UNIVERSIDADE FEDERAL DE SANTA CATARINA

ine
CTC • UFSC
Informática e Estatística

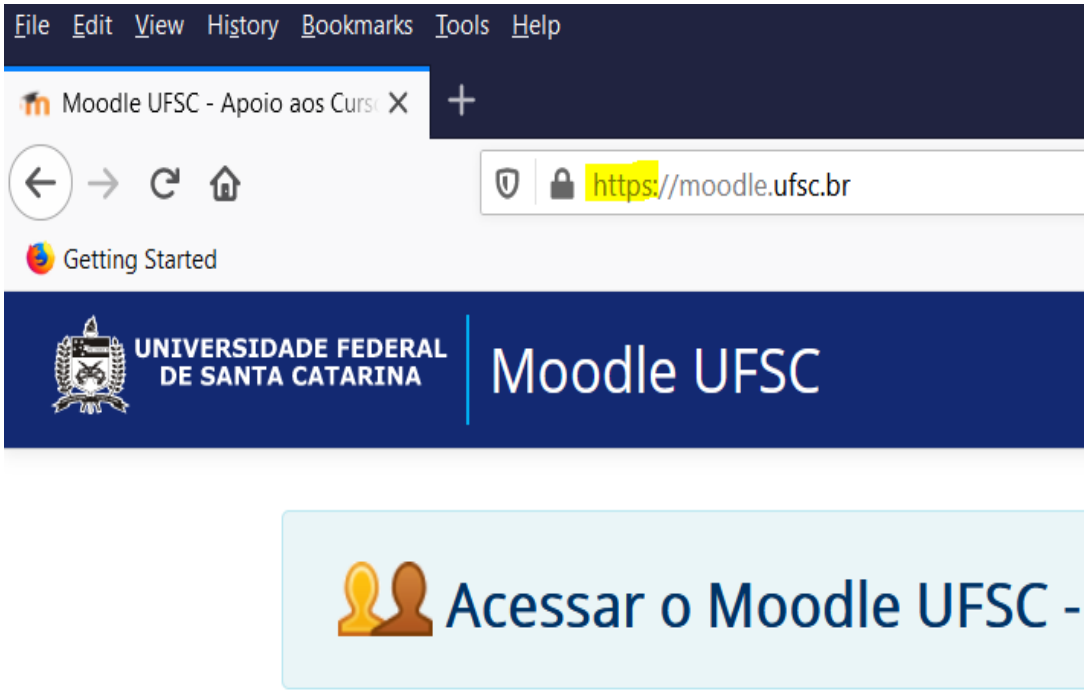
INE 5680
Segurança da Informação e de Redes

Criptografia Simétrica

Profa: Carla Merkle Westphall
carla.merkle.westphall@ufsc.br

Aplicações da Criptografia

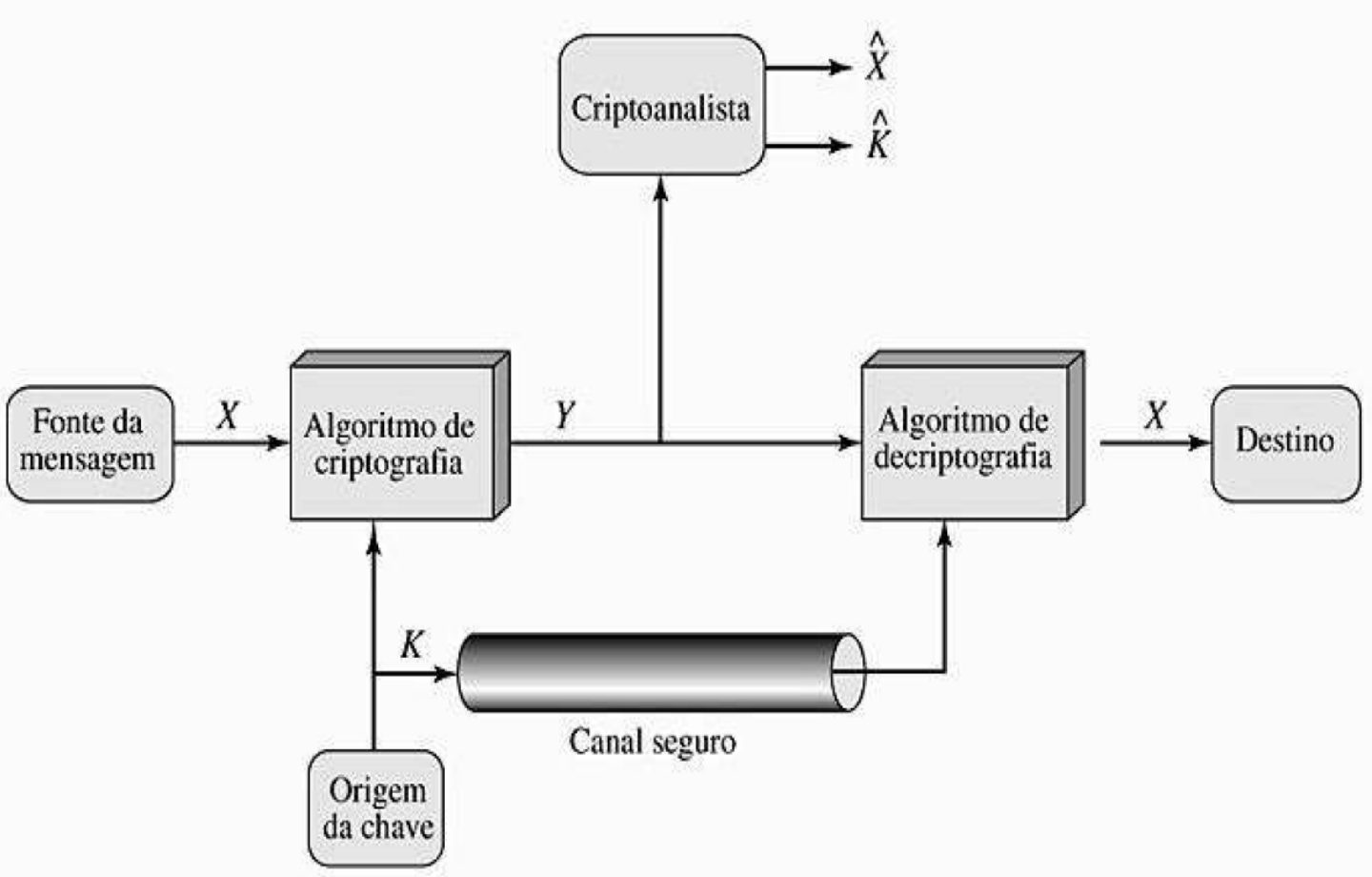
- **Comunicação segura (HTTPS, WhatsApp)**
  - 1 – definição de sessão para trocar chave
  - 2 – dados cifrados
- **Criptografia de arquivos**



Criptografia

- **Criptografia** – a ciência e arte da escrita (grafia) secreta (cripto) e do armazenamento secreto de informações
- Elementos:
  - Algoritmo (cifra) - são usados para transformar, a partir de uma chave, o texto em claro em um texto cifrado e vice-versa
  - Chaves - é um parâmetro do algoritmo que determina como seus dados são transformados
  - Comprimento das chaves
  - Texto em claro – mensagem no seu formato original
  - Texto cifrado – mensagem na forma cifrada

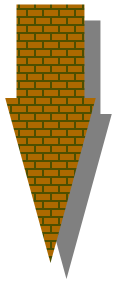
Criptografia Simétrica



Classificação

- **Criptologia é Criptografia + Criptoanálise**
- **Criptoanálise - estuda mecanismos para quebrar os textos cifrados (decifrar sem conhecer chave)**
- **Criptografia**
  - Simétrica
    - Cifras de fluxo (stream)
    - Cifras de bloco
  - Assimétrica
  - Protocolos (híbridos)

"ataque ao Iraque" - *texto em claro*



"buubdl bu njeojhiu" - *texto cifrado*

Princípios Gerais

- **Melhores criptosistemas:**
  - Chaves maiores
  - Chaves aleatórias
- **Bons criptosistemas produzem texto cifrado "aleatório"**
- **Melhores chaves são usadas apenas uma vez e são descartadas**
- **Princípio de *Kerckhoff*: um criptosistema deve ser seguro mesmo se o atacante souber vários detalhes sobre o sistema como os algoritmos de cifragem e decifragem. Somente a chave secreta o atacante não pode ter acesso.**

ATENÇÃO

Criptografia É:

- **Ferramenta poderosa**
- **Base de muitos mecanismos de segurança**

Criptografia NÃO É:

- **Solução para todos os problemas de segurança**
- **Confiável a menos que seja implementada e usada de forma correta**
- **Algo para tentar inventar por conta própria**
  - Muitos são os exemplos de projetos com falha

Criptografia - História

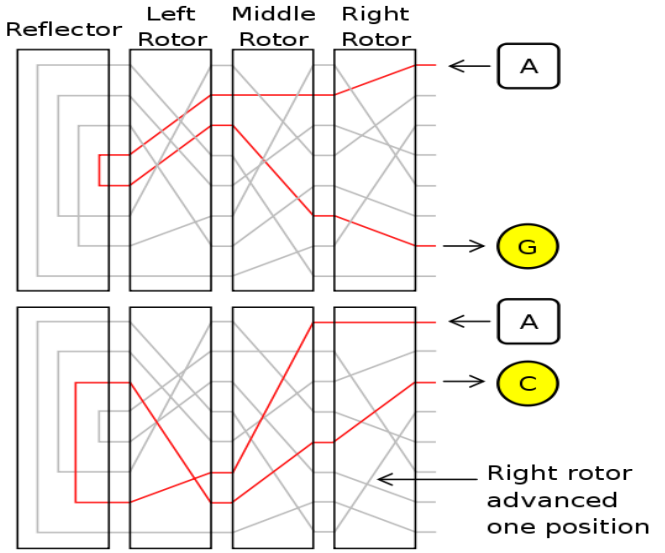
- Egípcios antigos cifravam alguns de seus hieróglifos
- O barro de Phaistos (1600 a.c) ainda não decifrado
- Cifrador de Júlio César, aproximadamente 60 AC
- Máquina Enigma – utilizada na Segunda Guerra Mundial pelos alemães para proteger as comunicações entre as embarcações e o comando

"A máquina Enigma não foi um sucesso comercial, porém foi aperfeiçoada até se transformar na ferramenta criptográfica mais importante da Alemanha durante a guerra. O sistema foi quebrado pelo matemático polonês Marian Rejewski que se baseou apenas em textos cifrados interceptados e numa lista de três meses de chaves diárias obtidas através de um espião".

(Fonte: http://www.numaboa.com.br/criptografia/historia/320-linha-do-tempo-atual?showall=&start=1)

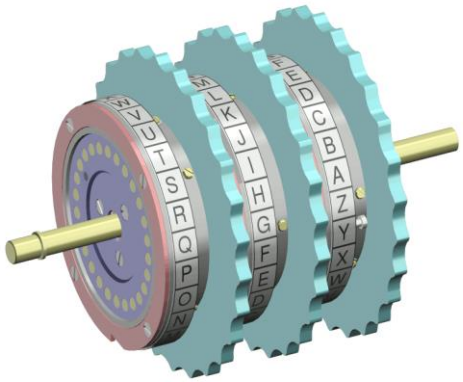
Linha de tempo da criptografia (Carl Ellison):
http://world.std.com/~cme/html/timeline.html

Máquina Enigma



Enigma: 3 a 5 rotores

chaves = 26^4 (2^{36} com o recurso de plugboard)



- Demonstração Máquina Enigma - Museu da UFRGS –

https://www.youtube.com/watch?v=VMJeDLv2suw

Pré-requisitos e Técnicas

Pré-requisitos:

- Teoria de Números
- Matemática Discreta
- Teoria da Informação
- Teoria de Probabilidade
- Complexidade Computacional
- Processamento de Sinais

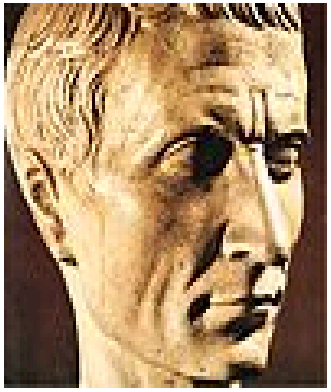
Técnicas de Cifragem Básicas:

- Substituição (Letras do texto plano são substituídas por outras letras)
- Permutação (ou transposição) – (Letras do texto plano são permutadas entre si)
- Combinações ou iterações das duas técnicas acima

Técnicas de Substituição - Cifrador de César

m:texto plano abcdefghijklmnopqrstuvwxyz

c:texto cifrado DEFGHIJKLMNOPQRSTUVWXYZABC



meditarproduzsabedoria

PHGMWDVTVRGXCVDEHGRULD

cifrar

c = E(m) = (m+3) mod 26

decifrar

m = D(c) = (c-3) mod 26

Técnica quebrada!

Executar o teste no browser (Ciphers):
http://www.cryptool-online.org/

Técnicas de Substituição - Cifrador 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

k: proverbio

m: meditar

c: bvrdxrs

Repetir:

1. Na linha da letra p (primeira letra da chave k)
2. Localizar coluna da letra m (primeira letra do texto plano m)
3. Encontrar a letra B (primeira letra do texto cifrado)

Técnica quebrada!

Quebra de cifras de substituição

- **Freqüência de letras em inglês: e (12.7%), t (9.1%), a (8.1%)**

UKBYBIPOUZBCUFEEBORUKBYBHOBBRFESPVKBWFOFERVNBCVBZPRUBOFERVNBCVBPCYYFVUFO
FEIKNWFRFIKJNUPWRFIPOUNVNIPUBRNCUKBEFWWFDNCHXCYBOHOPYXPUBNCUBOYNRVNIWN
CPOJIOFHOPZRVFZIXUBORJRUBZRBCHNCBBONCHRJZSFWNVRJRUBZRPCYZPUKBZPUNVPWPCYVF
ZIXUPUNFCPWRVNBCVBRPYYNUNFCPWWJUKBYBIPOUZBCUIPOUNVNIPUBRNCHOPYXPUBNCUB
OYNRVNIWNCPOJIOFHOPZRNCRVNBCUNENVVFZIXUNCHPCYVFZIXUPUNFCPWZPUKBZPUNVR

B	36
N	34
U	33
P	32
C	26

➔ E

➔ T

➔ A

NC	11
PU	10
UB	10
UN	9

digrams

➔ IN

➔ AT

UKB	6
RVN	6
FZI	4

trigrams

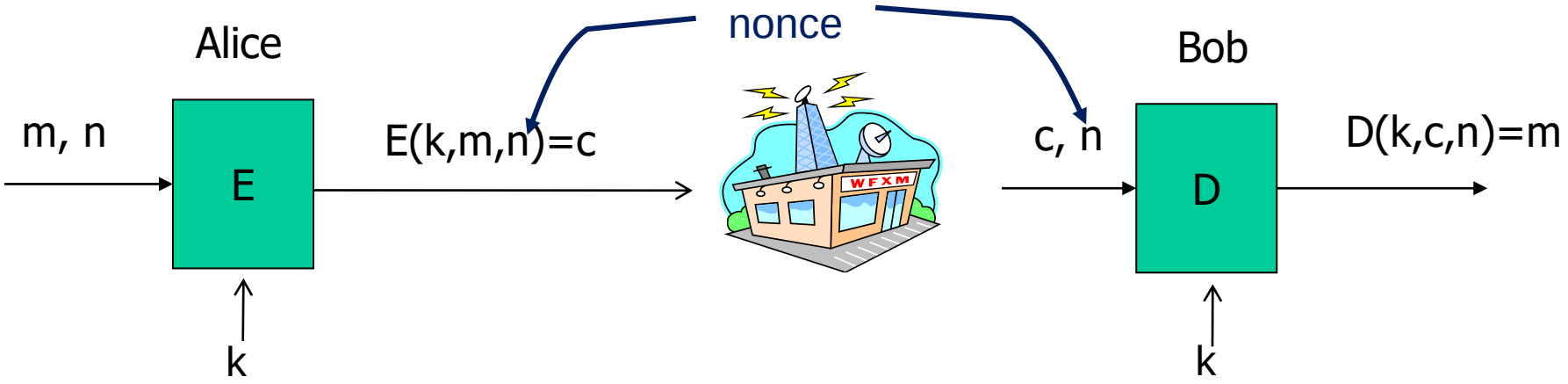
➔ THE

Criptografia Simétrica

- **Nomes alternativos: criptografia de chave simétrica, chave única, chave SECRETA**
- **Assume que as partes JÁ COMPARTILHAM a MESMA chave secreta**
- **Um cifra definida sobre os conjuntos (K, M, C)**
  - K é o conjunto de todas as chaves possíveis (*keyspace*)
  - M é o conjunto de todas as mensagens possíveis (*message space*)
  - C é o conjunto de todos os textos cifrados possíveis (*ciphertext space*)
- **é um par de algoritmos (*E*, *D*) "eficientes" (executam em tempo polinomial) onde**

  E: K x M → C , D: K x C → M
- **E: algoritmo de cifragem, D: o algoritmo de decifragem**
- **∀ m∈M, k∈K: D(k, E(k, m)) = m**
- **E é frequentemente "randomizado". D é determinístico (com a chave e o texto cifrado, o resultado é sempre igual) .**

Criptografia Simétrica



E, D: cifra **k: chave secreta (por ex. 128 bits)**

m: texto plano, c: texto cifrado, **n: nonce** (também conhecido por IV)

- **Nonce: um valor que muda de mensagem para mensagem**

  o par (k,n) nunca é usado mais de uma vez

- **Método 1: nonce é um contador (por exemplo, contador de pacotes)**
  - Usado quando a parte que cifra guarda o estado de msg para msg
  - Se o decifrador tem o mesmo estado, não é necessário enviar o nonce com o texto cifrado
- **Método 2: cifrador escolhe um nonce aleatório, $n \leftarrow \mathbb{N}$**

Algoritmo criptográfico é publicamente conhecido

- Nunca use uma cifra proprietária

Casos de uso – chave de uso único e uso múltiplo

Chave de uso único: (one time key)

- Chave é usada para cifrar uma mensagem
  - email cifrado: nova chave gerada para cada email
- Não precisa do nonce (definido com valor 0)

Chave de uso múltiplo: (many time key)

- Chave usada para cifrar múltiplas mensagens
  - SSL: mesma chave usada para cifrar muitos pacotes
- Precisa de um nonce único ou de um nonce aleatório

Exemplos

Chave	Nonce	Tamanho (chave + nonce)
000	00	8 chaves * 4 nonces = 32 possibilidades (chave + nonce) diferentes (2^5)
	01	
	10	
	11	
001	00	
	01	
	10	
	11	
010		
011		
100		
101		
110		
111		

Tempo médio para busca exaustiva da chave

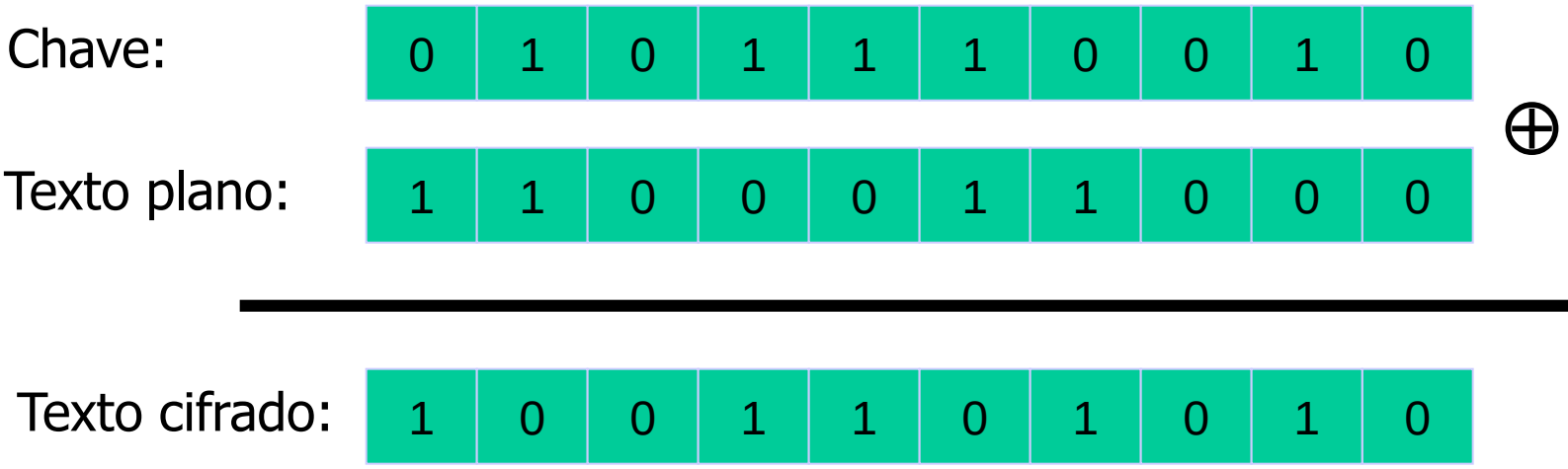
Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^{9} years	6.3×10^{6} years

Tipos de ataques a mensagens cifradas

Tipo de ataque	Conhecido ao criptoanalista	
Apenas texto cifrado	• Algoritmo de criptografia • Texto cifrado	*Termo em inglês:* *CCO - Ciphertext-only attack*
Texto claro conhecido	• Algoritmo de criptografia • Texto cifrado • Um ou mais pares de texto claro/texto cifrado formados com a chave secreta	*Termo em inglês:* *KPA- known-plaintext attacks*
Texto claro escolhido	• Algoritmo de criptografia • Texto cifrado • Mensagem de texto claro escolhida pelo criptoanalista, juntamente com seu texto cifrado correspondente, gerado com a chave secreta	*Termo em inglês:* *CPA - chosen-plaintext attacks*
Texto cifrado escolhido	• Algoritmo de criptografia • Texto cifrado • Texto cifrado pretendido, escolhido pelo criptoanalista, juntamente com seu texto claro decriptografado correspondente, gerado com a chave secreta	*Termo em inglês:* *CCA - chosen-ciphertext attacks*
Texto escolhido	• Algoritmo de criptografia • Texto cifrado • Mensagem de texto claro escolhida pelo criptoanalista, juntamente com seu texto cifrado correspondente, gerado com a chave secreta • Texto cifrado pretendido, escolhido pelo criptoanalista, juntamente com seu texto claro decriptografado correspondente, gerado com a chave secreta	

One Time Pad (chave de uso único)

- **Vernam (1917)**



$c = E(k, m) = k \oplus m$ **$D(k, c) = k \oplus c$** **$M=C=K=\{0,1\}^n$**

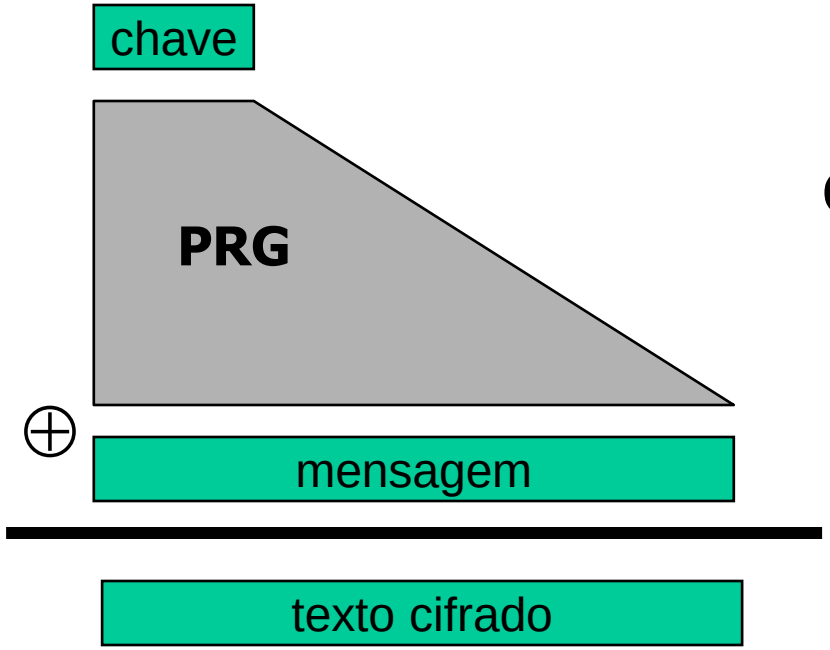
- Sabendo m e c, é possível calcular a chave?

- **Shannon '49:**
  - Texto cifrado não deve revelar NADA sobre o texto plano
  - One Time Pad é "seguro" contra ataques de texto cifrado (*ciphertext-only attacks*)

Cifras de Fluxo (Stream ciphers) (chave de uso único)

Problema: chave OTP tem tamanho igual ao da mensagem

<u>Solução</u>: chave Pseudo aleatória -- cifras de fluxo



$$c \leftarrow PRG(k) \oplus m$$

Stream ciphers: RC4 (113MB/sec) , SEAL (293MB/sec)

PRG: Pseudorandom Generator

Alerta no uso de cifras de fluxo

Chave de uso único!!

"Two time pad" (usar a mesma chave duas vezes) é inseguro:

$$\begin{cases} C_1 \leftarrow m_1 \oplus PRG(k) \\ C_2 \leftarrow m_2 \oplus PRG(k) \end{cases}$$

Agente malicioso (*eavesdropper*) faz:

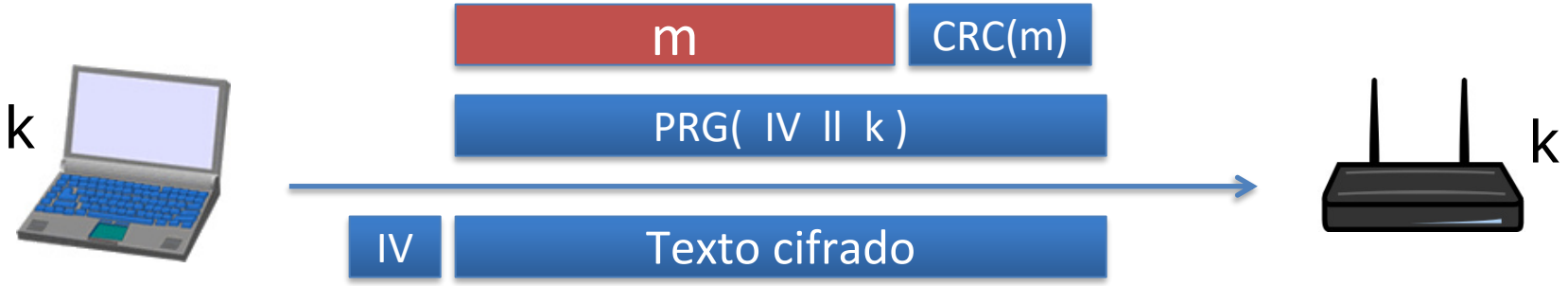
$$C_1 \oplus C_2 \rightarrow m_1 \oplus m_2$$

Usando análise de redundância de uma língua pode descobrir mensagens:

$$m_1 \oplus m_2 \rightarrow m_1 \text{ , } m_2$$

Exemplos reais

802.11b WEP:



Tamanho do IV (*Initialization Vector*): 24 bits

- IV repete depois de $2^{24} \approx 16M$ frames (depois de 50000 pacotes)
- Em algumas placas 802.11: IV resetado para 0 depois de desligado/ligado

key for frame #1: (1 ll k)
key for frame #2: (2 ll k)
…

24 bits 104 bits

60.000 pacotes - 80% de sucesso
40.000 pacotes capturados em menos de 1 min
Cálculo em 3 seg, 3 MiB, Pentium-M 1.7 GHz

- **WPA2 foi quebrado (security update – https://www.wi-fi.org/security-update-october-2017)**
- **WPA3 está disponível: https://www.wi-fi.org/discover-wi-fi/security**

Quebra do WPA2 (2017)

- https://www.krackattacks.com/
- Mathy Vanhoef e Frank Piessens publicaram e apresentaram artigo no congresso ACM CCS 2017
- O protocolo WPA2 usa um handshake de 4-vias para gerar uma chave de sessão
- Os autores demonstram o ataque da reinstalação da chave
- Fizeram testes no Android 6.0: força o cliente a usar uma chave previsível “tudo zero”. Assim, conseguem ter acesso ao tráfego decifrado

Exemplos reais

- **Projeto Venona: empenhava-se, desde 1943, na interceptação de radiocomunicações entre as redes de espionagem americana e a "central" em Moscou. Russos usaram algumas chaves “mais de uma vez” e assim uma porcentagem pequena das mensagens interceptadas foi decifrada inteiramente ou parcialmente (1%)**

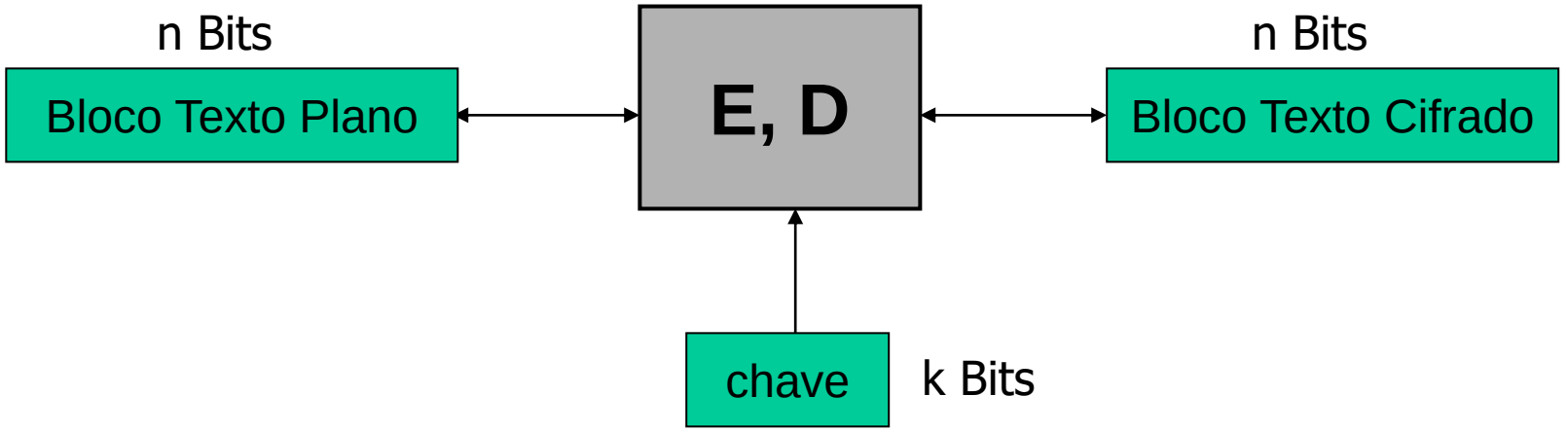
Alerta sobre PRG fracos

- Geradores de números aleatórios ou pseudo-aleatórios são cruciais para o bom funcionamento de algoritmos criptográficos
- Vários são os ataques e problemas
  - http://en.wikipedia.org/wiki/Random_number_generator_attack
  - https://www.owasp.org/index.php/Insecure_Randomness
- Por exemplo, nunca usar *random* da glibc (Kerberos V4)

```
glibc random():
      r[i] ← ( r[i-3] + r[i-31] )   % 2^32
      output  r[i] >> 1
```

- Problemas no /dev/random e /dev/urandom do Linux (https://eprint.iacr.org/2013/338)

Cifras de bloco



Exemplos:

1. 3DES: n= 64 bits, k = 168 bits
2. AES: n=128 bits, k = 128, 192, 256 bits

IV manipulado como parte do bloco do texto plano

Construindo uma cifra de bloco

Entrada: (m, k)

Repetir várias vezes a simples operação de "mistura"

- **DES:** **Repetir 16 vezes:**

$$\begin{cases} m_L \leftarrow m_R \\ m_R \leftarrow m_L \oplus F(k, m_R) \end{cases}$$

- **AES-128:** **Passo de mistura repetido 10 vezes**

Difíceis de projetar: devem resistir a ataques sutis

- **ataques diferenciais e lineares: com muitos pares de entrada/saída, tentam recuperar a chave**

- **força bruta – no DES, tempo total de ataque $\approx 2^{43}$ com 2^{42} pares aleatórios de entrada/saída**

Implementações

- **Em Javascript: crypto-js, http://crypto.stanford.edu/sjcl/**
- **Para usar em Java:**
  1. JCA dentro do JDK inclui dois componentes de software:
     - Framework que inclui pacotes java.security, javax.crypto, javax.crypto.spec, javax.crypto.interfaces
     - ***Providers* reais como Sun, SunRsaSign, SunJCE, que contêm as implementações criptográficas reais**
  2. http://www.gnu.org/software/gnu-crypto/
  3. http://bouncycastle.org/latest_releases.html
- **Existem instruções em hardware para acelerar o AES**
  - Intel Westmere: **aesenc, aesenclast (fazem um** round AES). **aeskeygenassist**: executa expansão de chaves AES
  - Instruções similares no AMD Bulldozer
- **Para usar em C++: Crypto++ : http://www.cryptopp.com/**

Comentários – algoritmos simétricos

Cifra	Autor	Comprimento da chave	Comentários
DES	IBM	56 bits	Muito fraco para usar agora
RC4	Ronald Rivest	1 a 2.048 bits	Atenção: algumas chaves são fracas
RC5	Ronald Rivest	128 a 256 bits	Bom, mas patenteado
AES (Rijndael)	Daemen e Rijmen	128 a 256 bits	Melhor escolha
Serpent	Anderson, Biham, Knudsen	128 a 256 bits	Muito forte
DES triplo	IBM	168 bits	Bom, mas está ficando ultrapassado
Twofish	Bruce Schneier	128 a 256 bits	Muito forte; amplamente utilizado

Fonte: Tanenbaum, Redes de Computadores, capítulo 8

Cifras de Fluxo e de Bloco

- **Algorithms, key size and parameters report – 2014 (link no moodle)**

Primitive	Classification	
	Legacy	Future
HC-128	✓	✓
Salsa20/20	✓	✓
ChaCha	✓	✓
SNOW 2.0	✓	✓
SNOW 3G	✓	✓
SOSEMANUK	✓	✓
Grain	✓	✗
Mickey 2.0	✓	✗
Trivium	✓	✗
Rabbit	✓	✗
A5/1	✗	✗
A5/2	✗	✗
E0	✗	✗
RC4	✗	✗

Primitive	Classification	
	Legacy	Future
AES	✓	✓
Camellia	✓	✓
Three-Key-3DES	✓	✗
Two-Key-3DES	✓	✗
Kasumi	✓	✗
Blowfish$^{\geq 80}$-bit keys	✓	✗
DES	✗	✗

Classification	Meaning
Legacy ✗	Attack exists or security considered not sufficient. Mechanism should be replaced in fielded products as a matter of urgency.
Legacy ✓	No known weaknesses at present. Better alternatives exist. Lack of security proof or limited key size.
Future ✓	Mechanism is well studied (often with security proof). Expected to remain secure in 10-50 year lifetime.

Performance: Crypto++ 5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux) https://www.cryptopp.com/benchmarks.html

	Cifra	tam bloco/chave	Velocidade (MiB/sec)
fluxo	**RC4**		**126**
	Salsa20/12		643
	Sosemanuk		**727**
bloco	**3DES**	**64/168**	**13**
	AES-128	**128/128**	**109**

Outras medidas de desempenho

- https://bearssl.org/speed.html
- https://www.cryptopp.com/
- https://www.cryptopp.com/wiki/ChaCha20Poly1305

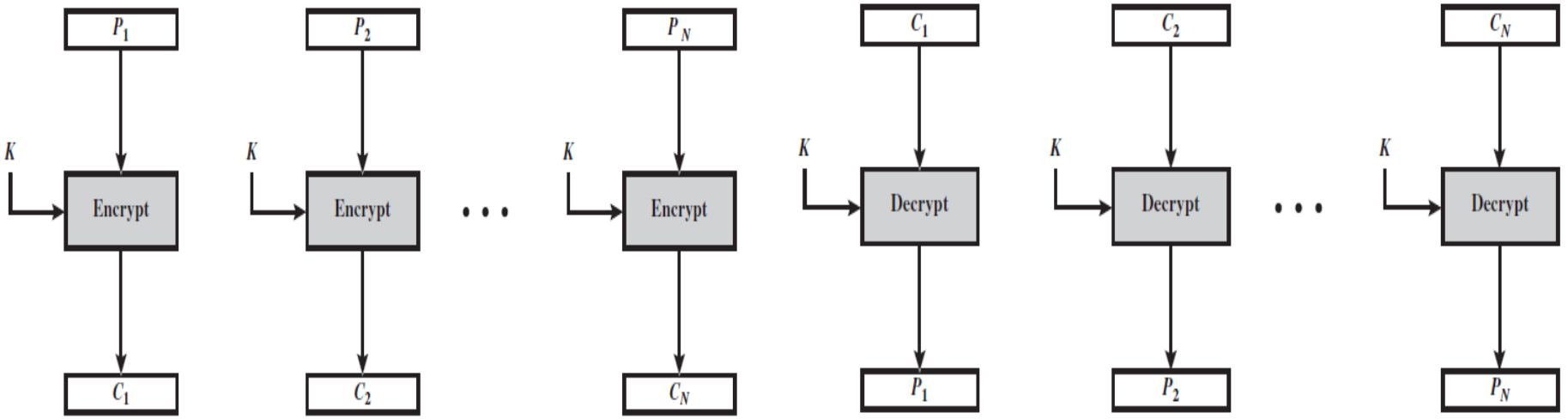
Cifras de Bloco

- **Uma cifra de bloco é muito mais do que apenas um algoritmo de criptografia, pode ser usado para :**
  - Construir diferentes tipos de esquemas de cifragem baseados em bloco
  - Implementar cifras de fluxo
  - Construir funções hash
  - Fazer MAC (message authentication codes)
  - Construir protocolos de estabelecimento de chaves
  - …
- **Modos de operação: maneiras diferentes de organizar a cifragem de longos textos planos, por exemplo email ou arquivo, com cifras de bloco**
- **Existem modos seguros e inseguros!**

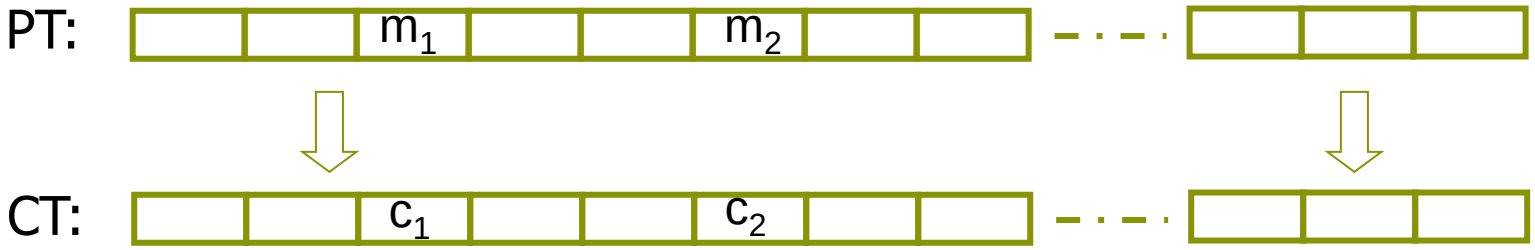
Modos de operação

- Electronic Code Book mode (ECB)
- Cipher Block Chaining mode (CBC)
- Output Feedback mode (OFB)
- Cipher Feedback mode (CFB)
- Counter mode (CTR)
- Galois Counter Mode (GCM)

Modo 1 - Electronic Code Book (ECB)



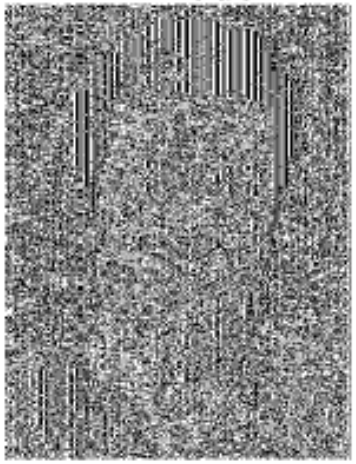
Não usar este modo! Problema: Se m_1=m_2 então c_1=c_2

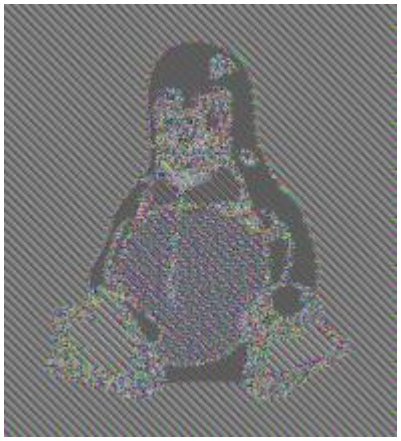


Problema visto em figuras

An example plaintext

Encrypted with AES in ECB mode



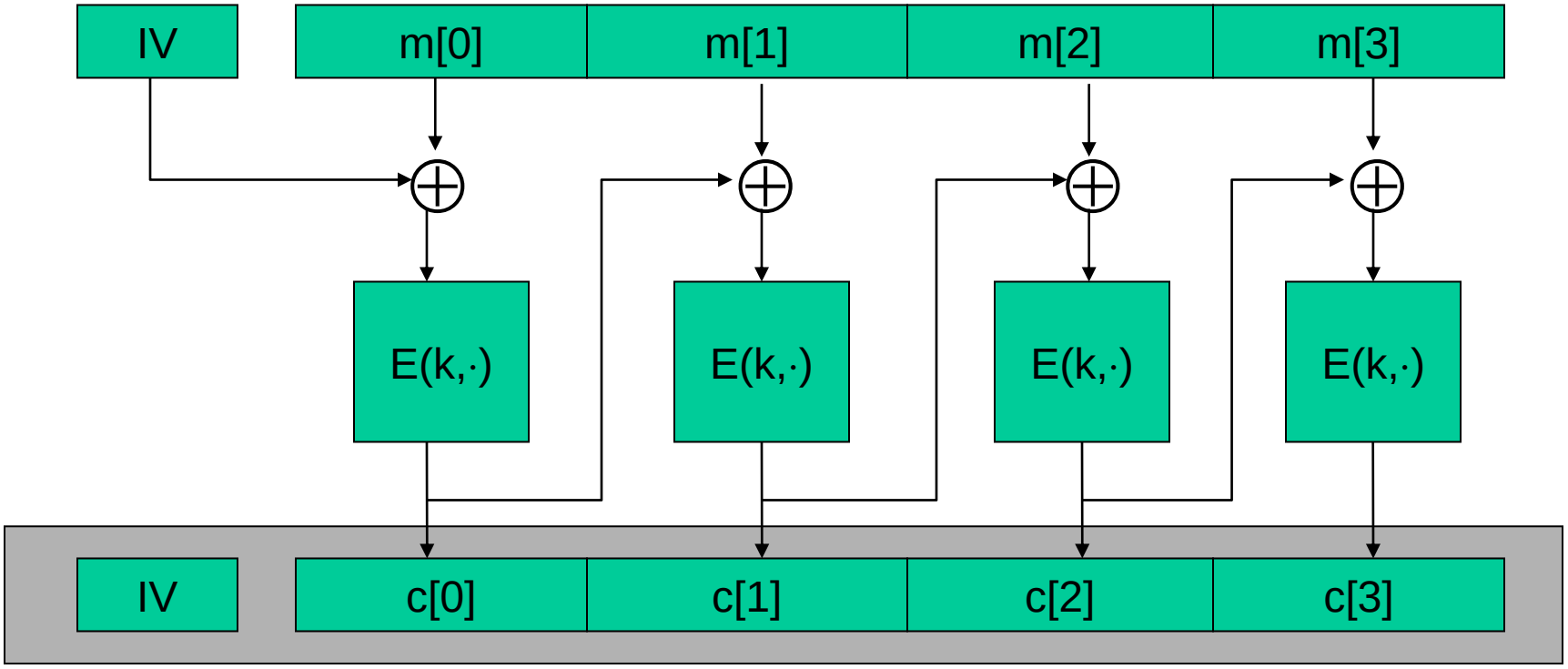


Comentários ECB

- Não deve ser usado para cifrar mensagens com tamanho maior que um bloco! Apenas mensagens de bloco único
- Desvantagem: Se $m_1=m_2$ então $c_1=c_2$ -> um texto cifrado de vários blocos pode revelar informações estatísticas sobre o texto plano, facilitando ataques
- Desvantagem: não protege a sequência de blocos cifrados. Um adversário pode modificar uma msg longa removendo ou reordenando blocos. Se o adversário tiver blocos cifrados com a mesma chave, pode inserir esses blocos no texto cifrado

Modo 2 - Cipher Block Chaining (CBC)

E e D são PRP (*Pseudo random permutation*) como 3DES ou AES.
Cipher Block Chaining com IV (*Initialization Vector*) aleatório:

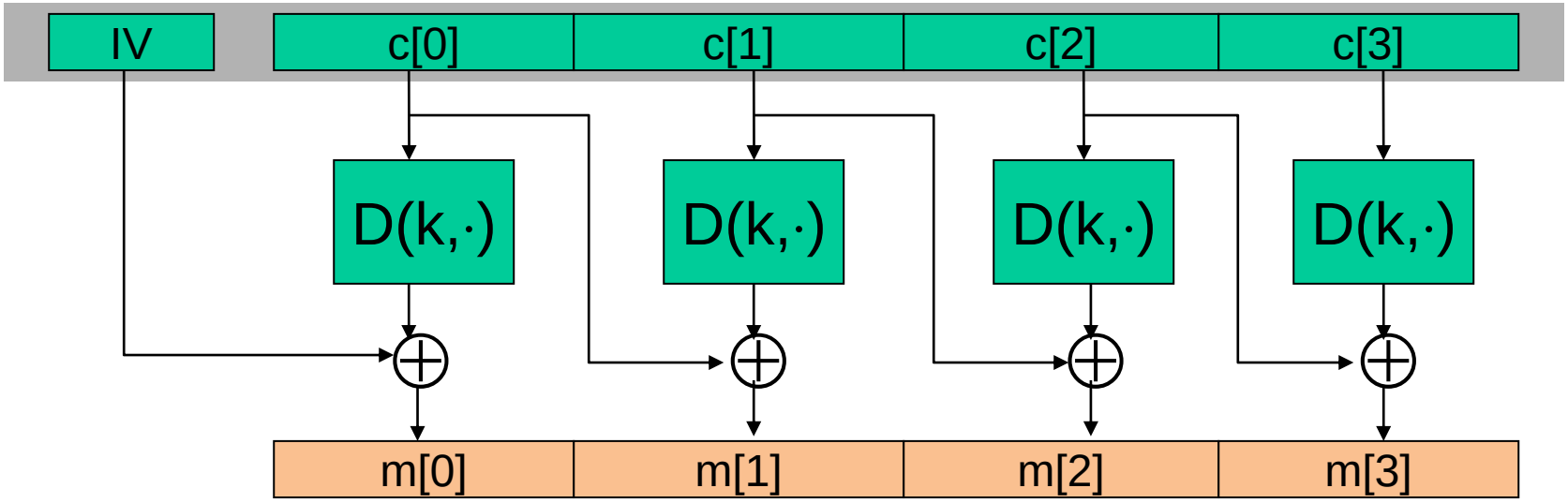


texto cifrado

Q: como decifrar?

Cipher Block Chaining (CBC) - decifragem

Em símbolos: $c[0] = E(k, IV \oplus m[0]) \Rightarrow m[0] = D(k, c[0]) \oplus IV$

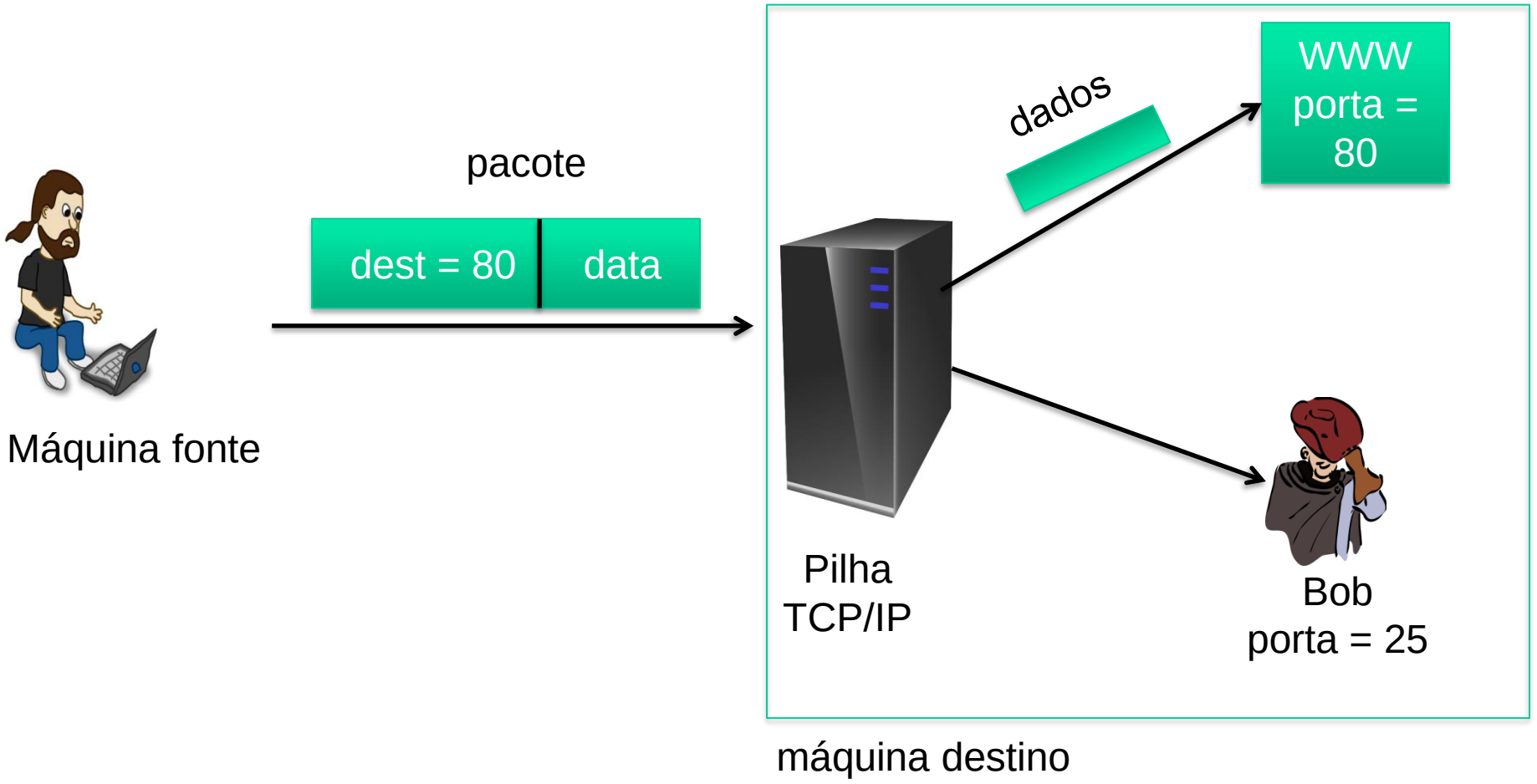


Para garantir a segurança, há LIMITE de mensagens que podem ser cifradas com a mesma chave (por exemplo, para um erro < $1/2^{32}$)

- **AES 128 bits de bloco: depois de 2^{48} blocos AES, DEVE ser trocada a chave**
- **3DES 64 bits de bloco: depois de 2^{16} blocos 3DES, DEVE ser trocada a chave**

Ataque ativo de adulteração (*tampering*)

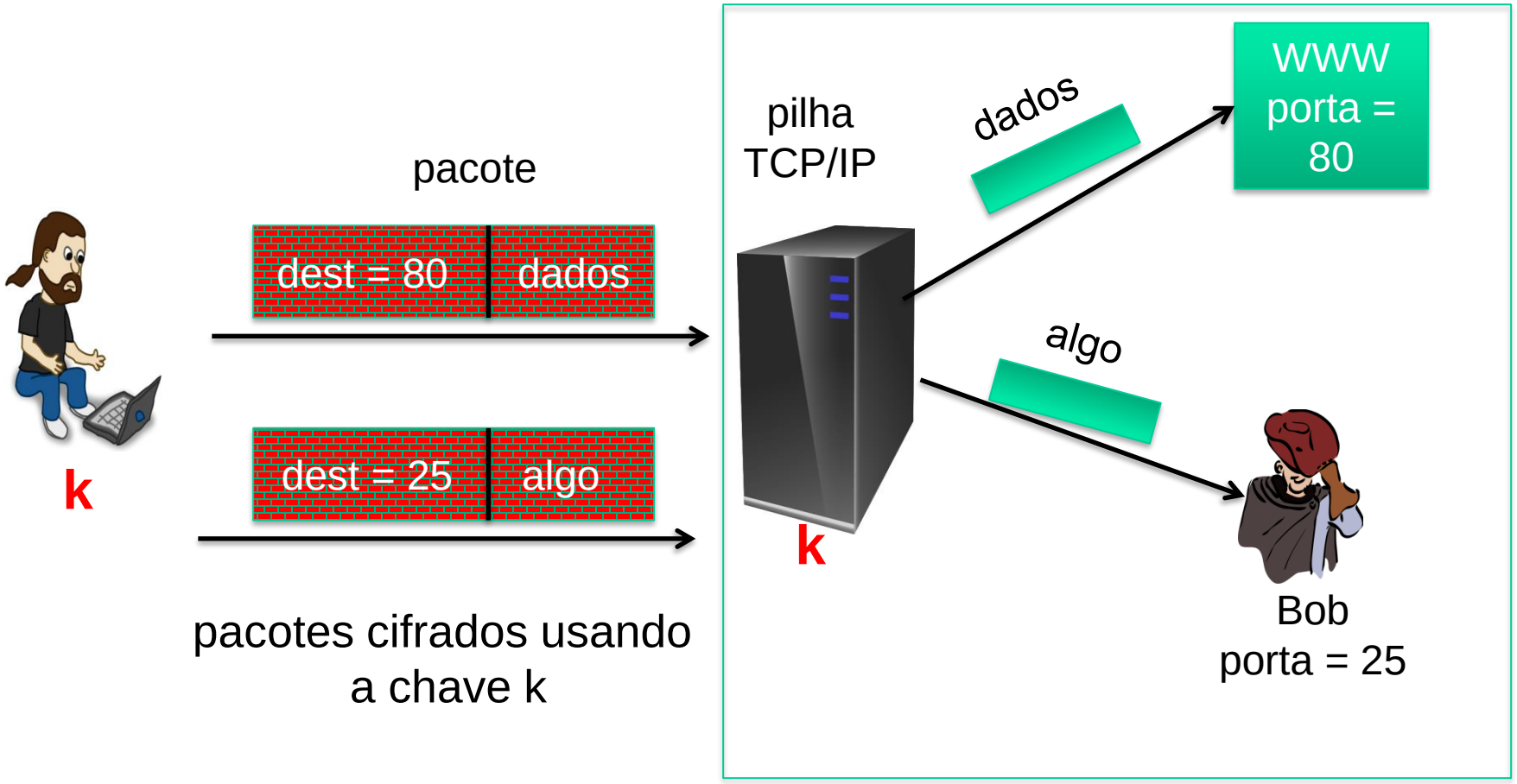
TCP/IP



Dan Boneh

Ataque ativo de adulteração (*tampering*)

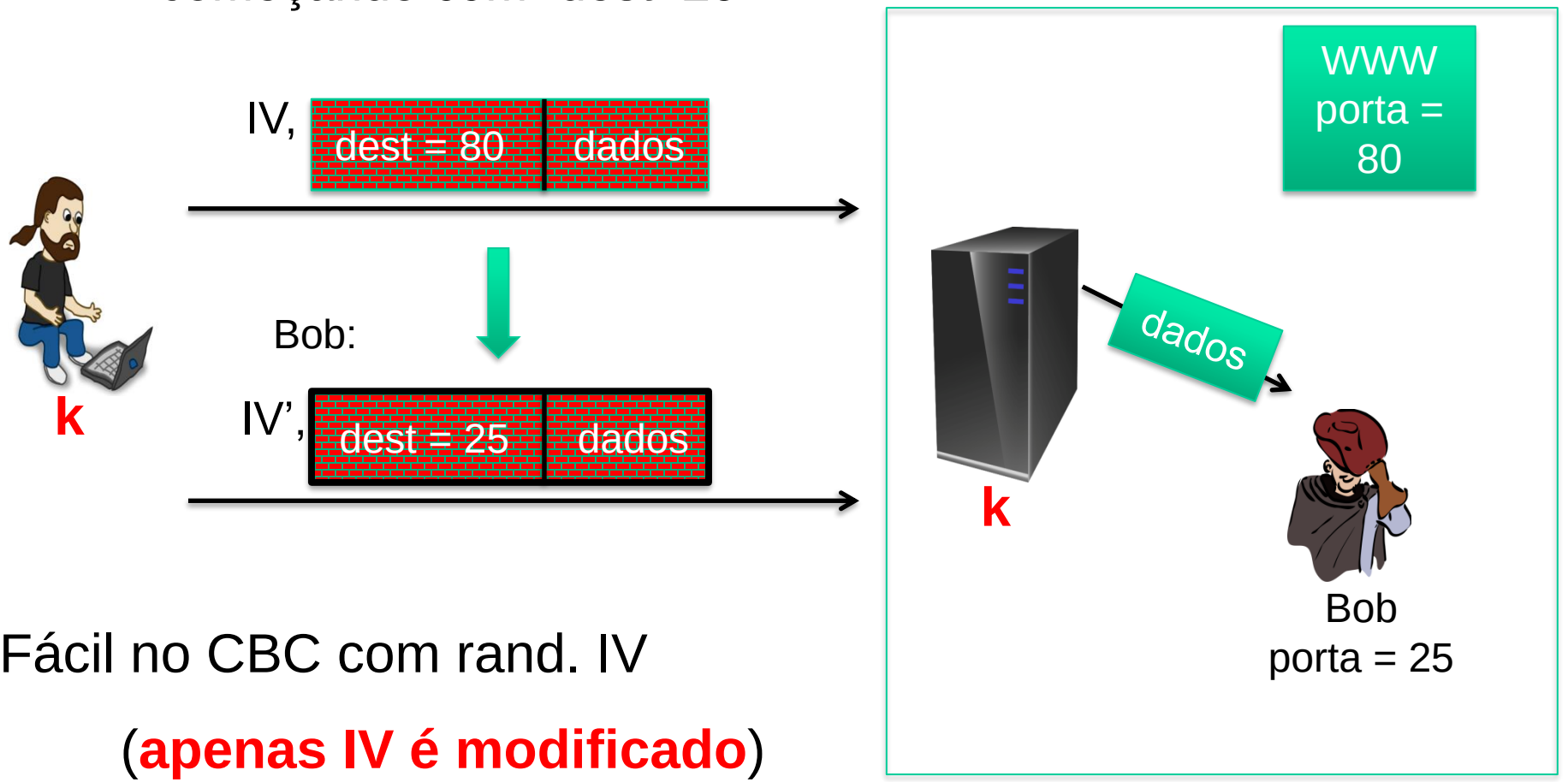
IPsec:



Dan Boneh

Lendo dados de alguém

Nota: atacante obtém decifragem de qualquer dado começando com “dest=25”

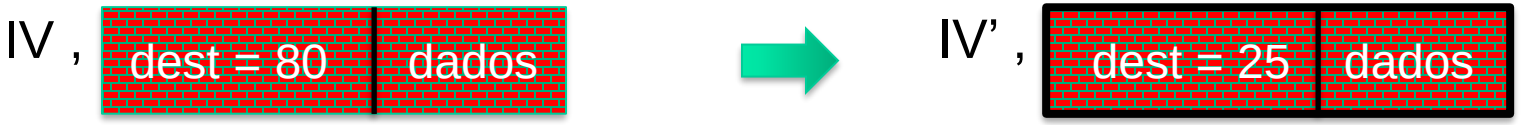


Fácil no CBC com rand. IV

(**apenas IV é modificado**)

Dan Boneh

Lendo dados de alguém - COMO



Cifragem é feita no CBC com um IV aleatório.

Como deve ser IV'? m[0] = D(k, c[0]) ⊕ IV = "dest=80…"

IV' = IV ⊕ (…25…)

IV' = IV ⊕ (…80…)

IV' = IV ⊕ (…80…) ⊕ (…25…)

It can't be done

D(k, c[0]) ⊕ IV' = D(k, c[0]) ⊕ IV ⊕ 80 ⊕ 25

= …25…

Dan Boneh

Casos de uso: como escolher IV

Chave uso único: IV não necessário (IV=0)

- email cifrado, nova chave para cada mensagem

Chave uso múltiplo: segurança contra ataques de texto plano

- Arquivos: mesma chave AES usada para cifrar muitos arquivos
- IPsec: mesma chave AES usada para cifrar muitos pacotes

Melhor: usar um IV *aleatório* para cada msg

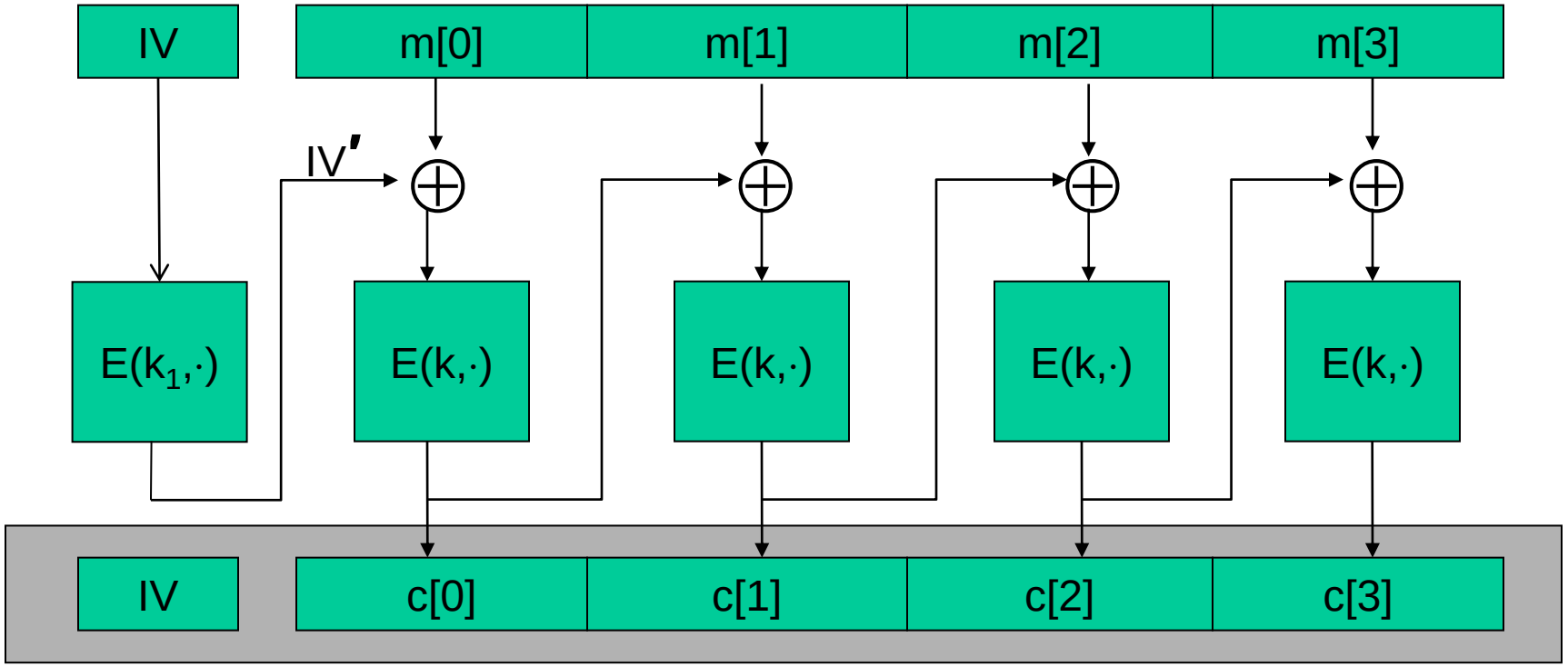
Pode usar IV *único* (por exemplo, contador)

mas o primeiro passo do CBC deve ser $IV' \leftarrow E(k_1, IV)$

benefício: será seguro transmitir IV com o texto cifrado

CBC com IVs únicos

IV único significa: (k,IV) é usado apenas para uma msg
como pode ser previsível, usar $E(k_1,\cdot)$ bom (PRF)

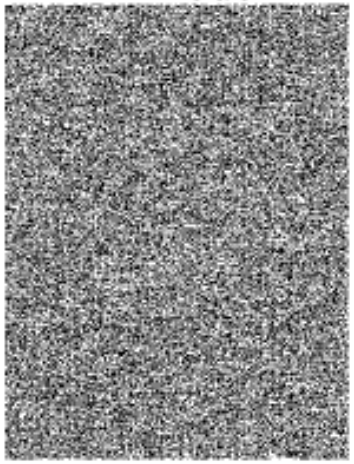


texto cifrado

Em figuras

An example plaintext

Encrypted with AES in CBC mode

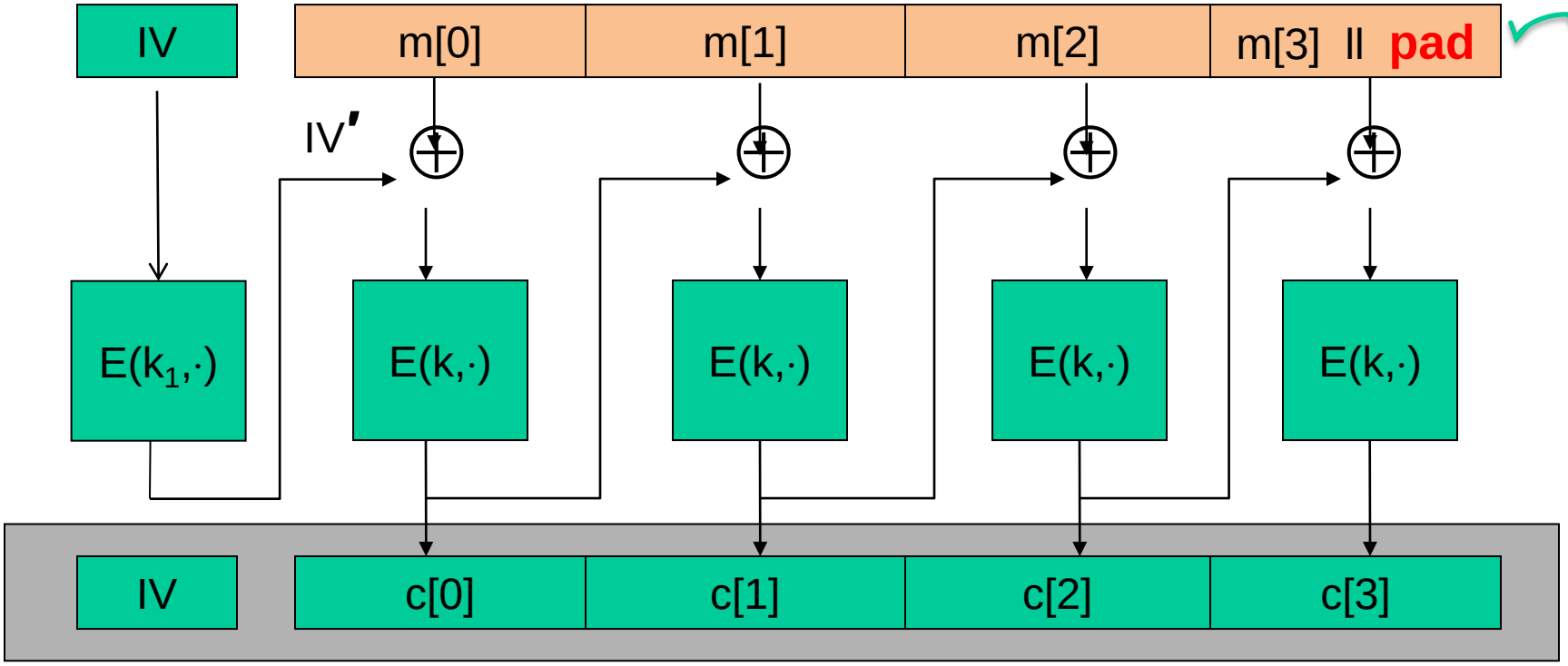


Um exemplo de Crypto API (OpenSSL)

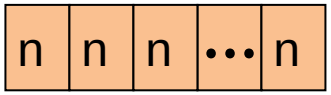
```
void AES_cbc_encrypt(
        const unsigned char *in,
        unsigned char *out,
        size_t length,
        const AES_KEY *key,
        unsigned char *ivec,          ← usuário fornece o IV
        AES_ENCRYPT or AES_DECRYPT);
```

Quando o IV (ou nonce) não é aleatório, deve ser cifrado antes de usar (antes de fornecer o parâmetro na função)

Padding do CBC



TLS: para n>0, n byte pad é



se pad não é necessário, adicionar um bloco dummy

(16 bytes com o número 16 em cada byte)

Esquema de Padding PKCS5 - Funcionamento

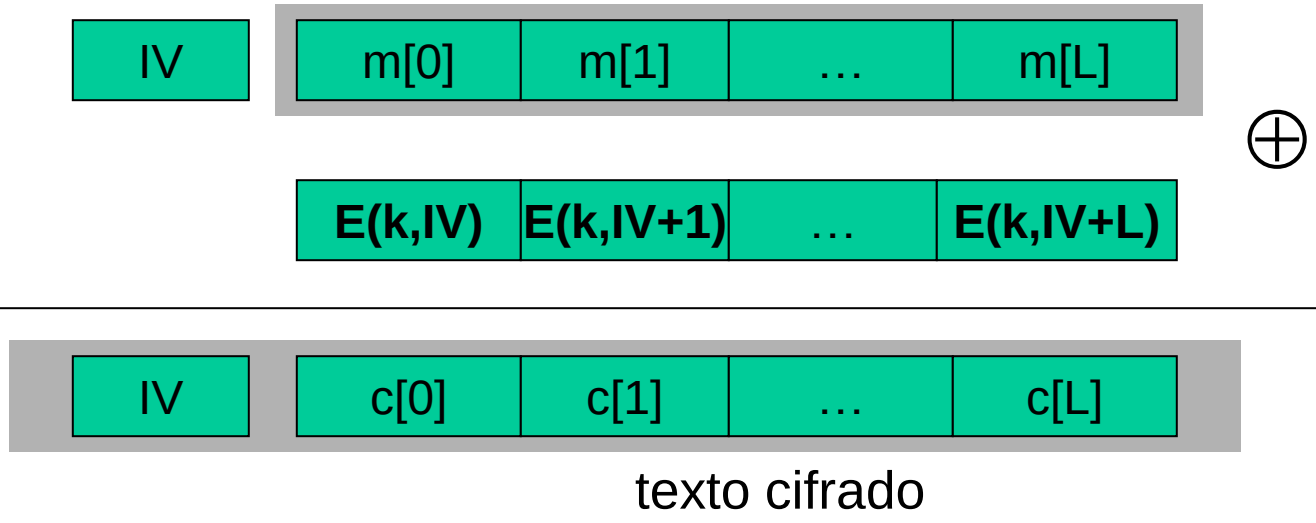
- **O número de bytes a ser "enchido/inserido" é igual a**
  - "8 - numeroDeBytes(textoPlano) mod 8"
  - Assim, 1 a 8 bytes serão inseridos (padded) ao texto plano dependendo do tamanho dos dados do texto plano

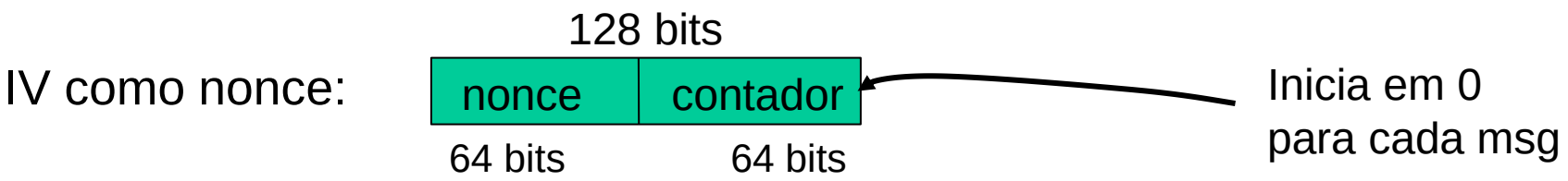
- **Todos os bytes inseridos tem o mesmo valor – o número de bytes inseridos**

```
Se numeroDeBytes(textoPlano) mod 8 == 7, PM = M + 0x01
Se numeroDeBytes(textoPlano) mod 8 == 6, PM = M + 0x0202
Se numeroDeBytes(textoPlano) mod 8 == 5, PM = M + 0x030303
   ...
Se numeroDeBytes(textoPlano) mod 8 == 0, PM = M
```

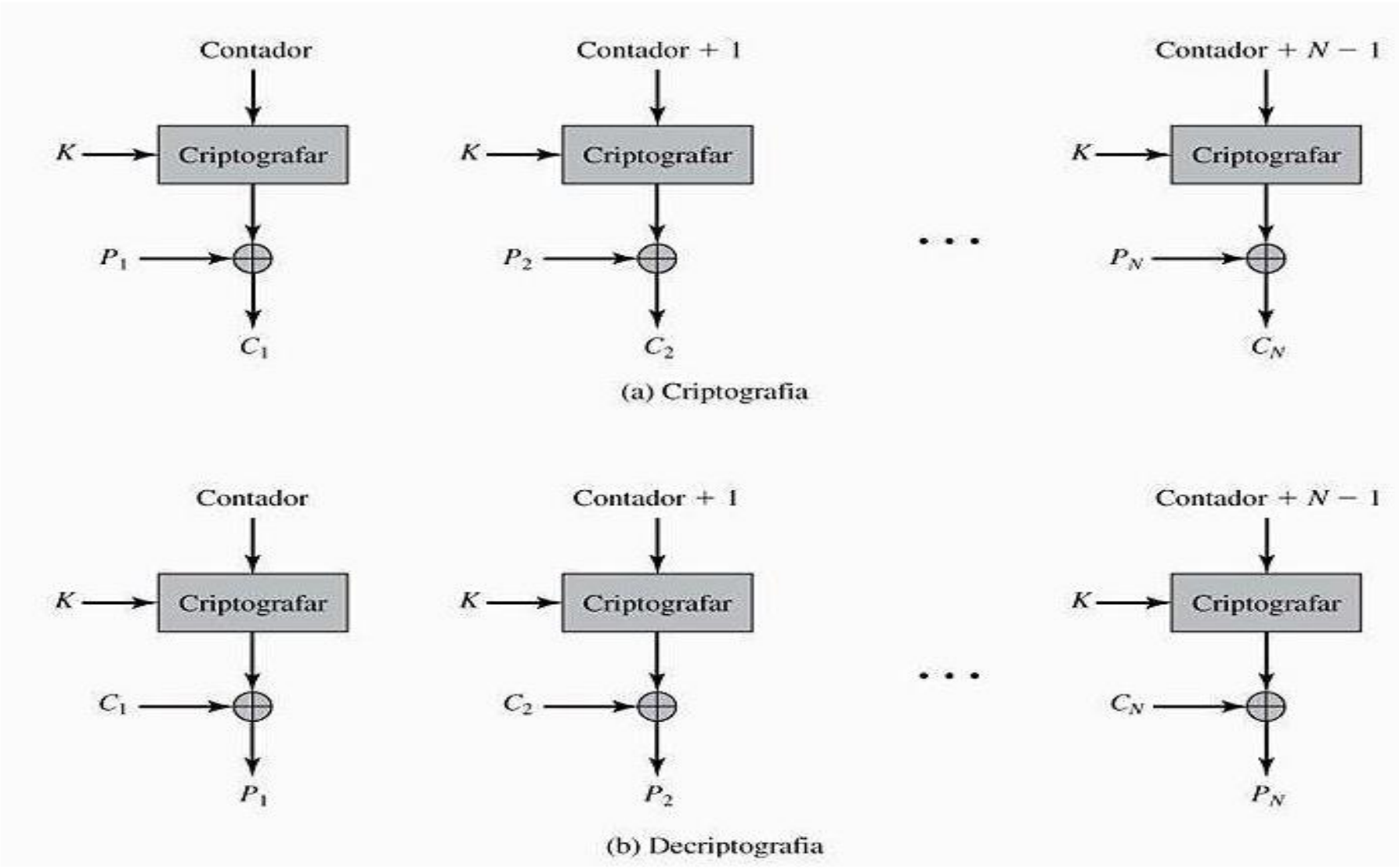
Modo 3 – Counter (CTR)

- Modo Counter com um IV aleatório: (cifragem paralela)
- Superior ao CBC - com AES 128 bits de bloco, depois de 2^{64} blocos AES, DEVE ser trocada a chave
- IV aleatório escolhido a cada nova msg ou IV como nonce





Modo 3 – Counter (CTR)



DES - Data Encryption Standard

- **Início 1970: Horst Feistel projetou Lucifer na IBM**

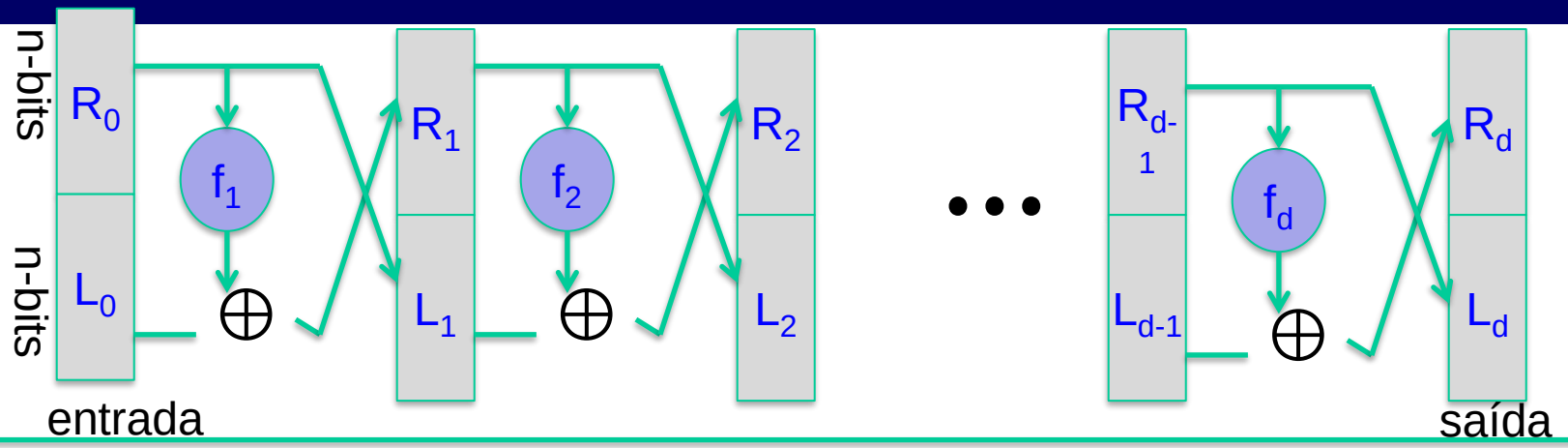
  chave = 128 bits ; bloco = 128 bits

- **1976: DES foi adotado como padrão americano**

  chave = 56 bits ; bloco = 64 bits

- **DES básico: chaves de 56 bits**
  - 1997 - quebrado por pesquisa exaustiva em apenas parte de um dia (22 horas e 15 minutos) usando hardware especializado
- **DES Triplo: chave efetiva de 112 bits**
  - Cifra, Decifra, Cifra usando o DES com três chaves diferentes
  - Impossível de ser "quebrado" com as técnicas conhecidas
- **2000: NIST adotou Rijndael como padrão com o nome AES (*Advanced Encryption Standard*) para substituir DES**

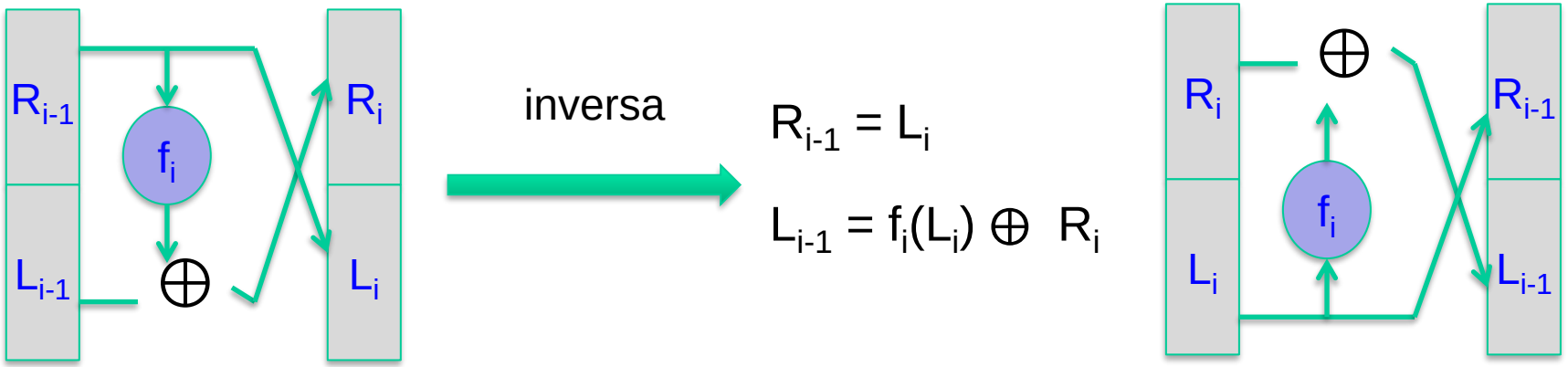
DES: ideia chave – Feistel Network



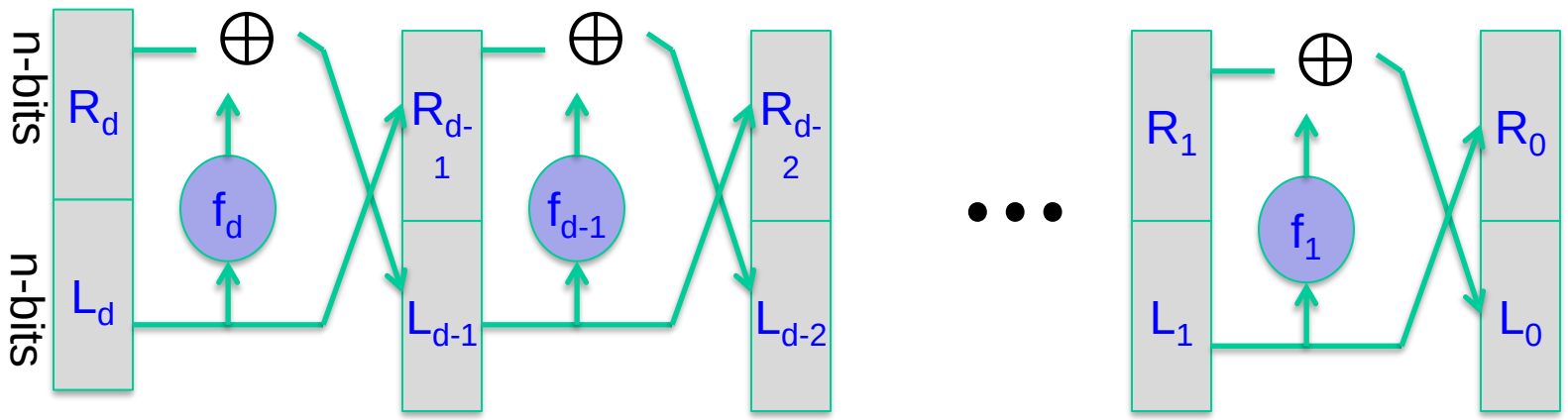
Afirmação: para todo $f_1, \ldots, f_d$: $\{0,1\}^n \longrightarrow \{0,1\}^n$

Feistel network $F: \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$ é inversível

Prova: construir inversa

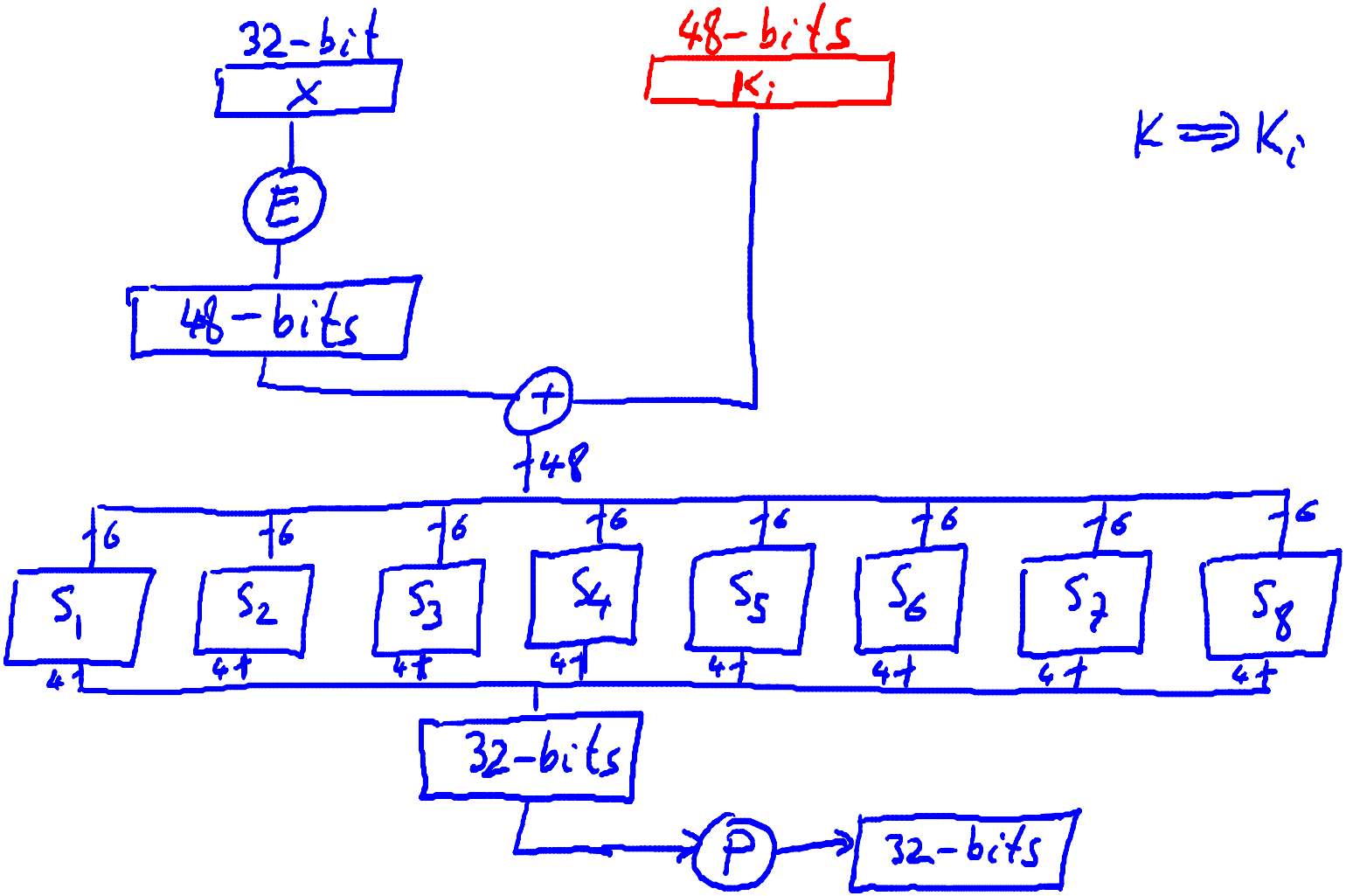


Circuito de decifragem



- Inversão é basicamente o mesmo circuito, com $f_1, \ldots, f_d$ aplicadas na ordem reversa
- Método geral para construir funções inversíveis (cifras de bloco) a partir de funções arbitrárias
- Usada em muitas cifras de bloco… mas não no AES
- Cifras de Feistel ou modificações : Blowfish, CAST-128, DES, RC5, Triple DES, Twofish, GOST 28147-89 (foi padrão russo)

A função $F(k_i, x)$



S-box: função $\{0,1\}^6 \longrightarrow \{0,1\}^4$, implementada como tabela

S-boxes

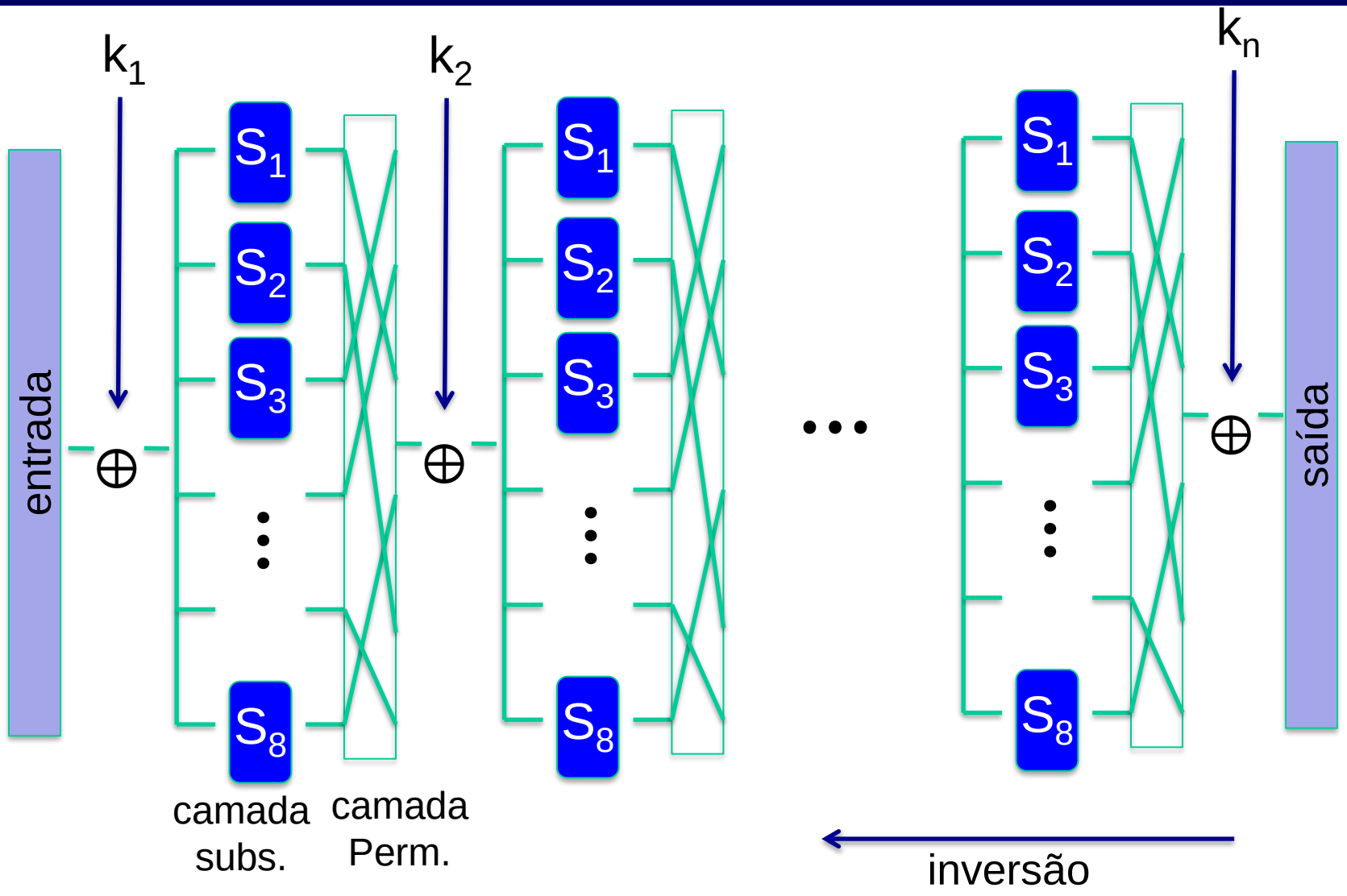
$$S_i: \{0,1\}^6 \longrightarrow \{0,1\}^4$$

S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

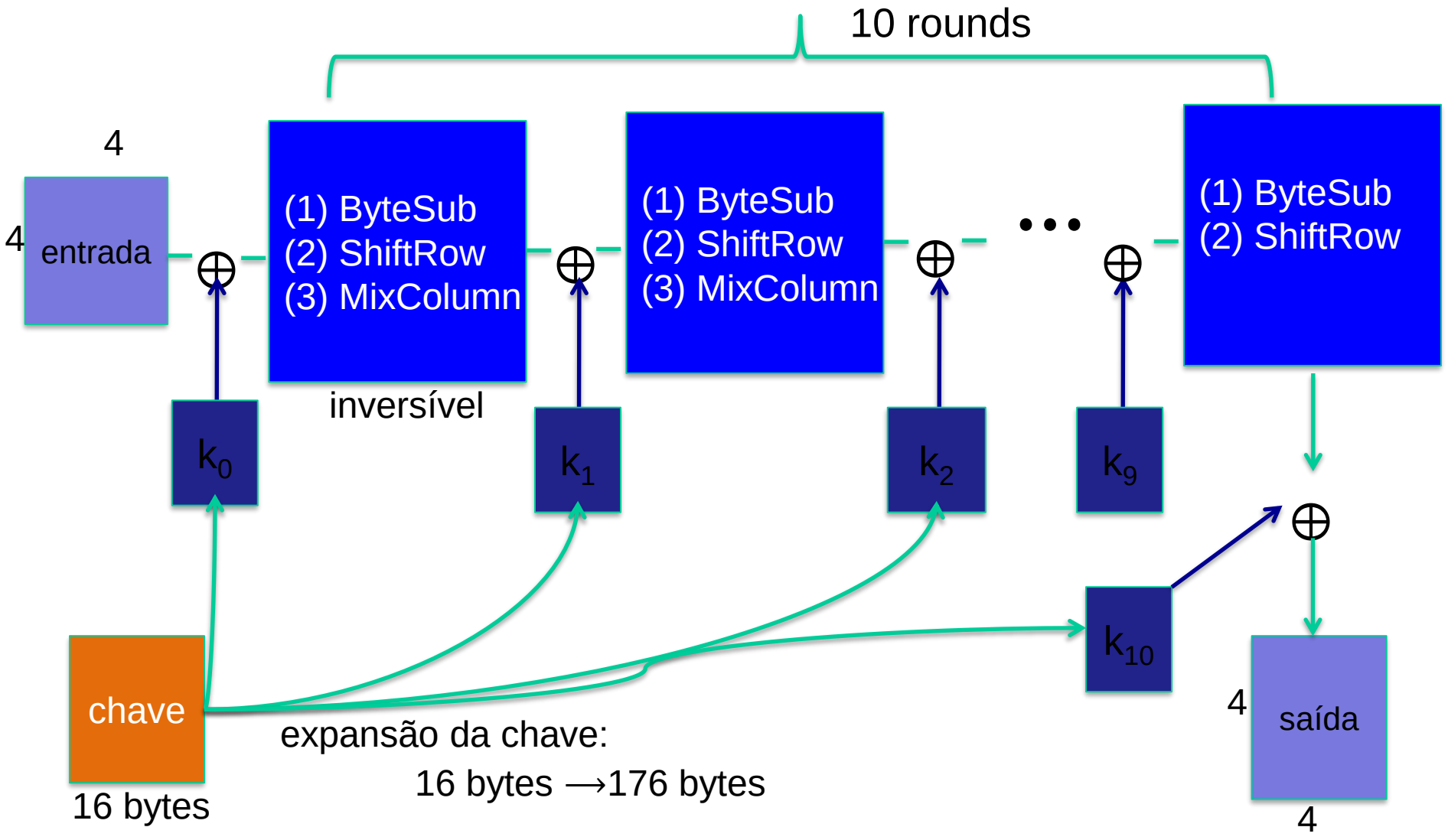
O processo AES

- **FIPS PUB 197** http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf
- **Em 1997 NIST fez uma concorrência entre "terceiras partes":**
  - o algoritmo deveria ser público e livre de *royalties*
  - algoritmos deveriam implementar criptografia de chave simétrica como um cifrador de bloco e suportar tamanhos de blocos de 128 bits e tamanhos de chaves de 128, 192 e 256 bits
- **1998: 15 submissões. Abril 1999 - cinco finalistas:**
  - Twofish - Bruce Shneier, Neils Ferguson e outros
  - RC-6 - RSA Laboratory
  - Mars - IBM
  - Serpent - Ross Anderson (UK), Eli Biham (Israel), e Lars Knudson (Noruega)
  - **Rijndael** - Vincent **Rij**men **e** Joan **Dae**men da Bélgica (ganhador)
- **Existem aproximadamente 10^{21} vezes mais chaves no AES de 128 bits de chave do que no DES 56 bits de chave**
- http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html

AES é rede Substituição-Permutação (não é Feistel)

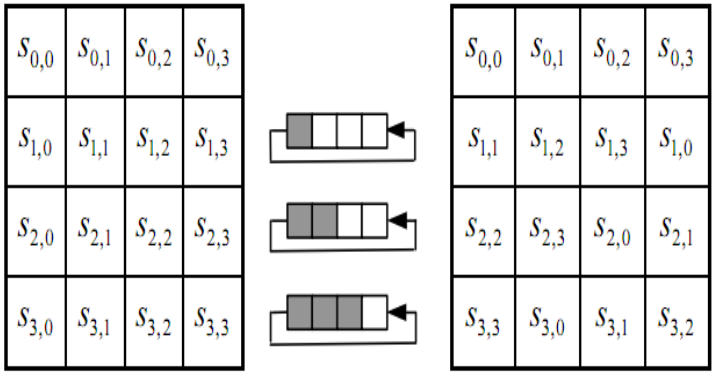


Esquema AES-128

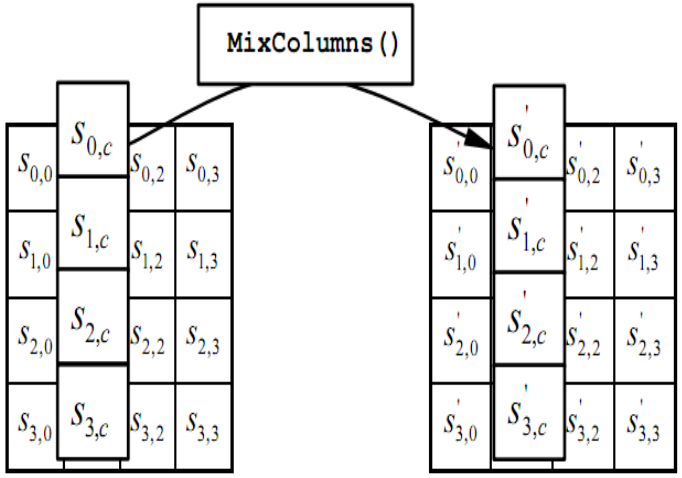


A função round

- **ByteSub: um 1 byte S-box; tabela 256 bytes (facilmente calculada)**
- **ShiftRows:**



- **MixColumns:**



Compensação tamanho código x desempenho

	tamanho código	desempenho
Funções round pré-computadas (24KiB or 4KiB)	maior	Mais rápido: tabelas e xors
Pre-computar apenas S-box (256 bytes)	menor	Mais lento
Sem pre-computação	pequeno	O mais lento

Comentários

- **Até aqui os algoritmos criptográficos (fluxo, bloco) nos seus vários modos fornecem APENAS CONFIDENCIALIDADE (proteção contra intromissão).**
  - Não fornecem integridade
  - Não fornecem autenticidade
- **Criptografia simétrica: Problema de distribuição de chaves dos sistemas de chave secreta**
  - deve-se compartilhar a chave secreta com outra parte antes de iniciar a conversação
  - se você quer conversar com n partes, são requeridas n chaves diferentes
- **Criptografia simétrica: no geral, mais rápidos que os sistemas assimétricos**
- **Capítulos 2,3,5,6 do livro do Stallings**
- **Links:** http://williamstallings.com/NetworkSecurity/styled-8/