COMPUTER SCIENCE DEPARTMENT
MET CS 789 CRYPTOGRAPHY

Course Overview

The course covers the main concepts and principles of cryptography with an emphasis put on public key cryptography. It begins with the review of integers and a thorough coverage of the fundamentals of finite group theory followed by the RSA ElGamal encryption, Diffie-Hellman key exchange, ElGamal Signatures, and the Digtial Signature Algorithm (DSA). Primitive roots in cyclic groups and the discrete log problem are discussed. Baby-step Giant-step and the Index Calculus probabilistic algorithms to compute discrete logs in cyclic groups are presented. Naor–Reingold and Blum–Blum–Shub Random Number Generators as well as Fermat, Euler and Miller-Rabin primality tests are thoroughly covered. Pollard's Rho, Pollard's p-1 factorization algorithms are presented. The course also covers oblivious transfer protocols and zero-knowledge proofs. There are numerous programming assignments, which are used in the final project to demonstrate the algorithms implemented by the students.

In addition to the specific topics detailed in the schedule of classes, below, practical application of the algorithms, attacks, and weaknesses are discussed throughout the class. Topics touched on include the differences between symmetric and public key cryptography, key distribution centers, digital certificates and signing, properties of secure hashes, public key infrastructure vs. web of trust, man-in-the-middle attacks, practical attacks on Diffie-Hellman Key Exchange, perfect forward secrecy, TLS/SSL authentication vs. key exchange, homomorphic encryption, and hardware random number generators.

Prerequisites

MET CS 248 Discrete Mathematics and MET CS 566 Analysis of Algorithms

Learning Objectives

By the end of this course, the student will have learned

1. Concepts of symmetric and public key cryptography;
2. The RSA and ElGamal asymmetric ciphers as well as the Diffie-Hellman Key Exchange Protocol and the Key Management Systems;
3. Algorithms to compute the Discrete Logarithm in cyclic groups, the Baby-step Giant-step Algorithm and the Index Calculus Algorithm;
4. Oblivious Transfer Protocols, Zero Knowledge Proof protocols, and ElGamal and the Digital Signature Algorithm;
5. Blum-Blum-Shub and Naor-Reingold pseudorandom number generators;
6. Probabilistic algorithms to check the primality of large numbers;
7. Factorization attacks including Pollard's Rho Method, and Pollard's p-1 Method.

Required Textbook

There is no required textbook for this course. The course notes should be sufficient for all the subjects covered. In the past, *Making, Breaking Codes: An Introduction to Cryptology* by Paul Garrett has been used, but unfortunately this textbook is now difficult to get and so won't be used, or required. There just aren't any other textbooks that cover all the materials for the course. However, there are some textbooks that you might find useful. They're optional.

Recommended Reference Textbooks—In Order of Preference

Garrett, Paul—*Making, Breaking Codes: An Introduction to Cryptology, 2nd Edition*
Prentice Hall, ISBN-10: 0-13-186146-8
Amazon: https://www.amazon.com/dp/B01FEKZJOC/ref=cm_sw_em_r_mt_dp_slgtFb2XW8ZHK

It's often ridiculously expensive new, if you can find it, but if you can get Garrett's book, it is quite good at explaining the basic mathematics covered in this course, but it isn't necessary. There are a lot of mistakes in the text, though, so be sure to check out the Errata[1].

Smart, Nigel—*Cryptography: An Introduction*
McGraw-Hill College, ISBN-13: 978-0077099879
Amazon: https://www.amazon.com/dp/0077099877/ref=cm_sw_em_r_mt_dp_EigtFbSE7YVY1

I quite like this book and it does a better job of explaining cryptology and topics that Garrett doesn't cover, but it doesn't provide the stronger mathematical treatment that you'll get from Garrett.

Other Resources

Forouzan, Behrouz—*Cryptography and Network Security*,
McGraw Hill, ISBN-13 978-0-0-7332753-2
Amazon: https://www.amazon.com/dp/0073327530/ref=cm_sw_em_r_mt_dp_gIgtFb6NP8JGY

This book is quite good and is the main text for MET CS 799, Advanced Cryptography. I find it superior in almost every way to the following Stallings text.

Stallings, William—*Cryptography and Network Security. Principles and Practice, 7th Edition*
Prentice Hall, ISBN-10: 0-13-444428-0
Amazon: https://www.amazon.com/dp/0134444280/ref=cm_sw_em_r_mt_dp_8VgtFbSX7ZVGZ

The Stallings book is one of the most popular cryptology textbooks, but it frankly isn't very good helping the student understand the mathematical underpinnings of cryptology. It is, however, useful as a reference work because it covers a large number of topics.

Evaluation and Grading

There will be a midterm exam and a final project. If any grading event will be missed, it is the responsibility of the student to arrange a mutually agreeable schedule for completion.

Class Participation	20%
Midterm	50%
Final Project	30%

Letter Grade	Numerical Range
A	95-100%
A-	90-94%
B+	87-89%
B	83-86%
B-	80-82%

[1] http://www-users.math.umn.edu/~garrett/crypto/Errata2.html

C+	77-79%
C	73-76%
C-	70-72%
D	60-69%
F	<60%

What does "Class Participation" mean? It is attending the lectures, asking questions, answering questions, and commenting on topics in the lecture, where appropriate. The lecture notes and the homework problems will be made available prior to class. **Please print the lecture notes and bring them with you to class, or otherwise have them available, so you can focus on the lecture and not on taking free form notes.** The Midterm is open book and open notes; you can even use the algorithms you developed in the course, but you must not browse the web or use any materials you haven't carried in with you.

The Final Project will be described in more detail later, but it entails creating a set of programs that allows you to encrypt, decrypt, and as an attacker, break encryption. All the programs you need to complete the Final Project are in the homework assignments. You may use any program language you like, but I strongly recommend a programming language with infinite magnitude integers that are unbounded in size, since the size of the integers you'll be working with will exceed the size of a single computer word, in most cases—Python is a good choice.

Note that **homework is not graded** and is used to challenge the student and help them grasp whether they have mastered the materials. It is extremely important to note that the programming assignments associated with the homework makes up the bulk of the final project. **If you don't do the homework assignments you won't be prepared for either the midterm or the final project, so do the homework assignments when assigned.** You don't want to wait until the end of the semester and have to rush and get your programming done all at once.

Many students find this class challenging. I very **strongly** encourage students to keep current with the homework, programming assignments, and form small study groups in order help each other to master the material.

Academic Honesty

The course is governed by the Academic Conduct Committee policies regarding plagiarism (any attempt to represent the work of another person as one's own). This includes copying (even with modifications) of a program or segment of code. You can discuss general ideas with other people, but the work you submit must be your own. Cheating and plagiarism will not be tolerated in any Metropolitan College course. Cheating will result in no credit for the assignment or examination and may lead to disciplinary actions. Please take the time to review the Student Academic Conduct Code[2]:

This should not be understood as a discouragement for discussing the material or your particular approach to a problem with other students in the class. On the contrary, you should share your thoughts, questions and solutions. Naturally, if you choose to work in a group, which is encouraged, you will be expected to come up with more than one and highly original solution rather than make the same mistakes. Your code should be your own.

[2] http://www.bu.edu/academics/policies/academic-conduct-code/

Class Location

The class will be at in the College of Arts and Sciences (CAS), Room B06A, 685–725 Commonwealth Avenue, at 6pm on Mondays. This is not a "blended" class. Unless you have talked to me ahead of time, I expect you to attend the lectures in person. Technically, the class runs to 845pm, however, I'll stay for questions afterwards and sometimes the class may run until 9pm to cover more material, so arrange your schedules accordingly.

Schedule of Classes

09/09 Integers—Divisibility, Unique Factorization, Euclidean Algorithm, Multiplicative Inverses, Equivalence Relations, and Modular Arithmetic

09/16 Groups—Definition of Groups and Subgroups, Lagrange's Theorem, Index of a Subgroup, Cyclic Subgroups, and Euler's Theorem

09/23 Exponentiation Algorithm, Fields, Primitive Roots, Discrete Logs, ElGamal Cipher, and Diffie-Hellman Key Exchange

09/30 Primitive Root Search Algorithm, Baby-Step Giant-Step Algorithm, The Index Calculus Algorithm, and Public-Key Ciphers

10/07 RSA Public Key Encryption

10/14 **NO CLASS**—Columbus Day—**Class on Tuesday**

10/15 RSA Public Key Encryption

10/21 The Chinese Remainder Theorem, Euler Criterion, and Roots Mod Composites

10/28 **Midterm Exam**

11/04 The Oblivious Transfer Protocol (factorization and discrete log), Zero Knowledge Proofs, ElGamal Signatures, and the Digital Signature Algorithm

11/11 Quadratic Reciprocity and Pseudoprimes

11/18 Pseudorandom Numbers, Fermat, Euler, and Miller-Rabin, Pseudo and Probabilistic Primes, and the Miller-Rabin Test

11/25 Random Number Generators—Linear Congruent Generator, Feedback Shift Generator, Naor-Reingold Number Generator, Blum-Blum-Shub Random Number Generator

12/02 Factorization Attacks—Pollard's Rho Method, Pollard's p-1 Method

12/09 No Class or Possible Makeup—Additional time to work on final project

12/16 **Final Project**

Instructor Information

Geoffrey Pascoe
Computer Science Department, Metropolitan College,
1010 Commonwealth Avenue, 3rd Floor
Boston, MA 02215
Cell: 603-866-1067
Email: gpascoe@bu.edu
Office hours by appointment via Zoom or Google Meeting