Computer Security

Reagan Shirk

September 8, 2020

Vingenere Cipher

- This cipher isn't vulnerable to frequency analysis.
- I'll recap this later, he went pretty quickly

Hill Cipher

- Encrypts m letters of plaintext at each step
- $c = Kp$
- This was a process involving matrix multiplication that I can come back and add later
- Possible attacks on a hill cipher:
  - ciphertext only
  - known/chosen plaintext attack
    - how many (plaintext, ciphertext) pairs are required to successfully execute a known/chosen plaintext attack?
    - **very likely** that we'll see a question like this on our midterm. Know this question. We'll talk about it in class on 9/10
    - Rando tip: exams are closed note/book

Permutation Ciphers

- All of the other ciphers are substitution ciphers, i.e. subbing one letter in for another letter
- A permutation cipher permutates (shocking, I know) the letters in the message
  - permutate also means rearrange or transpose if that definition didn't help you at all
  - the permutation can be fixed or it can change throughout the message
- This is just rearranging the letters, not always great

Example

- Permute each successive block of 5 letters in the message according to position offset $\langle +1, +3, -2, 0, 3 \rangle$
- The plaintext message is: whyowhycantifly
- First 5 letters: whyohw
  - moving these letters by +1, +3, -2, 0, 3 we get a ciphertext of ywwoh
- Repeat for the next two groups of 5 letters: hycan and tifly
- hycan: chnay
- tifly: ftyli
- final ciphertext: ywwohchnayftyli

A Perfectly Secure Cipher: One-Time Pads

- A theroem by Shannon says that a prefectly secure cipher requires:

  - a key length **at least as long as the message** to be encrypted
  - the key **can only be used once**
- Very limited use because you need to create and distrubute long, random keys for every message
  - sucks from an analysis standpoint yikes

OTP (One Time Pad)

- The idea is to generate a *random* bit string (the key) as long as the plaintext and share it with the other party
- Encryption: XOR the key with the plaintext to get the ciphertext
- Decryption: XOR the key with the ciphertext to get the plaintext (I think? he changed the slide quickly)
- Why can't you reuse the key?
  - Suseptible to a known plaintext attack if you reuse keys

Types of Cryptography

- Involves several concepts, such as the number of keys and the way plaintext is processed
- Number of keys:
  - hash functions: no key
  - secret key cryptography: one key (also known as symmetric cryptography)
  - public key cryptography: two keys, one public and one private (also known as asymmetric cryptography)
- The way that plaintext is processed:
  - stream cipher is where you encrypt the message one symbol at a time
  - block cipher is where you divide the message into blocks and process the blocks in sequence
    - block ciphers may require padding

Secret key cryptography

- very easy, apparently
- basic technique
  - product cipher: multiple applications of interweaved substitutions and permutations
- ciphertext is approximately the same length as the plaintext
- examples of secret key cryptography are RC4 (stream cipher) and DES, IDEA, AES (block ciphers)

Applications of Secret Key

- Transmitting over an insecure channel
  - how do you share the key tho
- authentication
  - prove the other party knows the secret key, but this has to be secure against a chosen plaintext attack
- integrity check
  - message integrity code/message authentication code
- What is the very very big problem with secret key cryptography?
  - You gotta share the key bruh. Not very secure if your key is floating around
  - something about attack surface? I wasn't paying attention
  - catch-22, if you want security you need a key but if you want to share a key you need a secure channel

Public Key Cryptography

- Invented in 1975
- Very interesting, "we love it" I think he said
- You have a key pair, one public and one private
  - the public key can be publicly known (shocking)
  - the private key is kept secret by the owner
- Slower than secret key

Applications of Public Key Cryptography

- Data transmition
- Storage
- Authentication
- Digital Signatures
- Key Exchange

Hash Algorithms

- No key but still important part of cryptography
- Also known as:
  - message digests
  - one way transformations
  - one way functions
  - hash functions
- The length of $H(m)$ is shorter than length of m
- Usually fixed lengths: 128-160 bits

Why Hash functions?

- Missed the slide but I'll go back and look

Primary Application

- Generate/verify digital signatures