0. Additional Books

The Internet has several good sources of information on Information Security. For those who want to learn more about the subject - beyond the scope of this course - can start with these two books.

Both are available free online (and legally so as long as you do not print it all out, or decide to sell it to your friends, etc.). If you have read these, and you like the books and you really like information security, then consider supporting the authors by buying the book.

These books are NOT prescribed for the course; they are for your additional interest only.

For an overall view of security (not very technical but covering a lot of topics from a system security perspective)

Security Engineering by Ross Anderson, Wiley.

http://www.cl.cam.ac.uk/~rja14/book.html

If you like the cryptography and protocols try

Handbook of Applied Cryptography

Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone

http://cacr.uwaterloo.ca/hac/

1. Introductory Concepts of Security (Tutorial 1 + Reading)

In Week 1 we will cover some basic security concepts, terminology and security standards.

Slides (pdf): Lecture 1-Introduction - DG.pdf

Notes: Lecture 1-Introduction - DG - Notes.pdf

The notes will often note whether a slide is only for reference or needs to be studied. Reference work is for interest, to complement other core concepts and you do not need to study these in detail for exam of mid-term.

Extra reading and questions for next week: Lecture 1 - Reading.pdf

Tutorial 1 (Week 2): Tutorial 1.pdf

Solutions (Tutorial plus extra reading):

Exercise 1 + Weekly Reading Solutions.pdf

1.1 Human aspect of security

For interest.

Even though we mostly consider security from a technical perspective in this course, you should also recognize the human aspect.

If you understand people, it is entirely possible to 'hack' without needing to understanding technology.

Watch: https://www.youtube.com/watch?v=L5J2PgGOLtE

2. Symmetric Encryption (Tutorial 2+3)

Lecture 2 deals with symmetric encryption algorithms. We start of looking at some historic approaches to symmetric encryption and look in detail at the algorithms and approaches we use today.

Lecture 2 is split across Weeks 2+3. I will post slide notes after lecture in Week 3.

The notes file will be ppt so the animation also works.

Slides: Lecture 2- Symmetric-Key-Encryption.pdf

Slides (ppt): Lecture 2- Symmetric-Key-Encryption-SlidesOnly.pptx

Notes: Lecture 2- Symmetric-Key-Encryption - Notes.pdf

Worked example of how Feistel structure encrypts/decrypts:

Feistel_example.pdf

Tutorial 2 (Week 4 Tue): tut02.pdf

Tutorial 2 Solutions: tut02_sol.pdf

Tutorial 3 (Week 4 Sunday): tut03.pdf

Tutorial 3 Solutions: tut03_sol.pdf

In Lecture 1 reading, Sony's 2011 data breach is an older but very common story. We know confidentiality is on of the main services to address this threat and that encryption can provide confidentiality. Unfortunately data loss like this is not uncommon - and in these cases records are not encrypted. For famous cases see:

https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

This link is a bit older but shows also some different ways the breaches were caused (for example, an employee losing an unencrypted disk by accident in the train or taxi on the way to work is also data loss, it is not different from hacking)

https://digitalguardian.com/blog/history-data-breaches

2.1 Encryption Optional Exercise 1

Try to do frequency analysis on the message on slide 33. If you have an answer that you think makes sense - you can send to me and I will tell you if it is right.

You can do it by yourself, or in a group.

2.2 Symmetric Encryption (Extra Reading)

We have discussed some basic ciphers/encryption - can you recognize some of these concepts in a real world system?

Please look at the EMVCo website - http://www.emvco.com/

EMVCo sets specifications for the majority of smart payment cards (and now mobile/payment tokens, etc.) used in electronic payments. About 73% of all card payment transactions conducted worldwide is done as specified by EMV. Look at the specifications and identify what symmetric encryption algorithms are being recommended for use.

Which algorithm would you use if you had a choice?

http://www.emvco.com/specifications.aspx?id=223

Look at "Book 2 - Security and Key Management"

Answers: TBP

This is not an assignment for grade, and you do not have to do it.

2.3 AES (for interest)

If you are interested in alternative explanation of how AES works see (try it - it gets more detailed as it gets along and you can stop at any point where you think you know as much as you are interested in):

http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html

Links to an external site.

If you would like to see how the key scheduling is down for 192 and 256 keys (essentially very similar except number of words increase), see:

https://www.samiam.org/key-schedule.html

2.4 Zodiac 340 Code (Interest Only)

In December 2020, some hobby code breakers solved a code that people have been trying to solve for 50 years. Substitution ciphers with some variations (unknown ordering of plaintext, custom alphabet, etc.) can be very tricky to solve even with modern technology.

https://www.bbc.com/news/world-us-canada-55285805

https://www.youtube.com/watch?v=-1oQLPRE21o&feature=youtu.be

3. Number Theory (Tutorial 4)

Lecture slides (pdf): Lecture 3- Number-Theory.pdf

Lecture Notes: Lecture 3- Number-Theory - Notes.pdf

Extra example of modular inverse and square and multiple. Note that the first modular inverse example corresponds to the complex RSA example in Lecture 4 and second modular inverse example corresponds to the complex ElGamal example in Lecture 4.

More detailed working of concepts (notes written by previous TA, who also took this course): Breaking Down Number Theory Notes.pdf

Tutorial 4: Tut4.pdf

Tutorial Solution: Tutorial4_sol.pdf

3.1 Number Theory and Practical Security (Interest Only)

Some people think that cryptographic algorithms and the underlying mathematics are mostly theoretical with little practical use. It is true that if you work in security, it is unlikely that someone asks you do design an algorithm like AES or RSA (and you should not do so really as there are enough good algorithms to use).

However, it is still important to have basic knowledge of how algorithms work at low level. A very practical area of security that is all about low level implementation of algorithm is side channel analysis. This is the idea that observing the device during calculation of algorithm can give us and idea what it is calculating. This is especially problematic if we are doing cryptographic calculation with secret/private keys.

I briefly told you about this in the lecture but this video also gives a good idea of some basic use of side channel (it is the issue surrounding RSA and square and multiply algorithm, they also mention Chinese Remainder Theorem)

https://www.rambus.com/side-channel-analysis-demo-mobile-device/

Side channel analysis can also be used other algorithms, for example AES (then it is not square and multiply, but inner working of the substitution/permutation):

https://www.youtube.com/watch?v=l5Oi9xNR60s&feature=emb_rel_end/

(Side note: Cryptography Research (now Rambus) was started by some guys who discovered differential power analysis and made some countermeasure for it - this made a lot of money as at the time everyone who made any crypto processor (e.g. every credit card with chip) had to pay them to use their idea. So very practical knowledge...)

4. Asymmetric Encryption

Lecture Slides (pdf): Lecture 4 - PKE.pdf

Slide Notes: Lecture 4 - PKE - Notes.pdf

Tutorial combined with Lecture 3 (Tutorial 4).

We have now done asymmetric and symmetric encryption, if you are interested you can look at recommended key lengths for different type of cryptography: https://www.keylength.com/

For interest - the alternative story of DH and RSA invention

https://cryptome.org/ukpk-alt.htm

http://aperiodical.com/2016/03/gchq-has-declassified-james-elliss-papers-on-public-key-cryptography

5. Integrity (Tutorial 5)

Lecture Slides (Updated): Lecture 5 - Signature-Hash-MAC.pdf

Lectures Notes: Lecture 5 - Signature-Hash-MACNotes.pdf

Tutorial 5: Tutorial5.pdf

Tutorial 5 Solution: Tutorial 5 Sol.pdf

If you are wondering why M should be less than n for RSA then look at Tutorial 4.

5.1 Signatures and hash (Reading for Interest)

Digital signature actually plays a large role in real life applications (so too does hashes if you consider that the hash creates the short digest from the message that is signed).

Many countries in the world assign Digital Signature the same important legally as hand signatures. For example, in Hong Kong the Electronic Transaction Ordinance states that digital signatures is a considered a valid signature if "it is reliable, appropriate and agreed by the recipient of the signature". This means the parties must agree to use it and the algorithm must be accepted as appropriate.

http://www.ogcio.gov.hk/en/regulation/eto/

In the lecture I also briefly mention the speeding fine that was cancelled because the offender had brought into question the integrity of the hash function used (MD5 - on which collisions could be found). You can read more about it here (it was in 2005 and a similar argument will likely fail today).

NSW speed cameras in doubt - National - theage.com.pdf

MD5 flaw pops up in Australian traffic court - CNET.pdf

As this was the last lecture with algorithms I just wish to remind you that it is important to have a basic understanding of algorithms, however it is unlikely that many of you will implement them. Using these will most often require you to use and apply the correct libraries, like for example:

http://developer.android.com/reference/javax/crypto/package-summary.htmlLinks to an external site.

https://www.openssl.org/

However, given what you learnt you should be able to choose the better algorithms and understand what services to use these for.

Finally, if you are more interested looking at things from a system security perspective you should remember that it does not matter if your cryptographic methods are good if your code is bad…

https://www.imperialviolet.org/2014/02/22/applebug.html

5.2 SHA-3 (Keccak) (for interest)

If you are really interested in how SHA-3 works on the inside (this is a for interest only, and bit beyond scope of this course).

Specification is here:

https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

Keccak maintains a website with much information

https://keccak.team/index.html

There is some pseudo-code of the internal permutations:

https://keccak.team/keccak_specs_summary.html

6. Authentication (Tutorial 6)

Slides: Lecture 6- Authentication.pdf

Authentication Notes: Lecture 6- Authentication-Notes.pdf

PPT Slides: Lecture 6- Authentication-Slides.pptx

Tutorial 6: Tutorial 6.pdf

Tutorial 6 Solution: Tutorial 6 Sol.pdf

7. Key Management (Tutorial 7+8)

Lecture Slides: Lecture 7 - Key management.pdf

Lecture Notes: Lecture 7 - Key management - Notes.pdf

PPT Slides: Lecture 7 - Key management_slides.pptx

Tutorial 7: Tutorial07.pdf

Tutorial 7 Solutions: Tutorial 7 Solutions.pdf

Tutorial 8: Tut08.pdf

Tutorial 8 Solutions: Tutorial 8_Solutions.pdf

7.1 Signal Protocol (Extra Reading)

For interest only.

The Signal protocols is used to provide key management for end-to-end encryption in messaging applications (like WhatsApp).

Given what we have done so far you can look at what is a modern, complex applied cryptography design. I think that most of what they mention you can start to recognize some things from what we did: Diffie-Hellman (although Elliptic Curve) and HMAC.

I made a summary of it in a few slides:

Signal Encrypted Messaging.pdf

Knowing how Elliptic curve works is not important to understand the overall Signal design, but if you do want to look into it briefly then see:

https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/

If you want to see the original source:

https://signal.org/docs/specifications/doubleratchet/

If you are interested in crypto and messaging, including group messaging, you can watch these computerphile videos:

Messaging/crypto: https://www.youtube.com/watch?v=DXv1boalsDI

Double ratchet: https://www.youtube.com/watch?v=9sO2qdTci-s

Group messaging: https://www.youtube.com/watch?v=Q0_lcKrUdWg

8. Computer Security (Tutorial 9)

Slides: Lecture 8 - CompSec.pdf

Notes: Lecture 8 - CompSec - Notes.pdf

Tutorial 9: Tut9.pdf

Tutorial 9 Solution: tut9_solutions.pdf

More information on Rainbow Table

https://www.geeksforgeeks.org/understanding-rainbow-table-attack/

This approach is used to find inputs that will result in a given hash value. So if you have a hash and you need to find an input that will result in this hash - this was an approach used for password searching (and other hash searches). The idea is that you do not necessarily find the password but and input that the verifier will hash to the correct value, i.e. you password is y and system stores h(y) - I find x so that h(x)=h(y) - it does not matter that I enter x as the result will be the same as if I entered y (the system will let me in).

This approach has advantages when storage was more limited and we cannot store all hash results. It is argued that with today's resources brute force/dictionary is improved.

We would start by hashing value a.

h1= h(a), h2= h(h1), h3= h(h2), h4= h(h3) ...

h100 = h(h99) ... h200 = h(h199) etc.

We would now store only some points: h1 h100 h200

now we find a candidate hash z = h(?)

We start hashing z

z1 = h(z) z2 = h(z1) z3 = h(z2)

Now we see that z3 is equal to h100. So we go back to the stored hash point prior to h100, which is h1. We start doing the hashes and find z= h(97), so a valid input? that will result in z if hashed is h(96).

9. Network Security (Tutorial 10+11)

Lecture Slides: Lecture 9 - Network.pdf

Slide notes: Lecture 9-Network-Notes.pdf

If you are interested on the open GSM base station project mentioned, see: http://openbts.org/w/index.php?title=Main_Page

Here is a guy running it, to make some phone calls, etc.: https://www.youtube.com/watch?v=pTb1_v8M6iA

Article to read (will discuss in class):

Tutorial 10: Tut10.pdf

Tutorial 10 Solutions: Tut10 solution.pdf

Tutorial 11: Tutorial 11.pdf

Tutorial 11 Solutions: Tut11 solution.pdf

10. Selected Topics

For the last lecture I select a topic that I think is of current importance. I have for this section selected some common web security issues.

This lecture is not examinable - it is just for interest so there are no notes.

This is if we happen to work through the normal course material and we have some spare time.

If you wish to know more see:

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

https://www.owasp.org/index.php/Main_Page

Lecture 10: Lecture10-Misc.pdf

11. Revision

Slides: No slides before lecture, will put up slides afterwards:

Revision.pdf

Please do take a look at this - I will talk about these in class:

signer-casestudy-sdc.pdf

We will do PS2 answers and very very quick overview of main points in the course.

Exam + Practice Questions

Practice question: Additional Questions for Practice.pdf

I will provide feedback/answer to you if you email your answers to me.

Exam:

Closed book exam

–1 A4 sheet (double side, you can put on it what you want)

–Calculator allowed (no phone)

Outline of the exam (2 hours, 100 marks):

–Q1: Symmetric Encryption (25 marks)

–Q2: Public Key Cryptography and Message Integrity (25 marks)

–Q3: Key Management and Authentication (19 marks)

–Q4: Network and Computer Security (21 marks)

–Q5: Security Engineering (10 marks)