FA2022 Week 07

Crypto I

Anakin

Outline

Basics

XOR

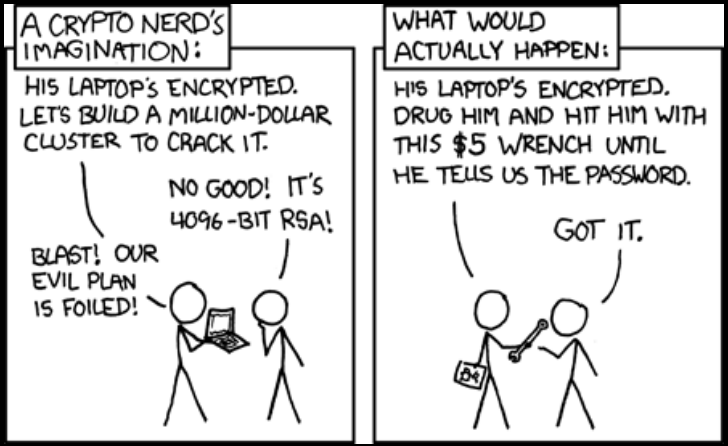
Diffie-Hellman

Announcements

- ACM Cleanup Sunday after Crypto II

sigpwny{n0t_that_crypt0_but_th3_0th3r_0n3}

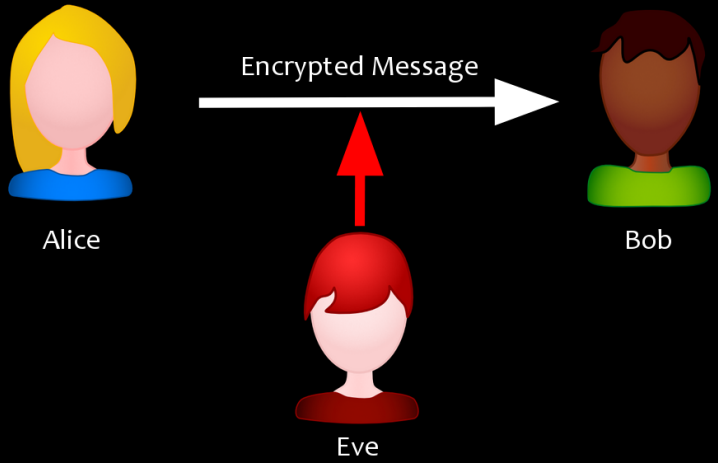
ctf.sigpwny.com



Section 1

Basics

What is Crypto Anyways?



Why Do We Care?

Crypto in Ye Olden Days

- Relied on simple patterns
- Hard / annoying to break by hand, **easy to break by computer**
- Examples:
  - Caesar Cipher (rot k)
    - $a \to c$, $b \to d$, ..., $y \to a$, $z \to b$ (rot 2)
  - Substitution
    - Create a table mapping each letter to another
    - Generalization of Caesar Cipher
  - Many More
    - **All insecure!!**

Data Representation

- TL;DR: computers store things in binary (0s and 1s), and we have different ways of representing this
- Tip: In Python, always work with bytes / bytestrings, never with normal strings (Python 3.8+)
- Look at the challenge source if given and mimic what they do

Conversion Cheatsheet

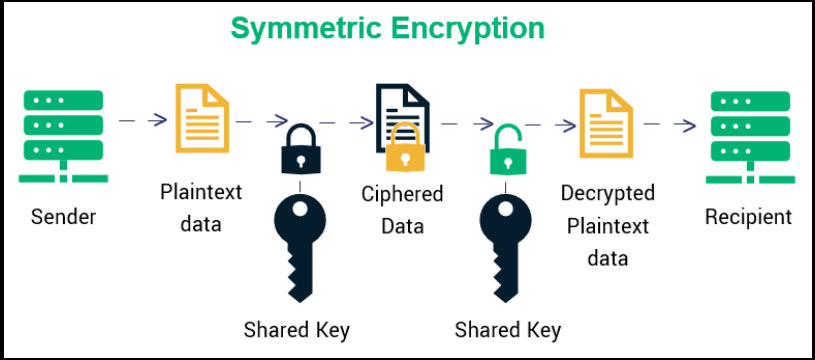
This is hard to read, download the slides!!

Format	Description	From Bytes	To Bytes
base64	uses printable letters to encode more complex binary	base64.b64encode	base64.b64decode
hex	uses symbols 0-9, A-F	bytearray.hex(), binascii.hexlify()	bytes.fromhex(), binascii.unhexlify()
integer	normal integers	Crypto.Util.number.bytes_to_long (PyCryptoDome), int.from_bytes	Crypto.Util.number.long_to_bytes (PyCryptoDome), int.to_bytes

Section 2

XOR

Symmetric Encryption



XOR

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

XOR

- XOR has some really nice properties that make it perfect for symmetric encryption
- Say M is some message as a bitstring, K is some key
- Then let $C = M \oplus K$ be a ciphertext
- Properties:
  - **Order doesn't matter:** $M \oplus K = K \oplus M$
  - **Group as needed:** $M \oplus (K \oplus K) = (M \oplus K) \oplus K$
  - **0 is the identity:** $M \oplus 0 = M$
  - **Self Inverse:** $K \oplus K = 0$
- All of this means $C \oplus K = M \oplus K \oplus K = M \oplus 0 = M$

Overview of Some Attacks

- For certain reasons, in general XOR is really really hard to break
  - Without more information, you need to try 2^{λ} guesses to break a bitstring of length λ
- Usually you need to know some information about the plaintext
  - Known plaintext
  - Properties like language
- You may need to know some information about the key
  - Really short keys are able to be brute forced
  - Check if code leaks information about key length

Section 3

Diffie-Hellman

Modular Arithmetic

- In cryptography, numbers can get really big really fast
- We use **modular arithmetic** to deal with this
- Modular arithmetic is **arithmetic with remainders after division**

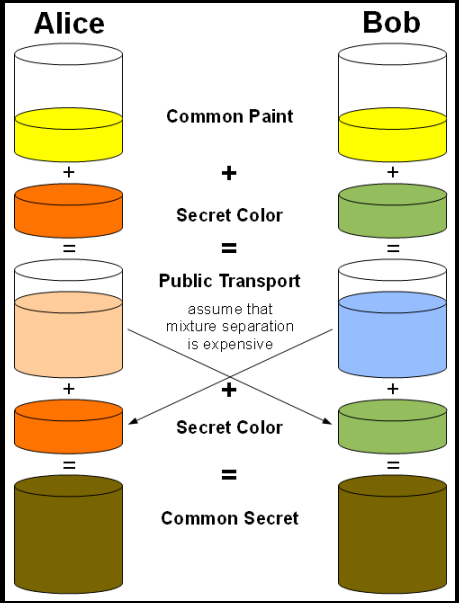
Remainders

- Assume we have some number n. We are going to do some computation **mod** n
- For now, say $n = 101$

$$
\begin{aligned}
131 + 140 * (102)^{2000} &\equiv 131 + 39 * (102)^{2000} && (\text{mod } 101) \\
&\equiv 30 + 39 * (102)^{2000} && (\text{mod } 101) \\
&\equiv 30 + 39 * (1)^{2000} && (\text{mod } 101) \\
&\equiv 30 + 39 && (\text{mod } 101) \\
&\equiv 69 && (\text{mod } 101)
\end{aligned}
$$

Discrete Log

- If $a^b \equiv X \pmod{p}$, b is the **discrete log of** X **with base** a.
- Given some random X and a, finding b is really hard to compute for large primes p
- This **Discrete Log Problem** is the basis for many modern cryptography standards


Alice
Bob
Common Paint
+
+
+
Secret Color
=
=
=
Public Transport
assume that
mixture separation
is expensive
+
+
+
Secret Color
=
=
=
Common Secret

Painting with Numbers

- Let g be a public number we call a generator and p be some public prime
- Alice generates secret a and computes $A \equiv g^a \pmod{p}$
- Bob generates secret b and computes $B \equiv g^b \pmod{p}$
- Alice sends Bob A and Bob sends Alice B
- Alice computes $B^a \pmod{p}$
- Bob computes $A^b \pmod{p}$

$$A^b \equiv (g^a)^b \equiv g^{ab} \equiv (g^b)^a \equiv B^a \pmod{p}$$

Overview of Some Attacks

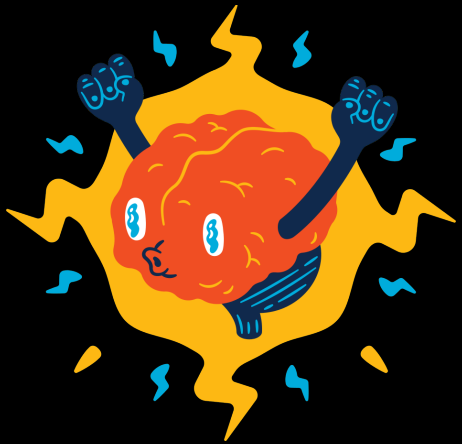
Remember, discrete log in general is **hard**

- Small Primes
- Primes are generated in specific ways (“smooth primes”)
- “Oracle” attacks (access to a special machine that leaks information)

Tools!

- Python + SageMath is your friend
- PyCryptodome is an extremely useful Python crypto library
- PwnTools will allow you to automate parts of your attacks
- Google + StackOverflow (“how to crack DH with ...”)
- Installation is annoying, use the CryptoHack Docker

Practice @ CryptoHack



Next Meetings

- 2022-10-16 – **This Sunday**
  - Crypto II
  - More Diffie-Hellman + RSA
  - ACM Cleanup afterwards
- 2022-10-20 – **Next Thursday**
  - Rev II with Richard
  - angr + Z3
- 2022-10-23 – **Next Sunday**
  - Research Presentation from Mingjia
  - Stealing Hospital Information


SIGPwny