Course program and reading list

Semester 1 Year 2024

School: Efi Arazi School of Computer Science M.Sc.

Cryptography

Lecturer:

Prof. Tal Moran talm@runi.ac.il

Teaching Assistant:

Mr. Ziv Huppert Ziv.Huppert@post.runi.ac.il

Course No.:	Course Type :	Weekly Hours :	Credit:
159	Elective	3	3

Course Requirements :	Group Code :	Language:
Final Paper	241015901	Hebrew

Prerequisites

Prerequisite:

52 - Calculus I
53 - Calculus II
54 - Linear Algebra I
55 - Linear Algebra II
56 - Discrete Mathematics
59 - Data Structures
69 - Logic And Set Theory
77 - Algorithms
417 - Introduction To Computer Science
643 - Automata And Formal Languages **OR** 3699 - Computational Models

---

Course Description

Cryptography is the science of designing algorithms and protocols that guarantee privacy, authenticity, and integrity of data when parties are communicating or computing in an insecure environment. The recent explosion of electronic communication and commerce has expanded the significance of cryptography far beyond its historical military role into all of our daily lives. For example, cryptography provides the technology that allows you to use your credit card to make on-line purchases without allowing other people on the internet to learn your credit card number.

The past 40 years have also seen cryptography transformed from an ad hoc collection of mysterious tricks into a rigorous science based on firm complexity-theoretic foundations. It is this modern, complexity-theoretic approach to cryptography that will be the focus of this course. Specifically, we will see how cryptographic problems can be given **precise mathematical definitions**. Then we will construct algorithms which **provably** satisfy these definitions, under precisely stated and widely believed **assumptions**. For example, we will see how to prove statements of the flavor "Encryption algorithm X hides all information about the message being transmitted, under the assumption that factoring integers is computationally infeasible."

Topics that we will cover include "classical" cryptographic methods, Shannon's theory of secrecy (and how to get around its limitations using computational assumptions), one-way functions, private-key and public-key encryption, digital signatures, pseudorandom generators. If time permits, we will also cover higher-level protocols such as zero-knowledge and secure computation, electronic cash, and the role of cryptography in network and systems security.

Course Goals

Definitions: Why it is important to precisely define cryptographic problems, and how to do so for several important problems (encryption, authentication, digital signatures, etc.). What are the kinds of subtleties that arise in such definitions, and how to critically evaluate and interpret cryptographic definitions.

Constructions and Proofs of Security: Examples of general and concrete solutions to various cryptographic problems, and how to prove that they satisfy the definitions mentioned above (based on precisely stated assumptions).

Foundations: The assumptions on which modern cryptography is based, and their implications.

Theory vs. Practice: This course will focus on theory, but we will discuss how the theory relates to what is actually done in practice.

Applications: If time permits, we will see one or two examples of how to address cryptographic issues in higher-level protocol problems, such as auctions, voting, or electronic cash.

Security: This is not a course on security, but if time permits, we will discuss how cryptography fits into the broader contexts of network and systems security.

Grading

Homework

Doing the problem sets is for most students the best way to master the course material. We will have between 5 and 7 problem sets throughout the semester. Each problem set has 4-6 questions. The due date will vary depending on the exercise, but it will typically be two weeks after the exercise is given. Sometimes, the last question in a problem set will be a more difficult "bonus" question.

Homework Policy

Students are encouraged to work together to do homework problems. Remember that what is important is a student's eventual understanding of homework problems, and not how that is achieved. In particular, what a student turns in as a homework solution is to be his or her own understanding of how to do the problem. Therefore, in preparing the draft of the homework to be turned in, a student *may not* consult the notes or homework solutions of another student or *any* solutions to homework problems posted on the web. In other words, **you are required to write your homework by yourself**.

The assignment questions have been carefully selected for their pedagogical value and may be similar to questions on problem sets from past offerings of this course or courses at other universities. Using any preexisting solutions from these other sources is **strictly prohibited**.

At the beginning of each submitted problem set students are required to write down a **collaboration statement** stating with whom they have collaborated, and from whom they have received help:

1. "I worked alone and only with course materials," or
2. "I collaborated on this assignment with (*students in class*), got help from (*people other than students and course staff*), and referred to (*citations to sources* ***other*** *than class material*)."

Whether you chose option 1 or 2 {will not affect your grade}. However, no problem set will be given credit unless it has a collaboration statement.

Unclear solutions will not be given full credit even if they are "correct." If you are concerned about how your homework has been graded, feel free to come and talk with me.

Exams and Grades

In addition to the homework sets, the course will have a final exam. Grades for the course will be based on the following weighting:

1. Problem sets: 20%
2. Final exam: 80%

To be eligible for the final exam, you are required to hand in *n-1* homework problem sets, where *n* is the number of problem sets we had in the course.
To pass the course **you must get at least 60/100 in the final exam** (and of course an average of 60/100 overall).

Reading List

The course will closely (but not completely) follow the book *Introduction to Modern Cryptography* by Jonathan Katz and Yehuda Lindell. A more advanced (graduate-level) exposition of the material can be found in Oded Goldreich's *Foundations of Cryptography* (Volumes I and II)}. More application-oriented crypto books are (note that these books take a much less careful approach to definitions and security proofs than we do in the course):

- A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.*
- D. R. Stinson. *Cryptography: Theory and Practice*.
- B. Schneier. *Applied Cryptography*.
- R. Anderson. *Security Engineering*

For those of you who are interested in more material on computational number theory and algebra, I recommend to take a look at the book *A Computational Introduction to Number Theory and Algebra* by Victor Shoup. You can find it online at https://www.shoup.net/ntb/.