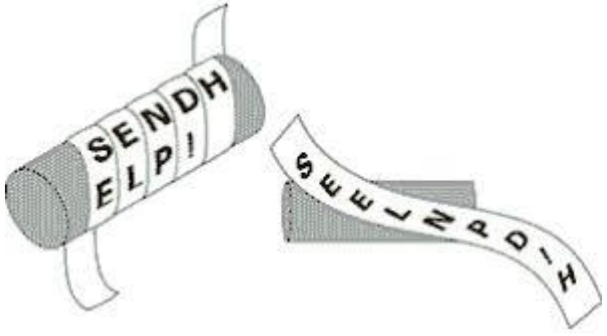
Security Tools Lab 1

Module 2 - Breaking of Simple Ciphers

History

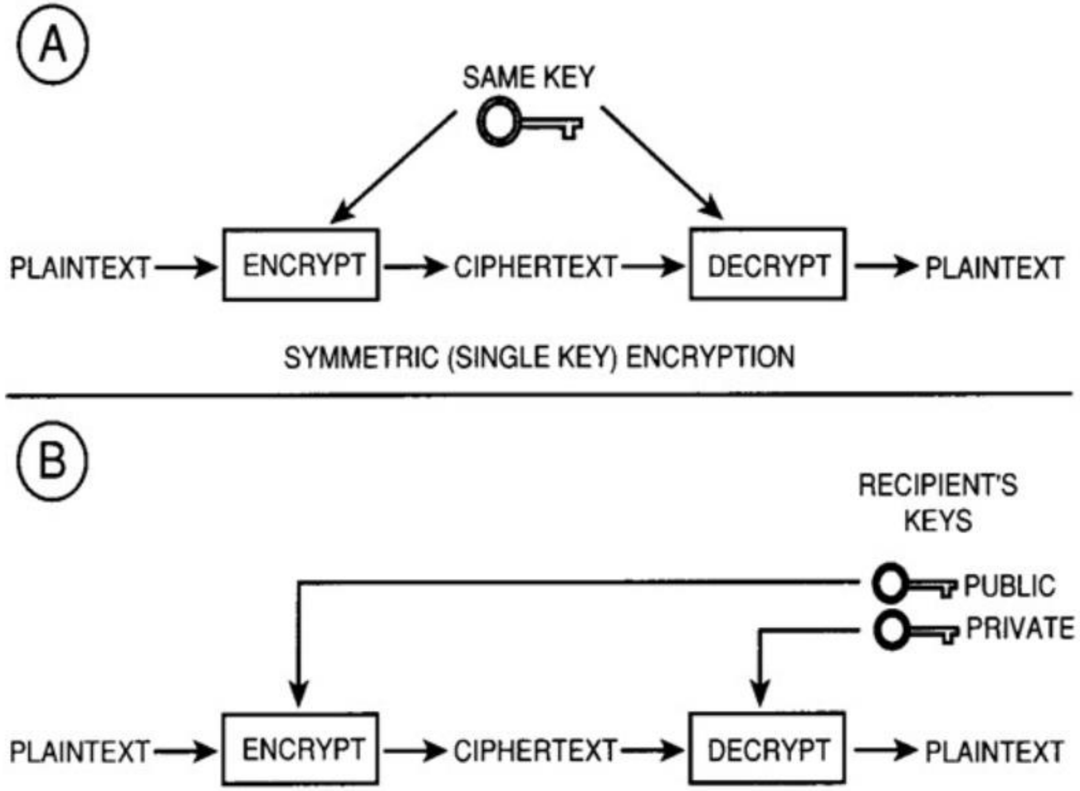
- Crypto (secret) graphy (writing)
  - Process of disguising a plaintext message into an unintelligible ciphertext
- Is it the solution to the security problem?
  - Or does it transform to a new problem(s)?
- How old?
  - 2000 BC – Encoded substitution Egyptian hieroglyphics
    - Used more for decorating tombs rather than concealing
  - 1500 BC - Mesopotamians encrypted cuneiform through substitution to hide information, eg, camouflaging the recipe for pottery glaze
  - 400 BC – Spartans using ‘Scytale’
    - Leather or paper wrapped around a rod
    - Transposition cipher, what is the key?

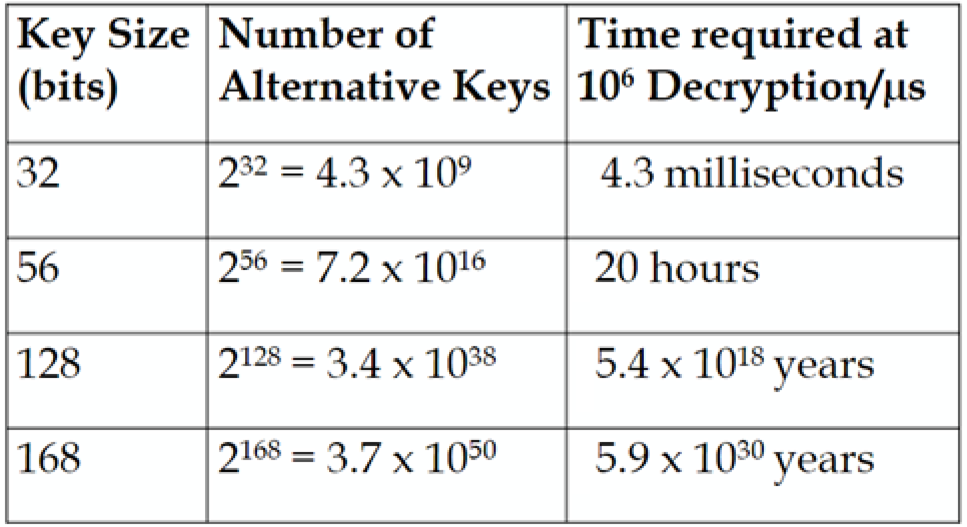
Hieroglyphic enciphering of proper names and titles, with ciphered on the left and plain equivalent on the right



Terminologies

- Plaintext
  - Original message
- Ciphertext
  - Transformed unintelligible message
- Cipher
  - Algorithm for transformation
- Key
  - Critical information known only to sender and receiver
- Cryptanalysis
  - Codebreaking, study of transforming ciphertext to plaintext without knowing the key

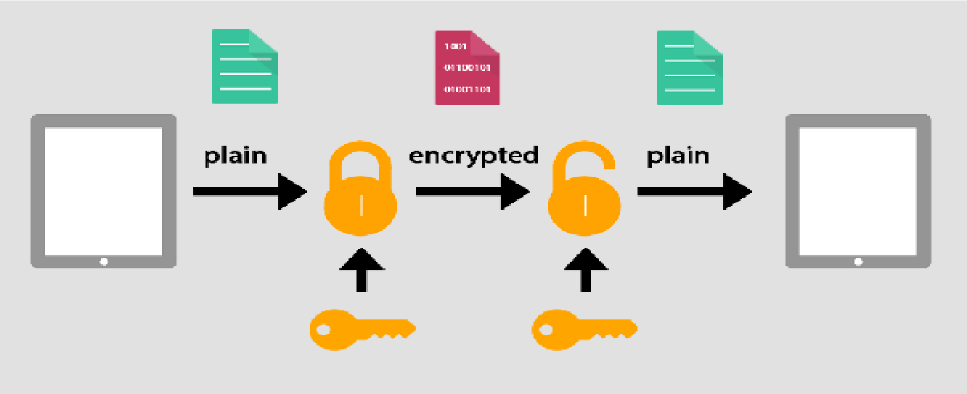


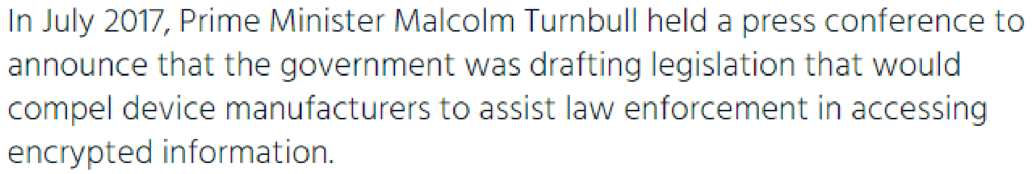


Key Size (bits)	Number of Alternative Keys	Time required at 10^6 Decryption/µs
32	$2^{32} = 4.3 \times 10^9$	4.3 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	20 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years

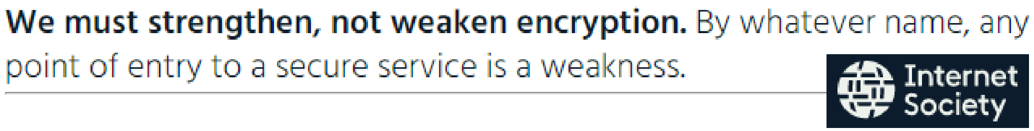
Legality

- Is encryption legal?
- Thin line : Privacy V/S National Security
- How to achieve control?
  - India’s limit on key length
  - China’s approval of encryption providers
  - Australia’s backdoor law
  - UK’s GCHQ ghost protocol (aka BCC law)
- Raises questions :
  - Is sharing really caring?
  - Weakening of cryptography?
  - Backdoor for one = backdoor for all
- Does heavy handed approach work?
  - Telegram banned in Russia
    - BUT still widely in use




In July 2017, Prime Minister Malcolm Turnbull held a press conference to announce that the government was drafting legislation that would compel device manufacturers to assist law enforcement in accessing encrypted information.

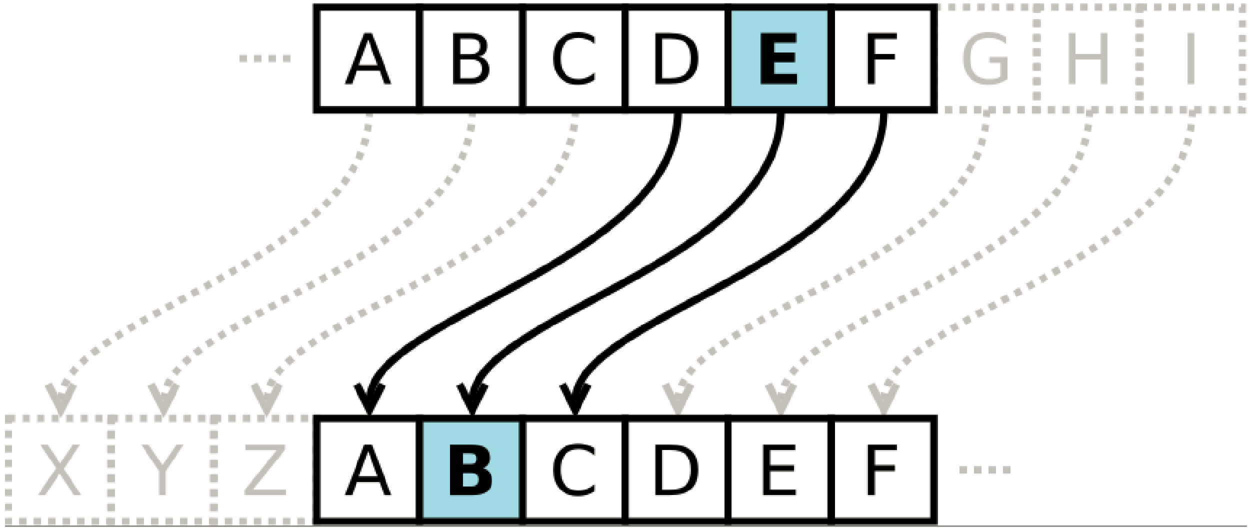
https://www.gp-digital.org/world-map-of-encryption/


We must strengthen, not weaken encryption. By whatever name, any point of entry to a secure service is a weakness.

Internet Society

Caesar’s cipher

- Monoalphabetic substitution cipher, where each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet, also known as a shift cipher
- This fixed number of positions is the secret key
- Original cipher : k = 3



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
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

Caesar's cipher

- Plaintext space P = Z_{26} = {0, 1, 2, ….., 24, 25}
  - $e_k(x) = (x+k) \bmod 26$ for $x \in P = y$
  - $d_k(y) = (y-k) \bmod 26$ for $y \in P = x$
- Correspondence between interger Z and alphabet
  - 0 -> A, 1 -> B, … , 24 -> Y, 25 -> Z
- Key k = 11
  - P = WEWILLMEETATMIDNIGHT
  - 22 4 22 8 11 11 12 4 4 19 0 19 12 8 3 13 8 6 7 19
  - Add 11 and reduced to mod 26
    - 7 15 7 19 22 22 23 15 15 4 11 4 23 19 14 24 19 17 18 4
    - C = HPHTWWXPPELEXTOYTRSE
- Cryptanalysis
  - Brute force performed by trying all 25 possible keys

k = 0 : HPHTWWXPPELEXTOYTRSE

k = 1 : GOGSVVWOODKDWSNXSQRD

k = 2 : FNFRUUVNNCJCVRMWRPQC

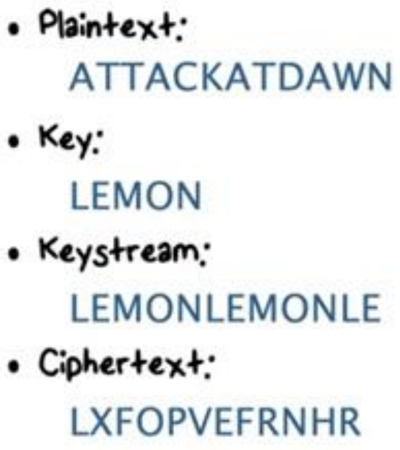
k = 3 : ….

k = x : ….

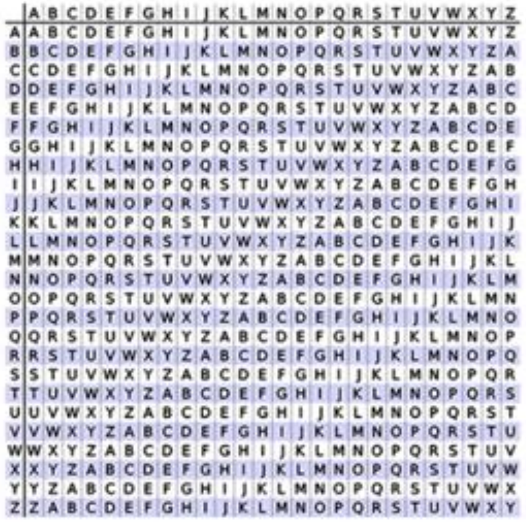
k = 11 : WEWILLMEETATMIDNIGHT

Substitution Cipher

- Monoalphabetic
  - Alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - Key: PDKIFMRBHSONCGXUTJWEYLQAZV
  - Cryptanalysis developed ~850 AD (Alkindus)
    - Broke Caesar’s cipher with ease
- Polyalphabetic
  - Can be substituted by different alphabets in different cases
  - Vigenere cipher, Enigma machines are more modern examples



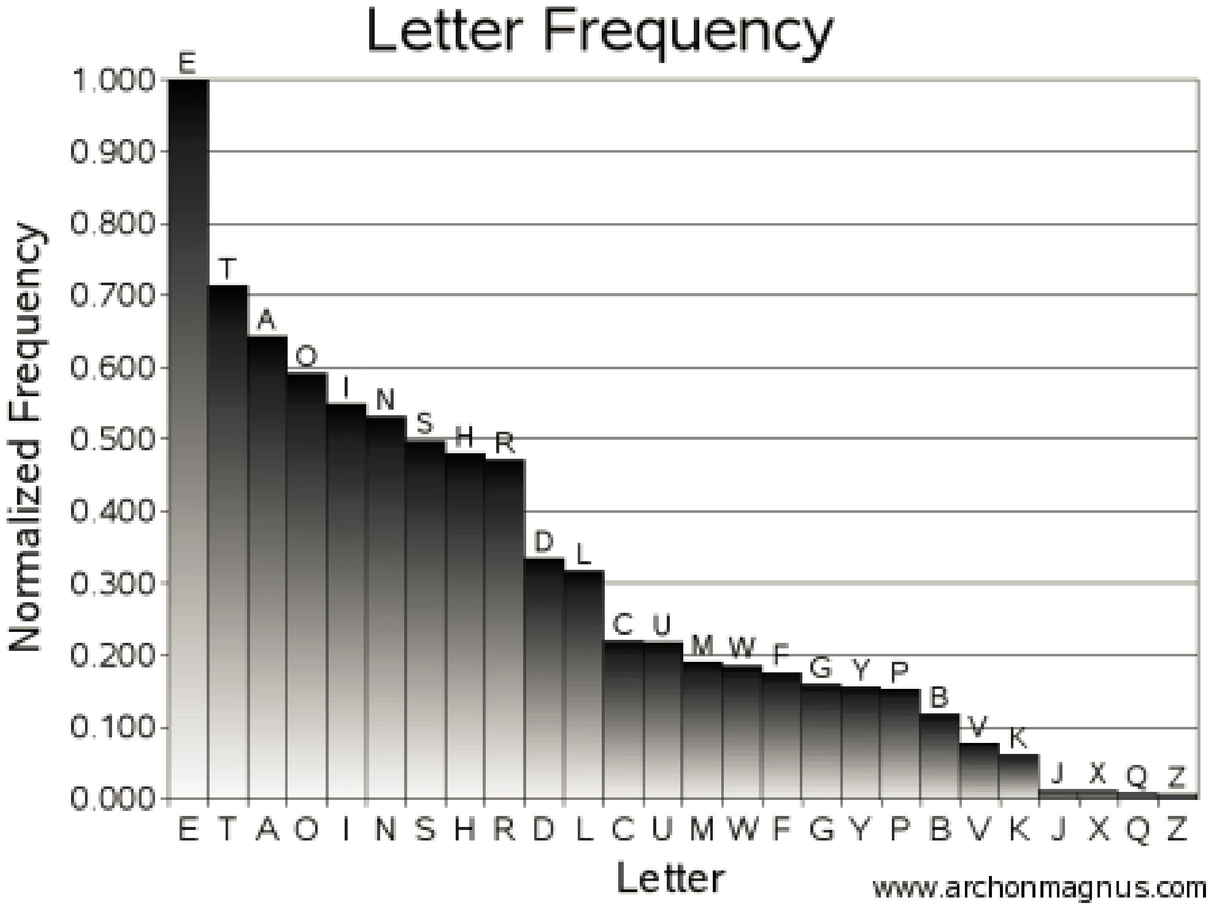
- Plaintext:
  ATTACKATDAWN
- Key:
  LEMON
- Keystream:
  LEMONLEMONLE
- Ciphertext:
  LXFOPVEFRNHR



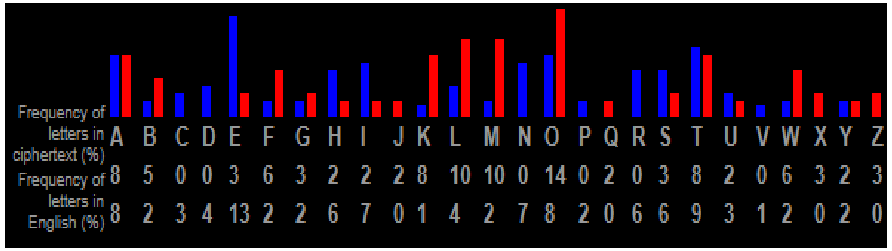
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T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

Substitution Cipher

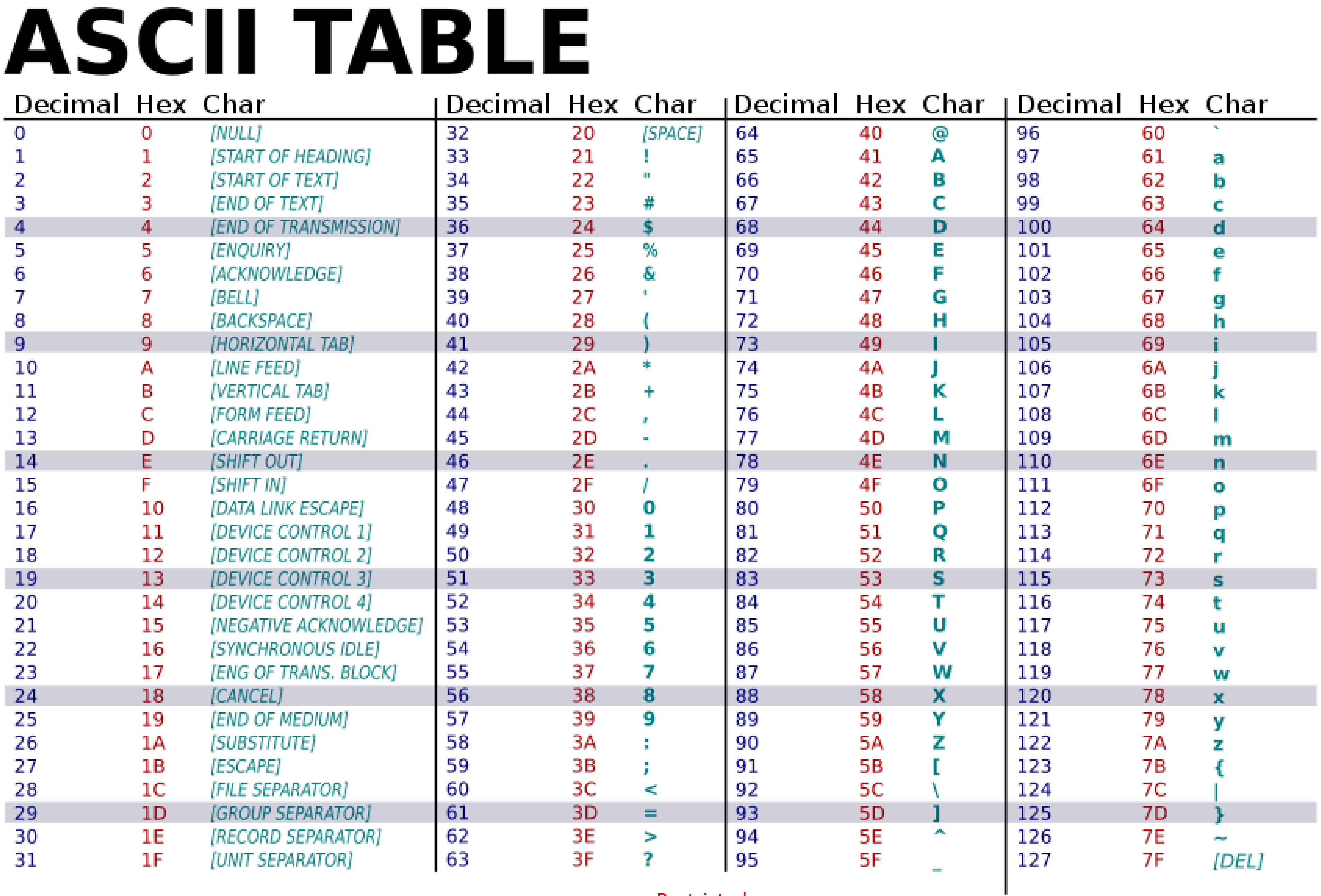
- Trivial to break with letter frequency



OM OL XTKB MKOXOAS MG ZKTAQ A LWZLMOMWMOGF EOHITK WLOFU YKTJWTFEB AFASBLOL



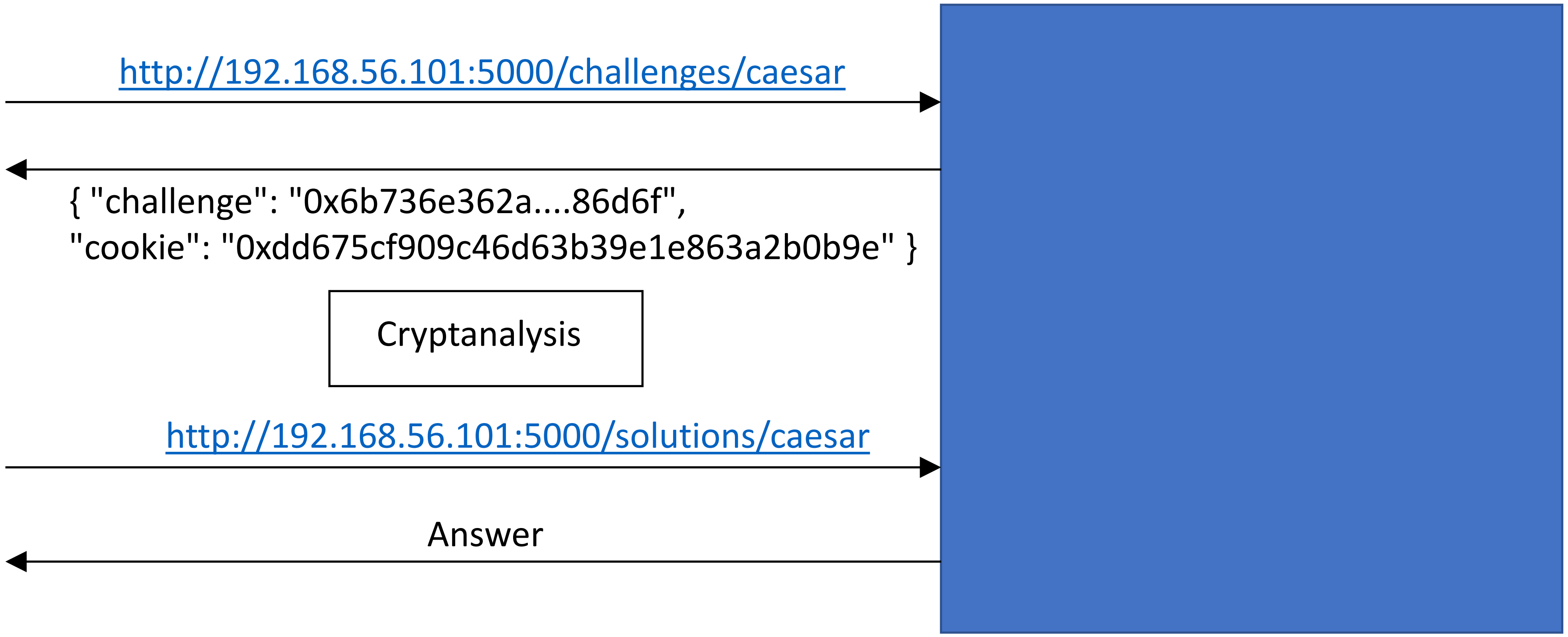
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Frequency of letters in ciphertext (%)	8	5	0	0	3	6	3	2	2	2	8	10	10	0	14	0	2	0	3	8	2	0	6	3	2	3
Frequency of letters in English (%)	8	2	3	4	13	2	2	6	7	0	1	4	2	7	8	2	0	6	6	9	3	1	2	0	2	0



ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Challenge Server



http://192.168.56.101:5000/challenges/caesar

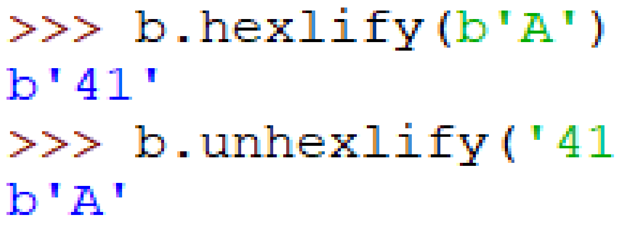
{ "challenge": "0x6b736e362a....86d6f",
"cookie": "0xdd675cf909c46d63b39e1e863a2b0b9e" }

Cryptanalysis

http://192.168.56.101:5000/solutions/caesar

Answer

Important methods



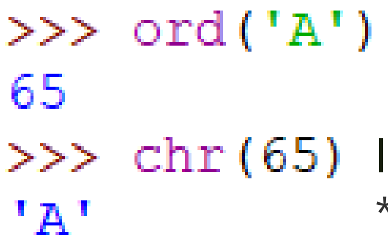
```
>>> b.hexlify(b'A')
b'41'
>>> b.unhexlify('41')
b'A'
```

Return the hexadecimal representation

Return the binary data represented by the hexadecimal string

Hexlify takes ‘bytearray’ while Unhexlify takes ‘string’

Both return ‘bytearray’



```
>>> ord('A')
65
>>> chr(65)
'A'
```

Return an integer of the given single Unicode character (decimal, not hex)

Inverse of ord() function, takes Unicode decimal and returns a string.
*Not the same as (un)hexlify

Setup

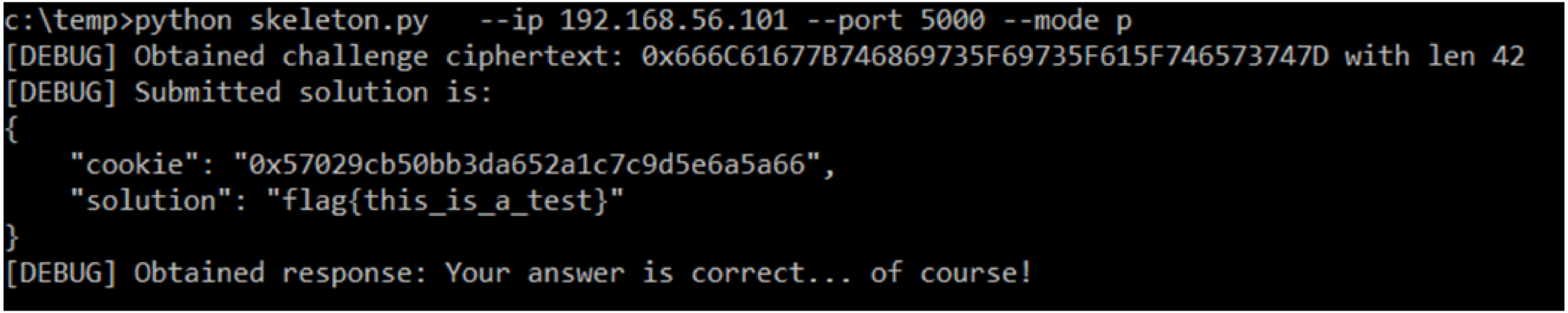
- Download VM from dropbox
- Double click VM file to open and import into VirtualBox
- Ensure network settings in VM are set to “Host-only Adapter”
- Run the VM and check file -> preferences -> Network – Host-only network
- Run Nmap on Kali to find IP and Port
- Access challenger server @ http://<IP>:<Port>
- Edit the provided python code (skeleton.py) to extract this hex ciphertext to convert it into plaintext

Caesar's challenge

- Convert the ciphertext obtained from the server into its plain text by using the Caesar’s method
- Replace the characters of the ciphertext with characters by changing the key. You need to use brute force in this case. Be careful of special character for new line and space.
- You will finally be able to discover a story hidden inside the ciphertext once you have converted it to plaintext.

To run the code

- Use the skeleton python file to carry out your exercise.
- Run it from the command line



```
c:\temp>python skeleton.py   --ip 192.168.56.101 --port 5000 --mode p
[DEBUG] Obtained challenge ciphertext: 0x666C61677B746869735F69735F615F746573747D with len 42
[DEBUG] Submitted solution is:
{
    "cookie": "0x57029cb50bb3da652a1c7c9d5e6a5a66",
    "solution": "flag{this_is_a_test}"
}
[DEBUG] Obtained response: Your answer is correct... of course!
```

--ip 192.168.56.101 --port 5000 followed by:

--mode p for plain
--mode c for Caesar
--mode s for substitution
--mode o for otp

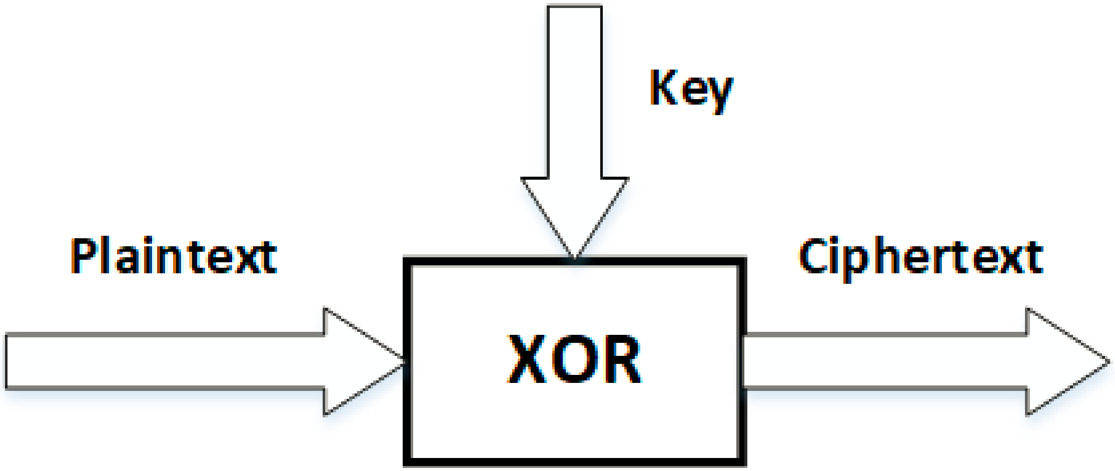
- You can also use Pycharm if you wish.

Substitution Cipher

- Convert the ciphertext obtained from the server into its plain text by using the substitution method
- Replace the characters of the ciphertext with characters of the letter frequency + special characters like “ ”, “,”, “?”, “!” etc.
- You will finally be able to discover a story hidden inside the ciphertext once you have converted it to plaintext. A story that you will be quite familiar with ;)

One-Time Pad (OTP)

- Key is random, is long as plaintext (at least), and is used only **ONCE**



E : P ⊕ K = C

D : C ⊕ K = P

What happens if re-used
e.g key is <space>

$$\begin{array}{r} \text{ABC} \\ \oplus\ \underline{\quad\quad} \\ \hline \end{array}$$

- This scheme **cannot** be *broken*
- No matter how strong an adversary is, he cannot learn anything about plaintext

One-Time Pad (OTP)

- Currently the plaintext associated with the given cipher text is “Student ID 1000000 gets 0 points”
- Manipulate the cipher text such that the plaintext associated with it contains the 1000000 replaced with your own school ID and the score 0 replaced with the maximum score you wish to get for this lab (6) ;)
    - Eg. “Student ID 1000003 gets 6 points”
- Remember
    - Ptxt (xor) OTP = Ctxt : Encryption
    - Ctxt (xor) OTP = Ptxt : Decryption
    - …
- You know the plaintext and you will receive the ciphertext from the challenge server. You’ll have to perform a ‘known-plaintext’ attack