A study guide for a first course in Computer Security

Daniel Bosk

Department of Information and Communication Systems
Mid Sweden University, SE-851 70 Sundsvall

1 Scope and aims

The course aims towards a good understanding for the requirements of a secure computer system. Problems such as authentication and access control; software security, such as buffer overflows; as well as operating system, library and application security mechanisms are treated in the course.

More specifically, after taking this course you should be able to:

- apply different cryptosystems and explain how these work,
- analyse the problems of authentication, access control and different solutions,
- explain how some common attacks on software works,
- analyse different operating system security mechanisms,
- analyse the functionality of different types of malware,
- explain different malware protection mechanisms,
- evaluate strengths and weaknesses of hardware-based security and full-disk encryption, as well as
- value and argue about different ethical aspects of computer security, e.g. surveillance.

2 Overview of structure and content

The course covers applied cryptography used in computer security, e.g. uses of cryptography for code obfuscation or digital rights management; authentication mechanisms, access control, and intrusion detection; software security, e.g. buffer overruns and interaction between programs; some security mechanisms provided by operating system and hardware; and malicious software and how these utilise the above weaknesses. Finally we discuss some ethical implications for computer engineers.

2.1 Teaching

The main course literature is *Computer Security* by Gollmann [17]. This is complemented by *Security Engineering* by Anderson [2]. The course is taught using lectures, individual laboratory assignments, workshops ("hackathon labs"), seminars, and finally a written exam. You can find a more detailed timetable,

containing lab sessions etc., in the following subsection. All assignments are numbered consecutively prefixed with an "L" for laboratory assignments, "H" for hackathons and "S" for seminar assignments. For general information about the examination of these and deadlines, see section 4. For detailed information, please see the instructions found in the course platform.

2.2 Schedule

To make your reading of the course easier, you are presented with a suggested schedule in this section. You are free to follow this schedule or any schedule you make for yourself, but the deadlines, laboratory sessions, and lectures will follow this schedule. You will find a short summary of the schedule in table 1. The detailed reading instructions for each item in the schedule can be found in the following sections.

3 Course content

This section summarizes the material covered by the lectures and assignments, i.e. what you should read for each of them. It is divided by topics and ordered according to progression of the course.

3.1 Foundations of security

In this learning session we will cover the foundations of security. By this we mean what security is all about, e.g. what properties we are interested in and what we want to achieve in our security work.

We will focus on Gollmann's chapter on "Foundations of Computer Security" [17, Chap. 3]. There he attempts at a definition of Computer Security and related terms, e.g. confidentiality, integrity, and availability, which we need for our treatment of the topic. After reading this chapter you you are encouraged to do exercises 3.2, 3.5, 3.6, 3.7 and 3.8 in [17]. Anderson also covers this in Chapter 1 of [2]. He also treats a wider area than just *computer* security, which is good for us, he covers many aspects of security in different examples.

3.2 Information theory

The area of Information Theory was founded in 1948 by Claude Shannon. It concerns information, e.g. how much information is contained in certain data. Equivalently, it is also a measure of uncertainty in information, and has thus plenty of application in security and cryptography.

The concept of entropy, the main part of information theory, is treated in a few short texts: *A Primer on Information Theory and Privacy* [12] and applied in "How Unique Is Your Browser?" [13], but also in "Chapter 6: Shannon entropy" [40]. This is then utilised in the text "Grundläggande lösenordsanalys" [8] (in Swedish), and "Of passwords and people: Measuring the effect of password-composition policies" [24] which treats passwords.

Table 1. A summary of the parts of the course and when they will (or should) be covered. The table is adapted to taking this course on half-time study rate, i.e. over ten weeks.

Course week	**Work**
1	Lecture: Course start/Foundations of security Lecture: Security usability
2	Individual study
3	Lecture: Information theory Lecture: Cryptography, part I Lecture: Cryptography, part II Lab: L0 (spuriouslab)
4	Lecture on Identification and authentication Seminar: S2 (pwdpolicies) Lab: L0 (spuriouslab), L1 (pwdguess), L3 (pricomlab)
5	Lecture: Access control Lecture: Reference monitors Lecture: Accountability Lab: L1 (pwdguess), L3 (pricomlab), L4 (tools)
6	Lecture on Trusted computing Lecture on Side-channels Lecture on Software security Lab: L1 (pwdguess), L3 (pricomlab), L4 (tools)
7	Hackathon: H5 (drmlab) Hackathon: H6 (stacksmash) Lab: L1 (pwdguess), L3 (pricomlab), L4 (tools)
8	Hackathon: H7 (malwarelab)
9	Presentation: L3 (tools) Seminar: S8 (ethics)
10	Exam: course exam Lab: last call for all labs Seminar: second call on all seminars, incl L4 presentation
+3 months	Exam: re-exam Seminar: final call on all seminars, incl L4 presentation
+6 months	Exam: last re-exam until next course

3.3 Cryptography

To fully understand how many security mechanisms can be implemented we need cryptography, as cryptography has a central role in a lot of security. This learning session is intended to give a high-level overview of cryptography: symmetric cryptography, public-key encryption, digital signatures, zero-knowledge proofs, and multi-party computation.

We will treat Chapter 5 in Anderson's *Security Engineering* [2] and Chapter 14 in Gollmann's *Computer Security* [17]. To practice your understanding of these mechanisms it is recommended to do exercises 14.2, 14.3 and 14.7 in [17].

3.4 L0 May the (brute) force be with you

This lab focuses on brute forcing approaches, specifically the probability of finding *the* correct solution. It begins by finding the unknown plaintext of a given ciphertext and follows by some time to reflect on the probability of the plaintext indeed being the correct plaintext. The next step is to generate two keys, which given a ciphertext, will yield two equaly probable plaintexts.

We use some classical ciphers. Those are best covered in Chap. 1 of *Cryptography : theory and practice* [37] or "En introduktion till kryptografi" [7]. Finally, you should read about spurious keys and unicity distance. The recommended literature is Chap. 2 in [37].

3.5 Identification and authentication

Authentication has always been a central part of security. An entity claims something, a property or an identity, authentication is about verifying or rejecting any such claim. We will cover a few different ways to do authentication: the traditional something you know, something you have and something you are; but also look beyond.

Why we want to do this, and how we can accomplish this is treated in Chapter 4 in [17]. Anderson also treats this topic (Chapter 2 in [2]), although in a wider perspective with less technical details. When you have studied this material you should do exercises 4.2, 4.3, 4.4 and 4.6 in [17].

3.6 Security usability

One important aspect of security, which traditionally is forgotten, is the users' weaknesses. The psychology of the human mind is therefore an important subject to discuss in the context of security. And consequently, we must adapt our systems to those limitations. How the users function and how to adapt systems to their limitations is at the centre of the usability area.

Anderson gives a short summary of the psychology of users, their strengths and weaknesses, in Chapter 2 "Usability and Psychology" in [2]. Also treated here is the ever-recurring problem of password policies. The material covering this area is the article "Of passwords and people: Measuring the effect of password-composition policies" [24] and its follow-up article "Can long passwords be secure and usable?" [36].

3.7 L1 Password cracking and social engineering

Before doing this laboratory assignment you should read Chap. 2 "Usability and Psychology" and Chap. 5 "Cryptography" in *Security Engineering* [2]. Further, you need a basic understanding of information theory [35] for this assignment, for this you are recommended to read "Chapter 6: Shannon entropy" [40].

Now that you have the basic theory, you should start reading the main material of this assignment. Start by reading the papers *Human Selection of Mnemonic Phrase-based Passwords* [25] and "Of passwords and people: Measuring the effect of password-composition policies" [24]. You should then read the follow-up paper to the latter: "Can long passwords be secure and usable?" [36]. After that you should read about some recent incidents where password databases have leaked, e.g. [10, 11, 19, 29].

For a more in-depth treatment on password guessing, you are recommended to read "Guessing human-chosen secrets" by Bonneau [6]. However, this is not a mandatory part of the assignment.

The final part of the theory concerns advanced persistent threats (APTs). You should read about these. First you should read about an incident striking the security company RSA, covered in "RSA: SecurID Attack Was Phishing Via an Excel Spreadsheet" [15]. Then you will read a paper on different approaches to APT, "Sherlock Holmes and The Case of the Advanced Persistent Threat" by Juels and Yen [21].

3.8 S2 Password policies

First you must read Chap. 2 "Usability and Psychology" in [2]. Further, you need a basic understanding of information theory [35] for this assignment, for this you are recommended to read "Chapter 6: Shannon entropy" [40].

Then, to participate in this seminar you must have read the papers "Of passwords and people: Measuring the effect of password-composition policies" [24], "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms" [22] and "Can long passwords be secure and usable?" [36]. In these papers the authors have studied how different password-composition policies affects users' choice of passwords.

3.9 L3 Private communication

Before starting this assignment you should have read chapters 5 and 23.4.4–5 in *Security Engineering* [2]. You should also read the paper "Exploring steganography: Seeing the unseen" [20] to fully understand how steganography works in practice. (Other recommended papers are "On the limits of steganography" [3] and "Hide and seek: An introduction to steganography" [33].)

During this assignment you should consult the documentation [23, 30, 32, 38] for instructions on how to use the specific softwares.

3.10 Access control

Once you have authenticated users you can support access control – and this is also one of the main reasons to authenticate them in the first place. Access control aims at controlling who may access what, and how they may access it. There are different models and ways to implement access control. We will give an overview of the possibilities.

This is treated by Chapter 5, followed by Chapters 11 and 12, in *Computer Security* [17]. You are also recommended to read Anderson's treatment of the subject, he treats this in Chapters 4, 8, and 9 in *Security Engineering* [2]. Finally, to establish your newly gained knowledge in this area, you should do exercises 5.1, 5.2, 5.5, 5.6, 5.8 and 5.9 in [17].

3.11 Reference monitors

The reference monitor enforces access control policies. It requires care to implement a reference monitor. To do this we need to explore what is the trusted computing base and enforcing access control on the lower layers in the system architecture.

Gollmann treats this area in Chapter 6 of his book *Computer Security* [17]. Exercises 6.1, 6.3 and 6.5 in [17] are recommended for your learning.

3.12 Accountability

The need for accountability has been apparent in civilisations for as long as they have existed. One of today's institutions which is historically renowned for keeping strict accounts is the state tax office, another is, of course, banks. We will explore some principles in keeping accounts and discuss ways to implement it in different, sometimes challenging, environments.

Anderson describes accountability through his experience from banks in Chapter 10 "Banking and Bookkeeping" in *Security Engineering* [2]. We will also use the secure logging system of Schneier and Kelsey [34] as an example of how to achieve secure logging in a challenging environment. The construction described therein is a method to safely store audit logs in an untrusted machine; in the scheme, all log entries generated prior to a compromise will be impossible for the attacker to read, modify, or destroy undetectably.

3.13 L4 Tools of the trade

Before starting this assignment you must have a wide grasp of the theory of security. If you do not, then you will not know of all available mechanisms. Hence you will neither know of all practicalities you will have to solve to use these as a developer.

3.14 Trusted computing

One can only do so much with software. The problem with software and general purpose processors is that the software can be modified and the processor will still execute it. Here we will explore how to ensure the integrity of the computer system before use. As an example, Alice has a laptop while travelling, how can she be sure no foreign intelligence agency inserted a modified version of the operating system during the customs inspection? Or, what about when she left the laptop in the hotel room while having breakfast, perhaps the hotel aide replaced the bootloader to break Alice's full-disk encryption? Another aspect of this is to protect parts of the system from Alice herself, this is what Digital Rights Management is all about. A content owner who only allows using his or her material in a certain way must have some means of ensuring this is enforced. These needs boils down to trusted computing.

We treat the material in Chapters 16, 18 and 22 in *Security Engineering* [2].

3.15 H5 Digital rights management

For this assignment you should first read Chap. 3–5, 16, 18, and 22 in *Security Engineering* [2]. Then you should read Chap. 10, 14–15 in *Computer Security* [17].

After reading the material given above you need to know about programming in assembler, specifically x86-64 assembler and some tools. For this you should read *x86-64 Machine-Level Programming* by Bryant [9]. You also need to be acquainted with some tools, study the manual pages for objdump(1), as(1), and gdb(1).

3.16 Side-channels

When looking at secure systems it is easy to assume they are safe just because the secret keys are not directly reachable. This is not always true. Even if the key storage is unreachable, there is some information that can be extracted anyway. For instance the fact *that* two principals are communicating, *when* they are communicating, the time each operation takes to perform, etc., is not provided any confidentiality. The information possible to extract from this is what is called side-channel information.

An overview of this area is provided in Chapters 17 and 23 of [2]. An interesting paper on this topic is *RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis* [16] where the authors extract RSA keys using acoustic side-channels, i.e. they analyse the sound emitted by the electrical circuitry to find the computations done and hence derive the RSA key used.

There is another aspect of this too, namely covert channels. Covert channels are channels over which communication can take place, even with limited bandwidth, despite the prohibition of this due to the security policy.

3.17 Software security

Perhaps the part of security most people intuitively associate with security, and computer security in particular, is software security. This part of computer security treats vulnerabilities in software, e.g. possibility of buffer overruns or code injections.

Gollmann treats this area in Chapter 10 of his book, *Computer Security* [17]. The recommended exercises to do after reading this material are 10.1, 10.3 and 10.4 in [17].

Anderson also treats this subject—in Chapter 4.4 and Chapter 18 of *Security Engineering* [2]—albeit with less technical details.

3.18 H6 Smashing the stack

To grasp this assignment you must first read Chap. 4, 8, 9, 18 in *Security Engineering* [2] and then you must read Chap. 5–7 (and optionally 8), 10–12, 20, in *Computer Security* [17].

After reading the material given above you need to know some assembly programming, specifically x86-64 assembler and some tools. For this you should read *x86-64 Machine-Level Programming* by Bryant [9]. You also need to be acquainted with some tools, for that reason, study the manual pages for objdump(1), as(1), and gdb(1).

Finally, you should read the main paper of this assignment: the classic paper on stack smashing, the first paper on the matter to be precise, "Smashing the stack for fun and profit" [1].

3.19 H7 Malicous software

To be able to do this assignment you should first read Chap. 5, 7, 10 in *Computer Security* [17]. Then you should read Sect. 21.3 in *Security Engineering* [2].

You should then read the classic paper "Reflections on trusting trust" by Thompson [39]. The assignment will focus on the ideas in this paper.

Although you can probably make it without knowing any assembly language in this assignment, it might come in handy. So, read up on some x86-64 assembly and some tools. For this you should read *x86-64 Machine-Level Programming* by Bryant [9]. You also need to be acquainted with some tools, for that reason, study the manual pages for objdump(1), as(1), and gdb(1).

3.20 S8 The computer engineer's code of ethics

This assignment is based on the Codes of Ethics of two engineering associations. Thus, before you start you must read *Code of Ethics: ACM Code of Ethics and Professional Conduct* [4], *Software Engineering Code of Ethics and Professional Practice* [5], and finally *IEEE Code of Ethics* [14].

Once you have read this you should read two articles analysing Snowden's revelations about the NSA surveillance techniques. The first one is "Making Sense

from Snowden: What's Significant in the NSA Surveillance Revelations" [27]. The second one is "Making Sense from Snowden: What's Significant in the NSA Surveillance Revelations" [26]. Finally, in your favourite search engine, search for the string

"nsa exploit of the day site:www.schneier.com".

Read about a few of the NSA exploits presented there.

Lastly, you should read about the "famous" Apple versus FBI case, where FBI wants Apple to engineer a weakened version of iOS so the FBI can break its security. In particular you should read the article "Apple engineers rebel, refuse to work on iOS amid FBI iPhone battle" [28].

3.21 Final exam

The final exam will assess how well you have achieved the intended learning outcomes of the course. Hence, it covers all the content given above.

4 Assessment

This section explains how the course modules are graded and mapped to LADOK. Table 2 visualizes the relations between modules, credits, grades and LADOK.

Table 2. Table summarizing course modules and their mapping to LADOK. P means pass, F means fail. A–E are also passing grades, where A is the best.

LADOK	Credits (ECTS)	Grade	Course Assignments
I104	0.0	P, F	L0
L104	3.0	P, F	L1, L3, L4, H5, H6
S104	1.5	P, F	S2, S7
T104	3.0	A–F	Exam
Total	7.5	A–F	(Determined by exam)

The written exam will be graded A–E for passing grades, F or Fx for failing grades. You will receive an Fx if you are very close to passing. In this case you may complement your written exam with an oral exam within a week from receiving the result. If you do not take this chance within a week you must retake the exam the next time it is given. The grade of the exam will also be the grade of the course total.

4.1 Handed-in assignments

In general, all hand-ins in the course must be in a "passable" condition; i.e. they must be well-written, grammatically correct and without spelling errors,

have citations and references according to [18] (see also [31] for a tutorial), and finally fulfil all requirements from the assignment instruction. If you hand something in which is not in this condition, you will receive an F with the comment "incomplete".

All handed-in material must be created by yourself, or, in the case of group assignments, created by you or one of the group members. When you refer to or quote other texts, then you must provide a correct list of references and, in the case of quotations, the quoted text must be clearly marked as quoted. If any part of the document is plagiarized you risk being suspended from study for a predetermined time, not exceeding six months, due to disciplinary offence. If it is a group assignment, all group members will be held accountable for disciplinary offence unless it is clearly marked in the work who is responsible for the part containing the plagiarism.

If cooperation takes place without the assignment instruction explicitly allowing this, this will be regarded as a disciplinary offence with the risk of being suspended for a predetermined time, not exceeding six months. Unless otherwise stated, all assignments are to be done individually.

4.2 "What if I'm not done in time?"

The deadlines on this course are of great importance, make sure to keep these! You must have completed the introductory assignment within its deadline. If you do not do this you will be deregistered from the course and your place will be open to other students.

For seminars and presentations there will be three sessions during the course of a year, if you cannot make it to any of those you will have to return the next time the course is given; i.e. up to a year later. All of these sessions will be in the course schedule (in the Student Portal). If you miss a deadline for the preparation for a seminar session, then you have to go for the next seminar even if the first seminar has not passed yet.

Since there might be few students attending these later seminars and presentations, they are merged into one session and you must participate in the entire session. This means that you might have to participate in the discussion of a seminar that you have already passed, but the same applies to the other students — who might have already passed the seminar that you must pass.

Written assignments are graded once during the course, most often shortly after the deadline of the assignment. After the course you are offered two more attempts within a year. In total you have three chances for having your assignments graded over the period of a year. After that you should come back the next time the course is given.

No tutoring is planned after the end of the course, i.e. after the last tutoring session scheduled in the course schedule. If you are not done with your assignments during the course and want to be guaranteed tutoring you have to reregister for the next time the course is given. Reregistration is a lower priority class of applicants for a course, all students applying for the course the first time have higher priority — this includes reserves too.

Thus, if you feel that you will not be done with the course on time, it is better to stop the course at an early stage. If you register a break within three weeks of the course start, you will be in the higher priority class of applicants the next time you apply for the course. You can register such a break yourself in the Student Portal.

References

[1] Aleph One. "Smashing the stack for fun and profit". In: *Phrack magazine* 7.49 (1996), pp. 14–16. URL: http://www-inst.eecs.berkeley.edu/~cs161/fa08/papers/stack_smashing.pdf.

[2] Ross J. Anderson. *Security Engineering. A guide to building dependable distributed systems*. 2nd ed. Indianapolis, IN: Wiley, 2008. ISBN: 978-0-470-06852-6 (hbk.) URL: http://www.cl.cam.ac.uk/~rja14/book.html.

[3] Ross J Anderson and Fabien AP Petitcolas. "On the limits of steganography". In: *Selected Areas in Communications, IEEE Journal on* 16.4 (1998), pp. 474–481. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=668971.

[4] Association for Computing Machinery. *Code of Ethics: ACM Code of Ethics and Professional Conduct*. Accessed on 4 April 2014. URL: https://www.acm.org/about/code-of-ethics.

[5] Association for Computing Machinery. *Software Engineering Code of Ethics and Professional Practice*. Accessed on 4 April 2014. URL: https://www.acm.org/about/se-code.

[6] Joseph Bonneau. "Guessing human-chosen secrets". PhD thesis. University of Cambridge, May 2012. URL: http://www.cl.cam.ac.uk/~jcb82/doc/2012-jbonneau-phd_thesis.pdf.

[7] Daniel Bosk. "En introduktion till kryptografi". 2015. URL: https://github.com/dbosk/introkrypt/releases/download/v1.0/introkrypt.pdf.

[8] Daniel Bosk. "Grundläggande lösenordsanalys". 2013. URL: http://ver.miun.se/courses/security/compendii/pwdanalysis.pdf.

[9] David R. Bryant Randal E. and O'Hallaron. *x86-64 Machine-Level Programming*. Sept. 2005. URL: https://www.cs.cmu.edu/~fp/courses/15213-s07/misc/asm64-handout.pdf.

[10] Graham Cluley. *The worst passwords you could ever choose exposed by Yahoo Voices hack*. July 2012. URL: http://nakedsecurity.sophos.com/2012/07/13/yahoo-voices-poor-passwords/.

[11] Nik Cubrilovic. *RockYou Hack: From Bad to Worse*. Dec. 2009. URL: http://techcrunch.com/2009/12/14/rockyou-hack-security-myspace-facebook-passwords/.

[12] Peter Eckersley. *A Primer on Information Theory and Privacy*. Jan. 2010. URL: https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy.

[13] Peter Eckersley. "How Unique Is Your Browser?" In: *Privacy Enhancing Technologies*. Springer. 2010, pp. 1–18. URL: https://panopticlick.eff.org/static/browser-uniqueness.pdf.

[14] Institute of Electrical and Electronics Engineers. *IEEE Code of Ethics*. Accessed on 4 April 2014. URL: http://www.ieee.org/about/corporate/governance/p7-8.html.

[15] Dennis Fisher. "RSA: SecurID Attack Was Phishing Via an Excel Spreadsheet". Apr. 2011. URL: https://threatpost.com/en_us/blogs/rsa-securid-attack-was-phishing-excel-spreadsheet-040111.

[16] Daniel Genkin, Adi Shamir, and Eran Tromer. *RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis*. Tech. rep. Cryptology ePrint Archive, Report 2013/857, 2013., 2013. URL: http://eprint.iacr.org/2013/857.

[17] Dieter Gollmann. *Computer Security*. 3rd ed. Chichester, West Sussex, U.K.: Wiley, 2011. ISBN: 9780470741153 (pbk.)

[18] D Graffox. *IEEE Citation Reference*. Sept. 2009. URL: http://www.ieee.org/documents/ieeecitationref.pdf.

[19] Troy Hunt. *A brief Sony password analysis*. June 2011. URL: http://www.troyhunt.com/2011/06/brief-sony-password-analysis.html.

[20] Neil F Johnson and Sushil Jajodia. "Exploring steganography: Seeing the unseen". In: *Computer* 31.2 (1998), pp. 26–34. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4655281.

[21] Ari Juels and Ting-Fang Yen. "Sherlock Holmes and The Case of the Advanced Persistent Threat". In: *LEET*. 2012. URL: https://www.usenix.org/system/files/conference/leet12/leet12-final29.pdf.

[22] Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms". In: *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE. 2012, pp. 523–537. URL: http://ieeexplore.ieee.org/abstract/document/6234434/.

[23] Werner Koch. *Using the GNU Privacy Guard*. Mar. 2012. URL: http://www.gnupg.org/documentation/manuals/gnupg.pdf.

[24] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Christin Nicolas, Lorrie Faith Cranor, and Serge Egelman. "Of passwords and people: Measuring the effect of password-composition policies". In: *CHI*. 2011. URL: http://cups.cs.cmu.edu/rshay/pubs/passwords_and_people2011.pdf.

[25] Cynthia Kuo, Sasha Romanosky, and Lorrie Faith Cranor. *Human Selection of Mnemonic Phrase-based Passwords*. Tech. rep. 36. Institute of Software Research, 2006. URL: http://repository.cmu.edu/isr/36/.

[26] Susan Landau. "Highlights from Making Sense of Snowden, Part II: What's Significant in the NSA Revelations". In: *IEEE Security & Privacy* 12.1 (2014), pp. 62–64. ISSN: 1540-7993. DOI: 10.1109/MSP.2013.161.

[27] Susan Landau. "Making Sense from Snowden: What's Significant in the NSA Surveillance Revelations". In: *IEEE Security & Privacy* 11.4 (2013), pp. 54–63. ISSN: 1540-7993. DOI: 10.1109/MSP.2013.90.

[28] Shaun Nichols. "Apple engineers rebel, refuse to work on iOS amid FBI iPhone battle". In: *The Register* (Mar. 2016). URL: http://www.theregister.co.uk/2016/03/18/apple_fighting_fbi_demand/.

[29] Jon Oberheide. *Brief analysis of the Gawker password dump*. Dec. 2010. URL: https://duo.com/blog/brief-analysis-of-the-gawker-password-dump/.

[30] Eng. Cosimo Oliboni. *OpenPuff v4.00 Steganography & and Watermarking*. July 2012. URL: http://embeddedsw.net/doc/OpenPuff_Help_EN.pdf.

[31] Joshua M. Paiz, Elizabeth Angeli, Jodi Wagner, Elena Lawrick, Kristen Moore, Michael Anderson, Lars Soderlund, Allen Brizee, and Russell Keck. *In-Text Citations: The Basics*. Nov. 2013. URL: https://owl.english.purdue.edu/owl/owlprint/560/.

[32] Niels Provos. *outguess - universal steganographic tool*. URL: http://manpages.ubuntu.com/manpages/utopic/man1/outguess.1.html.

[33] Niels Provos and Peter Honeyman. "Hide and seek: An introduction to steganography". In: *Security & Privacy, IEEE* 1.3 (2003), pp. 32–44. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1203220.

[34] Bruce Schneier and John Kelsey. "Secure audit logs to support computer forensics". In: *ACM Transactions on Information and System Security (TISSEC)* 2.2 (1999), pp. 159–176.

[35] C. E. Shannon. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27 (July 1948), pp. 379–423, 623–656.

[36] Richard Shay, Saranga Komanduri, Adam L Durity, Phillip Seyoung Huh, Michelle L Mazurek, Sean M Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. "Can long passwords be secure and usable?" In: *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM. 2014, pp. 2927–2936. URL: http://lorrie.cranor.org/pubs/longpass-chi2014.pdf.

[37] Douglas R. Stinson. *Cryptography : theory and practice*. 3rd ed. Boca Raton: Chapman & Hall/CRC, 2006. ISBN: 1-58488-508-4 (Hardcover).

[38] The Gpg4win Initiative. *The Gpg4win Compendium*. Aug. 2010. URL: http://wald.intevation.org/frs/download.php/775/gpg4win-compendium-en-3.0.0-beta1.pdf.

[39] Ken Thompson. "Reflections on trusting trust". In: *Communications of the ACM* 27.8 (1984), pp. 761–763. URL: http://dl.acm.org/citation.cfm?id=358210.

[40] Daniel Ueltschi. "Chapter 6: Shannon entropy". URL: http://www.ueltschi.org/teaching/chapShannon.pdf.