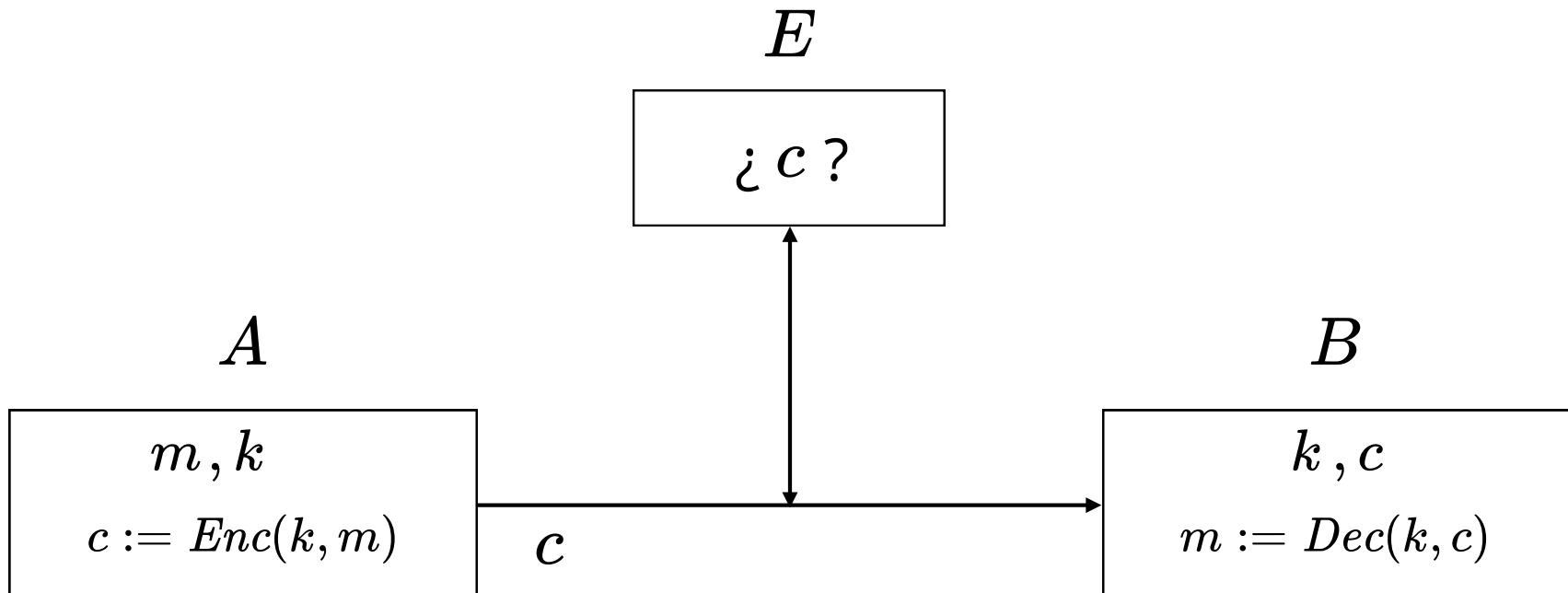


# IIC3253

Criptografía simétrica (de llave privada)

# Cifrado



# Cifrado del César

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

HOLA MUNDO  
EMIX JRKAM

MANDEN BITCOINS A UCRANIA  
JXKABK YFQZMFKP X RZOXKFX

¿Problemas?

# Cifrado del César + llave

Llave = shift **7**

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

HOLA MUNDO  
AIET FÑGWI

MANDEN BITCOINS A UCRANIA  
FTGWXG UBNVIBGM T ÑVLTGBT

¿Problemas?

La probabilidad de que un atacante "seleccione" o "adivine" la llave correcta debe ser muy baja.

⇒ El espacio de llaves posibles debe ser muy (muy) grande

¿Cómo podríamos agrandar el espacio de llaves siguiendo la idea de "sustituir"?

# Shift → Permutación

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

HOLA MUNDO  
RHKP AZJWH

¿Cuántas llaves posibles?

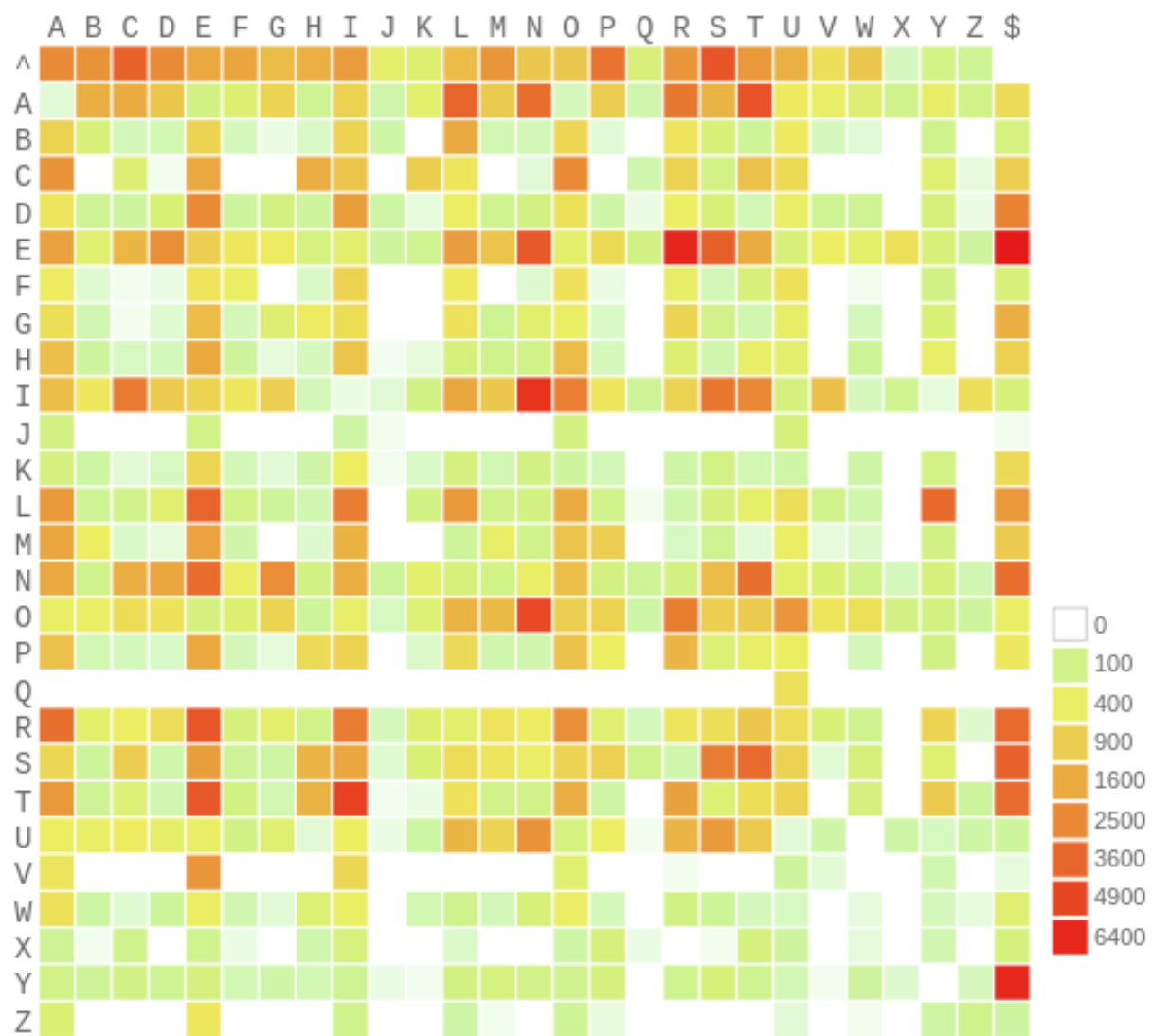
$27! = 10,888,869,450,418,352,160,768,000,000$

# Es este un buen cifrado?

## | The Most Used Letters in English

WordCheats.com







Un espacio de  
llaves grande  
es necesario,  
no suficiente.

# ONE-TIME PAD (OTP)

# Operación Módulo

(Recordatorio)

Dados  $a, n \in \mathbb{Z}$ , existe un único par de elementos  $(q, r) \in \mathbb{Z}^2$  tal que:

$$0 \leq r < |n|$$

$$a = q \cdot n + r$$

  
Cuociente Resto

Decimos entonces que  $a \bmod n = r$  y que  $a \equiv r \bmod n$

# Operación Módulo

(Ejemplos)

$$10 \bmod 3 = 1$$

$$28 \bmod 8 = 4$$

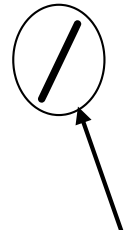
$$6 \bmod -20 = 6$$

$$-6 \bmod -20 = 14$$

Siempre esperaríamos que

$$n \cdot \left\lfloor \frac{a}{n} \right\rfloor + (a \bmod n) = a$$

Programando, esto se ve como

$$n * (a / n) + a \% n = a$$


División entera

```
1 # Python 2
2 print("La división entera entre 6 y -20 es:")
3 print(6 / -20)
```

Output: -1

```
1 // C++
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     cout << "La división entera entre 6 y -20 es: " << endl;
7     cout << (6 / 20) << endl;
8     return 0;
9 }
```

Output: 0



Esperamos que

$$n * (a / n) + a \% n = a$$

$$-20 * (6 / -20) + 6 \% -20 = 6$$

Python:  $-20 * -1 + 6 \% -20 = 6 \quad \Rightarrow 6 \% -20 = -14$

C++:  $-20 * 0 + 6 \% -20 = 6 \quad \Rightarrow 6 \% -20 = 6$



# Operación Módulo

Dados  $a, n \in \mathbb{Z}$ , existe un único par de elementos  $(q, r) \in \mathbb{Z}^2$  tal que:

$$0 \leq r < |n|$$

$$a = q \cdot n + r$$

Decimos entonces que  $a \bmod n = r$  y que  $a \equiv r \bmod n$





$$\text{Enc}(\text{SECRETKEY}, \text{HOLAMUNDO}) = \text{ZSNRPÑWHN}$$

¿Cómo decriptar?

$$\text{Dec}(\text{SECRETKEY}, \text{ZSNRPÑWHN}) = \text{HOLAMUNDO}$$

ZSNRPÑWHN	➡	—	26	19	13	18	16	14	23	7	13	
SECRETKEY		19	4	2	18	4	20	10	4	25		
		mod 27	7	15	11	0	12	-6	13	3	-12	
HOLAMUNDO	➡		7	15	11	0	12	21	13	3	15	

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	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



# Decantando OTP...

Dado un string  $a \in \{A, B, C, \dots, Z\}^*$ , llamaremos  $\bar{a}$  al string en  $\{0, 1, 2, \dots, 26\}^*$  correspondiente a  $a$ .

Dado un string  $s \in \{0, 1, 2, \dots, 26\}^*$ , llamaremos  $\bar{s}$  al string en  $s \in \{A, B, C, \dots, Z\}^*$  correspondiente a  $s$ .

Extendemos las operaciones de los enteros a strings de enteros, aplicando las operaciones por coordenada.

$$\begin{array}{r}
 \begin{array}{cccccccccc}
 & 26 & 19 & 13 & 18 & 16 & 14 & 23 & 7 & 13 \\
 - & 19 & 4 & 2 & 18 & 4 & 20 & 10 & 4 & 25 \\
 \hline
 \text{mod } 27 & 7 & 15 & 11 & 0 & 12 & -6 & 13 & 3 & -12 \\
 \hline
 & 7 & 15 & 11 & 0 & 12 & 21 & 13 & 3 & 15
 \end{array}
 \end{array}$$

Dados  $k, m \in \{A, B, C, \dots, Z\}^\ell$ , OTP se define por:

$$Enc(k, m) = \overline{(\bar{m} + \bar{k}) \bmod 27} =: c$$

$$Dec(k, c) = \overline{(\bar{c} - \bar{k}) \bmod 27} =? m$$

$$Dec(k, Enc(k, m)) = \overline{\overline{\overline{(\bar{m} + \bar{k}) \bmod 27} - \bar{k}} \bmod 27}$$

$$= \overline{((\bar{m} + \bar{k}) \bmod 27 - \bar{k}) \bmod 27}$$

$$= \overline{(\bar{m} + \bar{k} - \bar{k}) \bmod 27} = \overline{\bar{m} \bmod 27}$$

$$= \overline{\bar{m}} = m$$

Generalmente abusaremos de la notación y supondremos que nuestros mensajes son directamente arreglos de números

Dados  $k, m \in \{0, \dots, N - 1\}^\ell$ , OTP se define por:

$$Enc(k, m) = (m + k) \bmod N =: c$$

$$Dec(k, c) = (c - k) \bmod N = m$$

# ¿Qué tan bueno es OTP?

¿Qué pasa si veo un mensaje cifrado  $c$  pasar?

Aquí un ejemplo:

```
c = "YFTGXEIWIWEHAGQGESLPKRVLMYGXSJJIQZVIYHVBRJGNTR"  
m = "ESTEMENSAJEESLITERALMENTEIMPOSIBLEDEDESCRIPTAR"  
k = "UNACLAVEINADIVINABLEYNISIQUIERAPORFUERZABRUTA"
```