**Introduction to Cryptography with Coding Theory**

Chapter 1: Overview of Cryptography and Its Applications

* **Cryptology**: study of communication over nonsecure channels.
* **Cryptography**: process of designing cryptologic systems.
* **Cryptanalysis**: breaking cryptographic systems.
* **Coding theory**
* Representing input information symbols by output symbols called code symbols
* Covers compression, secrecy, error correction
* Possible attacks
* Cyphertext only
* Known plaintext (cribs)
* Chosen plaintext: temporary access to encryption machine to deduce key. Eve can put plaintext into system and get ciphertext (can encrypt)
* Chosen cyphertext: access to decryption machine to deduce key. Eve can derive plaintext from ciphertext (can decrypt)
* **Kerckhoff's principle:** one should always assume the enemy knows the encryption method being used when assessing the security of a cryptosystem.
* Symmetric and Public Key algorithms
* **Symmetric Key**. Encryption and decryption keys both known to both parties.
* **Steam ciphers.** Data fed into algorithm in small pieces (bits, chars) and output is returned in small pieces.
* **Block ciphers.** Block of input bits collected and all at once and output as block.
* **Public Key.** Encryption key is public, but decryption key infeasible to find without information known only to receiving party. RSA is the most popular implementation of this.
* Public key encryption more computationally expensive, and generally not used to encrypt large quantities of data.
* **Codes:** words and letter combinations are replaced by code words.
* **Cipher:** encrypts every string of characters via algorithm.
* Factorization takes significantly longer than multiplication.

Chapter 2: Classical Cryptosystems.

* **Shift cipher (Caesar cipher).** x --> x + k, where k = the key.
* **Affine cipher.** Shift cipher variation. x --> ax + b, where gcd(a, 26) = 1.
* **Vigenère cipher.** Shift cipher variation. Key is a vector representative of a word of length n. For instance, the key k = vector = (21,4,2,). Encrypt by shifting first letter of plaintext by k[0], second letter by k[1], etc. After k[n-1] go back to k[0].
* **Substitution cipher.** Permuation of the alphabet is chosen and applied to plaintext.
* Examples include shift and affine ciphers
* Can be cracked with frequency counts
* **Block ciphers**. Encrypt blocks of several characters simultaneously. A change of one plaintext character could the entire ciphertext block.
* **Playfair cipher**. Key is a word, repeated letters removed, remaining used to populate 5x5 matrix (with i and j as one). Divide plaintext into groups of 2, padding with an x if odd.
* If 2 letters not in same row or column, replace each letter by the letter that is in its row and is in the column of the other letter.
* If 2 letters are in same row, replace each letter with letter immediately to the right of itself.
* If 2 letters are in same column, replace each letter with letter immediatlely below itself.
* **Playfair** cipher example of block cipher--takes two-character blocks
* **ADFGX cipher**.
* Starts with letters of alphabet in random 5x5 matrix (i and j as one).
* Rows and columns of matrix are labeled ADFGX from origin
* Each plaintext letter replaced by the labels of its row and column.
* Then choose a keyword, and label columns of matrix by letters of keyword and put result of initial step into another matrix.
* Now reorder the columns alphabetically and get the ciphertext by reading down the columns, left to right.
* **Hill cipher**.
* Choose integer n, and key is n by n matrix M whose entries are integers mod26
* Message is written as series of row vectors (size n) multiplied by key mod26
* **Electronic Codebook (ECB) Mode**. Convert plaintext to ciphertext block by block.
* **Diffusion**. A change in one character in the plaintext/ciphertext should correspond to multiple character changes in the other.
* **Confusion**. Key does not relate in a simple way to the ciphertext, i.e. each character of the cipher text should depend on several parts of the key.
* A **Linear Congruential Generator** produces a sequence of numbers x1, x2, ..., where:
* xsubn = a\*xsub(n-1) + b (mod m

Lectures

* If a + b = 0 (mod 26), then we say that b is a's additive inverse, e.g. the additive inverse of 5 is 21 (mod 26)
* Multiplicative inverse of a is the value b such that a \* b = 1 mod 26
* One-time pad is a Vigenère cipher with a randomly created key of the same as the plaintext. Key can NEVER be used again also.
* Decimation cipher is a multiplication equivalent to the shift cipher
* Transposition cipher. Permuted plaintext.
* Pick keyword of length n
* Populate n-column matrix with plaintext
* Reorder columns alphabetically by column letter of keyword
* Print column by column left to right for ciphertext
* ADFGVX
* Both transposition and substitution
* Compose 6 by 6 matrix labeled ADFGVX on both rows and columns with matrix randomly populated with 26 digits and 10 numbers
* Substitute plaintext letter for row-column pair
* Transpose ciphertext row by row under keyword (with padding)
* Alphabetize columns of keyword
* Write columns out into line to get ciphertext
* Hill
* Block cipher -- encipher two or more characters as a unit
* Start with matrix M that's square of length n with values mod 26. This is the key. Determinant of key must be coprime with 26
* Transform plaintext into numerical representation
* Break up into n-sized pieces
* Multiply each piece by the key to get new vector of ciphertext (mod 26)
* Diffusion. A cipher has the property of diffusion if changing a single character in the plaintext results in many changes in the ciphertext (at least half (the bits)). Small change diffuses to affect quite a bit of the ciphertext.
* Confusion. A cipher has the property of confusion if changing a single key character results in many changes in the ciphertext (about half (the bits)).
* Primes: infinity amount. Between n and 2n there is always a prime (Bertrand's postulate)
* Fundamental theorem of arithmetic says that every natural number greater than 1 can be uniquely factored into prime numbers. All numbers have only one prime factorization.
* Fermat's little theorem. If p is prime, then for a value a >= 2 and <= p-1, it will be true that a raised to the p-1 power = 1(mod p); however, just because it is true does not necessarily mean that p is prime. But if I choose 10 random as, and all 10 are true for p, then there is only a 1/2^10 chance that p is not prime. Known as random, or monte carlo algorithm.
* Addition in binary is xor (^), and multiplication in binary is logical and (&)
* One-time Pad. plaintext XOR key = ciphertext, and ciphertext XOR key = plaintext.
* If you XOR two one-time pad ciphertexts, you can recover the key by
* Blum-Blum-Shub good, but computationally expensive. LFSR (Linear Feedback Shift Register) is less secure, less random, but much faster.
* A flipflop (FF) is a memory device that holds one bit
* a | b means a divides b, which is true if and only if b % a = 0 (a goes into b)
* Prime number theorem (density): number of primes < x is approximately x/ln(x) ...generally
* Fundamental theorem of arithmetic. Every positive integer can be uniquely factored into prime numbers: no two prime factorizations are alike...they act as fingerprints
* To find a-inverse, mod b: ax + by = 1. by mod b - 0; 1 mod b = 1;

ax + by (mod b) is ax = 1(mod b)

* DES is a symmetric (private) block cipher that encrypts 64-bit (8-byte, 8-character) blocks
* Can obtain non-linearity not only by raising to power, but also by writing functions/operations so that distributive property doesn't hold (DES, for example)
* Equivalent relations have reflexivity, transitivity, and symmetry. They break up all values into five disjoint sets, such that the intersection of all sets is the empty set and the union of all the sets is all of the values. Called equivalence classes.
* Random algorithm: select 2 <= a <= p-1 at random. If a ^ p-1 % p != 1, return false (p is not prime). Else repeat (k times). If you fall out of the loop return true. If it's false, it's 100 percent that p is not prime. If true, there is still a small probability that p is not prime. Probability of being wrong is approximately 1 / 2^k.
* (a\*b)%p = ((a%p) \* (b%p)) % p
* Repeated squaring: to calculate a^b, if a == 0 return 1, else if b % 2 == 0, return ((a^(b/2))^2), else return a \* a^(b-1).
* Euler's phi function. phi(n) = number of values 1 <= a <= n such that gcd(n, a) = 1. E.g. phi(10) = 4 (1, 3, 7, 9). phi(pq) where p and q are both prime is

(p-1)(q-1).

* Euler's theorem. If gcd(a, n) == 1, then a ^ (phi(n) === 1 mod n. If a = 25 and n = 32, then 25 ^ (16) === 1 mod 32.
* Monte Carlo algorithm. Random algorithm with one-sides error (false positives).
* If eve can get two different ciphertext bits (with the same key), then she can do:

b.c1 xor b.c2 = b.pt1 xor b.key xor b.pt2 xor b.key, and this is equal to:

b.p1 xor b.p2 xor b.k xor b.k == b.p1 xor b.p2, which means that I am now doing an encryption on a one-time pad where the key bits are not random

* Non-linearity of s-box. Inputs, x and y, are 6 bits. We have an s-box, S, that outputs S(x) and S(y) that are 4 bits. Non-linearity means that for all x and y, such that x is not equal to y, S(x) xor S(y) != S(s xor y). S(x) + S(y) != S(x + y).