Dylan Silva-Rivas

Cryptography Notes

Contents

# Chapter One: Intro to Cryptography Concepts.

## OSI Security Architecture

### Summary

International Telecommunication union (ITU)

## Security Attacks

### Summary

There are different attack classifications: Passive attacks and active attacks. Passive attack = Attempts to make use of information from the system but not affecting the system. Active attack = Complete opposite of passive attack. Active attacks consist of masquerade, replay, data modification, and denial of service …

## X.800 Security Mechanisms

### Summary

Some mechanisms in x.800 security mechanisms: Encipherment, digital signature, access control, and data integrity. Encipherment = Converting data to something unintelligible. Digital Signature = Data appended to encrypted data allowing recipient to prove data integrity.

## Private-Key Cryptography

### Summary

# Chapter 2 – Introduction to Number Theory

## Division Algorithm

### Summary

* +n and nonnegative integer a, if you divide a by n we get quotient q and an integer remainder
* a=qn+r, 0 <= r <n
  + Ex: 39 mod 45
    - Dividend = 39 = a
    - Modulus = 45 = n
    - Quotient = q
    - Remainder = r
  + a = qn + r 🡺 39 = q(45) + r, q = 0, r = 39

## Euclidean Algorithm (Greatest common divisor)

### Summary

Algorithm: Let’s say we are trying to find the greatest common divisor (GCD) for 60 and 24. Set the largest integer = qn + r, so in this case 🡺 60 = 24(q) + r.

Steps after setting up the problem:

* 60 = 24(q) + r 🡺 60 = 24(2) + 12
* 24 = 12(q) + r 🡺 24 = 12(2) + 0, r = 0 so the algorithm ends and gcd = 12 which is n.

Also remember that gcd(a, b) = gcd(a, -b) = gcd(-a, b) = gcd(-a,- b)

## Congruent Modular Arithmetic

### Summary

Two integers a and b are congruent modulo if a mod n = b mod n, which is written as a b (mod n). Example:

## Modular Exponentiation

### Summary

## Extended Euclidean Algorithm

### Summary

When you have gcd(a, b) you still have to apply the regular Euclidean algorithm then working back up the algorithm. The purpose of this algorithm is to compute the linear combination of form: gcd(a, b) = pa + sb = d

Example:

81 = 1(57) + 24 🡺 3 = 2(81-1(57)) + 9 🡺 3 = 2(24) + 9

57 = 2(24) + 9 🡺 3 = 2(24) + (57 – 2(24)) 🡺 3 = 2(24) + 9

24 = 2(9) + 6 🡺 3 = 9 – 1(24 – 2(9)) 🡺 3 = 9 – 1(6)

9 = 1(6) + 3 🡺 3 = 9 + 1(-6)

6 = 2(3) + 0

57 – 2(24) = 9

## Solving method for extended Euclidean algorithm

Solve 212 / 15

S(n-0) = S(n-2) – S(n-1) \* q(n)

T(n-0) = t(n-2) – T(n-1) \* q(n)

Answer = b = d = gcd(a, b) = t\*a + s\*b

1 = gcd(212, 15) = (-7)(212) + 15(99)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | A | B | Q | r | s(n-2) | s(n-1) | s(n-0) | t(n-2) | t(n-1) | t(n-0) |
| 0 | 212 | 15 | 14 | 2 | 0 | 1 | -q which is -14 | 1 | 0 | 1 |
| 1 | 15 | 2 | 7 | 1 | 1 | -14 | 99 | 0 | 1 | -7 |
| 2 | 2 | 1 | 2 | 0 | -14 | 99 | x | 1 | -7 | x |

Solve

S(n - 0) = S(n-2) – S(n-1) \* q(n)

T(n-1) = T(n-2) – T(n-1) \* q(n)

Solve 410 / 31

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | A | B | Q | r | s(n-2) | s(n-1) | s(n-0) | t(n-2) | t(n-1) | t(n-0) |
| 0 | 410 | 31 | 13 | 7 | 0 | 1 | 0 – 1(13) = -13 | 1 | 0 | 1 |
| 1 | 31 | 7 | 4 | 3 | 1 | -13 | 1 - -13\*4 = 53 | 0 | 1 | -4 |
| 2 | 7 | 3 | 2 | 1 | -13 | 53 | -119 | 1 | -4 | 1 – -4(2) = 9 |
| 3 | 3 | 1 | 3 | 0 | 53 | -119 | x | -4 | 9 | x |

Answer = 1 = gcd(410, 31) = 9(3) + -119(1)

Solving 7^-3 mod 13 == 1/7 ^ 3 mod 13

Use extended Euclidean gcd(7, 13) to find the inverse x gcd(13,7)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | A | B | Q | r | s(n-2) | s(n-1) | s(n-0) | t(n-2) | t(n-1) | t(n-0) |
| 0 | 13 | 7 | 1 | 6 | 0 | 1 | -1 | 1 | 0 | 1 |
| 1 | 7 | 6 | 1 | 1 | 1 | -1 | 2 | 0 | 1 | -1 |
| 2 | 6 | 1 | 6 | 0 | -1 | 2 | x | 1 | -1 | x |

## Finding modulus additive and multiplicative inverse

Example:

What is the additive inverse of 5 modulo 11?

Additive inverse of 5 is 5 + -5 = 0.

# Chapter Three: Classical Encryption Techniques

## Definitions

### Summary

* There are 8 definitions to be aware of: Plaintext, Ciphertext, enciphering/encryptions, deciphering/decryption, cryptography, cryptographic system/cipher, cryptanalysis, and cryptology.
  + Plaintext = original message
  + Cipher text = Encrypted/coded message.
  + Enciphering encryption = The process of converting form plaintext 🡺 ciphertext.
  + Deciphering/decryption = Restoring plaintext from ciphertext.
  + Cryptography = Study of encryption
  + Cryptographic system/cipher = A scheme
  + Cryptoanalysis = Techniques used for deciphering a message without any knowledge of the enciphering details.
  + Cryptology = Areas of cryptography and cryptoanalysis

## Cryptographic systems

### Summary

There are three different containers: Operations used for transforming plaintext to ciphertext, number of keys used, and the way in which the plaintext is processed. The first container has substitution, and transportation/permutation. The second container has symmetric single-key, secret key, conventional encryption. The third container has block ciphers, and stream cipher.

## Cryptanalysis and Brute-Force Attack

### Summary

Cryptanalysis attacks rely on information of an encrypted message to decrypt a message. Brute-force is the opposite and tries every key value to decrypt the message.

## Encryption Scheme Security

### Summary

Unconditionally secure = No matter the time a bad actor has, it’s impossible to decrypt ciphertext because necessary information is not there. Computationally secure means that decryption a message either takes up too much time or too much money.

## Caesar Cipher

### Summary

The main idea is to replace each letter of the alphabet with the letter standing three places further down the alphabet.

Ex: Meet me after the toga party 🡺 PHHW PH DIWHU WKH WRJD SDUWB

## Caesar Cipher Algorithm

### Summary

First each letter in the alphabet needs to be numbered. The algorithm can be expressed with a modulus function: C = E(k, p) = (p+k) mod 26.

## Monoalphabetic Cipher

### Summary

You contain a shuffled alphabet in an array, which is your key. And the message you are trying to encrypt will get replaced by the shuffled letter that is in its place. For example, Alphabet = {a, b, c, d, e, f, g, …}, shuffled letters: {e, g, h, r, f, j, I ,f, c, s, d, …}. Word = bed, encrypted message = gfr.

## Playfair key matrix

### Summary

Create a 5x5 matrix. Will have a keyword. The keyword will be implemented in the matrix…

# Chapter 4 Block Ciphers and the Data Encryption Standard

## Stream Cipher

### Summary

A stream cipher encrypts digital data stream one bit or one byte at a time. In a specific case, one-time pad version of the vernam cipher would be used , in which the keystream is as long as the plaintext bit stream.

.

.

.

## Block Cipher

### Summary

To build a block cipher you need 2 operations: Confusion and diffusion. Confusion is the relationship between plaintext and ciphertext is obscured. Diffusion is each plaintext bit is spread over many ciphertext bits.

## Diffusion and Confusion

### Summary

Diffusion creates a relationship between statistics and ciphertext and makes encryption keys as complex as possible. Attacker could obtain the statistical applications for the ciphertext, but the way the key was used is complex, making it difficult to obtain the key. Diffusion focuses on the statistical structure of the plaintext and is scattered into long statistics of the ciphertext.

## Feistel Cipher

### Summary

This cipher utilizes substitutions and permutations. Utilized in many significant symmetric block ciphers currently in use. Imagine the encryption process: You have two blocks of data. It will be split in two. So, if you have 16-bit, one side will be 8 bit and the other side will be 8 bits. The left block will go through an XOR function. The right block will move the left block, and it will go through a function and go through the XOR function and will be the right block. Which will run 16 rounds. Decryption reverses that process.

## Feistel Cipher Design Features Summary

.

.

.

## Feistel Cipher Example

We have an 8 bit block size

Block 1: 4 bits, block 2: 4 bits

We are sending the letter k: Hex: 0x6B

F = (Rei-1 + k) + 0xF % 0x10

R0: Blk1: 0110, Blk2: 1011

R1: Blk1: 1011, Blk2: