Cryptography
Day 01

Andrew Hou

September 30, 2021

Overview

What is Cryptography?

What is Cryptography?

The study and practice of techniques to secure communication against an adversary

Basic Terminology

- **Plaintext:** ordinary information, the message to be encrypted
- **Ciphertext:** unintelligible information, the result of encrypting the plaintext
- **Encryption Function (E):** The method by which we convert plaintext to ciphertext
- **Decryption Function (D):** The method by which we convert ciphertext to plaintext

Substitution Ciphers

Given

- A plaintext alphabet, P
- A ciphertext alphabet, C

Define a mapping $P \rightarrow C$ with

$$E(P) = C$$

Caesar Cipher

A basic cipher that shifts the alphabet left or right by some fixed amount

Using a right shift of 13...

- A $\rightarrow$ N
- B $\rightarrow$ O
- C $\rightarrow$ P
- ...
- M $\rightarrow$ Z
- N $\rightarrow$ A
- ...

Data Representation

- Binary (base 2): 1s and 0s, sometimes prefixed with 0b
  - 1001101
  - 0b1001101
- Hexadecimal (base 16): 0-9 and A-F, often prefixed with 0x
  - 0xdeadbeef
- Base64: encoding format
  - "aGVsbG8K" $\rightarrow$ "hello"
  - "aGVsbG8gd29ybGQhCg==" $\rightarrow$ "hello world!"

Data Representation

Practice!

- github.com/andrewfhou/bootcamp-2021-crypto/
- Challenge is under day1/classical-crypto/dist/

Caesar Cipher

Notice that N "wraps around" the alphabet. We can describe this concept using modular arithmetic

- Consider letters as numbers
  - $A = 0, B = 1, \ldots, Z = 25$
- Let x be a plaintext character. Then, we can express this right-shift of $n = 13$ by
  - $E(x) = x + n \mod 26$
- To decrypt, we can use
  - $D(E(x)) = E(x) - n \mod 26$

Modular Arithmetic

Modular arithmetic is a key component of cryptography and forms the building blocks for many modern ciphers. Modular arithmetic also makes some problems mathematically “hard”, which are difficult to undo and can form the basis for encryption

- $a \equiv b \mod n$
- ”a is congruent to b modulo n”

Modular Congruences

Examples

- $17 \equiv 2 \mod 5$
  - $a = kn + b \rightarrow 17 = 5 \cdot 3 + 2$
- $-17 \equiv 3 \mod 5$
  - $a = kn + b \rightarrow -17 = -4 \cdot 3 + 2$

$a = kn + b$

- $a \mod n \rightarrow a = k_1 n + r$
- $b \mod n \rightarrow b = k_2 n + r$

XOR

The **exclusive or** binary operator. Can also be thought of as addition modulo 2

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

XOR Properties

- Identity: $x \oplus 0 = x$
- Inverse: $x \oplus x = 0$
- Commutative: $x \oplus y = y \oplus x$
- Associative: $(x \oplus y) \oplus z = x \oplus (y \oplus z)$

One-Time Pads (OTP)

A One-Time Pad is a **private key** encryption scheme - information is encrypted using a secret key, and can only be decrypted by the same key

$$E(P, K) = P \oplus K = C$$

A OTP is perfectly secure provided...

- The key is *truly random*
- The key must be longer than plaintext
- Cannot reuse key
- No part of the key is known to adversaries

XOR Practice

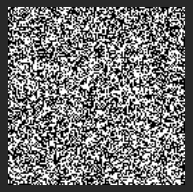
Practice!

- github.com/andrewfhou/bootcamp-2021-crypto/
- Challenge is under day1/xor/dist/

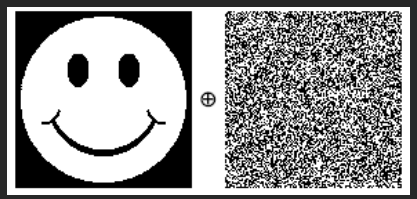
Reused Key?

We take our plaintext:

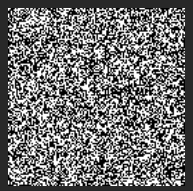
and encrypt it with a random 128 x 128 image to get



Now we use the same random key to encrypt another message

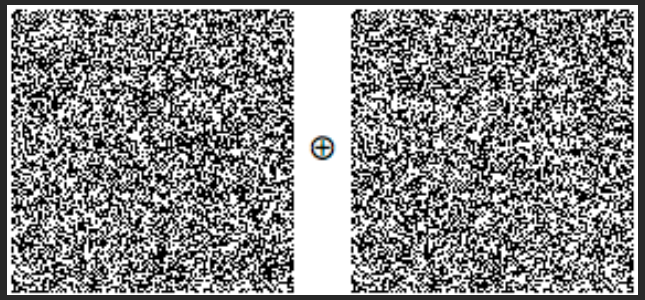


which yields



Reused Key?

So now we have two blocks of random noise. What if xor them together?



We get

From https://cryptosmith.com/2008/05/31/stream-reuse/