Coding and Cryptography Fall 2016
Topics

Ciphers

For each of these, you should be able to encipher or decipher a short message by hand, if given a key, and be able to implement these ciphers on computer. You should be able to describe the size of the keyspace. We will cover a variety of ciphers for their historical and mathematical importance, including some or all of the following, or others:

1. Caesar cipher
2. Scytale
3. Substitution cipher
4. Vigenere cipher
5. Affine cipher
6. Hill cipher
7. One-time pad
8. Enigma
9. RSA
10. Three-pass protocol
11. ElGamal using Discrete Logs
12. Elliptic Curve Encryption

Symmetric vs. Public-key cryptography

You'll learn the historical importance and mathematical meaning of this dichotomoy. We'll see many examples of each.

Number Theory Topics

For each of these, you should know the appropriate definitions or statements and be able to work with them in a novel situation (check if something

satisfies a definition or hypothesis, give examples and non-examples, prove a small statement or consequence), and be able to do standard computations by hand efficiently.

1. Modular arithmetic
2. Divisibility
3. GCD
4. Solving $ax + by = d$
5. Finding an inverse modulo n
6. Efficient arithmetic modulo n
7. Chinese Remainder Theorem
8. Fermat's Theorem
9. Euler's Theorem
10. Inverting Matrices Mod n
11. You'll be expected to do small novel proofs using the basic definitions
12. Primitive roots
13. Discrete logs
14. How to find square roots mod p if p is 3 modulo 4
15. How to use Chinese Remainder Theorem to find roots modulo a composite number by looking at its prime factors
16. How to test if an element is a square by taking $(p-1)/2$ power
17. How to use 4 square roots of one value to find a nontrivial factor of the modulus
18. Legendre and Jacobi symbols and their efficient computation
19. Elliptic curves

Cryptographic 'Hard Problems'

You'll be able to state each of these hard problems precisely, and explain what is the significance of a 'hard problem'.

1. Factoring
2. Discrete Logarithm
3. Elliptic Curve Discrete Logarithm
4. Lattice problems

Cryptanalysis and related topics

Cryptanalysis is the art of breaking ciphers. We'll learn various methods.

1. The method of frequency analysis as used against a Caesar cipher or substitution cipher

2. Cryptanalysis of the vigenere cipher
3. Cryptanalysis of affine and hill, as done on your homework.
4. Cryptanalysis of Enigma
5. RSA: Be able to factor n if you know $\phi(n)$, and be able to compute $\phi(n)$ if you know the factorisation of n.
6. RSA: Be able to use the previous bullet to decipher a message you don't have the private key for, if you know $\phi(n)$ or the factorisation of n.
7. Primality testing: 4 square roots of an element mod n
8. Primality testing: Fermat test
9. Primality testing: Miller-Rabin
10. Primality testing: Solovay-Strassen
11. Which primes are safe to use in RSA? Special attacks.
12. Factoring: Fermat factorization
13. Factoring: $p-1$ method
14. Factoring: Quadratic Sieve
15. RSA Attack: short plaintext and padding
16. RSA Attack: Timing attacks and double-and-add
17. Baby-Step Giant-Step for computing a Discrete Log
18. Birthday Attack

Protocols for Cryptography and Coding

Protocols are ways of using cryptographic tools to accomplish particular goals. You should be able to describe and implement some basic protocols, such as

1. Sending a secret message with ciphers above
2. Treaty Verification
3. Diffie-Hellman Key Exchange
4. Digital signatures
5. Bit committment
6. Quantum key distribution

Error correcting codes

Error correcting codes are not cryptography per se, as the encoding and decoding is doable by everyone. Here are some codes you may be expected to know how to implement:

1. Repetition codes

2. Parity check codes
3. Hamming codes
4. New vocabulary: alphabet, binary code, ternary code, q-ary code, length of a code, codeword, block codes, Hamming distance, Hamming weight, minimum distance, nearest neighbour decoding, code rate (AKA information rate), linear code, information symbols, check symbols, syndrome, coset leader, coset, metric, code parameters: (n, M, d) and $[n, k]$ or $[n, k, d]$.
5. determine from minimum distance how many errors can be corrected and detected
6. Know what it means for two codes to be equivalent
7. Basics of (mostly prime) finite fields
8. Basics of vectors spaces, subspace, dimension, basis.
9. Compute the size of a vector space or subspace based on its dimension and the size of the finite field.
10. Linear codes: generating matrix, parity check matrix
11. Decoding a linear code using the syndrome and parity check matrix

Computer Science topics

Along the way, we'll study some basic computer science topics. These include:

1. Pseudorandom number generators
2. time complexity of algorithms such as gcd and successive squaring
3. DES and Rijndael
4. Modes of operation for block ciphers: electronic codebook and cipher block chaining
5. Basics of how passwords are handled by a server
6. The use of chinese remainder theorem to make computations efficient on a computer; the use of chinese remainder theorem in radar.
7. Runtimes (exponential, subexponential, polynomial)
8. Key distribution and certifying trusted authorities
9. Bitcoin

Quantum and Post-Quantum Cryptography

This is a trip: quantum computers are coming and will break our current cryptography. We'll learn how, and how to save ourselves from it.

1. Definition of a qubit

2. Polarization experiment
3. Quantum Key Distribution
4. Discrete Fourier Transform
5. Schor's Algorithm
6. Lattice-based cryptography