Cryptography and Cracking

October 9, 2024

Outline

1. Intro to Cryptography
    a. What is cryptography, and how does it protect information?
2. Classical and Modern Cryptography
    a. How did cryptography evolve from pen-and-paper to computer algorithms?
3. Attacking Modern Cryptography
    a. How is information secured today, and how can we break it?

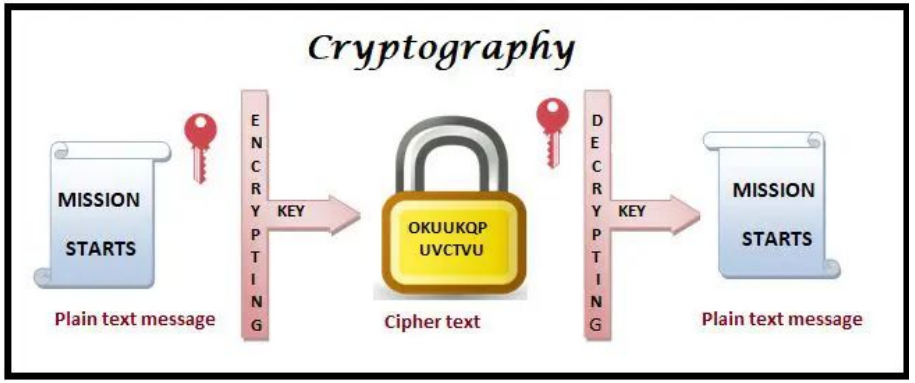
1. Cryptography

How do you stop people from reading your stuff?

How do you stop people from reading your stuff?

- Come up with a system to “hide” information from people not intended to receive it

- **Plaintext** - the unencrypted information
- **Ciphertext** - the encrypted information
- **Cipher/Algorithm** - The algorithm/process used to encrypt/decrypt information

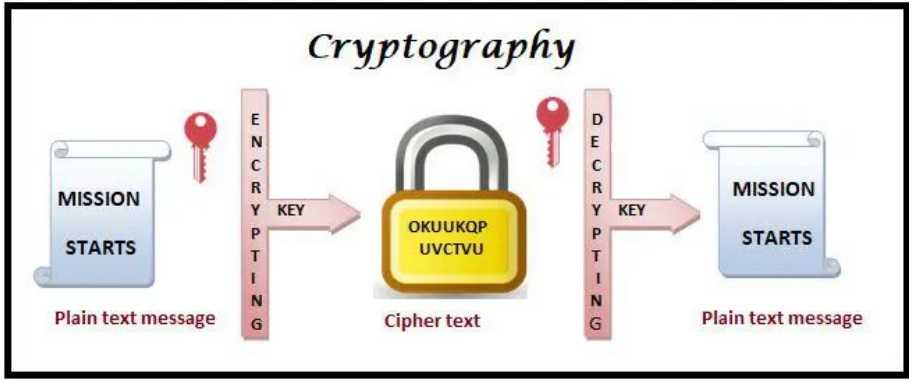


How do you stop people from reading your stuff?

Objectives of Cryptography

1) Hide the original message being sent (the main goal)

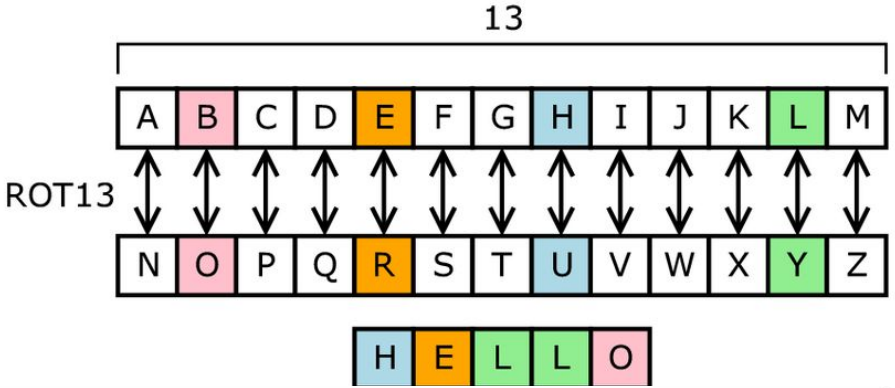
2) Hide additional information about the message that might hint towards its contents
   a) Think: is there anything in the message itself that tells me what it COULD be saying?



The Old Solution (Classical Ciphers)

Classical Cipher - Cryptography conducted mainly through pen and paper

- **Substitution Cipher** - Uses a system of replacing each letter/character with another one
- **Transposition Cipher** - Rearrange the letters in a way that hides the original message



Plain text: MEET ME AFTER THE TOGA PARTY

Row 1: M M T H G R

Row 2: E T E F E T E O A A T

Row 3: E A R T P Y

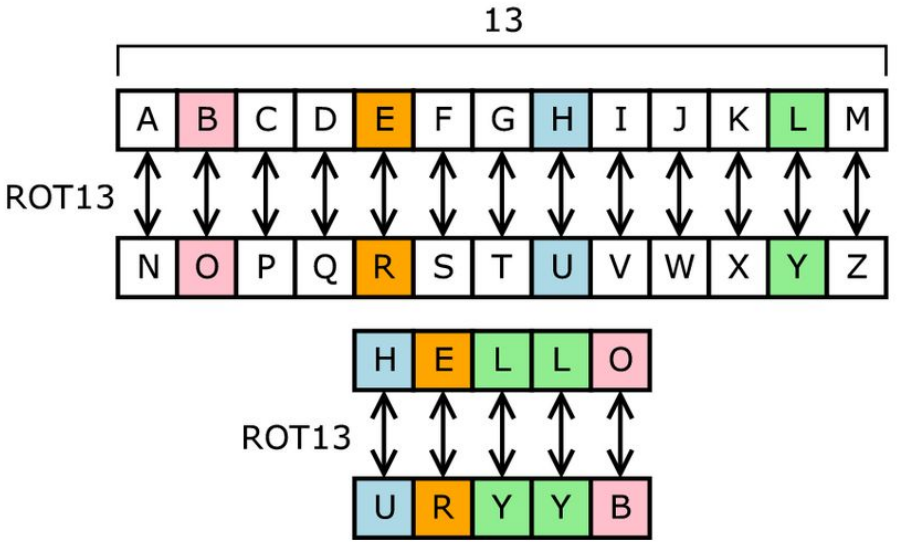
Cipher Text: MMTHGR ETEFETEOAAT EARTPY

Classical Shift Ciphers

- **Caesar Cipher:** Easiest and most well-known substitution cipher
  - “Rotate” the letters by a certain amount to encrypt and decrypt

- **ROT13** - a “symmetric” shift of the letters 13 places

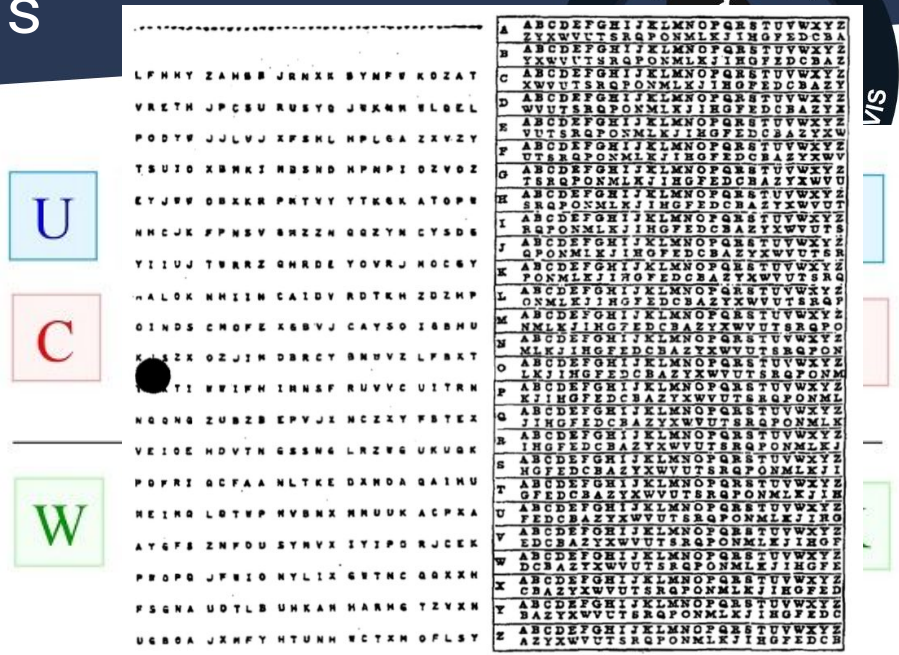
https://gchq.github.io/CyberChef/



Classical Shift Ciphers

- **Vigenere Cipher:** What if we shifted each character by a different amount?
  - Given a “key,” shift each letter by the corresponding number value on the key

- **One-Time Pad:** A “perfectly secret” cipher that uses a perfectly random key

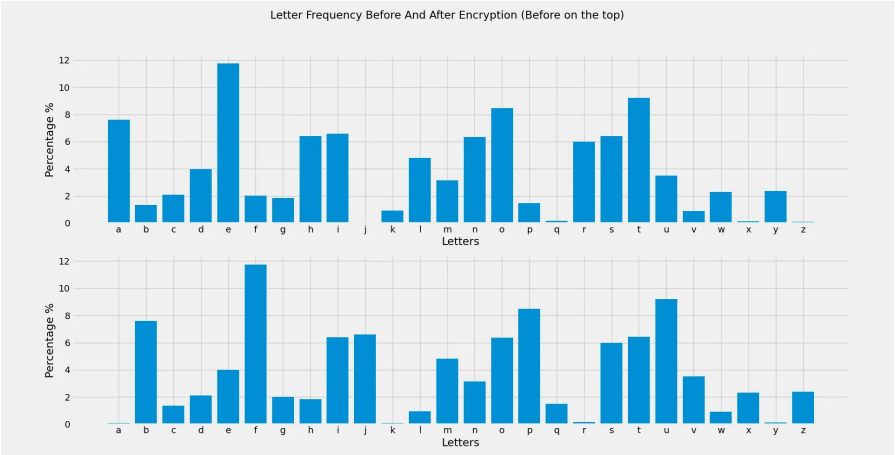


https://gchq.github.io/CyberChef/

So Why Do Classic Ciphers Suck?

Cryptanalysis - Study of a cipher to find weaknesses that allow it to be decoded by an attacker

- Frequency Analysis - check how often letters show up in encrypted text vs in plain English
- Known-plaintext Attacks - Use “cribs” to figure out what plaintext encrypts to a specific ciphertext
- Good for by-hand encryption, but defeatable with statistics **and** computers!



2. Modern Cryptography

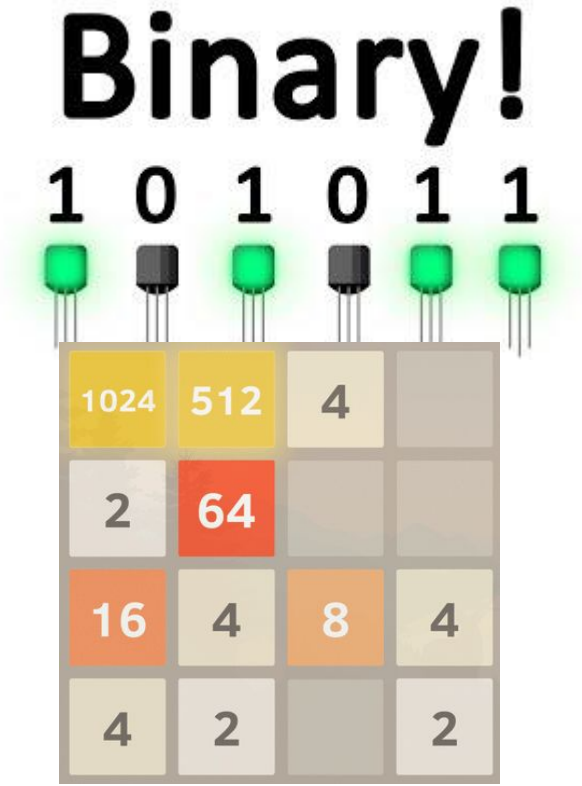
What can be done to improve on classical ciphers?

Step 1: Encoding for Computers

How do you represent numbers and letters inside a computer?

- Binary/Base 2

OK, so you can make numbers. How do you make letters then?



Step 1: Encoding for Computers

Side Note:

- The smallest “chunk” of memory that a computer can ask for is **1 byte** (8 binary bits, 1 character)

- It becomes useful to represent binary numbers with another power of 2
  - 2 Hex digits conveniently represent 1 byte AND use less digits

Denary/Decimal	Binary	Hexadecimal
Base 10 Number System	Base 2 Number System	Base 16 Number System
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Step 1: Encoding for Computers

Answer: Convert numbers to letters

- **Encoding standards** set what computer “numbers” represent which characters
  - ASCII - The classic, but only has 127 characters
  - UTF-8 - Supports all characters, the standard for displaying text

Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	|
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

Step 1: Encoding for Computers

Exercise: What characters are being encoded in Hex?

Use CyberChef.

https://zh.wikipedia.org/wiki/%E5%8E%9F%E7%A5%9E

YW1vbmd1c3N1c21vcmJpdXNzdXM=

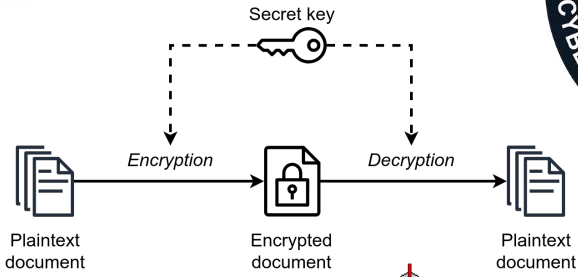
- Looks like a code, but in fact it’s another type of encoding!

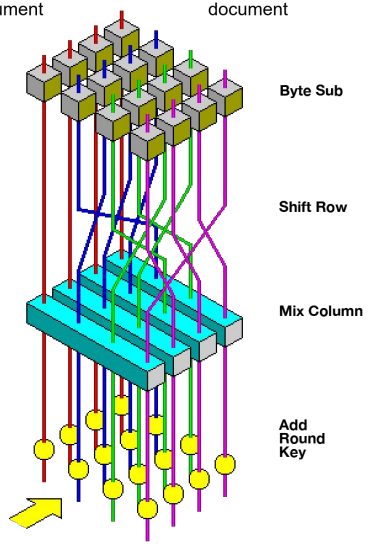
- Base64!

Step 2: Convert the Bits

Symmetric

- The same key is used to encrypt and decrypt the plaintext.
  - Simple XOR cipher, RC4, AES

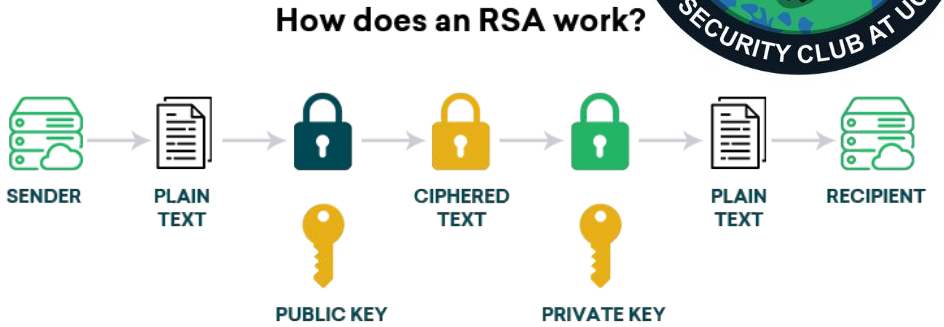




Step 2: Convert the Bits

Asymmetric *aka Public-Key Cryptography*

- Uses different keys for encryption/decryption
- Relies on mathematical difficulties (EX: Factor of 2 primes) to make it hard for computers to crack it



- Used in combination with symmetric in order to achieve efficient security
  - Asymmetric used to share keys used in symmetric

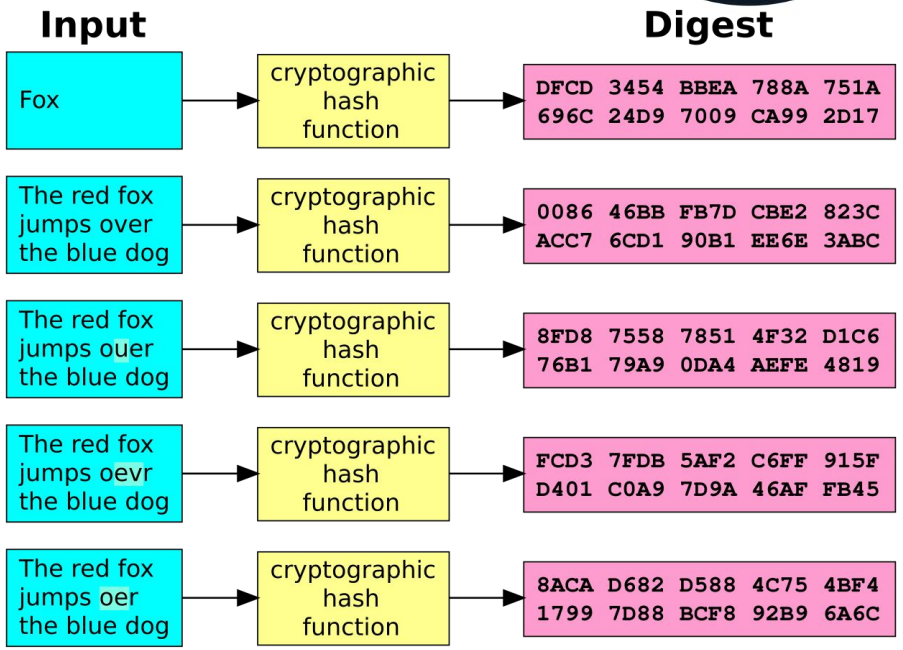
Step 2.5: Hash Functions

One-Way Function

- Easy to do an operation one way, but “hard” to do it the other way
- Can we use those functions to do OTHER things?

Hashing

- Produces a “fingerprint” for a given stream of bits that can’t be used to recover those bits!
- Used to “hash” passwords for secure storage, check if files are corrupted, etc.



3.

Attacking Modern Cryptography

How can we attack hashes if we can’t reverse them?

Guess and Check

- If it's not practical to "reverse" the hash algorithm, then why don't we just guess it?

- It's how "password cracking tools" work:
  - Given an encrypted hash, guess the password that goes with it

```
Session..........: hashcat
Status...........: Cracked
Hash.Name........: SolarWinds Serv-U
Hash.Target......: e983672a03adcc9767b24584338eb378:00
Time.Started.....: Sun May 23 11:43:13 2021 (1 sec)
Time.Estimated...: Sun May 23 11:43:14 2021 (0 secs)
Guess.Mask.......: ?a?a?a?a?a?at [7]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........: 24620.9 MH/s (32.19ms) @ Accel:32 Loops:1024 Thr:1024 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 31606272000/735091890625 (4.30%)
Rejected.........: 0/31606272000 (0.00%)
Restore.Point....: 0/857375 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:35840-36864 Iteration:0-1024
Candidates.#1....: 4{,erat -> cyr ~}t
Hardware.Mon.#1..: Temp: 62c Fan: 31% Util:100% Core:1920MHz Mem:7000MHz Bus:16
```

Length of Password	Combinations	Time to Crack (yrs)	Time to Crack (s)
4	456976	0.0	0.000228488
5	11881376	0.0	0.005940688
6	308915776	0.0	0.154457888
7	8031810176	0.0	4.015905088
8	208827064576	0.0	104.4135323
9	5429503678976	0.0	2714.751839
10	141167095653376	0.0	70583.54783
11	3670344486987780	0.1	1835172.243
12	95428956661682200	1.5	47714478.33
13	2481152873203740000	39.3	1240576437
14	64509974703297200000	1022.8	32254987352
15	1677259342285730000000	26592.8	8.3863E+11
16	436087428994289000000000	691412.1	2.18044E+13
17	11338273153851500000000000	17976714.2	5.66914E+14
18	2947951020001390000000000	467394568.1	1.47398E+16

People Are Bad At Making Passwords

- People tend to use things they can actually remember to make passwords
  - “password” and other variations
  - Easy number patterns i.e. “12345”
  - The current year
  - **Actual words**

- Instead of guessing every possible combination, we can narrow it down A LOT

Probably not your first choice of password

	2019		2020		2021		2022	
	Password	Number of users	Password	Number of users	Password	Number of users	Password	Number of users
1	12345	2,812,220	123456	2,543,285	123456	103,170,552	password	4,929,113
2	123456	2,485,216	123456789	961,435	123456789	46,027,530	123456	1,523,537
3	123456789	1,052,268	picture1	371,612	12345	32,955,431	123456789	413,056
4	test1	993,756	password	360,467	qwerty	22,317,280	guest	376,417
5	password	830,846	12345678	322,187	password	20,958,297	qwerty	309,679
6	12345678	512,560	111111	230,507	12345678	14,745,771	12345678	284,946
7	zinch	483,443	123123	189,327	111111	13,354,149	111111	229,047
8	g_czechout	372,278	12345	188,268	123123	10,244,398	12345	188,602
9	asdf	359,520	1234567890	171,724	1234567890	9,646,621	col123456	140,505
10	qwerty	348,762	senha	167,728	1234567	9,396,813	123123	127,762
11	1234567890	329,341	1234567	165,909	qwerty123	8,933,334	1234567	110,279
12	1234567	261,610	qwerty	156,765	000000	8,377,094	1234	106,929

People Are Bad At Making Passwords

- Instead of guessing every combo, why don't we try lists of common passwords and combinations?

- rockyou.txt - The most well known password wordlist!
  - 32 MILLION unencrypted passwords breached
  - In every Kali installation
  - (Most cyber competition challenges will have you use this)

rockyou

```
jason@kali:/usr/share/wordlists$ ls -l
total 188756
lrwxrwxrwx 1 root root        26 Jul 22 23:33 amass → /usr/sh
lrwxrwxrwx 1 root root        25 Jul 22 23:33 dirb → /usr/sha
lrwxrwxrwx 1 root root        30 Jul 22 23:33 dirbuster → /us
lrwxrwxrwx 1 root root        41 Jul 22 23:33 fasttrack.txt →
lrwxrwxrwx 1 root root        45 Jul 22 23:33 fern-wifi → /us
lrwxrwxrwx 1 root root        28 Jul 22 23:33 john.lst → /usr
lrwxrwxrwx 1 root root        27 Jul 22 23:33 legion → /usr/s
lrwxrwxrwx 1 root root        46 Jul 22 23:33 metasploit → /u
ts
lrwxrwxrwx 1 root root        41 Jul 22 23:33 nmap.lst → /usr
-rw-r--r-- 1 root root 139921507 Jul 17  2019 rockyou.txt
-rw-r--r-- 1 root root  53357329 May 12  2023 rockyou.txt.gz
lrwxrwxrwx 1 root root        39 Jul 22 23:33 sqlmap.txt → /u
lrwxrwxrwx 1 root root        25 Jul 22 23:33 wfuzz → /usr/sh
lrwxrwxrwx 1 root root        37 Jul 22 23:33 wifite.txt → /u
jason@kali:/usr/share/wordlists$
```

People Are Bad At Making Passwords

```
jason@kali:/usr/share/wordlists$ grep 'password123' rockyou.txt
password123
password1234
password12345
password123456789
password123456
mypassword123
password1234567
password12345678
123password123
password1234567890
password1235
password1232
newpassword123
password123@
password12345678910
password1234567-
password1234321
password123xx
password123w
password123awaywego
password123a
password123C
password123987
password1239
password1238\\'
password12380
password123789
password12356//
password123555no
password1234??123?
password12345?
password123456__
password1234567899
password123456.
password123321`
password1233
password123123
password1231
password1230
```

Hashcat Demo

- Try and crack *sample_hashes.txt*
- (IF TIME) NTLM Hash demo (*ntlm_sample.txt*)

Useful Resources:

- https://hashcat.net/wiki/doku.php?id=hashcat
  - Hashcat Manual
- Googling *"How to do x in hashcat"*

```
Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: D:\Jason\TA\Cybersecurity\sample_hashes.txt
Time.Started.....: Wed Aug 07 15:34:11 2024 (1 sec)
Time.Estimated...: Wed Aug 07 15:34:12 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (D:\Jason\NCL\Tools\rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........: 11494.1 kH/s (4.34ms) @ Accel:2048 Loops:1 Thr:32 Vec:1
Recovered........: 5/5 (100.00%) Digests (total), 5/5 (100.00%) Digests (new)
Progress.........: 11796480/14344384 (82.24%)
```

NTLM - The Windows Hash Format


```
Administrator:500:CEEB0FA9F240C200417EAF40CFAC29C3:D280553F0
103F2E643406517296E7582:::
```

Fields

1) Username
2) SID
3) LM Hash
4) NTLM Hash

Command

sample_hashes.txt

hashcat -m (hash type) *-a 0* (hash file) (wordlist) *--show*

ntlm_sample.txt

john --format=NT --rules -w=(wordlist) (hash file)

Hashcat Demo

OK, what if someone is using a password that isn't on the list?

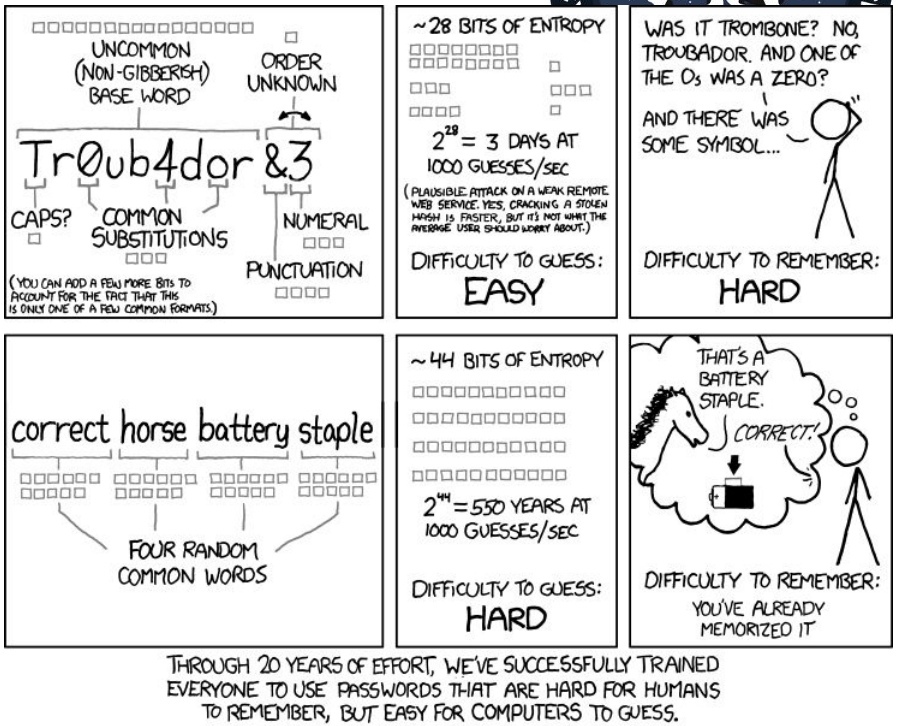
- MORE Wordlists!
  - Crackstation (15 **GB!**)
    - https://crackstation.net/crackstation-wordlist-password-cracking-dictionary.htm
  - Default Password Wordlists!
    - https://www.google.com/search?q=default+password+wordlists

- The possibilities (and required computing power) is ENDLESS!

Tips to Secure Your Passwords

1) Don't use anything that can be easily guessed!
   a) "password," "qwerty," and all its variations
   b) Your name, birthday, etc.
   c) If you're tired of changing stuff, use a password manager

2) Longer is better! The process of "breaking" passwords is literally to guess every possible combination
   a) "Special Characters" create extra work for the hacker!



Review

Cryptography

The study of hiding information from people who shouldn't read it.

Classical ciphers use pen and paper to hide data, but no longer work.

Modern Cryptography

Converts information to bits and bytes, and operates on those bits.

Modern encryption methods use mathematics and complex operations enabled by computers to hide data WHILE preventing the ciphertext from revealing clues about the plaintext.

Cracking Hashes

Involves a process of guessing and checking possible passwords against a password hash.

People are bad at making passwords, so it's a bit easier than "checking every single possibility."

Thanks!

Any questions?