Segurança e Criptografia com Python

OI!

Me chamo Elma Santos

Técnica em informática (**IFRN**). Graduanda em tecnologia da informação (**UFRN**)

CODE MINER

pyladies

MULHERES NA TI

@elmasantos

@elmasnts

Segurança da informação

“[...] preservar a **integridade**, a **disponibilidade** e a **confidencialidade** dos recursos do sistema de informação”

Confidentiality, Integrity and Availability (CIA triad)

- **Confidencialidade**: A proteção dos dados contra a revelação não autorizada.
- **Integridade**: A garantia de que um dado recebido está exatamente da mesma forma como foi enviado por uma entidade autorizada.
- **Disponibilidade**: Garantia de que certo recurso estará disponível para utilização.
- **Autenticidade, Controle de Acesso, Auditoria**

Cifragem - mecanismo de segurança

Criptografia

kryptós = oculto/escondido + gráphein= escrita

Termos importantes

- **Texto claro**: é a mensagem em sua forma original
- **Texto cifrado**: é a mensagem codificada
- **Encriptação**: processo de converter um texto claro em um texto cifrado
- **Decriptação**: restauração do texto claro a partir do texto cifrado
- **Cifra**: esquema utilizado para encriptação
- **Criptoanálise**: uso de técnicas empregadas para decifrar uma mensagem sem conhecimento dos detalhes de encriptação

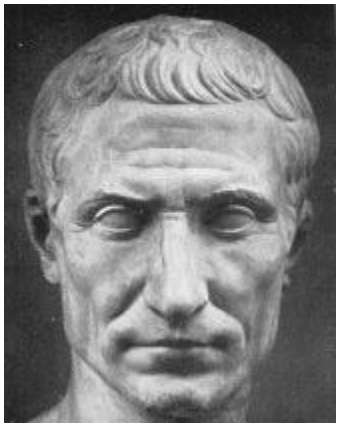
Recaptulando...

texto claro > encriptação > texto cifrado

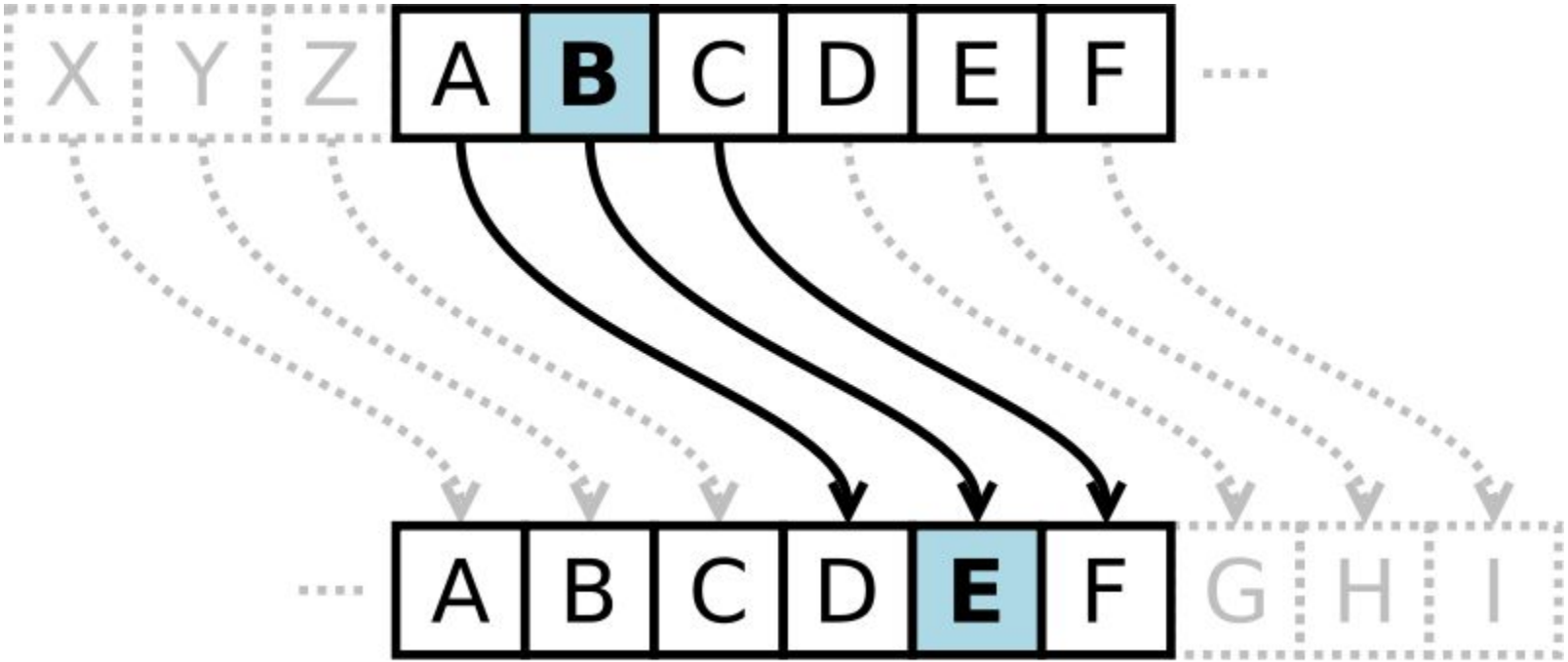
texto cifrado > decriptação > texto claro

Cifra de César

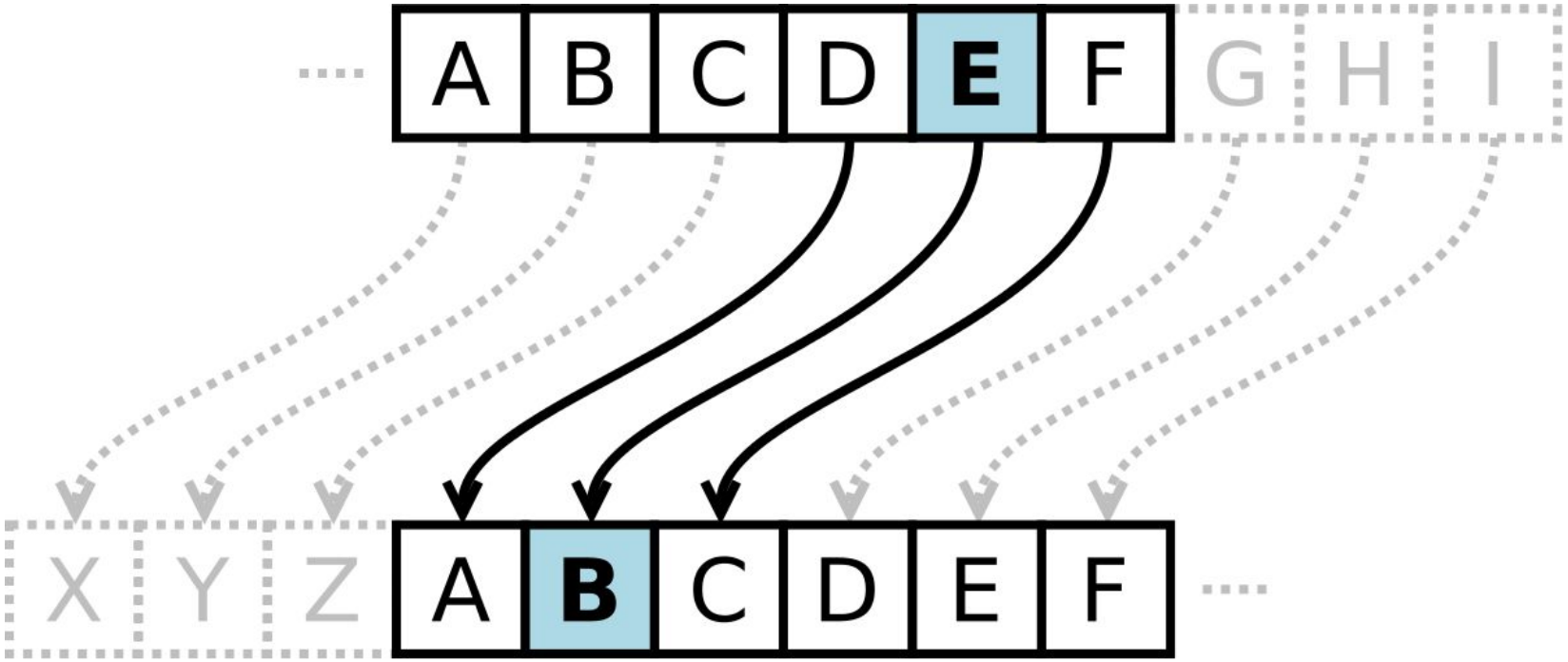
O imperador romano Júlio César utilizou uma cifra em suas correspondências pessoais em 50 a.c.



Cifra de César



Cifra de César



Até tu, Brutus? = Dwh wx, Euxwxv?

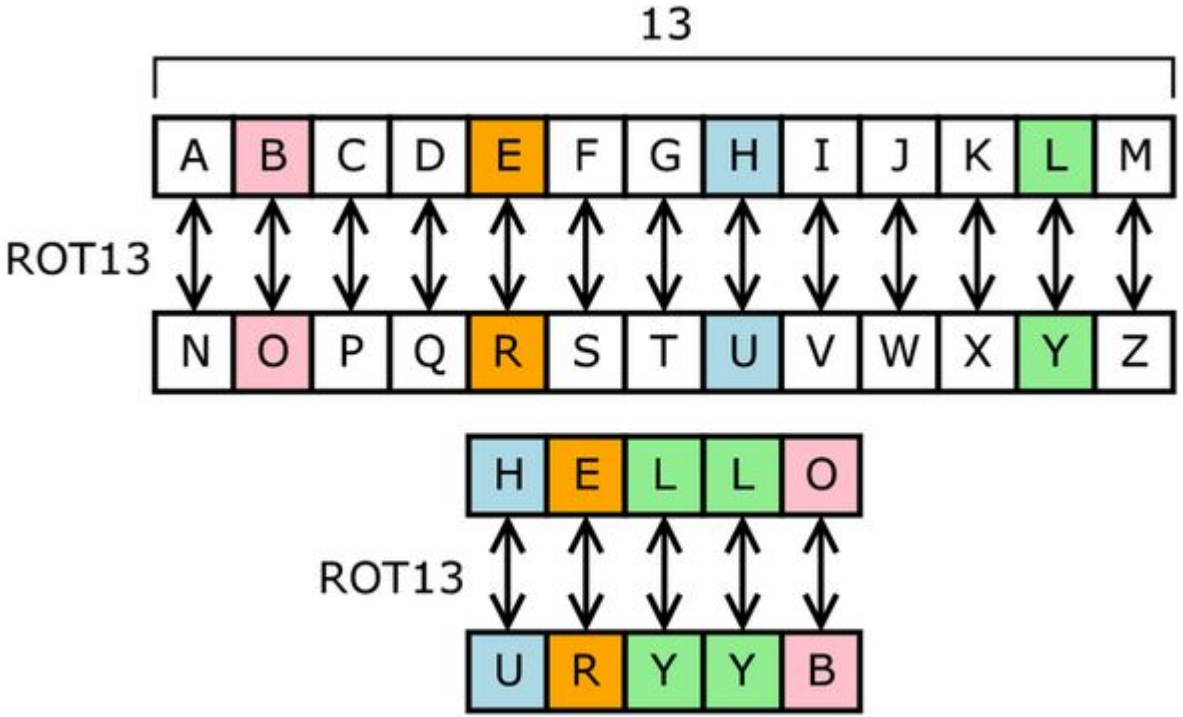
$C = E(k,p) = (p+k) \bmod 26,\ k=1..25$

$C = E(3,p) = (p+3) \bmod 26$

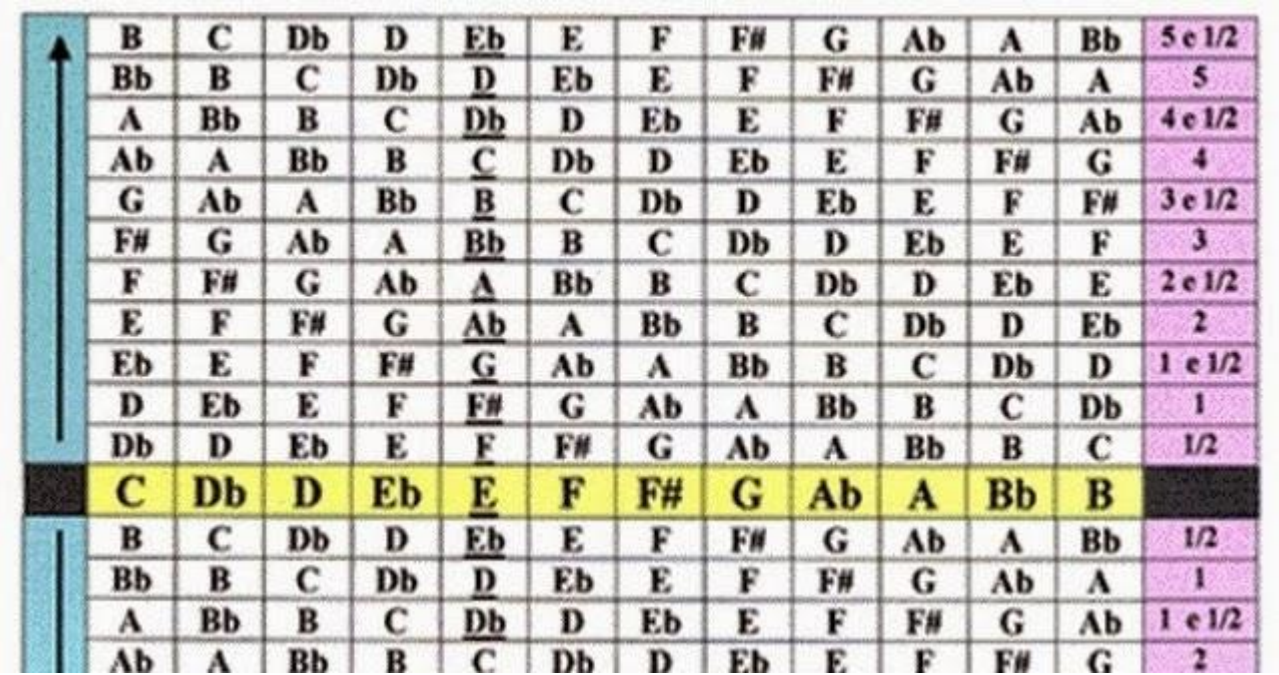
caesarCipher.py

```
# we need 2 helper mappings, from letters to ints and the inverse
L2I = dict(zip("ABCDEFGHIJKLMNOPQRSTUVWXYZ",range(26)))
I2L = dict(zip(range(26),"ABCDEFGHIJKLMNOPQRSTUVWXYZ"))

key = 3
plaintext = "DEFEND THE EAST WALL OF THE CASTLE"

# encipher
ciphertext = ""
for c in plaintext.upper():
    if c.isalpha(): ciphertext += I2L[ (L2I[c] + key)%26 ]
    else: ciphertext += c

# decipher
plaintext2 = ""
for c in ciphertext.upper():
    if c.isalpha(): plaintext2 += I2L[ (L2I[c] - key)%26 ]
    else: plaintext2 += c

print plaintext
print ciphertext
print plaintext2
```

Cifra de substituição: um símbolo do texto claro é substituído por outro



Cifra de transposição: é feita a permutação das posições dos símbolos.



B	C	Db	D	Eb	E	F	F#	G	Ab	A	Bb	5 e 1/2
Bb	B	C	Db	D	Eb	E	F	F#	G	Ab	A	5
A	Bb	B	C	Db	D	Eb	E	F	F#	G	Ab	4 e 1/2
Ab	A	Bb	B	C	Db	D	Eb	E	F	F#	G	4
G	Ab	A	Bb	B	C	Db	D	Eb	E	F	F#	3 e 1/2
F#	G	Ab	A	Bb	B	C	Db	D	Eb	E	F	3
F	F#	G	Ab	A	Bb	B	C	Db	D	Eb	E	2 e 1/2
E	F	F#	G	Ab	A	Bb	B	C	Db	D	Eb	2
Eb	E	F	F#	G	Ab	A	Bb	B	C	Db	D	1 e 1/2
D	Eb	E	F	F#	G	Ab	A	Bb	B	C	Db	1
Db	D	Eb	E	F	F#	G	Ab	A	Bb	B	C	1/2
C	**Db**	**D**	**Eb**	**E**	**F**	**F#**	**G**	**Ab**	**A**	**Bb**	**B**	
B	C	Db	D	Eb	E	F	F#	G	Ab	A	Bb	1/2
Bb	B	C	Db	D	Eb	E	F	F#	G	Ab	A	1
A	Bb	B	C	Db	D	Eb	E	F	F#	G	Ab	1 e 1/2
Ab	A	Bb	B	C	Db	D	Eb	E	F	F#	G	2

Cifra de transposição de colunas

VAMOS EMBORA, FOMOS DESCOBERTOS

Z E B R A S
V A M O S E
M B O R A F
O M O S D E
S C O B E R
T O S J E U

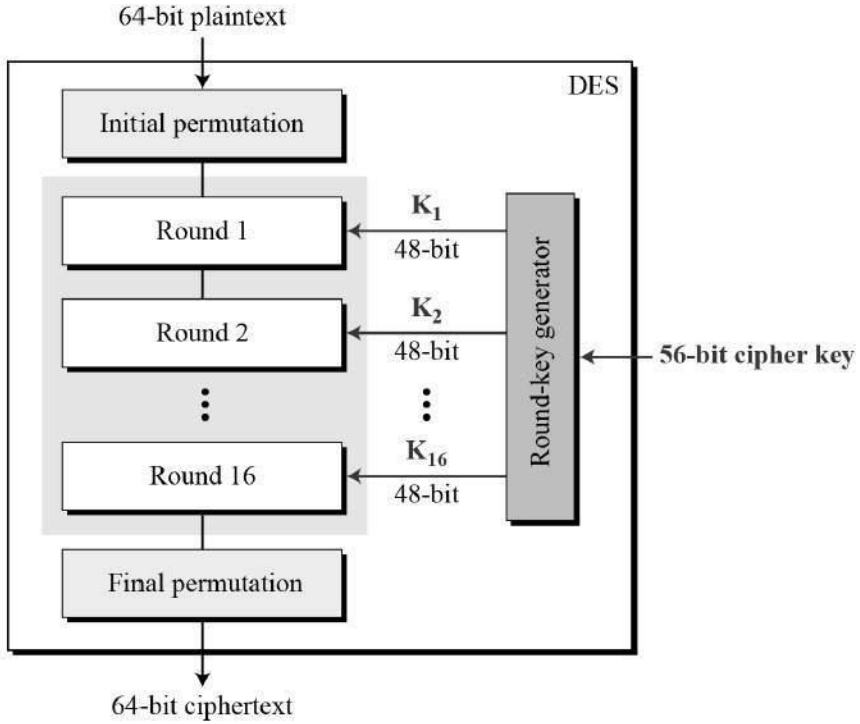
SADEE MOOOS ABMCO ORSBJ EFERU VMOST

Enigma

- Encriptar e decriptar códigos de guerra
- Conjunto de cilindros rotativos, cada um com 26 pinos de entrada e 26 pinos de saída.
- Era utilizada uma complexa cifra de substituição.

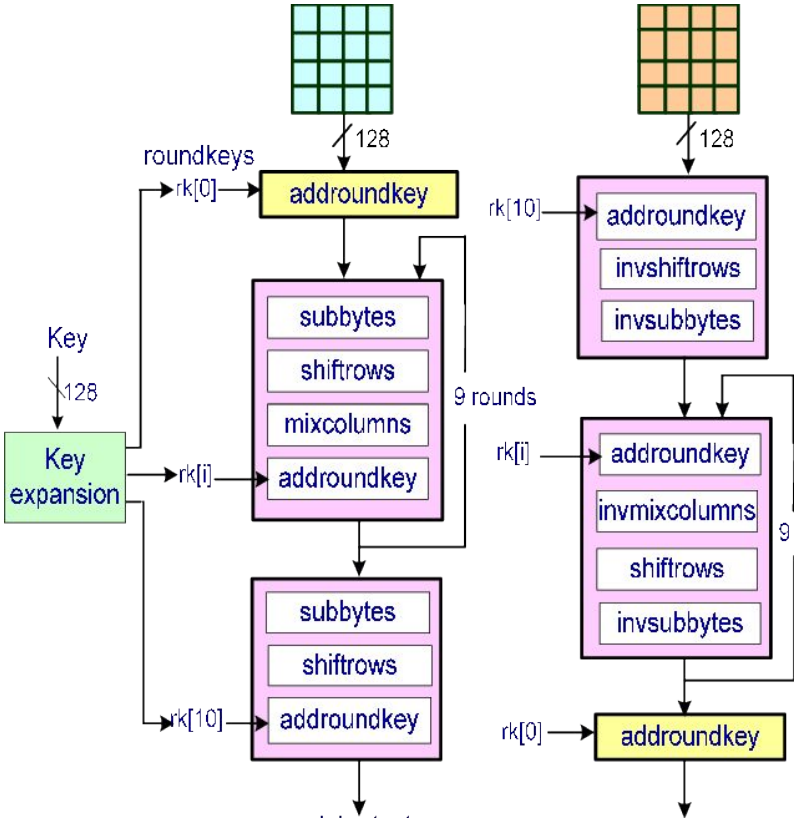
Avanço da criptografia

- Com o avanço da computação, a criptografia avançou do hardware para o software
- Data Encryption Standard (**DES**). Desenvolvida pela IBM e NSA em 1977.
- **72 quadrilhões** de chaves
- Com o tempo tornou-se um algoritmo inseguro



Advanced Encryption Standard (AES)

- 2001
- Criptoanálise por força bruta levaria **trilhões** de anos
- Corta os dados em blocos de 16 bytes e aplica uma série de substituições e permutações baseadas no valor da chave



PyCrypto

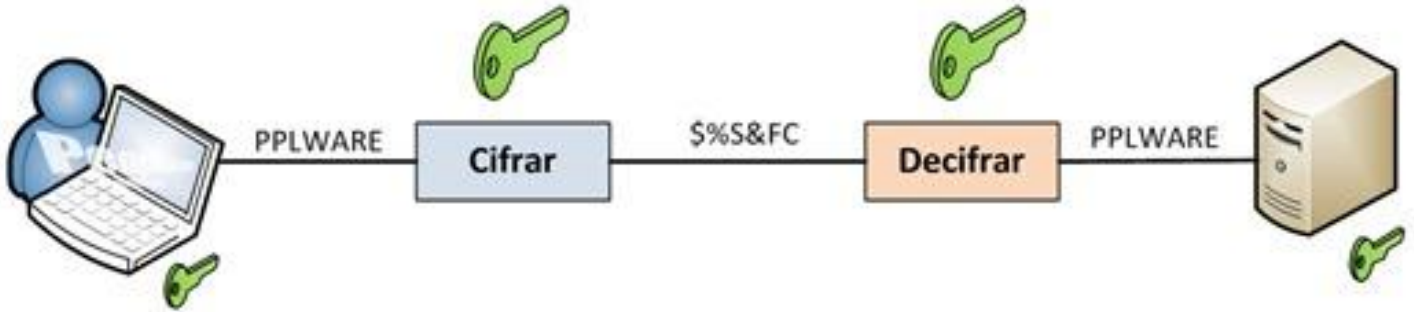
- Biblioteca Python que fornece algoritmos de encriptação como AES, DES, RSA
- Instalação: pip install pycrypto

PyCrypto

```
from Crypto.Cipher import AES
# Encryption
encryption_suite = AES.new('This is a key123', AES.MODE_CBC, 'This is an
IV456')
cipher_text = encryption_suite.encrypt("A really secret message. Not for
prying eyes.")

# Decryption
decryption_suite = AES.new('This is a key123', AES.MODE_CBC, 'This is an
IV456')
plain_text = decryption_suite.decrypt(cipher_text)
```

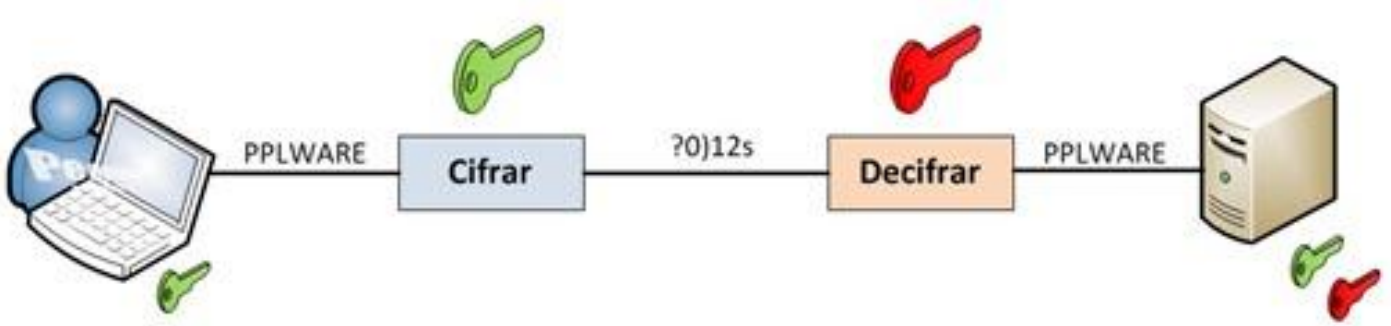
Cifragem simétrica

- A cifra de césar, o DES e o AES são cifras simétricas.



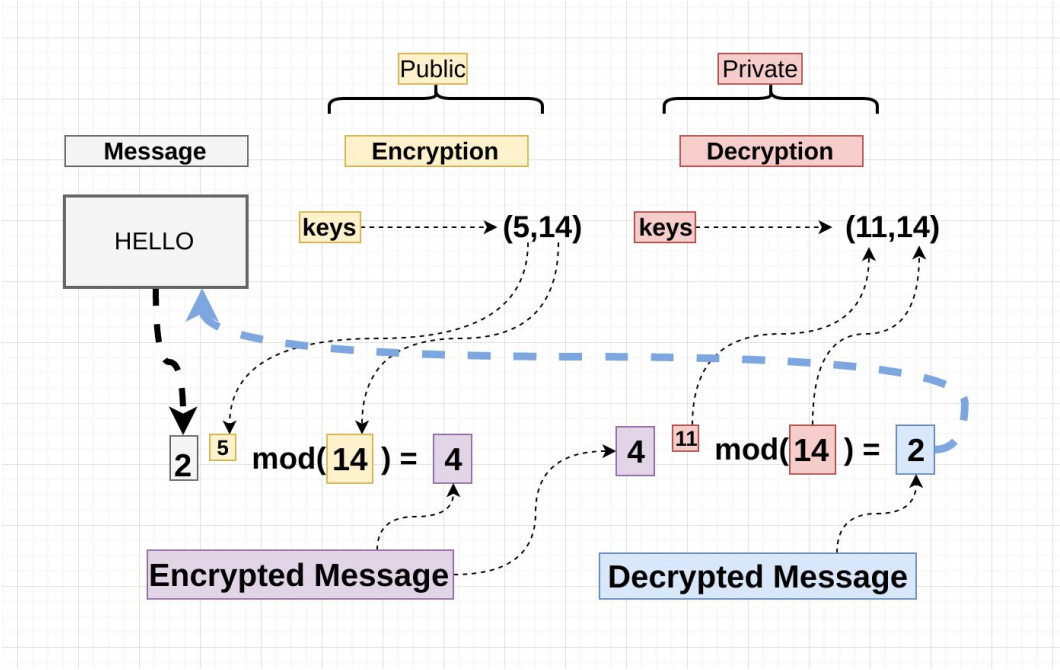
Cifragem assimétrica

- Usa um par de chaves distintas (chave privada e chave pública)



RSA

- Algoritmo de encriptação assimétrica
- Rivest, Shamir e Adleman



Cryptography

- Biblioteca mais popular
- Instalação:

```
pip install cryptography
```

Hacking Secret Ciphers with Python - https://inventwithpython.com/hacking/
Cryptography with Python Tutorial - https://www.tutorialspoint.com/cryptography_with_python/index.htm
Bibliotecas - https://docs.python-guide.org/scenarios/crypto/

Obrigada! :)

telegram: @elmasnts