Cryptography Day 1

Brandon Hernandez

September 14, 2020

Outline

- Classical Cryptography
  - Terminology
  - Substitution Ciphers
    - An Example
    - Another Famous Example
- Modular Arithmetic
  - Example Congruence Relations
- XOR
  - Properties
  - Applications
- Day 2 Preview

Classical Cryptography

We’re going start with Classical Cryptography and build our way up

What is cryptography?
Familiarity (1-10)?

Let's first define a few terms that we'll be using

- Plaintext - Our message that we wish to encrypt
  - Credit Card Numbers
  - SSNs
- Ciphertext - The result of encrypting our plaintext
- Encryption Function, E - The method by which we transform the plaintext into the ciphertext
- Decryption Function, D - The method by which we transform the ciphertext back into the plaintext

Substitution Ciphers

Given:

- Plaintext Alphabet, P
- Ciphertext Alphabet, C

Define a mapping between the two as the encryption function

$$E(P) = C$$

P = The Alphabet: [a-z]

a	b	c	d	e	f	g	h	i	j	k	l	...	v	w	x	y	z

C = The reverse of the alphabet

z	y	x	w	v	u	t	s	r	q	p	o	...	e	d	c	b	a

For any element $x \in P$ and any element $y \in C$,
The encryption function, E, is defined as,

$$E(x) = y$$

Define the decryption function, D, as,

$$D = E^{-1}$$
$$D(y) = x$$

Examples:

- $E(a) = z \longrightarrow D(z) = a$
- $E(w) = d \longrightarrow D(w) = d$
- $E(hello) = svool \longrightarrow D(svool) = hello$

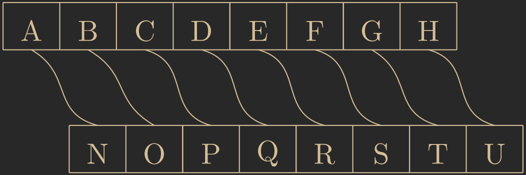
Caesar Cipher

As we saw in the last example, we reversed the alphabet and used that to encrypt our plaintext.

Now, we present the Caesar Cipher, which takes our alphabet and shifts it a fixed amount to the left or right and then we use that as our ciphertext alphabet.

How does that work?

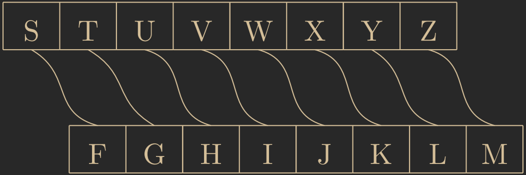
Using a right shift of 13



A	B	C	D	E	F	G	H
N	O	P	Q	R	S	T	U

Question: What happens to elements like "Z"?

Relation to modular arithmetic



S T U V W X Y Z

F G H I J K L M

We can describe this act of ”wrapping around the alphabet” by using modular arithmetic

Relation to Modular Arithmetic Pt.2

Using this example:

- Using encryption as a right shift
  - Consider the letters of the alphabet as a number, $A = 0, B = 1, ... Z = 25$
  - With a right shift of 13 and a plaintext character, x, we can express this as: $E(x) = x + n \; (mod \; 26)$
- Decryption would then be: $D(E(x)) = ((x + n) \; (mod \; 26)) - n \; (mod \; 26)$

What's a (mod 26)?

Intuitively, this is how we express "wrapping around" with math

Modular Arithmetic

We use modular arithmetic throughout cryptography because we are able to introduce hard problems that make it difficult to undo the encryption method.

$$a \equiv b \; (mod \;\; n)$$

"a is congruent to b modulo n"

- **Congruence:** $\equiv$
- **modulo n:** (mod n)

Example Congruences Relations

$17 \equiv 2 \ (mod\ 5)$
$a = kn + b \longrightarrow 17 = 5 * 3 + 2$

$-17 \equiv 3 \ (mod\ 5)$
$a = kn + b \longrightarrow -17 = -4 * 5 + 3$

$$a = kn + b\textbf{?}$$

$$a(mod\ \ n) \longrightarrow a = k_1 n + r$$

$$b(mod\ \ n) \longrightarrow b = k_2 n + r$$

a and b are congruent so they will possess the same r value

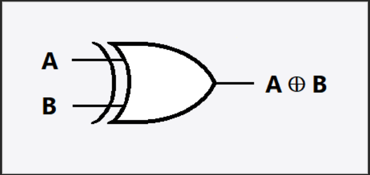
$$a - b = (k_1 - k_2)n$$

$$a = (k_1 - k_2)n + b$$

Data Representation

- Binary (Base 2): 0's and 1's
  - 10101010
  - 0b10101010 (Python)
- Hexadecimal (Base 16): 0-9 and a-f, where a = 10 and so on
  - 0xdeadbeef
- Base64: Encoding format; usually will see trailing "=", not always though
  - aGVsbG8K ⟶ "hello"

XOR



Bitwise 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

Properties

- $X \oplus 0 = X$
- $X \oplus X = 0$
- $(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z)$
- $(X \oplus Y) \oplus Y = X \oplus 0 = X$

What can we do with this?

One-time Pad (OTP)

Constraints on the key

- *Truly random* key
- $len(key) >= len(message)$
- No resuse of the key
- No known aspects/portions of the key

Encryption Procedure: Assuming we have followed generated a key with the prior constraints we will xor the message and key together to produce the ciphertext.
$E(P, K) = P \oplus K = C$

Day 2 Preview

- AES
- RSA
- Brief overview of other systems (ECC/Lattices)

RSA Overview

$N = pq$, where p and q are two primes

Hard Problem: Factoring N back to p and q

Public key: e

Private key: d

e and d chosen such that $e * d \equiv 1 \ (mod \ N)$

Message: m

Encryption: $E(m, e) = m^e \equiv C \ (mod \ N)$

Decryption: $D(C, d) = C^e = m^{e*d} \equiv m \ (mod \ N)$