



NTNU

Norwegian University of
Science and Technology

COURSE INTRODUCTION

TTM4205 – Lecture 1

Tjerand Silde

18.08.2026

Overview

Course Staff

Motivation

Course Description

Earlier Semesters

Fall 2026

Background Knowledge

Contents

Course Staff

Motivation

Course Description

Earlier Semesters

Fall 2026

Background Knowledge

Tjerand Silde

- ▶ Associate Professor in Cryptology at IIK
- ▶ Research Group Leader at the NTNU Applied Cryptology Lab (NaCl)
- ▶ PhD in crypto and privacy from NTNU
- ▶ Work as Security and Cryptography Expert at startup Pone Biometrics
- ▶ Teach TTM4135 Applied Cryptography and Network Security in the spring
- ▶ Have earlier taught Linear Algebra (M3) and Discrete Mathematics at NTNU



Teaching Assistants

The teaching assistants run the exercise classes and the ChipWhisperer lab sessions, and follow up questions on the Canvas forum:

- ▶ Jo Remvik – jo.v.remvik@ntnu.no
- ▶ Thomas Houillier – tbhouill@stud.ntnu.no

The CW SCA lab tutorial on September 15th will be given by Caroline Sandsbråten (PhD candidate in crypto at NTNU).

Contents

Course Staff

Motivation

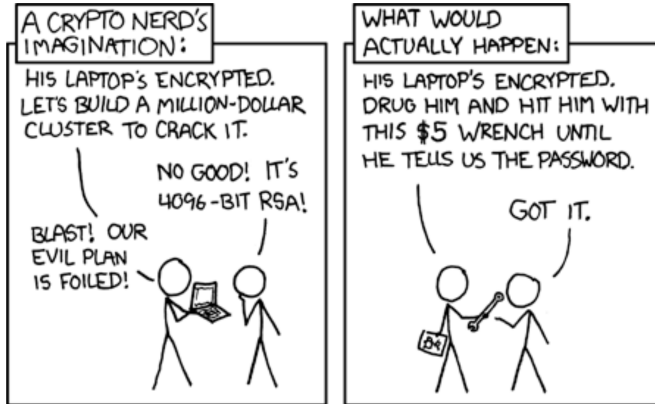
Course Description

Earlier Semesters

Fall 2026

Background Knowledge

Mathematical Security vs. Real-World Security



Mathematical Security vs. Real-World Security

It is somewhere in between the above, and we need to protect against:

- ▶ correctness errors and lack of parameter checks
- ▶ side-channel and fault injection attacks
- ▶ weak or faulty randomness generation
- ▶ mismatch when composing protocols
- ▶ lack of integrity checks and bad padding

Context

- ▶ IIK created a new MTKOM profile: Cryptographic Engineering
- ▶ We wanted a new practical engineering course in cryptography
- ▶ There is a high demand from academia, industry, and government
- ▶ Very few people know cryptographic engineering in Norway...

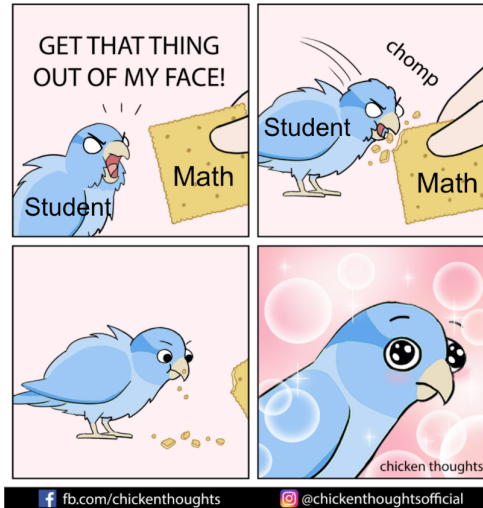
Context

Personal Reasons

I wanted to create a fun and exciting course that I wish I could have taken as a student, and acquire new knowledge that I can use for my own research.

I have included topics I hope you will find interesting and the industry will appreciate that you are familiar with. I want the course to be practical, project- and group-based, with oral presentation and reports instead of a final exam.

Goal



Contents

Course Staff

Motivation

Course Description

Earlier Semesters

Fall 2026

Background Knowledge

Course Content

The course covers how to implement, analyse, attack, protect and securely compose cryptographic algorithms in practice. It goes in depth on how to

- ▶ implement computer arithmetic
- ▶ attack implementations using side-channel attacks and fault injection
- ▶ exploit padding oracles and low-entropy randomness
- ▶ utilise techniques to defend against these attacks
- ▶ securely design misuse-resistant APIs



Learning Outcome

Knowledge

Advanced knowledge about the mathematical building blocks underlying modern cryptography, properties of and applications of cryptographic primitives, challenges and common mistakes when implementing cryptography, side-channel attacks and countermeasures, and high level design principles for secure use of cryptography in practice.

Learning Outcome

Skills

Able to implement the underlying mathematics and high-level protocols used in symmetric key and public key cryptosystems, perform simple side-channel attacks and implement countermeasures, analyse side-channel countermeasures and design misuse resistant APIs for cryptography.

Learning Outcome

General competence

Experience on how to organise projects in small groups, conduct experiments, and write academic reports.

Learning Methods and Activities

Lectures, individual assignments, group projects, and laboratory exercises.

Further on Evaluation

Portfolio assessment is the basis for the grade in this course. The portfolio consists of one or more projects covering implementation, analysis, attacks and protection of cryptographic primitives, including a final practical assignment given at the end of the semester. This will be announced at the beginning of the term.

Further on Evaluation

The work on all tasks composes 100 % of the final grade. The results for the projects are given in points and in %-scores. The entire portfolio is assigned a letter grade. All assignments will be given in English only and reports must be submitted in English.

Further on Evaluation

If a student has the final grade F/failed, the student must repeat the entire course. Also in the case a student wants to improve their grade, they must repeat the entire course.

Recommended Previous Knowledge

The following or equivalent courses are recommended:

- ▶ TMA4140 Discrete Mathematics
- ▶ TDT4100 Object-Oriented Programming
- ▶ TDT4120 Algorithms and Data Structures
- ▶ TTM4135 Applied Cryptography and Network Security

It is also recommended to take TTM4138 Wireless Network Security, TTM4195 Blockchain Technologies and Cryptographic Tools, and/or TMA4160 Cryptography at the same time as this course.

Course Materials

To be announced at the beginning of the term.

The main course material will be given in the form of slides, notes, manuals, research papers, books and recordings.

Useful course material:

- ▶ ChipWhisperer: <https://www.newae.com/chipwhisperer>
- ▶ *Serious Cryptography* by Jean-Philippe Aumasson
- ▶ *Real World Cryptography* by David Wong
- ▶ *The Hardware Hacking Handbook* by van Woudenberg and O'Flynn



Contents

Course Staff

Motivation

Course Description

Earlier Semesters

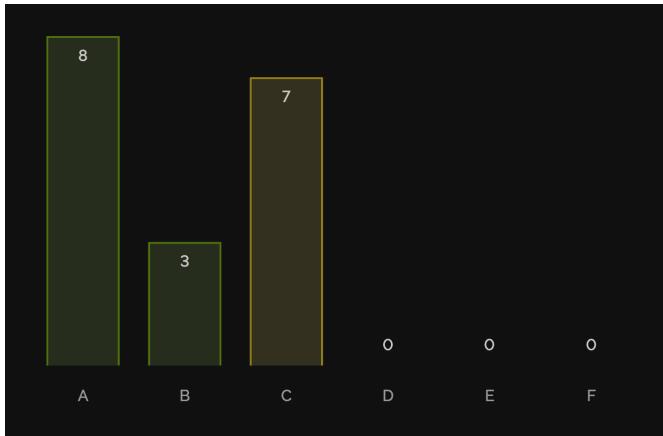
Fall 2026

Background Knowledge

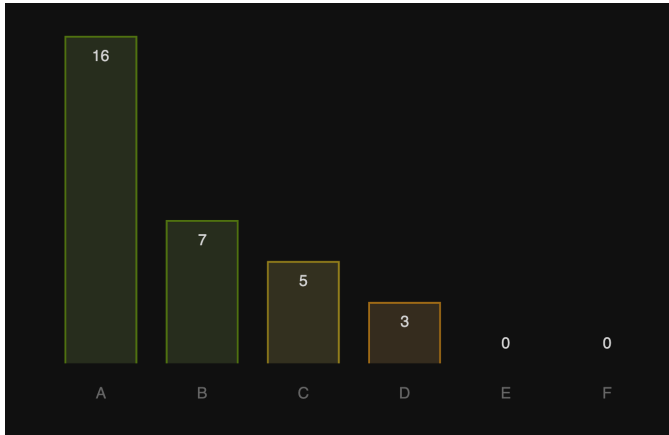
Course Information

The course material, grade distributions, and student evaluations from fall 2023, 2024, and 2025 are available on the current course website. The content is similar this year, but we have made some improvements based on the student feedback and experience from the lectures and grading.

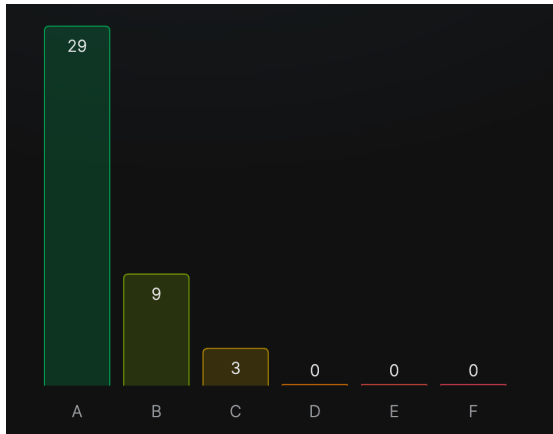
Grade Distribution 2023



Grade Distribution 2024



Grade Distribution 2025



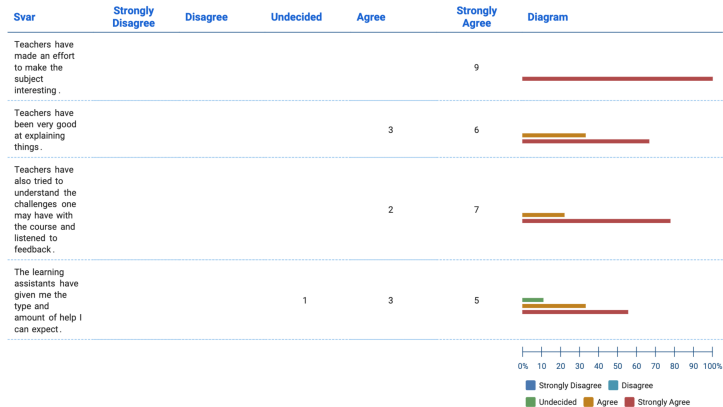
Learning Activities

Learning Activities



Course Information

Teachers and Learning Assistants



Contents

Course Staff

Motivation

Course Description

Earlier Semesters

Fall 2026

Background Knowledge

Course Information

This is the fourth time this course has ever been organized. We have planned well, but some things might go differently, and your feedback is essential.

We will make adjustments during the semester and provide help to everyone.

All course material will be made available at <http://ttm4205.iik.ntnu.no>.



Lecture Plan

This course consist of lectures and lab/exercise sessions, but note that the format varies based on the topic of the week. Watch the lecture plan carefully at the course website. The Friday session on October 23rd is already canceled.

Sessions fall 2026: Tuesdays at 12:15-14:00 (lecture OR lab) and 14:15-16:00 (exercises OR lab) in H2, and Fridays at 12:15-14:00 (lecture) in R10. Today is a double lecture: this introduction, followed by *From Enigma to Crypto Wars*.

SCA lab tutorial by Caroline Sandsbråten on September 15th, and guest lecture by Diego F. Aranha (Aarhus University) on October 27th.



Lecture Plan

Week	Date	Format	Responsible	Topic	Resources
34	18/8	Lecture x2	Tjerand	Intro + From Enigma to Crypto Wars	
34	21/8	Lecture	Tjerand	Randomness 1: Entropy	
35	25/8	Lecture	Tjerand	Randomness 2: Breaking ECDSA	
35	25/8	Exercises	Jo/Thomas	Exercise Class	
35	28/8	Lecture	Tjerand	Randomness 3: Randomization	
36	1/9	Lecture	Tjerand	Legacy Crypto and Attacks on TLS	
36	1/9	Exercises	Jo/Thomas	Exercise Class	
36	4/9	Lecture	Tjerand	Padding Oracles 1: CBC and SHA	
37	8/9	Lecture	Tjerand	Padding Oracles 2: RSA Encryption	
37	8/9	Exercises	Jo/Thomas	Exercise Class	
37	11/9	Lecture	Tjerand	Side-Channel Attack (SCA): Intro	



Website and Canvas

All course content lives on the course website <https://ttm4205.iik.ntnu.no>: the schedule, slides, three assignments, and the material from earlier years.

Canvas is used for announcements, the discussion forum, submitting the assignments, and signing up for groups. Ask your questions in the forum rather than by email, so that everyone gets the answer; email me about personal matters at tjerand.silde@ntnu.no.

The forum supports $\text{\LaTeX}$, and we recommend it for anything with maths.



Portfolio Assignments

The evaluation consists of three assignments worth 100 points in total, and there is no final exam:

- ▶ Weekly Problems (40 points, individual)
- ▶ ChipWhisperer Lab (30 points, groups of 2–3)
- ▶ Technical Essay (30 points, groups of 2–3)

You keep the same group for the lab and the essay. The letter grade follows from the official NTNU grading scale for percentage points.

Use of AI

The same rule applies to all three assignments: what you hand in must be your own work, and you may not use AI tools to solve an assignment or to produce your write-up. Polishing a finished submission is fine.

Two of the assignments then allows a little more: the lab lets you use AI to troubleshoot the technical setup, and the essay lets you use it to research your topic and review a draft. The essay requires a footnote saying how you used it.

The exact wording is on the first page of each assignment.



Why We Restrict It

This is not about distrust, and not about AI being useless. It is about the order you learn things:

- ▶ You cannot judge an answer in a field you have not learned yet: subtly wrong cryptographic code looks exactly like correct code
- ▶ Asking the right question is most of the work, and that skill comes from struggling with the problem yourself first
- ▶ Intuition for what is plausible is built by getting things wrong and finding out why, which is exactly the part a tool takes away from you
- ▶ The failures here are silent: a broken implementation still encrypts, still verifies, and still passes your tests



Why We Restrict It

The restriction is temporary and deliberate: learn it properly now, while you have the time, the exercises, and people to ask, and then the tools become genuinely useful to you later.

Employers will hand you the tools anyway. What they need is your judgment to tell when the output is nonsense, and that is what we are building here.

The guidelines are similar as for a master's thesis: the work and writing must be yours, and you declare what tools you used. Treat this as practice.

Weekly Problems



Weekly Problems

This individual assignment is worth 40 points. It contains 27 problems worth 2 points each:

- ▶ Mathematics problems
- ▶ Coding problems
- ▶ CryptoHack problems

That is more than the cap, so you choose freely which ones to solve and do not have to solve them all. Solutions written in $\text{\LaTeX}$, using our template. The deadline is **December 4th at 23:59**.



ChipWhisperer Lab



ChipWhisperer Lab

This group assignment is worth 30 points and covers:

- ▶ Side-channel attacks (measure time and voltage during computation)
- ▶ Fault injections (make things go wrong or skip instructions)
- ▶ Analyse captured data (mean, average, difference, graphs)

The four parts contain 37 points of work, so you can skip tasks that do not work out and still get full marks. Report written in $\text{\LaTeX}$, using our template. The deadline is **December 4th at 23:59**.

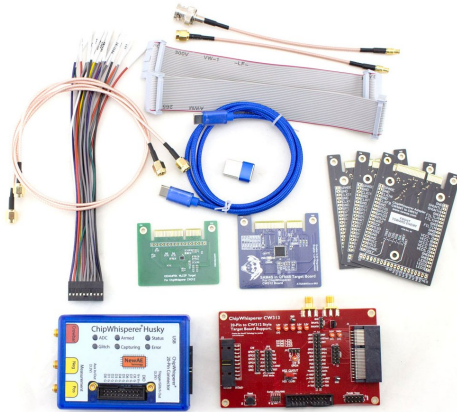


Figure: ChipWhisperer Husky

Technical Essay



Technical Essay

This group assignment is worth 30 points. You write a technical essay and present it, on a topic either not covered by the lectures or covered in more depth. We suggest some topics, but you are welcome to propose your own.

Technical Essay

Most important guidelines:

- ▶ Groups of 2 or 3 students, the same as for the ChipWhisper lab
- ▶ Essays of 8 to 10 pages (+ front page and references), written in $\text{\LaTeX}$
- ▶ The essay is graded on technical content and significance, analysis and reasoning, and structure, clarity and referencing: 10 points each
- ▶ The presentations are mandatory: without it, essays are not graded

Technical Essay

Deadlines:

- ▶ Topic/scope/group approval: **October 30th**
- ▶ Short oral presentations: **November 10th, 13th, or 20th**
- ▶ Final submission: **December 18th at 23:59**

In the proposal, name at least two topics from our list in order of preference, or one of your own with a title, scope, and two references. Rank the three presentation dates as well, since we assign the slots from those rankings.

Course Material

- ▶ We will make all the slides available on the course website
- ▶ You do not need to buy any books but we give recommendations
- ▶ We keep many of the recommended books, and more, in the CRYPTO-LAB, and you are welcome to borrow them during the semester
- ▶ You can make an account for free at <https://cryptohack.org>
- ▶ We provide $\text{\LaTeX}$ templates for all three assignments
- ▶ Each group borrows a ChipWhisperer box from the CRYPTO-LAB in Electro A175, and may take it outside of the lab if you prefer

Reference Group

We highly value constructive feedback and encourage you to join the reference group. This is a course in active development, and you will have more impact than in most other reference groups.

Send me an email to join. We plan three meetings during the semester.

Contents

Course Staff

Motivation

Course Description

Earlier Semesters

Fall 2026

Background Knowledge

Encryption Security

We have two common notions for encryption security:

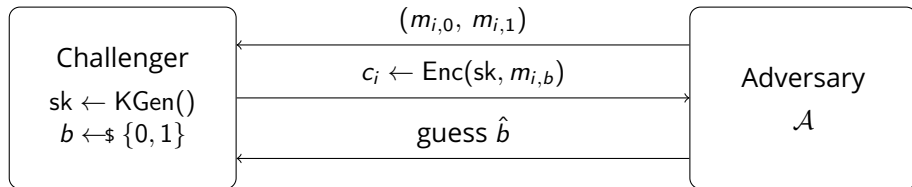
- ▶ IND-CPA: indistinguishability under chosen-plaintext attack
- ▶ IND-CCA: indistinguishability under chosen-ciphertext attack

The difference lies in what kind of power we give to the adversary.

RSA or ElGamal encryption is only IND-CPA secure by default.

IND-CPA: the Attack Game

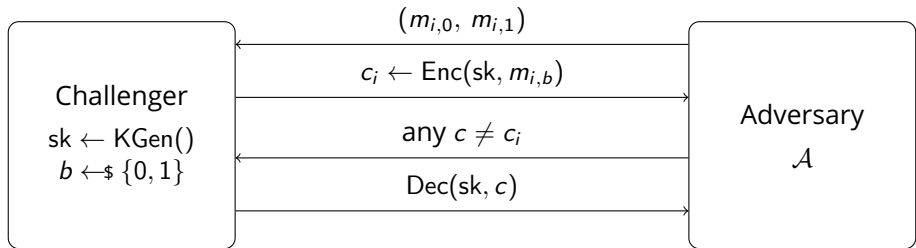
The challenger picks a bit $b \leftarrow \$ \{0, 1\}$ and a key. The adversary $\mathcal{A}$ sends message pairs and gets back encryptions of the b -th one; it wins by guessing b .



$\mathcal{A}$ wins if $\hat{b} = b$, and the scheme is secure if $\Pr[\hat{b} = b] \leq 1/2 + \varepsilon$ for a negligible ε . Note that $|m_{i,0}| = |m_{i,1}|$, and that the adversary may query many pairs.

IND-CCA: the Attack Game

The same game, but $\mathcal{A}$ also gets a decryption oracle, and may query it on any ciphertext except the challenge ciphertexts.



This is the notion that matters in practice: padding oracles, error messages, and timing differences are all ways of giving the attacker this oracle for free.

Encryption Security: Where to Read More

These games follow Boneh and Shoup, *A Graduate Course in Applied Cryptography*, and are redrawn here in a compact form:

- ▶ Chapter 5: chosen plaintext attacks and semantic security
- ▶ Chapter 9: chosen ciphertext attacks and authenticated encryption
- ▶ Chapters 11 and 12: the same notions for public-key encryption

The book is free at <https://toc.cryptobook.us>.



Finite Fields

A *field* is a set with addition and multiplication where every non-zero element has a multiplicative inverse. A *finite* field has finitely many elements.

- ▶ Every finite field has order p^k for a prime p and an integer $k \geq 1$
- ▶ We almost always work with $\mathbb{F}_p = \mathbb{Z}_p = \{0, 1, \dots, p-1\}$, where addition and multiplication are taken $\mod p$
- ▶ Inverses exist because $\gcd(a, p) = 1$ for all $a \neq 0$: the extended Euclidean algorithm gives a^{-1} , and Fermat gives $a^{-1} = a^{p-2} \mod p$
- ▶ $\mathbb{F}_p^* = \mathbb{F}_p \setminus \{0\}$ is cyclic of order $p-1$, which is where discrete logarithms live

Careful: $\mathbb{Z}_n$ for composite n is a *ring*, not a field. Elements sharing a factor with n have no inverse, and finding one is exactly what breaks RSA.

Elliptic Curves

Let $E_{a,b} : y^2 = x^3 + a \cdot x + b$ be an elliptic curve over a finite field $\mathbb{F}_p$ of prime order p , where $a, b \in \mathbb{F}_p$. The group $E_{a,b}(\mathbb{F}_p)$ consists of the point at infinity $\mathcal{O}$ and all pairs $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ satisfying the curve equation. We write $\eta_{a,b}$ for the number of points, which need not be prime.

$E_{a,b}(\mathbb{F}_p)$ is a group, so we can add points, $P + Q$, and multiply by a scalar, $[k]P = P + \dots + P$ (k times). The point $\mathcal{O}$ is the identity, and $-P = (x, -y)$.

Two facts we will use repeatedly:

- ▶ Scalar multiplication is easy, but recovering k from $[k]P$ (the discrete logarithm) is believed to be hard: this is what ECDSA and ECDH rest on
- ▶ If $\eta_{a,b} = q \cdot h$ for a large prime q , we work in the subgroup $\mathbb{G}_q$ of order q , and h is the *cofactor*

Elliptic Curves

Given $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ in $E_{a,b}(\mathbb{F}_p)$, the sum $P + Q$ is computed as:

1. If $P = \mathcal{O}$, output Q . If $Q = \mathcal{O}$, output P .
2. If $x_1 = x_2$ and $y_2 = -y_1$, then output $\mathcal{O}$.
3. Otherwise, let $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = -y_1 - \lambda \cdot (x_3 - x_1)$, where

$$\lambda = \begin{cases} \frac{3x_1^2 + a}{2y_1} & \text{if } P = Q \\ \frac{y_1 - y_2}{x_1 - x_2} & \text{otherwise,} \end{cases}$$

and output $R = (x_3, y_3)$.

Note the branches: the formula you run depends on the inputs, and that is already a potential side channel. It is also why implementations must check that a point is on the curve and in the right subgroup.

Chinese Remainder Theorem

Let $n = p \cdot q$ for distinct primes p and q . Then the map

$$\mathbb{Z}_n \longrightarrow \mathbb{Z}_p \times \mathbb{Z}_q, \quad x \longmapsto (x \bmod p, x \bmod q)$$

is a bijection, and it respects both addition and multiplication.

Given (x_p, x_q) we recover

$$x = x_q + q \cdot (u \cdot (x_p - x_q) \bmod p), \quad u = q^{-1} \bmod p.$$

So we can compute in the two small components instead of the big one.

This generalises to any pairwise coprime moduli $n_1, \dots, n_k$.

Chinese Remainder Theorem

Why we care in this course

RSA signing and decryption are about four times faster with the CRT: compute $m_p = c^d \bmod p$ and $m_q = c^d \bmod q$, then recombine. Almost every real RSA implementation does this.

It is also a rich source of failures. If a fault hits only one of the two branches, a single faulty signature factors n , and if the recombination step can be tampered with, the result leaks information about the primes.

You will meet both the speed-up and the attacks in the assignments.

What We Assume You Know

We build directly on TTM4135 Applied Cryptography and Network Security, and will use the following without reintroducing it:

- ▶ Modular arithmetic and fields
- ▶ Euclid, CRT, Fermat and Euler
- ▶ Block ciphers: DES and AES
- ▶ Modes of operation and padding
- ▶ Stream ciphers
- ▶ Hash functions and MACs
- ▶ RSA and discrete-log cryptosystems
- ▶ Diffie–Hellman key exchange
- ▶ Digital signatures and certificates
- ▶ Key establishment and PKI
- ▶ TLS 1.2 and 1.3



Where to Read Up

If any of this is rusty, the TTM4135 slides and exercise sheets are public at <https://ttm4135.iik.ntnu.no> and are the fastest way to refresh it.

For a proper treatment, these are excellent and cover everything above:

- ▶ Katz and Lindell, *Introduction to Modern Cryptography*
- ▶ Boneh and Shoup, *A Graduate Course in Applied Cryptography*, free at <https://toc.cryptobook.us>

Ask in the forum if something is unclear. You are not expected to know everything on day one, but you should know where to look it up.

Questions?