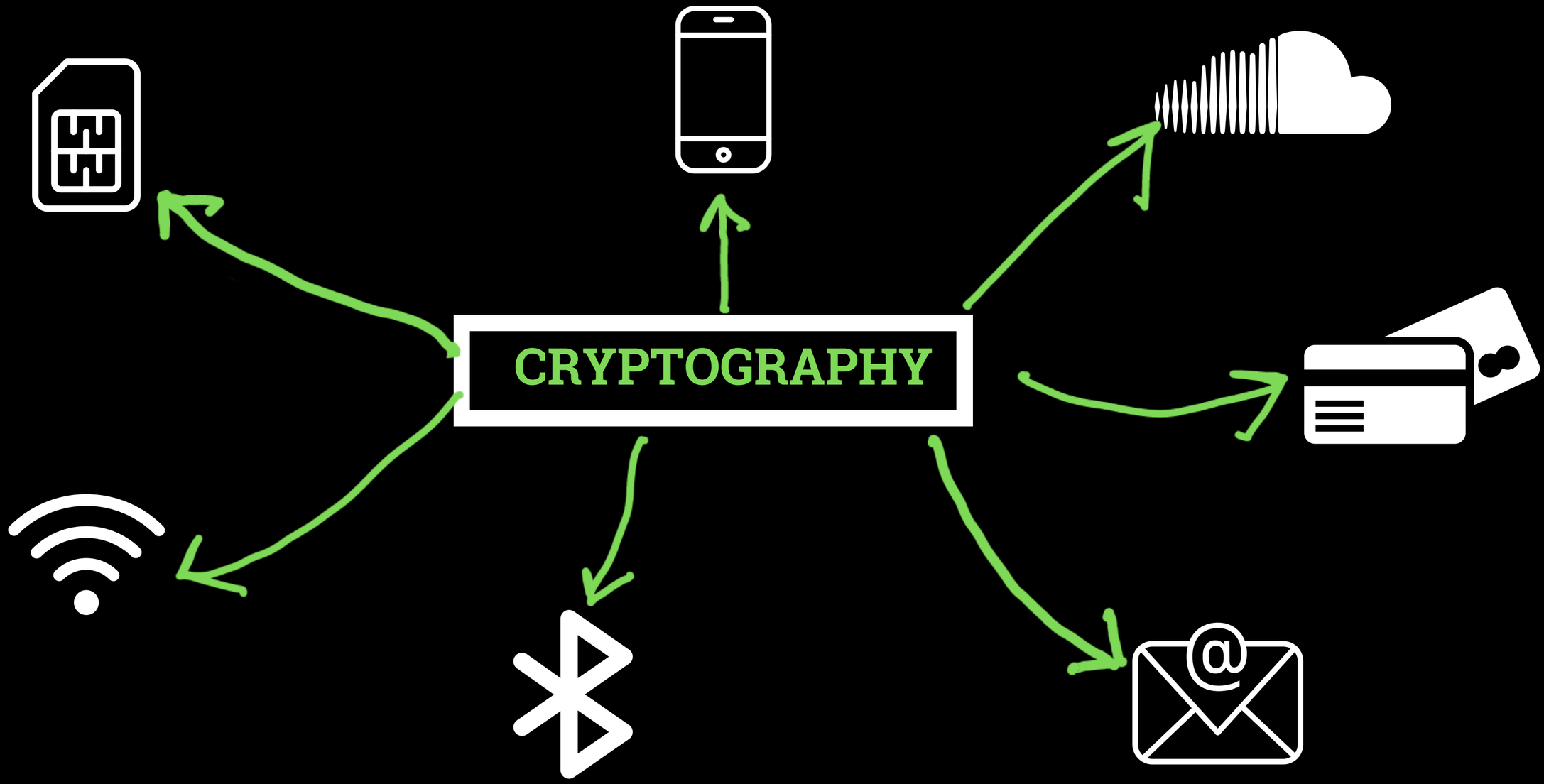
CRYPTOGRAPHY

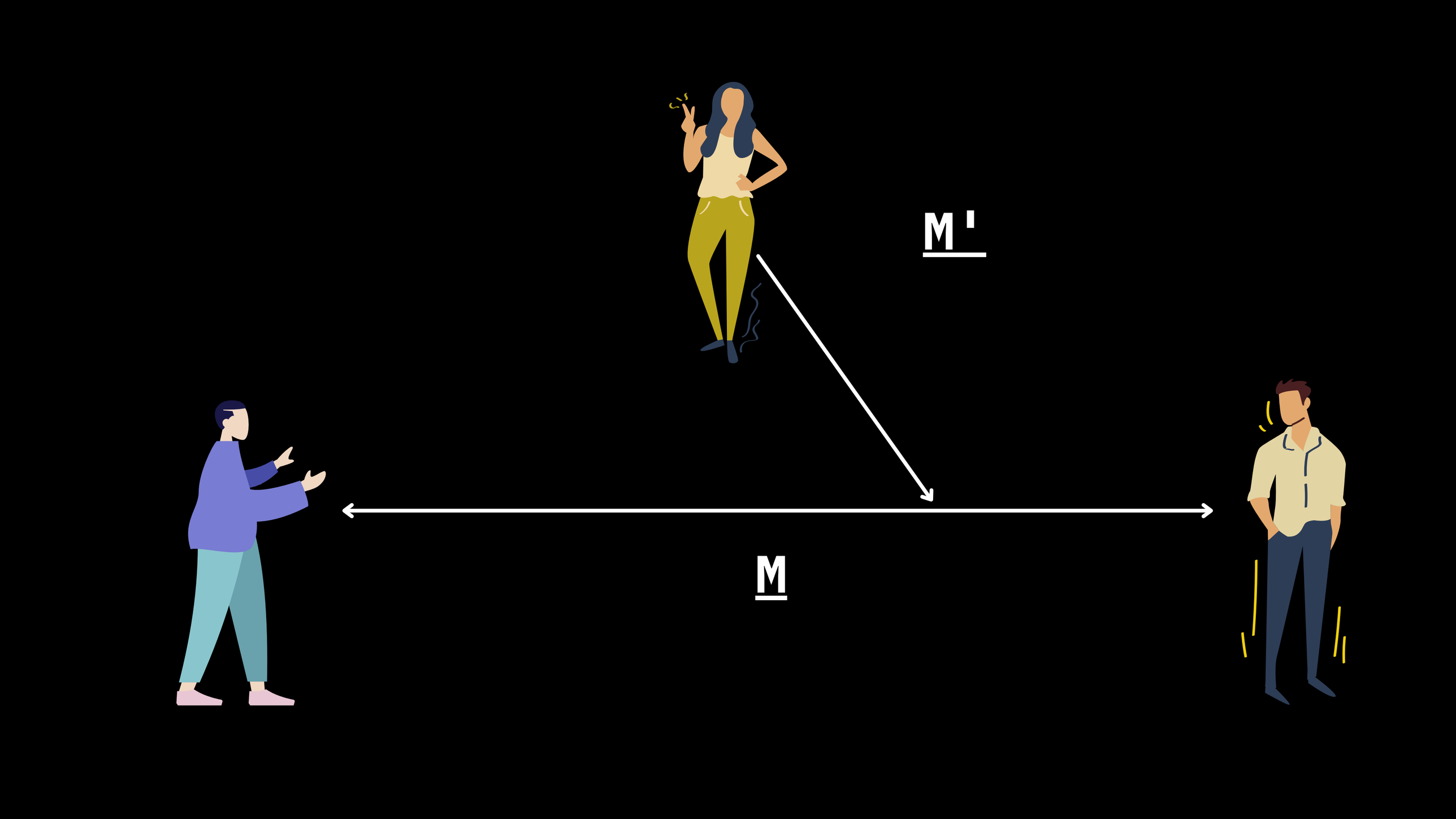
PART-I

CRYPTOGRAPHY IS EVERYWHERE!!



THE SETTING OF MODERN CRYPTO

- Alice and Bob are talking over an insecure channel
- How can Alice and Bob make sure that if anyone is intercepting their messages, they are unable to read them?


M'
M

THINGS WE WILL DISCUSS

- What is Cryptography ?
- Bits, Bytes and Hex
- Base64, Base32 and ASCII
- Cryptography principles and terms.
- Ancient Ciphers (Caesar cipher)
- Encoding VS Encryption

WHAT IS CRYPTOGRAPHY

- “The art of writing or solving codes”
- Keeping secret information secret – takes the form of a cryptosystem, a scheme which specifies methods for encrypting and decrypting information.

WHY IS CRYPTOGRAPHY NEEDED?

For a secure world

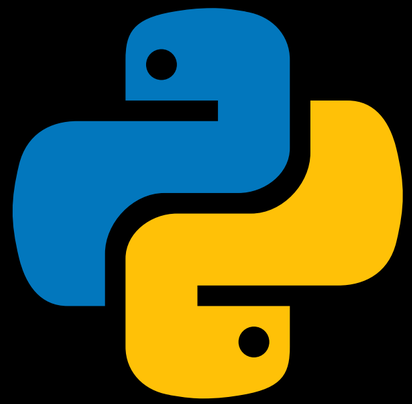
To know more, checkout this video

STUFF YOU MAY ENCOUNTER

There are so many standards!!

We will be discussing the following:

- Bits and Bytes
- Hex
- Base32 and Base64
- ASCII



BITS AND BYTES

- Computers use bits to represent everything, since transistors lend themselves nicely to holding either a “1” or “0” value.
- **Bit** = **b**inary dig**it** (binary is base 2)
- e.g. 0b101 = $(101)_2 = 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = (5)_{10}$
- A byte is 8 bits, and so its maximum value is $(1111\ 1111)_2 = 255$
- Bits and bytes are the building blocks of computer data; almost always you will see bits in multiples of 8 (bytes), because it is the standard we have adopted for designing our hardware.

HEXADECIMAL (A.K.A HEX)

- Hexadecimal is another common way to represent bytes, it's base 16. Since we only have 10 arabic numerals, we augment this with "A", "B", "C", "D", "E", and "F" (case doesn't matter).
  - "A" = 10, "B" = 11, ... "F" = 15
- Since $16 = 2^4$, that means that one hex character equals 4 bits exactly. Thus, one byte can be represented by two hex characters.
- e.g. 0x4e = $(4e)_{16} = 4 * 16^1 + 14 * 16^0 = (78)_{10}$

HEXADECIMAL (A.K.A HEX)

- We mostly look at bytes in hex, because it takes up significantly less space than bits (and it’s easier to count off two hex chars than it is to count off 8 bits)
- And looks nice:

01001000 01000101 01001100 01001100 01001111 00100000
01010111 01001111 01010010 01001100 01000100

48 45 4c 4c 4f 20 57 4f 52 4c 44

BASES
BINARY-TO-TEXT ENCODING

- BASE - n is just n characters mapping to n whole numbers.
- So, BASE - 2 is binary (0s & 1s), BASE - 16 is Hex (0-9, A-F) and similarly:

- **BASE32**: 32 characters - A-Z , 2-7
- **BASE64**: 64 characters - A-Z, a-z, 0-9, +, /
  (There is also a URL safe variant that has -, _ instead of +, /)

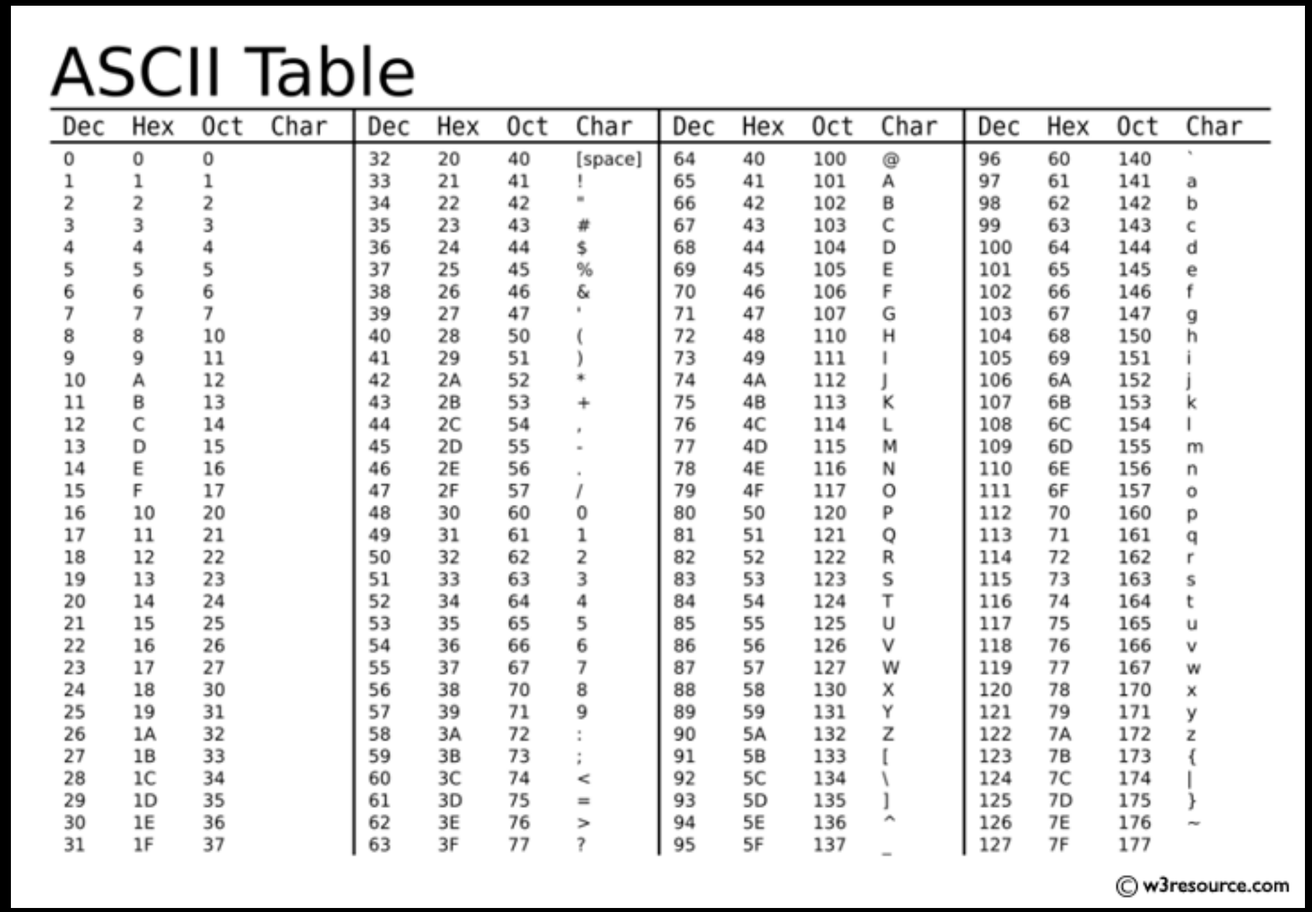
BASES

- Specifically **base64** is a popular form of encoding and can be seen at many places, e.g. in URLs, in PEM files (more on that later) etc.

- Let us see it in **Python** !

ASCII

- American Standard Code for Information Interchange
- We use one byte for each character, with the values listed in the table.
- e.g. “A” = 65
- 0x4e = “N”



ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

IS IT CRYPTOGRAPHY?

NO !

There’s no key for this, so anyone can do these operations, and it provides no security to our message.
For that matter, these representations are all essentially equivalent to a computer -- they only look different to us.

CRYPTOGRAPHY PRINCIPLES AND TERMS

- **Plaintext**: the message to be encrypted, like “ATTACK_AT_DAWN”.
- **Ciphertext**: the encrypted message (corresponds to a plaintext).
- **Key**: a “parameter” to a cryptosystem that is used in the encryption or decryption process, upon which its security depends.
- **Kerckhoff’s Principle**: a cryptosystem should be secure even if everything about it is public knowledge, except for the key.
- **Keyspace**: the set of all possible keys for a cryptosystem.
- **Cryptanalysis**: the art of deciphering coded messages without being told the key

CRYPTOGRAPHY PRINCIPLES AND TERMS

- **Plaintext**: the message to be encrypted, like “ATTACK_AT_DAWN”.
- **Ciphertext**: the encrypted message (corresponds to a plaintext).
- **Key**: a “parameter” to a cryptosystem that is used in the encryption or decryption process, upon which its security depends.
- **Kerckhoff’s Principle**: a cryptosystem should be secure even if everything about it is public knowledge, except for the key.
- **Keyspace**: the set of all possible keys for a cryptosystem.
- **Cryptanalysis**: the art of deciphering coded messages without being told the key

CRYPTOGRAPHY PRINCIPLES AND TERMS

- **Plaintext**: the message to be encrypted, like “ATTACK_AT_DAWN”.
- **Ciphertext**: the encrypted message (corresponds to a plaintext).
- **Key**: a “parameter” to a cryptosystem that is used in the encryption or decryption process, upon which its security depends.
- **Kerckhoff’s Principle**: a cryptosystem should be secure even if everything about it is public knowledge, except for the key.
- **Keyspace**: the set of all possible keys for a cryptosystem.
- **Cryptanalysis**: the art of deciphering coded messages without being told the key

CRYPTOGRAPHY PRINCIPLES AND TERMS

- **Plaintext**: the message to be encrypted, like “ATTACK_AT_DAWN”.
- **Ciphertext**: the encrypted message (corresponds to a plaintext).
- **Key**: a “parameter” to a cryptosystem that is used in the encryption or decryption process, upon which its security depends.
- **Kerckhoff’s Principle**: a cryptosystem should be secure even if everything about it is public knowledge, except for the key.
- **Keyspace**: the set of all possible keys for a cryptosystem.
- **Cryptanalysis**: the art of deciphering coded messages without being told the key

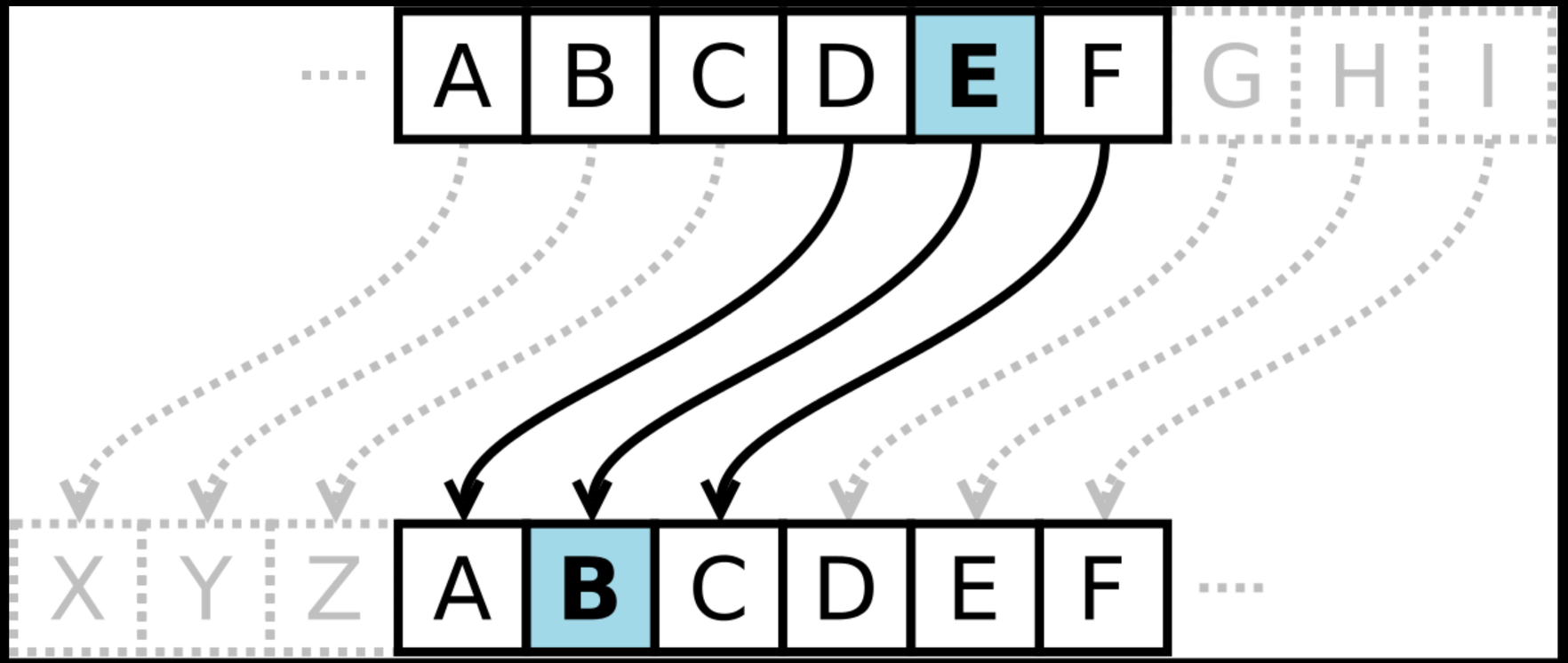
CRYPTOGRAPHY PRINCIPLES AND TERMS

- **Plaintext**: the message to be encrypted, like “ATTACK_AT_DAWN”.
- **Ciphertext**: the encrypted message (corresponds to a plaintext).
- **Key**: a “parameter” to a cryptosystem that is used in the encryption or decryption process, upon which its security depends.
- **Kerckhoff’s Principle**: a cryptosystem should be secure even if everything about it is public knowledge, except for the key.
- **Keyspace**: the set of all possible keys for a cryptosystem.
- **Cryptanalysis**: the art of deciphering coded messages without being told the key

CRYPTOGRAPHY PRINCIPLES AND TERMS

- **Plaintext**: the message to be encrypted, like “ATTACK_AT_DAWN”.
- **Ciphertext**: the encrypted message (corresponds to a plaintext).
- **Key**: a “parameter” to a cryptosystem that is used in the encryption or decryption process, upon which its security depends.
- **Kerckhoff’s Principle**: a cryptosystem should be secure even if everything about it is public knowledge, except for the key.
- **Keyspace**: the set of all possible keys for a cryptosystem.
- **Cryptanalysis**: the art of deciphering coded messages without being told the key

CIPHERS



ANCIENT CIPHERS

- **Caesar Cipher**: Shifting letters over by some number (rot k):
  a -> c, b -> d, ..., y -> a, z -> b (rot 2)
- **Substitution Cipher**:
  Create a table mapping each letter to another
  To crack: use frequency analysis
- **Vigenere Cipher**:
  like Caesar Cipher, but each character is shifted by a keyphrase, rather than just one number

CAESAR CIPHER

Key: a “shift”, a number from 1 to 25 (inclusive).
Encryption Algorithm: to obtain the ciphertext, replace each letter in the plaintext with the corresponding letter shifted down the alphabet by {key} places.
Decryption: simply shift backwards through the alphabet instead.
Example: with key = 3, plaintext = “ATTACKATDAWN”, the ciphertext becomes “DWWDFNDWGDZQ”

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
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

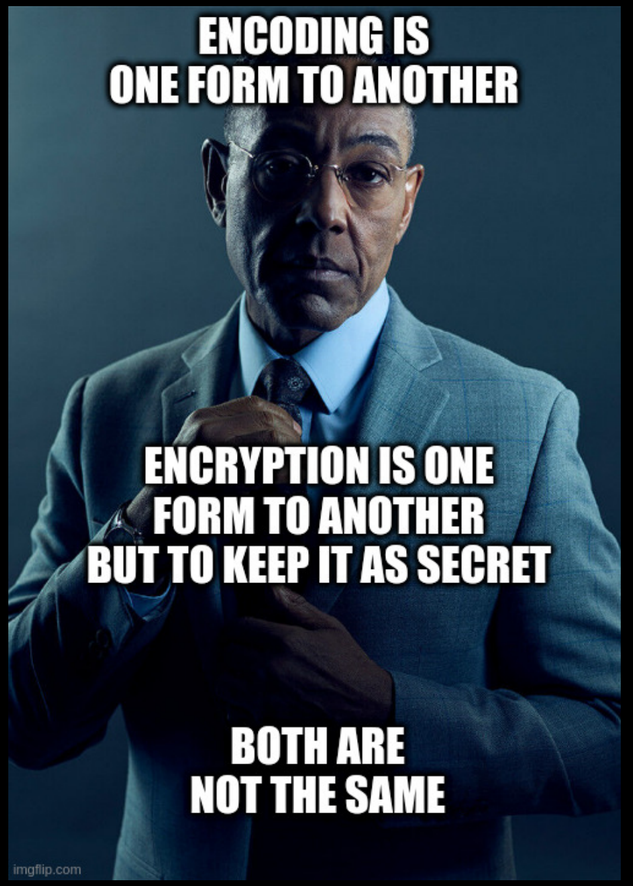
CAESAR CIPHER

- First we translate all of our characters to numbers,
  'a'=0, 'b'=1, 'c'=2, ... , 'z'=25.
- We can now represent the Caesar cipher as an encryption function, e(x), where x is the character we are encrypting, as:
  e(x) = (x+k) % 26
- Where k is the key (the shift) applied to each letter. After applying this function the result is a number which must then be translated back into a letter. The decryption function is :
  e(x) = (x-k) % 26

ENCODING VS ENCRYPTION

ENCODING is the process of converting data into a format universally accepted by various platforms

ENCRYPTION transforms data, in such a way that only specific individuals can reverse the transformation.



THANK YOU

CONTACT US

@nitdgplug

/nitdgplug

@lugnitdgp

@nitdgplug

GNU/Linux Users' Group NIT-Dgp

nitdlug@gmail.com

GNU/Linux Users' Group NIT-Dgp