Criptografia

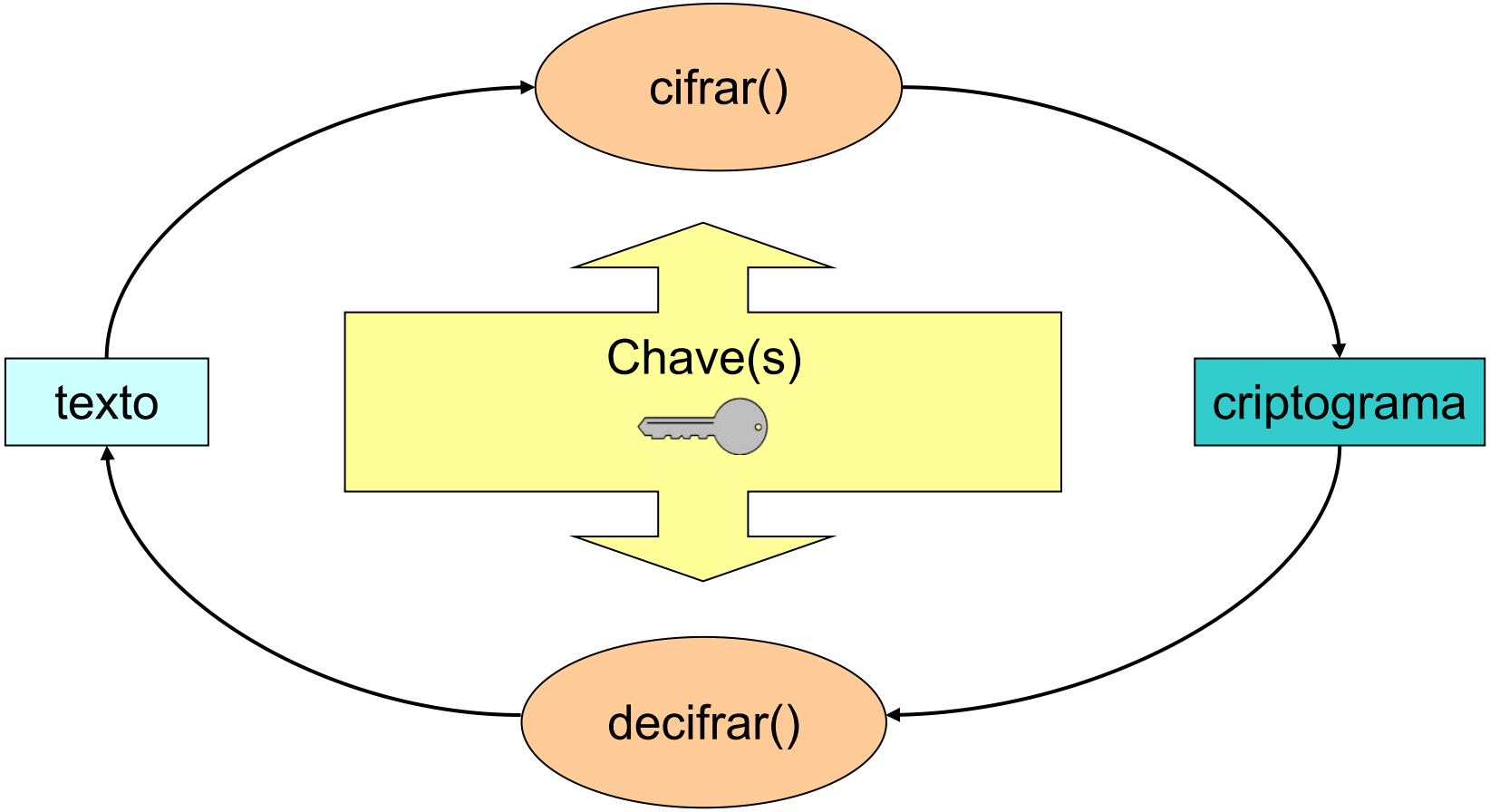
Terminologia

- **Criptografia**
  - Arte ou ciência de escrever de forma escondida/confidencial
    - do Gr. kryptós, oculto + graph, r. de graphein, escrever
  - Inicialmente para garantir a privacidade da informação
  - Esteganografia
    - do Gr. steganós, oculto + graph, r. de graphein, escrever
- **Criptanálise**
  - Arte ou ciência de quebrar sistemas criptográficos ou informação criptografada
- **Criptologia**
  - Criptografia + criptanálise

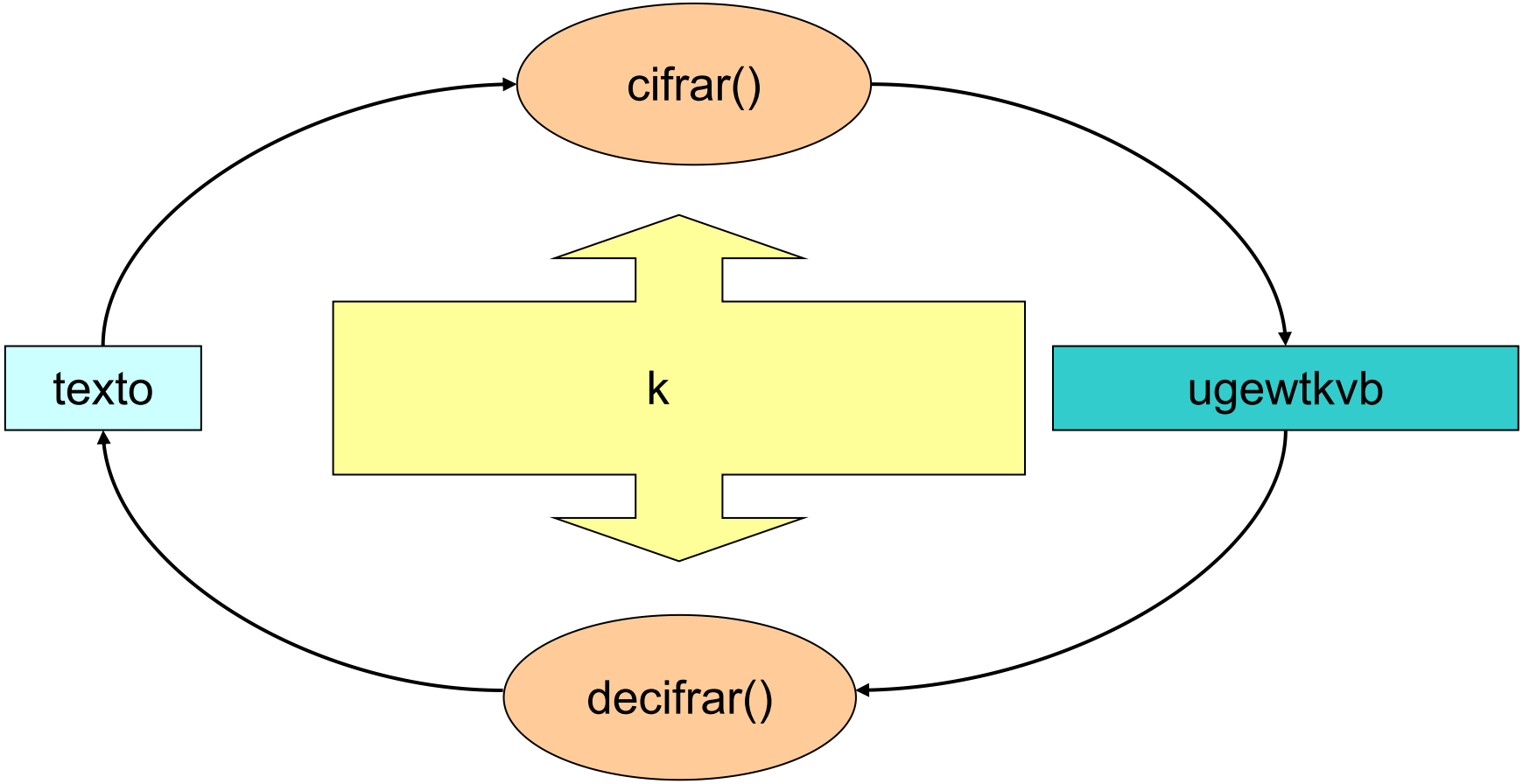
Terminologia

- **Cifra**
  - Técnica concreta de criptografia
- **Operação de uma cifra**
  - **Cifra**: texto em claro -> criptograma
  - **Decifra**: criptograma -> texto em claro
- **Algoritmo:** modo de transformação de dados
- **Chave:** parâmetro do algoritmo
  - Influencia a operação do algoritmo

Operações



Operações



Casos de uso (Cifras Simétricas)

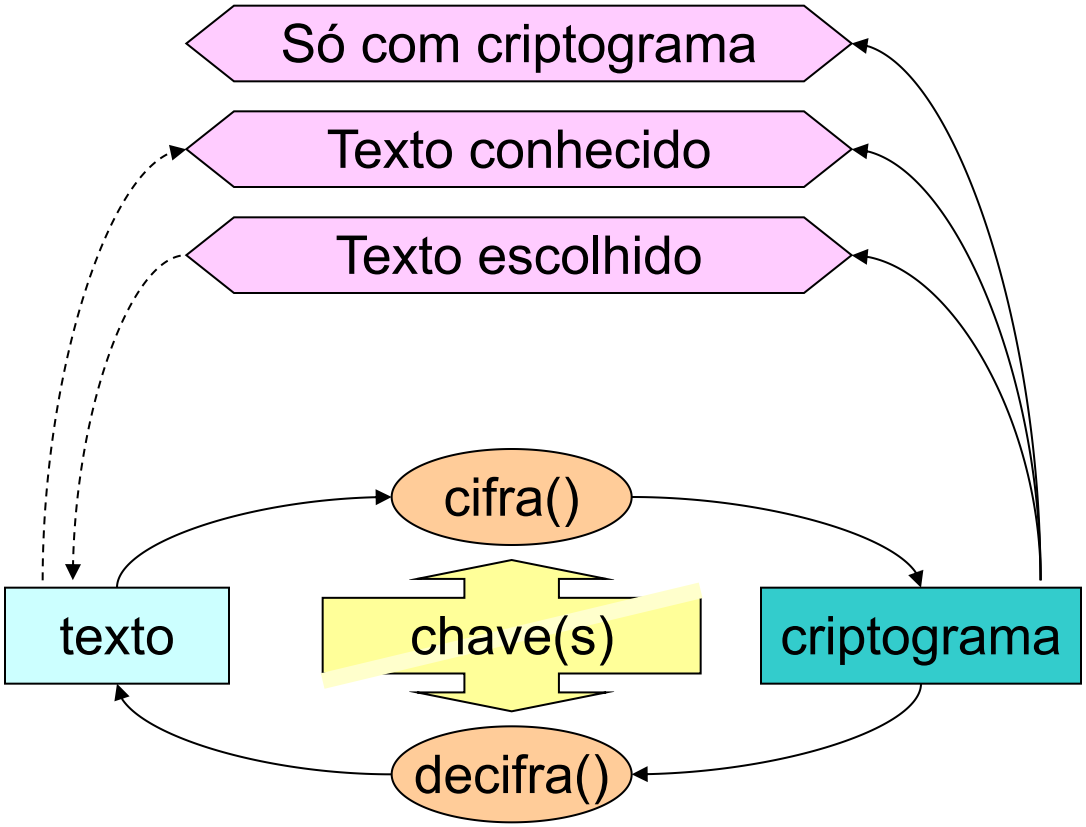
- **Proteção própria com chave K**
  - Alice cifra texto P com chave K
    -> `Alice: C = {P}`$_k$
  - Alice decifra C com chave K
    -> `Alice: P'= {C}`$_k$
  - P' deverá ser igual a P (deve ser verificado)

- **Comunicações seguras com chave K**
  - Alice cifra texto P com chave K
    -> `Alice: C = {P}`$_k$
  - Bob decifra C com chave K
    -> `Bob: P' = {C}`$_k$
  - P' deve ser igual a P (deve ser verificado)

Criptanálise: Objetivos

- **Obtenção do texto original**
  - Relativo a um criptograma
- **Obtenção de uma chave de cifra**
  - Ou de uma equivalente
- **Obtenção do algoritmo de cifra**
  - Ou de um equivalente
  - Normalmente os algoritmos não são secretos, mas existem exceções:
    - Lorenz, A5 (GSM), RC4, Crypto-1 (Mifare)
    - Algoritmos para DRM (Digital Rights Management)
  - Por engenharia reversa

Ataques por Criptanálise



Ataques por Criptanálise

- **Força Bruta (ataque genérico)**
  - Pesquisa exaustiva sobre todo o espaço de chaves, até se encontrar uma chave adequada
  - Não é prática para espaços de dimensão grande
    - ex. chaves de 128 bits possuem um espaço de 2^{128} bits.
  - É importante que exista aleatoriedade na chave.

- **Ataques mais inteligentes**
  - Reduzir o espaço de pesquisa para uma dimensão menor:: palavras, números, conjunto reduzido, alfabeto
  - Identificar padrões em algumas operações, etc..

Evolução das Cifras

- **Manuais:** Algoritmos de substituição ou transposição

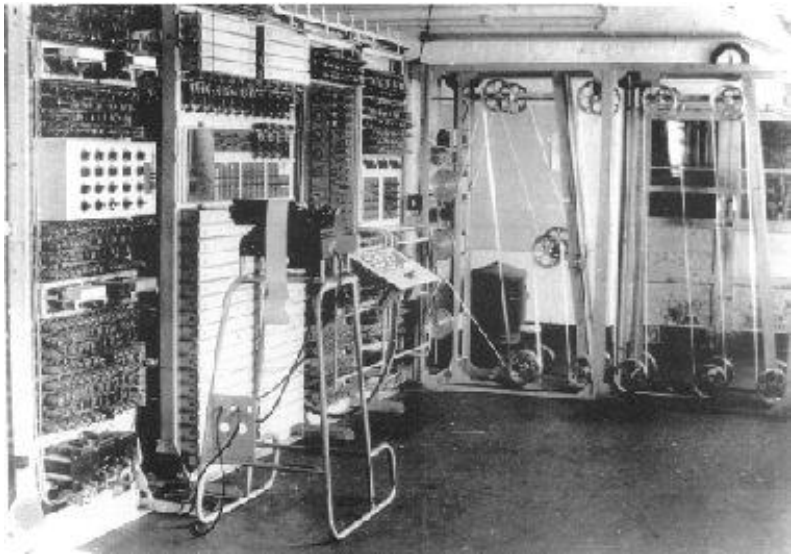
Fonte: Wikimedia Commons e CryptoMuseum

Evolução das Cifras

- **Mecânicas**
  - A partir do Séc. XIX
    - Máquina Enigma
    - M-209 Converter
  - Algoritmos de substituição ou transposição
    - Elementos críticos para a 2ª Grande Guerra

Evolução das Cifras

- **Cifras Informáticas**
  - Surgem com o uso dos computadores
  - Algoritmos de substituição mais complexos
  - Algoritmos matemáticos de grandes números ou problemas complexos
  - Utilizados de forma comum (e transparente) no dia a dia



Cifras: Tipos Básicos

- **Transposição:** O texto original é “baralhado”

O	O	I	B	H
T	O	N	A	A
E	R	A	R	D
X	I	L	A	O
T	G	E	L	

- **Resultado:** ooibh tonaa erard xilao tgel

Cifras: Tipos Básicos

- **Transposição:** Permutações intra-blocos

P	E	R	M	U
T	A	C	O	E
S	I	N	T	R
A	B	L	O	C
O	S			

- **Resultado:**
  - (13524) -> pruem tceao snrit alcbo os
  - (25413) -> eumpr aeotc irtsn bcoal so

Cifras: Tipos Básicos

- **Substituição**
  - Cada símbolo original é substituído por outros
  - Considera símbolos como letras, dígitos e pontuação
  - Na realidade são blocos de bits

- **Estratégias de substituição**
  - Mono alfabética (um para um)
  - Poli-alfabética (muitos para um)
  - Homofónica (um para muitos)

Cifras: Mono-alfabéticas

- **Usam apenas um alfabeto de substituição**
  - Com um número de elementos #A

- **Exemplos**
  - Aditivas (ou de translação)
    - `cripto - letra = (letra + chave) mod #A`
    - `letra = (cripto - letra – chave) mod #A`
    - `Número de chaves efetivas = #A`
    - Cifra de César (ROT-x)
  - Com frase-chave
    - `ABCDEFGHIJKLMNOPQRSTUVWXYZ`
    - `QTUWXYZCOMFRASEHVBDGIJKLNP`
    - Número de chaves efetivas = #alfabeto! -> 26! ≈ 288

- **Problemas**
  - Reproduzem padrões do texto original
  - Letras, digramas, trigramas, etc.
  - A análise estatística facilita a criptanálise
  - “The Gold Bug”, Edgar Alan Poe

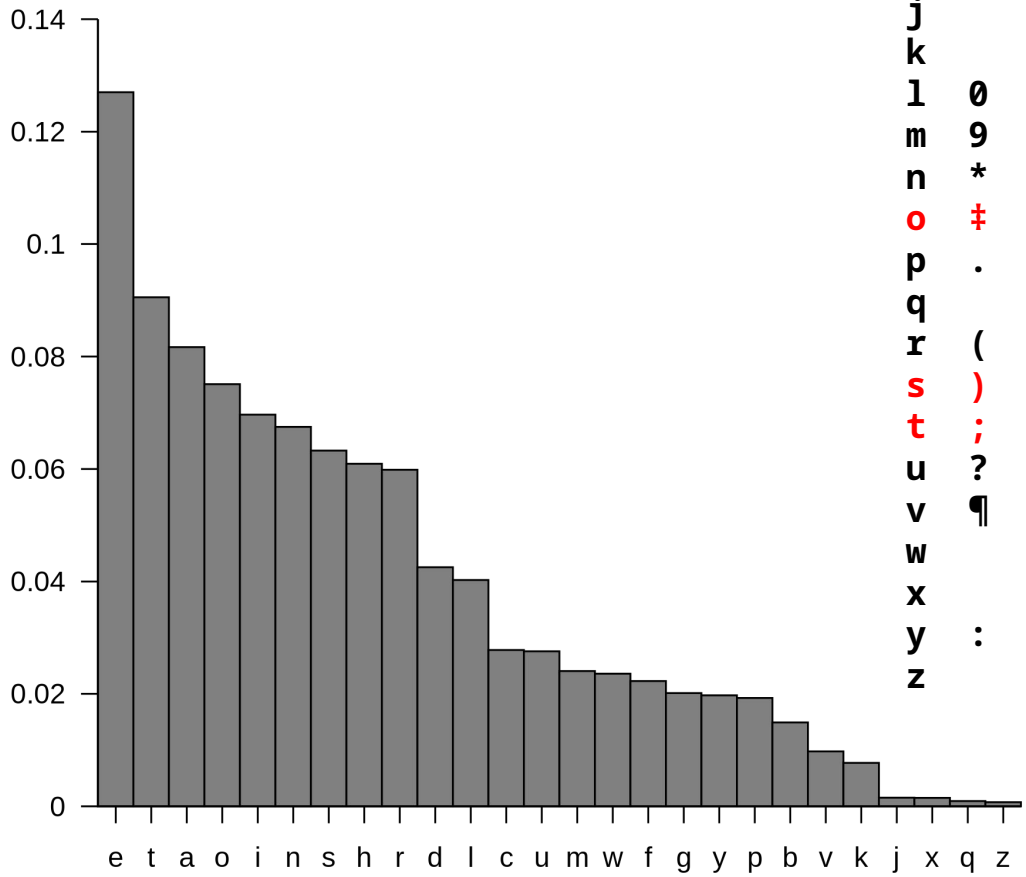
Cifras: Mono-alfabéticas

```
a good glass in the
bishop's hostel in the
devil's seat fifty-one
degrees and thirteen
minutes northeast and
by north main branch
seventh limb east side
shoot from the left eye
of the death's-head a
bee line from the tree
through the shot forty
feet out
```

```
53‡‡†305))6*;4826)4‡.)
4‡);806*;48†860))85;1‡
(;:‡*8†83(88)5*†;46(;8
8*96*?;8)*‡(;485);5*†2
:*‡(;4956*2(5*—4)88*;4
069285);)6†8)4‡‡;1(‡9;
48081;8:8‡1;48†85;4)48
5†528806*81(‡9;48;(88;
4(‡?34;48)4‡;161;:188;
‡?;
```

Cifras: Mono-alfabéticas

```
53‡‡†305))6*;4826)4‡.)4‡);80
agoodglassinthebishopshostel

6*;48†8¶60))85;1‡(;:‡*8†83(88)
inthedevilsseatfortyonedegrees

5*†;46(;88*96*?;8)*‡(;485);5*†
andthirteenminutesnortheastand

2:*‡(;4956*2(5*-4)8¶8*;40692
bynorthmainbranchseventhlimb

85);)6†8)4‡‡;1(‡9;48081;8:8‡1
eastsideshootfromthelefteyeof

;48†85;4)485†528806*81(‡9;48
thedeathsheadabeelinefromthe

;(88;4(‡?34;48)4‡;161;:188;‡?;
treethroughtheshotfiftyfeetout
```



a	5	(12)
b	2	(5)
c	-	(1)
d	†	(8)
e	8	(33)
f	1	(8)
g	3	(4)
h	4	(19)
i	6	(11)
j		
k		
l	0	(6)
m	9	(5)
n	*	(13)
o	‡	(16)
p	.	(1)
q		
r	(	(10)
s	)	(16)
t	;	(26)
u	?	(3)
v	¶	(2)
w		
x		
y	:	(4)
z		

Cifras: Mono-alfabéticas

- **Frequência de Pares**
  - AO, NO, AS, OS, SO, UM, IA, NA...
- **Frequência de Triplos**
  - QUE, NAO, EST, ENT, ÇÃO, TRA...
- **Probabilidades condicionais**
  - P(A | B) diferente de P(Z | B)

Cifras: Poli-alfabéticas

- **Usam N alfabetos de substituição**
  - Têm período **N**
- **Exemplo:** Cifra de Vigenère
- **Problemas**
  - Conhecido o período, podem ser analisadas como N mono alfabéticas
    - O período pode ser descoberto usando estatística
  - Método de Kasiski
    - Fatorização de distâncias entre blocos iguais do criptograma
  - Índice de coincidência
    - Fatorização de deslocamentos relativos que produzem mais coincidências na sobreposição do criptograma

Cifra de Vigenère

	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

- **Exemplo de se cifrar a letra M com a chave S, resultando no criptograma E**
- **Criada por Blaise Vigenère (final séc XVI)**
  - le chiffre indéchiffrable!
- **Quebrada no séc XIX por Charles Babbage e Friedrich Kasiski**

Cifra de Vigenère

- **Texto:**

  Eles não sabem que o sonho é uma constante da vida
  tão concreta e definida como outra coisa qualquer,
  como esta pedra cinzenta em que me sento e descanso,
  como este ribeiro manso, em serenos sobressaltos
  como estes pinheiros altos

- **Cifra com o quadrado de Vigenère e chave "poema"**

texto	`elesnaosabemqueosonhoeumaconstantedavidataoconcretaedefinida`
chave	`poemapoemapoemapoemapoemapoemapoemapoemapoemapoemapoema`
criptograma	`tzienpcwmbtaugedgszhdsyyarcretpbxqdpjmpaiosoocqvqtpshqfxbmpa`

Criptanálise de um criptograma Vigenère

Teste de Kasiski

- **Localizar padrões comuns no criptograma**
- **Calcular afastamento entre padrões**
- **O maior divisor comum sugere a dimensão da chave (gcd)**

tzienpcwmbtaugedgszhdsyyarcretpbxqdpjmpaiosoocqvqtpshqfxbmpa

- **Com o texto indicado:**

mpa	$20 = 2 \times 2 \times 5$
tp	$20 = 2 \times 2 \times 5$

- **Com o poema completo:**

$$175 = 5 \times 5 \times 7$$
$$105 = 3 \times 5 \times 7$$
$$35 = 5 \times 7$$
$$20 = 2 \times 2 \times 5$$

Criptanálise de um criptograma Vigenère

- **Índice de coincidência (c/ poema completo)**
  - Sobreposição de uma cópia, com afastamento
  - Contagem dos carateres que se repetem

D	I	P (%)
1	6	3.2
2	6	3.2
3	5	2.7
4	7	3.8
5	15	8.2
6	3	1.6
7	6	3.3
8	5	2.8
9	10	5.6
10	6	3.4
11	8	4.5
12	6	3.4
13	6	3.4
14	7	4.0
15	11	6.3
16	10	5.8
17	6	3.5
18	2	1.2
19	8	4.7
20	23	13.6
21	4	2.4
22	3	1.8
23	7	4.2
24	9	5.5
25	12	7.3
26	6	3.7
27	6	3.7
28	6	3.7
29	7	4.4
30	9	5.7

D	I	P (%)
31	9	5.7
32	7	4.5
33	6	3.8
34	5	3.2
35	17	11.0
36	5	3.3
37	4	2.6
38	4	2.6
39	7	4.7
40	14	9.4
41	5	3.4
42	6	4.1
43	5	3.4
44	6	4.1
45	5	3.5
46	3	2.1
47	7	4.9
48	2	1.4
49	10	7.1
50	10	7.2
51	10	7.2
52	4	2.9
53	3	2.2
54	6	4.4
55	16	11.9
56	3	2.3
57	2	1.5
58	2	1.5
59	5	3.8
60	7	5.4

D	I	P (%)
61	1	0.8
62	5	3.9
63	6	4.8
64	6	4.8
65	11	8.9
66	7	5.7
67	6	4.9
68	6	5.0
69	5	4.2
70	14	11.8
71	5	4.2
72	6	5.1
73	7	6.0
74	7	6.1
75	4	3.5
76	3	2.7
77	1	0.9
78	9	8.1
79	8	7.3
80	7	6.4
81	5	4.6
82	6	5.6
83	3	2.8
84	2	1.9
85	8	7.7
86	6	5.8
87	4	3.9
88	2	2.0
89	5	5.0
90	9	9.1

D	I	P (%)
91	4	4.1
92	0	0.0
93	3	3.1
94	2	2.1
95	3	3.2
96	2	2.2
97	2	2.2
98	2	2.2
99	4	4.4
100	2	2.2
101	0	0.0
102	6	6.9
103	2	2.3
104	6	7.1
105	10	11.9
106	4	4.8
107	3	3.7
108	3	3.7
109	2	2.5
110	9	11.4
111	2	2.6
112	4	5.2
113	3	3.9
114	5	6.7
115	8	10.8
116	4	5.5
117	3	4.2
118	2	2.8
119	3	4.3
120	3	4.3

D	I	P (%)
121	4	5.9
122	3	4.5
123	0	0.0
124	3	4.6
125	7	10.9
126	1	1.6
127	1	1.6
128	2	3.3
129	2	3.3
130	6	10.2
131	1	1.7
132	4	7.0
133	2	3.6
134	1	1.8
135	4	7.4
136	3	5.7
137	0	0.0
138	2	3.9
139	4	8.0
140	2	4.1
141	3	6.2
142	1	2.1
143	3	6.5
144	4	8.9
145	7	15.9
146	2	4.7
147	1	2.4
148	0	0.0
149	0	0.0
150	1	2.6

D	I	P (%)
151	1	2.6
152	2	5.4
153	0	0.0
154	0	0.0
155	5	14.7
156	0	0.0
157	1	3.1
158	0	0.0
159	1	3.3
160	3	10.3
161	0	0.0
162	0	0.0
163	0	0.0
164	1	4.0
165	0	0.0
166	1	4.3
167	2	9.1
168	0	0.0
169	1	5.0
170	2	10.5
171	0	0.0
172	0	0.0
173	0	0.0
174	0	0.0
175	3	21.4
176	0	0.0
177	1	8.3
178	0	0.0
179	0	0.0
180	2	22.2

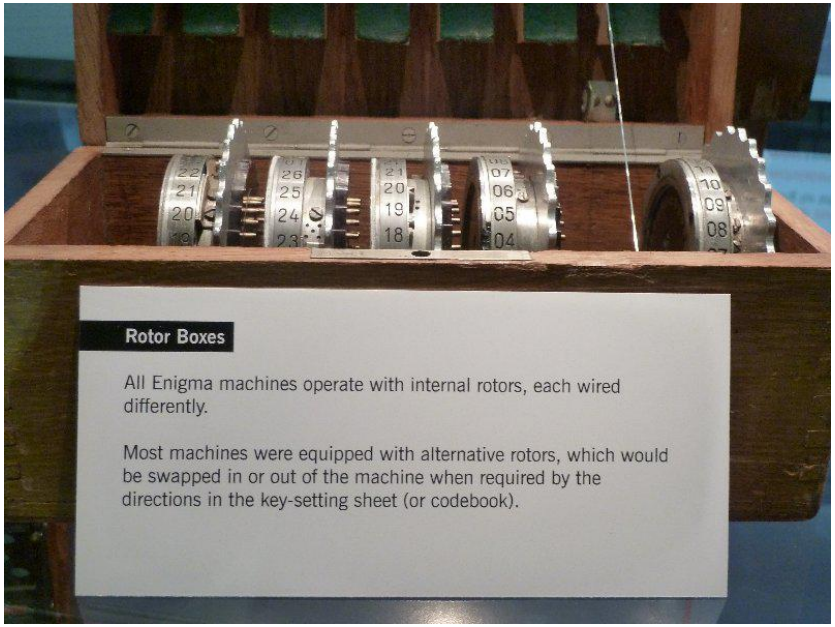
Máquinas de Rotores

Máquinas de Rotores

- **As máquinas de rotores concretizam cifras poli-alfabéticas complexas**
  - Cada rotor efetua uma permutação do alfabeto
    - Que consiste num conjunto de substituições
  - A posição do rotor concretiza um alfabeto de substituição
  - A rotação de um rotor concretiza uma cifra poli-alfabética
  - Acumulando vários rotores em sequência e rodando-os de forma diferenciada consegue-se uma cifra poli-alfabética complexa

- **A chave de cifra é:**
  - O conjunto de rotores usado
  - A ordem relativa dos rotores
  - A posição de avanço do rotor seguinte
  - A posição original dos rotores

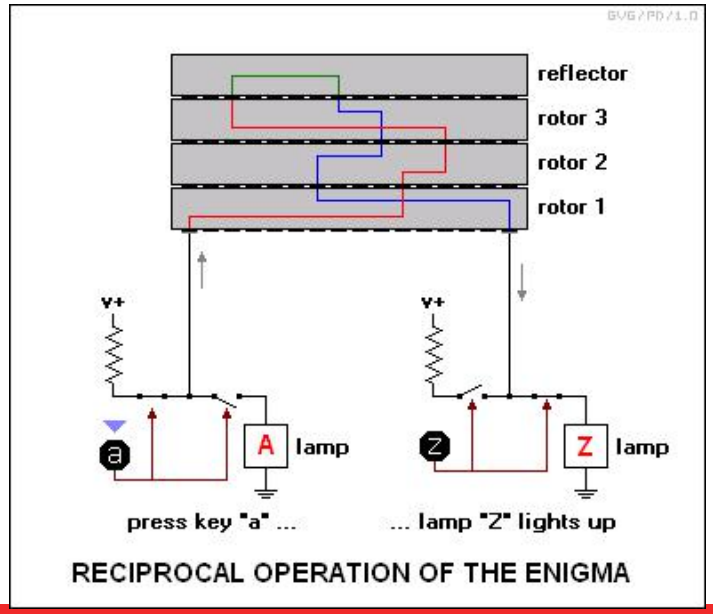
- **Rotores simétricos (bidirecionais) permitem decifras usando cifras duplas**
  - Usando um disco refletor (meio-rotor)



Sarah Witherby, www.flickr.com

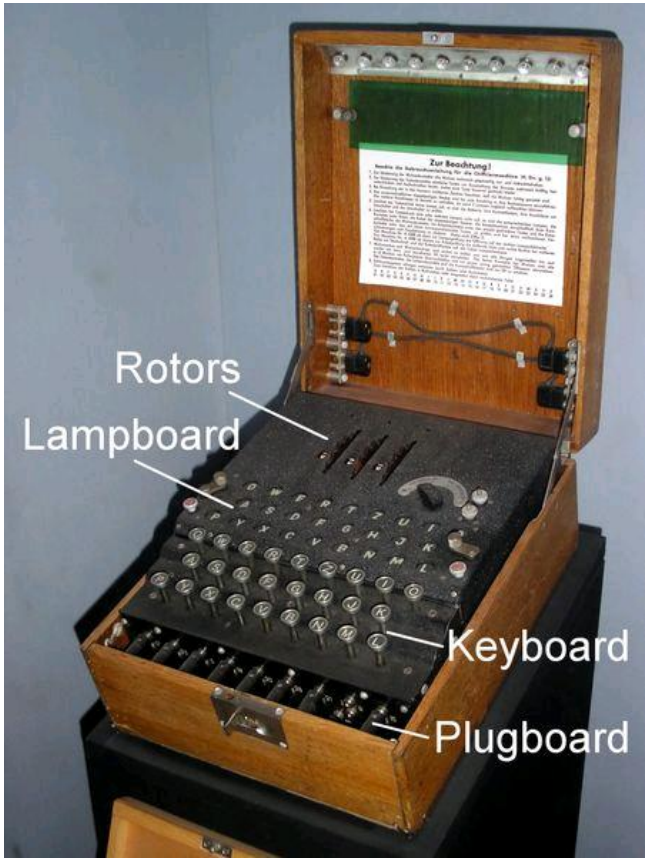
Máquinas de Rotores

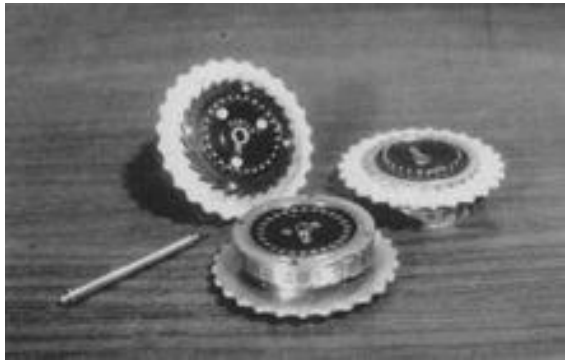
- **Operação recíproca com um refletor**
  - O operador emissor carrega em “A” (o texto em claro) e obtém “Z” como criptograma, o qual é transmitido
  - O operador recetor carrega em “Z” (o criptograma) e obtém “A” como texto em claro
  - Uma letra nunca pode ser cifrada para si própria!



Enigma

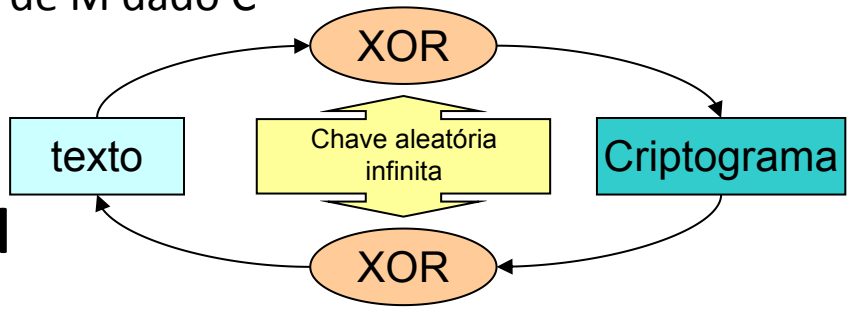
- **Máquina de rotores usada pelos Alemães na 2ª GG**
- **Originalmente apresentada em 1919**
  - Enigma I, com 3 rotores
- **Foram usadas diversas variantes**
  - Com diferentes números de rotores
  - Com cablagem para permutar alfabetos
- **Seleções de chaves distribuídas em livros de códigos**
- **https://observablehq.com/@tmcw/enigma-machine**





Criptografia: Aproximações Teóricas

- **Espaço de texto**
  - Número de combinações de texto diferentes (M)
- **Espaço do criptograma**
  - Número de combinações de criptograma diferentes (C)
- **Espaço das chaves**
  - Número de chaves diferentes para um algoritmo de cifra (K)
- **Cifra perfeita**
  - Dado $c_j \in C$, $H(M \mid C) = H(M)$
    - H (M | C) é a entropia condicional de M dado C
    - H (M) é a entropia de M
  - $\#K \geq \#C \geq \#M$
- **Cifra de Vernam: One-time pad**



Criptografia: Aproximações Práticas

- **Teoricamente seguras vs. seguras na prática**
  - Uso teórico != exploração prática
  - Práticas incorretas podem comprometer boas cifras
  - Exemplo: reutilização de one-time-pads

- **Cifras seguras na prática**
  - A segurança é assegurada pela dificuldade computacional de realizar a criptanálise
    - Usando força bruta
  - Têm uma segurança baseada em limites razoáveis:
    - Custo de uma solução técnica de criptanálise
    - Infraestrutura reservada para a criptanálise
    - Tempo útil de criptanálise

Criptografia: Aproximações Práticas

5 critérios de Shannon

1. **A quantidade de secretismo oferecida**
   - e.g o comprimento da chave
2. **A complexidade na escolha das chaves**
   - e.g. geração da chave, deteção de chaves fracas
3. **A simplicidade da realização**
4. **A propagação de erros**
   - Relevante em ambientes onde surgem erros (e.g. canais de comunicação ruidosos)
5. **A dimensão do criptograma**
   - Relativamente aos respetivos textos originais

Criptografia: Aproximações Práticas

- **Confusão**
  - Complexidade na relação entre o texto, a chave e o criptograma
    - Os bits resultantes (criptograma) devem depender dos bits de entrada (texto e chave) de um forma complexa
- **Difusão**
  - Alteração de grandes porções do criptograma em função de uma pequena alteração do texto
    - Se um bit de texto se alterar, então o criptograma deverá mudar substancialmente, de uma forma imprevisível e pseudoaleatória
  - Efeito de avalanche

Criptografia: Aproximações Práticas

Assumir sempre o pior caso

- **O criptanalista conhece o algoritmo**
  - A segurança está na chave
- **O criptanalista possui grande número de criptogramas gerados com um algoritmo e chave**
  - Os criptogramas não são secretos
- **Os criptanalista conhecem parte dos textos originais**
  - É normal haver alguma noção do texto original
  - Ataques com texto conhecido ou escolhido

Robustez criptográfica

- **A robustez dos algoritmos e a sua resistência a ataques**
  - Ninguém consegue avaliar a robustez de forma precisa
    - Podem especular ou demonstrar usando outras suposições
  - São robustos até que alguém os quebre
  - Existem orientações públicas sobre o que deve/não deve ser usado
    - Antecipar problemas futuros
- **Algoritmos públicos, sem ataques conhecidos, supostamente são mais robustos**
  - Mais investigadores à procura de fraquezas
- **Algoritmos com chaves maiores são tendencialmente mais robustos**
  - Mas frequentemente também são mais lentos.

Robustez criptográfica: AES

- **1997: NIST lançou desafio para o próximo Advanced Encryption Protocol**
  - de conhecimento e utilização públicos, simétrico, chaves de 128, 192 e 256 bits
- **1998: 15 candidatos apresentados por investigadores**
  - CAST-256, Crypton, DEAL, DFC, Frog, HPC, LOKI97, Magenta, MARS, RC6, Rijndael, Safer+, Serpent, Twofish
  - Comunidade tentou encontrar problemas nos candidatos
- **1999: 5 propostas demonstraram ser seguras**
  - MARS, RC6, Rijndael, Twofish
  - Novamente a comunidade tentou encontrar problemas e avaliar a performance
- **2001: Rijndael selecionado como o vencedor**
  - Versões reduzidas do MARS foram quebradas , RC6 e Twofish são seguros
- **2002: Publicado como FIPS PUB 197 e largamente utilizado**

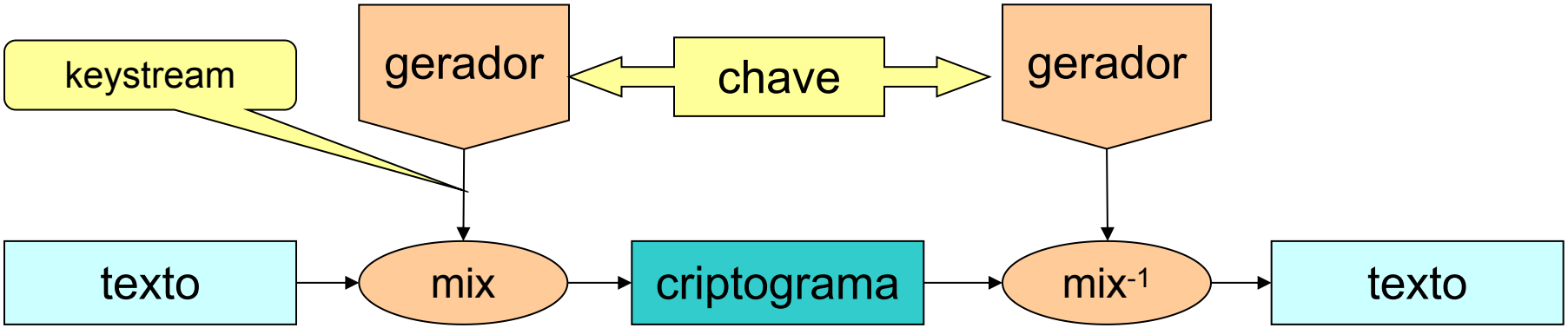
Cifras Contínuas (Stream)

- **Mistura de uma chave contínua (keystream) com o texto ou criptograma**
  - Chave contínua aleatória (cifra de Vernam, com one-time pad)
  - Chave contínua pseudoaleatória (produzida por gerador)
- **Função de mistura invertível**
  - e.g. XOR bit a bit (⊕)

$$C = P \oplus ks \quad P = C \oplus ks$$

- **Cifra poli-alfabética**
  - Cada símbolo da chave contínua define um alfabeto

Cifras Contínuas (Stream)



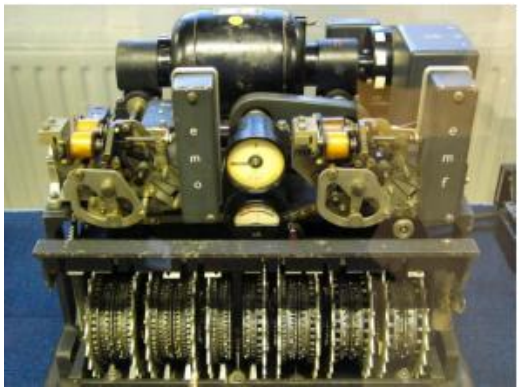
Cifras Contínuas (Stream)

- **Keystream pode ser infinita, mas possui um período**
  - Período depende do gerador
- **Questões práticas de segurança**
  - Cada keystream só pode ser usada **uma vez!**
  - Caso contrário, a soma dos criptogramas fornece a soma dos textos

$$C1 = P1 \oplus Ks,\ C2 = P2 \oplus Ks \rightarrow C1 \oplus C2 = P1 \oplus P2$$

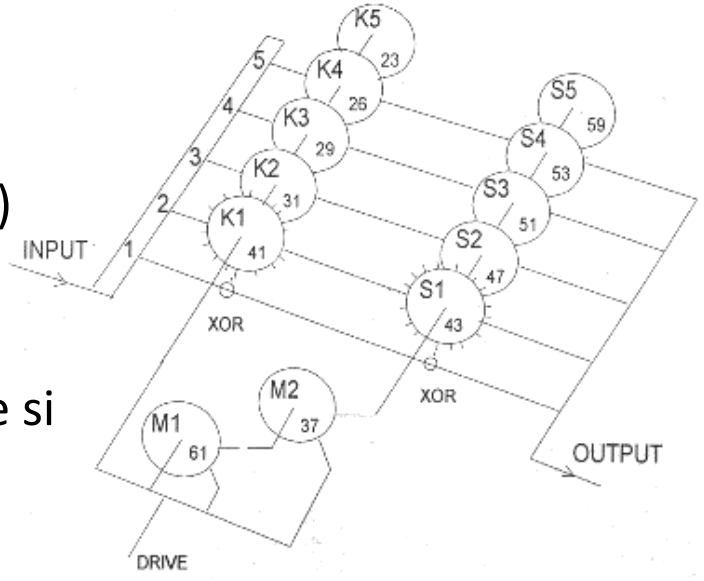
  - Dimensão do texto **tem de ser menor** que o período
    - Exposição da keystream é **total com textos escolhidos/conhecidos**
    - Período permitem analistas conhecer partes do texto
  - Controlo de integridade **é mandatório**
    - **Não existe difusão**, apenas confusão
    - Criptogramas podem ser **manipulados livremente**

Lorenz (Tunny)



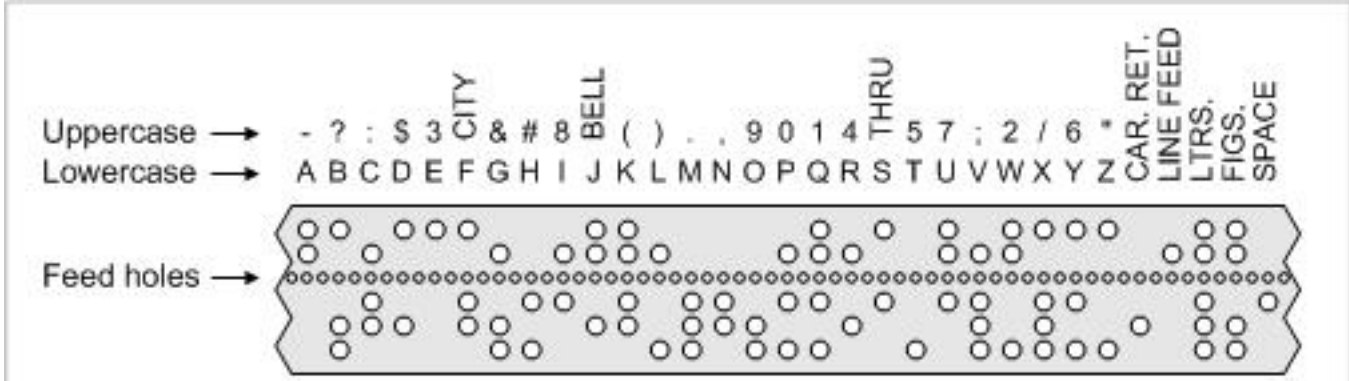
- **Cifra contínua com 12 rotores**
  - Usada pelos alemães durante a 2 G. Guerra
  - Cada caratere de 5 bits é misturado com 5 keystreams

- **Operação**
  - 5 rotores movendo-se regularmente (χ)
  - 5 rotores movendo-se irregularmente (ψ)
  - 2 rotores motorizados
    - para acionar os rotores (ψ)
  - Número de espaços é sempre primo entre si



Criptanálise da Lorenz

- **A estrutura interna não era conhecida**
  - Apenas foi conhecida depois do final da guerra
  - Sabiam que a máquina existia porque intercetavam mensagens cifradas com 5 bits
    - Usando Códigos Baudot de 32 símbolos (e não Morse)



Paper tape showing the five-bit Baudot Code

De interesse: 2014, The Imitation Game

Criptanálise da Tunny

O erro (30 de agosto de 1941)

- **Um operador alemão tinha uma grande mensagem para enviar (~4,000 carateres)**
  - Configurou a sua Lorenz e enviou um indicador de 12 letras (posição inicial dos rotores) para o recetor
  - Depois de ter escrito ~4,000 caracteres, manualmente, recebeu do recetor “envie outra vez” (em texto)
- **O operador emissor recolocou a sua Lorenz na mesma posição inicial**
  - Mesma chave contínua! Completamente proibido!
- **O emissor recomeçou o envio da mensagem, manualmente**
  - Mas escreveu algo ligeiramente diferente! (abreviaturas)

Criptanálise da Tunny

$$C0 = T0 \oplus Ks$$

$$C1 = T1 \oplus Ks$$

$$T1 = C0 \oplus C1 \oplus T0 \rightarrow \text{Variações do Texto}$$

Se parte to texto inicial (T0) for conhecido, as variações podem ser encontradas

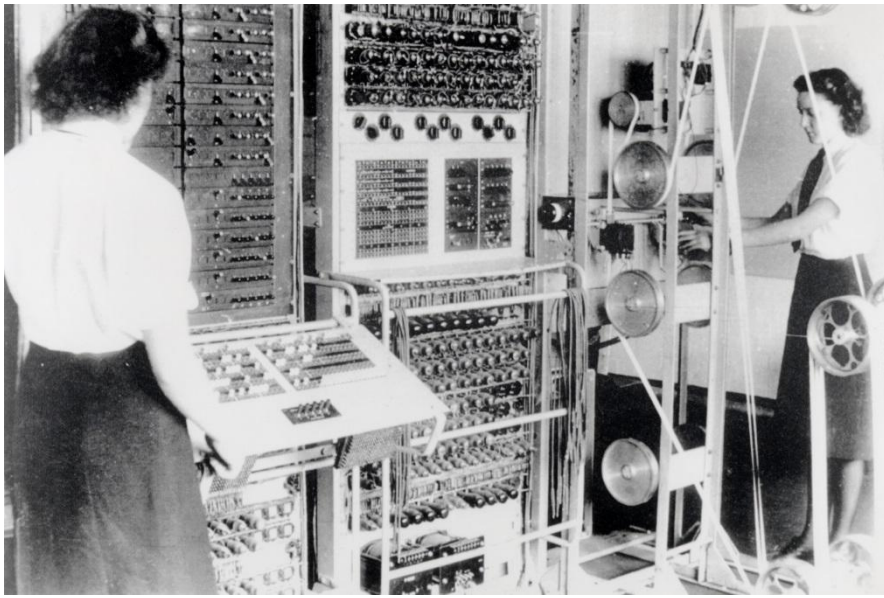
Criptanálise da Tunny

- **A mensagem começava com um texto padrão: SPRUCHNUMMER — número de mensagem**
  - Na primeira vez o operador escreveu: **S P R U C H N U M M E R**
  - Na segunda vez escreveu: **S P R U C H N R**
  - Assim, imediatamente após o N os dois criptogramas eram diferentes!

- **As mensagens foram completamente decifradas por John Tiltman, em Bletchley Park, usando combinações aditivas dos criptogramas (chamados Depths)**
  - A segunda mensagem era cerca de 500 caracteres mais curta que a primeira

- **Assim se conseguiu obter, pela 1ª vez, um exemplar longo de uma chave contínua Lorenz**
  - Tiltman ainda não sabia como a Lorenz operava, apenas sabia que o que tinha era o resultado da sua operação!

Tunny



- **A estrutura da cifra foi deduzida a da chave contínua capturada**
  - Mas a decifra dependia do conhecimento da posição inicial dos rotores
- **Os alemães começaram a usar números para definir o estado inicial dos rotores**
  - Bill Tutte desenvolveu um método para o encontrar
  - A máquina Colossus foi desenvolvida para o aplicar
- **Colossus**
  - Conceção começou em março de 1943
  - O Colossus Mark 1 (1500 válvulas) operacional em jan. de 1944
  - Reduziu o tempo de criptanálise de semanas para horas

Cifras Modernas: Tipos

- **Quanto à operação**
  - Por blocos (mono-alfabéticas)
  - Contínuas (poli-alfabéticas)
- **Quanto ao tipo de chave**
  - Simétricas (chave secreta ou segredo partilhado)
    - Potencialmente sujeitas a caução (escrowing)
  - Assimétricas (chave pública)
- **Combinatória**

	Cifras Por Blocos	**Cifras Contínuas**
Cifras Simétricas		
Cifras Assimétricas		

Cifras Simétricas

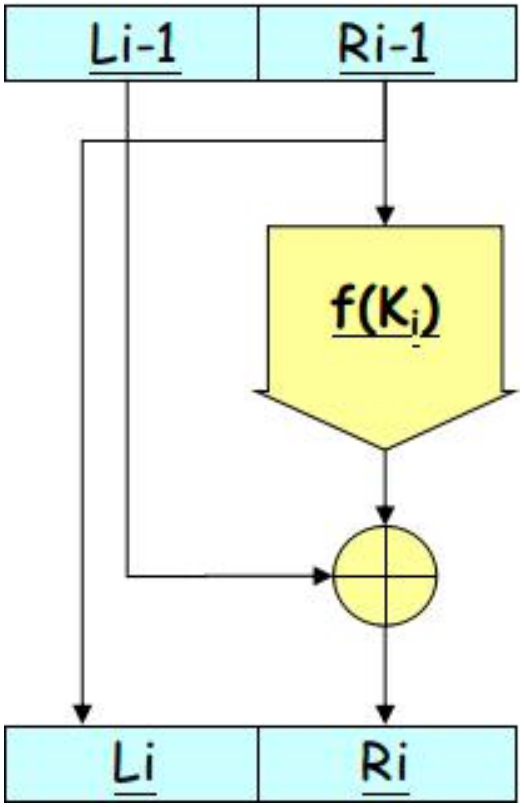
- **Chave secreta: partilhada por 2 ou mais interlocutores**
- **Permitem**
  - Confidencialidade para todos os conhecedores da chave
  - Autenticação de mensagens (cifra por blocos)
    - Quando se usam cifras por blocos
- **Vantagens**
  - Desempenho (normalmente muito eficientes)
- **Desvantagens**
  - N interlocutores, 2 a 2 secretamente -> N x (N-1)/2 chaves
- **Problemas**
  - Distribuição de chaves

Cifras Simétricas por Blocos

- **Aproximações usadas**
  - Blocos de grande dimensão: 64, 128, 256, etc.
- **Difusão, confusão**
  - Permutação, substituição, expansão, compressão
  - Redes de Feistel com múltiplas iterações
  - Ou redes de substituição-permutação
- **Algoritmos mais usados**
  - DES (Data Enc. Stand.), D=64; K=56
  - IDEA (Int. Data Enc. Alg.), D=64; K=128
  - AES (Adv. Enc. Stand., aka Rijndael), D=128, K=128, 192, 256
  - Outros (Blowfish, CAST, RC5, etc.)

Redes de Feistel

$$L_i = R_{i-1} \qquad R_i = L_{i-1}\ f(R_{i-1} \oplus, K_i)$$



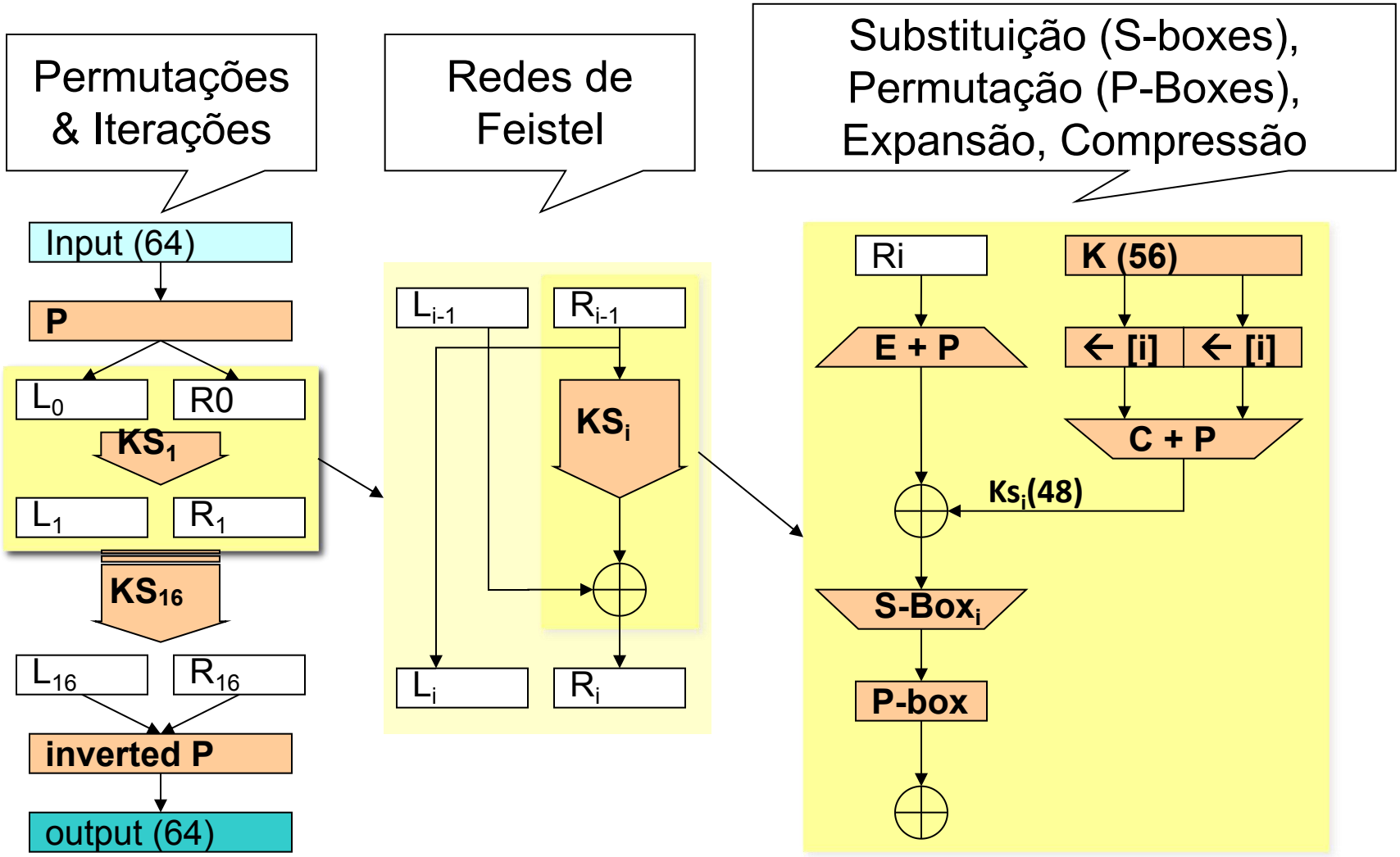
Redes de Substituição-Permutação

- **S-Box: (Substituição) baseado num bit da entrada, troca bits da saída**
  - substituição não é direta (1 para 1)
  - ideal: a alteração de um bit provoca a alteração de todos os bits
  - prática: a alteração de um bit provoca a alteração de pelo menos metade dos bits

- **P-Box: (Permutação) - permuta a posição de bits entre entrada e saída**
  - ideal: permuta a posição de todos os bits

Operação de ambas depende da chave

DES: Data Encryption Standard



DES: robustez

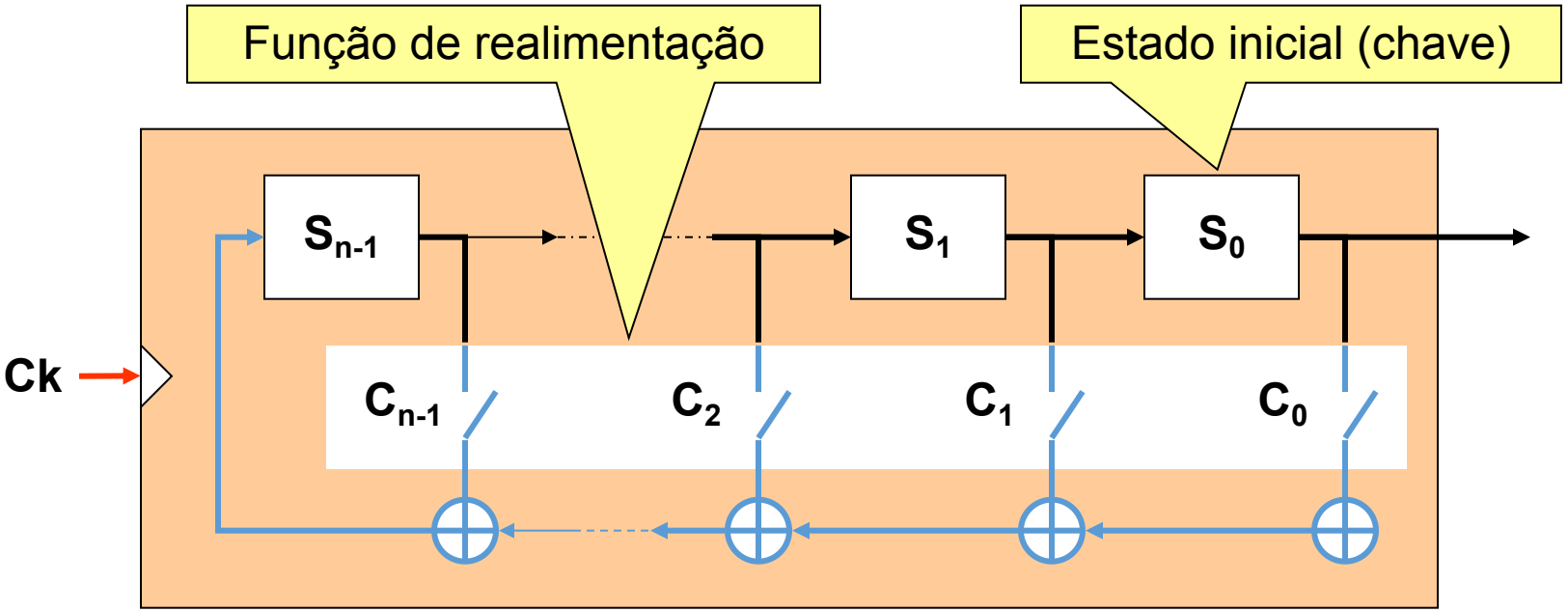
- **Escolha de chaves**
  - Chaves fracas, semi-fracas e quasi-fracas
  - Fáceis de identificar
- **Ataques conhecidos**
  - Pesquisa exaustiva
- **Dimensão das chaves: 56 bits são atualmente insuficientes**
  - A pesquisa exaustiva é técnica e economicamente viável
- **Solução: cifra múltipla**
  - Cifra dupla não é completamente segura (teoricamente ...)
  - Cifra tripla: 3DES (Triple-DES)
    - Com duas ou três chaves
    - Chaves equivalentes de 112 ou 168 bits

Cifras Simétricas Contínuas

- **Aproximações usadas**
  - Desenho de geradores pseudo-aleatórios seguros
    - Baseados em LFSRs
    - Baseados em cifras por blocos
  - Outras aproximações (famílias de funções, etc.)
  - Normalmente sem sincronização
  - Normalmente sem possibilidade de acesso aleatório rápido

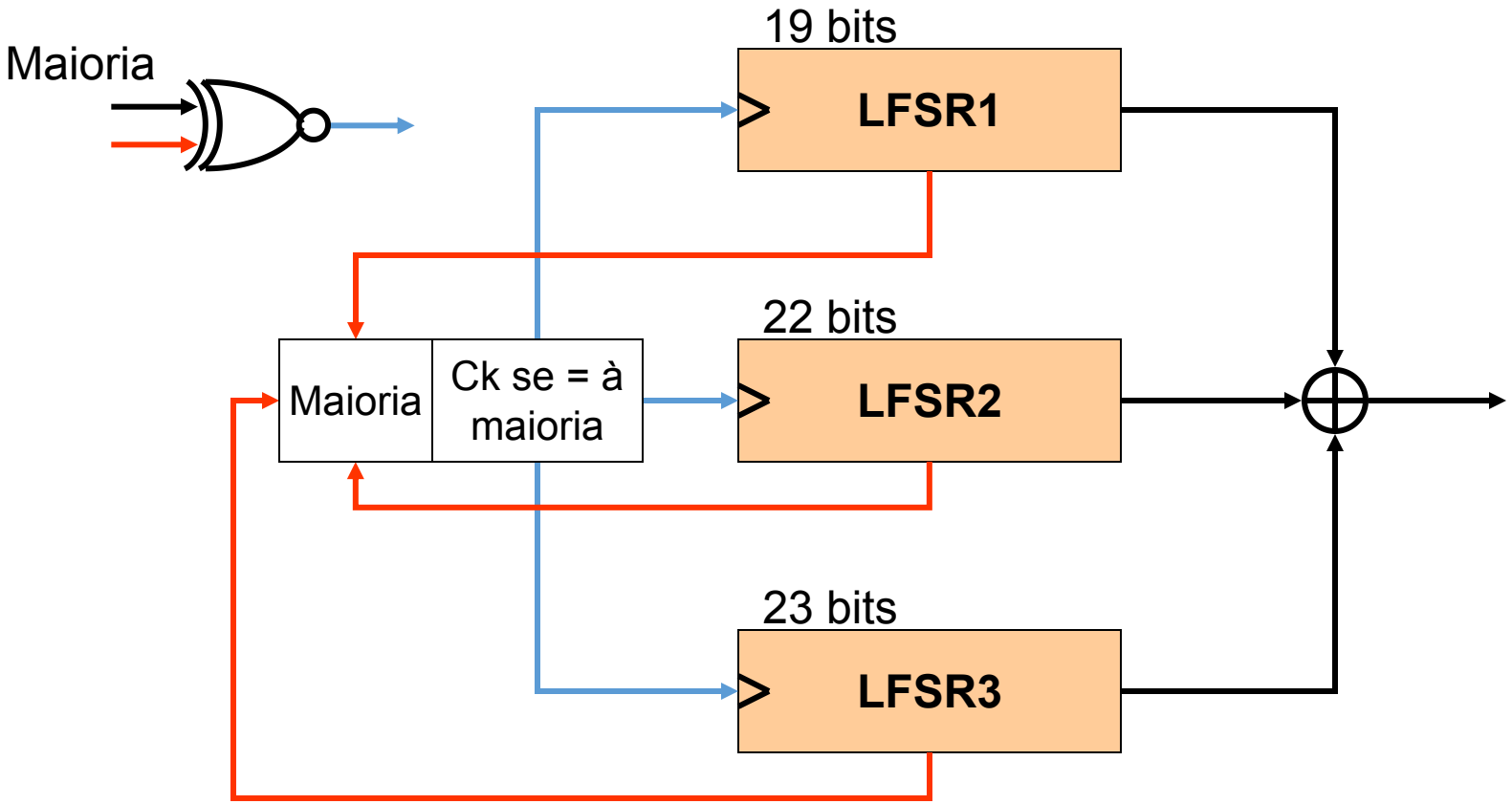
- **Algoritmos mais comuns**
  - A5/1 (US, Europe), A5/2 (GSM)
  - RC4 (802.11 WEP/TKIP, etc.)
  - E0 (Bluetooth BR/EDR)
  - SEAL (c/ acesso aleatório)
  - Chacha20
  - Salsa20

Linear Feedback Shift Register (LFSR)



- **2^n-1 sequências não nulas**
  - Se uma delas possuir um período 2^n-1 então todas o têm
- **Funções de realimentação primitivas**
  - Todas as sequências não nulas têm comprimento 2^n - 1

Geradores com composições de LFSR: A5/1 (GSM)



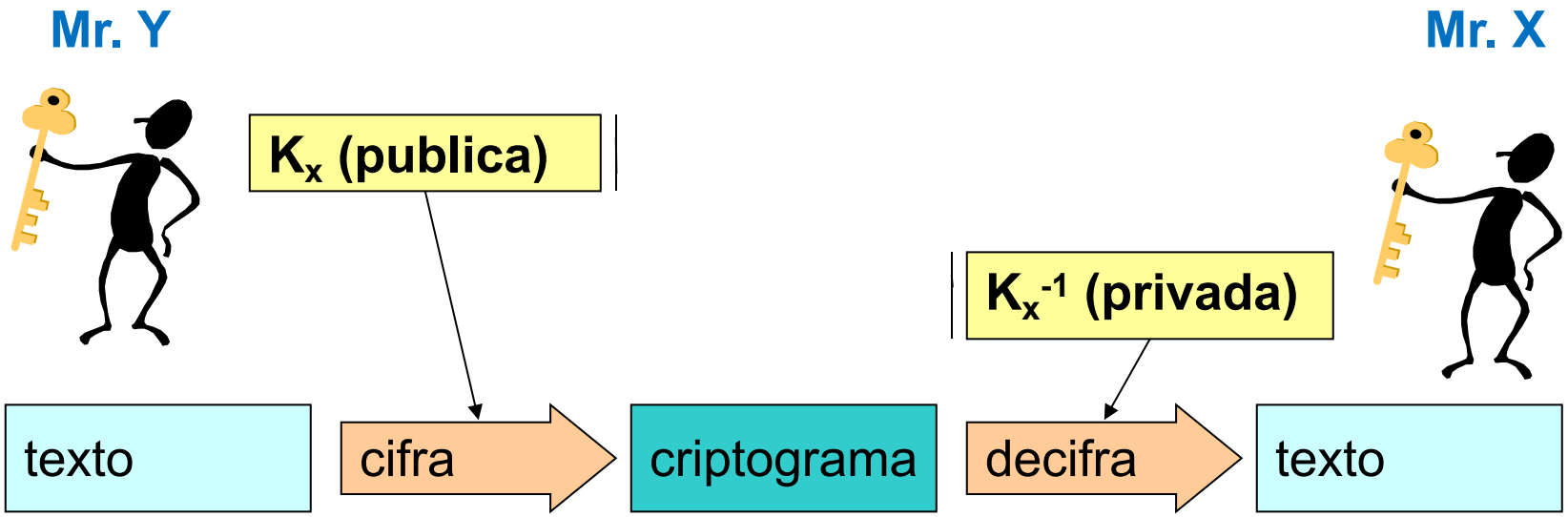
Cifras Assimétricas por Blocos

- **Par de chaves**
  - Uma privada, pessoal e intransmissível
  - Uma pública
- **Permitem**
  - Confidencialidade sem troca de segredos
  - Autenticação
    - De conteúdos (integridade), De autoria (assinaturas digitais)
- **Desvantagens**
  - Desempenho (normalmente pouco eficientes)
- **Vantagens**
  - N interlocutores, N pares de chaves
- **Problemas**
  - Distribuição de chaves públicas
  - Tempo de vida dos pares de chaves

Cifras Assimétricas por Blocos

- **Aproximações: complexidade matemática**
  - Cálculo de logaritmos discretos
  - Fatorização de grandes números
  - Problema da mochila (knapsack)
- **Algoritmos mais usados**
  - RSA
  - ElGamal
  - Curvas elípticas (Elliptic Curve Cryptography, ECC)
- **Outras técnicas com chave pública**
  - Diffie-Hellman (negociação de chaves)

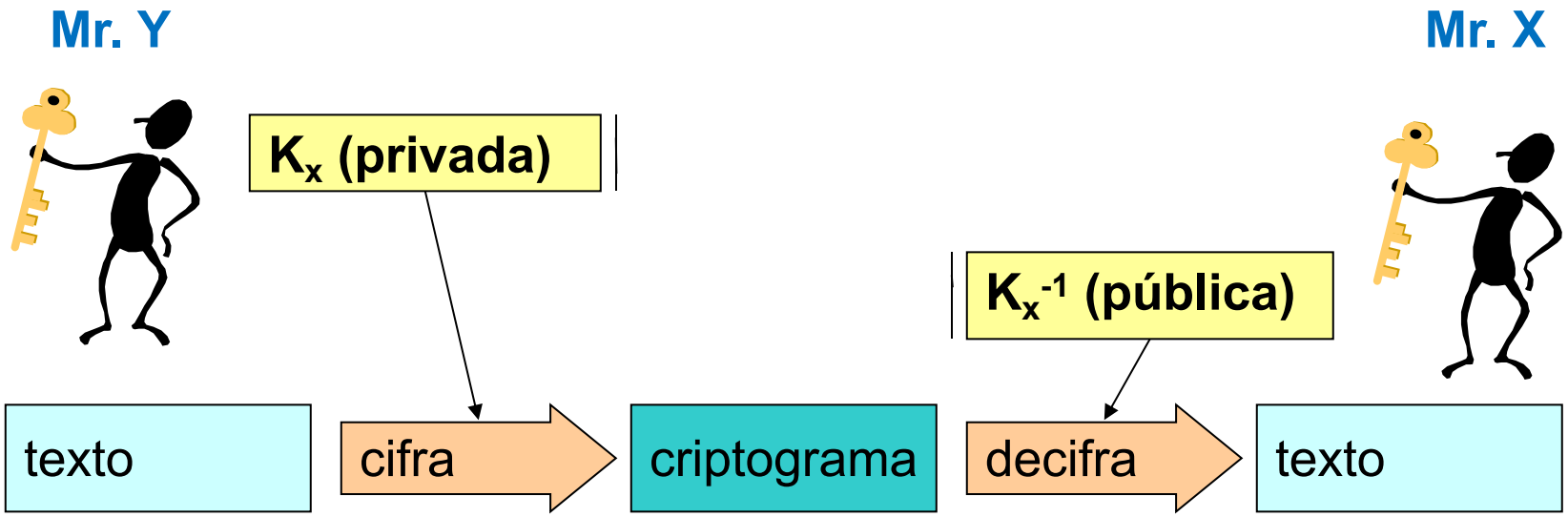
Confidencialidade c/ Cif. Assimétricas



- **Menos chaves**
  - C = E(K, P) P = D(K^{-1}, C)
  - Para ter confidencialidade **Y** basta conhecer a chave pública de **X** (K_X)

- **Não há autenticação de origem**
  - **X** não sabe quem produziu o criptograma
  - Se K_X for efetivamente pública, qualquer um o pode fazer

Autenticidade c/ Cif. Assimétricas



- **O criptograma não pode ser alterado**
  - C = E(K^{-1}, P) P = D(K, C);
  - Só X conhece a chave K_X^{-1} com que foi gerado
- **Não há confidencialidade**
  - Quem conhecer K_X decifra o criptograma
  - Se K_X for verdadeiramente pública, qualquer um o pode fazer

RSA (Rivest, Shamir, Adelman) 1978

- **Complexidade matemática**
  - Dificuldade de Fatorização de grandes números
  - Dificuldade de cálculo de logaritmos discretos

- **Operações e chaves**
  - $K = (e, n)$ $K^{-1} = (d, n)$
  - $C = P^e \bmod n$ $P = C^d \bmod n$
  - $C = P^d \bmod n$ $P = C^e \bmod n$

- **Escolha dos valores das chaves**
  - n de grande dimensão (centenas ou milhares de bits)
  - $n = p \times q$ p e q primos, de grande dimensão
  - Escolher e coprimo de $(p-1) \times (q-1)$
  - Procurar um d tal que $e \times d \equiv 1 \bmod (p-1) \times (q-1)$
  - Não se consegue deduzir d a partir de e ou de n

RSA (Rivest, Shamir, Adelman) 1978

- **p = 5 q = 11 (pequenos números primos)**
  - n = p x q = 55
  - (p-1)(q-1) = 40
- **e = 3**
  - Coprimo de 40
- **d = 27**
  - e x d ≡ 1 mod 40
- **P = 26 (note que P, C∈ [0, n-1])**
  - C = P^e mod n = 26^3 mod 55 = 31
  - P = C^d mod n = 31^{27} mod 55 = 26

ElGamal - 1974

- **Semelhante ao RSA**
  - Baseado apenas na dificuldade de cálculo de logaritmos discretos
- **Uma variante é muito usada em assinaturas digitais**
  - DSA (*Digital Signature Algorithm*)
  - US *Digital Signature Standard* (DSS)
- **Operações e chaves (para assinaturas)**
  - $\beta = \alpha^x \bmod p$ $K = (\beta, \alpha, p)$ $K^{-1} = (x, \alpha, p)$
  - k aleatório, $k \times k^{-1} \equiv 1 \bmod (p-1)$
  - Assinatura de M: (γ,δ) $\gamma = \alpha^k \bmod p$ $\delta = k^{-1} (M - x\gamma) \bmod (p-1)$
  - Validação da assinatura de M: $\beta^\gamma \gamma^\delta \equiv \alpha^M \pmod p$
- **Problema**
  - O valor de k precisa de ser secreto
  - O seu conhecimento revela x!

Diffie-Hellman

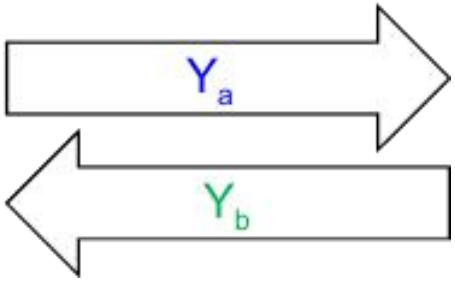
q (primo de elevada dimensão)

α (raiz primitiva mod q)

a = random

$Y_a = \alpha^a \bmod q$

$K_{ba} = Y_b^{\,a} \bmod q$



b = random

$Y_b = \alpha^b \bmod q$

$K_{ab} = Y_a^{\,b} \bmod q$

$K_{ba} = K_{ab}$

Diffie-Hellman - Ataque por MitM

a = random

$Y_a = \alpha^a \bmod q$

$K_{ca} = Y_c^{\,a} \bmod q$

Y_a

Y_c

c = random

$Y_c = \alpha^c \bmod q$

$K_{ac} = Y_a^{\,c} \bmod q$

$K_{cb} = Y_b^{\,c} \bmod q$

Y_b

Y_c

b = random

$Y_b = \alpha^b \bmod q$

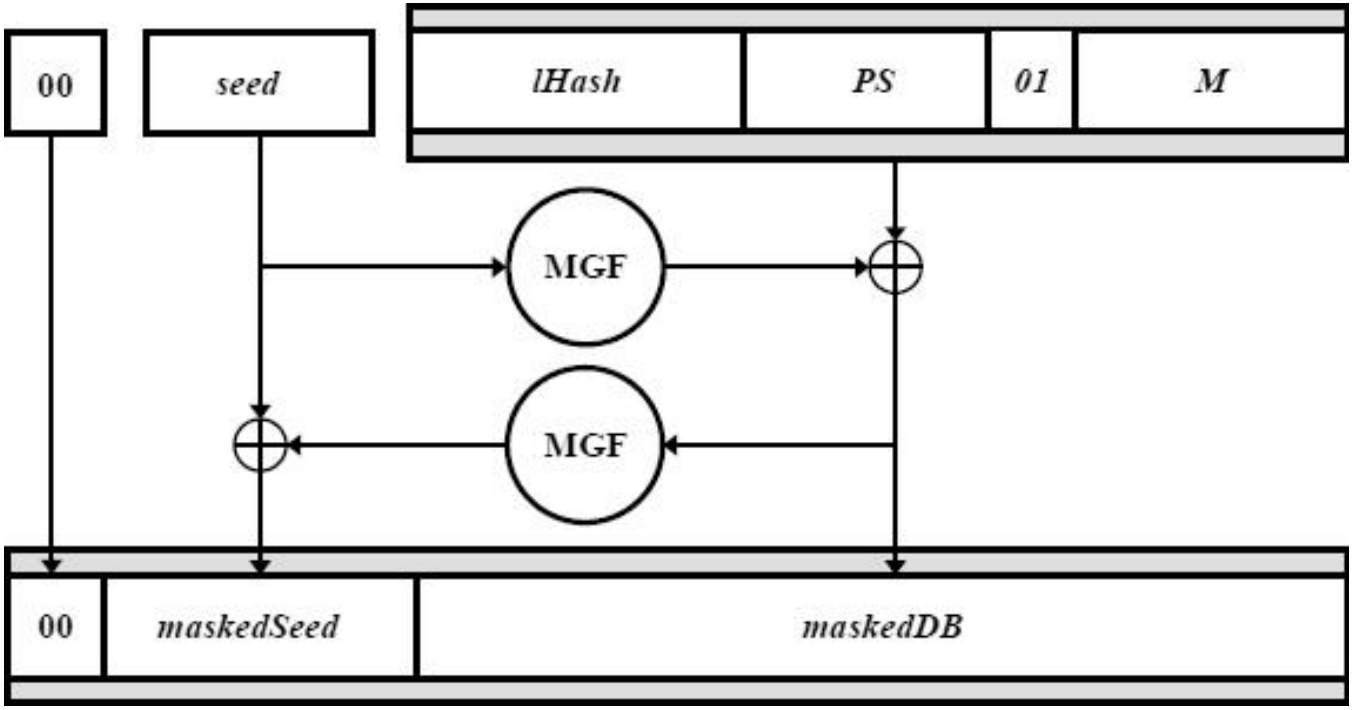
$K_{cb} = Y_c^{\,b} \bmod q$

Randomização de cifras com chave pública

- **O resultado de uma cifra com chave pública não deverá ser determinístico (previsível)**
  - **N** cifras do mesmo valor, coma mesma chave, devem produzir **N** resultados diferentes
  - Objetivo: impedir a descoberta de valores cifrados por tentativa e erro
- **Técnicas**
  - Concatenação do valor a cifrar com dois valores
    - Um fixo (para controlo de erros)
    - Um aleatório (para randomização)

Randomização de cifras com chave pública: OAEP

- **lHash: Digest sobre Label**
- **seed: Random**
- **PS: zeros**
- **M: Texto**
- **MGF: Mask Generation Function**



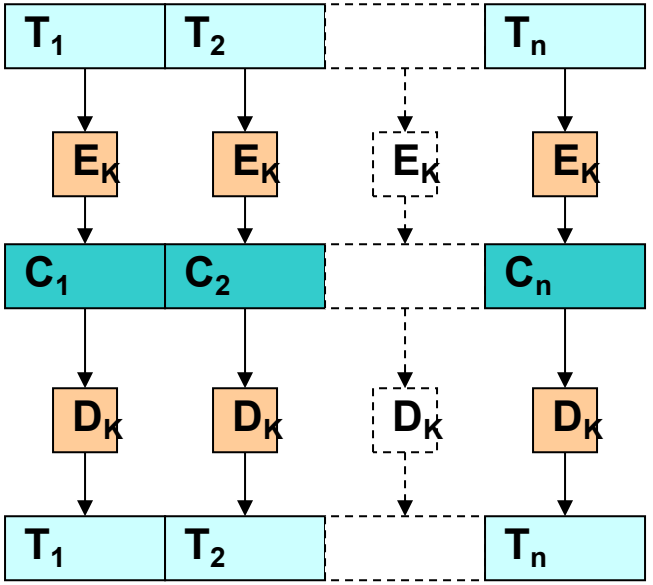
Utilização de cifras por blocos: Modos

- **Propostos inicialmente para o DES**
  - ECB (Electronic Code Block)
  - CBC (Cipher Block Chaining)
  - OFB (Output Feedback Mode)
  - CFG (Cipher Feedback Mode)
- **Modos podem ser usados com outras cifras (em teoria)**
- **Podem existir outros modos:**
  - CTR (Counter Mode)
  - GCM (Galois/Counter Mode)
  - Tweaks...

Modos: Electronic Code Block

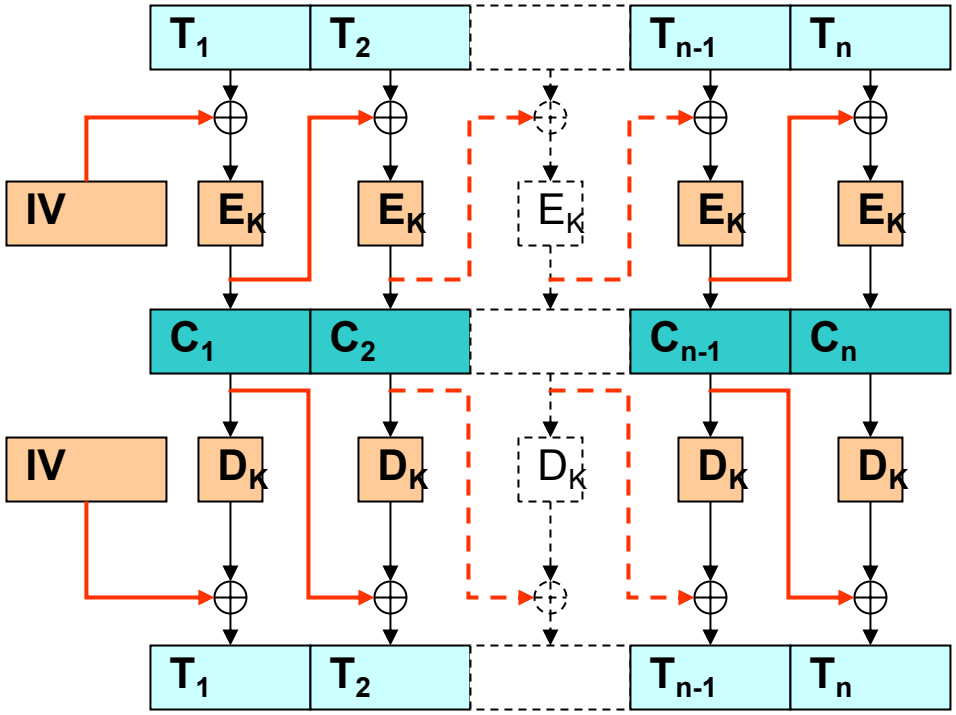
- **Cifra direta de cada bloco: $C_i = E_k(T_i)$**
- **Decifra direta de cada bloco: $T_i = D_k(C_i)$**
- **Blocos são independentes**
  - Sem feedback
- **Problema:**

se $T_1 = T_2$ então $C_1 = C_2$

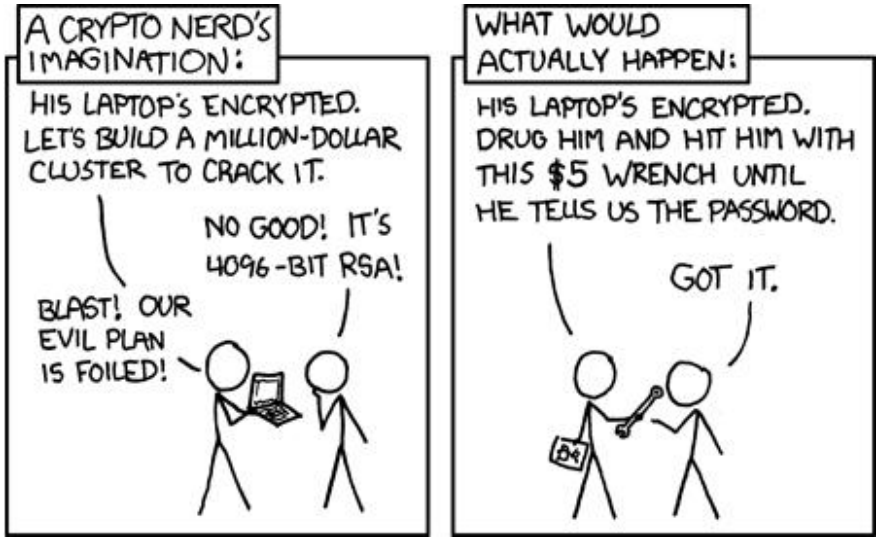


Modos: Cipher Block Chaining (CBC)

- **Cifra de cada bloco T_i com feedback de C_{i-1}**
  - $C_i = E_K(T_i \oplus C_{i-1})$
- **Decifra de cada bloco C_i com feedback de C_{i-1}**
  - $T_i = D_K(C_i) \oplus C_{i-1}$
- **Bloco inicial usa IV**
  - Initialization Vector
  - Valor aleatório
  - Pode estar em claro

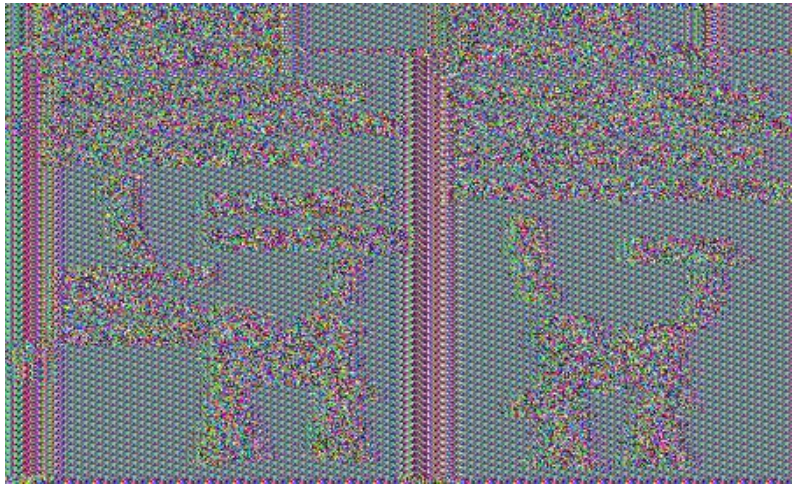


ECB vs CBC: Propagação de Padrões

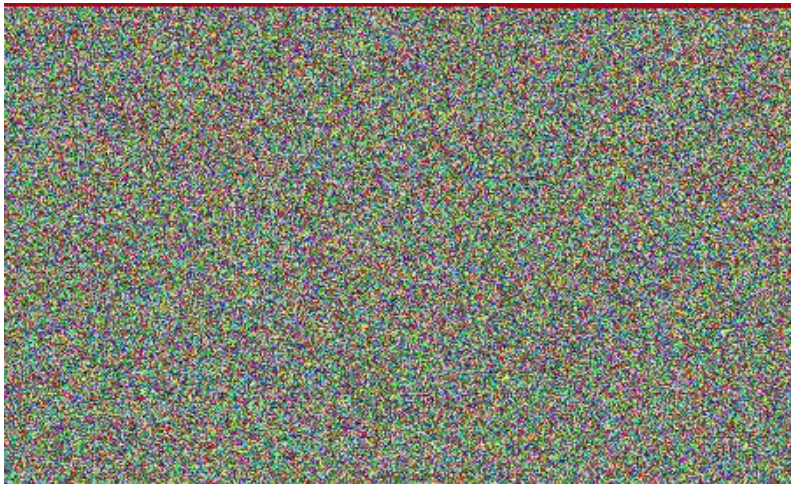


https://xkcd.com/538/

ECB



CBC

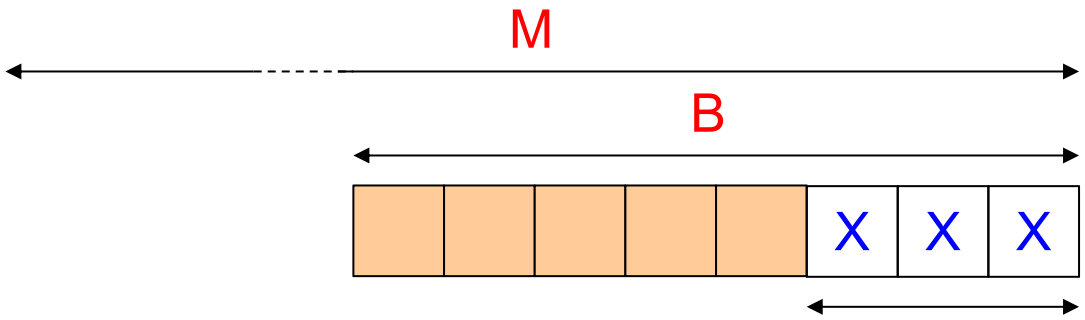


Modos: ECB/CBC problemas de alinhamento

- **Modos ECB/CBC necessitam de textos com dimensão múltipla da dimensão do bloco**
  - Cifra é aplicada por blocos de texto
- **Blocos incompletos (o último) necessitam de tratamento diferenciado**
  - na cifra e na decifra
- **Resultado é um bloco**
  - Criptograma pode ser maior do que o texto

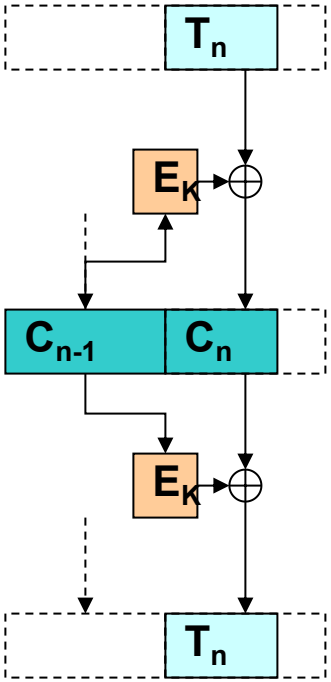
Modos: ECB/CBC problemas de alinhamento

- **Alternativa: Excipiente (Padding)**
- **PKCS#7**
  - **X = B - (M mod B)**
  - **X bytes extra, com valor X**
  - **Se M mod B = 0, adicionar um bloco inteiro com valor B**
- **PKCS#5: igual a PKCS#7 mas com B=8**



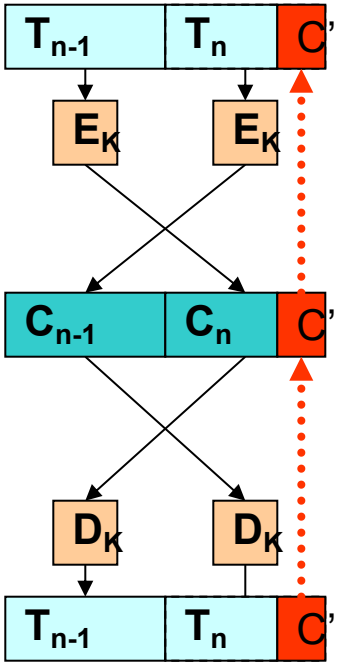
Modos: ECB/CBC problemas de alinhamento

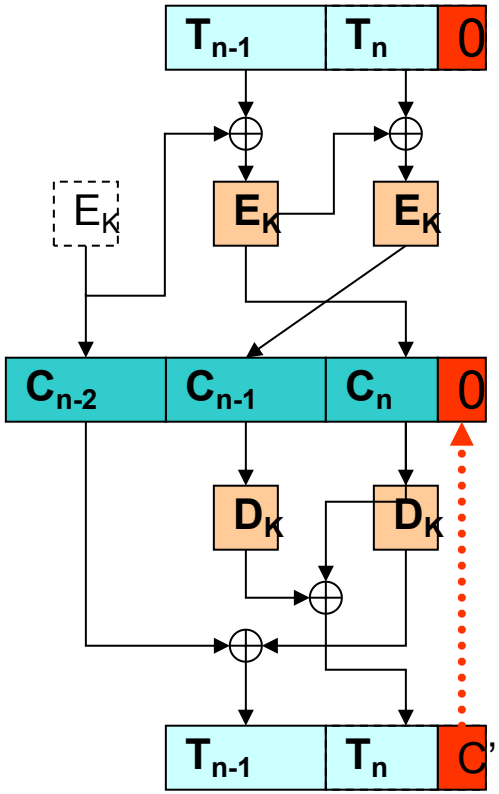
- **Cifrar o último bloco de forma diferenciada**
  - usar um processo semelhante a uma cifra contínua



Modos: ECB/CBC problemas de alinhamento

- **Ciphertext Stealing**
  - Troca ordem de cifra/decifra dos dois últimos blocos
  - a) Usa parte do criptograma do penúltimo para preencher último
  - b) Usa excipiente fixo e cifra contínua antes de cifra por blocos





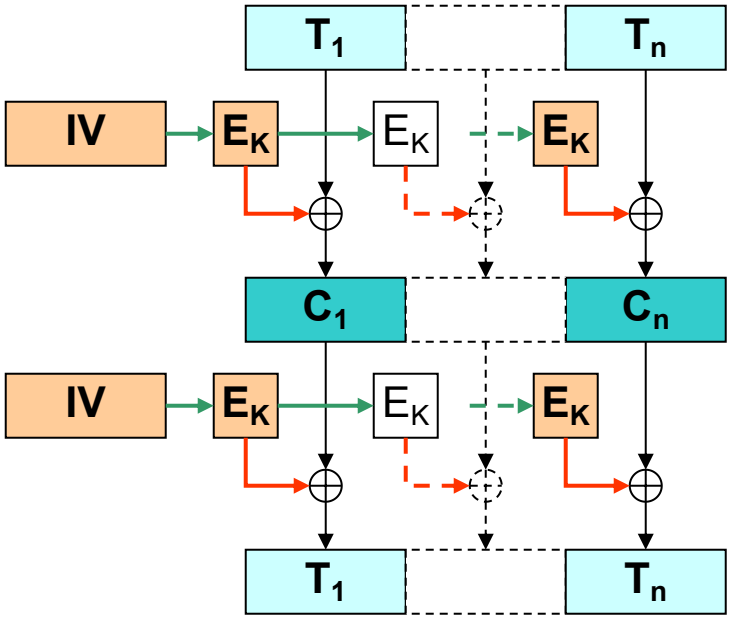
Modos: n-bit OFB (Output Feedback)

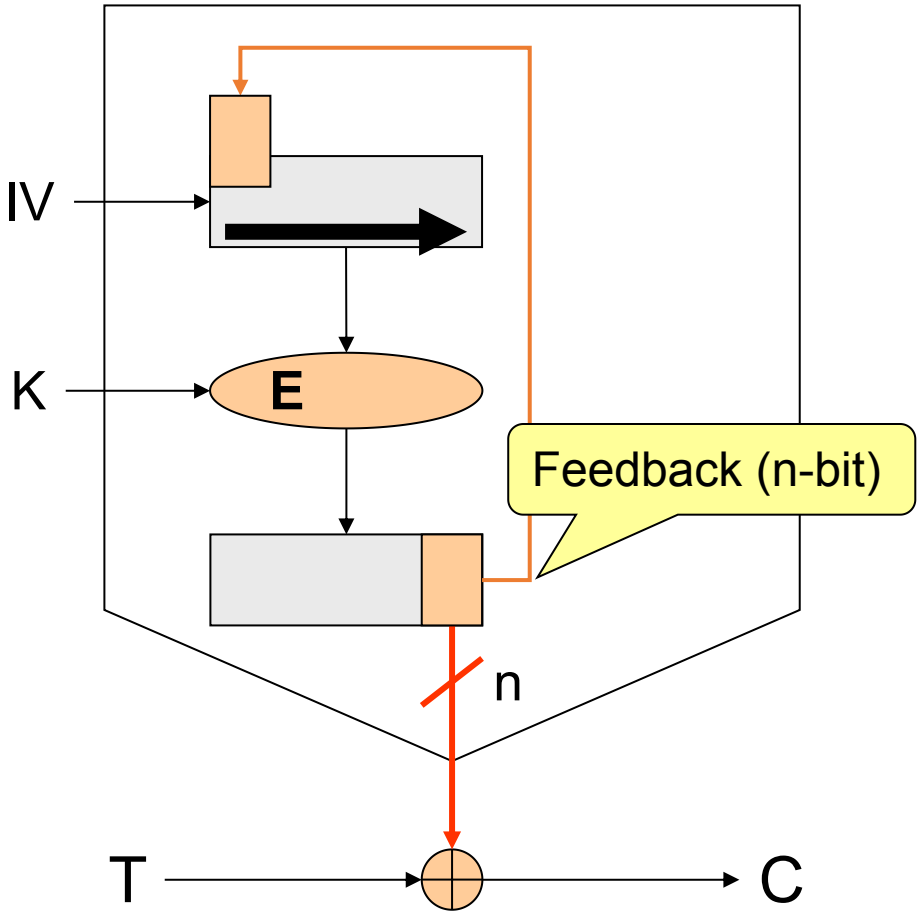
$C_i = T_i \oplus E_K(S_i)$

$T_i = C_i \oplus E_K(S_i)$

$S_i = f(S_{i-1}, E_K(S_{i-1}))$

$S_0 = IV$





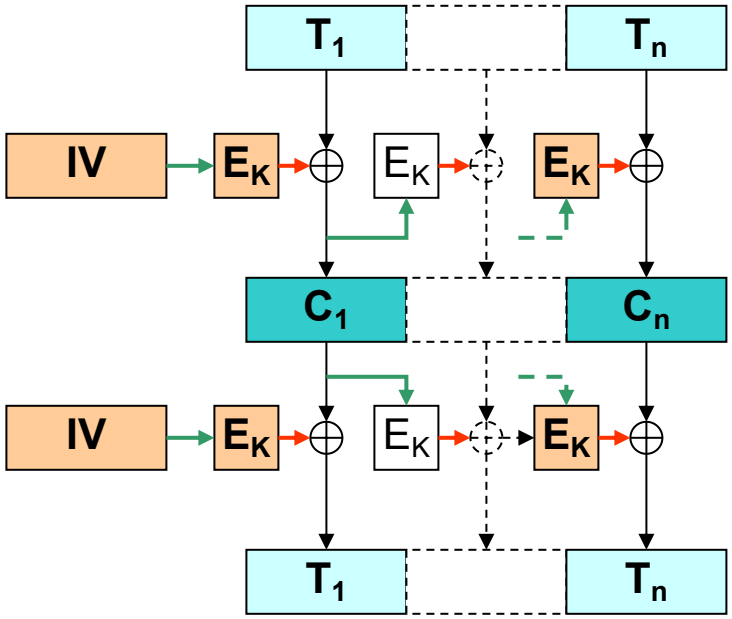
Modos: n-bit CFB (Ciphertext Feedback)

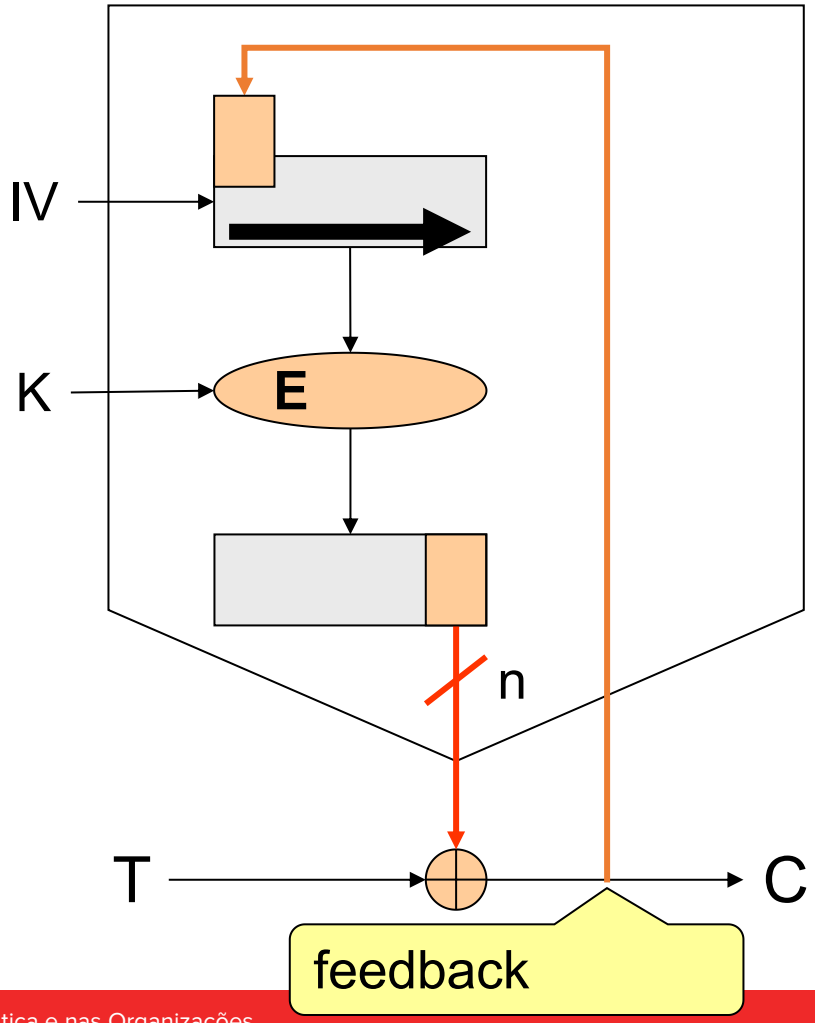
$C_i = T_i \oplus E_K(S_i)$

$T_i = C_i \oplus E_K(S_i)$

$S_i = f(S_{i-1}, C_i)$

$S_0 = IV$





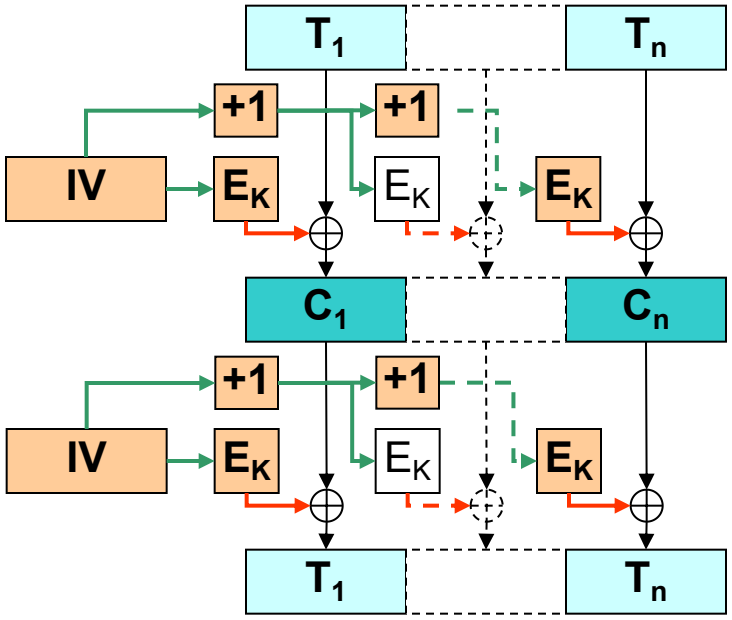
Modos: n-bit CTR (Counter)

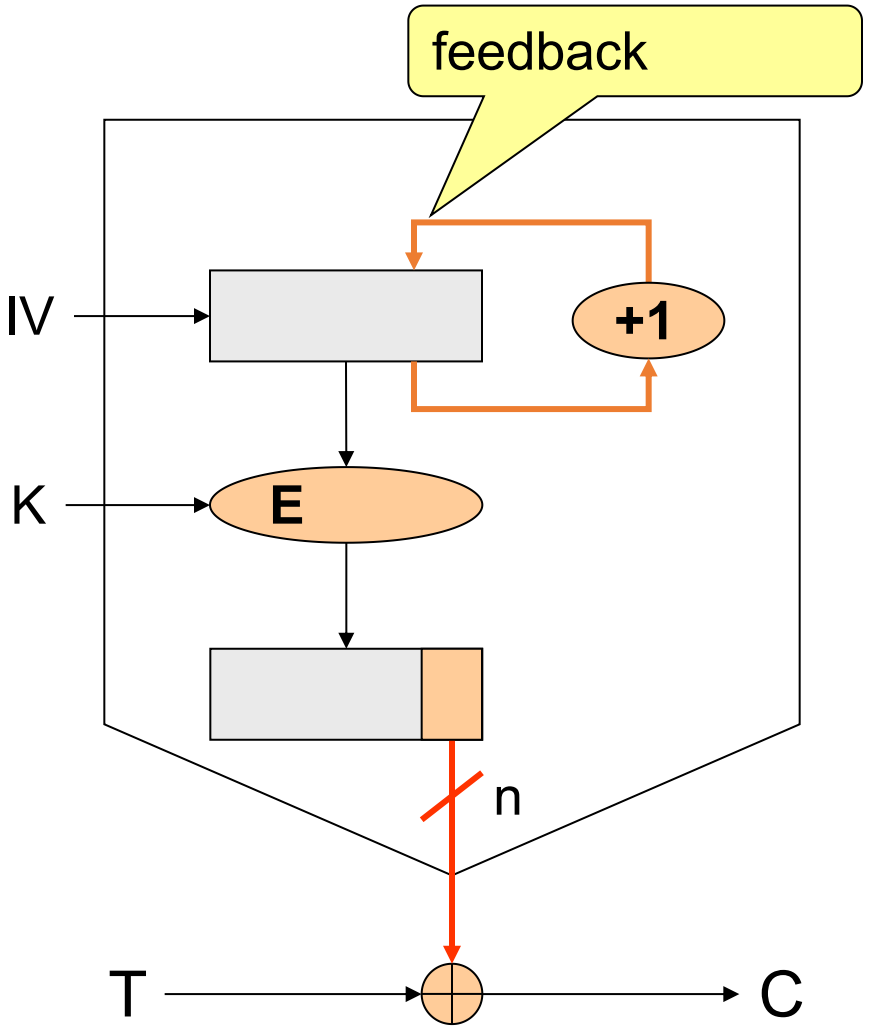
$C_i = T_i \oplus E_K(S_i)$

$T_i = C_i \oplus E_K(S_i)$

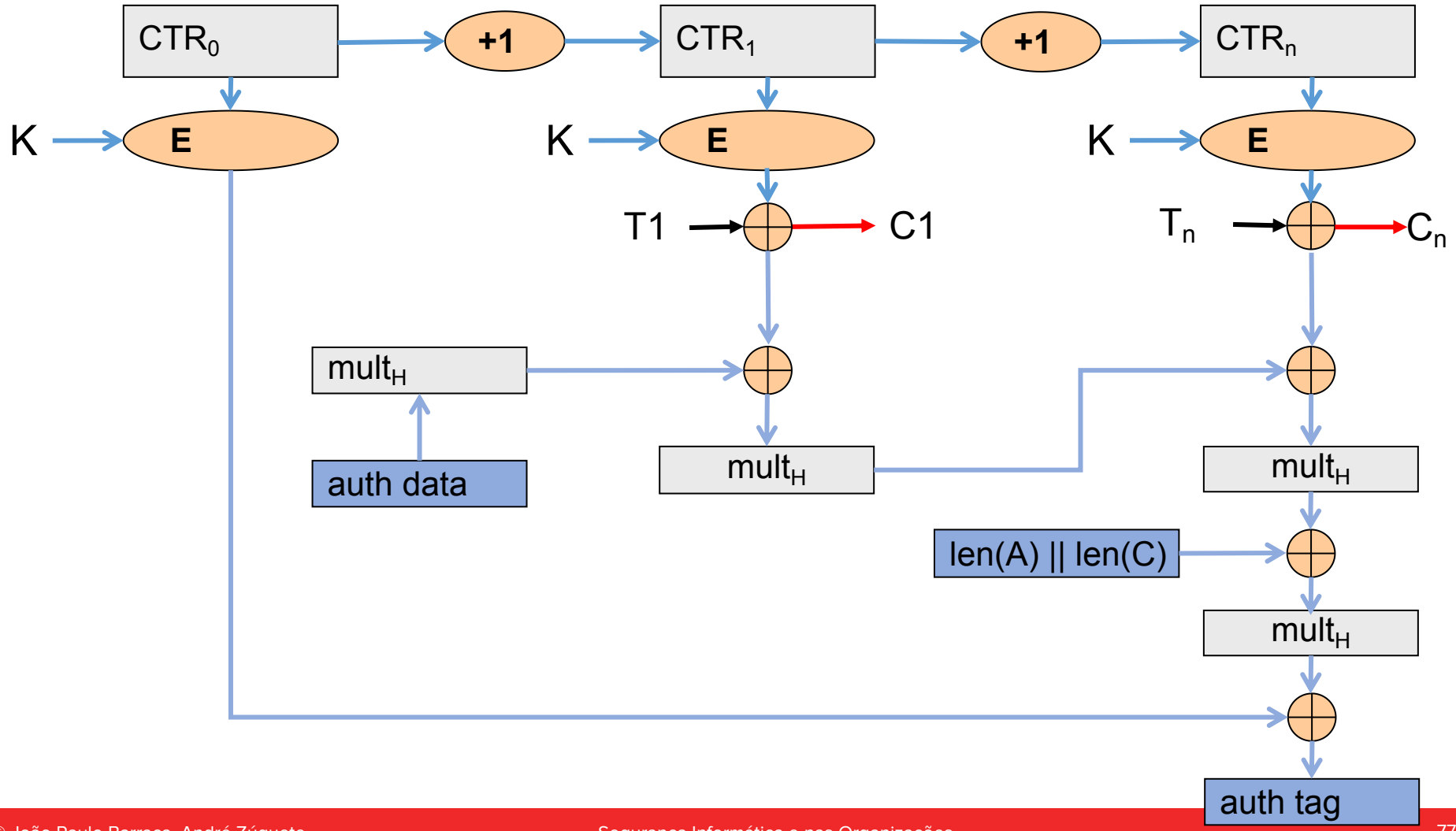
$S_i = S_{i-1}+1$

$S_0 = IV$





Modos: Galois w/ Counter Mode (GCM)



Modos: Comparação

	Bloco		Contínua			
	ECB	**CBC**	**OFB**	**CFB**	**CTR**	**GCM**
Ocultação de padrões no texto		✓	✓	✓	✓	✓
Confusão na entrada da cifra		✓		✓	Contador Secreto	Contador Secreto
Mesma chava para mensagens diferentes	✓	✓	Outro IV	Outro IV	Outro IV	Outro IV
Dificuldade de alteração	✓	✓ (...)				✓
Pré-processamento			✓		✓	✓
Paralelização	✓	decrifra	com pré. proc.	decifra	✓	✓
Acesso aleatório uniforme						
Propagação de erros		próximo bloco		alguns bits seguintes		detetado
Capacidade de re-sincronização	Perda de blocos	perda de blocos		perda de múltiplos n-bits		detetado

Modos: Reforço da Segurança

Cifra Múltipla

- **Cifra dupla**
  - Violável por intromissão em 2^{n+1} tentativas
    - Com 2 ou mais blocos de texto conhecido
    - Usando 2^n blocos de memória ...
  - Não é (teoricamente) muito mais segura ...

- **Cifra tripla (EDE): $C_i = E_{K1}(D_{K2}(E_{K3}(T_i)))$ $P_i = D_{K3}(E_{K2}(D_{K1}(C_i)))$**
  - Normalmente usa-se $K_1=K_3$
  - Se $K_1=K_2=K_3$ transforma-se numa cifra simples

Modos: Reforço da Segurança (Cifra dupla)

Ataque Meet in the midle

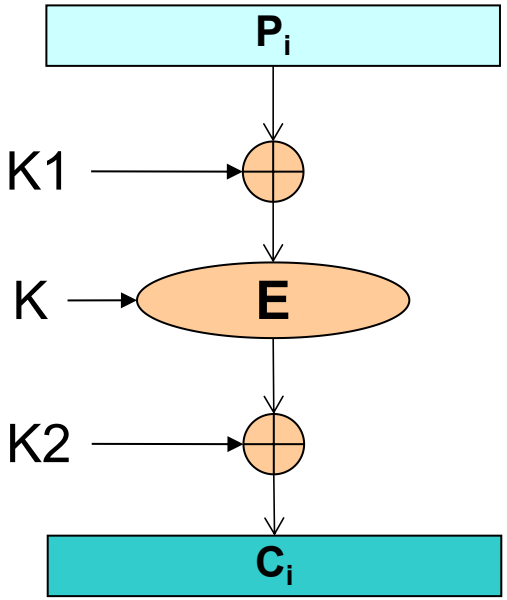
- **Cifra dupla com duas chaves K_a e K_b**
  - $C = E_b(k_b, E_a(k_a, T))$
  - $T = D_a(k_a, D_b(k_b, C))$
- **Logo: $D_b(k_b, C) = E_a(k_a, T)$**
- **Se C e T forem conhecidos, podem-se calcular:**
  - Todos os valores $D_b(k_b, C)$, variando k_b
  - Todos os valores $E_a(k_a, T)$, variando k_a
- **Chaves encontradas quando se verificar a igualdade**
  - Complexidade esperada: $2^{len(ka) + len(kb)}$
  - Complexidade real: $2^{len(ka)} + 2^{len(kb)}$

Modos: Reforço da Segurança

Branqueamento/whitening

Técnica simples e eficiente de introdução de confusão

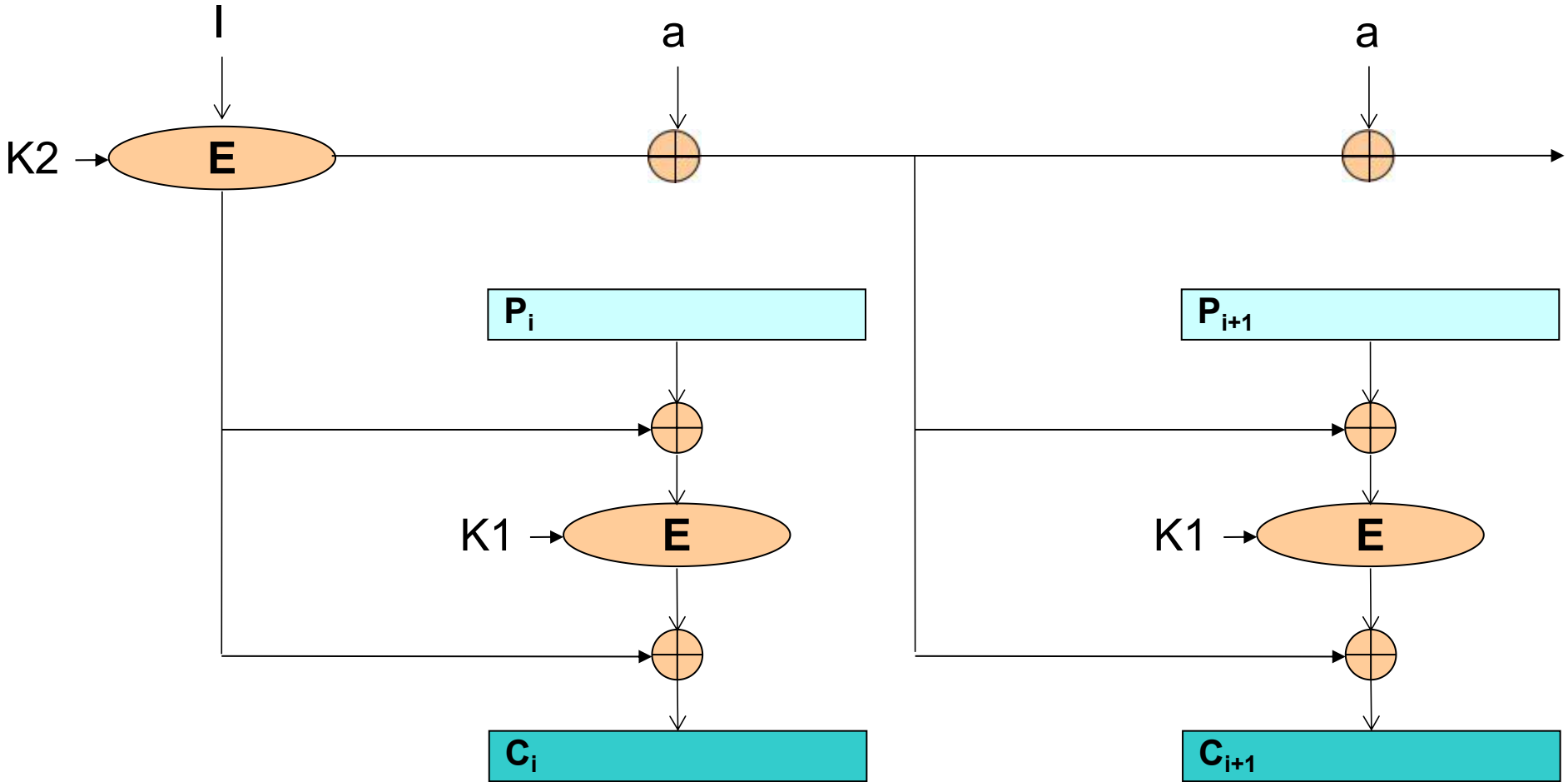
- $C_i = E_K(K_1 \oplus T_i) \oplus K_2$
- $T_i = K_1 \oplus D_K(K_2 \oplus C_i)$



Modos: Reforço da Segurança

XOR-Encrypt-XOR (XEX)

XTS = XEX + Ciphertext Stealing



Aumento de performance: Cifra Híbrida

- **Combinação de Cifra Assimétrica com Simétrica**
  - Usar o melhor de dois mundos, evitando os problemas
  - Cifra Assimétrica: utilização de chaves públicas (mas lenta)
  - Cifra Simétrica: Rápida (mas com fraca troca de chaves)

- **Aproximação:**
  1. Obter K_{pub} do destinatário
  2. Gerar K_s de forma aleatória
  3. Calcular $C_1 = E_{sim}(K_s, T)$
  4. Calcular $C_2 = E_{asim}(K_{pub}, K_s)$
  5. Enviar $C_1 + C_2$
     - C1 = Texto cifrado com chave simétrica
     - C2 = Chave simétrica cifrado com chave pública do destinatário
       - Também pode conter o IV

Funções de Síntese (digest)

- **Resultado de dimensão constante com entradas de dimensão variável**
  - Uma espécie de "impressão digital" dos textos
- **Resultados muito diferentes para entradas similares**
  - Funções de dispersão criptográficas unidirecionais
- **Propriedades relevantes:**
  - Resistência à descoberta de um texto
    - Dada uma síntese, é difícil encontrar um texto que o produza
  - Resistência à descoberta de um 2º texto
    - Dado um texto, é difícil encontrar um segundo texto com a mesma síntese
  - Resistência à colisão
    - É difícil encontrar dois textos com a mesma síntese
    - Paradoxo do aniversário

Funções de Síntese: Dimensão

- **Considerando o textos:**
  - T1:“Hello User_A!”, T2:“Hello User_B!”, T3:“Hello User_XY!”
- **Diferentes algoritmos produzem valores de dimensão diferente, mas independente da dimensão do texto**
  - MD5:
    - T1: `70df836fdaf02e0dfc990f9139762541`
    - T3: `a08313b553d8bf53ca7457601a361bea`
  - SHA-1:
    - T1: `f591aa1eabcc97fb39c5f422b370ddf8cb880fde`
    - T3: `c28b0520311e471200b397eaa55f1689c8866f25`
  - SHA-256:
    - T1: `9649d8c0d25515a239ec8ec94b293c8868e931ad318df4ccd0dffd67aff89905`
    - T3: `8fc49cde23d15f8b9b1195962e9ba517116f45661916a0f199fcf21cb686d852`

Funções de Síntese: Dimensão

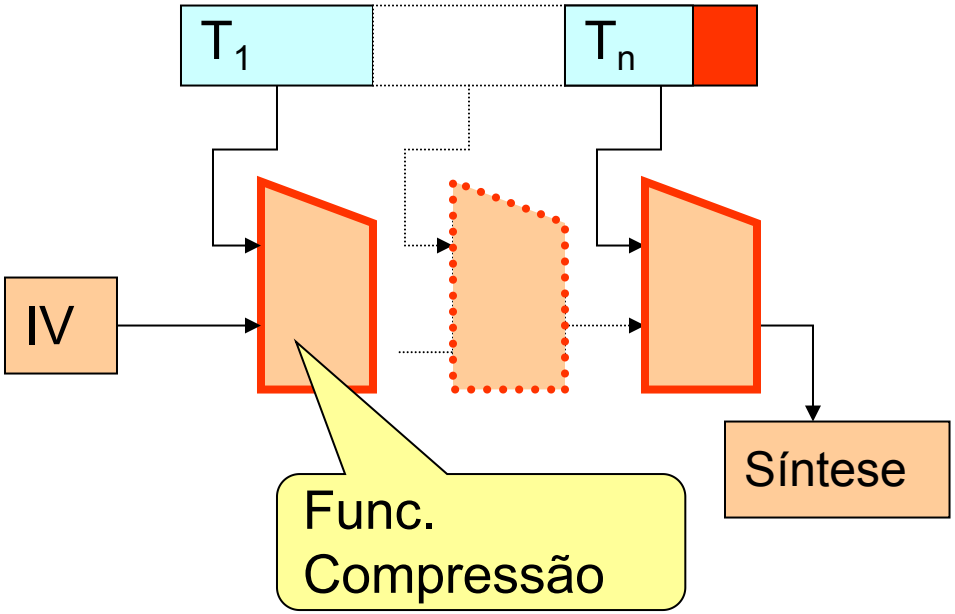
- **Considerando o textos:**
  - T1:“Hello User_A!”, T2:“Hello User_B!”, T3:“Hello User_XY!”
- **Uma pequena alteração no texto produz uma alteração drástica no resultado**
  - MD5:
    - T1: `70df836fdaf02e0dfc990f9139762541`
    - T2: `c32e0f62a7c9c815063d373acac80c37`
  - SHA-1:
    - T1: `f591aa1eabcc97fb39c5f422b370ddf8cb880fde`
    - T2: `bab31eb62f9612667585240 71a7ad8221bc8700b`
  - SHA-256:
    - T1: `9649d8c0d25515a239ec8ec94b293c8868e931ad318df4ccd0dffd67aff89905`
    - T2: `e663a01d3bec4f35a470aba4baccece79bf484b5d0bffa88b59a9bb08707758a`

Funções de Síntese (digest)

- **Aproximações**
  - Difusão e confusão em funções de compressão
  - Construção Merkle-Damgård
    - Compressão iterativa
    - Padding com o comprimento

- **Algoritmos mais comuns**
  - MD5 (128 bits)
    - Já não é seguro! É fácil descobrir colisões!
  - SHA-1 (Secure Hash Algorithm, 160 bits)
    - Já não é seguro! É fácil descobrir colisões! (em 2017)
  - SHA-2, aka SHA-256/SHA-512, SHA-3, etc.

Funções de Síntese



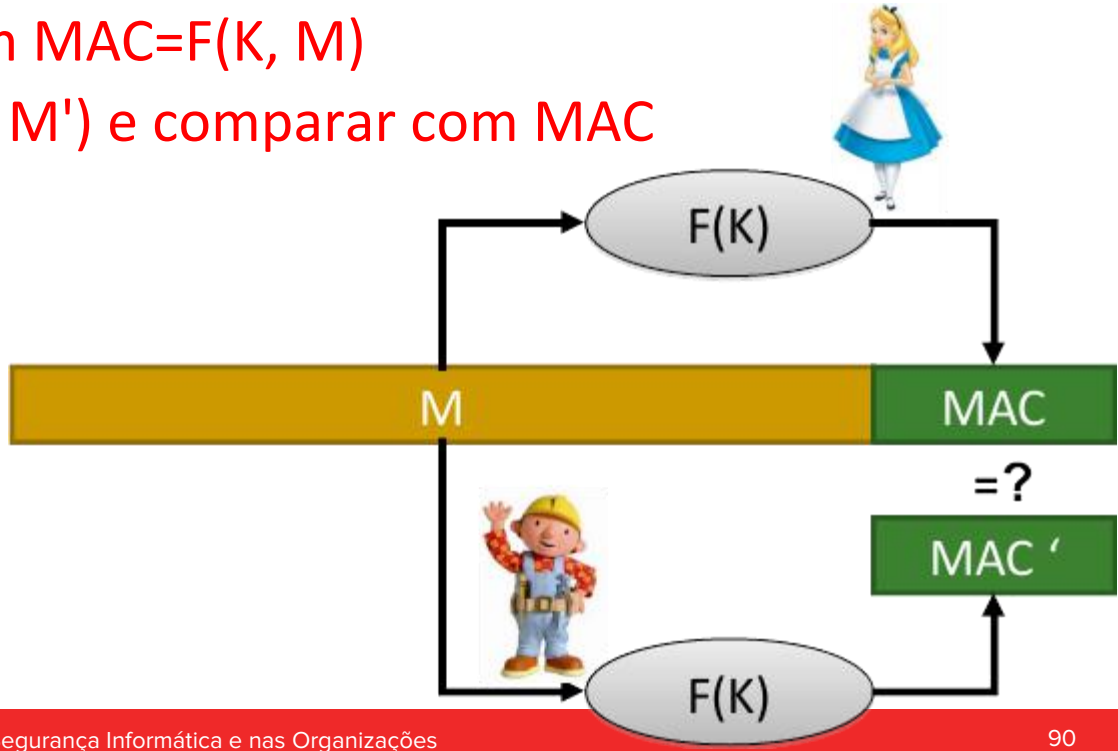
Message Integrity Code (MIC)

- **Fornecem capacidade de detetar alterações por máquinas**
  - Erros de comunicação/armazenamento
  - De caráter aleatório ou não controlado
- **Envio: Calcular S(T) e enviar T + MIC**
  - com T=texto e MIC=síntese(T)
- **Receção: Receber dados (T') e verificar se S(T') = MIC**
  - Calcular S'=síntese(T)
  - Validar se S(T') == MIC
- **Não protege contra alterações deliberadas**
  - Atacante pode manipular T em T' e calcular novo MIC

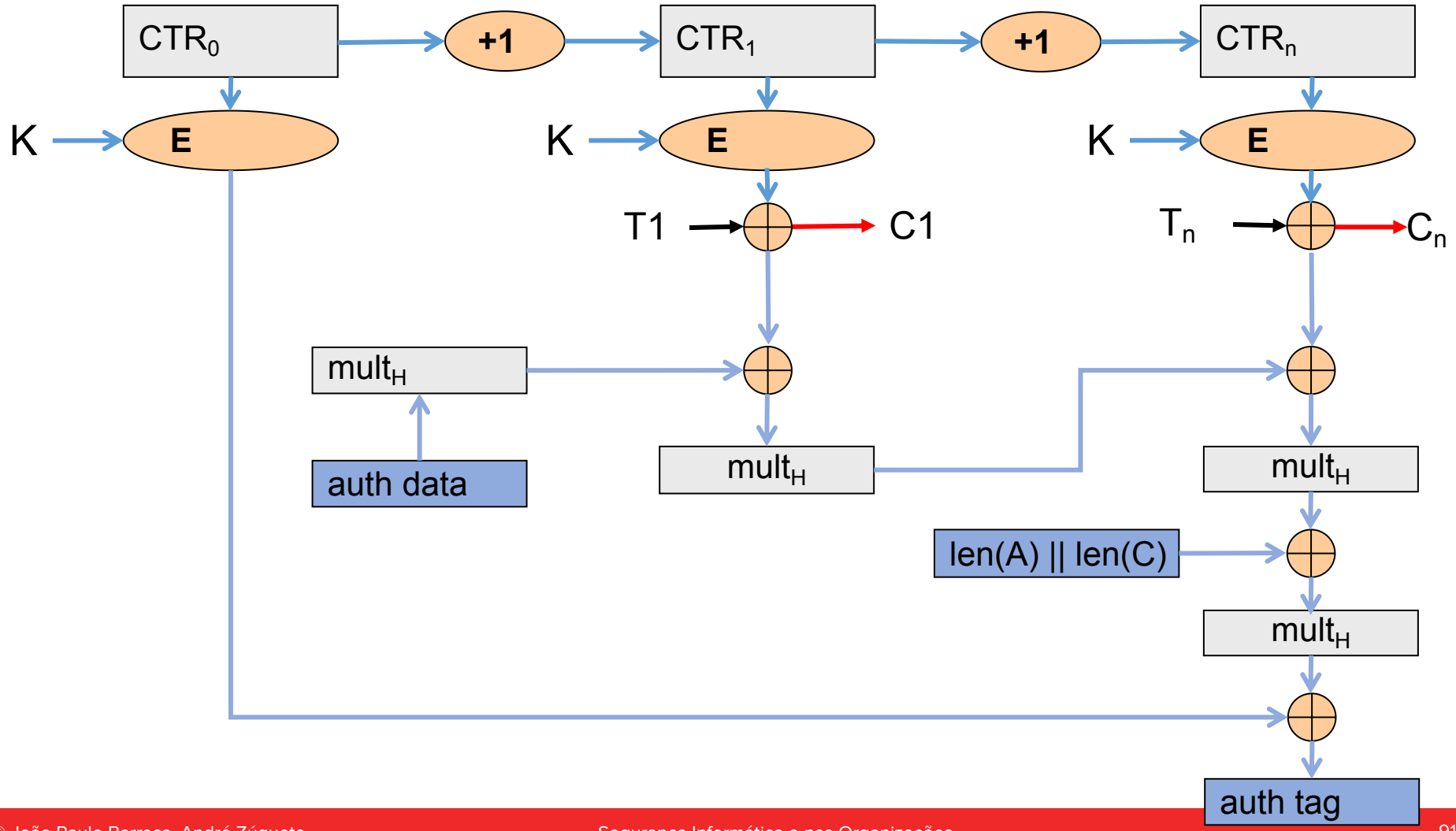
Message Authentication Code (MAC)

- **Síntese gerada com recurso a uma chave**
  - Só os conhecedores da chave conseguem gerar/validar o MAC

- **Utilizada para garantir autenticidade/integridade**
  - Enviar: M + MAC, com MAC=F(K, M)
  - Receber: Calcular F(K, M') e comparar com MAC



MAC: Cifras com Autenticação (CTR)



MAC: Aproximações

- **Cifrando uma síntese normal**
  - Por exemplo, com uma cifra simétrica por blocos
- **Usando uma função chaveada, realimentação e propagação de erros**
  - ANSI X9.9 (ou DES-MAC) com DES CBC (64 bits)
- **Usando uma chave nos parâmetros da função**
  - Keyed-MD5 (128 bits): MD5(K, keyfill, texto, K, MD5fill)
- **Construção HMAC: H(K, opad, H(K, ipad, texto))**
  - ipad = 0x36 B vezes, opad = 0x5C B vezes
  - HMAC-MD5, HMAC-SHA, etc.

Cifra e Autenticação

- **Encrypt-then-MAC: MAC calculado do criptograma**
  - Permite verificar a integridade antes da decifra

- **Encrypt-and-MAC: MAC é calculado do texto**
  - MAC não é cifrado
  - Fornece informação acerca do texto original (se igual a outro)

- **MAC-then-Encrypt: MAC é calculado do texto**
  - MAC é cifrado
  - Obriga a decifra completa antes da validação do MAC
    - Erros só são detetados após a decifra e validação

Assinaturas Digitais

- **Autenticam o conteúdo de documentos**
  - Garantem a sua integridade
- **Autenticam o autor**
  - Garantem a identidade do autor/criador
- **Previnem repudiação do conteúdo**
  - Autor não pode negar a sua criação
    - (só ele tem acesso à chave privada)

Assinaturas Digitais (aproximações)

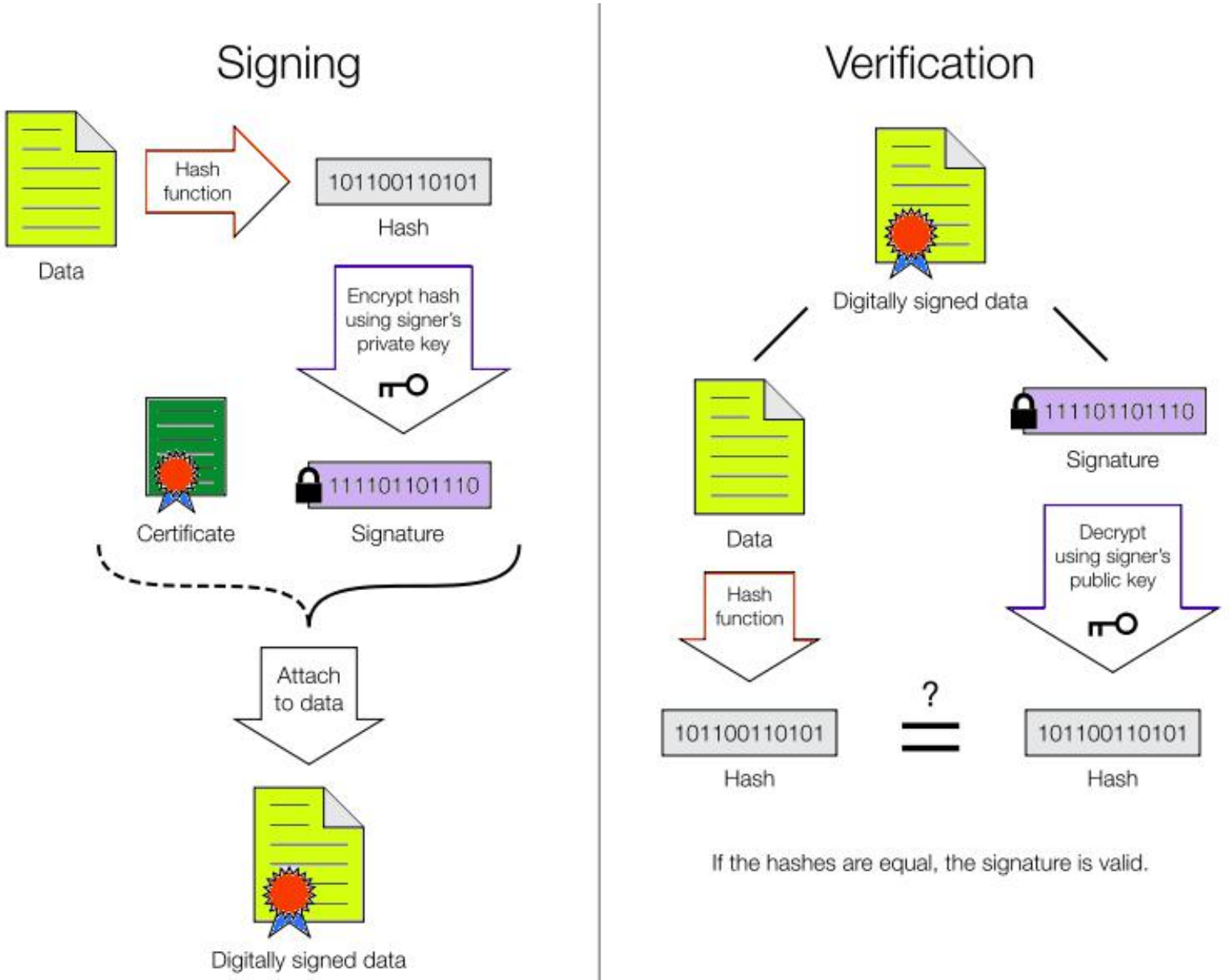
- **Cifra Assimétrica sobre Síntese**
  - Síntese usada por questões de desempenho
  - Cifra assimétrica para garantir autenticidade

Assinar: $A_x(\text{doc}) = \text{info} + E(K_x^{-1}, \text{digest}(\text{doc} + \text{info}))$

$\text{info} \rightarrow K_x$

Verificar:

$$D(K_x, A_x(\text{doc})) \equiv \text{digest}(\text{doc} + \text{info})$$


Signing
Data
Hash function
101100110101
Hash
Encrypt hash using signer's private key
Certificate
111101101110
Signature
Attach to data
Digitally signed data
Verification
Digitally signed data
Data
111101101110
Signature
Decrypt using signer's public key
Hash function
?
101100110101
Hash
101100110101
Hash
If the hashes are equal, the signature is valid.

Assinatura digital num email

```
From - Fri Oct 02 15:37:14 2009
[…]
Date: Fri, 02 Oct 2009 15:35:55 +0100
From: User From <user.from@ua.pt>
Organization: UA
MIME-Version: 1.0
To: User To <user.to@ua.pt>
Subject: Teste
Content-Type: multipart/signed; protocol="application/x-pkcs7-signature"; micalg=sha1; boundary="------------ms050405070101010502050101"

This is a cryptographically signed message in MIME format.

--------------ms050405070101010502050101
Content-Type: multipart/mixed;
 boundary="------------060802050708070409030504"

This is a multi-part message in MIME format.
--------------060802050708070409030504
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: quoted-printable

Corpo do mail

--------------060802050708070409030504–
--------------ms050405070101010502050101
Content-Type: application/x-pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"
Content-Description: S/MIME Cryptographic Signature

MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMIAGCSqGSIb3DQEHAQAAoIIamTCC
BUkwggSyoAMCAQICBAcnIaEwDQYJKoZIhvcNAQEFBQAwdTELMAkGA1UEBhMCVVMxGDAWBgNV
[…]
KoZIhvcNAQEBBQAEgYCofks852BV77NVuww53vSxO1XtI2JhC1CDlu+tcTPoMD1wq5dc5v40
Tgsaw0N8dqgVLk8aC/CdGMbRBu+J1LKrcVZa+khnjjtB66HhDRLrjmEGDNttrEjbqvpd2QO2
vxB3iPTlU+vCGXo47e6GyRydqTpbq0r49Zqmx+IJ6Z7iigAAAAAAAA==
--------------ms050405070101010502050101--
```

Assinaturas cegas

- **Assinaturas pode ser efetuadas de forma cega**
  - Assinante não consegue observar os conteúdos assinados
  - Semelhante a assinar um envelope com um documento e um papel químico
- **Servem para garantir o anonimato e a não alteração da informação assinada**
  - O assinante X sabe quem lhe pede a assinatura (Y)
  - X assina T_1, mas Y depois recupera a assinatura sobre T_2
    - T_2 não é qualquer, está relacionado com T_1
  - O requerente pode apresentar T_2 assinado por X
    - Mas não pode alterar T_2
    - X não consegue associar T_2 ao T_1 que viu e assinou

Derivação de Chaves

- **Algoritmos requerem chaves de dimensão fixa**
  - 56, 128, 256... bits
- **Necessário derivar chaves de várias fontes**
  - Segredos partilhados
  - Passwords geradas por humanos
  - Códigos PINs e segredos pequenos..
- **Fonte original pode ter baixa entropia**
  - reduz necessidade de um ataque de força bruta
  - Necessário existir uma transformação complexa entre fonte e chave
- **Necessário poder-se chegar a múltiplas chaves para a mesma password**
  - Evitar deduzir a password a partir da chave gerada

Derivação de Chaves

- **Reforço das chaves: Aumento da segurança de uma password**
  - Tipicamente definida por humanos
  - Tornar os ataques por dicionário impraticáveis

- **Expansão das chaves: Aumento da dimensão de uma password**
  - Expansão até ao pretendido para o algoritmo
  - Eventualmente também a geração de outros valores como chaves para MACs

Derivação de Chaves

- **Derivação de chaves impõe a existência de:**
  - um Sal que torna a geração única
  - um problema custoso
  - um grau de complexidade

- **Dificuldades computacionais:** Transformação requer recursos computacionais relevantes para ser realizada

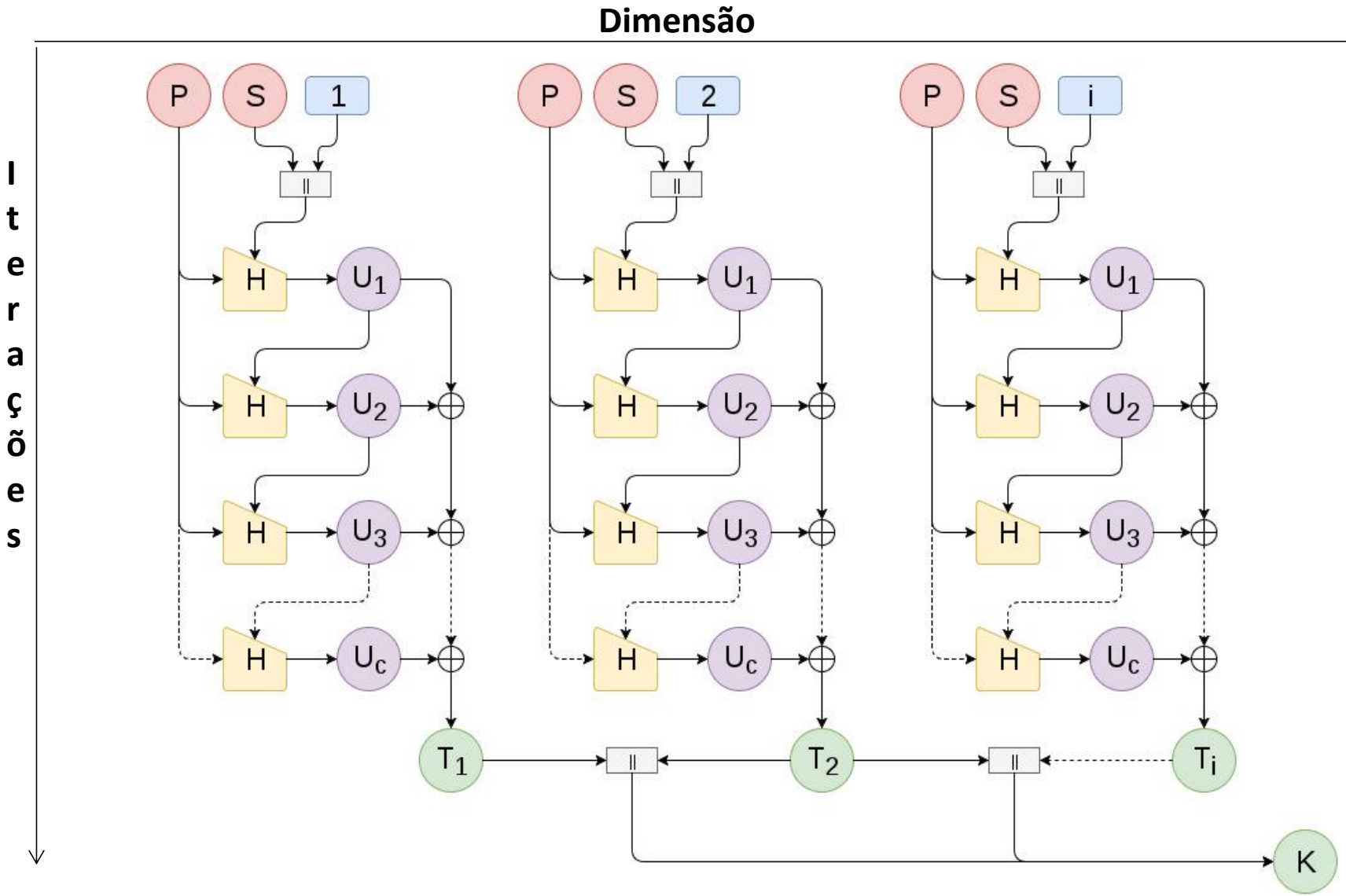
- **Dificuldades de armazenamento:** Transformação ocupa recursos de armazenamento relevantes (memória)

Derivação de Chaves: PBKDF2

Password Based Key Derivation Function 2

- **Produz uma chave com um custo computacional pré-definido**
- **K = PKBDF2(PRF, Sal, Iterações, Password, dim)**
  - PRF: Pseudo-Random-Function: Uma síntese
  - Sal: Um valor aleatório
  - Iterações: O custo (um valor nas centenas de milhares)
  - Password: Um segredo
  - Dim: a dimensão do resultado pretendido
- **Operação: Realiza N x dim operações do PRF, com base no SAL e password**
  - Quanto maior o valor de N, maior o custo

Derivação de Chaves: PBKDF2



Derivação de Chaves: scrypt

- **Produz uma chave com um custo de armazenamento pré-definido**

- **K = scrypt(Password, Sal, N, p, dim, r, hLen, MFlen)**
  - Password: um segredo a expandir
  - Sal: Um valor aleatório
  - N: parâmetro de custo
  - p: Parâmetro de paralelização. $p \leq (2^{32} - 1) * hLen / MFLen$
  - dim: a dimensão da chave a produzir
  - r: o tamanho dos blocos a usar (tipicamente 8)
  - hLen: dimensão da função de síntese (32 para SHA256)
  - MFlen: octetos na mistura interna (tipicamente 8 x r)

Derivação de Chaves: scrypt