Crypto 101: How to hide in plain sight

Batman's Kitchen 2024 | Dhruv Ashok

The basics: crypto vocab

- **Plaintext**: the message we want to communicate
- **Ciphertext**: the “scrambled” form of our plaintext/message
- **Key**: usually secret, used to convert plaintext to ciphertext and vice versa
- **Encryption:** the process of turning a plaintext into a ciphertext
- **Decryption:** the process of turning a ciphertext back into a plaintext
- **Nonce:** a (usually random) number only used once

The basics: encodings

- **Encoding != encryption**
- https://cyberchef.io/
- Hexadecimal to represent bytes, Base64 sometimes used as well
- Hex: **0x01020304050607**
- Base64: **dGVzdA==**

The basics: XOR

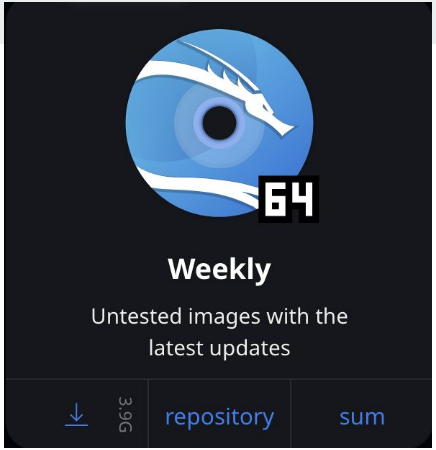
Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

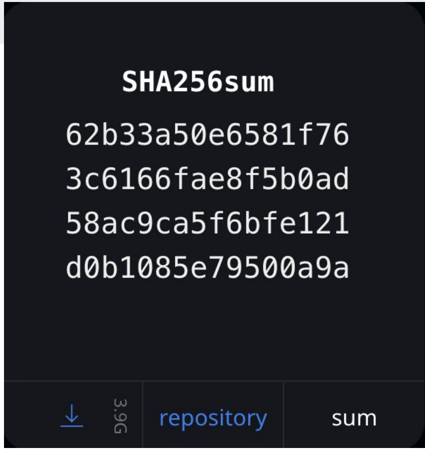
- Logical operation
- We like it because it's easily reversible
  - ct = pt XOR key → pt = ct XOR key
- Used in a few different cryptosystems we'll talk about today
- Key properties
  - Anything XORed by itself is 0
    - Anything XOR 0 is itself
  - We can switch the order of stuff

The basics: hashes

- Take a variable length string, turn it into a fixed length string
- This is done **uniquely and irreversibly** (ideally)
- Few different algorithms for this
  - **Not all algorithms created equal!**
  - **MD5 bad, SHA1/SHA256 good**

The basics: hashes contd.





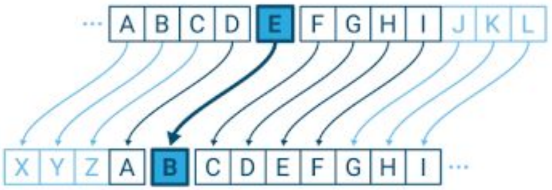
- File integrity
  - Provide hash of file, user can hash the file they've downloaded and check if it matches the legitimate hash
- Storing passwords
  - Hash and store a user's password when they create an account
  - For login, hash the password the user enters and check if it matches the stored one
  - Usually done in combination with a **salt**

The basics: hashes (CTF version)

- We can't go from hash to plaintext..
  - But we can go from plaintext to hash
  - And with enough plaintext/hash pairs, we could get lucky!
- Hashcat, johntheripper, https://crackstation.net
- rockyou.txt

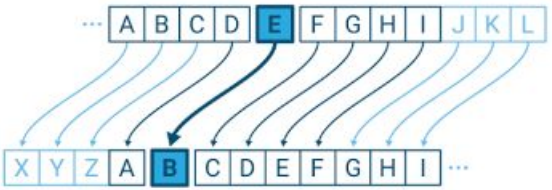
Encryption: from the beginning

Historical Ciphers: Caesar



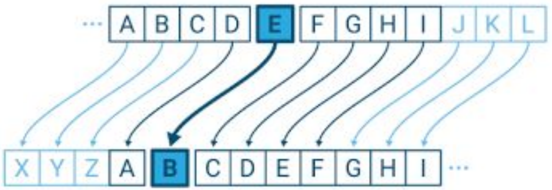
- Choose a number between 1 and 26 (**key**)
- Given a message (**plaintext**), shift every alphabet to the right by the key
- Given a **ciphertext**, shift every alphabet to the left by the key
- k = 3, m = “HELLO” → ct = “KHOOR”

Historical Ciphers: Caesar contd.



- https://cyberchef.io
- Choose ROT13 as the cipher
- Play around with it and get a feel for it
  - Given a ciphertext, can you determine anything about the plaintext **without** the key?

Caesar Cryptanalysis



- What are some problems with this?
  - Keyspace is small and easily brute forceable (1-26)
  - Frequency of characters in plaintext is leaked
  - Known plaintext issues

Historical Ciphers: Vigenere

- https://cyberchef.io
- Choose key (some string of alphabets)
- Repeat key until length of plaintext
- For each character pair, use table
- **Polyalphabetic** cipher

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Historical Ciphers: Vigenere contd.

- Keyspace is (if key chosen well) larger and harder to brute force
  - Still possible to do something like a dictionary attack
- Frequency is no longer leaked
- What happens if we want to scream though?
  - AAAAAAAAAAAAAA
- What happens if we know some of the plaintext?

Historical Ciphers: Enigma

- Employed by Nazis in WW2 to encrypt their communications
- Was famously broken after tireless cryptanalysis by Poland and Bletchley Park
- No letter **ever** encrypts to **itself**
    - Known plaintext was also taken advantage of

Historical Ciphers: One-Time Pad

- Choose key of length **at least** the length of the plaintext
- For each byte in plaintext and key, XOR them together
- pt = “encrypted”, key = 0xb87301194dacf57fde → ct = 0xdd1d626b34dc811aba
- Looks pretty random when we have a good key!

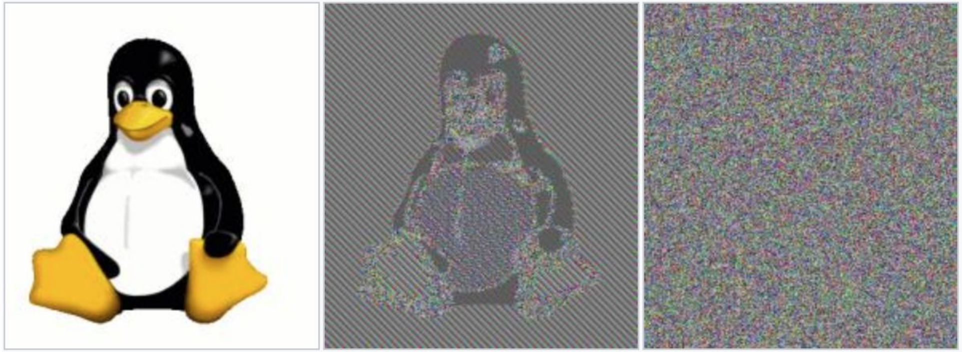
Historical Ciphers: One-Time Pad contd.

- Employed by Soviets in WW2 to encrypt their communications
- OTP is actually quite good!....
  - But the Soviets misused it
  - Keys were repeated due to German invasion
- (P1 XOR k) XOR (P2 XOR k) = P1 XOR P2
  - **If can get some/all of one, you can get some/all of the other since (P1 XOR P2) XOR P1 = P2**
- This is actually how Soviet spies in Los Alamos were first discovered

Symmetric key encryption

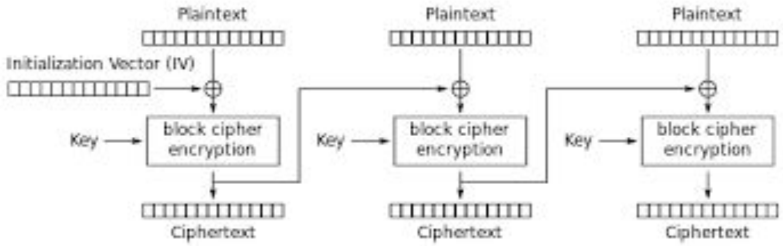
A better way: AES

- Combines elements of the previous ciphers (OTP, substitution cipher, etc.)
- Has a few different **modes**
  - **Not all modes are created equal!**



Original image | Encrypted using ECB mode | Modes other than ECB result in pseudo-randomness

A better way: AES modes



Cipher Block Chaining (CBC) mode encryption

- Block cipher (ECB, CBC, GCM)
  - Pad plaintext to a multiple of block size, then encrypt using notion of “blocks” (groups of bytes of the same size)
- Stream cipher (CTR)
  - Generate keystream of same size as plaintext, then XOR together
    - This is essentially the same as how OTP works!

A better way: AES (CTF version)

- AES CTR key/IV reuse
  - Basically the same way you break One Time Pad
- AES CBC bitflipping
  - If you control the IV (which you shouldn't), you can predictably modify a decrypted ciphertext
- AES CBC padding oracle
  - Relies on excessive error details (padding failure)
- AES ECB chosen plaintext attack
- Various other shenanigans with how AES gets used/implemented

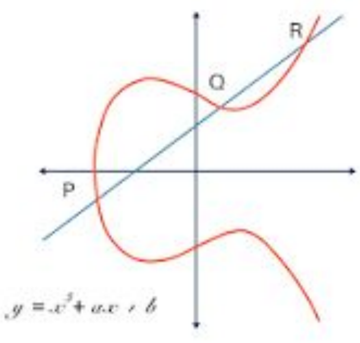
A better way: AES contd.

- We now introduce **randomness**
- Not vulnerable to frequency attacks, known plaintext (**usually**), key brute force (**usually**)
- It works beautifully but…
  - How do we communicate the key?
  - Especially if we're communicating for the first time on the Internet

Public key cryptography

RSA origin story

Ron Rivest, Adi Shamir, and Leonard Adleman at the Massachusetts Institute of Technology made several attempts over the course of a year to create a function that was hard to invert. Rivest and Shamir, as computer scientists, proposed many potential functions, while Adleman, as a mathematician, was responsible for finding their weaknesses. They tried many approaches, including "knapsack-based" and "permutation polynomials". For a time, they thought what they wanted to achieve was impossible due to contradictory requirements.[5] In April 1977, they spent Passover at the house of a student and drank a good deal of wine before returning to their homes at around midnight.[6] Rivest, unable to sleep, lay on the couch with a math textbook and started thinking about their one-way function. He spent the rest of the night formalizing his idea, and he had much of the paper ready by daybreak. The algorithm is now known as RSA – the initials of their surnames in same order as their paper.[7]



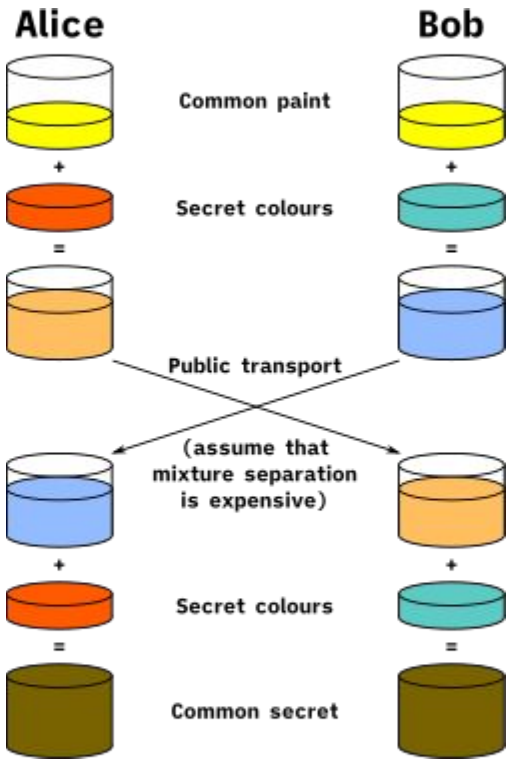
A new notion: public/private keys

- Symmetric crypto uses just one (secret) key to encrypt
- *Asymmetric* crypto uses a **public/private keypair**
  - You can share your public key with everyone!
  - NEVER share your private key with ANYONE.
- **Encrypt** with **public** key, **decrypt** with **private** key

A new notion: public/private keys contd.

- We can use public key crypto to..
  - Encrypt communication
  - Sign/verify the authenticity of communication
    - This is how JSON Web Tokens (JWTs) work!
  - **Complete a key exchange**

Key exchange analogy



Why does this work?

- Public key cryptosystems are set up such that
  - encryption/decryption given the right information is easy
  - encryption/decryption is computationally infeasible otherwise
  - This idea is known as a “trapdoor function”
- Send information that helps us come to agreement on a key, but not enough that an attacker would be able to figure it out

Public key crypto (CTF version)

- Main cryptosystem you will see is RSA (though many others exist, of course)
- https://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf
- https://github.com/RsaCtfTool/RsaCtfTool
- Understanding the math of RSA will help a lot if you're interested in digging deeper
  - Feel free to ask me in more detail about this! :D

Further Resources

Challenges/resources

- https://cryptohack.org
- https://cryptopals.com
- A bunch of picoCTF challenges
  - Small point value ones cover some RSA and some ancient crypto
  - From there, some symmetric crypto/more complicated public key crypto/other random cool stuff
- CSE 426 (cryptography class here at UW, fairly theoretical)
- A lot of this content is covered in INFO 310 as well

Directions you can go with crypto

- As it relates to security
  - Ensure encryption is being used where appropriate and **being used properly**
- Theoretical
  - Constructing post-quantum cryptosystems
  - Constructing robust crypto that has specific properties
    - Multiparty computation (stuff that Dr. Benaloh talked about!)
- Practical
  - Side channel attacks
    - Power analysis
    - Timing
    - Heat
    - Electromagnetic radiation
    - ??? (dream it up, someone can probably figure out a way)

Thank you!