CS 427: Cryptography

Instructor: Dr. Mike Rosulek, <rosulekm@eecs.oregonstate.edu>

Meets: MWF 1, in MFD 105

Website: http://eecs.oregonstate.edu/~rosulekm/crypto
Please check often for announcements, homeworks, etc.

Office hours: MW 2-3, or by appointment, in my office (KEC 3063)

Prerequisites: MTH 232, programming fluency (in a language of your choice). CS 321 or 325 helpful but not required.

Textbook: There is no required text. Lecture notes and in-class slides will be posted on the course website. For a good second opinion, you can use the following resources:

- *A Course in Cryptography*, Rafael Pass & abhi shelat (free online).
- *Cryptography, an Introduction*, Nigel Smart (free online).
- *Introduction to Modern Cryptography*, Jonathan Katz & Yehuda Lindell

Software: We'll be carrying out computations on very large ($\sim 2^{1000}$) integers, which requires some special libraries. Examples in class will use `pari/gp`, a free open-source library for number theory & abstract algebra. It's installed on `nome` and `flip`, and can be downloaded from http://pari.math.u-bordeaux.fr.

Course Overview

Historical cryptography was an arms race between the ingenuity of the "good guys" in designing ciphers and the "bad guys" in breaking them. Modern cryptography, however, allows us to *mathematically prove* statements about security. A major theme in this course is to understand how to talk about security in such a *rigorous* way. In this course you should expect to learn how to:

- State and interpret the standard formal definitions for the most common cryptographic security properties (privacy and authentication).
- Formally prove security properties of sound cryptographic constructions, and break the security of unsound ones.
- Choose the appropriate cryptographic primitive for a task (block ciphers, hash functions, MACs, public-key encryption, etc.) while avoiding common pitfalls.

Along the way, you will also learn how the most common cryptographic constructions work.

Assessment

40% **Problem sets:** Expect roughly 6 homework assignments, with a mixture of math, programming, computation, security proofs, attacks, problem solving. Submissions must be typed and submitted on TEACH. **No late homeworks!** If you don't already have my approval to submit a late homework, it's worth a zero.

40% **Exams:** There will be a midterm exam and final, each worth 20%.

20% **Small project:** Find an example of cryptography being used inappropriately or implemented incorrectly, *in a real-world system.* Write a 2-page report on the problem:

- Clearly and concisely describe the problem. What was the intent of the designer? How was the solution implemented? *Why is it wrong?* How could an attacker exploit the system? What could an attacker accomplish by the exploit?
- Was the problem reported responsibly?
- Describe a fix for the problem. Which cryptographic primitives security definitions from class are appropriate for achieve the original intent?

Some good starting places for finding such exploits include: CVE, netsec subreddit, ...

5% **Bounties:** I will assign bonus points for reporting bugs & typos in the course lecture notes. The totals at the end of the quarter will be curved.

Other Policies

Cheating: Academic dishonesty (including plagiarism and cheating) will not be tolerated. Consult the university's student conduct code for more details. I will follow the guidelines given there, and seek out the maximum allowable penalty for violations that occur in this course.

If you have a question about what constitutes academic dishonesty, please ask me.

Disabilities: Accommodations are collaborative efforts between students, faculty, and Disability Access Services (DAS). Students with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 737-4098.