Instructor: Nathan Pflueger (pronounced “fleeger”)
email: npflueger@amherst.edu
office: SMUD 401

Times and locations: Mon, Wed, Fri 1:00-1:50 SMUD 207

Course webpage `http://npflueger.github.io/252/` (I rarely use Moodle)

How to reach me Come to office hours! No appointment is needed. Besides that, I generally reply to email within 24 hours. However, **I may not read or reply to email on weekends, outside business hours, or on Thursdays**, which is the day I devote primarily to research.

Course content

Math 252 is a **mathematics course**, but it has a special twist: you will learn mathematics via both **proofs** and some basic **programming**. I do not assume you have programmed before, and many previous students have told me that learning these basics has been a huge side benefit of this course!

The course has four primary **learning goals:**

- You will write simple versions of the major **public-key cryptographic algorithms** used in practice, and get your hands dirty experimenting with some variations on them as well as some attacks against them. You will also be able to explain the **underlying mathematics** and prove basic properties about these systems and attacks on them.
- Serve as a “**programming for math students**” course, using Python. We will use Python to solve mathematical problems and try experiments. **No prior programming experience is necessary or expected!**
- Provide a bridge from students’ first exposure to proofs to the more abstract and sophisticated 300-level courses, especially Math 350 (Abstract Algebra).
- Most importantly, you will gain confidence and strength in **problem-solving.** Mathematics is a versatile and useful discipline largely because it trains you to problem-solve in novel situations, rely on your own ideas, and place faith in the soundness of your own reasoning. I will often challenge you to adapt ideas from class to novel situations to train these skills.

Course topics: Sections refer to the course textbook. Number of weeks is approximate.

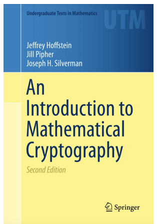
1. (3 weeks) Modular arithmetic and algorithmic aspects of it (§1.2-5)
2. (3 weeks) Discrete logarithms and Diffie-Hellman key exchange. (§2.2-9)
3. (2 weeks) Integer factorization and the RSA cryptosystem. (§3.1-5)
4. (1 week) Digital signatures based on both factoring and discrete logarithms. (§4)
5. (2 weeks) Elliptic curves their use in discrete-logarithm-based algorithms (§6.1-4).
6. (2 weeks) The NTru lattice-based cryptosystem (excerpts from §7)

Throughout the course, we will discuss both *cryptography* (encryption and signing algorithms) and *cryptanalysis* (algorithms to break cryptosystems). Strictly speaking, the course ought to be called *Cryptology*, which refers to both crytography and cryptanalysis.

We will meet examples of the main characters of abstract algebra, “groups,” “rings,” and “fields,” but you are not expected to know anything about these, and we will not examine the general theory of them. Take Math 350 if you want to know more!

Caution

Remember **real-world cryptography applications are only as secure as their weakest link, which is usually not the mathematics**! You are encouraged to consider the Computer Science department’s courses on cryptography and security to see other aspects of the subject, including the many practical aspects you must consider before applying these tools to sensitive data.



Undergraduate Texts in Mathematics
UTM
Jeffrey Hoffstein
Jill Pipher
Joseph H. Silverman
An Introduction to Mathematical Cryptography
Second Edition
Springer

Prerequisites A course with proofs, such as Math 220 or 271 (but not necessarily either of these), or instructor permission. Prior programming experience is **not** required, nor is prior coursework in abstract algebra.

Textbook *An Introduction to Mathematical Cryptography,* ***Second Edition***, by Hoffstein, Pipher, and Silverman. In addition to the paper text, you can download the entire book for free on Springerlink. If you are off-campus, you should first look up “Springerlink” (one word) in the Amherst College library catalog, and follow the link in the catalog entry, to get full access.

Course structure

Grading: Grades are based on the following categories. The exact cutoffs for each letter grade are not set in advance; I calibrate them at the end based the difficulty and score distribution of the exams. There is no set curve, but typically the median grade is around B+.

Written homework	10%	both usually due Wednesdays at 10pm
Programming homework	10%	
Midterm 1	20%	Friday, October 18, in class
Midterm 2	20%	Friday, November 22, in class
Final exam	30%	Date/time to be set by registrar (will be three hours)
Your best exam	10%	(midterm or final; added to its original weight)

Before setting the final grade cutoffs, I anonymize my grading spreadsheet and remove students with substantial preparation beyond the prerequisites, e.g. students who have taken Math 350 or prior programming courses. In practice, though, I have never observed a significant difference in grade distribution between students with extra preparation and those without it.

Expectations You should expect to spend at least eight hours studying and working on problem sets outside of class each week. Of that time, I recommend that you spend at least two hours reviewing your notes, the textbook, and previous assignments. Distributing your practice and review throughout the semester will be much more effective than concentrating your review and studying right before exams or due dates. You are expected to attend class every day, arrive on time, and be respectful. You are expected to know about any announcement I make in class or by email.

I encourage you to **stop me to ask questions**. Active participation helps but your brain in the mode that will make new connections and learn well. If you are feeling lost, there is almost certainly someone else feeling the same thing; asking a question may help many of your classmates as well! If you are new to programming, working on that skill will account for a lot of your time (but it will pay very high dividends, as my previous students repeatedly tell me!). Some assigned problems will be quite challenging, and **you do not need to complete all problems to earn a good grade in the course**. However, I recommend attempting all assigned problems.

You will sometimes be expected to take the initiative in looking up information about the tools you need, especially when programming. I will of course be happy to help, as will our Math Fellow.

The nature of a course like this, which connects to several different disciplines, is that some students may have an easier time with the material than others (e.g. if they already know how to program, or already know some abstract algebra). If it seems like the course is easier for some of your classmates, do not be discouraged! You are in good company, and this just means that you have a lot to gain from the course. I hope that by the end of the course all students will reach the same place, and you will all be versatile and valuable problem-solvers wherever you take your skills next.

Course policies

Dropped assignments To compensate for illness and other emergencies, your **lowest two homework scores will be dropped**. If you cannot make a due date due to an emergency, my advice is to skip the assignment, but study and understand the problems when you have time, and focus on keeping up with the new material in the course. You do not need to apologize or provide any reasons for skipping an assignment or turning it in unfinished; please choose what is best for your time, health, and well-being. Remember that **the primary purpose of the homework is not evaluation, but to help you learn the material and guide your studying**, so you should still work through all problems on any assignment your drop, and ask me about them as needed.

Homework deadlines and late policy Homework will be **due at 10pm**, typically on Wednesdays, via Gradescope. To allow for technical difficulties or other last-minute issues, Gradescope will allow you to submit homework after the deadline, however your score will be reduced by 2% **per hour** after the deadline (scaled continuously, e.g. being fifteen minutes late results in a 0.5% deduction). Please try to turn in your work by the due time (I don't want to be responsible for lost sleep!), but don't worry about short delays. **I generally do not grant extensions**, but instead drop two assignments (see above).

Missed exams The midterm dates are listed above. **Put them on your calendar now**. Exam dates/times are fixed and may not be rescheduled except in the case of an extenuating circumstance (illness, emergency, religious conflict, etc.) with a note from a health professional or dean. In such a case, please let me know at the start of the semester, or as soon as possible. Other than by reason of a valid extenuating circumstance, a missed exam will be counted as 0. In valid extenuating circumstances, I will often simply excuse the exam and count your final exams grade in its place. This is because I usually release exam solutions soon after the exam.

The final exam date is set by the registrar, and should be available on the registrar's website partway through the term. **Do not schedule travel before the end of exam week until the final exam date has been determined by the registrar.**

Accommodations I strive to make this course welcoming to all students. If you would like to discuss your learning needs with me, please schedule a meeting so that we can work together to support your academic success. Anyone who may require an accommodation based on the impact of a disability should contact me to make arrangements. I rely on Accessibility Services for assistance in verifying the need for accommodations and developing accommodation strategies, so you should contact them at `accessibility@amherst.edu` or 413-542-2337. If you require accommodations on exams, please arrange this with me at least one week in advance.

Intellectual responsibility

- **Homework:** Mathematics is a collaborative subject; open and generous communication is one of its core values. Therefore you are strongly encouraged to work with other students, ask many questions, and learn from as many people as possible. However, you must write up the solution yourself. **All your submitted work must be your work, written in your own words**. Copying solutions from other students, solutions manuals, online databases, or generative AI is plagiarism; such copying will result in a 0 on the assignment and will be reported to Community Standards. You are also expected to **list each person your worked with** on the front of your homework assignment.

- **Exams:** You will be allowed **one page of notes (front and back)** for each exam. No calculators or other aids are permitted. Cell phones should be stowed out of sight during exams. Use of cell phones or other devices during the exams will be grounds to receive a 0 on the exam. You are bound by the college's honor code, and all work must be entirely your own on exams.

For homework and exams, I reserve the right to give no credit for any work that appears suspicious.

Tips and resources

Come to office hours! I am happy to answer your questions and also talk about the course in general. Even if you don't have specific questions, you can come to review material, listen to other students' questions, or just to chat. There is a desk in my office and several just outside where you are welcome to work, chat, and listen in. Office hours are the best way I have to learn about you and how you're doing in the course and the college, so please visit!

Focus on practice and improvement. Every homework problem, or example and class or the book is an opportunity to practice. Take these opportunities, and make the most of them!

Distribute your practice. Study a bit every day, not just before exams. Treat every homework problem as a chance to practice and study.

Actively seek opportunities to practice. Ask me questions, ask classmates questions, read examples in the book, and try problems that haven't been assigned.