lot but all items have a good motivation. Two principles at ***Do more*** are extra outside ordinary functions but essential for future ("clean up" and "make logs"). The remaining one, *Do cross-checks* is extra "inside", meaning more

[38] Note that while the structural principle of *isolation* can rarely be fully implemented it is the principle of *allowing less* that helps in filtering the interaction between compartments like network segments.

work during the ordinary functioning. It combines van Oorschot's two quite detailed principles *Independent confirmation* and *Request-response integrity*. The first one of them augments *Verify before trusting* in (less trusted) situations where there is reason to be more doubtful, and the latter reminds that one such case is when you get a response to your request. Cross-check is due to make sure the response matches the request. These matters are well understood by those who already design protocols, but a note for others can be useful.



The variety of principles in the four sources shows that creating abstractions leads to different emphases. Smith has a strong educational motivation and so have Cyber Curricula and van Oorschot – and the current document, too. If not the lists and maps in this document, then possibly the layered design of Common criteria or IT-Grundschutz can give a starting point for organizing your own tool-box of principles. A further example of layers is in the a U.S.-defence-oriented document Cyber-security Maturity Model Certification (CMMC Model overview, v2.0, 2021) where page 4 gives a graph showing how the practices accumulate when maturity increases. The total number of practices is over 170.

As a conclusion it is perhaps good to remind that this section deals with creating secure systems and not with secure practices of using them. The matters are also more abstract than security mechanisms or good security habits, like

Updates	Good passwords
Backups	"Hygienic" browsing
Antivirus	Awareness training
Firewalls	Secure initialization
Multifactor authentication	Encryption

or even having a policy and enforcing it. Many of these are fairly direct consequences of the principles though. And of course, the designer and implementer of a system must have their production environment and work habits secured.

24. Example: Trusting a program can be complicated

Assume your laptop is running Microsoft Windows and you are editing some files that you need to copy over the internet to a computer at the university. How can you make such file transfers securely?

What you are reading now is an exercise that presents the solution in the third paragraph and then spends 2,5 pages asking questions – and hardly providing answers. The intention is to familiarize you with all sorts of issues that are related to trusting a program. There are no other exercises of this kind on the course, but this example may lead you to think of some other situation as a possible topic for your Essays exercise.

Start by finding the correct site to download WinSCP. Is it winscp.net?

Download the program file and start investigating it. You can do this even if you don't use Windows. The file is less than 11 megabytes and doesn't do any harm.

There used to be three checksums given on the download page (nowadays only the two latter) like this:
MD5: 1dd9535e9f9ee30679b14a6fc16c87b9
SHA-1: b93eb228de3293f74b54974296c72eb6e4fab046
SHA-256: 79de2d5cba143cba220ecf6c76d9e07407243e554ba524a78365ccd881b80214

What coding is this, with abcdef?

You may guess that SHA-256 has 256 bits. How many have SHA-1 and MD5?

The site indeed calls these checksums, but their calculation is much more complicated than mere summation. What is a more precise term for these "fingerprints"?

How can you calculate the fingerprint of the downloaded file? In Windows the keyword can be FCIV. Find the fingerprint of the file and compare it with the one given.

What does a successful comparison prove?

If the file were not ok, why would the page give a correct fingerprint?

So, a matching fingerprint only proves that the file is the same as on the page.

Could you find support elsewhere for the fingerprint, and what would it help?

Now look more closely into the file, not into its contents but its metadata. In MS-Windows go to >Properties >Digital signatures. Or find a certificate decoder in the web, like https://www.sslshopper.com/certificate-decoder.html to decode this


```
-----BEGIN CERTIFICATE-----
MIIFeTCCBGGgAwIBAgIQAjJGbclbQOydIdkymr/NXTANBgkqhkiG9w0BAQsFADBs
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSswKQYDVQQDEyJEaWdpQ2VydCBFViBDb2RlIFNpZ25p
bmcgQ0EgKFNIQTIpMB4XDTIwMDIxMDAwMDAwMFoXDTIzMDIxNzEyMDAwMFowgZcx
EzARBgsrBgEEAYI3PAIBAxMCQ1oxHTAbBgNVBA8MFFByaXZhdGUgT3JnYW5pemF0
aW9uMREwDwYDVQQFEwg4NzMzMTUxOTELMAkGA1UEBhMCQ1oxDzANBgNVBAcTBlBy
YWd1ZTEXMBUGA1UEChMOTWFydGluIFByaWtyeWwxFzAVBgNVBAMTDk1hcnRpbiBQ
cmlrcnlsMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArjgofJe/Yqm1
CDbY+bsHzq51KVtLZiHMP4vf6u9wNfdsZ93Zr4Z7BS85U/cac0ab7v+zfQSb9SAp
v8K9heu/s02D/5u1St2NFrW9uztFfZeE9L5Vy7IHhqFAdk1ZDTgM5zkuu1DsK/g8
1vjYHFulKdCM/4DVO8dXNQiQfU1o1mMMTIh+ygudKvZtRwQdpJUTkKu26Y4HSDYw
GTtB/BwzTX1T3TCCCOmGSXaqBNrXgQgfgjAs9Zfa5GCoaQOUo8SlH8emNiVuHzJ4
yDixNeiTOLQvJNfLGHAPZShXnrf2BSbYrpBxKCDmucuhMQfgI3VUboR1ohPzZUQ6
eRYSLltk7wIDAQABo4IB6TCCAeUwHwYDVR0jBBgwFoAUj+h+8G0yagAFI8dwl2o6
kP9r6tQwHQYDVR0OBBYEFB6VJN3DGK95aOAEc/7thaBFoj6lMCYGA1UdEQQfMB2g
GwYIKwYBBQUHCAOgDzANDAtDWi04NzMzMTUxOTAOBgNVHQ8BAf8EBAMCB4AwEwYD
VR0lBAwwCgYIKwYBBQUHAwMwewYDVR0fBHQwcjA3oDWgM4YxaHR0cDovL2NybDMu
ZGlnaWNlcnQuY29tL0VWQ29kZVNpZ25pbmdTSEEyLWcxLmNybDA3oDWgM4YxaHR0
cDovL2NybDQuZGlnaWNlcnQuY29tL0VWQ29kZVNpZ25pbmdTSEEyLWcxLmNybDBL
BgNVHSAERDBCMDcGCWCGSAGG/WwDAjAqMCgGCCsGAQUFBwIBFhxodHRwczovL3d3
dy5kaWdpY2VydC5jb20vQ1BTMAcGBWeBDAEDMH4GCCsGAQUFBwEBBHIwcDAkBggr
BgEFBQcwAYYYaHR0cDovL29jc3AuZGlnaWNlcnQuY29tMEgGCCsGAQUFBzAChjxo
dHRwOi8vY2FjZXJ0cy5kaWdpY2VydC5jb20vRGlnaUNlcnRFVkNvZGVTaWduaW5n
Q0EtU0hBMi5jcnQwDAYDVR0TAQH/BAIwADANBgkqhkiG9w0BAQsFAAOCAQEAkACN
0vCCRLPCJfpMyMIXIIYysXez4+ODX4xvUQfWZIrCqzvs+d15rK+KrD2YYyQUdf+i
sBkVGX0jtTT3jP6KQXRZFPo+esnHKv2NPGh3feDRP7w8iSc//PHMD9uuUuGndvIl
YaRgqVNlZ4Ne11PKXL8szR1b/6MbkpdK1cbx/zTnG5UY4/oeAIz1CBUqFh6OIVLP
O3KrJO5afSjnP0KH5Vq4ha9JzwIWMKgJY22CI8OFgRLMpajNbriLdoXCLWkiM4tv
oM/jzrqIOSATO6+cXRF1x0abIGl3mVCL1XfIqdYmrtuHx1m5FOG7uhBdE1sWer7y
EGdvHzDGZOCZFPW7Uw==
-----END CERTIFICATE-----
```


What sort of decoding happened? In other words what coding is it that starts like this `MIIFeTCCBGGgAwIBAgIQAjJGbclbQOydIdkymr/NXTANBgkqhkiG9w0BAQsFADBs` ?

You will see also a person's name. Whose? Is he the attacker who modified the program and digitally signed it? Who gave him the certificate for the code signing key?

(Go to >Details and >View Certificate.) What evidence is given to support the validity of the signature?

(Go to >Issuer Statement.) Which CP and CPS are valid for the certificate in question? See how these documents look like and what they tell us about. (CP=Certificate Policy, CPS=Certification Practices Statement)

So, if you trust the certificate and the signature on the program, what evidence do you have that the signer has authority to sign the code?

You apparently get to evaluate the download site. There you get to a similar chain of investigation that first leads to a certificate authority. Then it leads to evaluating the trustworthiness of the site itself. Did they write the code or are they just distributing it? How would you trust that the site is not spreading malware?

Can you find support through some kinds of reputation checks – on the publisher and on the program itself? Does the publisher provide evidence that the code was made securely?

Can you see the software license somewhere before starting to install the program? Yes, but can you find the page, without this direct link? When you see it is a GNU GPL, are you relieved that you don't need to read more? Did you notice that there are some picture elements that are not under the GPL, and there are a few extra paragraphs in the license concerning them? And at the end there is also a note on privacy.

You need not install the program, but what is your conclusion, aided with your antimalware software, whether it is safe to install it?

Because the WinSCP file is digitally signed, the operating system automatically attempts to verify the signature during installation. Most likely it accepts the signature because it regards the certificate as valid. So, you may think you did the above pondering in vain. Not really, the operating system is no wiser than you, but it just has the CA certificate preinstalled. And it probably still asked you to accept the installation. And furthermore, it probably asked you to provide admin credentials before proceeding.

This is just one example – and quite high in the hierarchy between hardware and humans – of the security tasks of an operating system. This is essential in modern networked computing, but the more profound security services are lower in the hierarchy. Some of them are almost as old as operating systems themselves, for instance keeping user's code from poking or even peeking into the OS code. Some of the low-level controls are newer, like ASLR, about which you may find out yourself now, or when facing the MCQ on the Maso pages.

Let us get back to WinSCP. If you install and start using it, some important questions still remain.

The program does file transfer between hosts over the internet. How do you know what protection is needed and whether that protection is given? Firstly, how is the destination host authenticated? Secondly how is your transmission encrypted?

Knowing that WinSCP, or some similar program on your own machine, handles the security task properly is largely based on trust and the observation that many others have trusted without any problems. But crowds are not always right. And if you are one of the early adopters, you might need to do your own testing by seeing what sort of traffic the program generates. A packet sniffer will help in this, but probably you would need several other tools, also such that are designed to do cryptanalysis or other sorts of attacks.

New programs to be installed for the tests would start this story over, and go beyond the course objectives. It is obvious though that you must not see any cleartext in the traffic, and you must not see repeating patterns when you modify the input, i.e. the file to transport. And of course, you should see an SSH handshake. SSH, the Secure SHell, originally a Finnish design from 1995 by Tatu Ylönen, will appear as a basis of WinSCP in what follows.

The main reason to trust the operation of WinSCP is that it provides support for SFTP and SCP and not just FTP. What does this mean? In other words, what are these protocols?

It is apparent that the authors and publishers of WinSCP are honest and try to do their work well. Although WinSCP can be considered a secure program, it has also had bugs, such as an integer overflow.

It is described on https://www.cvedetails.com/cve/CVE-2013-4852/in a single sentence, which can be structured as follows:

Integer overflow
in PuTTY 0.62 and earlier,
WinSCP before 5.1.6,
and other products that use PuTTY
allows remote SSH servers to
cause a denial of service (crash) and
possibly execute arbitrary code in certain applications that use PuTTY
via a negative size value
in an RSA key signature
during the SSH handshake,
which triggers a heap-based buffer overflow.

Here PuTTY is the SSH client program on which WinSCP is based. That is, WinSCP talks to the SSH server at the remote machine through PuTTY.

What stage of using WinSCP does the CVE-2013-4852 vulnerability affect?

a) Verification of the installation file signature discussed in the beginning of this section,
b) server program authentication to the client program,
c) authentication of the client user to the server or
d) the stage at which the program checks the integrity of the file transferred between machines?

How did you reach this conclusion?

25. Trust, assurance and levels of security

You surely understood what trust means while reading section 24. Perhaps you thought that trust is not a cybersecurity concept but something psychological. Trust certainly appears mostly outside information security, but if the broad term *dependability* (including safety) from section 6 is taken into use, you may realize that trust is what most everything in our daily live depends on. And trust is not only what individuals feel, but it is also "felt" collectively. Trust can be said to be a belief in the future, and even willingness to accept uncertainty.

There is a vast literature on factors that affect people's trust in various on-line situations like e-commerce. A fairly early article on this is by Wang and Emurian (2005) and from the link you can find many new articles citing that paper, and also many earlier references.

Maps and lists of trust

The first concept map below presents some general properties of trust, while the second one delves deeper into the realm of information security. For an introduction to the transition from the general to the specific one here is a list of matters where the concept of trust ordinarily appears in computing:

- You can trust the sender of an email or another message, and then for instance open an attached document. Here it is well-known what can go wrong and how an attacker can try to make it happen. Also in all other points of this list you can investigate the role of trust by checking the adversarial viewpoint.
- Whitelist: A list of entities that are considered trustworthy and are granted access or privileges (quoted from a glossary)
- A certificate policy dictates, among other things, how well the owner of a public key must be validated before a certificate is granted. There are many quality criteria, but the following is a very common setting.
- Web sites may have ordinary public key certificates or EV certificates. EV stands for extended validation, which means that the certificate authority, CA, has done such validation towards the identity of the key owner. The "ordinary" certificates also have degrees: DV, OV and IV standing for domain-, organization- and individual-validated.

- Web browsers, and other software, have preinstalled certificates of many CAs. They act as anchors of trust to the user who will then extend his trust to certificates issued by these CAs, and consequently to the connection protected by the certified key, and in most cases also to the site that is under such protection. The last step is *not justified if* it involves trust in other qualities of the site.
- eIDAS stands for Electronic Identification and Trust Services Regulation (910/2014/EU). The eIDAS Levels of Assurance Low, Substantial, High mean various degrees of confidence in the claimed identity of a person. From the link you can check what each of these can require.
- Certificate authorities are just one example of trust services. Their service in general is to sell statements about some entity to be trusted in some matter – and being themselves trusted to say so. A certificate authority does this for a key in the matter of it belonging to a web site or person. Another type of trust service can be a bank which testifies a person's identity.
- Trusted Introducer is the support infrastructure that the European CERT community established for security and incident response teams (CSIRTs). These teams exchange very sensitive information about attacks, and only trusted teams can become CSIRTs.
- Trusted computing base, TCB, is a concept from 1980's, which means it is quite fundamental. It is that part of an information system where our security policy holds without further verifications, i.e. all accesses are legitimate. This is somewhat theoretical, though, because also TCB can contain programming or hardware errors. Actually, a TCB is what we *must* trust to proceed in development of more sophisticated features of the system.
- Trusted platform module is a more recent incarnation of a TCB by the industry association Trusted computing group. It can appear under various names, for instance Embedded security subsystem.
- Trusted Path is a term referring to a user's way to access the trusted computing base. In a way, the whole path, for instance from the keyboard through its driver to the OS kernel, then becomes part of the TCB.

AI, blockchains, computational trust

The discussion in this subsection deals with trust but it meanders through several curves, like AI, to trust levels which are related to recommendations, reputation and computational trust.

Rob Bainbridge advocates, or reminds of, AI-supported trust as an augmentation of the dimension security–usability. He writes (2018) "Trust is the fundamental part of control and can be heavily exploited using contextual and behavioural analysis to both improve user experience and increase security effectiveness."

Bruce Schneier would probably oppose to this, based on some of his earlier essays related to trust. One (Feb 2019) is on blockchain but includes trust-related reasoning beyond that. A slightly more recent commercial essay by EY (2020) gives an example of using blockchain to create trust in the Xbox ecosystem. The main topic of that essay is how differently regulators and enterprises currently see the prospects of trust in AI, that is, AI as target, not a tool for trust. Another blockchain example deals with the cold-chain for fish delivery, in an article about automating trust (2019).

Security engineers will have an interesting task in helping people trust in AI, but many issues are beyond mere security improvements[39]. Still, the talk about trusted intelligence in another EY article (2020) brings some of this hype back to the basic information security: To reduce the need of trust, attribute-based access control (ABAC) implements the principle of need-to-know using finer-grained judgments than role-based AC (RBAC). The ABAC model is not included in Exams 1 or 2, though, but you can study that in your essay.[40] You may get the "RB vs. AB" idea from the analog of a kitchen: a cook and a cleaner are different roles, but if these roles have access to the kitchen, then chili peppers can be accessed by both. ABAC would make a difference, preventing access to peppers even from the cook if the current recipe (=attribute list) does not include peppers.

Before entering the AI matters, security engineers should understand the levels of trust in the same way as they have been dealing with risks. You have encountered the problem of estimating and even

[39] like trust barometers show: bling-bling examples from 2020 and 2021.

[40] ABAC has become more common, as evidenced by its appearance in the CISSP exam content in 2018.

understanding probability if you studied risk-assessment in the NIST-Intro (cf. sec. 9)[41]. Now assume risks have been managed and due countermeasures are in place. How certain are you that things are "going to be fine"?

To approach certainty, you also need some assurance that the security mechanism function properly. You probably did not create all of them yourself and so you must trust someone else's work. It is good if the designer and manufacturer have followed the best practices and security design principles as stipulated in section 23. There are many layers of trust needed to assure you that things are like that, for instance the organizational practices in the designer's firm, the designer's education, and the standards of the educational institution.[42] Security audits usually reach only the first of these. Hence you cannot get rid of the psychosocial aspects of trust like those mentioned at the left edge in the first map below. Maybe some of these aspects can be embedded into AI someday, and hopefully AI can then be trusted.

In case you are striving for maximal available assurance, you can build on the trusted platform modules, mentioned above. And on top of them you can use the zero-trust architecture which is described in the second map below by the properties *verifying all requests* and *minimizing domains*. How to carry out this is an important topic, but it is one of those that are left out of this course, and consequently listed in section 30.

If you can partially restrict access in case you are not fully assured, you should put this in effect – to improve user experience in a system, in contrast to full restrictions. In the kitchen analog you could check whether the cooks are sober, have washed their hands, protected their hair, are possibly acting on behalf of the real cook, or if there is a fully trusted cook present etc. If something is not perfect, the most valuable or dangerous ingredients could be withheld. In a similar way – and more realistically – levels of trust can be introduced to the field of security engineering. In any case some measures of assurance are used to increase the probability that trust is not misplaced – or otherwise too high. If there are different levels of trust then different levels of access will be given.

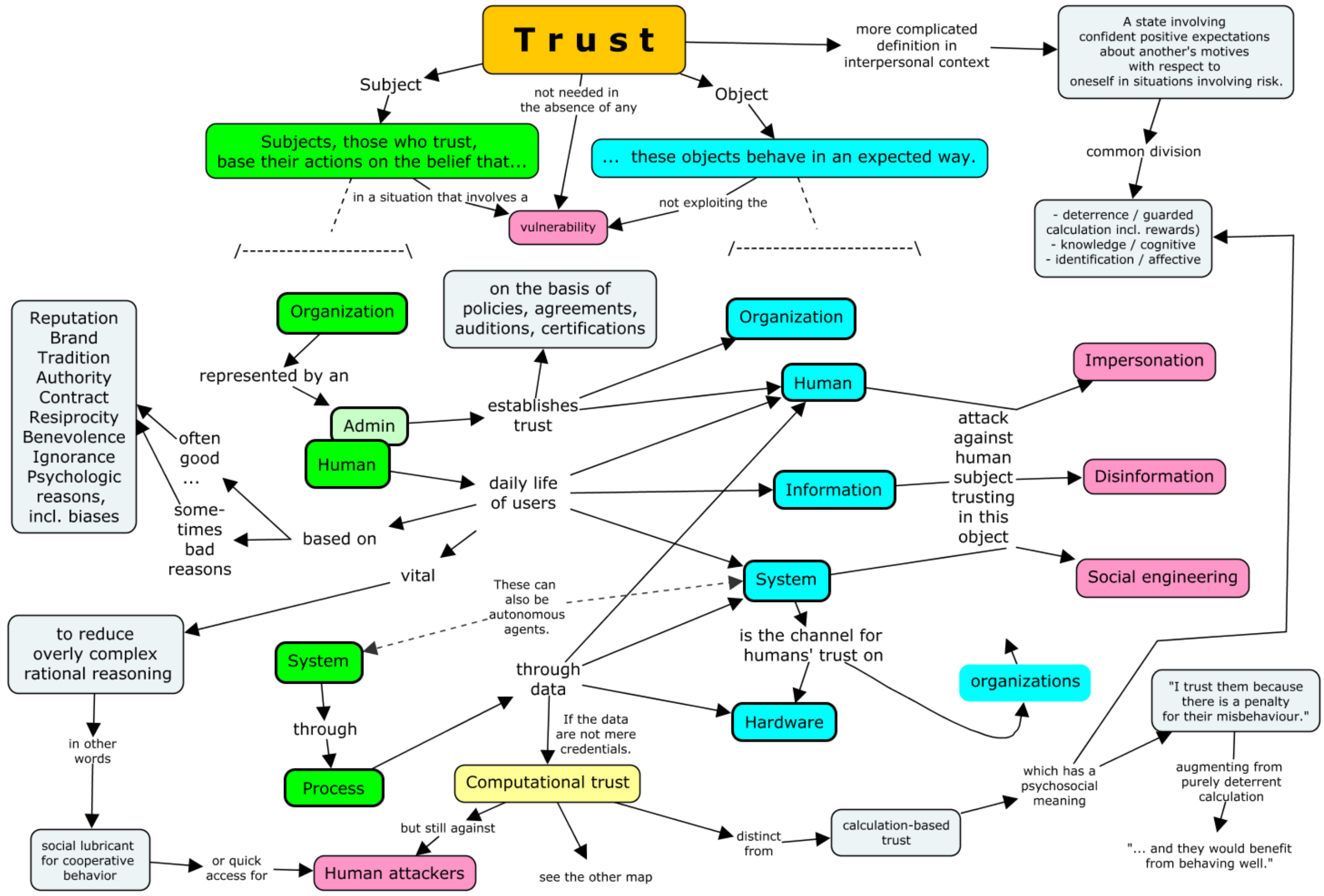
Trust that is evaluated at "run-time", especially at varying levels, is called computational trust, and the computations are part of trust management. This is a very popular research (and business) area. The survey article by Ruan and Durresi (2016) seems to be a suitable hub for you to get to the topic and fresh publications (74 as of April 2021).

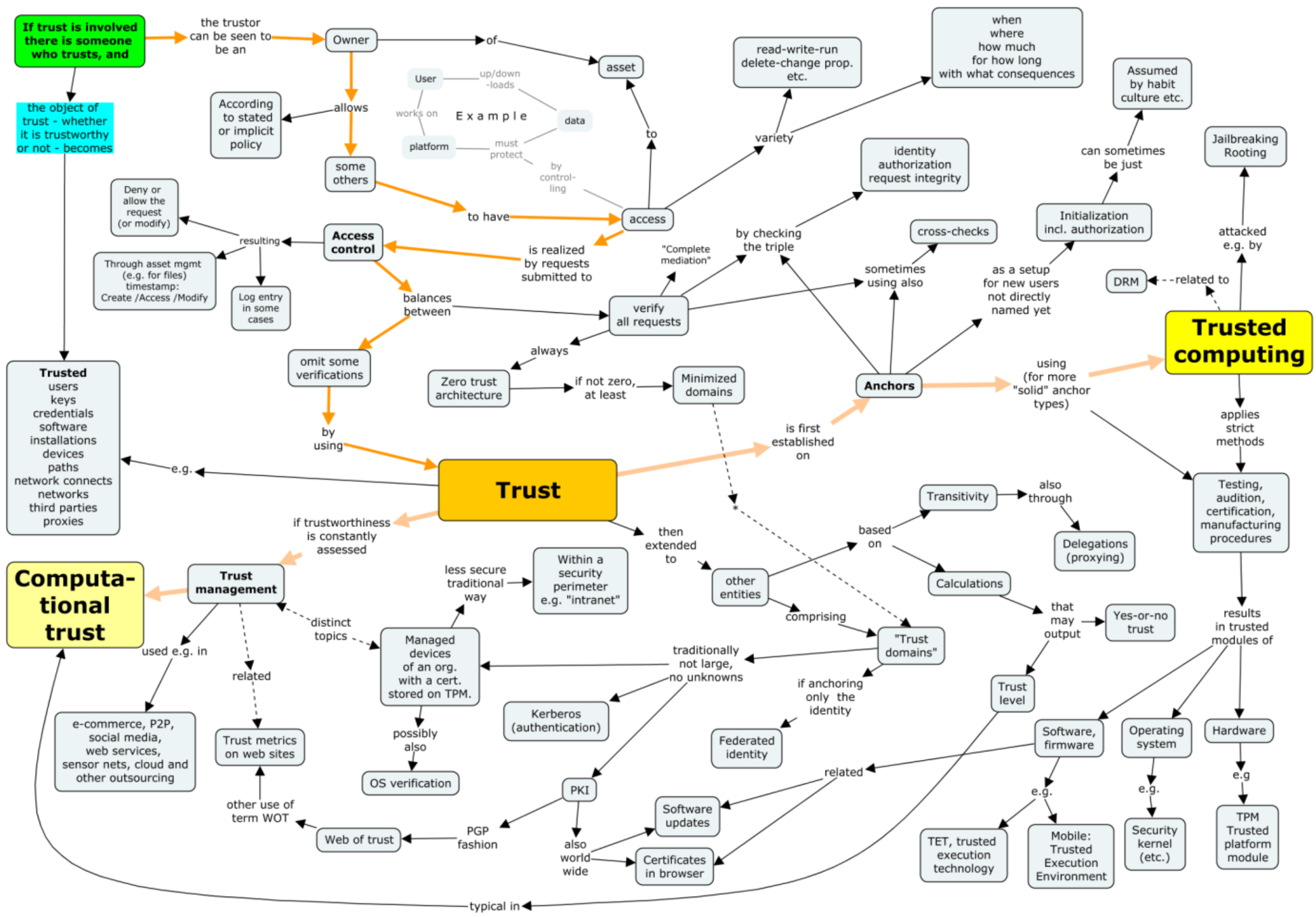
The maps

The first map mainly displays entities, subject and objects, between of which a trust relationship can be established. Notice that when a process authenticates a human user, a system or hardware, it can be said to trust these entities. But if there is nothing more involved than authentication, albeit with a lot of mathematics, this sort of trust is not computational – and neither calculational (which could be also phrased speculative).

The second map also starts with the subjects and objects at the top left corner, but it then proceeds through the data owner to data access and introduces trust in the middle as something that is needed when everything cannot be verified. No attacks are shown in this map[43], but you must always remember to think that there is something adversarial hanging below most of the nodes.

[41] … and also if you did the Harpo Survey exercise to your acquaintances.
[42] For a more technical example of many stages when booting a computer, see section 11.5.4 Code and data integrity checks in CyBOK (find the link in sec. 5 above)
[43] except two (at top right) that are usually carried out by the device owner.


Trust
more complicated definition in interpersonal context
A state involving confident positive expectations about another's motives with respect to oneself in situations involving risk.
Subject
not needed in the absence of any
Object
Subjects, those who trust, base their actions on the belief that…
… these objects behave in an expected way.
common division
in a situation that involves a
vulnerability
not exploiting the
- deterrence / guarded calculation incl. rewards)
- knowledge / cognitive
- identification / affective
Reputation
Brand
Tradition
Authority
Contract
Resiprocity
Benevolence
Ignorance
Psychologic reasons, incl. biases
Organization
represented by an
on the basis of policies, agreements, auditions, certifications
Organization
Impersonation
Admin
establishes trust
Human
Human
often good …
attack against human subject trusting in this object
some-times bad reasons
daily life of users
Information
Disinformation
based on
vital
System
Social engineering
These can also be autonomous agents.
to reduce overly complex rational reasoning
System
is the channel for humans' trust on
organizations
through data
"I trust them because there is a penalty for their misbehaviour."
through
Hardware
in other words
If the data are not mere credentials.
augmenting from purely deterrent calculation
which has a psychosocial meaning
Process
Computational trust
but still against
distinct from
calculation-based trust
social lubricant for cooperative behavior
or quick access for
Human attackers
see the other map
"… and they would benefit from behaving well."


If trust is involved there is someone who trusts, and
the trustor can be seen to be an
Owner
of
asset
read-write-run delete-change prop. etc.
when where how much for how long with what consequences
the object of trust - whether it is trustworthy or not - becomes
User
up/down -loads
works on
Example
data
According to stated or implicit policy
allows
platform
must protect
by control-ling
to
variety
identity authorization request integrity
Assumed by habit culture etc.
can sometimes be just
Jailbreaking Rooting
some others
to have
access
Deny or allow the request (or modify)
resulting
Access control
is realized by requests submitted to
"Complete mediation"
by checking the triple
cross-checks
Initialization incl. authorization
attacked e.g. by
Through asset mgmt (e.g. for files) timestamp: Create /Access /Modify
Log entry in some cases
balances between
verify all requests
sometimes using also
as a setup for new users not directly named yet
DRM
related to
Trusted computing
omit some verifications
always
Zero trust architecture
if not zero, at least
Minimized domains
Anchors
using (for more "solid" anchor types)
Trusted users keys credentials software installations devices paths network connects networks third parties proxies
by using
is first established on
applies strict methods
e.g.
Trust
Testing, audition, certification, manufacturing procedures
Transitivity
also through
if trustworthiness is constantly assessed
then extended to
based on
Delegations (proxying)
Computa-tional trust
Trust management
less secure traditional way
Within a security perimeter e.g. "intranet"
other entities
Calculations
that may output
Yes-or-no trust
results in trusted modules of
distinct topics
Managed devices of an org. with a cert. stored on TPM.
comprising
"Trust domains"
used e.g. in
related
traditionally not large, no unknowns
if anchoring only the identity
Trust level
Kerberos (authentication)
Federated identity
Software, firmware
Operating system
Hardware
e-commerce, P2P, social media, web services, sensor nets, cloud and other outsourcing
Trust metrics on web sites
possibly also
OS verification
related
e.g.
e.g.
e.g
PKI
other use of term WOT
Web of trust
PGP fashion
Software updates
also world wide
Certificates in browser
TET, trusted execution technology
Mobile: Trusted Execution Environment
Security kernel (etc.)
TPM Trusted platform module
typical in

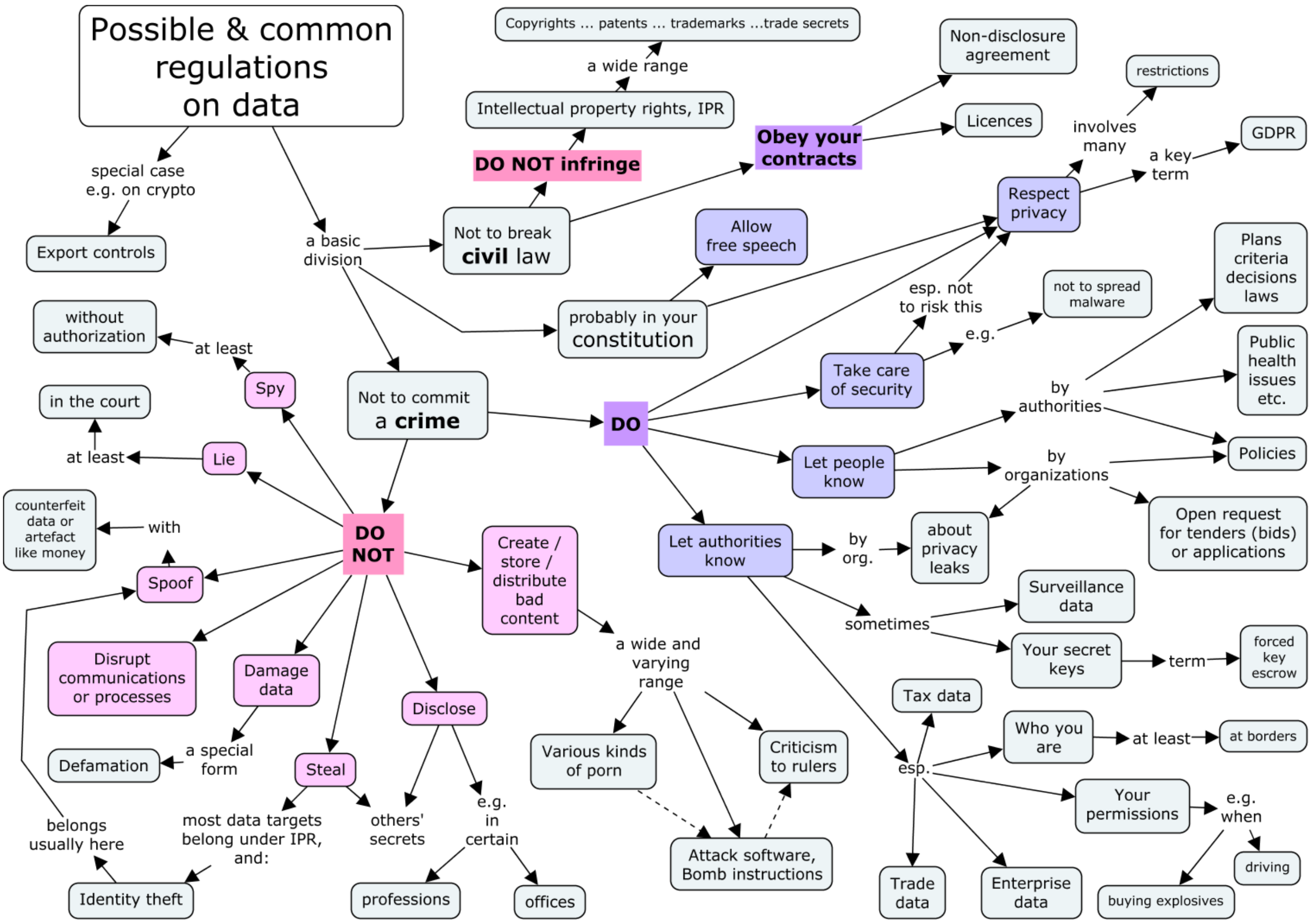
26. Regulations

The concept map of this section focuses on laws, but there are various other levels of regulations, too. Please note especially the TAU Terms of Use of Access Rights and Code of Sanctions in Cases of Access Rights Violations. The latter with its three sanction tables resembles already the criminal law: instead of incarceration it offers “outcarceration” in the form if restricting access rights if they have been abused.

The map for table of contents mentions that data laws of your country are not explained here. The reason is quite obvious. Those who take the comparable course in Finnish must know about the Finnish criminal law, data protection law and copyright law. Even if you may not be a citizen of an EU member country, you must be aware of GDPR, the General Data Protection Regulation of the European Union.

The following concept map organizes many dos and don’ts that relate to information security and are likely to occur in the legislation of your country. You have seen some issues already in other maps: in section 6 (harmful content, immaterial rights), in section 3 (“pax computationis”) and in section 2 (value of data). The “basic division” does not apply so clearly in common law, and religious laws are totally out of this map. For the criminal law case, it is worth noting that some crimes are punishable even if they do not succeed, but there is evidence of an attempt. Port scanning may already be evidence of a break-in, which was not completed, yet.

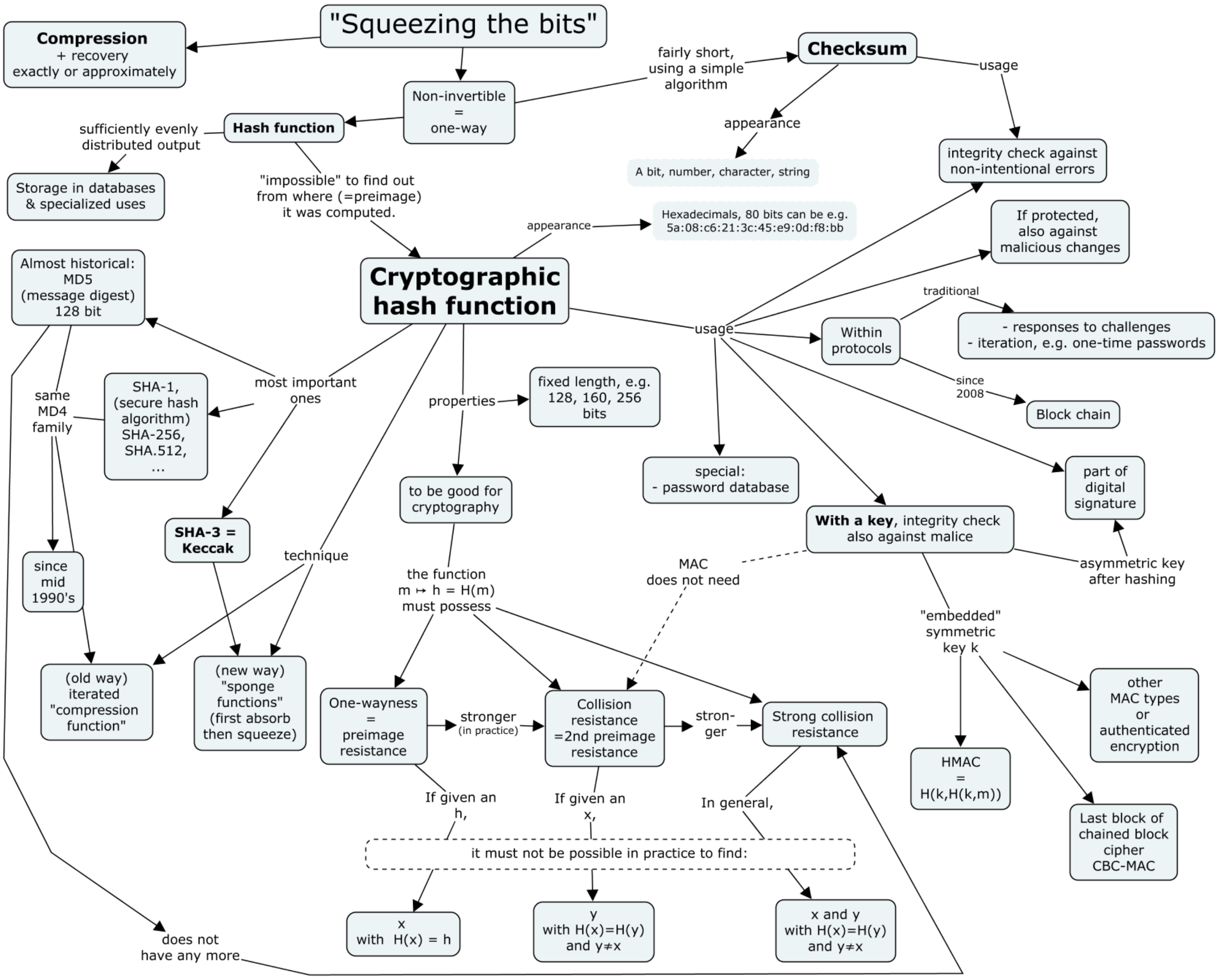
Even without entering a cyber security profession you probably will face situations where your organization must pay attention to compliance with regulations. Interactions across borders will provide a lot of complexity to this task.



27. Crypto

There was a time when cryptography (or ciphers) was one of the first ideas to come into people's mind when they heard about computer security. Now in this document, which supports a basic course in cyber security, this topic is mapped only in the end. It was already used in several places before, though, for instance in section 24, and in the map of mechanisms in section 20.

We begin with a map of hash functions. The entry point to the map is at the top where "bits are squeezed" in general. Compression is not a cryptographic operation, but lossless compression can be used prior to encryption. In this map compression is just a partial reminder of the fact that you must be careful with the terminology. Furthermore, as you can see, hash functions have other uses than cryptography, and there is information security usage to squeezed bits that are not cryptographically hashed. Checksums are sometimes called fingerprints, which is a term that can appear also in connection with a cryptographic hash.

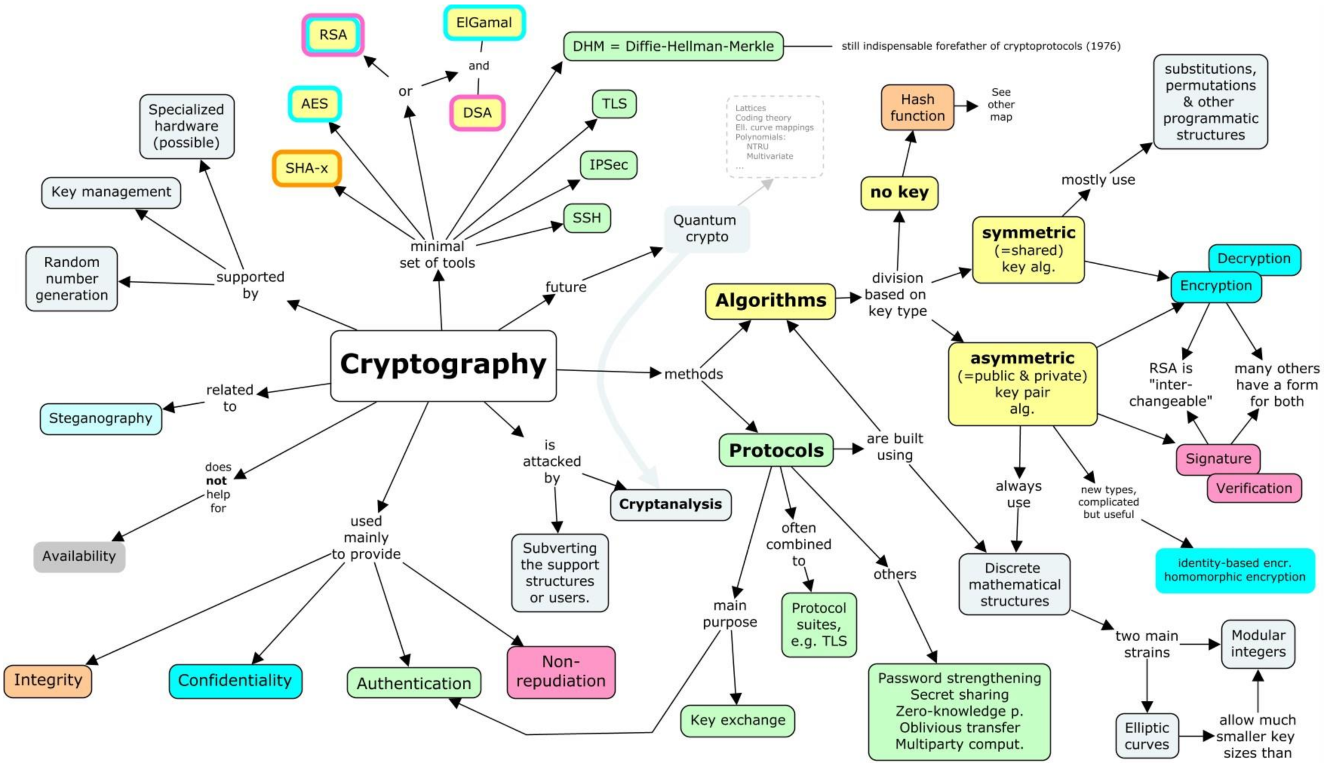


For the rest of cryptography, including hashes, you can find large maps on two CyBOK pages: Crypto "proper" and Applied crypto. These are actually knowledge *trees*, graphic tables of content for the two CyBOK chapters, 10 and 18. If you want to support your study by reading CyBOK, choose Applied crypto. The theoretical chapter 10 does not contain many things that you would need here. Most notable exceptions are the One-time pad and Modes of operation (of block ciphers). Chapter 10 also mentions certain kinds of attacks, like *chosen-plaintext*. These definitions actually mean *adversarial models:* Strength against certain kinds of attacks is the basis for various levels of cryptographic security. "Security games" is one way of defining such strength and also a methodology to prove it. These games are certainly interesting but not very entertaining.

An essential component that is needed to make cryptography secure, that is, to resist cryptanalysis, is *randomness*. Random numbers, or random bit strings are used as keys, or keying material from which keys and other parameters are *derived*. This is another usage for hash functions (not mentioned in the map). The randomness property – or more accurately *unpredictability* – of keys is so crucial for security that methods of producing truly random bits are needed to seed the key derivation functions. A true source of randomness, like atmospheric noise, is something else than an algorithm, but this does not throw randomness out of the crypto realm. One possibility is to collect a *pool of entropy* from measurements of such computing hardware and operations, which usually have unpredictable variations.

Key management is not included in the crypto theory chapter of CyBOK, and that makes sense because as such it is not part of cryptography. Of course *key exchange,* or *agreement* is an essential crypto protocol, and new keys can be derived from earlier ones, but ultimately keys must be managed in some other way[44]. This is an important part of the applied crypto chapter of CyBOK.

Below is a map of “entire” cryptography. The node of hash functions is linked to the above map, and similarly there could be more detailed maps for several other nodes. Instead, there is some internal linking in this map by the use of colours: they are connecting different usages to different algorithm names and types.



Mathematical research together with systems thinking is constantly bringing and improving new approaches that make cryptography applicable to new situations. Block chains were mentioned in the hash map above. In the map below *homomorphic encryption* means "calculable cryptotexts". Of course, you could always add or multiply two encrypted bit strings together, but it is difficult to get anything reasonable – let alone the sum or product of the plaintexts – when decrypting the result. The challenge is to keep the structure calculable but inaccessible to the cryptanalyst.

---

[44] You could not protect a key with cryptography without having another key, and so on. So you keep the last key, or passphrase, in your memory or on paper or other media – probably in a safe. Well, you say, you can split that key with your friends cryptographically. But where do they put their share?

System developers have many ideas to put in practice as soon as fully homomorphic encryption systems become common. The same is true for *identity-based encryption*, where “you are your key”, and for many other advances in cryptography. Even if you do not delve into the mathematical study, you can prosper as a designer of unprecedented systems that rely on security provided by these advances. That should serve as a motivation to study the non-entertaining basics properly. For less-imaginary study goals the motivation for the basics lies just in what the map tries to teach: organizing the main concepts in a meaningful way.

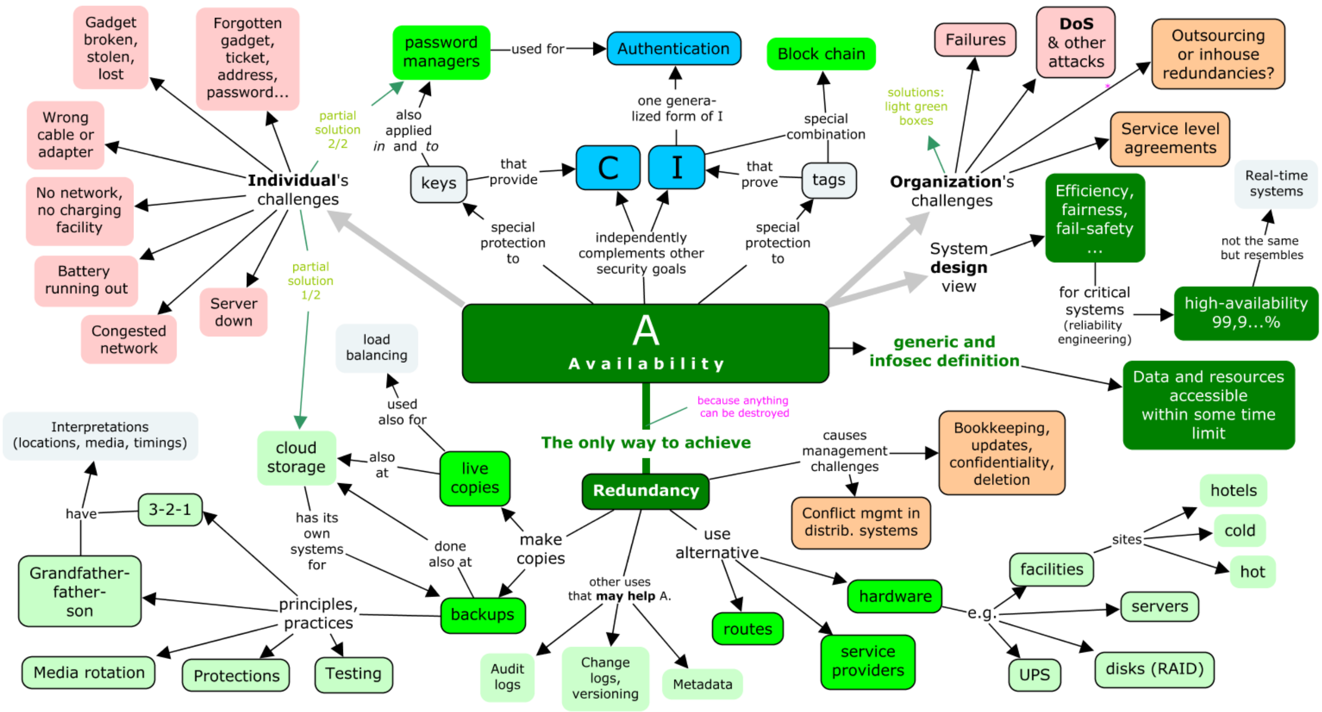
28. Availability

The course includes a small exercise, where you are supposed to take into use and evaluate your choice of tools for password management and cloud storage. The exercise bears the name Availability, and the concept map below shows how these tools are situated in the field of availability. That field is “a critical aspect of cybersecurity, as it ensures that services are provided continuously and reliably, which is essential for businesses, governments, and individuals to function effectively.”

The quote above was written by someone else than this author. It is from the “discussion” at aichatonline.org/app in June 2023 and appeared as the last item in the bulleted list (below), that was the answer to this request:

> This is a link to my concept map (made with IHMC CmapTools) about availability as a cybersecurity concept, complementing "C"=confidentiality and "I"=integrity, [a temporary link at Harpo, giving the source code]. Please write out *the whole of* this as a linear text. Add important points of view if they are missing, *but don't change terms that already exist in the map.* Do not emphasize problems with availability but rather solutions that improve availability.

The two italicized passages were added to the original request, but the result was still almost the same – seemingly “inspired” by the map but still explaining things in the way the (vast) learning materials of the AI had expressed it. The italicized passages in the list below give comments on this. Read the list, study the map, spot the differences, and make a truly intelligent synthesis in your own mind!



- Availability is a cybersecurity concept that complements Confidentiality and Integrity, *but it goes beyond AI capability to grasp the connection made in the map. Can you see it?*
- Ensuring availability means that information and services are accessible to authorized users when needed.
- Availability can be threatened by various factors, such as cyber attacks (e.g. DDoS attacks), natural disasters (e.g. power outages), and human errors (e.g. misconfigurations).
- Some solutions that can improve availability include redundancy, failover mechanisms, load balancing, backup and recovery plans, and disaster recovery plans.
- Redundancy involves having multiple instances of critical components (e.g. servers, network links) to minimize the impact of failures. *Redundancy applies also to what happens when you replicate data during backup.*
- Failover mechanisms ensure that if one component fails, another one can take over its role seamlessly. *This, too, is redundancy (and actually just repeats the last bullet). So, it is rather that the word* redundancy *two bullets above is redundant.*
- Load balancing distributes traffic across multiple components to avoid overloading any single one.
- Backup and recovery plans involve regularly backing up data and having a plan in place to restore it in case of a disaster.
- Disaster recovery plans involve having a plan in place to recover from a major disruption (e.g. a hurricane). *Besides backups, this may require* facilities *as noted at the bottom right corner.*
- In addition to technical solutions, ensuring availability also requires establishing clear policies and procedures, regular training and awareness raising for employees, and effective communication with stakeholders. *Very nice addition, which was in the map only hinted at "principles and procedures". This bullet was given exactly with the same words at the two attempts. So, it must be quite accurate, or what do you think?*

Despite certain doubts, also expressed above (unedited), it took two months for this author to realize the truth. The AI or actually its front end using the ChatGPT API had “lied”, that it can read cmap source code and it would be ok to submit it as a link to a file. In reality, at least through this service, it is not possible to access external links directly. This can be evidenced with any question including a link. As already mentioned, the language model, the AI, has a vast “knowledge” of cyber security concepts including availability. As the question was so verbose, the AI “politely” ignored the map input and gave an overview of the topic. They match fairly well because the map is also a result of wide cyber security knowledge. This is another example (besides those in a separate document) of your challenges if you start doing the Essay exercise at Harpo.

29. What is not mapped or listed?

There are many areas of cybersecurity that are not separately treated in this document. Most of them are present in some maps or lists of general kind, like curricula and principles, and of course everything fits in the ontology. Some new details will appear in section 30 below.

The very first map mentions several topics that are not addressed specifically in the current document: they range from cryptographic (big) details to laws, ethics and awareness. There is at least one important area missing there and deserving a mention here, although it appears in several other places. It is **privacy**, represented for instance in the map of section 6 as both a core and special area of information security. Interestingly, it does not appear in the map of cyber security domains by Henry Jiang, copied for instance into Ray Toal’s course.

Everybody should have a good understanding of privacy, because it is constantly and often explicitly challenged in everyday life. No map or list is shown here. Instead, it would be instructive for you to make one yourself. To encourage you, here is a link to Ian Oliver's rather casual chart of thoughts related to privacy. It is a tree but there are coinciding leaves.

Privacy is a key focus in the TAU course *Digital shadow: privacy and anonymity*. The Wikipedia page on privacy is very large, over 40% of this document. Still, it does not deal with **Differential privacy**, which has its own quite mathematical page reaching the length of 1/3 of the Privacy page.

30. What next

This section mentions some cyber security related issues that are beyond the current course, but which you are likely to encounter later, some even outside a security career. Some of them may also lose significance while the digital environment changes and other types of developments replace them. An example of such is the mobile agents, which were still studied at the turn of the century but not much anymore – while everyone is carrying lots of mobile agents with them.

The concept map below the table of contents mentions some topics that could be included in this document but are omitted. They are suitable topics for the course Cyber Security 2.

The listing is about big and small issues are roughly sorted by the time of appearance of the technologies:

- **MLS**, or multilevel security, originally in military. Includes formal models of information flow, such as the famous Bell-LaPadula model.
- **Digital heritage**: what happens to digital assets (including SoMe presence and the "digital self" (cf. below)) after death.
- **Escrow** of keys or other data (also related to heritage)
- **Anonymity**, realized for instance by **TOR**, onion routing.
- **PUF**, physically unclonable function, a circuit that operates deterministically but in a way that cannot be predicted.
- **Zero trust** architecture (but there is a brief discussion in section 25)
- **Cognitive** cyber security, use of AI in detecting vulnerabilities and attacks by defenders. Of course AI is also used by the attackers, and let an AI "itself" continue on this (June 2023; remember again to judge this yourself, like in section 28)

> Attackers can use AI in a variety of ways to carry out cyber attacks. For example, they can use AI to automate social engineering attacks, such as phishing scams, by creating more convincing and personalized messages that are harder for humans to detect. AI can also be used to bypass security measures, such as authentication systems or firewalls, by identifying vulnerabilities and exploiting them. In addition, attackers can use AI to analyze and monitor their targets' behaviors and activities, allowing them to develop more effective attack strategies.

- **Blockchain**: the consensus mechanisms, applications including NFT (below), criticism.
- **NFT**, non-fungible token. Representing real-world items in the form of bitcoin-like tokens.
- **Life recorder**, voice, video and other data of every moment of our life.
- **Neuroimplants**, information technology operating inside our bodies.
- **Digital self**, our mind represented outside our brain with the aid of robots and AI, in interaction with similar selves of other people.
- **Formal methods** are likely to make a comeback to prove security in a larger scale than in 1990's when they were applied on cryptographic protocols.[45]

[45] A quote from the report of a 2015 workshop : *Adoption of formal methods in various areas (...) has dramatically improved the quality of computer systems. We* ***anticipate*** *that formal methods can provide similar improvement in the security of computer systems*. A summary of a lot more skeptical 2013 book edited by H. Garavel. The book is an interesting example of a large collection of Wikipedia links, not quite unlike section 8 of this document, but it also

- **Quantum** cryptography will not be in your next mobile, but quite possibly in your last one. Even then it is not using quantum computing but new algorithms that don't get broken by it. For that purposes a new course was included in TAU curriculum in 2024 with the name *Post-quantum cryptography engineering*.
- **Side-channels.** This is not a new topic at all, but it entered as a new course in the TAU curriculum in 2024. An open source book is used on that course.
- **"Security and trust in the age of algorithms".** This is one of the three themes for strategic research programmes proposed in 2021 by the Academy of Finland (nowadays *Research Council of Finland*). This goes into the societal area of matters discussed in section 25, but you may well find yourself as a research assistant or researcher in one of the projects of the programme SHIELD that lasts until 2028.

In a very practical sense, what happens next, is that you will pass and leave this course and continue your studies – and perhaps start developing one of the areas above.

contains 92 (book) pages of references. On the positive side of formal methods, note that the CyBOK, as mentioned in sections 3 and 5, introduced formal methods as one of its knowledge areas in version 1.1 in 2021. In version 1.0 they were already included as one of the cross-cutting topics.