MET CS 789 Cryptography

Course Overview

This course covers the main concepts and principles of Cryptography with the main emphasis on Public Key Cryptography. It begins with the review of integers and a thorough coverage of the Group Theory fundamentals followed by the RSA and ElGamal ciphers, Oblivious Transfer Protocols, Zero Knowledge Proofs, Cryptographically Secure Hash Functions, Digital Signatures, Message Authentication, Integrity, Confidentiality, Nonrepudiation, Distribution of Secret session Keys, Pseudorandom Numbers and Random Number Generators along with various factorization attacks will also be covered. There will be programming assignments to code the Euclidean Algorithm, the Fast Exponentiation Algorithm, the Primitive Root Search Algorithm, the Baby-step Giant-step Algorithm, the Index Calculus Algorithm, the Miller- Rabin Test, the Noar-Reingold Random Number Generator, the Blum-Blum-Shub Random Number Generator and the Pollard's p-1 method.

Prerequisites

MET CS 248 Discrete Mathematics and CS 566 Analysis of Algorithms

Learning Objectives

By the end of this course, the student will learn:

1. Asymmetric ciphers, the RSA and ElGamal as well as Diffie-Helman key exchange protocol and key management system
2. Algorithms to compute the Discrete Logarithm in cyclic groups, the Baby-step Giant-step Algorithm and the Index Calculus Algorithm.
3. Oblivious transfer protocols
4. Cryptographically secure hash functions
5. Digital signatures
6. Message authentication, integrity, confidentiality, nonrepudiation
7. Key management
8. Various random number generators
9. Probabilistic algorithms to check the primality of numbers
10. Factorization attacks: Pollard's Rho Method, Pollard's p-1 Method, Dixon Algorithm, Non-Sieving Quadratic Sieve

Method of Instruction

This is a lecture-based course with several programming assignments

Evaluation and Grading

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

Grades will be based on:

- Class participation 10%
- Midterm 50%
- Final Project 40%

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 students, but the work you submit must be your own.

Instructor Information

Anatoly Temkin, Ph.D.
Department of Computer Science
Boston University Metropolitan College
1010 Commonwealth Avenue, 3d floor
Boston, MA 02215
Office: 617-353-2566
Cell: 617-953-8378
Email: **temkin@bu.edu**

Office Hours: Monday 5-6pm

Classes are scheduled at CGS, Room 515

Schedule

Date	Topic	Reference
9/8	Integers (Divisibility, Unique Factorization, Euclidean Algorithm, Multiplicative Inverses, Equivalence Relations, Integers $\bmod n$)	Chapter 7
9/15	Groups (Definition of Groups and Subgroups, Lagrange's Theorem, Index of a Subgroup, Cyclic Subgroups, Euler's Theorem)	Chapter 17
9/22	Fields, Generators in Groups, ElGamal Cipher, Exponentiation Algorithm	Chapters 22, 27, 28
9/29	The Diffie-Helman Key Exchange Protocol, Primitive Root Search Algorithm, Baby-step Giant-step Algorithm, The Index Calculus Algorithm	Chapters 10, 27

10/6	Communication in Networks, Key Management, Electronic Key Management System, The RSA Cipher	Chapter 10
10/14	**Substitute Monday Schedule of Classes**	
10/20	Chinese Remainder Theorem, n-th roots, Euler Criterion, Principal Square Roots	Chapter 13
10/27	Oblivious Transfer Protocol (Factorization and Discrete Log Based), The Digital Signature Algorithm, Zero Knowledge Proofs	Chapter 18
11/3	Cryptographically Secure Hash Functions, Digital Signatures, RSA, ElGamal digital signatures protocols, Schnorr digital signature algorithm, Blind digital signature	
11/10	**Midterm Exam**	
11/17	Message Authentication, Integrity, Confidentiality, Non-repudiation, Symmetric Key Distribution by the Key Distribution Center.	
11/24	Pseudorandom Numbers, Fermat, Euler and Strong Pseudoprime numbers, Solovay-Strassen and Miller-Rabin Tests, Blum-Blum-Shub and Naor-Reingold Random Number Generators.	Chapter 16
12/1	Factorization Attacks (Pollard's Rho Method, Pollard's $p-1$ Method, Dixon Algorithm, Non-Sieving Quadratic Sieve)	Chapters 24, 25
12/8	Course review	
12/15	**Final Project**	

Required Book

Paul Garrett: Making, Breaking Codes: An Introduction to Cryptology, Prentice Hall, ISBN#:0-13-030369-0

Recommended Book

Behrouz Forouzan: Cryptography and Network Security, McGraw Hill,
ISBN#: 978-0-07-287022-0

Homework assignments

p.111, # 7,8,9,11,14,16
p.118, # 1,2,3,4,5,6
p.121, # 1,8,9
p.123, # 1 to 10; p.126, # 1
p.135, #1,2,3,4,9,10,14,15
Write a C++, Java, or Python code for the Euclidean Algorithm
Write a C++, Java, or Python code that finds two integers, x and y, for given integers m and n, such that $xm + yn$ yields the smallest positive integer.
p.267, #1,3,4,5,6
p.268, #3,4,5,6,7
p.271, #2,3
p.275, #1,2,10

12.5.01, 12.5.06
Write a C++, Java, or Python code for the Exponentiation Algorithm

Write a C++, Java, or Python code for a Primitive Root Search Algorithm

Write a C++, Java, or Python code for a Baby-step Giant-step Algorithm

Have an example of the Diffie-Hellman Key Exchange Protocol, assuming it takes place in $Z_p^{\times}$,

where p = 9511

10.2.02, 10.2.03, 10.2.06,10.2.08

13.1.01, 13.2.02, 13.2.03, 13.3.01, 13.3.07, 13.8.01, 12.6.01, 12.6.07, 12.6.03, 12.7.01, 12.7.02, 12.7.03 and an additional exercise: Solve $x^2 \equiv -1 \bmod 13 \cdot 17 \cdot 29$

Have an example of the Oblivious Transfer Protocol (factorization based), where $p = 31$ and $q = 103$
Have an example of the Oblivious Transfer Protocol (discrete log based), where $p = 103$

16.2.01, 16.6.01, 16.6.02

Write a C++, Java, or Python code for the Miller- Rabin Test

P. 335, #21.3.02, 21.3.03, 21.3.04;

P. 336, # 21.4.01, 21.4.03

Write a C++, Java, or Python code for the Noar-Reingold Random Number Generator

Write a C++, Java, or Python code for the Blum-Blum-Shub Random Number Generator

24.1.01, 24.1.02, 24.1.03
24.2.02, 24.2.03

Write a C++, Java, or Python code for the Pollard's $p-1$ method