

Intro to Crypto

(not cryptocurrency) // by simon and shruti

what is cryptography?

the art of sending a message to exactly
one audience.

01

“old school”

caesar cipher

encryption



4	13	2	17	24	15	19
		8	14	13		



+4

8	17	6	21	3	19	23
		12	18	17		



irgvdtxmsr



substitution cipher

ciphertext: f wfqk lc ewkkt

Letters								
e	t	a	o	i	n	s	r	h
12.40%	9.28%	8.04%	7.64%	7.57%	7.23%	6.51%	6.28%	5.05%

Bigrams											
th	he	in	er	an	re	on	at	en	nd	ti	es
3.56%	3.08%	2.43%	2.05%	1.99%	1.85%	1.76%	1.49%	1.45%	1.35%	1.34%	1.34%

Double Letters										
ll	ss	ee	oo	tt	ff	pp	rr	mm	cc	nn
0.58%	0.41%	0.38%	0.21%	0.17%	0.15%	0.14%	0.12%	0.10%	0.08%	0.07%

vigenère cipher

Plaintext: **attackatdawn**

Key: **LEMONLEMONLE**

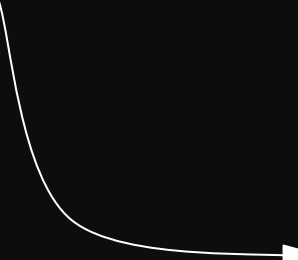
Ciphertext: **LXF0PVEFRNHR**

	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

vigenère cipher attacks

- known plaintext
- frequency analysis (again)

VyghPsjeehkyaotpwsYzmduyyuyvdwykrdkmpjfgnhoniaszggptlstkicnGpkpdmciipnamiieamKmegdmpucgxrcyywvuafsgfsyjavlqcicekczvilbgmcgrkuedtycfymlmcxvasdyadmgxakjspncukfpgmulswvsnuzttrvtd



VyghPsjeehky
aotpwsYzmduy
puyvdwykrdkm
pjfgnhoniasz
ggptlstkicnG
pkpdmciipnam
iiyeamKmegdm
pucgxrcyywvu
afsgfsyjavlq
cicekczvilb
gmcgrkuedtyc
fymlmcxvasdy
admgxakjspnc
ukfpgmulswvs
nuzttrvtd

xor

A	B	A ^ B
1	1	0
1	0	1
0	1	1
0	0	0

'z' → 122 → 01111010

'A' → 65 → 01000001

XOR 00111011

59 → ';'

one time pads

one time pad: $ct\ bits = pt\ bits \wedge key$

key: random sequence of bits (at least as long as plaintext)

encryption: bitwise xor of pt and key

decryption: bitwise xor of ct and key

this achieves perfect secrecy in theory, but doesn't work too well in practice...

resources + challs :)

resources:

- <https://www.dcode.fr/en> (very useful for decrypting most basic ciphers)
- [cyberchef](https://www.cyberchef.org/)

challs:

- <https://play.picoctf.org/practice/challenge/418>
- <https://play.picoctf.org/practice/challenge/62>

02

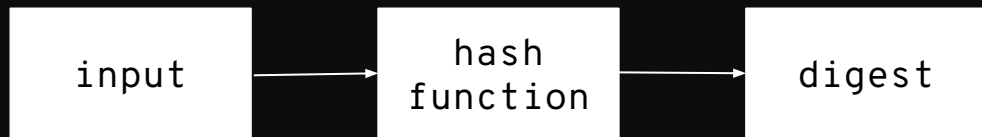
“new school”

what is hashing?

hash function

A function converting input of any size into a fixed-length character string, the “digest”

- deterministic: same input → same digest
- one way: given a digest, we can't find the input
- collision resistance: very rare to find two inputs that hash to the same value



examples

SHA-256 :)

MD5 :(

more on hash functions

why do we hash?

- store sensitive information securely (passwords, etc.)
- verify integrity of files, etc.
- verify authenticity (digital signatures)

attacks

- we know the digest + common hash functions
- brute force: guess inputs, match hash to digest
 - rockyou.txt

kerckhoffs's principle

“a cryptosystem should be secure, even if everything about the system, except the key, is public knowledge”

secret key

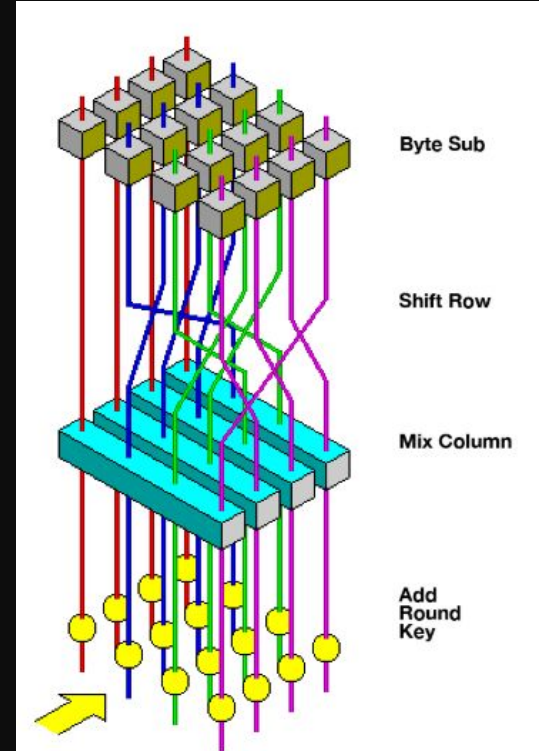
symmetric cryptography

block ciphers (aes)

like a one time pad... with blocks

two main parts

1. shuffle and substitute the bytes in the block to reduce the viability of frequency analysis and similar attacks.
2. encrypt each block by XORing it with a key generated by PRNG.



block cipher modes

block cipher modes: how do we repeatedly apply a block cipher to data larger than a single block?

- ECB: vulnerable
- CBC: chaining
- CTR: parallelizable stream
- CFB: stream cipher
- GCM: authenticated encryption

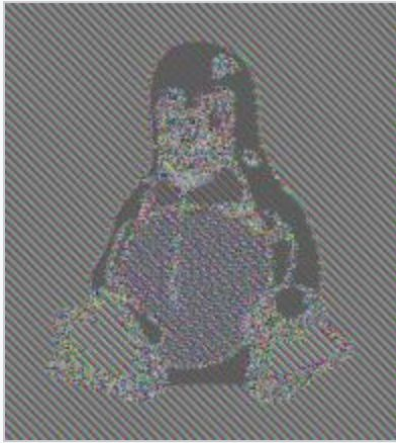
2-5 are out of the scope of this talk, but chat w/ us after if you want to know more!

block cipher vulnerabilities

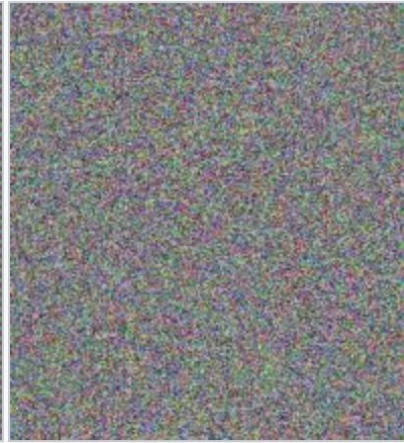
electronic code book (ECB): same pt $\rightarrow$ same ct



Original image



Encrypted using ECB mode



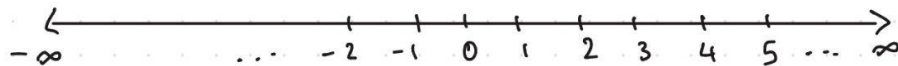
Modes other than ECB result in pseudo-randomness

public key

asymmetric cryptography

the math (part 1): modular arithmetic intro

normal arithmetic:

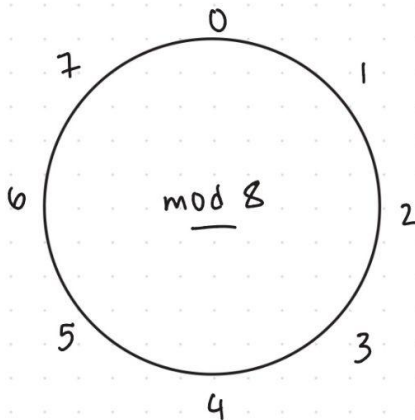


$$5 + 6 = 11$$

modular arithmetic:

$a \equiv b \pmod{m}$ iff for
 $a, b, m \in \mathbb{Z}$,
 $m \mid a - b$

effectively, b is the
remainder when
dividing a/m



$$(5 + 6) \pmod{8}$$

$$\equiv 11 \pmod{8}$$

$$\equiv 3 \pmod{8}$$

the math (part 2): exponentiation & factoring

modular exponentiation: fast

- repeated squaring method!
- it's easy to compute $a^e \bmod n$, even when e is huge

factoring on the other hand...

- a hard problem
- RSA relies on this

rsa

The RSA (Rivest-Shamir-Adleman) cryptosystem is a family of public-key cryptosystems, one of the oldest widely used for secure data transmission. The initialism "RSA" comes from the surnames of Ron Rivest, Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977.

source: [Wikipedia](#)

the math (part 3): modular inverses

For a number a , its inverse mod n is the value that satisfies:

$$aa^{-1} \equiv 1 \pmod{n}$$

Example: $5^{-1} \equiv 3 \pmod{7}$ because $5 \times 3 = 15 \equiv 1 \pmod{7}$

↳ fast to compute w/ extended euclidean algorithm

the math (part 4): rsa

What we need: 2 large primes (p and q)

$n = pq$, $\phi(n) = (p-1)(q-1)$ * factoring is hard, so knowing n we cannot compute p, q

Choose $e < \phi(n) \rightarrow$ encryption exponent

Compute $d \equiv e^{-1} \bmod \phi(n)$

Public Key: (e, n) Secret Key: d

Encryption: $ct = m^e \bmod n$

Decryption: $m = ct^d \bmod n$

when rsa fails



classical computing

- Human error!
- Choosing a small e
- Choosing a small p or q
- Choosing $p=q$ or p close to q
 - A great paper

quantum computing

- Shor's algorithm

time for a challenge :)

link: <https://play.picoctf.org/practice/challenge/470>

to convert long integer to flag format:

```
pt_bytes = pt_long.to_bytes((pt_long.bit_length() + 7) // 8, 'big')
print("flag: ", pt_bytes.decode())
```

or, with crypto library:

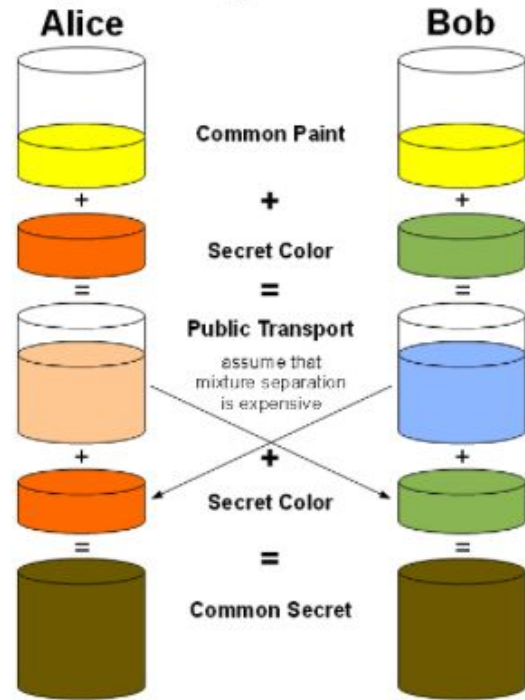
```
from Crypto.Util.number import long_to_bytes
pt_bytes = long_to_bytes(pt_long)
print("flag: ", pt_bytes.decode())
```

more resources + challs

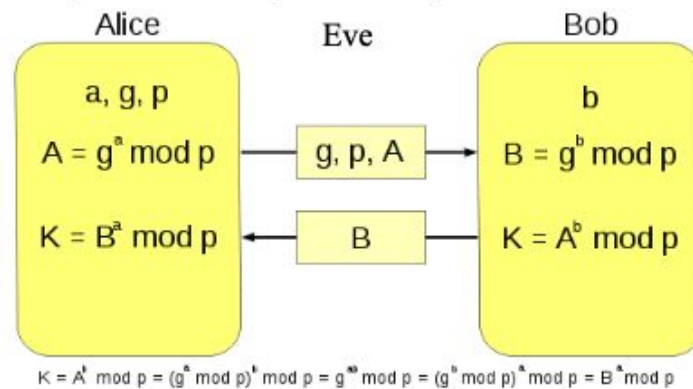
- <https://cryptopals.com/>
- <https://cryptohack.org/>
- <https://play.picoctf.org/practice/challenge/162>

diffie hellman intro

Use of Colours as Opposed to Numerical Values:



Use of Mathematical Sequences Portrayed via Variables:



03

what's next?

other cryptographic methods exist

- elliptic curves
- lattice-based

quantum...

- nothing is safe
- except maybe lattice-based crypto
- a lot of research into post quantum cryptography (PQC)

thank you



CREDITS: This presentation template was created by [Slidesgo](#), and includes icons, infographics & images by [Freepik](#)