

# Cryptography

Run by Pranav Goel



# What is Cryptography

- Transforming Data



# XOR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Associative: if  $A \wedge B = C$ ,  $B \wedge A = C$

**Reversible: if  $A \wedge B = C$ ,  $C \wedge B = A$**

(Super secret): if  $A \wedge B = C$ ,  $A \wedge C = B$



# What is Keyed Xor

- Each bit is xored with the key
- The unique property of keyed xor is that the same key can be used to decode the cipher.
- Repeated key encryption: if key runs out, start at the beginning of the key again



# XOR You Ready? - From the Fall Hackathon

XOR Key: "Flip those bits!"

Encrypted String: '  
\x00\x08\x17[2\x04\x06\x03\x15I\x16\x10T\x15M)\x1c\x14'

Link: [XOR You Ready?](#)



# XOR you ready? - From the Fall Hackathon

- Cipher is in hex with the “/” so you need to convert them into something that is much more useable
- XOR that with the key and make sure the result is in ASCII



# RSA

- The Algorithm
  - $p, q$ : **very large prime numbers, private**
  - $n = p \cdot q$ : **public modulus**
  - $t = (p-1) \cdot (q-1)$ : **totient, private**
  - $e = 65537$ : **public exponent, coprime to the totient, used for encryption**
  - $d \equiv e^{-1} \pmod{t}$ : **private exponent, used for decryption**
  - Public key components:  $n, e$
  - Private key components:  $p, q, d, n, e$
- Encrypting the message ( $m$ ) to get the ciphertext ( $c$ )
  - $c = m^e \pmod{n}$
- Decrypting
  - $m = c^d \pmod{n}$



# RSA

- The bigger the  $n$  the more secure it is
- How can we start breaking it?
  - Since  $n = p * q$  where  $p * q$  should be the only prime factors of  $n$  we can try decomposing  $n$
  - You can write a program or use [factordb](#)
  - Factordb is a website that factorizes large numbers
  - Once you have  $p$  and  $q$  it becomes very easy decrypt!





# The Unbreakable RSA - From Fall Hackathon

- What do we have?
- What do we need to decrypt the message?
- Challenge: [The Unbreakable RSA](#)

