

# Fundamentos de Segurança

---

Prof. Marcos Dantas

[mdo@ufc.br](mailto:mdo@ufc.br)

# Sumário

- Fundamentos
  - Criptografia Simétrica
  - Criptografia Assimétrica
  - Integridade
  - Autenticação
  - Assinatura Digital e Certificação Digital
  - Distribuição de Chaves
  - Serviços de Rede
-

# Sumário

- Fundamentos
  - Criptografia Simétrica
  - Criptografia Assimétrica
  - Integridade
  - Autenticação
  - Assinatura Digital e Certificação Digital
  - Distribuição de Chaves
  - Serviços de Rede
-

# Segurança da Informação - Serviços

## Tripé da Segurança:

### ■ Confidencialidade:

- apenas o transmissor e o receptor pretendido deveriam “entender” o conteúdo da mensagem
- Transmissor criptografa mensagem, Receptor decodifica a mensagem

### ■ Integridade de mensagens:

- Transmissor e receptor querem assegurar que as mensagens não foram alteradas, (em trânsito, ou depois) sem detecção
- Alteração consiste em:
  - Escrita, mudança, deleção, criação, duplicação.

### ■ Disponibilidade:

- Serviços devem ser acessíveis e disponíveis para os usuários autorizados

## Serviços Adicionais:

### ■ Autenticação:

- Transmissor e receptor querem confirmar a identidade um do outro

### ■ Não-repudiação:

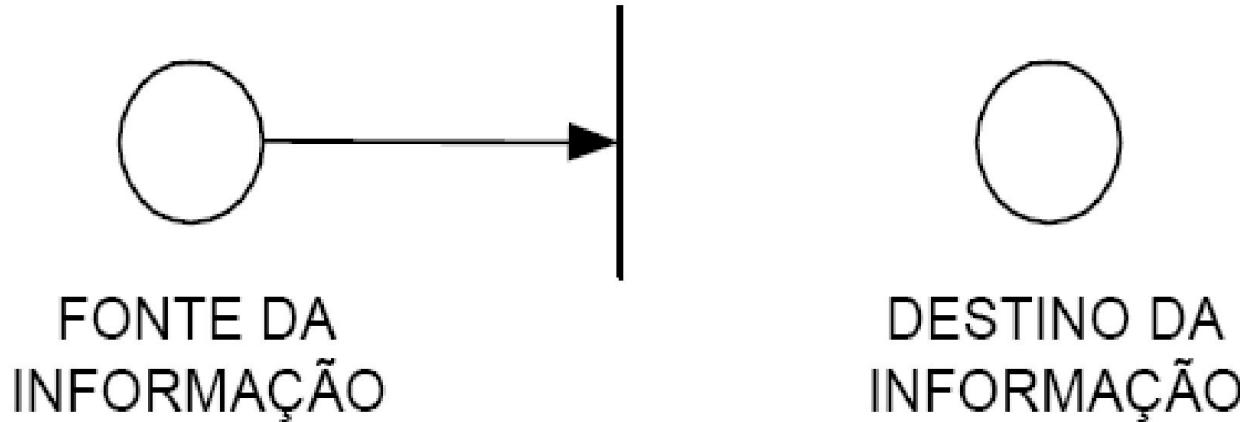
- Previne que alguém negue o envio e/ou recebimento de uma mensagem.

# Exemplo de aplicação: Compra pela Internet

- ❑ Informação que permite a transação - valor e descrição do produto adquirido
    - precisa estar disponível no dia e na hora que o cliente desejar efetuá-la
      - ❑ (disponibilidade).
  - ❑ O valor da transação não pode ser alterado
    - ❑ (integridade).
  - ❑ Somente o cliente que está comprando e o comerciante devem ter acesso à transação
    - ❑ (confidencialidade).
  - ❑ O cliente que está comprando deve ser quem diz ser
    - ❑ (autenticidade).
  - ❑ O cliente tem como provar o pagamento e o comerciante não tem como negar o recebimento
    - ❑ (não-repúdio).
  - ❑ O conhecimento do conteúdo da transação fica restrito aos envolvidos
    - ❑ (confidencialidade).
-

# Tipos de Ataques

## Interrupção

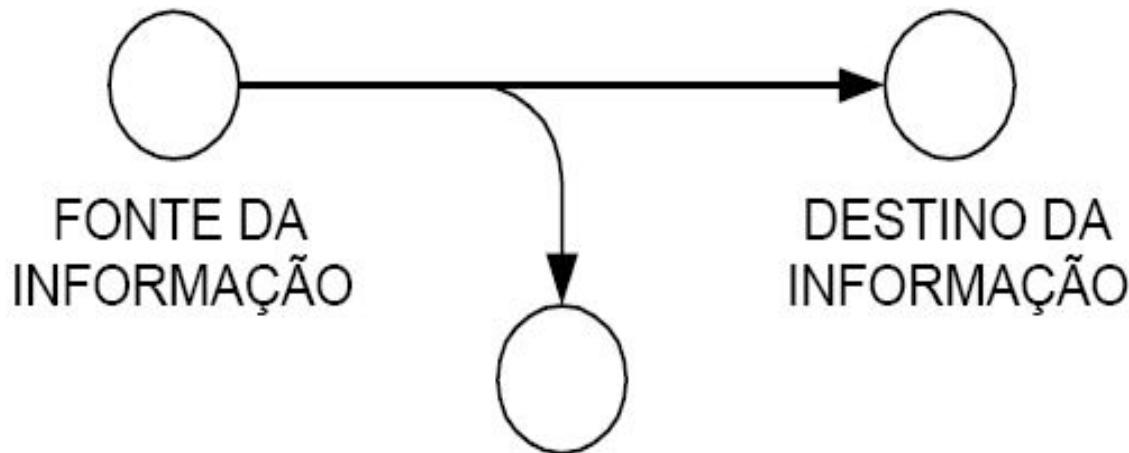


- Um recurso do sistema é destruído ou se torna indisponível ou inútil.
- Este é um ataque à **disponibilidade**.
- Exemplos:
  - destruição de uma peça de hardware como um disco rígido;
  - o corte de uma linha de comunicação;
  - tornar indisponível um sistema de gerência de arquivos.



# Tipos de Ataques

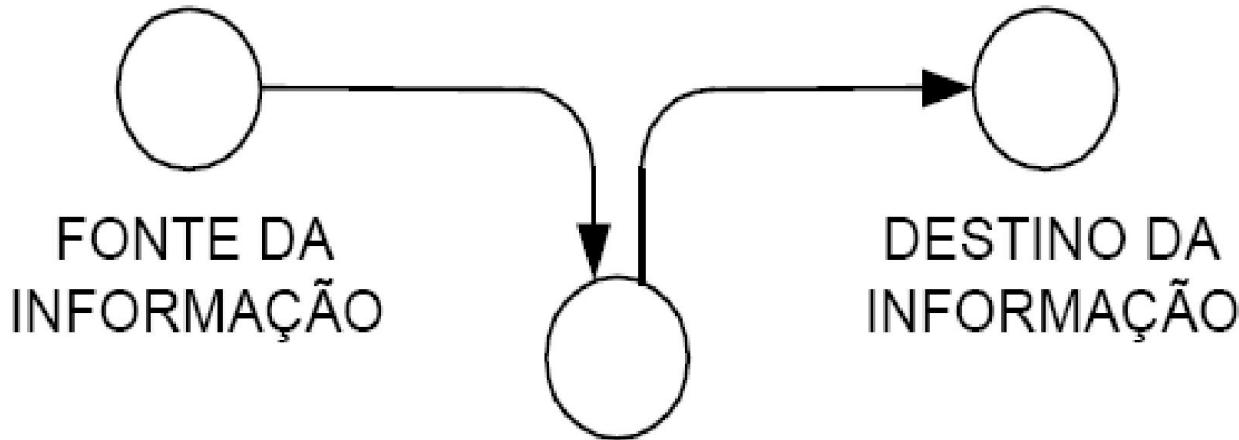
## Interceptação



- ❑ Uma parte não autorizada ganha acesso a um recurso.
- ❑ Este é um ataque à **confidencialidade**.
- ❑ A parte não autorizada pode ser pessoa, programa ou computador.
- ❑ Exemplos:
  - ❑ grampos em linhas para capturar dados da rede;
  - ❑ cópia de programas ou arquivos;
  - ❑ análise de tráfego.

# Tipos de Ataques

## Modificação

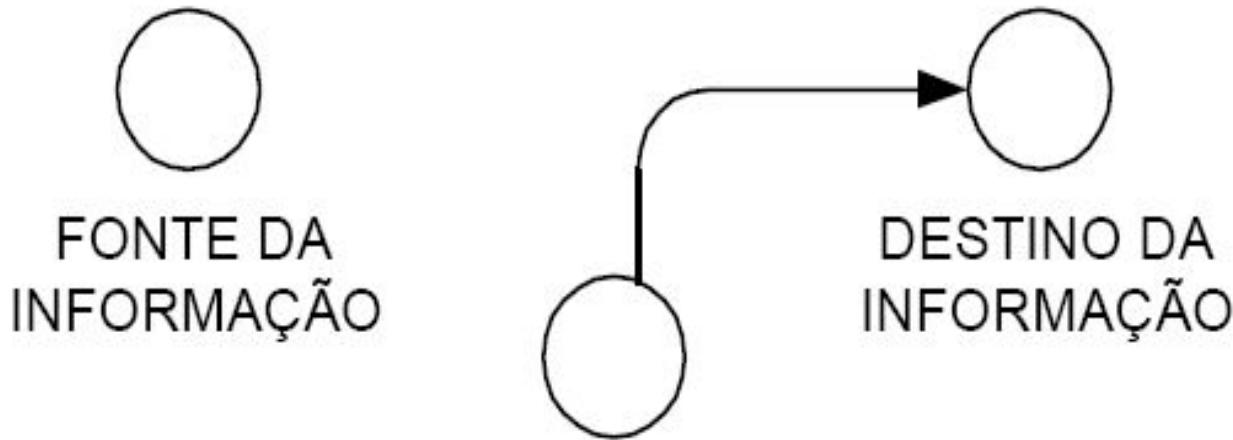


- Uma parte não autorizada não somente ganha acesso, mas também adultera o bem.
- Este é um ataque à **integridade**.
- Exemplos:
  - mudança de valores em um arquivo de dados;
  - alteração de um programa de tal forma que ele se comporte de maneira diferente;
  - modificação do conteúdo da mensagem sendo transmitida.

---

# Tipos de Ataques

## Fabricação



- Uma pessoa não autorizada insere objetos no sistema.
- Este é um ataque à **autenticidade**.
- Exemplos:
  - inserção de mensagens maliciosas na rede;
  - adição de registros em um arquivo.

# O que é Criptografia?





# **Criptografia**

**(*kriptos* = oculto + *graphos* = grafia)**

**“Arte ou a ciência de escrever em  
cifras (código).”**

# Criptografia - Fundamentos

- **Criptografia** - Conjunto de técnicas que permitem tornar “incompreensível” uma mensagem originalmente escrita com clareza, de forma a permitir que apenas o destinatário a decifre e a compreenda.
  - **Criptoanálise** - do grego **kryptos + análisis (decomposição)** - ciência que estuda a decomposição do que está oculto ou a “quebra” do sistema criptográfico.
  - **Criptologia** - Criptografia + Criptoanálise.
-

# Criptografia

- “Do correio eletrônico à telefonia celular, do acesso seguro a servidores WEB à moeda eletrônica, **a criptografia é parte essencial dos sistemas de informação de hoje.**”
  - Pode **prevenir fraudes** em comércio eletrônico e **garantir a validade** de transações financeiras.
  - Usada apropriadamente, protege a anonimidade e fornece provas de identidade de pessoas.
-

# Criptografia - Fundamentos

## Pré-requisitos da Criptografia

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

# Criptografia - Terminologia

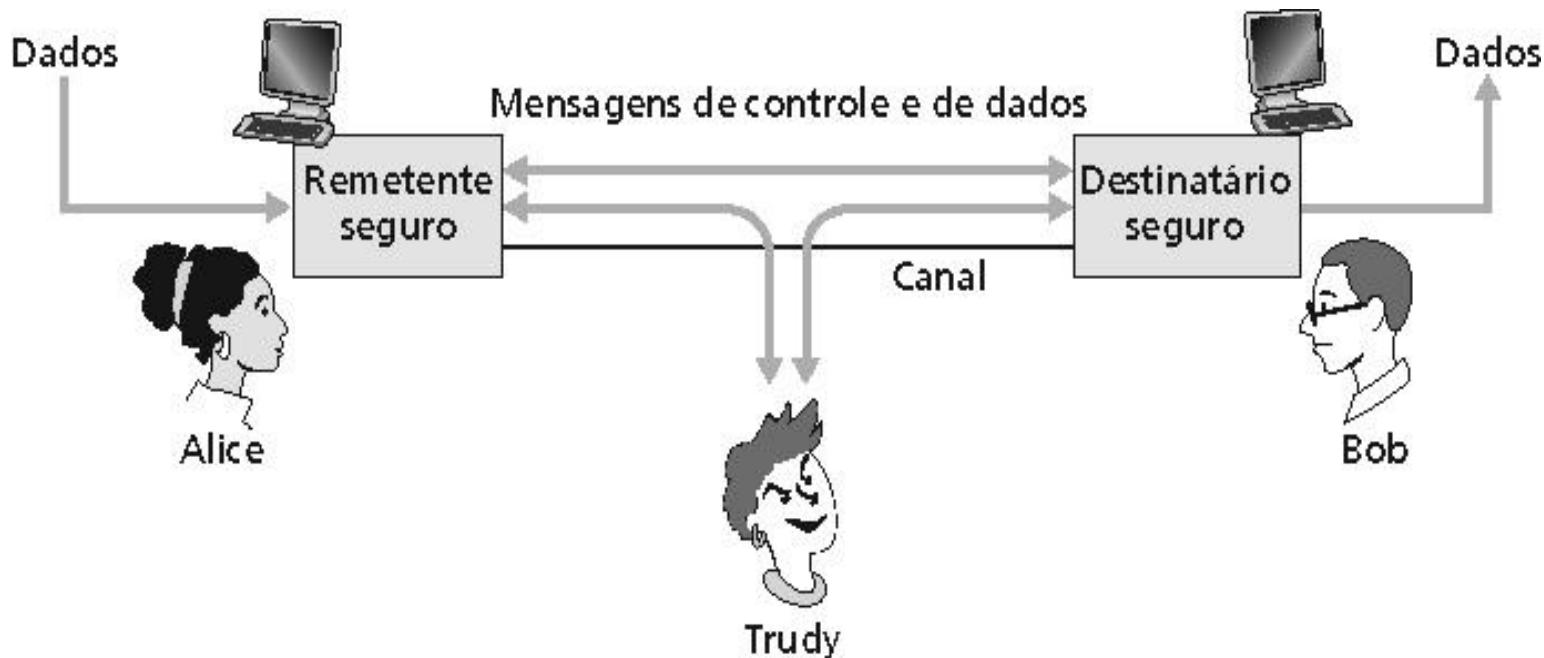
Termo	Descrição
<b>Texto claro, simples (<i>plain text</i>) ou mensagem</b>	<b>Mensagem original</b>
<b>Cifração ou criptografia</b>	<b>Processo de “embaralhar” a mensagem de forma a ocultar seu conteúdo de outrem</b>
<b>Texto cifrado (<i>cipher text</i>, <i>Encrypted Text</i>) ou criptograma</b>	<b>Mensagem cifrada</b>
<b>Decifração ou descriptografia</b>	<b>Processo inverso de recuperação da mensagem a partir do criptograma</b>
<b>Chave criptográfica</b>	<b>Parâmetro de controle. Segredo por meio do qual a mensagem pode ser cifrada ou decifrada</b>

# Criptografia - Terminologia

Termo	Descrição
<b>Algoritmo criptográfico</b>	<b>Transformação matemática - converte uma mensagem em claro em uma mensagem cifrada e vice-versa.</b>
<b>Alice</b>	<b>Origem - Cifra uma mensagem.</b>
<b>Bob</b>	<b>Destino - Decifra uma mensagem.</b>
<b>Trudy</b>	<b>Intruso – tenta interceptar e decifrar a mensagem.</b>

# Amigos e inimigos: Alice, Bob, Trudy

- Bem conhecidos no mundo da segurança de redes
- Bob e Alice (amantes!) querem se comunicar “seguramente”
- Trudy, a “intrusa” pode interceptar, apagar, acrescentar mensagens



# Quem poderiam ser Bob e Alice?

- ... bem, Bobs e Alices do *mundo real*!
  - Browser/servidor Web para transações eletrônicas (ex.: compras on-line)
  - Cliente/servidor de banco on-line
  - Servidores DNS
  - Roteadores trocam atualizações de tabela de roteamento
  - **Outros exemplos?**
-

# Fundamentos de Criptografia

- Componentes básicos para o ciframento de uma mensagem:
  - **algoritmo**
  - **chave**

Princípio de Kerckhoff (1883): *Todos os algoritmos devem ser públicos; apenas as chaves são secretas.*

**Algoritmo secreto:** segurança pela obscuridade.

# Fundamentos de Criptografia

## Vantagens importantes para o uso de chaves

- Permite a utilização do mesmo algoritmo criptográfico para a comunicação com diferentes receptores, trocando apenas a chave.
  - Permite trocar facilmente a chave no caso de uma violação, mantendo o mesmo algoritmo.
  - Número de chaves possíveis depende do tamanho (número de bits) da chave.
    - **Exemplo:** uma chave de 8 bits permite uma combinação de no máximo 256 chaves. Quanto maior o tamanho da chave, mais difícil quebrá-la.
-

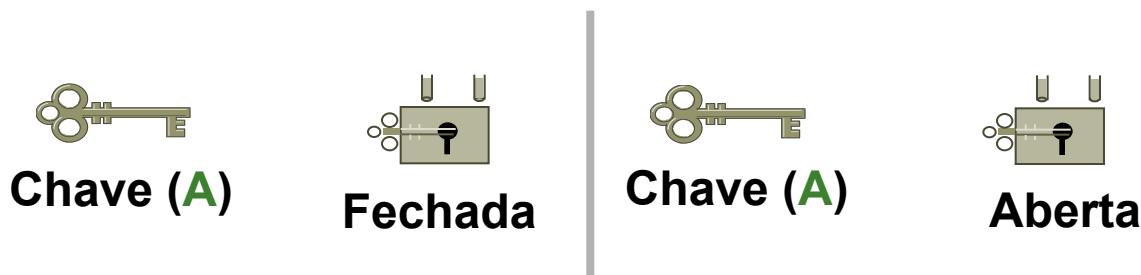
# Algoritmos Púlicos vs Privados

Algoritmos proprietários	Algoritmos públicos
São aqueles que somente poucas pessoas conhecem o código	São aqueles que todos conhecem o código
<b>Vantagens:</b> Geralmente realizar criptoanálise conhecendo o código é difícil, sem conhecer o código é ainda mais difícil.  <b>Desvantagens:</b> O código somente foi avaliado por poucas pessoas, com isso, podem existir fragilidades não descobertas. Pode ser feita engenharia reversa em cima de um produto que implemente o algoritmo e pode ser descoberto o código.	<b>Vantagens:</b> O código foi avaliado por muitas pessoas tornado o algoritmo mais confiável. Maior facilidade de padronização e produção por fabricantes diferentes.  <b>Desvantagens:</b> No caso de descoberta de uma vulnerabilidade no algoritmo, imediatamente todos os usuários estão comprometidos.

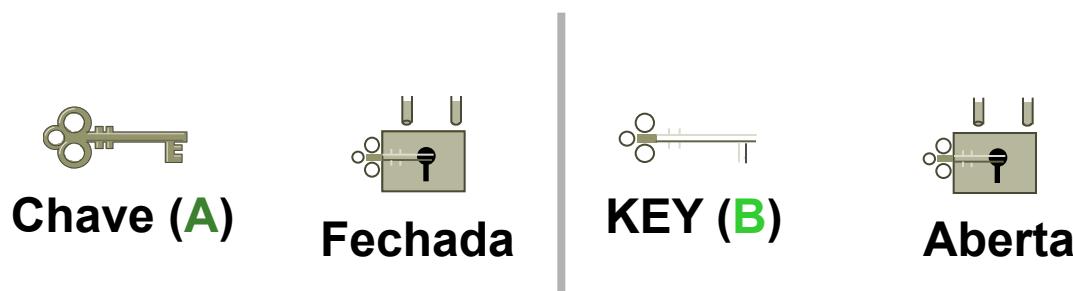
# Criptografia - Tipos

## Tipos básicos de Criptografia (em relação ao uso de chaves)

- Criptografia Simétrica (chave secreta)



- Criptografia Assimétrica (chave pública)

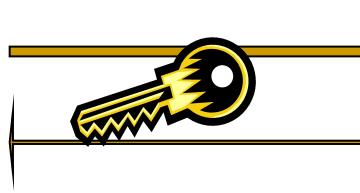


# Sumário

- Fundamentos
  - Criptografia Simétrica
  - Criptografia Assimétrica
  - Integridade
  - Autenticação
  - Assinatura Digital e Certificação Digital
  - Distribuição de Chaves
  - Serviços de Rede
-

# Criptografia Simétrica

**Texto claro**



**Mensagem cifrada**

- Utiliza uma mesma chave tanto para cifrar como para decifrar (ou pelo menos a chave de decifração pode ser obtida trivialmente a partir da chave de cifração)
- A mesma chave utilizada para “fechar o cadeado” é utilizada para “abrir o cadeado”.

# Criptografia Simétrica

**Criptografia Simétrica - Requer uma chave compartilhada**

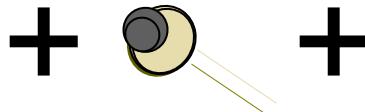
## Criptografia

Para: Banco  
De: Affonso

Data: 16, Abr, 2001

Transferir R\$ 2,5  
milhões da conta  
254674-12 para  
a conta 071517-08

Affonso



Algoritmo



\*> \*ql3\*UY  
#~00873/JDI  
c4(DH: IWB(883  
LKS9UI29as9eea  
qw9vijhas9djerhp7  
(\*Y23k^wbvlqkwc  
zqw-\_89237xGyjdc  
Biskdue di7@94

## Descriptografia

\*> \*ql3\*UY  
#~00873/JDI

c4(DH: IWB(883

LKS9UI29as9eea  
qw9vijhas9djerhp7  
(\*Y23k^wbvlqkwc  
zqw-\_89237xGyjdc  
Biskdue di7@94

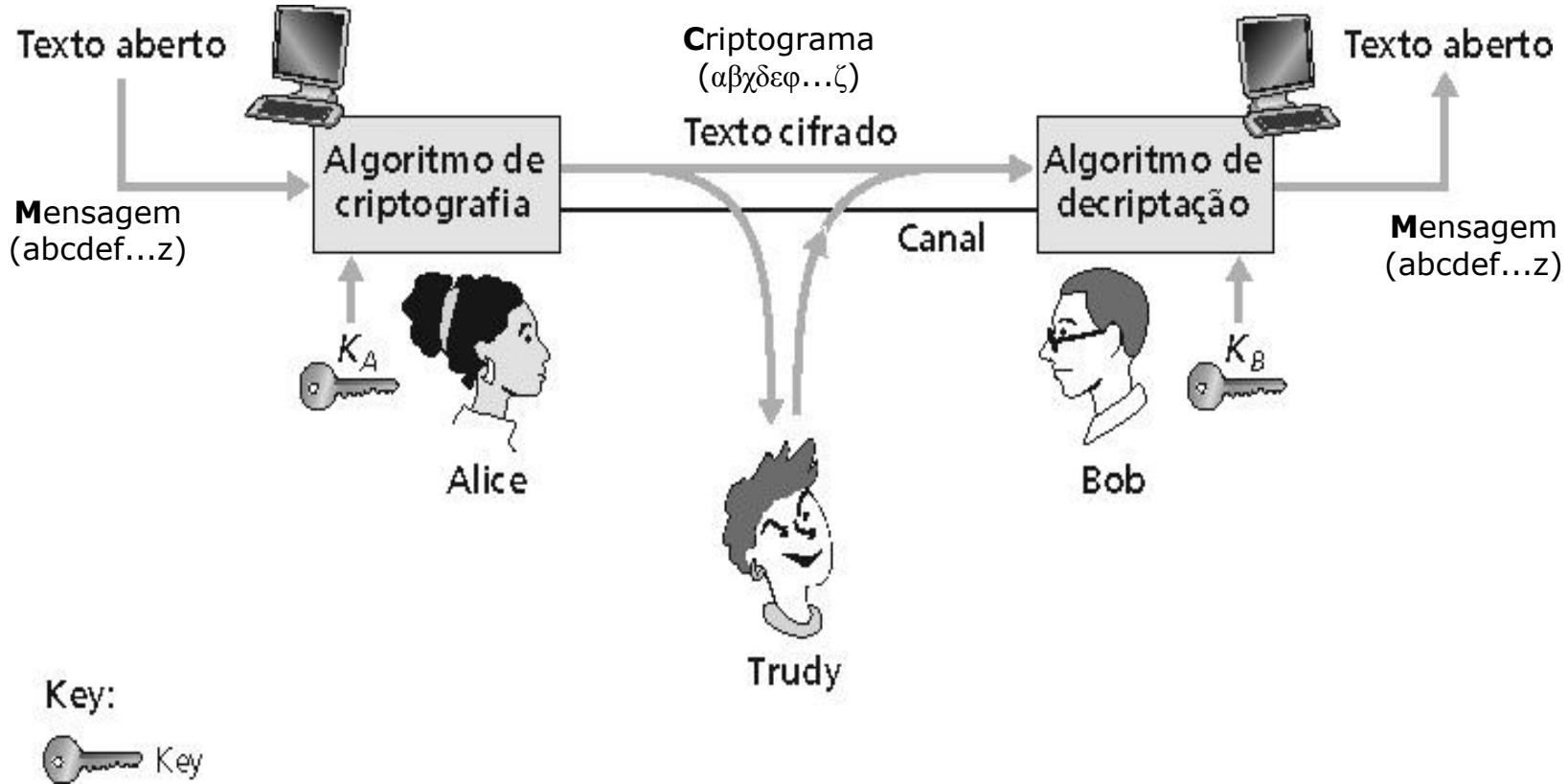


Algoritmo



Para: Banco  
De: Affonso  
Data: 16, Abr, 2001  
Transferir R\$ 2,5  
milhões da conta  
254674-12 para  
a conta 071517-08  
Affonso

# A linguagem da criptografia

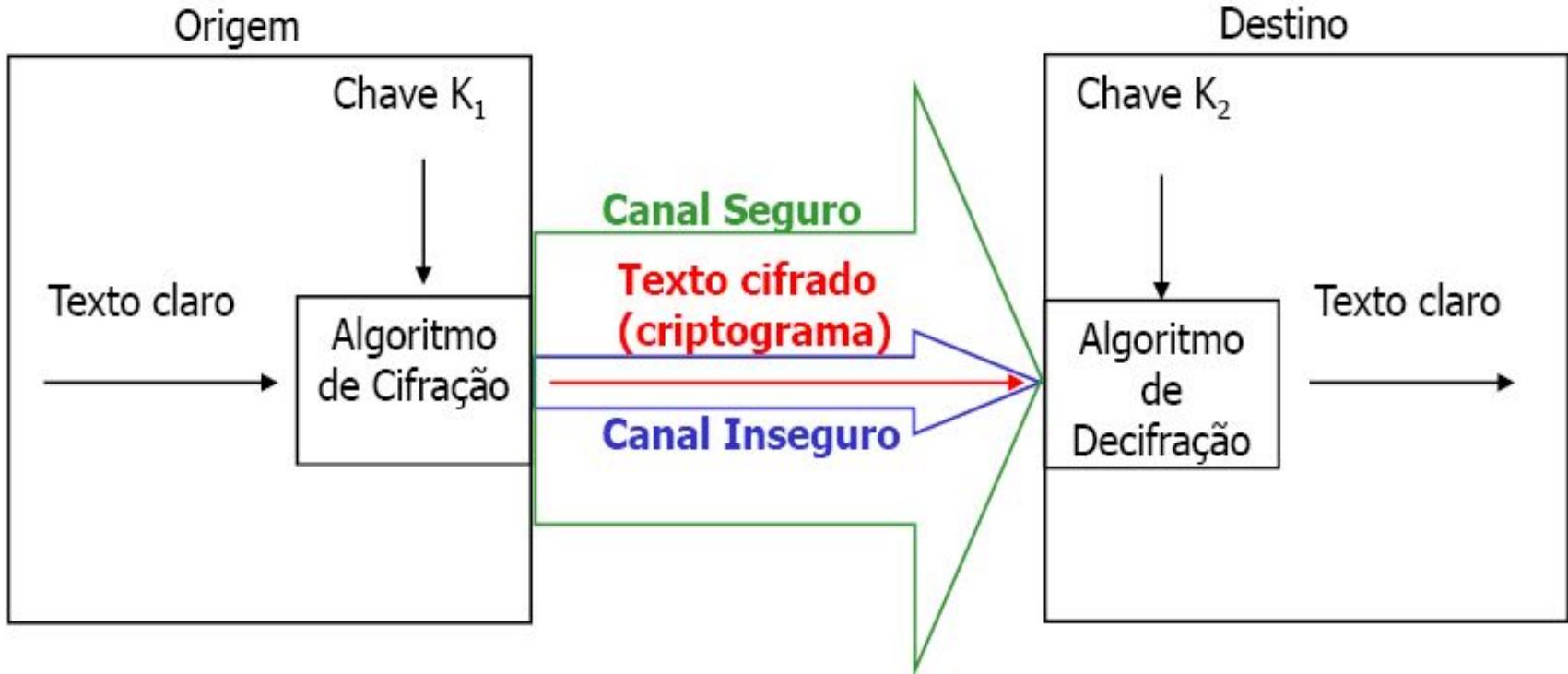


**Chave simétrica** de criptografia: as chaves do transmissor e do receptor são idênticas

# Criptografia Simétrica

- Alice **cifra** uma mensagem - utiliza um algoritmo de ciframento e uma chave secreta para transformar uma mensagem clara em um texto cifrado.
  - Bob **decifra** uma mensagem - utiliza o algoritmo de deciframento correspondente e a mesma chave para transformar o texto cifrado em uma mensagem em claro.
  - Trudy - não possui a chave secreta, mesmo conhecendo o algoritmo, não consegue decifrar a mensagem.
  - **A segurança do sistema reside não mais no algoritmo e sim na chave empregada.** É ela que agora, no lugar do algoritmo, deverá ser mantida em segredo por Alice e Bob.
-

# Elementos de um sistema criptográfico



- Problema importante: Distribuição da chave

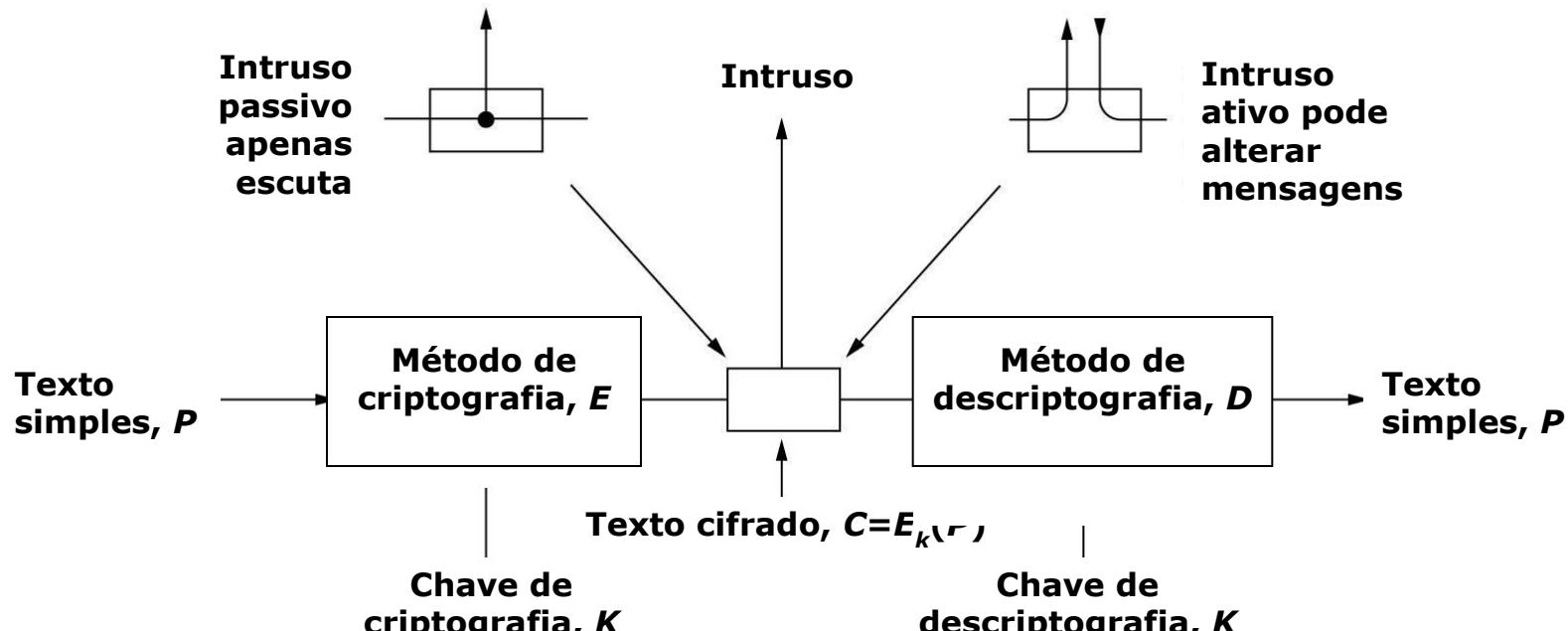
# Criptografia Simétrica

- **Algoritmos simétricos** - exigem que a chave seja mantida secreta, do conhecimento exclusivo dos dois interlocutores.
- É requerido um *canal seguro* que permita a um usuário transmitir a chave ao seu interlocutor.

Se uma pessoa quer se comunicar com outra com segurança, ela deve passar primeiramente a chave utilizada para cifrar a mensagem. Este processo é chamado **distribuição de chaves**.

# Criptografia Simétrica

## Modelo de criptografia (para uma cifra de chave simétrica)

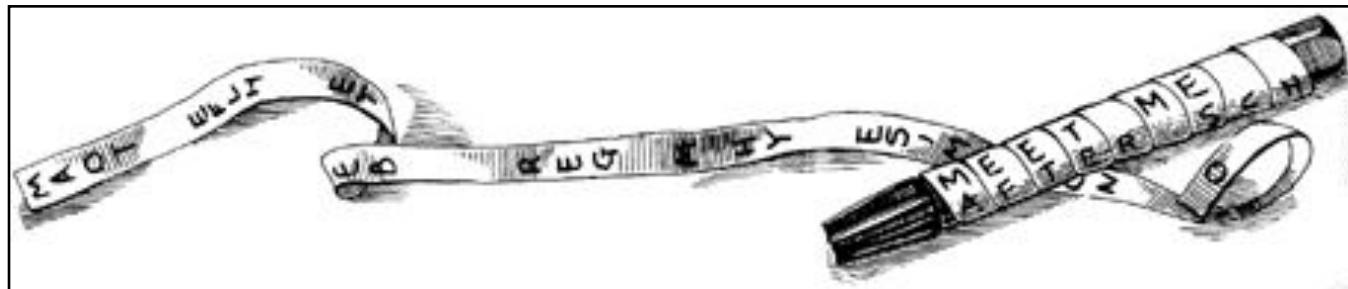


$$D_k(E_k(P)) = P$$

# Criptografia de chave simétrica – Bastão de Licurgo

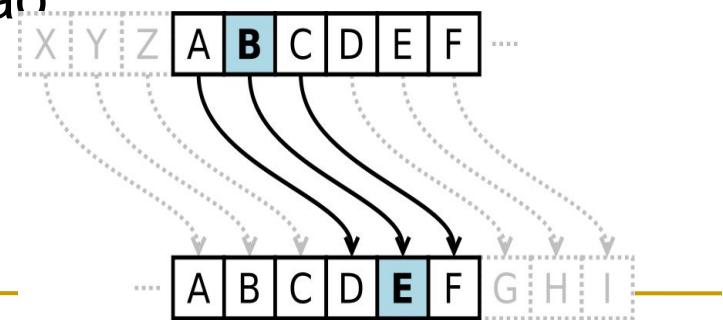
## □ 487 a.C. - Bastão de Licurgo

- O remetente escreve a mensagem ao longo do bastão e depois desenrola a tira, a qual então se converte numa sequência de letras sem sentido. O mensageiro usa a tira como cinto, com as letras voltadas para dentro. O destinatário, ao receber o "cinto", enrola-o no seu bastão, cujo diâmetro é igual ao do bastão do remetente. Desta forma, pode ler a mensagem.



# Criptografia de chave simétrica – Cifra de Cesar

- 50 a.C. - a **cifra de César** é uma das mais simples e conhecidas técnicas de encriptação e foi usada por Júlio César para se comunicar secretamente com os seus generais .
- Nessa cifra, cada letra do texto limpo é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes. Por exemplo, com uma troca de 3 posições, A seria substituído por D, B viraria E e assim por diante.
- Como todas as cifras de substituição monoalfabéticas, a cifra de César é facilmente decifrada por meio de estatística de freqüência das letras em um idioma e na prática não oferece nenhuma segurança na comunicação



# Criptografia de chave simétrica – Cifra de Cesar

- A encriptação e decriptação usando a cifra de César pode ser representada pela transformação das letras em números ( $A = 0$ ,  $B = 1, \dots, Z=25$ ) , conforme a seguir:

- Encriptação:  $C = E_K(P) = (P + k)mod26$
- Decriptação:  $P = D_K(C) = (C - k)mod26$

Sendo:

**P**: cada letra do texto limpo

**C**: cada letra do texto cifrado

**K**: o número de deslocamentos no alfabeto

---

# Criptografia de chave simétrica – Cifra de Cesar

Letra	Posição	Soma Algébrica	Soma Módulo 26	Nova Letra
A	0	$0 + 3 = 3$	$(0 + 3) \text{ MOD } 26 = 3$	D
B	1	$1 + 3 = 4$	$(1 + 3) \text{ MOD } 26 = 4$	E
...	...	...	...	...
X	23	$23 + 3 = 26$	$(23 + 3) \text{ MOD } 26 = 0$	A
Y	24	$24 + 3 = 27$	$(24 + 3) \text{ MOD } 26 = 1$	B
Z	25	$25 + 3 = 28$	$(25 + 3) \text{ MOD } 26 = 2$	C

Cifre a mensagem abaixo com K=3  
Texto plano: “**bob, i love you. alice**”

Texto cifrado: “**ere, I oryh brx. dolfh**”

# Exercício

- Decifre
    - ezi giwev qsvmxyvm xi wepyxerx
    - $K = 4$
  - Mensagem
    - ???
-

# Exercício

- Decifre

- ezi giwev qsvmxyvm xi wepyxerx
  - K = 4

- Mensagem

- Ave Cesar Morituri Te Salutant*



# Criptografia de chave simétrica

**Código de substituição:** substituindo uma coisa por outra

- Código monoalfabético: substituir uma letra por outra

texto aberto: abcdefghijklmnopqrstuvwxyz  
↓                    ↓  
Chave : mnbvvcxzdasdfghjklpoiuytrewq

Ex.:

texto aberto: bob. i love you. alice

texto cifrado: nkn. s gktc wky. mgsbc

P.:\_Quão difícil é quebrar esse código simples?

- Força bruta (quantas tentativas?)
- Outro método?

# Exercício – Cifra Monoalfabética

- Resolva a cifra monoalfabética a seguir.
- WV QHSGMZTHADSA, IVA QSDHA VZXZAUDABWMSQA WR  
IVA QSDHA CW JIBJMSMISQAZ ZXCW QACA UWMHA CZ  
MWKMZ WV QUAHZ WR JIBJMSMISCA GZH IVA ZIMHA  
UWMHA XZ MWKMZ QSDHACZ, CW DZHVA QZXJMAXMW.  
WJMW MSGZ CW QSDHA WR WKMHWVAVWXMW  
NIUXWHANWU GAHA AGUSQAQZWJ
- Chave: *a-b-q-c-w-d-t-r-s-f-e-u-v-x-z-g-y-h-j-m-i-n-l-k-o-p*
- Força Bruta?
  - Comente a fragilidade desse sistema criptográfico, apesar do espaço de chaves ser da ordem 26!.

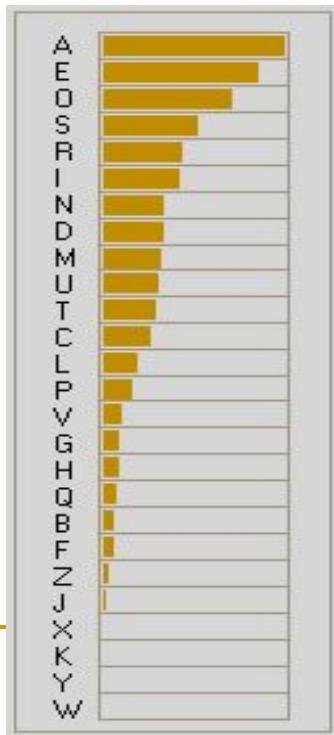
# Exercício – Cifra Monoalfabética

- EM CRIPTOGRAFIA, UMA CIFRA MONOALFABETICA EH UMA CIFRA DE SUBSTITUICAO ONDE CADA LETRA DO TEXTO EM CLARO EH SUBSTITUIDO POR UMA OUTRA LETRA NO TEXTO CIFRADO, DE FORMA CONSTANTE. ESTE TIPO DE CIFRA EH EXTREMAMENTE VULNERAVEL PARA APLICACOES



# Exercício – Cifra Monoalfabética

- WV QHSGMZTHADSA, IVA QSDHA VZXZAUDABWMSQA WR  
IVA QSDHA CW JIBJMSMISQAZ ZXCW QACA UWMHA CZ  
MWKMZ WV QUAHZ WR JIBJMSMISCA GZH IVA ZIMHA  
UWMHA XZ MWKMZ QSDHACZ, CW DZHVA QZXJMAXMW.  
WJMW MSGZ CW QSDHA WR WKMHWVAVWXMW  
NIUXWHANWU GAHA AGUSQAQZWJ



DE	1.76	QUE	0.96
RA	1.67	ENT	0.56
ES	1.65	COM	0.47
OS	1.51	NTE	0.44
AS	1.49	EST	0.34
DO	1.41	AVA	0.34
AR	1.33	ARA	0.33
CO	1.31	ADO	0.33
EN	1.23	PAR	0.30
QU	1.20	NDO	0.30
ER	1.18	NAO	0.30
DA	1.17	ERA	0.30
RE	1.14	AND	0.30
CA	1.11	UMA	0.28
TA	1.10	STA	0.28
SE	1.08	RES	0.27
NT	1.08	MEN	0.27
MA	1.06	CON	0.27
UE	1.05	DOS	0.25
TE	1.05	ANT	0.25

# Cenários de Quebra

- Ataque exclusivo ao texto cifrado
    - Análise estatística
  - Ataque com texto aberto conhecido
    - É conhecido trechos em claro do texto cifrado
  - Ataque com texto aberto escolhido
    - A mensagem a ser cifrada pode ser escolhida pelo intruso
    - “*The quick fox jumps over the lazy brown dog*”
      - Cifra polialfabética
-

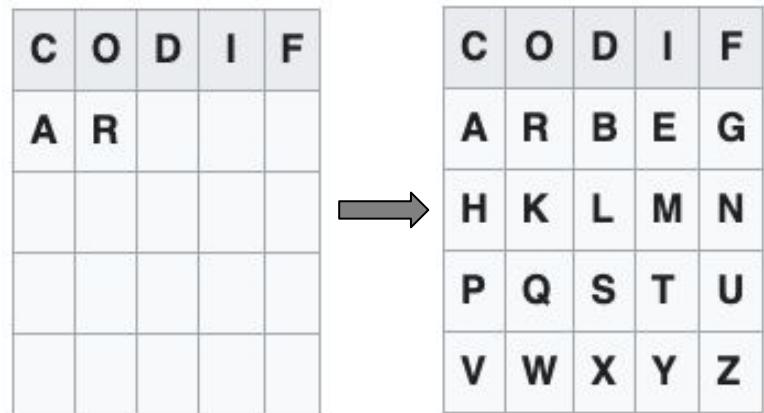
# Cifra Polialfabética

- Conjuntos de substituições monoalfabéticas
- A chave indica qual cifra monoalfabética utilizar
- Ex:
  - Plano: Testando
    - k: chave
  - Cifra: VLSOEPKO

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	
0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	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	
2	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	
3	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	
4	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	
5	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	
6	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	
7	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	
8	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	
9	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	
10	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	
11	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	
12	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	
13	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	
14	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	
15	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	
16	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	
17	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	
18	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	
19	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	
20	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	
21	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	
22	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	
23	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	
24	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	
25	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	

# Cifra de Playfair

- Passo 1: Escolha da mensagem e da senha.
  - M: OS INIMIGOS ESTÃO AVANÇANDO
  - K: Codificar
- Passo 2: Criação da matriz de cifragem
- Passo 3: Formatação da mensagem.
  - Letras agrupadas em pares
  - Letras maiúsculas
  - Remover sinais gráficos
  - Se repetir ou faltar, usar X ou Z
    - ex: CARRO → CA RX RO
    - ex: NUNCA → NU NC AX



- Passo 4: Cifragem
  - Regras aos pares:
    - **Linha - direita**
      - DO → ID
    - **Coluna - abaixo**
      - IM → ET
    - **Quadrado - horizontal**
      - OS → DQ

# Cifra de Playfair

- Exemplo
  - M: OS INIMIGOS ESTÃO AVANÇANDO
  - K: Codificar
  - C: DQFMETFEDQBTEPCRCHHFGHID

C	O	D	I	F
A	R			

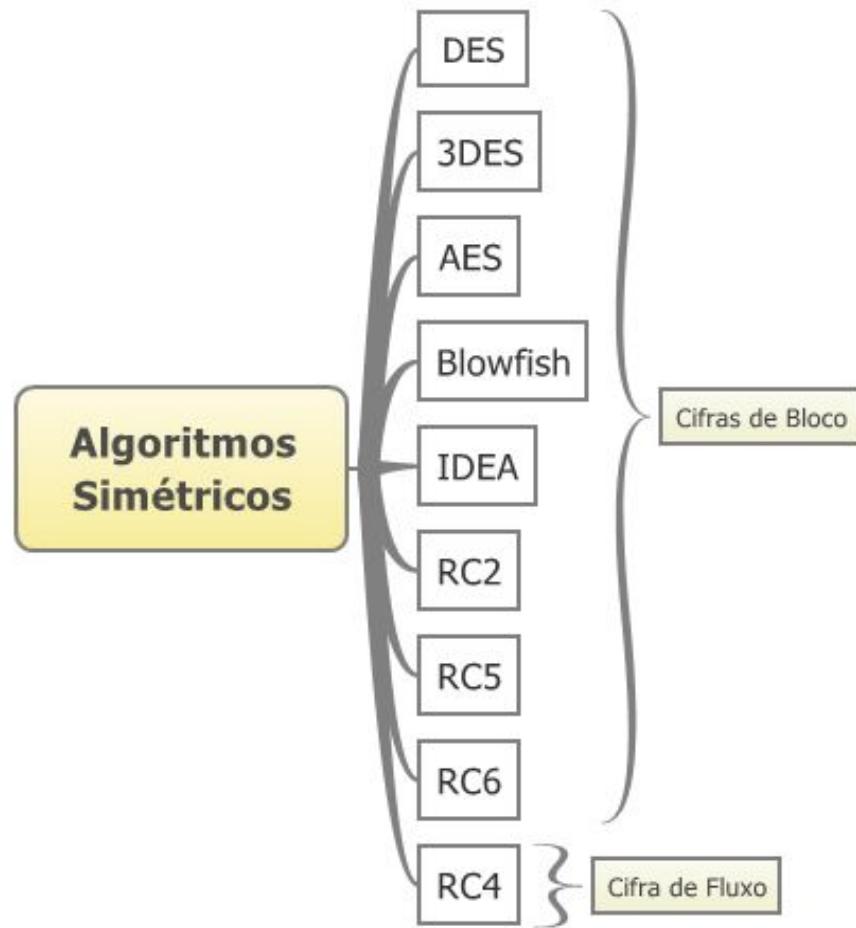


C	O	D	I	F
A	R	B	E	G
H	K	L	M	N
P	Q	S	T	U
V	W	X	Y	Z

## Passo 4: Cifragem

- Regras aos pares:
  - Linha - direita
    - DO → ID
  - Coluna - abaixo
    - IM → ET
  - Quadrado - horizontal
    - OS → DQ

# Criptografia de chave simétrica

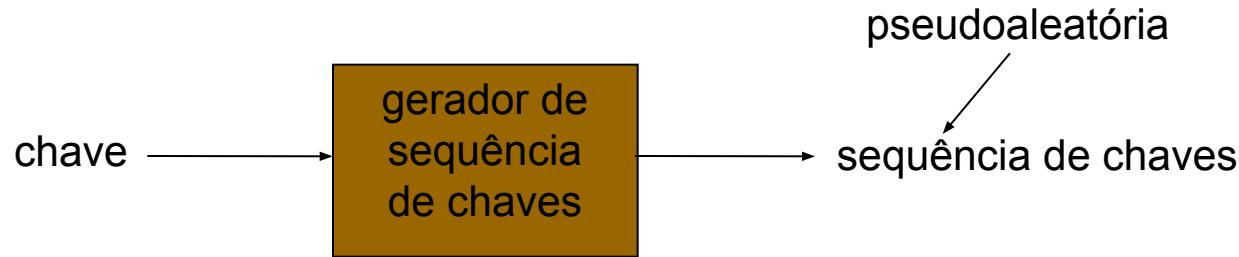


# Dois tipos de cifras simétricas

- Cifras de fluxo
  - criptografam um bit por vez
- Cifras de bloco
  - Quebram a mensagem de texto aberto em blocos de mesmo tamanho
  - Criptografam cada bloco como uma unidade



# Cifras de fluxo



- Combinam cada bit da sequência de chaves com bit de texto aberto para obter bit de texto cifrado
  - $m(i) = i^{\text{o}} \text{ bit da mensagem}$
  - $ks(i) = i^{\text{o}} \text{ bit da sequência de chaves}$
  - $c(i) = i^{\text{o}} \text{ bit do texto cifrado}$
  - $c(i) = ks(i) \oplus m(i)$  ( $\oplus = \text{OR exclusivo, ou XOR}$ )
  - $m(i) = ks(i) \oplus c(i)$
-

# Cifra de fluxo RC4

- RC4 é uma cifra de fluxo popular
  - chave pode ter de 1 a 256 bytes
  - usada por WEP para 802.11
  - pode ser usada em SSL



## Cifras de bloco

- Mensagem a ser criptografada é processada em blocos de k bits (p. e., blocos de 64 bits).
- Mapeamento 1-para-1 é usado para mapear bloco de k bits de texto aberto para bloco de k bits de texto cifrado

Exemplo com k = 3:

<u>entrada</u>	<u>saída</u>
000	110
001	111
010	101
011	100

<u>entrada</u>	<u>saída</u>
100	011
101	010
110	000
111	001

Qual é o texto cifrado para 010110001111 ?

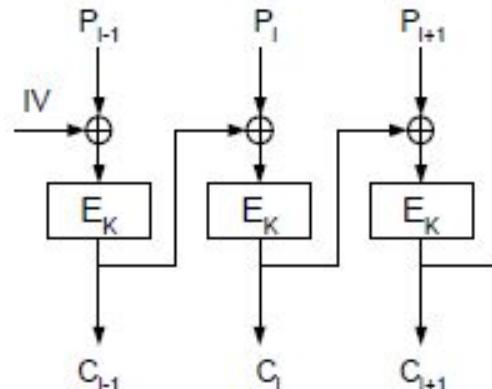
---

# Criptografando uma mensagem grande

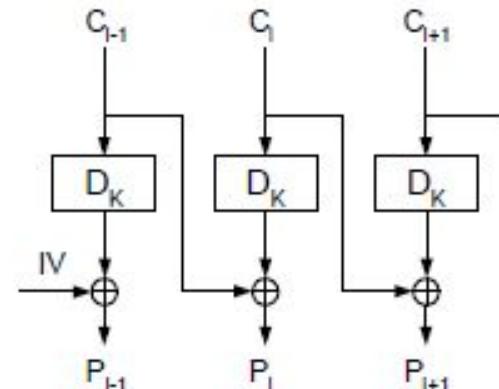
- Por que não apenas quebra a mensagem em blocos de 64 bits e criptografar cada bloco separadamente?
  - Se o mesmo bloco de texto aberto aparecer duas vezes, gerará o mesmo texto cifrado.
- Que tal:
  - gerar número aleatório de 64 bits  $r(i)$  para cada bloco de texto aberto  $m(i)$
  - calcular  $c(i) = K_S(m(i) \oplus r(i))$
  - transmitir  $c(i), r(i), i = 1, 2, \dots$
  - no destinatário:  $m(i) = K_S(c(i)) \oplus r(i)$
  - problema:
    - ineficaz, precisa enviar  $c(i)$  e  $r(i)$

# Cipher Block Chaining (CBC)

- CBC gera seus próprios números aleatórios
  - faça a criptografia do bloco atual depender do resultado do bloco anterior
  - $c(i) = K_s(m(i) \oplus c(i-1))$
  - $m(i) = K_s(c(i)) \oplus c(i-1)$
- Como criptografamos o primeiro bloco?
  - vetor de inicialização (IV): bloco aleatório =  $c(0)$
- mude IV para cada mensagem (ou sessão)
  - garante que, ainda que a mesma mensagem seja enviada repetidamente, o texto cifrado será completamente diferente a cada vez



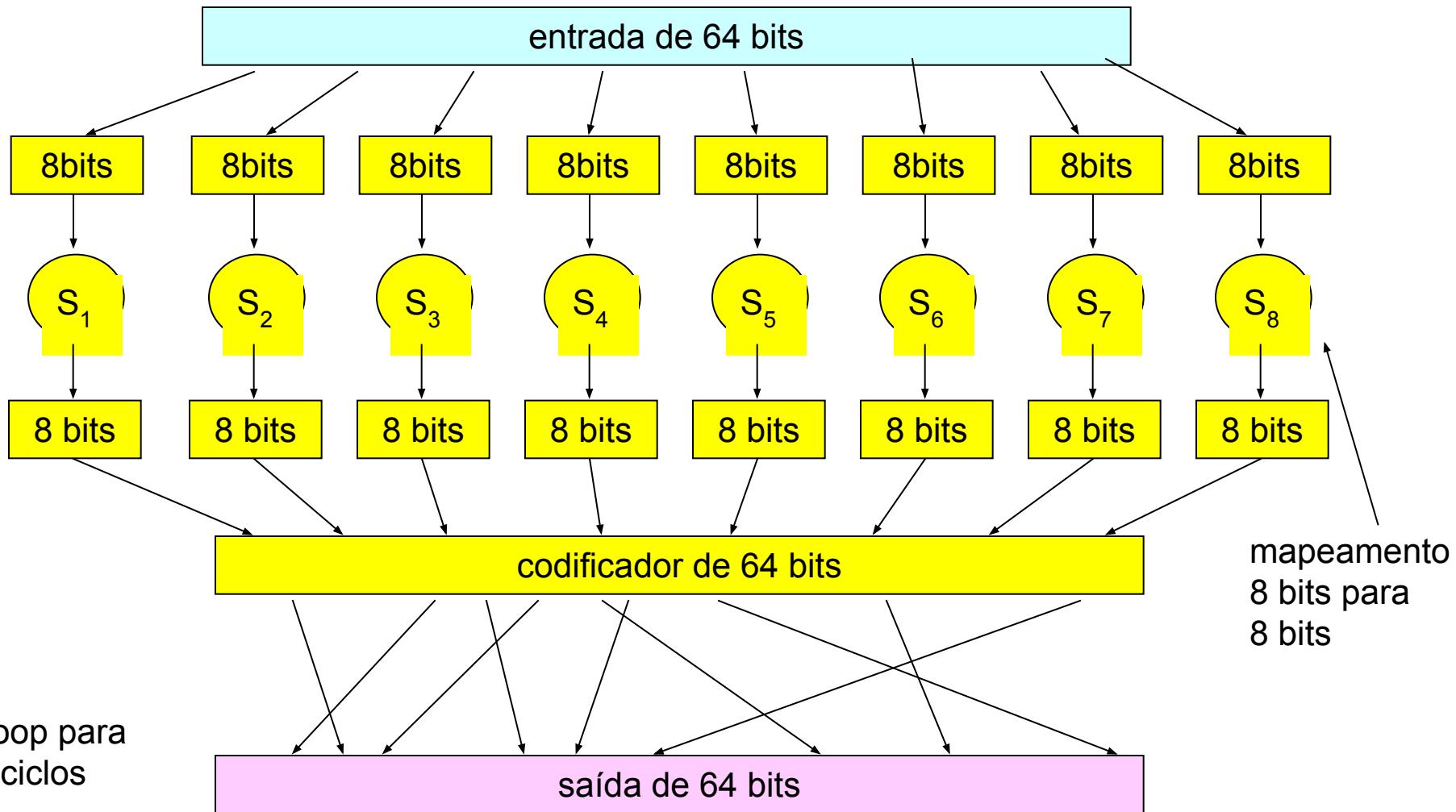
cifração CBC



decifração CBC

- Quantos mapeamentos existem para  $k = 3$ ?
    - Quantas entradas de 3 bits?
    - Quantas permutações das entradas de 3 bits?
    - **Resposta: 40.320 ; não muitas!**
  - Em geral,  $2^k!$  mapeamentos; imenso para  $k = 64$
  - Problema:
    - Requer tabela com  $2^{64}$  entradas, cada entrada com 64 bits
  - Solução
    - use função que simula tabela permutada aleatoriamente
-

# Função de protótipo



# O que acontece no protótipo?

- se um único ciclo, então um bit de entrada afeta no máximo 8 bits de saída.
  - no 2º ciclo, os 8 bits afetados são espalhados e inseridos em múltiplas caixas de substituição.
  - quantos ciclos?
    - quantas vezes você precisa misturar cartas?
    - torna-se menos eficiente quando  $n$  aumenta
-

# Difusão e Confusão

## ■ Difusão



*sem difusão*



*com difusão*



## ■ Confusão

- Dificulta o relacionamento entre o texto cifrado e a chave
-

# *One Time Pad (OTP) – Cifra de Chave Única*

- O *One Time Pad* em dados binários utiliza uma chave gerada de forma verdadeiramente aleatória (nunca reutilizada), do tamanho do texto.
  - A operação de cifração consiste na operação XOR bit a bit do texto em claro com a chave.
  - Neste método de cifração, a busca exaustiva da chave não é eficaz, uma vez que todos os possíveis textos são gerados.
-

# *One Time Pad (OTP)*

<b>texto cifrado</b>	<b>chave de busca exaustiva</b>	<b>texto em claro analisado</b>
0101	0000	0101
0101	0001	0100
0101	0010	0111
0101	0011	0110
0101	0100	0001
0101	0101	0000
0101	0110	0011
0101	0111	0010
0101	1000	1101
0101	1001	1100
0101	1010	1111
0101	1011	1110
0101	1100	1001
0101	1101	1000
0101	1110	1011
0101	1111	1010

# One Time Pad (OTP) – Cifra de Chave Única

Mensagem 1	<b>1001001 0100000 1101100 1101111 1110110 1100101 0100000 1111001 1101111 1110101 0101110</b>
K 1:	<b>1010010 1001011 1110010 1010101 1010010 1100011 0001011 0101010 1010111 1100110 0101011</b>
Texto cifrado:	<b>0011011 1101011 0011110 0111010 0100100 0000110 0101011 1010011 0111000 0010011 0000101</b>
K 2:	<b>1011110 0000111 1101000 1010011 1010111 0100110 1000111 0111010 1001110 1110110 1110110</b>
Texto claro 2:	<b>1000101 1101100 1110110 1101001 1110011 0100000 1101100 1101001 1110110 1100101 1110011</b>

## Tentativa de quebra

- Mensagem original:
  - "I LOVE YOU."
- Texto claro obtido com k2
  - "ELVIS LIVES"

Hex	Value																		
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p	80	~	90	g
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q	81	^	91	z
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r	82	;	92	o
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s	83	*	93	l
04	EOT	14	DC4	24	\$	34	4	44	D	54	T	64	d	74	t	84	-	94	u
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u	85	,	95	o
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v	86	=	96	u
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w	87	:	97	o
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x	88	)	98	u
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y	89	,	99	z
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z	8A	,	9A	~
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{	8B	;	9B	^
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C		8C	,	9C	~
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}	8D	,	9D	~
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	-	8E	,	9E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL	8F	,	9F	~

# *One Time Pad* (OTP) – Cifra de Chave Única

## ■ Exercício

- Explique porque o algoritmo *One Time Pad* é considerado uma cifra perfeita (inquebrável).
  - Faça o ataque força bruta do texto cifrado: 1001
  - Qual chave gerou esse texto cifrado?



# DES: criptografia com chave simétrica

## DES: Data encryption standard

- Padrão de criptografia dos Estados Unidos [NIST 1993]
- Chave simétrica de 56 bits, 64 bits de texto aberto na entrada
- Quão seguro é o padrão DES?
  - DES Challenge: uma frase criptografada com chave de 56 bits (“strong cryptography makes the world a safer place”) foi decodificada pelo método da força bruta em 4 meses
    - Não há ataque mais curto conhecido
- Tornando o DES mais seguro
  - Use três chaves em seqüência (3-DES) sobre cada dado



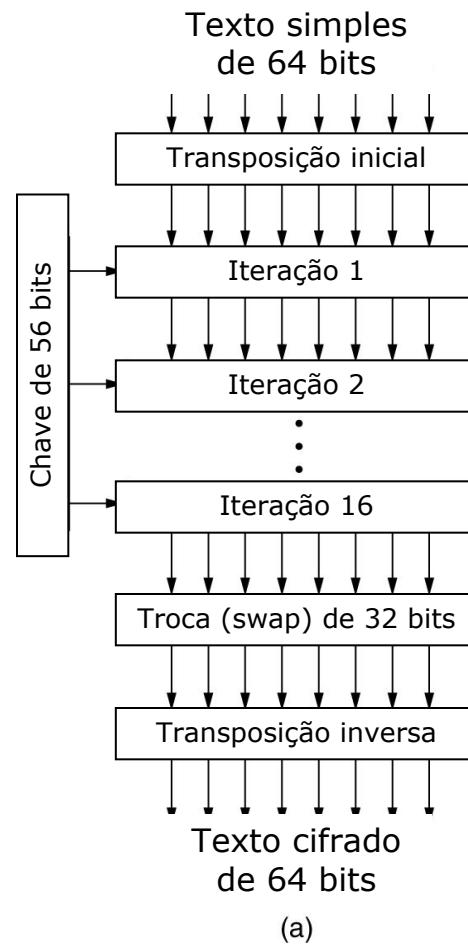
# Criptografia de chave simétrica: DES

## Operação do DES

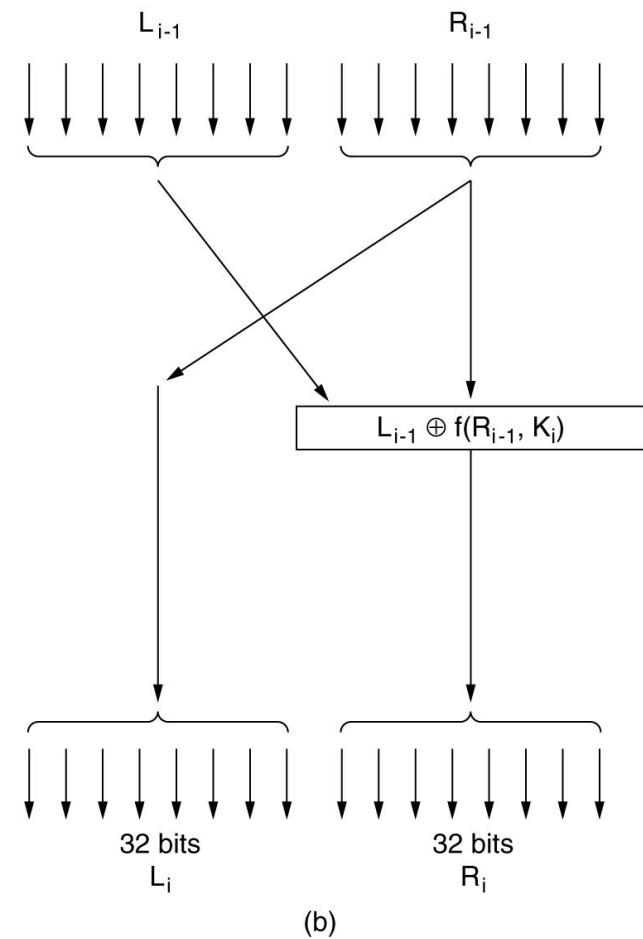
Permutação inicial

16 rodadas  
idênticas de função  
de substituição,  
cada uma usando  
uma diferente  
chave de 48 bits

Permutação final

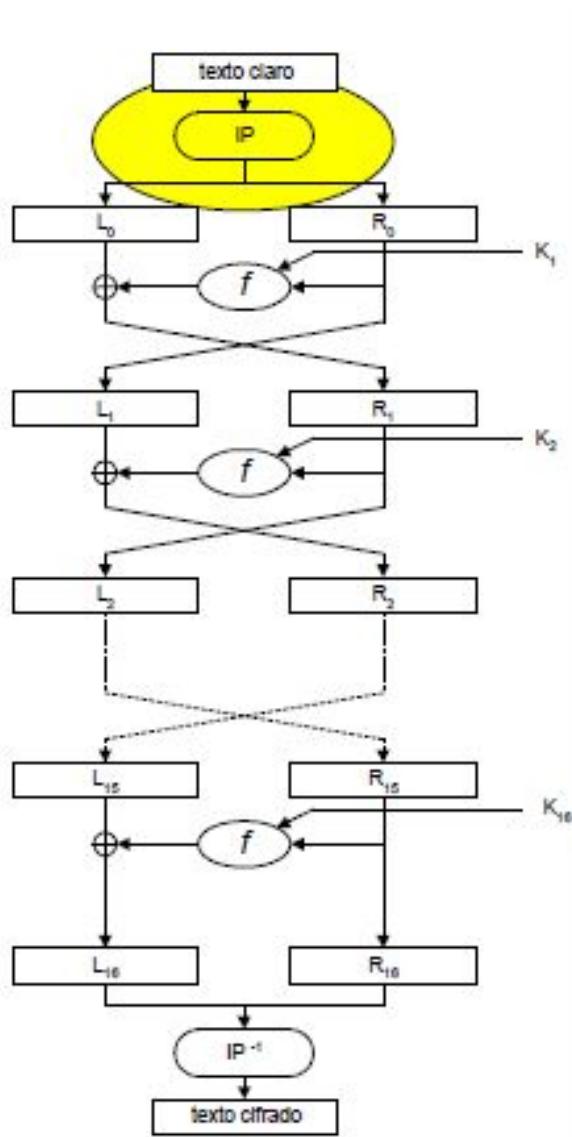


(a)



(b)

# Permutação Inicial

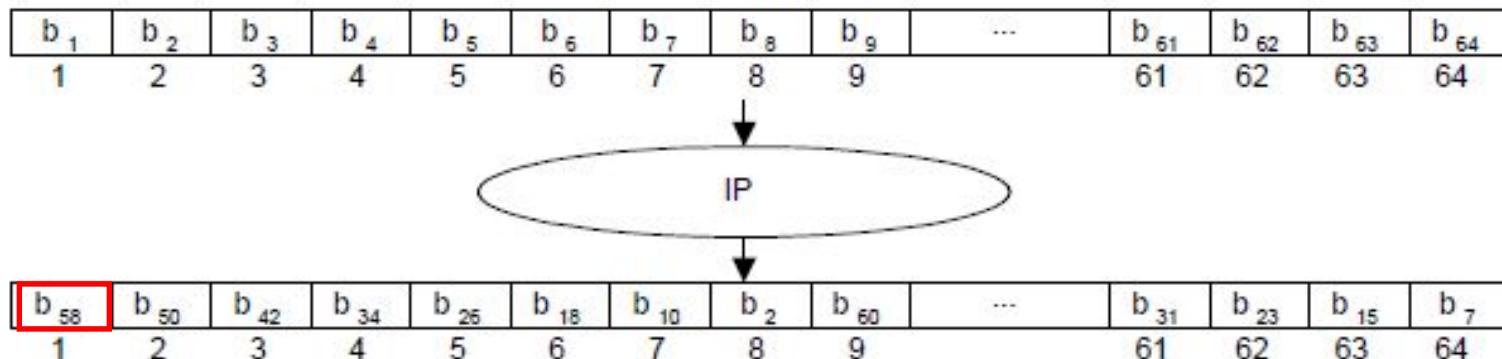


# Permutação Inicial - Matriz de Permutação

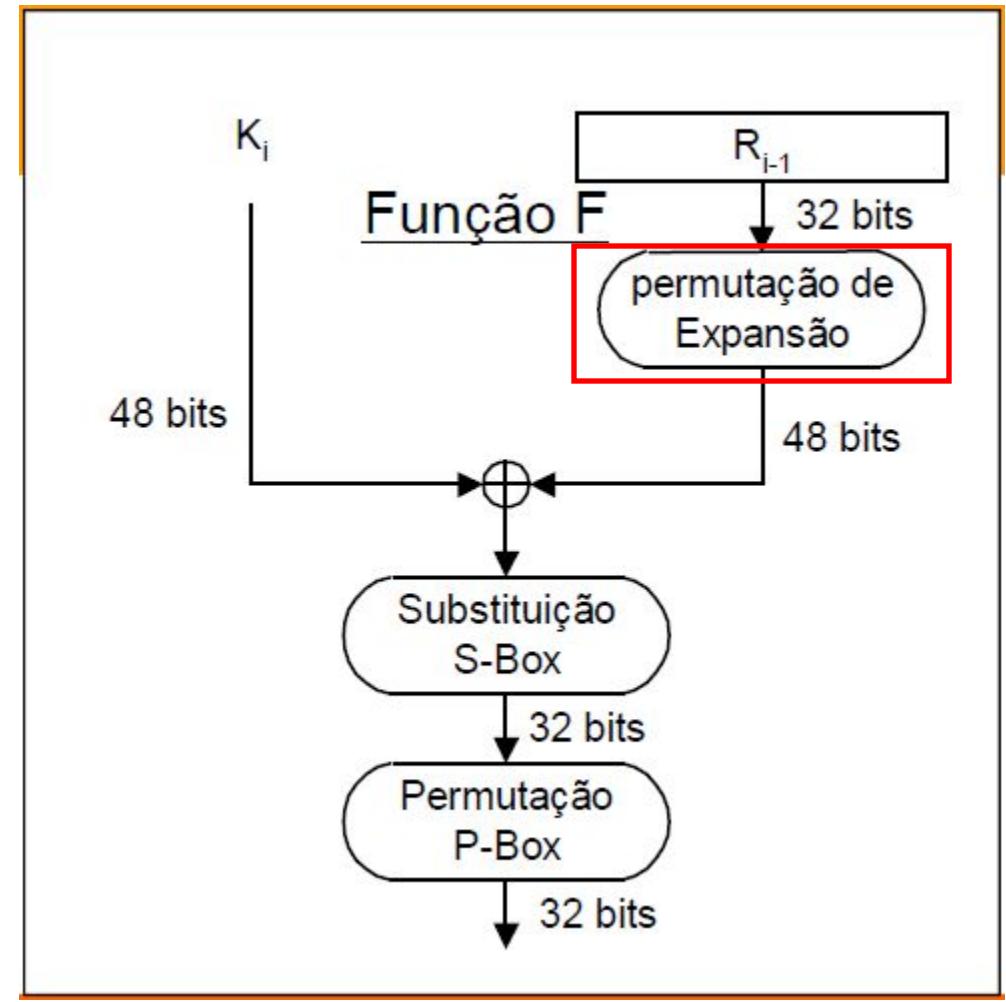
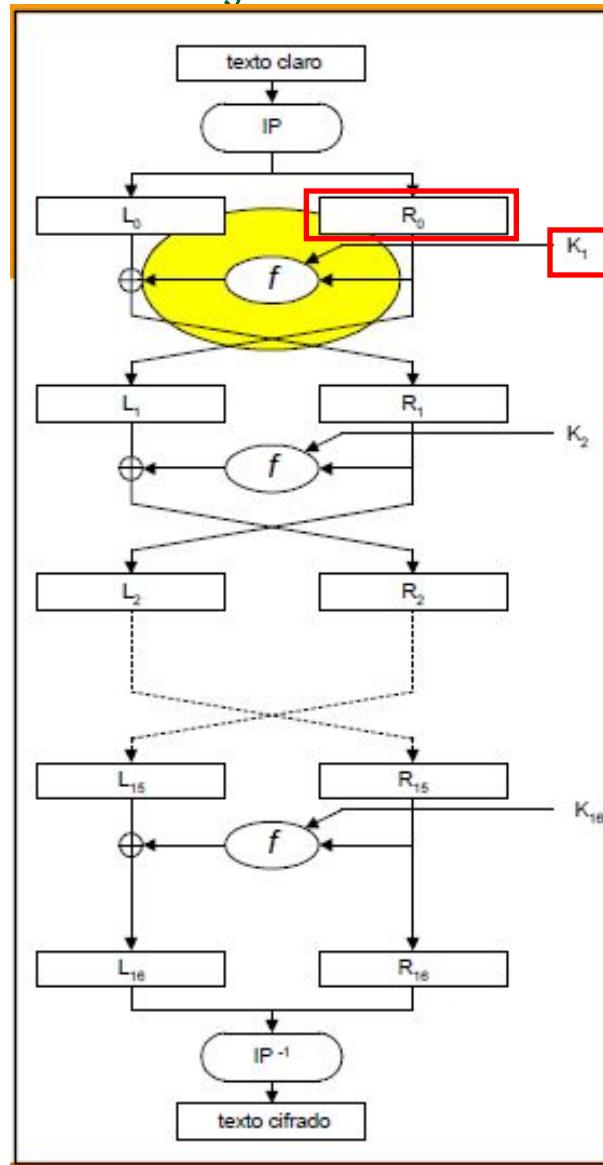
Permutação inicial (IP)

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

OU SEJA



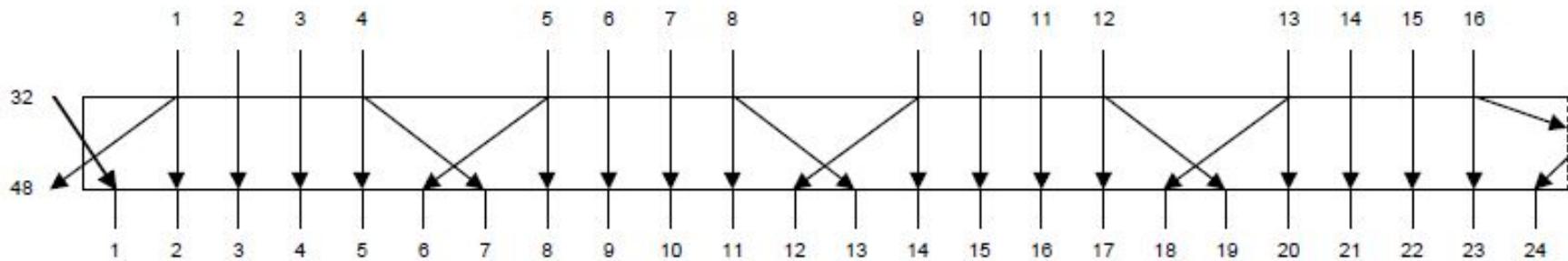
# Função de Substituição - Feistel



# Função de Substituição – Feistel Permutação de Expansão

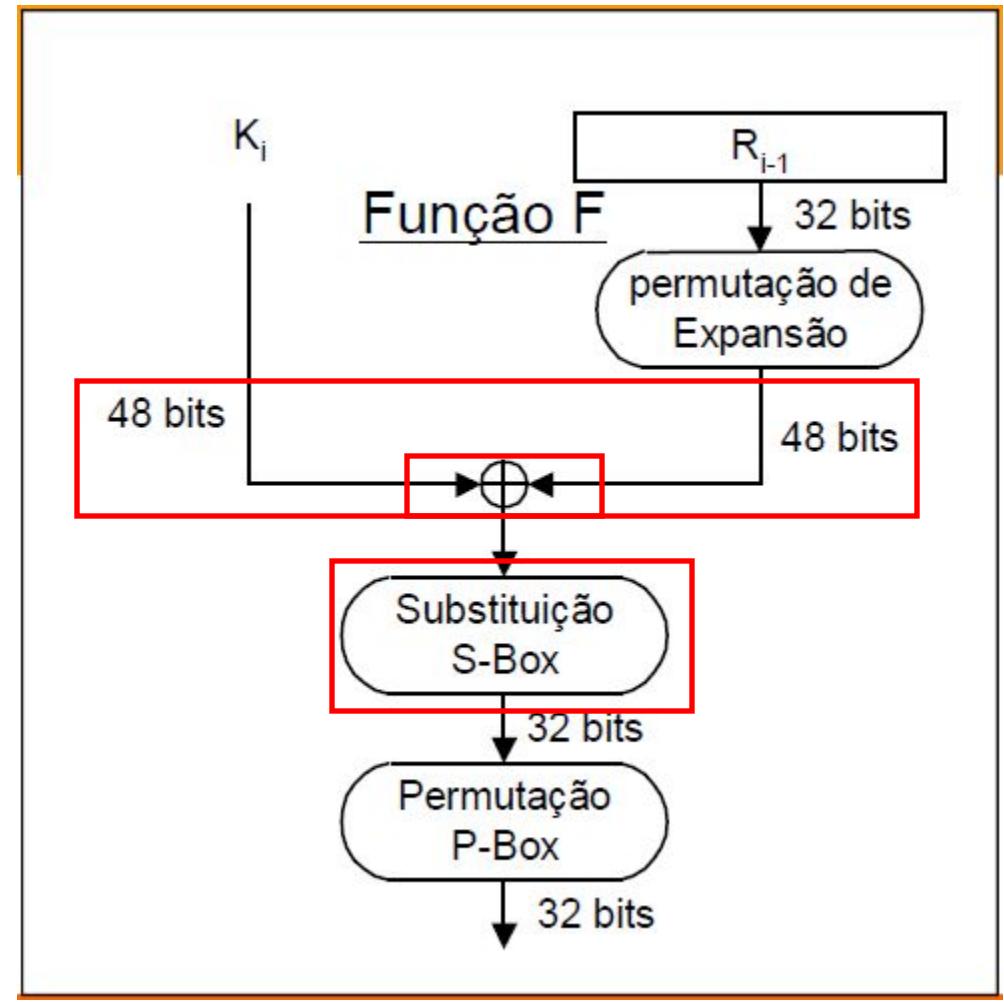
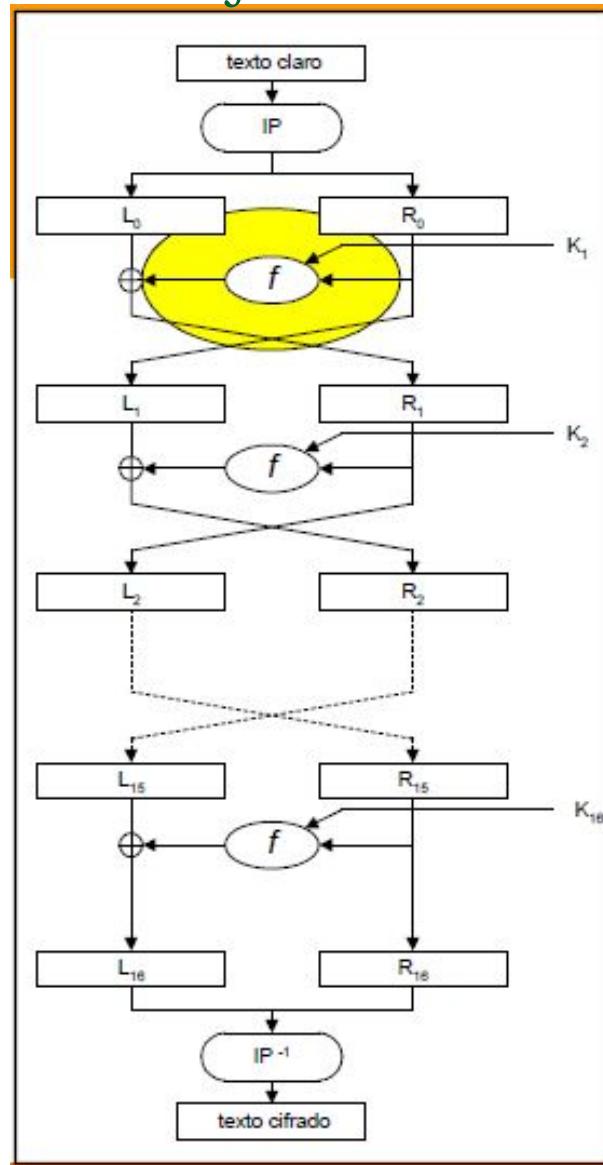
32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

OU SEJA



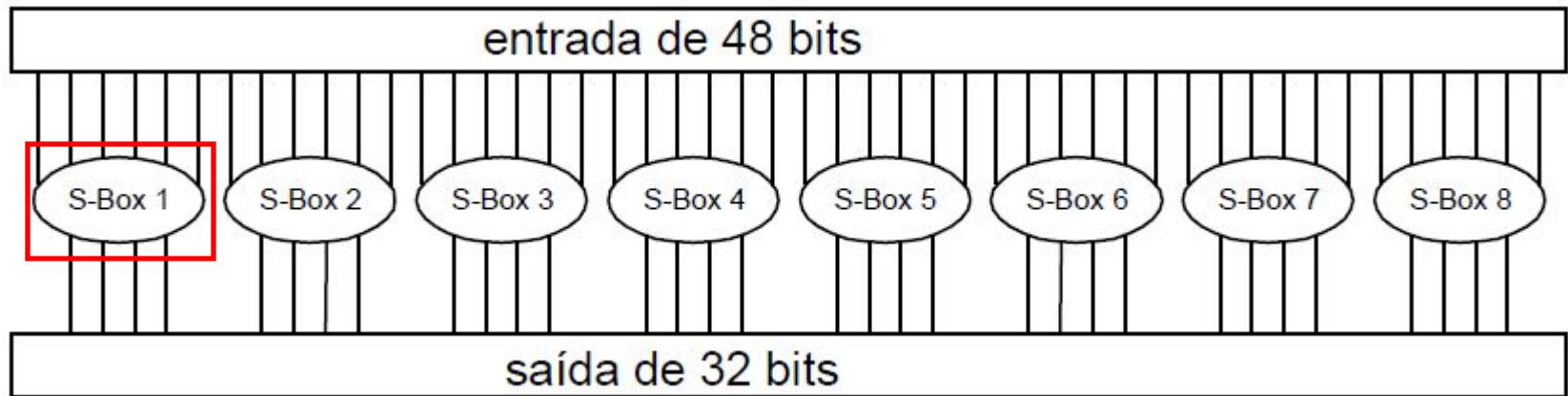
---

# Função de Substituição - Feistel



# Função de Substituição – Feistel

## Substituição S-Box



# Função de Substituição – Feistel

## Substituição S-Box

S-box1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Entrada na base 2 – 6bits

Posições 123456

Bits 1 e 6 = linha

Bits 2,3,4 e 5 = coluna

Exemplo

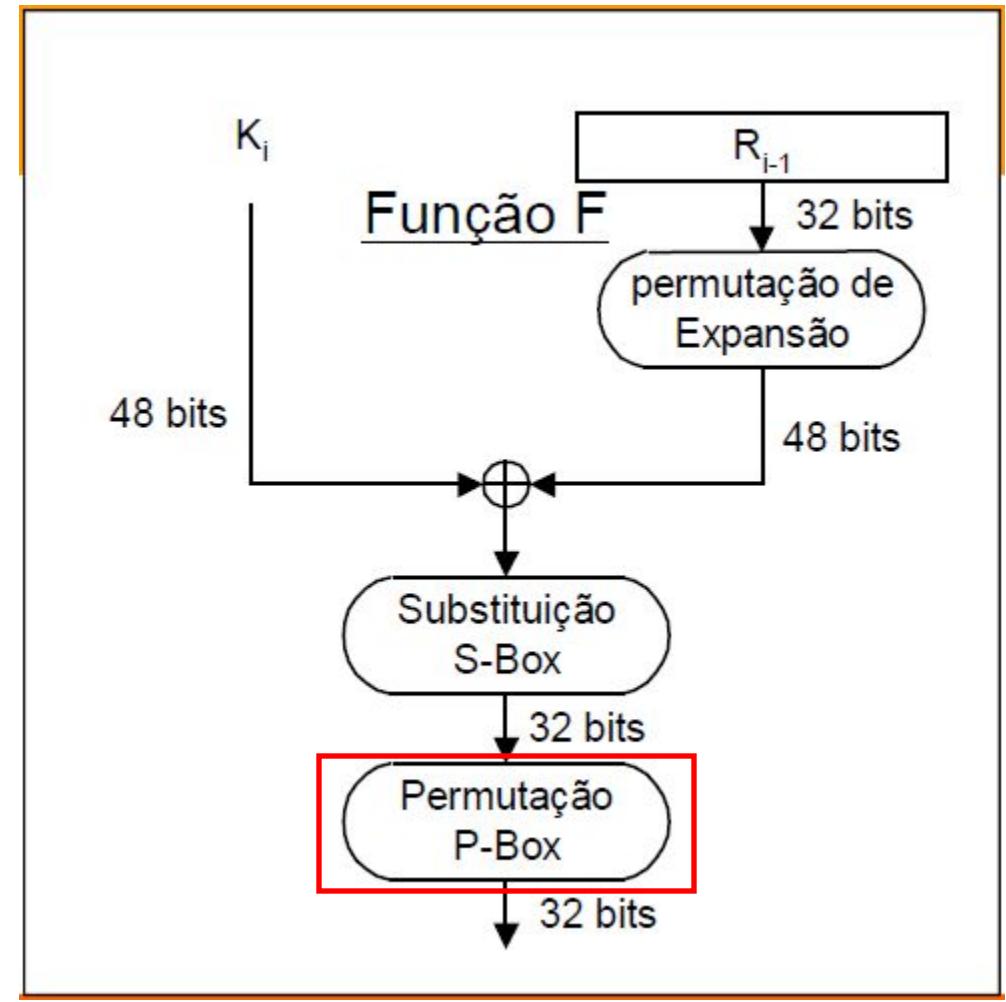
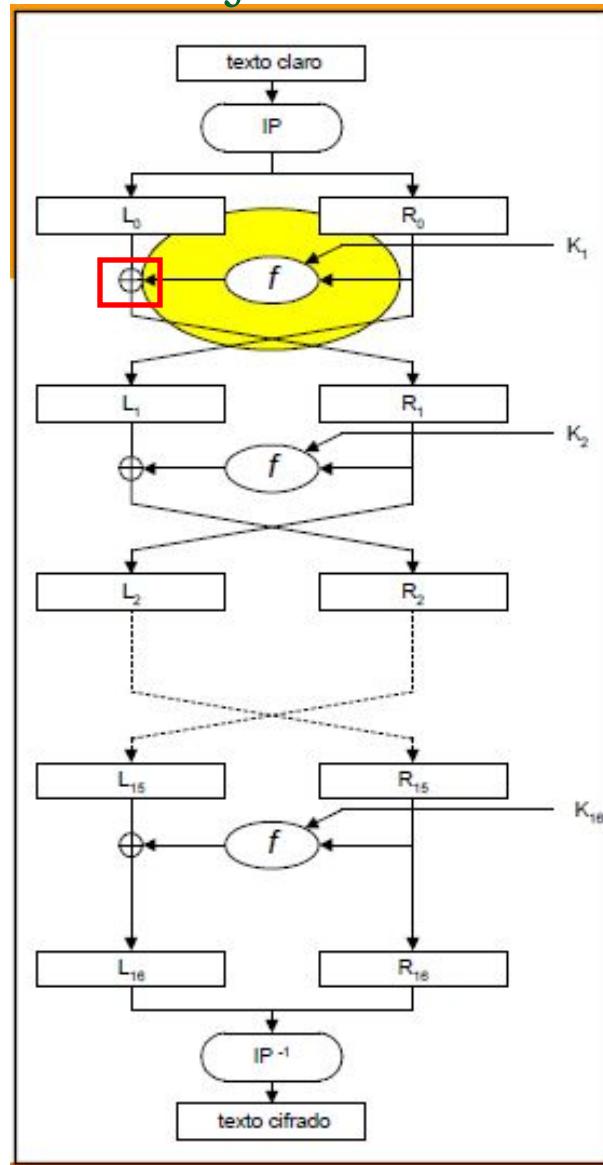
011101

01 = linha 1

1110 = coluna 14

Saída: 3 = 0011

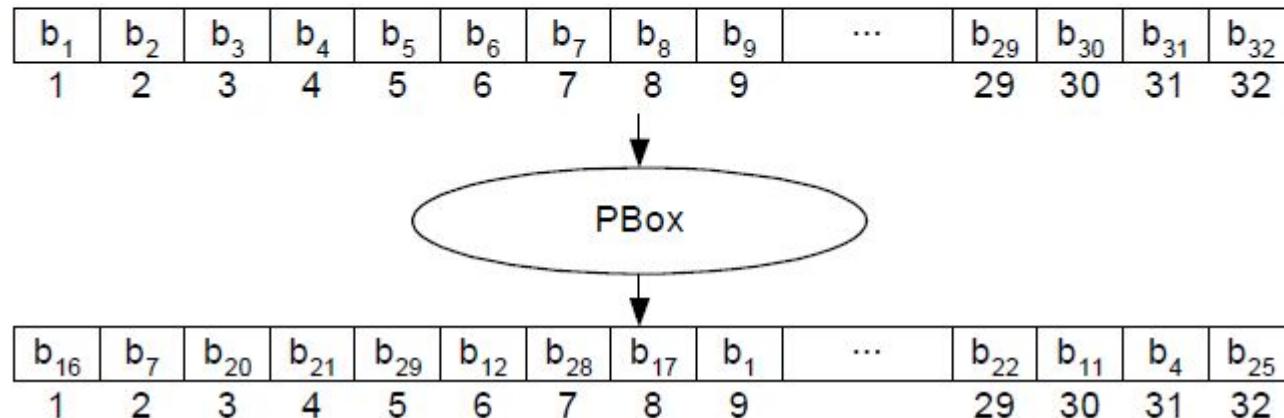
# Função de Substituição - Feistel



# Função de Substituição – Feistel Permutação P Box

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

OU SEJA



# Permutação Final ( $IP^{-1}$ )

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

OU SEJA

$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$	...	$b_{61}$	$b_{62}$	$b_{63}$	$b_{64}$
1	2	3	4	5	6	7	8	9		61	62	63	64

$IP^{-1}$

$b_{40}$	$b_8$	$b_{48}$	$b_{16}$	$b_{56}$	$b_{24}$	$b_{64}$	$b_{32}$	$b_{39}$	...	$b_{49}$	$b_{17}$	$b_{57}$	$b_{25}$
1	2	3	4	5	6	7	8	9		61	62	63	64

Prepara o texto cifrado para o processo inverso (Decifração)

# DES - Ilustrado

Mensagem

"Criptologia sempre NumaBoa"

Hexadecimal

43 72 69 70 74 6F 6C 6F

67 69 61 20 73 65 6D 70

72 65 20 4E 75 6D 61 42

6F 61 0D 0A

Mensagem

"Criptologia sempre NumaBoa"

Hexadecimal

43 72 69 70 74 6F 6C 6F

67 69 61 20 73 65 6D 70

72 65 20 4E 75 6D 61 42

6F 61 0D 0A 00 00 00 00

# DES - Ilustrado

## Mensagem

"Criptologia sempre NumaBoa"

## Hexadecimal

43 72 69 70 74 6F 6C 6F

67 69 61 20 73 65 6D 70

72 65 20 4E 75 6D 61 42

6F 61 0D 0A 00 00 00 00

## Cifrado

Chave 0E329232EA6D0D73

A1 BF 4C 8C 1F 44 6A 4C

CA 4D E4 28 6E DE 99 50

F5 59 66 2B B5 09 D9 3C

4B A7 70 FA E2 4B B3 C2

# DES - Ilustrado

Mensagem = 0123456789ABCDEF (hexadecimal)

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010  
1011 1100 1101 1110 1111

L = 0000 0001 0010 0011 0100 0101 0110 0111

R = 1000 1001 1010 1011 1100 1101 1110 1111

# DES - Ilustrado

**Chave = 133457799BCDFF1**

K = 00010011 00110100 01010111 01111001 10011011 10111100  
11011111 11110001

# DES - Ilustrado

A partir da chave de 56 bits, criar 16 subchaves de 48 bits

Permutação de 64 bits para 56 (remoção dos bits de paridade)

PC-1							
57	49	41	33	25	17	9	
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
63	55	47	39	31	23	15	
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

**K** = 00010011 00110100 01010111 01111001  
10011011 10111100 11011111 11110001

**K+** = 1111000 0110011 0010101 0101111  
0101010 1011001 1001111 0001111

C0 = 1111000 0110011 0010101 0101111

D0 = 0101010 1011001 1001111 0001111

# DES - Ilustrado

## Deslocamentos a esquerda

Dezesseis blocos  $\mathbf{C}_n$  e  $\mathbf{D}_n$ ,  $1 \leq n \leq 16$ .

Cada par de blocos  $\mathbf{C}_n$  e  $\mathbf{D}_n$  é formado pelo par anterior  $\mathbf{C}_{n-1}$  e  $\mathbf{D}_{n-1}$

Número de Iterações	Número de deslocamentos à esquerda
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

# DES - Ilustrado

## Deslocamentos a esquerda

$C_0 = 1111000011001100101010101111$

$D_0 = 0101010101100110011110001111$

$C_1 = 1110000110011001010101011111$

$D_1 = 1010101011001100111100011110$

$C_2 = 1100001100110010101010111111$

$D_2 = 0101010110011001111000111101$

$C_3 = 000011001100101010101111111$

$D_3 = 0101011001100111100011110101$

$C_4 = 0011001100101010101111111100$

$D_4 = 0101100110011110001111010101$

$C_5 = 1100110010101010111111110000$

$D_5 = 0110011001111000111101010101$

$C_6 = 00110010101011111111000011$

$D_6 = 1001100111100011110101010101$

$C_7 = 11001010101111111100001100$

$D_7 = 0110011110001111010101010110$

$C_8 = 00101010101111110000110011$

$D_8 = 1001111000111101010101011001$

$C_9 = 010101010111111100001100110$

$D_9 = 0011110001111010101010110011$

$C_{10} = 010101011111110000110011001$

$D_{10} = 1111000111101010101011001100$

$C_{11} = 010101111111000011001100101$

$D_{11} = 1100011110101010101100110011$

$C_{12} = 010111111100001100110010101$

$D_{12} = 0001111010101010110011001111$

$C_{13} = 011111110000110011001010101$

$D_{13} = 01111010101011001100111100$

$C_{14} = 1111111000011001100101010101$

$D_{14} = 11101010101100110011110001$

$C_{15} = 1111100001100110010101010111$

$D_{15} = 1010101010110011001111000111$

$C_{16} = 1111000011001100101010101111$

$D_{16} = 0101010101100110011110001111$

# DES - Ilustrado

Montar as 16 chaves com 48 bits

Aplicar a tabela PC2 aos 16 pares

**C<sub>n</sub>D<sub>n</sub>**

PC-2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

**C<sub>1</sub>** = 111000011001100101010101011111  
**D<sub>1</sub>** = 1010101011001100111100011110  
**K<sub>1</sub>** = 000110 110000 001011 101111  
111111 000111 000001 110010

**K<sub>2</sub>** = 011110 011010 111011 011001 110110 111100  
100111 100101  
**K<sub>3</sub>** = 010101 011111 110010 001010 010000 101100  
111110 011001  
**K<sub>4</sub>** = 011100 101010 110111 010110 110110 110011  
010100 011101  
**K<sub>5</sub>** = 011111 001110 110000 000111 111010 110101  
001110 101000  
**K<sub>6</sub>** = 011000 111010 010100 111110 010100 000111  
101100 101111  
**K<sub>7</sub>** = 111011 001000 010010 110111 111101 100001  
100010 111100  
**K<sub>8</sub>** = 111101 111000 101000 111010 110000 010011  
101111 111011  
**K<sub>9</sub>** = 111000 001101 101111 101011 111011 011110  
011110 000001  
**K<sub>10</sub>** = 101100 011111 001101 000111 101110 100100  
011001 001111  
**K<sub>11</sub>** = 001000 010101 111111 010011 110111 101101  
001110 000110  
**K<sub>12</sub>** = 011101 010111 000111 110101 100101 000110  
011111 101001  
**K<sub>13</sub>** = 100101 111100 010111 010001 111110 101011  
101001 000001  
**K<sub>14</sub>** = 010111 110100 001110 110111 111100 101110  
011100 111010  
**K<sub>15</sub>** = 101111 111001 000110 001101 001111 010011  
111100 001010  
**K<sub>16</sub>** = 110010 110011 110110 001011 000011 100001  
011111 110101

# DES - Ilustrado

Codificar cada bloco de 64 bits de dados

Aplicar a Permutação Inicial IP

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

**M** = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

**IP** = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

Dividir o bloco permutado em duas metades **L<sub>0</sub>** e **R<sub>0</sub>**

**L<sub>0</sub>** = 1100 1100 0000 0000 1100 1100 1111 1111

**R<sub>0</sub>** = 1111 0000 1010 1010 1111 0000 1010 1010

# DES - Ilustrado

## 16 rodadas

- Aplicar a função  $f(R_{n-1}, K_n)$  e fazer o XOR a metade esquerda
  - $L_n = R_{n-1}$
  - $R_n = L_{n-1} + f(R_{n-1}, K_n)$
- Para  $n = 1$  temos
  - $K_1 = 000110\ 110000\ 001011\ 101111\ 111111\ 000111\ 000001\ 110010$
  - $L_1 = R_0 = 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010$
  - $R_1 = L_0 + f(R_0, K_1)$

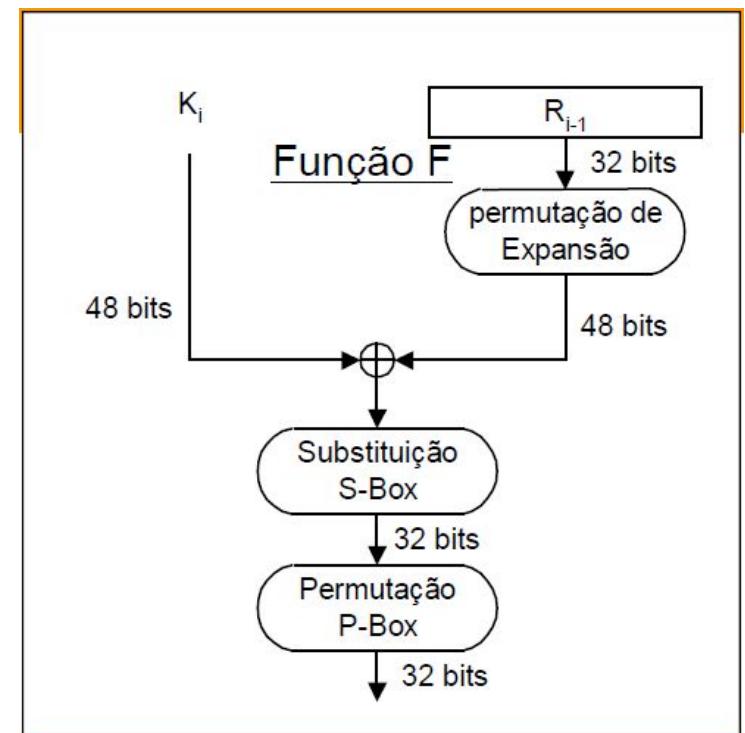
# DES - Ilustrado

funcão  $f(R_{n-1}, K_n)$

## Permutação de expansão

Tabela E de Seleção de bits

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



$R_0 = 1111 \ 0000 \ 1010 \ 1010 \ 1111 \ 0000 \ 1010 \ 1010$

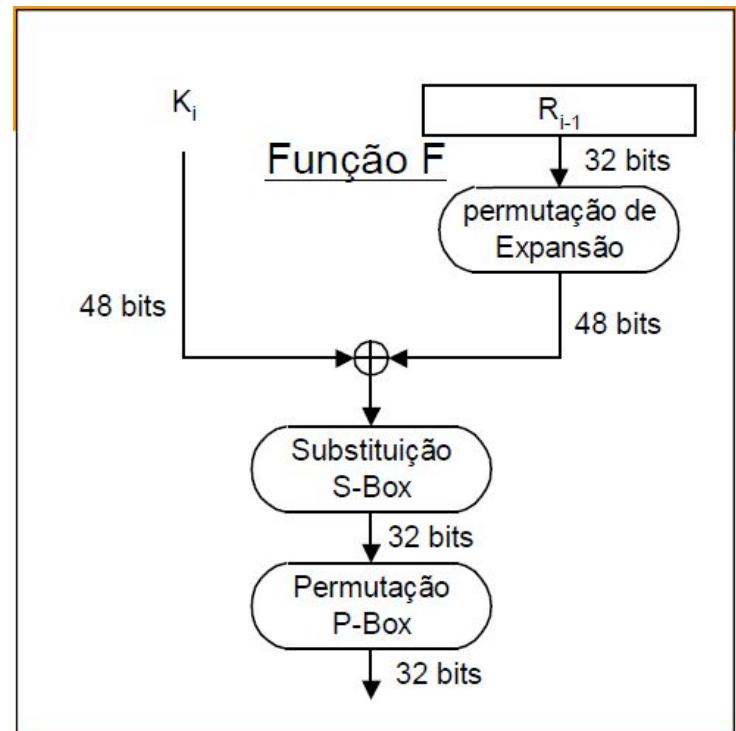
$E(R_0) = 011110 \ 100001 \ 010101 \ 010101 \ 011110 \ 100001 \ 010101 \ 010101$

# DES - Ilustrado

função  $f(R_{n-1}, K_n)$

XOR com a chave

$K_n + E(R_{n-1})$



$K_1 = 000110\ 110000\ 001011\ 101111\ 111111\ 000111\ 000001\ 110010$

$E(R_0) = 011110\ 100001\ 010101\ 010101\ 011110\ 100001\ 010101\ 010101$

$K_1 + E(R_0) = 011000\ 010001\ 011110\ 111010\ 100001\ 100110\ 010100\ 100111$

<http://www.numaboa.com.br/criptografia/bloco/313-des2>

# DES - Ilustrado

funcão  $f(R_{n-1}, K_n)$

Aplicar as "S Boxes"

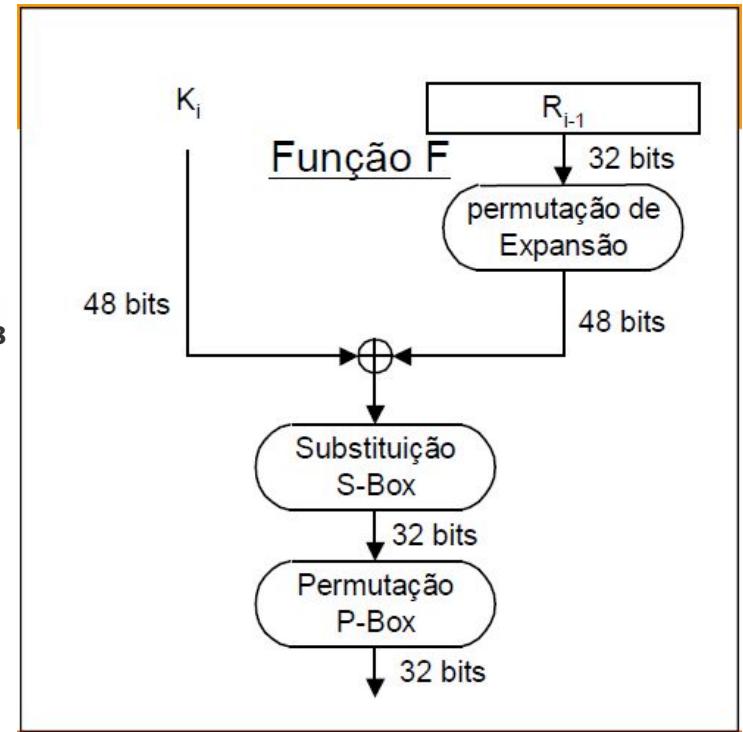
$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$

Bloco 1: 011000

linha: 00 = 0

coluna: 1100 = 12

valor: 5 = 0101



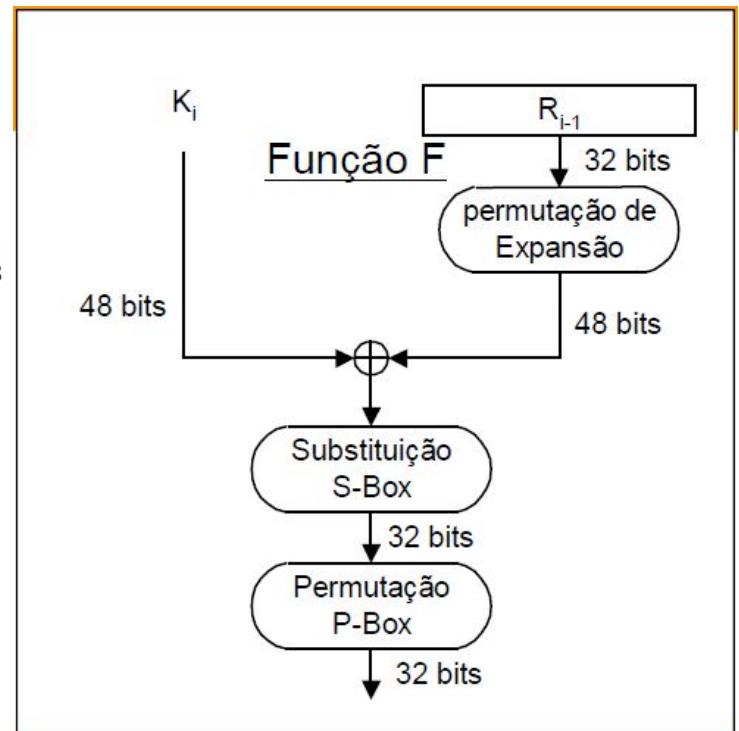
Função S1																
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0   14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1   0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
2   4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3   15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	

# DES - Ilustrado

funcão  $f(R_{n-1}, K_n)$

Aplicar as "S Boxes"

$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$



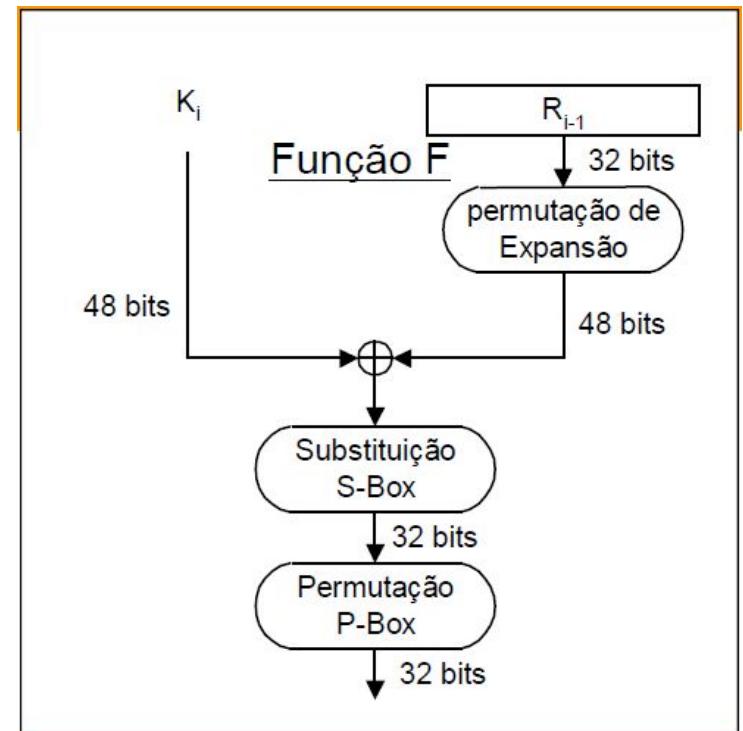
$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8) = 0101\ 1100\ 1000\ 0010\ 1011\ 0101\ 1001\ 0111$

# DES - Ilustrado

funcão  $f(R_{n-1}, K_n)$

Aplicar permutação P-Box

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25



$$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8) = 0101\ 1100\ 1000\ 0010\ 1011\ 0101\\ 1001\ 0111$$

obtemos

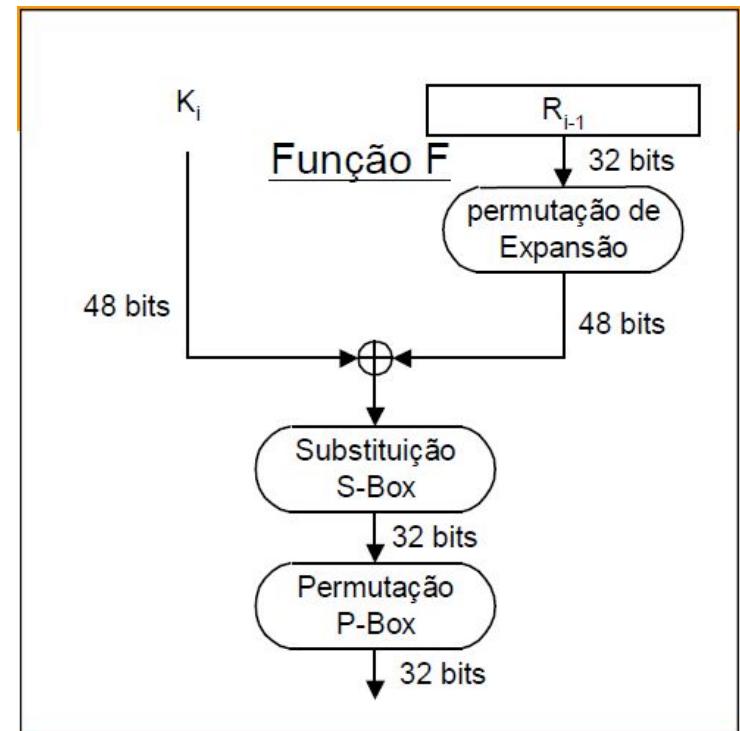
$$f = 0010\ 0011\ 0100\ 1010\ 1010\ 1001\ 1011\ 1011$$

# DES - Ilustrado

funcão  $f(R_{n-1}, K_n)$

Aplicar permutação P-Box

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25



$$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8) = 0101\ 1100\ 1000\ 0010\ 1011\ 0101\\ 1001\ 0111$$

obtemos

$$f = 0010\ 0011\ 0100\ 1010\ 1010\ 1001\ 1011\ 1011$$

# DES - Ilustrado

## 16 rodadas

- Aplicar a função  $f(R_{n-1}, K_n)$  e fazer o XOR a metade esquerda

- $L_n = R_{n-1}$

- $R_n = L_{n-1} + f(R_{n-1}, K_n)$

- Para  $n = 1$  temos

$$L_0 = \begin{array}{cccccccccc} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

$$f(R_0, K_1) = \begin{array}{cccccccccc} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{array}$$

$$R_1 = \begin{array}{cccccccccc} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}$$

# DES - Ilustrado

Fim das 16 rodadas

Swap das metades  $L_{16}$  e  $R_{16}$  para  $R_{16}L_{16}$

$L_{16}$  = 0100 0011 0100 0010 0011 0010 0011 0100

$R_{16}$  = 0000 1010 0100 1100 1101 1001 1001 0101

$R_{16}L_{16}$  = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100

Aplica a Permutação Inversa

IP <sup>-1</sup>							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

$IP^{-1}$  = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101

**M = 0123456789ABCDEF é C = 85E813540F0AB405.**



# AES: Padrão avançado de criptografia

- Novo (nov/2001) padrão do *NIST(National Institute of Standards and Technology)* para chaves simétricas, substituindo o DES
- Processa dados em blocos de 128 bits
- Chaves de 128, 192, ou 256 bits
- Decodificação por força bruta (tentar cada chave) leva 1 segundo no DES e 149 trilhões de anos no AES



# Criptografia Simétrica

## □ Vantagem

- Rapidez na criptografia e descriptografia da informação.

## □ Desvantagens

- A chave deve ser trocada entre as partes e armazenada de forma segura, o que nem sempre é fácil de garantir;
- A criptografia simétrica não garante a identidade de quem enviou ou recebeu a mensagem (autenticidade e não-repudiação).



# Sumário

- Fundamentos
  - Criptografia Simétrica
  - **Criptografia Assimétrica**
  - Integridade
  - Autenticação
  - Assinatura Digital e Certificação Digital
  - Distribuição de Chaves
  - Serviços de Rede
-

# Criptografia de chave pública

## *Chave simétrica*

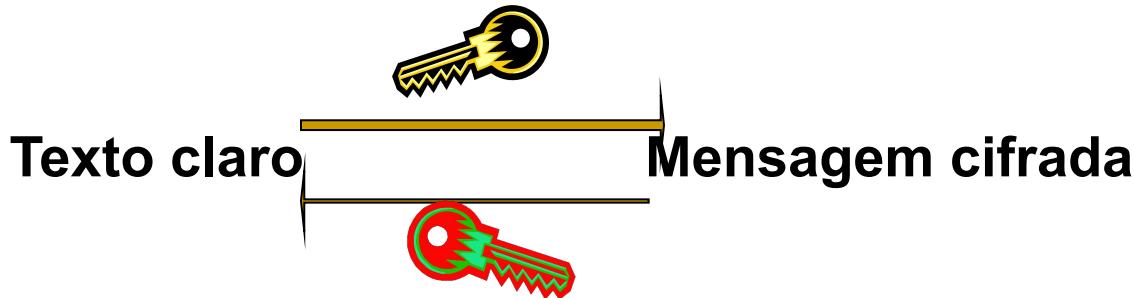
- Exige que o transmissor e o receptor compartilhem a chave secreta
- P.: Como combinar a chave inicialmente (especialmente no caso em que eles nunca se encontram)?



## *Chave pública*

- Abordagem radicalmente diferente [Diffie-Hellman76, RSA78]
- Transmissor e receptor **não** compartilham uma chave secreta
- A chave de criptografia é **pública** (conhecida por **todos**)
- Chave de decriptografia é **privada** (conhecida somente pelo receptor)

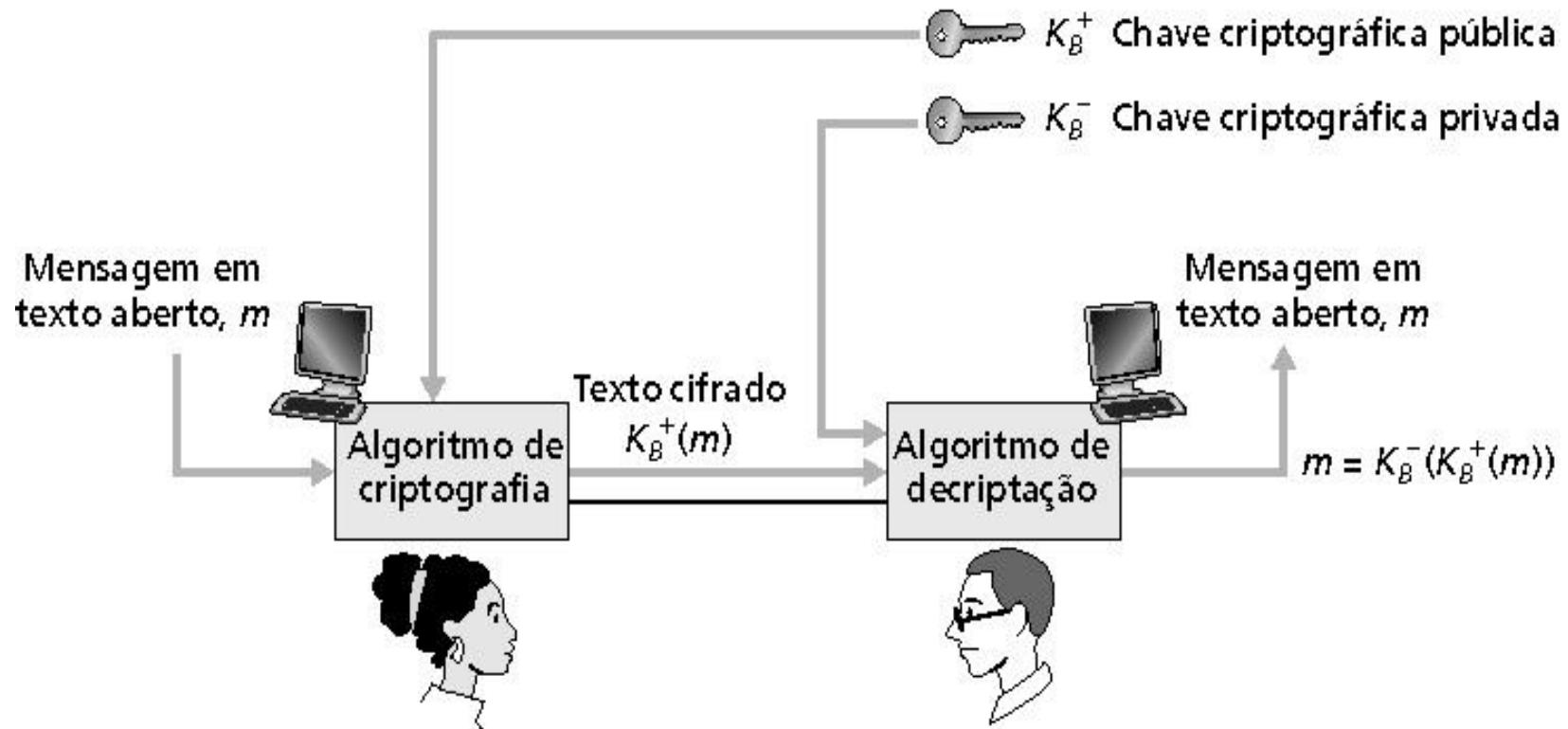
# Criptografia Assimétrica



- As chaves são sempre geradas aos pares: uma para cifrar e a sua correspondente para decifrar.
- A chave pública é divulgada, a chave privada é proprietária (normalmente não abandona o ambiente onde foi gerada).

**Uma chave é utilizada para “fechar o cadeado” e outra chave, diferente, mas relacionada à primeira, é utilizada para “abrir o cadeado”**

# Criptografia de chave pública (cont.)



# Criptografia Assimétrica

**Criptografia Assimétrica** - Não possui segredos compartilhados

## Criptografia

Para: Banco  
De: Affonso  
Data: 16, Abr, 2001  
Transferir R\$ 2,0 milhões da conta 254674-12 para a conta 071517-08  
Affonso



Algoritmo



\*> \*ql3\*UY  
#~00873/JDI  
c4(DH: IWB(883  
LKS9UI29as9%#@  
qw9vijhas9djerhp7  
(\*Y23k^wbvlqkwc  
zqw-\_89237xGyjdc  
Biskdue di7@94

## Descriptografia

\*> \*ql3\*UY  
#~00873/JDI  
c4(DH: IWB(883  
LKS9UI29as9%#@  
qw9vijhas9djerhp7  
(\*Y23k^wbvlqkwc  
zqw-\_89237xGyjdc  
Biskdue di7@94



Algoritmo



Para: Banco  
De: Affonso  
Data: 16, Abr, 2001  
Transferir R\$ 2,0 milhões da conta 254674-12 para a conta 071517-08  
Affonso

As duas chaves são relacionadas através de um processo matemático, usando funções unidirecionais para a codificação da informação.

# Criptografia Assimétrica

- **Algoritmos assimétricos** - permitem que **a chave de cifração possa ser tornada pública**, disponibilizando-a em um “canal público” (Ex.: repositório de acesso público) - **chave-pública**.
- Qualquer um pode cifrar mensagens com uma dada chave-pública.
- Somente o destinatário, detentor da correspondente chave de decifração (**chave-privada**, ou **secreta**), poderá decifrar a mensagem.

A **chave-privada** não precisa e nem deve ser dada a conhecer a ninguém, devendo ser **guardada em segredo** pelo seu detentor apenas, que deve também ter sido o responsável pela geração do seu *par de chaves*, enquanto a chave-pública pode ser publicada livremente.

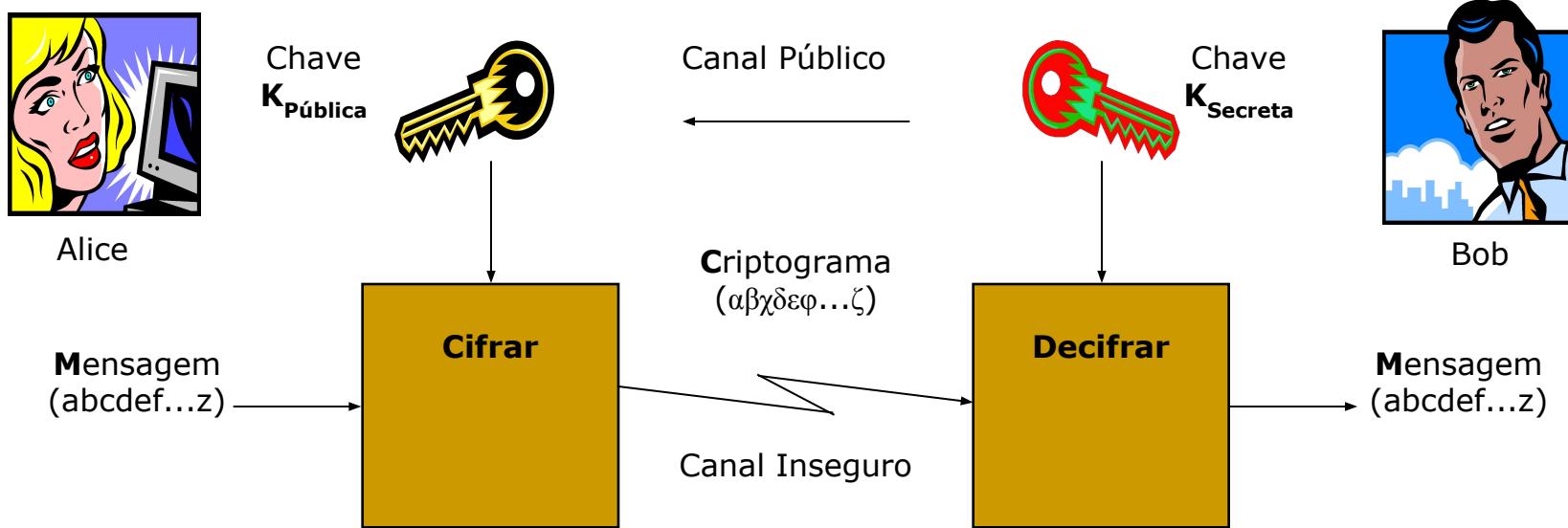
# Criptografia Assimétrica

- Importante
  - É computacionalmente inviável determinar a chave de decriptografia dado apenas o conhecimento do algoritmo e da chave de criptografia



# Criptografia Assimétrica

Uso de algoritmo criptográfico **assimétrico** (chave pública).

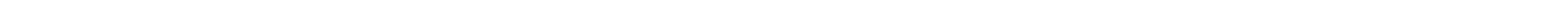


Para que Alice envie uma mensagem confidencial a Bob, ela deve encriptar essa mensagem com a chave pública de Bob que, de posse de sua chave privada, consegue descriptá-la. Como, em tese, ninguém tem acesso à chave privada de Bob, ninguém pode descriptar a mensagem.

# Criptografia Assimétrica

## □ Descrição do funcionamento do sistema (forma simplificada)

- Bob e todos os que desejam comunicar-se de modo seguro geram uma chave de ciframento e sua correspondente chave de deciframento.
- Bob mantém secreta a chave de deciframento; esta é chamada de sua *chave privada*.
- Bob torna pública a chave de ciframento: esta é chamada de sua *chave pública*.
- Qualquer pessoa pode obter uma cópia da chave pública. Bob encoraja isto, enviando-a para seus amigos ou publicando-a em boletins. Eva não tem nenhuma dificuldade em obtê-la.

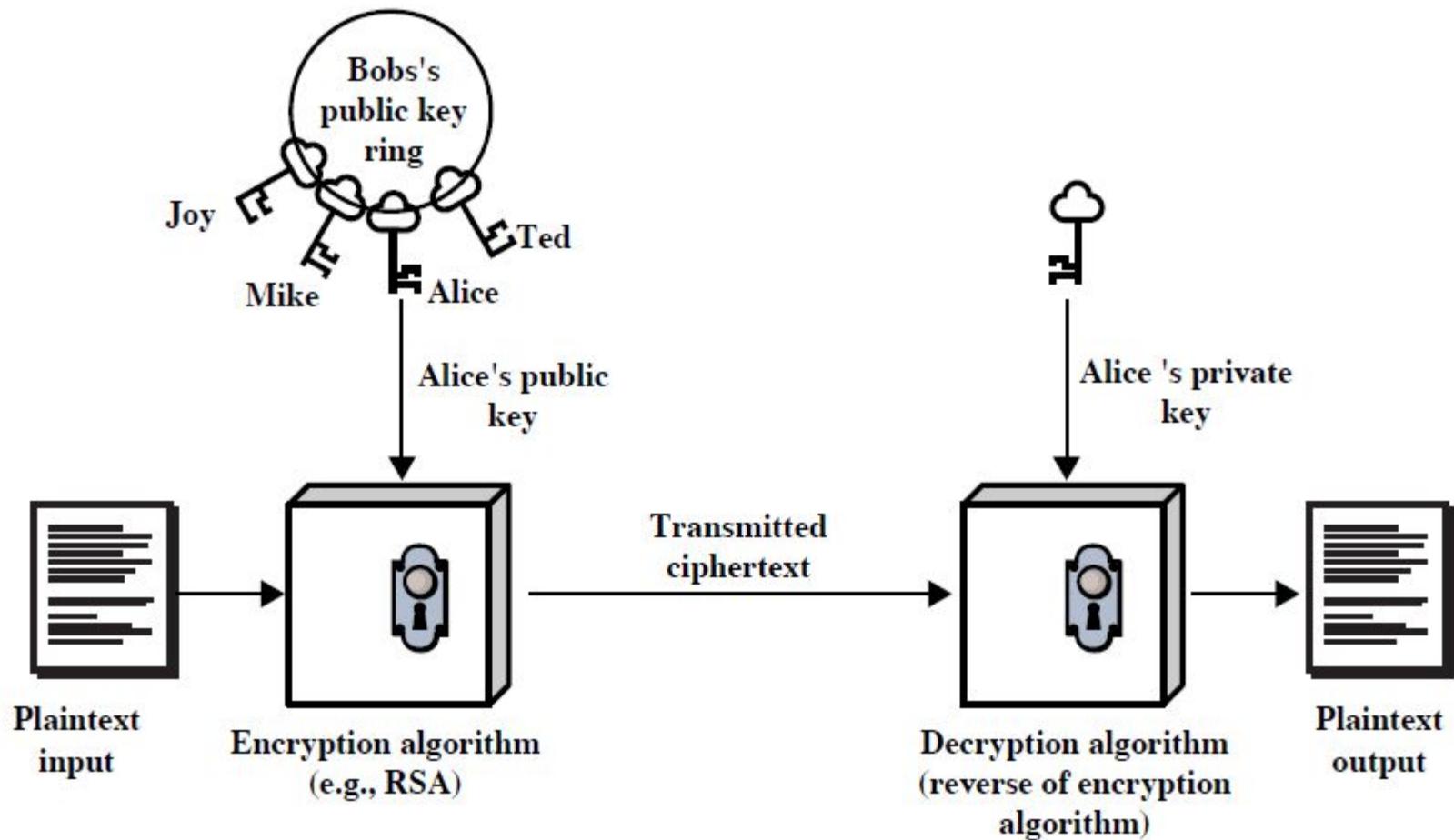


# Criptografia Assimétrica

## □ Descrição do funcionamento do sistema (forma simplificada)

- Alice deseja enviar uma mensagem a Bob: precisa primeiro encontrar a chave pública dele. Feito isto, ela cifra sua mensagem utilizando a chave pública de Bob, despachando-a em seguida.
  - Bob recebe a mensagem, a decifra facilmente com sua chave privada.
  - Trudy, que interceptou a mensagem em trânsito, não conhece a chave privada de Bob, embora conheça sua chave pública. Mas este conhecimento não a ajuda a decifrar a mensagem.
  - Mesmo Alice, que foi quem cifrou a mensagem com a chave pública de Bob, não pode decifrá-la agora.
-

# Chave Pública vs Privada



# Algoritmo de criptografia de chave pública

Requisitos:

- ① precisa de  $K_B^+(\cdot)$  e  $K_B^-(\cdot)$  tais que  
$$K_B^-(K_B^+(m)) = m$$
- ② dada a chave pública  $K_B^+$ , deverá ser impossível calcular chave privada  
 $K_B^-$
- ③ dada a chave pública  $K_B^+$  e o texto cifrado C, deverá ser inviável encontrar o texto claro m

---

**RSA:** Algoritmo de Rivest, Shamir, Adelson

## Pré-requisito: aritmética modular

- $x \bmod n = \text{resto de } x \text{ quando dividido por } n$
- Fatos:
  - $[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$
  - $[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$
  - $[(a \bmod n) * (b \bmod n)] \bmod n = (a*b) \bmod n$
- Assim,
$$(a \bmod n)^d \bmod n = a^d \bmod n$$
- Exemplo:  $x = 14$ ,  $n = 10$ ,  $d = 2$ :
$$(x \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$$
$$x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$$

# RSA: aprontando

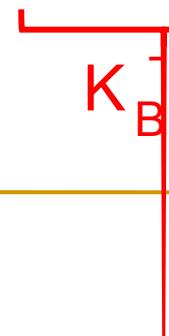
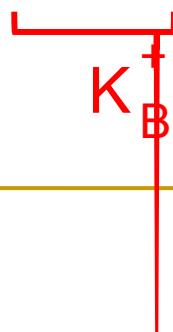
- Uma mensagem é um padrão de bits.
- Um padrão de bits pode ser representado exclusivamente por um número inteiro.
- Assim, criptografar uma mensagem é equivalente a criptografar um número.

## Exemplo

- $m = 10010001$ . Essa mensagem é representada exclusivamente pelo número decimal 145.
  - Para criptografar  $m$ , criptografamos o número correspondente, que gera um novo número (o texto cifrado).
-

# RSA: Criando par de chave pública/privada

1. Escolha dois números primos grandes  $p, q$ .  
(p. e., 1024 bits cada)
2. Calcule  $n = pq, z = (p-1)(q-1)$
3. Escolha  $e$  (com  $e < n$ ) que não tenha fatores comuns com  $z$ . ( $e, z$  são “relativamente primos”).
4. Escolha  $d$  tal que  $ed-1$  seja divisível exatamente por  $z$ .  
(em outras palavras:  $ed \bmod z = 1$  ).
5. Chave pública é  $(n, e)$ . Chave privada é  $(n, d)$ .



# RSA: criptografia, decriptação

0. Dados  $(n,e)$  e  $(n,d)$  conforme calculamos
1. Para criptografar a mensagem  $m (< n)$ , calcule

$$c = m^e \text{ mod } n$$

2. Para decriptar padrão de bits recebido,  $c$ , calcule

$$m = c^d \text{ mod } n$$

A mágica  
acontece!

$$m = \underbrace{(m^e \text{ mod } n)^d}_{c} \text{ mod } n$$

## Exemplo de RSA:

Bob escolhe  $p = 5$ ,  $q = 7$ . Depois,  $n = 35$ ,  $z = 24$ .

$e = 5$  (assim,  $e$ ,  $z$  relativamente primos).

$d = 29$  (assim,  $ed-1$  divisível exatamente por  $z$ ).

Criptografando mensagens de 8 bits.

criptografia:	<u>padrão de bits</u>	<u><math>m</math></u>	<u><math>m^e</math></u>	<u><math>c = m^e \text{ mod } n</math></u>
	00001100	12	248832	17

decriptação:	<u><math>c</math></u>	<u><math>c^d</math></u>	<u><math>m = c^d \text{ mod } n</math></u>
	17	481968572106750915091411825223071697	12

# Por que RSA funciona?

- Deve mostrar que  $c^d \bmod n = m$ 
    - onde  $c = m^e \bmod n$
  - Fato (teoria dos números):
    - para qualquer  $x$  e  $y$ :  $x^y \bmod n = x^{(y \bmod z)} \bmod n$
    - Desde que:  $n = pq$  e  $z = (p-1)(q-1)$
  - Assim,
$$\begin{aligned} c^d \bmod n &= (m^e \bmod n)^d \bmod n \\ &= m^{ed} \bmod n \\ &= m^{(ed \bmod z)} \bmod n \\ &= m^1 \bmod n \\ &= m \end{aligned}$$
-

# RSA: outra propriedade importante

A propriedade a seguir será *muito* útil adiante:

$$K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$$

use chave pública  
primeiro, seguida  
por chave privada

use chave privada  
primeiro, seguida  
por chave pública

*O resultado é o mesmo!*

---

Por que  $K_B^{-1}(K_B^+(m)) = m = K_B^+(K_B^{-1}(m))$ ?

Segue diretamente da aritmética modular:

$$\begin{aligned}(m^e \bmod n)^d \bmod n &= m^{ed} \bmod n \\&= m^{de} \bmod n \\&= (m^d \bmod n)^e \bmod n\end{aligned}$$



# Por que RSA é seguro?

- Suponha que você conheça a chave pública de Bob ( $n, e$ ). Qual é a dificuldade de determinar  $d$ ?
- Basicamente, é preciso encontrar fatores de  $n$  sem conhecer os dois fatores  $p$  e  $q$ .
- Fato: fatorar um número muito grande é difícil.

## Gerando chaves RSA

- É preciso achar números primos  $p$  e  $q$  grandes



# Chaves de sessão

- Exponenciação é computacionalmente intensa
- DES é pelo menos 100 vezes mais rápido que RSA

## Chave de sessão, $K_s$

- Bob e Alice usam RSA para trocar uma chave simétrica  $K_s$
  - Quando ambos tiverem  $K_s$ , eles usam a criptografia de chave simétrica
-

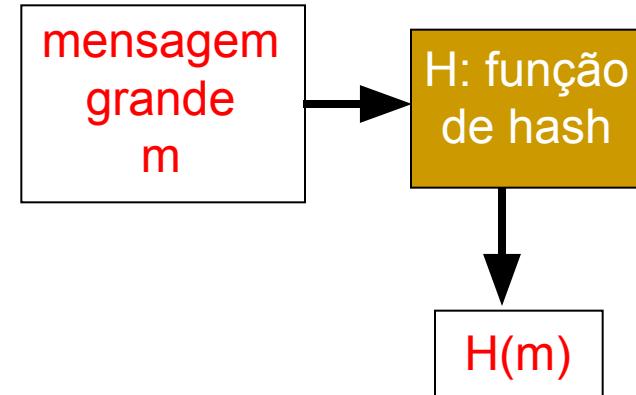
# Sumário

- Fundamentos
  - Criptografia Simétrica
  - Criptografia Assimétrica
  - **Integridade**
  - Autenticação
  - Assinatura Digital e Certificação Digital
  - Distribuição de Chaves
  - Serviços de Rede
-

# Integridade de mensagem

- permite a comunicação das partes para verificar que as mensagens recebidas são autênticas.
    - conteúdo da mensagem não foi alterado
    - origem da mensagem é quem/o que você pensa ser
    - mensagem não foi reproduzida - replay
    - sequência de mensagens é mantida
-

# Resumos de mensagens: h

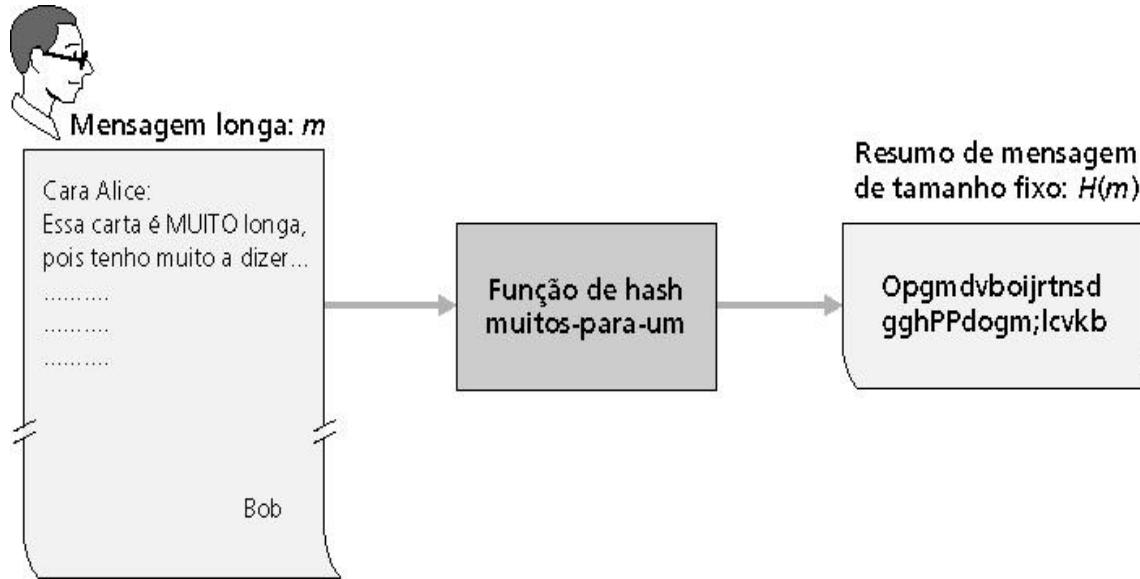


## Propriedades das funções de hash:

- Muitos-para-um
  - $H$  pode ser aplicado a um bloco de dados de tamanho qualquer.
  - $H$  produz um resumo (impressão digital) de tamanho fixo (*Função de Condensação*).
- $H(m)$  é relativamente fácil de calcular para qualquer  $m$ .
- Dado um resumo da mensagem  $h$ , é computacionalmente impraticável encontrar  $m$  tal que  $h = H(m)$  - *Funções Unidirecionais*
- Para qualquer mensagem  $m$ , é computacionalmente inviável encontrar  $n \neq m$  tal que  $H(n) = H(m)$ .
- É computacionalmente inviável encontrar qualquer par  $(m, n)$  tal que  $H(n) = H(m)$ .

---

# Resumos de mensagens



Aplicar função hash  $H$  a  $m$  para obter um resumo de tamanho fixo,  $H(m)$

# Soma de verificação da Internet: resumo de mensagem fraco

soma de verificação da internet tem propriedades da função de hash:

- produz resumo de tamanho fixo (soma de 16 bits) de entrada
- é muitos-para-um
- mas, dada mensagem com dado valor de hash, é fácil achar outra mensagem com o mesmo valor de hash.
- exemplo: soma de verificação simplificada: soma porções de 4 bytes de cada vez:

mensagem    formato ASCII

I	O	U	1	49	4F	55	31
0	0	.	9	30	30	2E	39
9	B	O	B	39	42	D2	42

mensagem    formato ASCII

I	O	U	9	49	4F	55	39
0	0	.	1	30	30	2E	31
9	B	O	B	39	42	D2	42

---

B2 C1 D2 AC

mensagens diferentes  
mas somas de verificação idênticas!

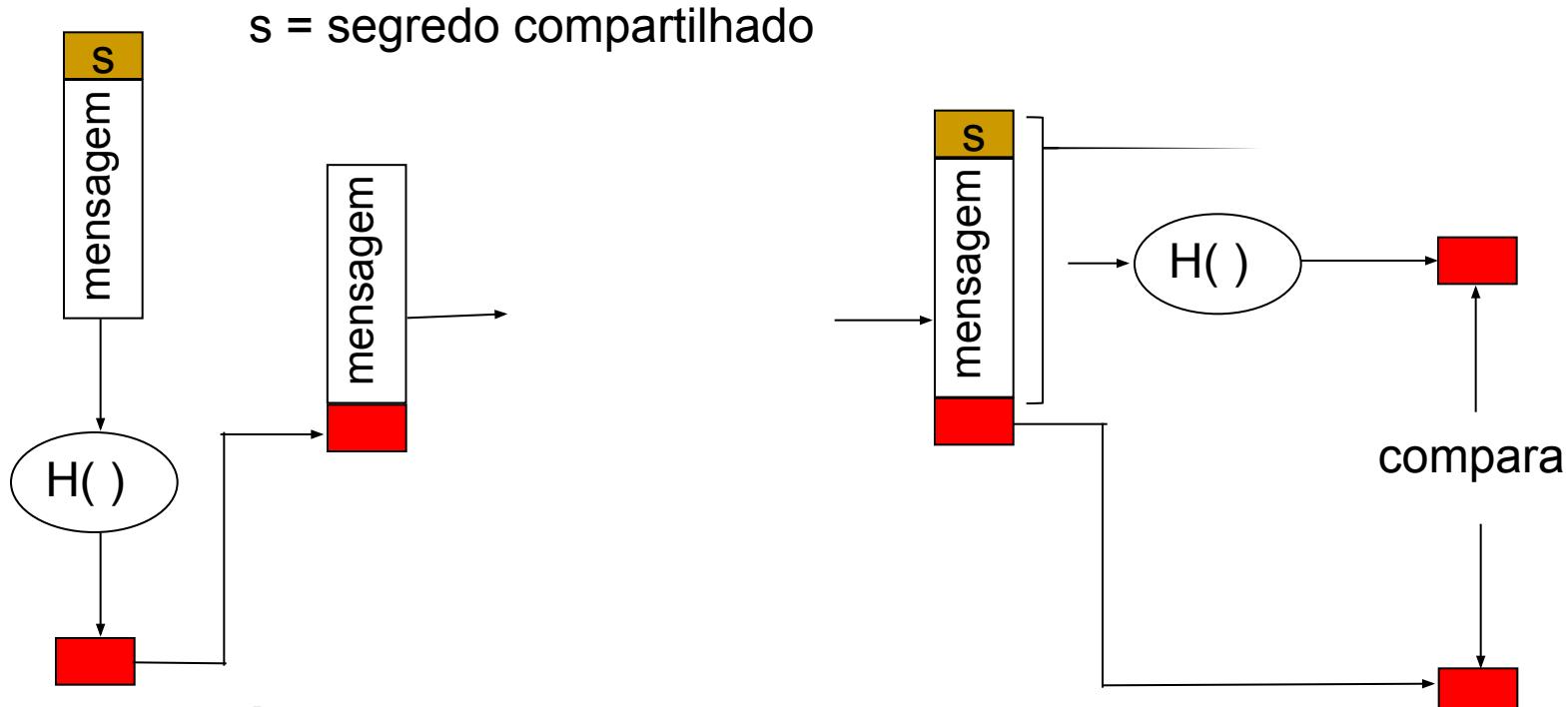
---

B2 C1 D2 AC

# Algoritmos de função de hash

- função de hash MD5 bastante usada (RFC 1321)
  - calcula resumo de mensagem de 128 bits em processo de 4 etapas.
- SHA-1 também é usado.
  - padrão nos EUA [NIST, FIPS PUB 180-1]
  - resumo de mensagem de 160 bit

# Message Authentication Code (MAC)



- **autentica remetente**
- **verifica integridade da mensagem**
- sem criptografia!
- também chamado “hash chaveado”
- notação:  $MD_m = H(s||m)$  ; envia  $m||MD_m$

# Exemplo: OSPF

- lembre-se de que OSPF é um protocolo de roteamento intra-AS
- cada roteador cria mapa do AS inteiro (ou área) e executa algoritmo do caminho mais curto pelo mapa.
- roteador recebe anúncios de estado do enlace (LSAs) de todos os outros roteadores no AS.

## Ataques:

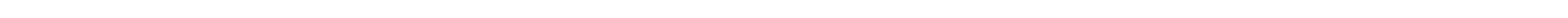
- inserção de mensagem
- exclusão de mensagem
- modificação de mensagem
- como sabemos se uma mensagem OSPF é autêntica?

# Autenticação OSPF

- dentro de um sistema autônomo, roteadores enviam mensagens OSPF entre si.
  - OSPF oferece escolhas de autenticação
    - Sem autenticação
    - Senha compartilhada: inserida em aberto no campo de autenticação de 64 bits no pacote OSPF
    - hash criptográfico
  - hash criptográfico com MD5
    - MD5 é executado sobre uma concatenação do pacote OSPF e chave secreta compartilhada
    - hash MD5 então anexado ao pacote OSPF; encapsulado no datagrama IP
-

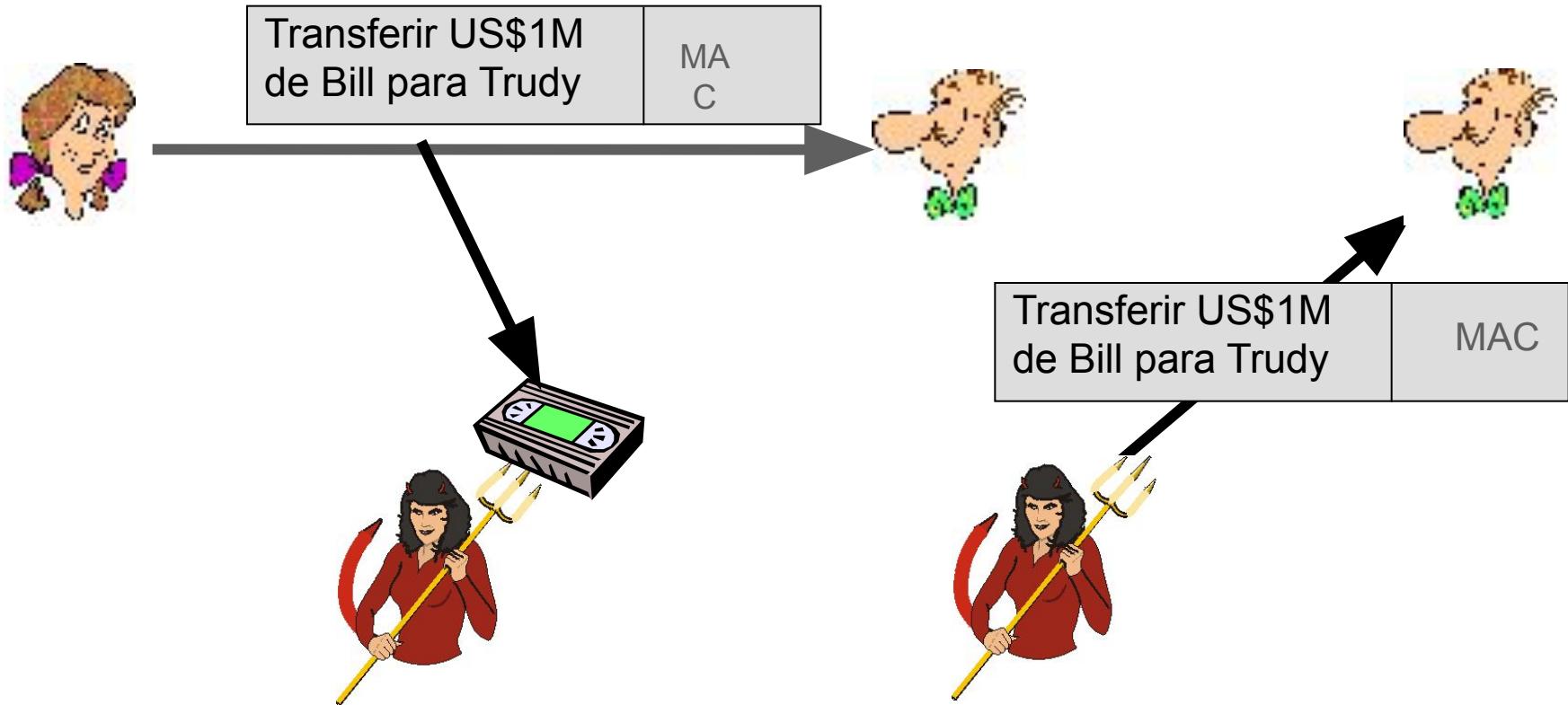
# Autenticação do ponto final

- deseja ter certeza do remetente da mensagem – *autenticação do ponto final*
- supondo que Alice e Bob tenham um segredo compartilhado, MAC oferecerá autenticação do ponto final
  - sabemos que Alice criou a mensagem
  - mas ela a enviou?

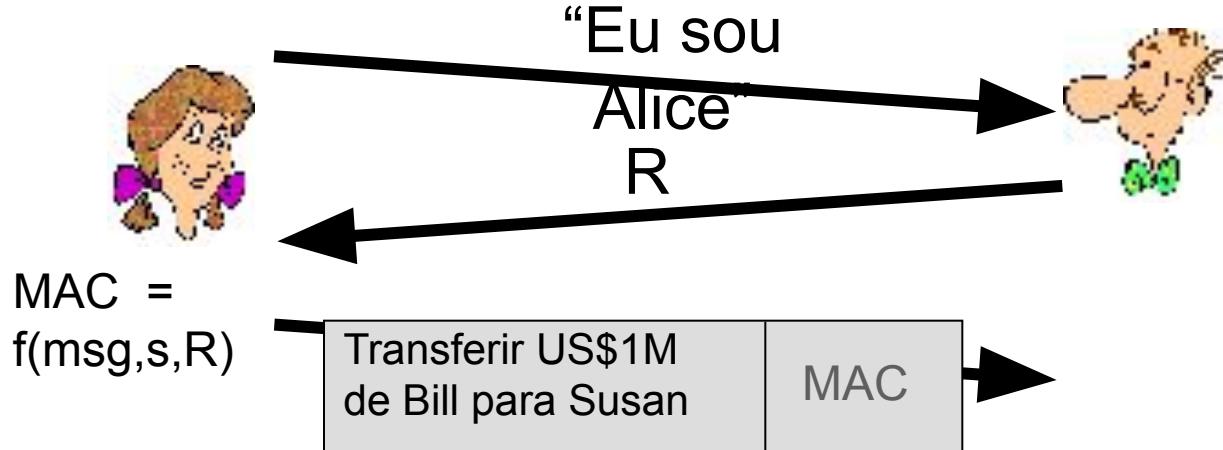


# Ataque de reprodução

$$\text{MAC} = f(\text{msg}, s)$$



# Defendendo contra ataque de reprodução: nonce



# Sumário

- Fundamentos
  - Criptografia Simétrica
  - Criptografia Assimétrica
  - Integridade
  - Autenticação
  - Assinatura Digital e Certificação Digital
  - Distribuição de Chaves
  - Serviços de Rede
-

# Autenticação

**Objetivo:** Bob quer que Alice “prove” sua identidade para ele

**Protocolo ap1.0:** Alice diz “Eu sou Alice”.

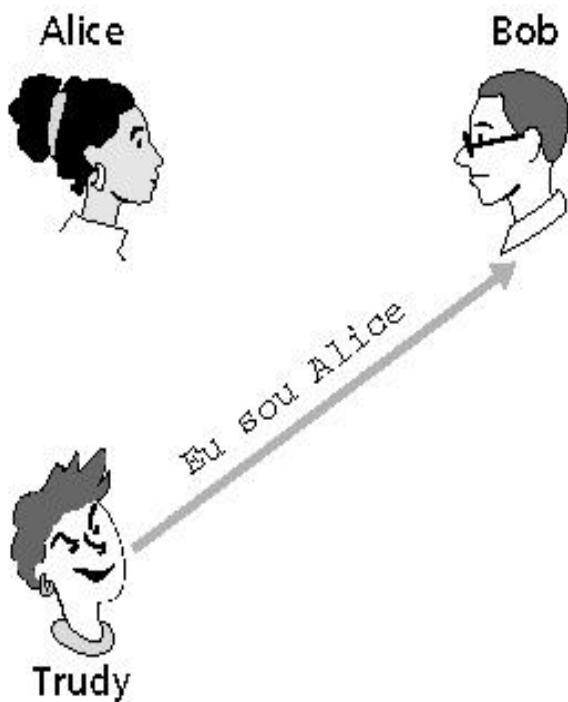


Cenário de falha??

# Autenticação (cont.)

**Objetivo:** Bob quer que Alice “prove” sua identidade para ele

**Protocolo ap1.0:** Alice diz “Eu sou Alice”.



Numa rede,  
Bob não pode “ver” Alice,  
então Trudy simplesmente  
declara  
que ela é Alice

# Autenticação: outra tentativa

**Protocolo ap2.0:** Alice diz “Eu sou Alice” e envia seu endereço IP junto como prova.



Cenário de falha??



# Autenticação: outra tentativa (cont.)

**Protocolo ap2.0:** Alice diz “Eu sou Alice” num pacote IP contendo seu endereço IP de origem.



Trudy pode criar um pacote “trapaceando” (*spoofing*) o endereço de Alice

# Autenticação: outra tentativa (cont.)

**Protocolo ap3.0:** Alice diz “Eu sou Alice” e envia sua senha secreta como prova.

Cenário de falha??



Legenda:

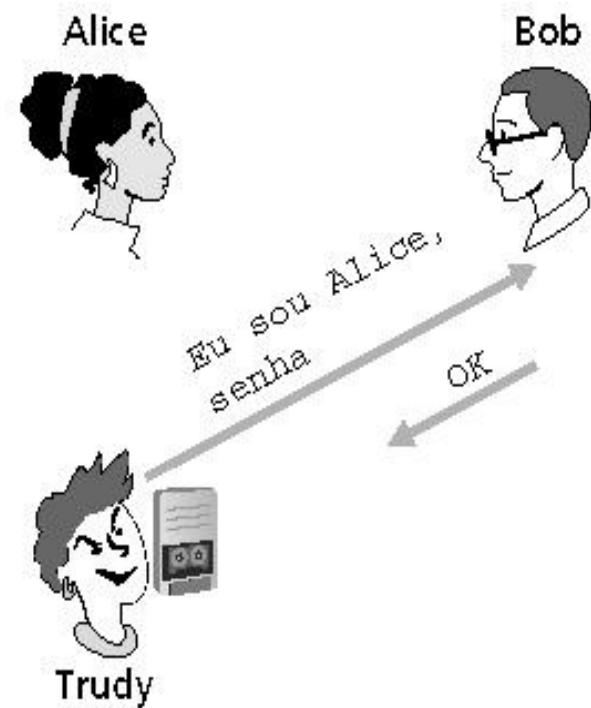


Gravador

# Autenticação: outra tentativa (cont.)

**Protocolo ap3.0:** Alice diz “Eu sou Alice” e envia sua senha secreta como prova.

**ataque de playback:**  
Trudy grava o pacote de Alice e depois o envia de volta para Bob.



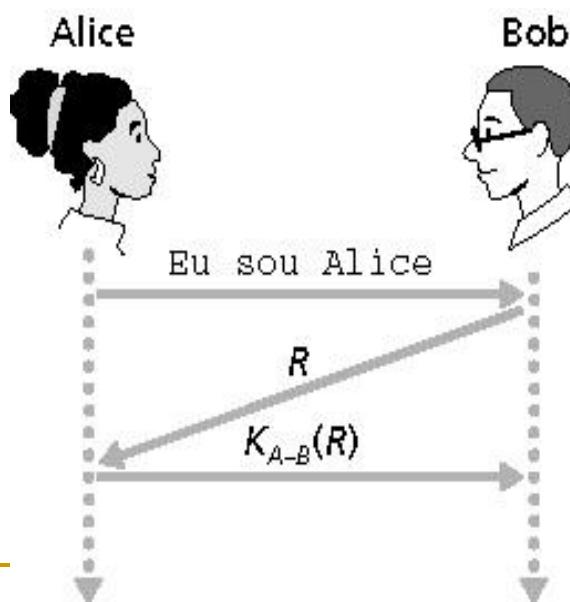
# Autenticação: mais uma tentativa (cont.)

**Protocolo ap3.1:** Alice diz “Eu sou Alice” e envia sua senha secreta *criptografada* para prová-lo.

**Meta:** evitar ataque de reprodução (playback).

**Nonce:** número (R) usado apenas uma vez na vida.

**ap4.0:** para provar que Alice “está ao vivo”, Bob envia a Alice um **nonce**, R. Alice deve devolver R, criptografado com a chave secreta comum.



Falhas, problemas?

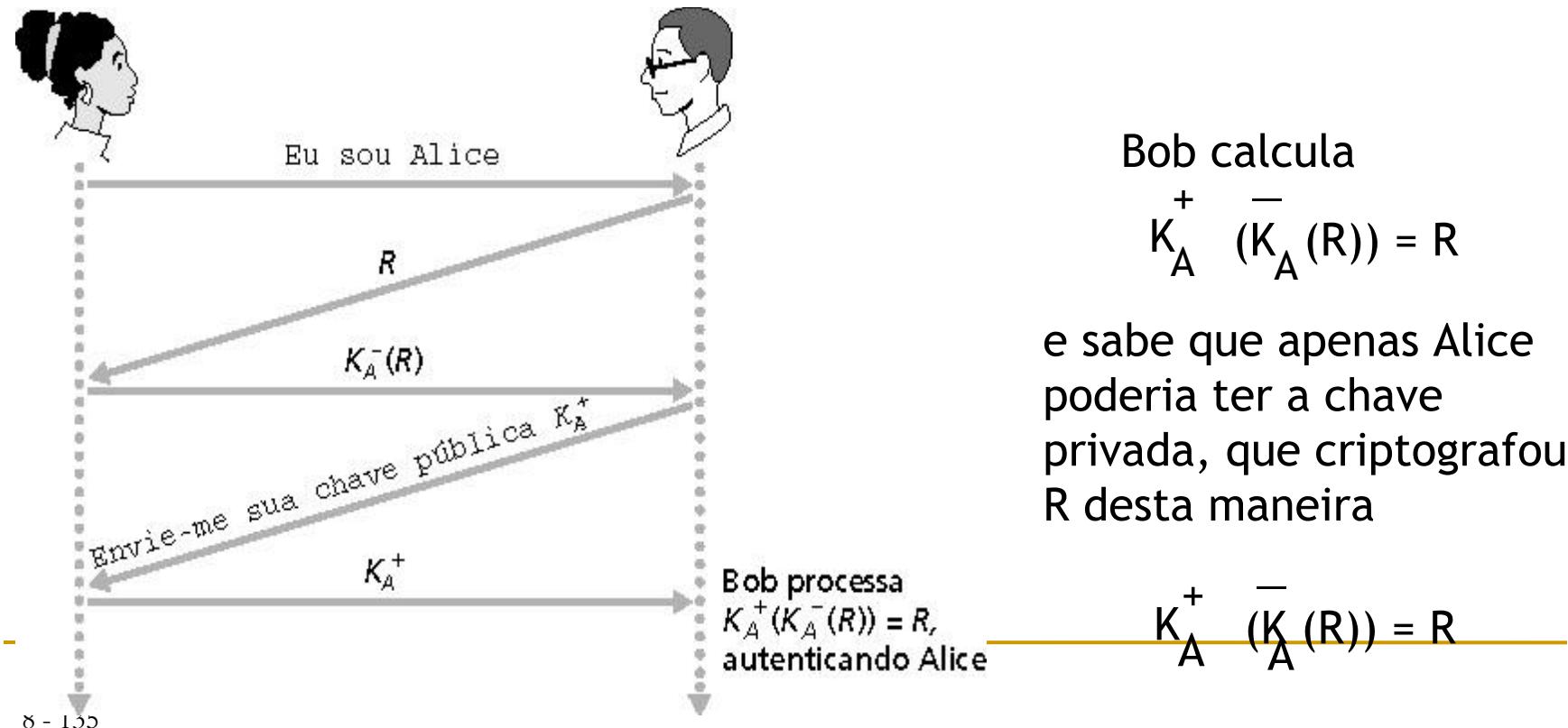
Alice está ao vivo,  
e apenas Alice  
conhece a chave  
para criptografar o  
nonce, então ela  
deve ser Alice!

# Autenticação: ap5.0

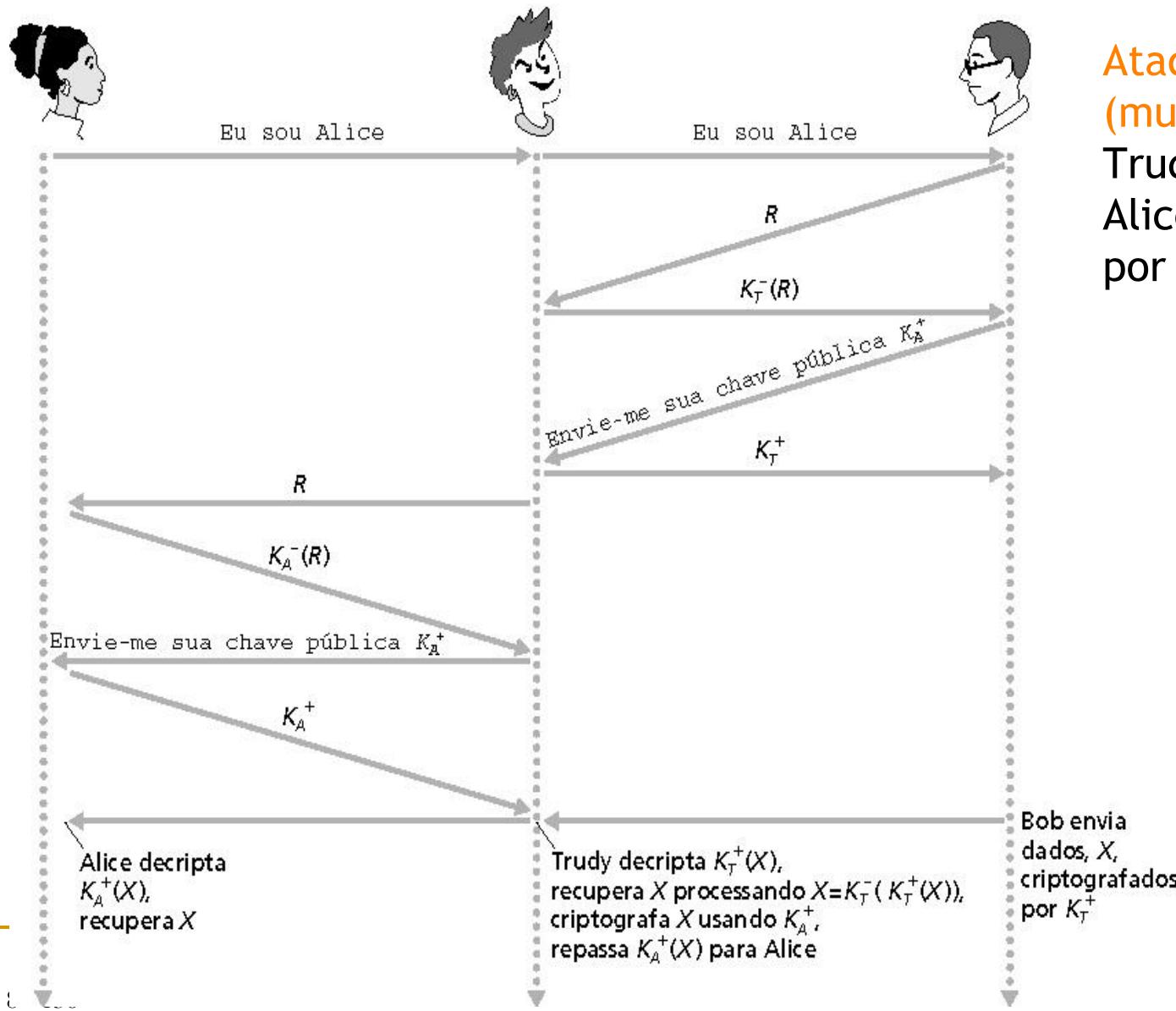
ap4.0 exige chave secreta compartilhada.

- É possível autenticar usando técnicas de chave pública?

**ap5.0:** usar nonce, criptografia de chave pública.



# ap5.0: falha de segurança



Ataque do homem (mulher) no meio:  
Trudy se passa por Alice (para Bob) e por Bob (para Alice)

## ap5.0: falha de segurança

**Ataque do homem no meio:** Trudy se passa por Alice (para Bob) e por Bob (para Alice)

Difícil de detectar:

- O problema é que Trudy recebe todas as mensagens também!
- Bob recebe tudo o que Alice envia e vice-versa. (Ex.: então Bob/Alice podem se encontrar uma semana depois e recordar a conversação.)

# Sumário

- Fundamentos
  - Criptografia Simétrica
  - Criptografia Assimétrica
  - Integridade
  - Autenticação
  - Assinatura Digital e Certificação Digital
  - Distribuição de Chaves
  - Serviços de Rede
-

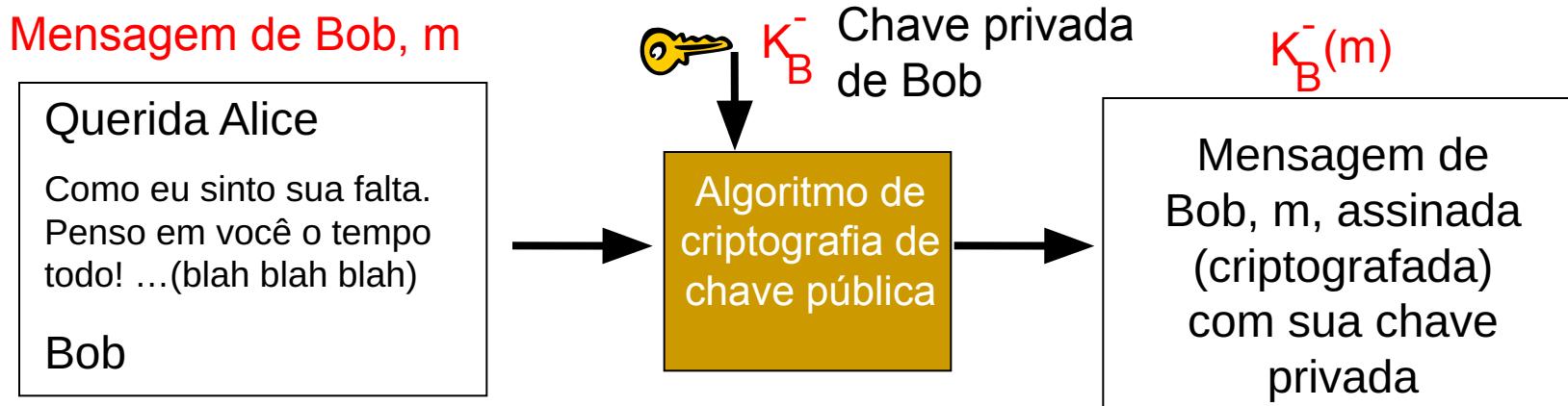
# Assinaturas digitais

Técnica criptográfica semelhante a assinaturas escritas a mão.

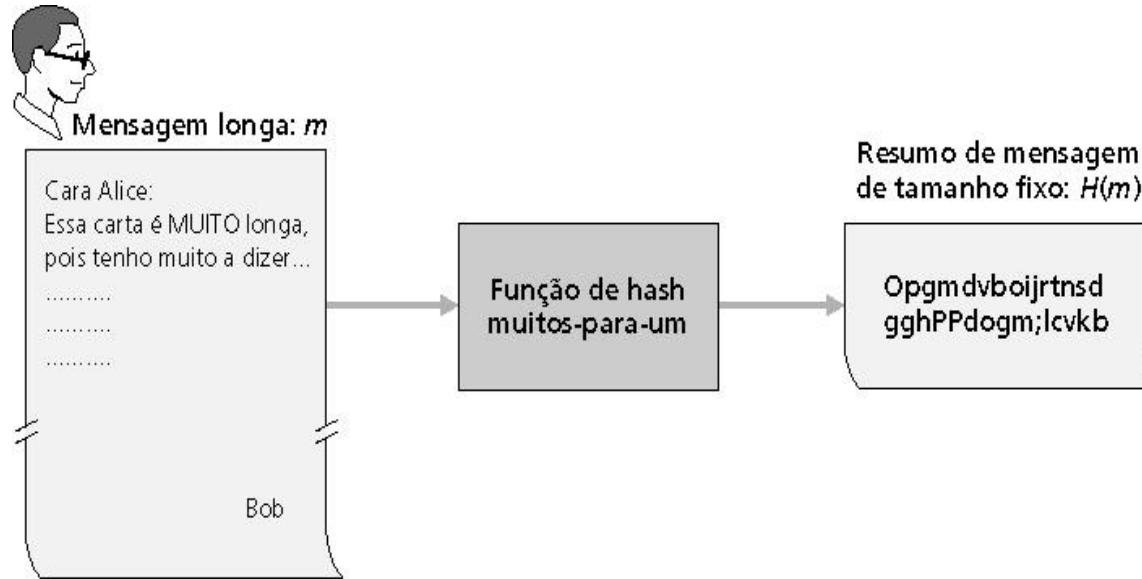
- remetente (Bob) assina documento digitalmente, estabelecendo que é o dono/criador do documento.
  - objetivo semelhante a um MAC, exceto que agora usamos criptografia de chave pública.
  - **verificável, não falsificável**: destinatário (Alice) pode provar a alguém que Bob, e ninguém mais (incluindo Alice), deverá ter assinado o documento.
-

## assinatura digital simples para mensagem $m$ :

- Bob assina  $m$  criptografando com sua chave privada  $K_B^-$ , criando mensagem “assinada”,  $K_B^-(m)$



# Resumos de mensagens nas Assinaturas Digitais



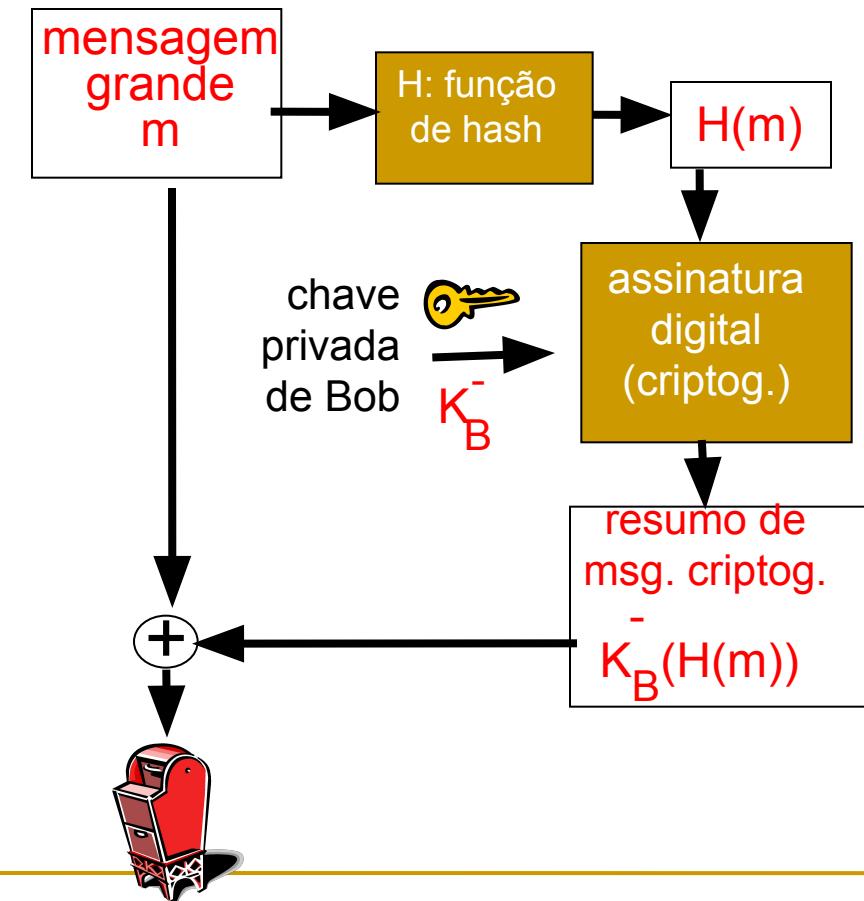
Computacionalmente caro criptografar mensagens longas com chave pública

**Meta:** assinaturas digitais de comprimento fixo, facilmente computáveis, “impressão digital”

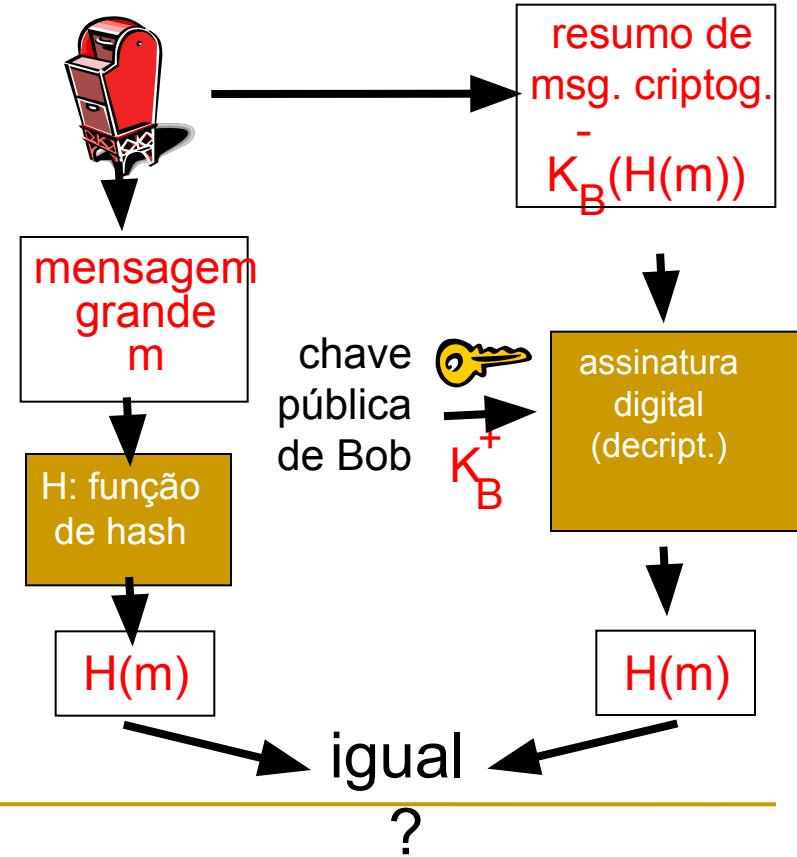
- Aplicar função hash  $H$  a  $m$  para obter um resumo de tamanho fixo,  $H(m)$

# Assinatura digital = resumo de mensagem assinada

Bob envia mensagem assinada em forma digital:



Alice verifica assinatura e integridade da mensagem assinada em forma digital:



# Assinaturas digitais (mais)

- Suponha que Alice receba msg  $m$ , assinatura digital  $K_B^-(m)$
- Alice verifica  $m$  assinada por Bob aplicando chave pública de Bob  $K_B^+ a K_B^-(m)$ , depois verifica  $K_B^+(K_B^-(m)) = m$ .
- se  $K_B^+(K_B^-(m)) = m$ , quem assinou  $m$  deve ter usado a chave privada de Bob.

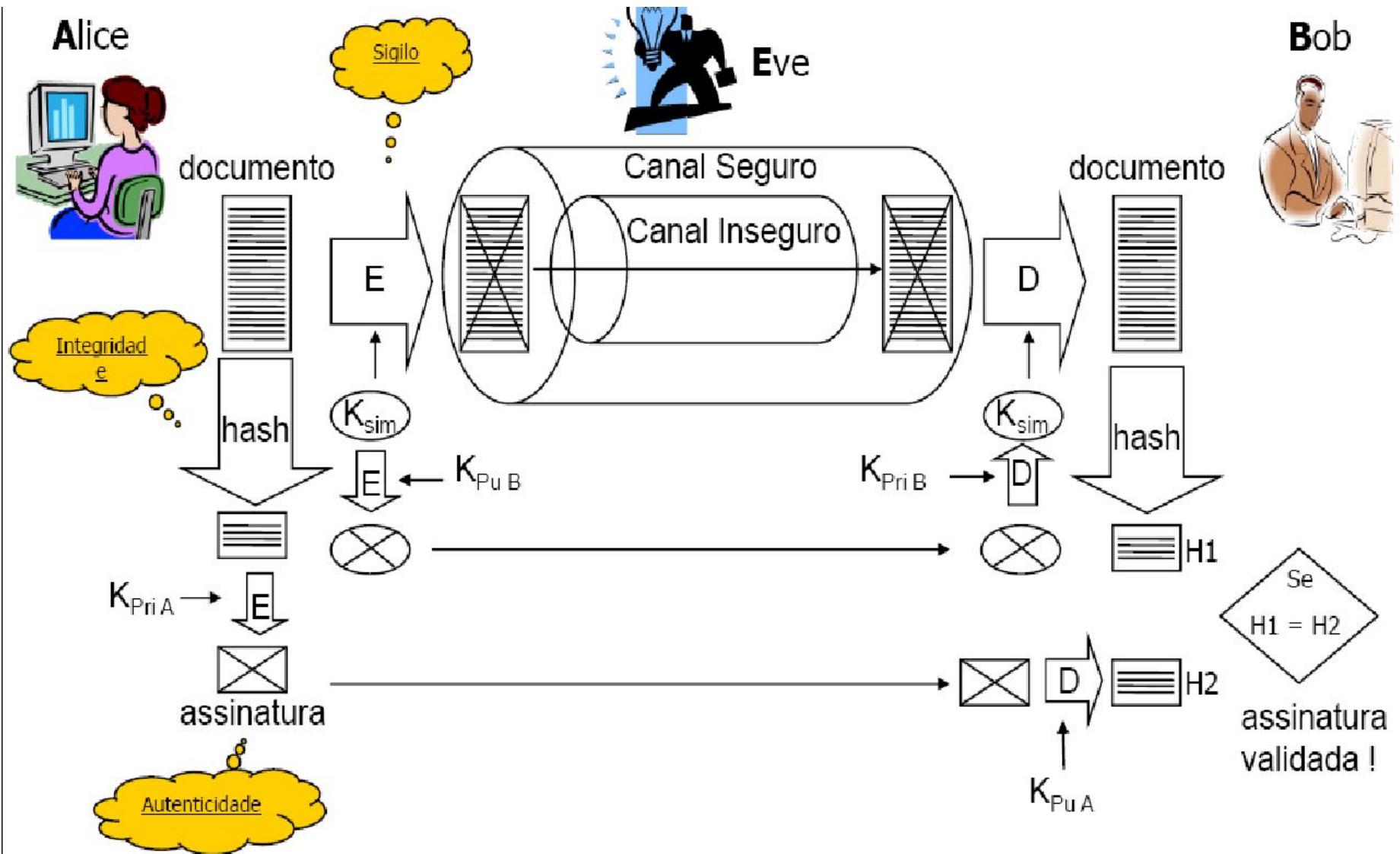
Assim, Alice verifica se:

- Bob assinou  $m$ .
- Ninguém mais assinou  $m$ .
- Bob assinou  $m$  e não  $m'$ .

Não repudição:

- Alice pode levar  $m$  e assinatura  $K_B^-(m)$  ao tribunal e provar que Bob assinou  $m$ .
-

# Integração

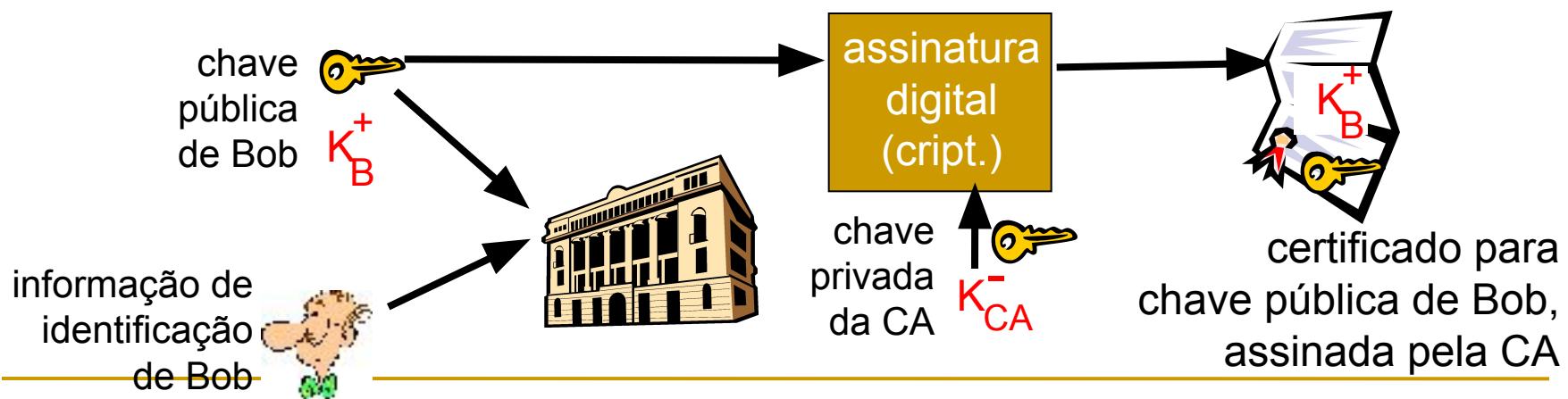


# Certificação de chave pública

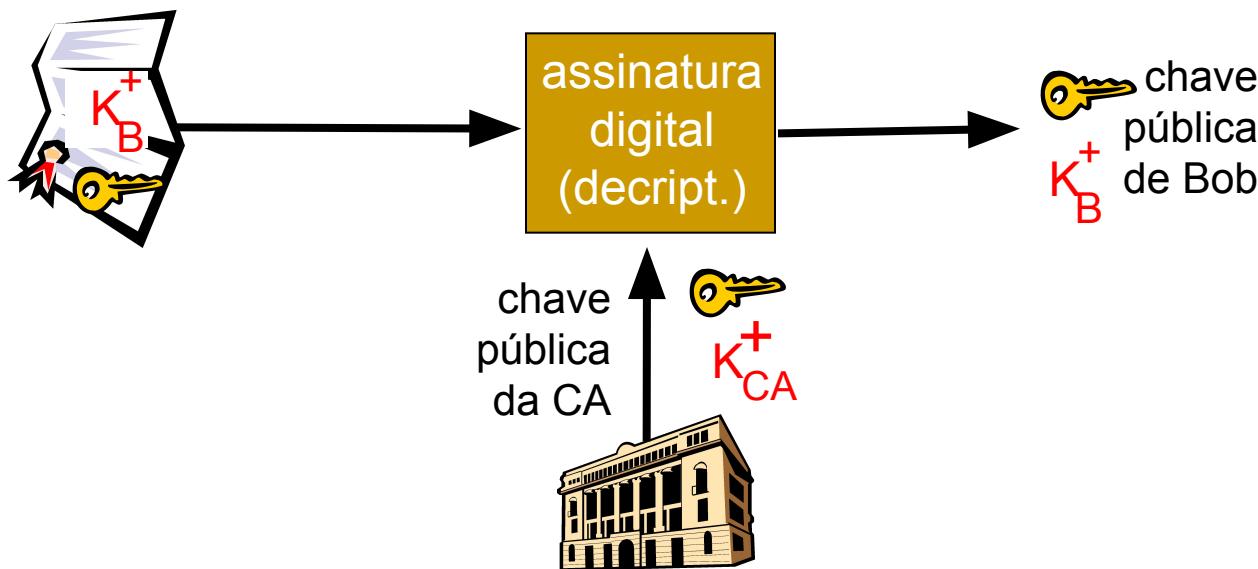
- motivação: Trudy prega peça da pizza em Bob
  - Trudy cria pedido por e-mail:  
*Prezada pizzaria, Por favor, me entregue quatro pizzas de calabresa. Obrigado, Bob.*
  - Trudy assina pedido com sua chave privada
  - Trudy envia pedido à pizzaria
  - Trudy envia à pizzaria sua chave pública, mas diz que é a chave pública de Bob.
  - pizzaria verifica assinatura; depois, entrega quatro pizzas para Bob.
  - Bob nem sequer gosta de calabresa.

# Autoridades de certificação

- **autoridade de certificação (CA):** vincula chave pública à entidade particular, E.
- E (pessoa, roteador) registra sua chave pública com CA.
  - E fornece “prova de identidade” à CA.
  - CA cria certificado vinculando E à sua chave pública.
  - certificado contendo chave pública de E assinada digitalmente pela CA – CA diz “esta é a chave pública de E”



- quando Alice quer a chave pública de Bob:
  - **recebe certificado de Bob (Bob ou outro).**
  - **aplica chave pública da CA ao certificado de Bob, recebe chave pública de Bob**



# Exemplo de certificado

I hereby certify that the public key

19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A  
belongs to

Robert John Smith

12345 University Avenue

Berkeley, CA 94702

Birthday: July 4, 1958

Email: bob@superduper.net.com

SHA-1 hash of the above certificate signed with the CA's private key



# Campos do X.509 – RFC 2459

Campo	Significado
Version	A versão do X.509
Serial number	Este número, somado ao nome da CA, identifica de forma exclusiva o certificado
Signature algorithm	O algoritmo usado para assinar o certificado
Issuer	Nome X.500 da CA
Validity period	A hora inicial e final do período de validade
Subject name	A entidade cuja chave está estando certificada
Public key	A chave pública do assunto e a ID do algoritmo que a utiliza
Issuer ID	Uma ID opcional que identifica de forma exclusiva o emissor do certificado
Subject ID	Uma ID opcional que identifica de forma exclusiva o assunto do certificado
Extensions	Muitas extensões foram definidas
Signature	A assinatura do certificado (assinado pela chave privada da CA)

## ■ certificado contém:

- nome do emissor
  - nome da entidade, endereço, domínio etc.
  - chave pública da entidade
  - assinatura digital (assinada com a chave privada do emissor)
-

# Certificado X.509 do BB

**Visualizador de certificados: "www2.bancobrasil.com.br"**

**Geral** | **Detalhes**

**Este certificado foi homologado para estes usos:**

Certificado para servidor|SSL

---

**Expedido para:**

Nome Comum (CN) www2.bancobrasil.com.br  
Empresa (O) Banco do Brasil S.A.  
Unidade Organizacional (OU) DITEC  
Número de série 08:69:A3:8E:3D:3A:BA:C0:44:B1:C2:D3:A3:E3:63:31

**Expedido por:**

Nome Comum (CN) Thawte SSL CA  
Empresa (O) Thawte, Inc.  
Unidade Organizacional (OU) <Não faz parte do certificado>

**Validade:**

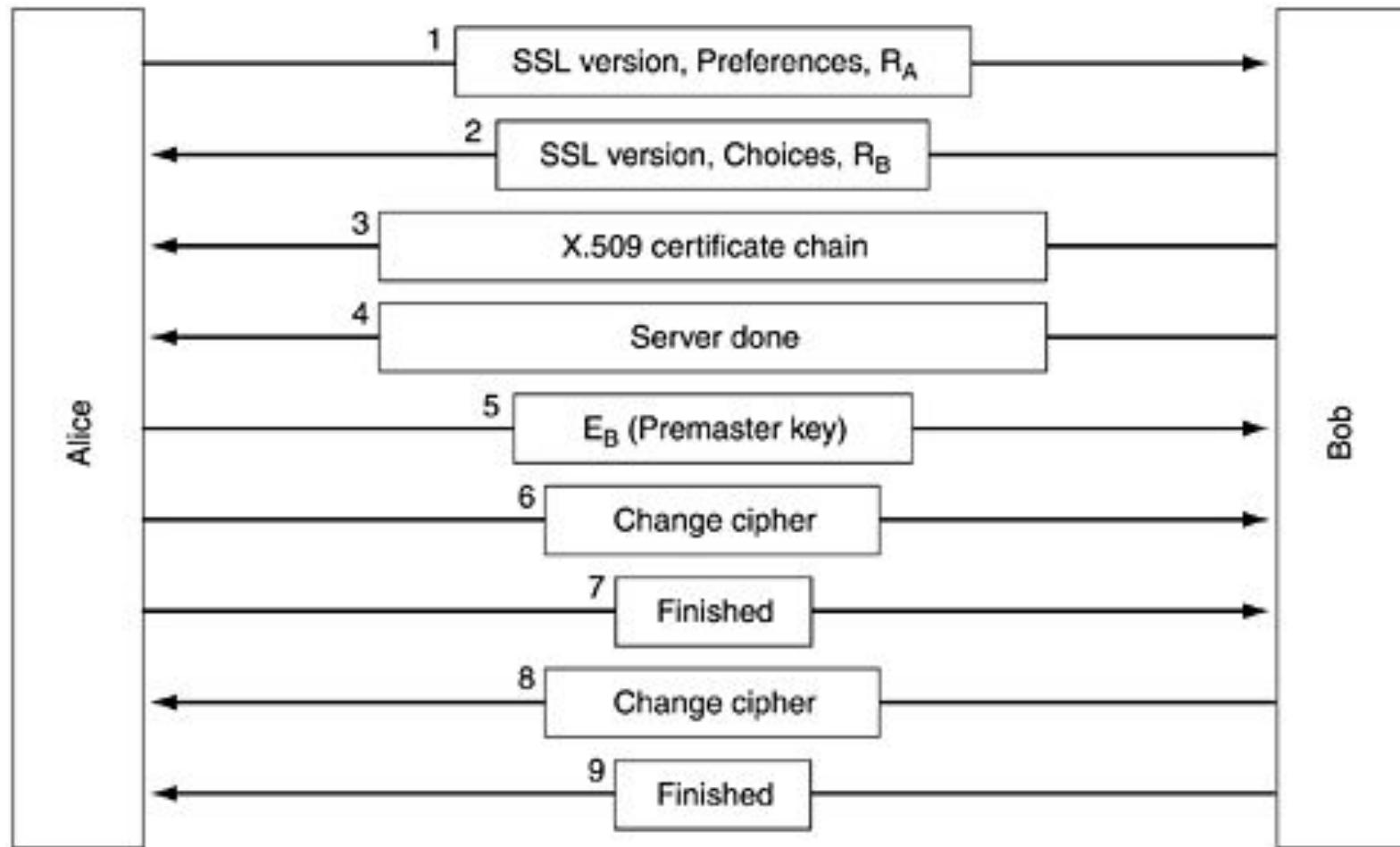
Expedido em 8/6/2011  
Válido até 8/7/2012

**Assinaturas:**

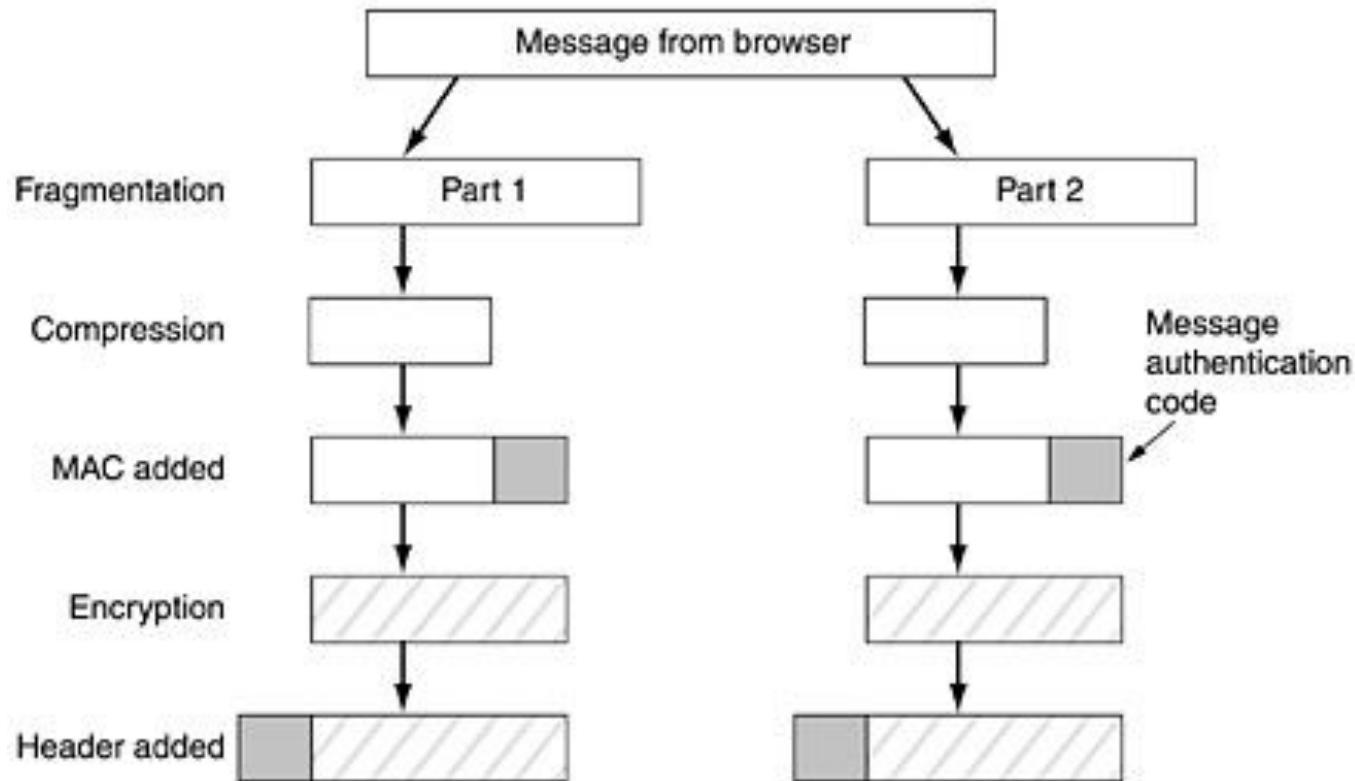
Assinatura SHA1 D9:8B:66:E5:51:21:06:C6:47:88:78:3A:C3:21:E0:FD:60:FE:9C:E2  
Assinatura MD5 27:79:0C:B2:28:69:59:69:6D:79:E0:25:9C:2A:E2:2E

# SSL – *Secure Sockets Layer*

(Estabelecimento de Conexão)



# SSL – Transmissão de Dados



# Certificados digitais - Exemplos



# e-CPF, e-CNPJ, RIC

- Motivação: Há tempos que as pessoas utilizam assinaturas à caneta, carimbos, selos, entre outros recursos, para comprovar a autenticidade de documentos, expressar concordância com determinados procedimentos, declarar responsabilidades, etc.
  - Hoje, muitas dessas atividades podem ser feitas através da internet.
-

# Como obter o Certificado Digital passo a passo

- 1- Escolher uma **Autoridade Certificadora** (SERPRO, CAIXA ECONOMICA FEDERAL, SERASA, RECEITA FEDERAL,Certisign,FENACON) da ICP-Brasil;



# Como obter o Certificado Digital passo a passo

- 2- Solicitar no próprio portal da internet da AC escolhida a emissão de certificado digital de pessoa física (ex: **e-CPF**) e/ou jurídica (ex: **e-CNPJ**). Os tipos mais comercializados são:
  - **A1** (validade de um ano – armazenado no computador/smartphone) ou **A3** (validade de até três anos – armazenado em **cartão** ou **token criptográfico**).
-

# Como obter o Certificado Digital passo a passo

- 3- Depois da solicitação, a AC vai confirmar o pedido, em geral via e-mail, e encaminhará os contatos da Autoridade de Registro (AR) mais próxima do cliente, para que seja agendada uma visita presencial, quando o interessado levará os documentos e será identificado. Quem escolher o **certificado tipo A3** poderá receber na própria AR o **cartão ou token** com o certificado digital.
-

# Como obter o Certificado Digital passo a passo

- 4- Aguardar uma notificação da AC para baixar o certificado.

e-CPF A3 Certisign - em cartão + leitora USB - 1 ano	▼	R\$ 290,00
e-CPF A3 Certisign em cartão (requer leitora) - 1 ano	▼	R\$ 160,00
e-CPF A3 Certisign (exige mídia criptográfica) - 1 ano	▼	R\$ 110,00
e-CPF A3 Certisign (exige mídia criptográfica) - 3 anos	▼	R\$ 165,00
e-CPF A3 Certisign em cartão (requer leitora) - 3 anos	▼	R\$ 215,00
e-CPF A3 Certisign - em cartão + leitora USB - 3 anos	▼	R\$ 365,00
e-CPF A1 Certisign - 1 ano	▼	R\$ 110,00
e-CPF A3 Certisign + token - 3 anos	▼	R\$ 365,00

# Documentos Necessários

## ■ Pessoa Física e-CPF

- ❑ Cédula de Identidade ou Passaporte, se estrangeiro, original e duas cópias;
  - ❑ Cadastro de Pessoa Física (CPF), original e duas cópias;
  - ❑ Comprovante de Residência, original e duas cópias;
  - ❑ Número de Identificação Social - NIS (PIS, PASEP, ou CI), se informado, original e duas cópias;
  - ❑ Cadastro Específico do INSS - CEI, se informado, original e duas cópias;
  - ❑ Título de Eleitor, se informado, original e duas cópias;
  - ❑ Duas fotos 3x4 recentes;
  - ❑ Termo de Titularidade devidamente preenchido;
  - ❑ Comprovante do depósito de pagamento do certificado.
-

# Aplicações

## ■ Cartório Eletrônico

- Certidão de protesto; Registro Civil (certidão de nascimento, de casamento, de óbito); Certidão de Registro; Registro de Imóveis; Tabelionato de Notas (certidão de escritura e de procuração).

□ <http://www.cartorio24horas.com.br/>

## ■ Pregão Eletrônico

□ <http://www.pregao.com.br/>

□ <http://www.comprasnet.gov.br/>

# Aplicações

- Receita Federal
  - Central Virtual de Atendimento ao Contribuinte
    - <http://www.receita.fazenda.gov.br/Atendvirtual/default.htm>
    - <http://www.receita.fazenda.gov.br/atendvirtual/servdisponivel.htm>
  - Nota Fiscal Eletrônica
    - <http://www.nfe.fazenda.gov.br/portal/>



# Aplicações

- Programa Universidade para Todos – PROUNI
    - O sistema é acessado pela instituição de ensino superior por meio de certificado digital.
  - Sistema de Diárias e Passagens - Serv. Público
-

# Public-Key Infrastructure (PKI)

- Certificados e Autoridades de certificação
- ICP-Brasil -  
<https://www.gov.br/iti/pt-br/assuntos/icp-brasil>

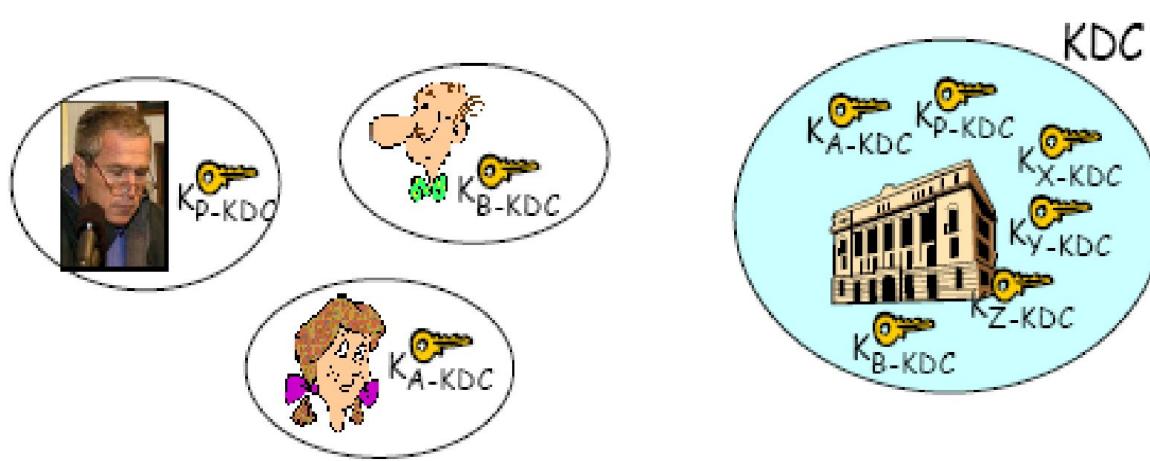


# Sumário

- Fundamentos
  - Criptografia Simétrica
  - Criptografia Assimétrica
  - Integridade
  - Autenticação
  - Assinatura Digital e Certificação Digital
  - Distribuição de Chaves
  - Serviços de Rede
-

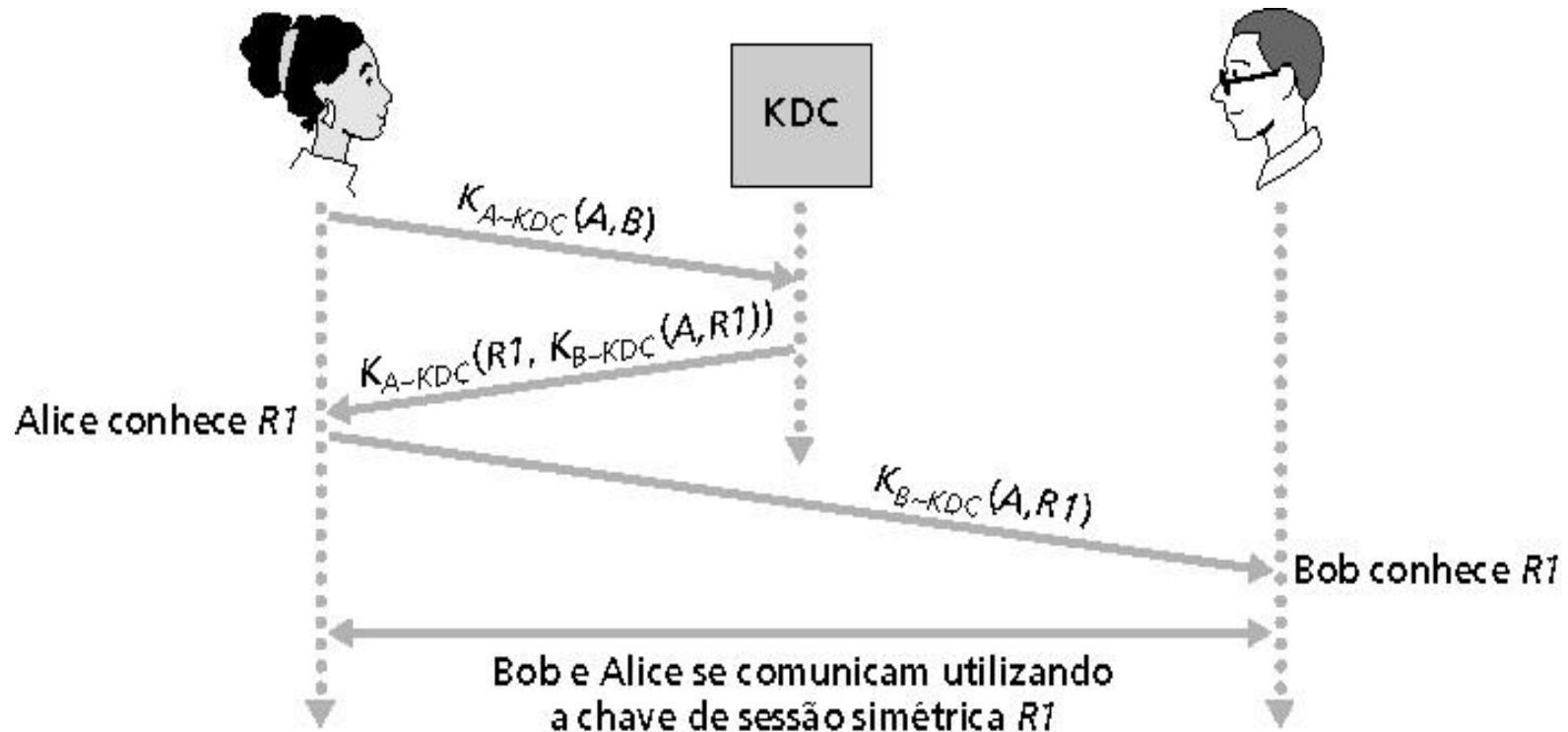
# Centro de distribuição de chave (KDC)

- Alice e Bob necessitam de uma chave simétrica comum
- **KDC**: servidor compartilha diferentes chaves secretas com *cada* usuário registrado (muitos usuários)
- Alice e Bob conhecem as próprias chaves simétricas,  $K_{A-KDC}$   $K_{B-KDC}$ , para comunicação com o KDC



# Centro de distribuição de chave (KDC)

P.: Como o KDC permite que Bob e Alice determinem uma chave simétrica comum para se comunicarem entre si?



# Troca de chaves *Diffie-Hellman*

Depende para sua segurança da dificuldade de se calcular logaritmos discretos

## Logaritmo Discreto

- **Raiz primitiva de um número primo  $p$ :** suas potências modulo  $p$  geram todos os inteiros positivos de 1 até  $p - 1$ . Ou seja, se  $a$  é uma raiz primitiva do número primo  $p$ , então os números

$$a \bmod p, a^2 \bmod p, a^3 \bmod p, \dots, a^{p-1} \bmod p$$

São distintos e consistem nos inteiros de 1 até  $p - 1$  em alguma permutação

Para qualquer inteiro  $b$  e uma raiz primitiva  $a$  do número primo  $p$ , podemos encontrar um exponente **exclusivo**  $i$  tal que

$$b \equiv a^i \pmod{p}$$

O expoente  $i$  é referenciado como logaritmo discreto de  $b$  na base  $a$ , mod  $p$

---

# Troca de chaves *Diffie-Hellman*

## Global Public Elements

$q$  prime number

$\alpha$   $\alpha < q$  and  $\alpha$  a primitive root of  $q$

## User A Key Generation

Select private  $X_A$   $X_A < q$

Calculate public  $Y_A$   $Y_A = \alpha^{X_A} \text{ mod } q$

## User B Key Generation

Select private  $X_B$   $X_B < q$

Calculate public  $Y_B$   $Y_B = \alpha^{X_B} \text{ mod } q$

# Troca de chaves *Diffie-Hellman*

Calculation of Secret Key by User A

$$K = (Y_B)^{X_A} \bmod q$$

Calculation of Secret Key by User B

$$K = (Y_A)^{X_B} \bmod q$$

A chave calculada K é a mesma para Bob e Alice. Essa chave é denominada chava de sessão.

---

# Troca de chaves *Diffie-Hellman*

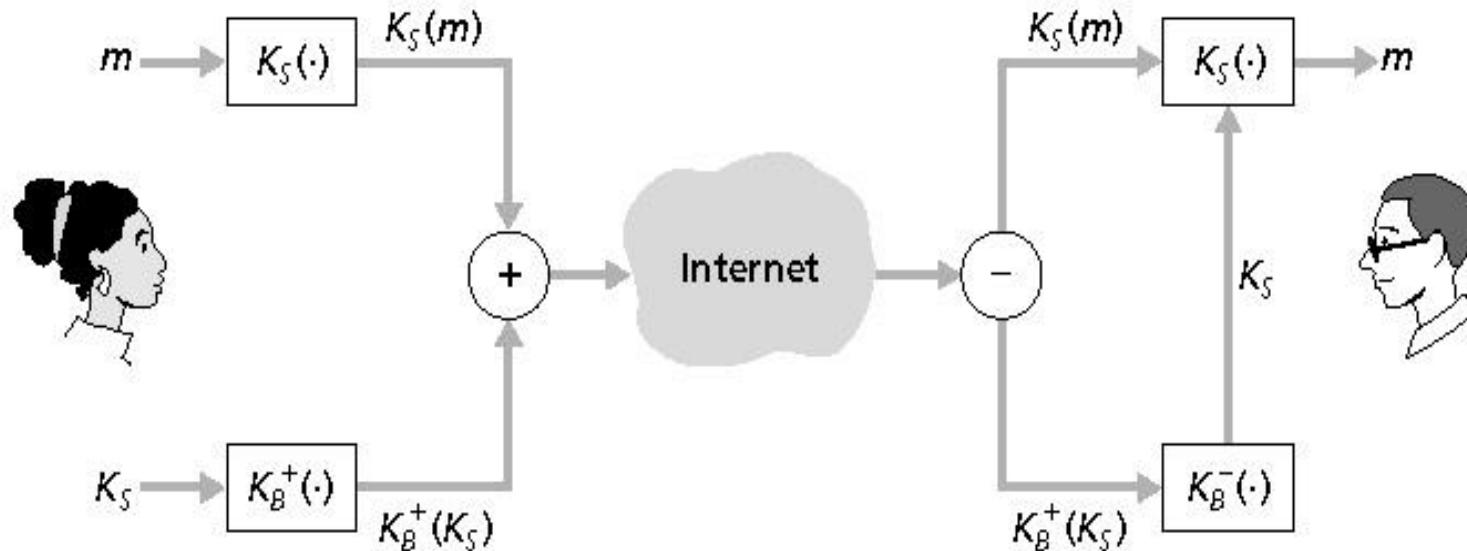
- Exemplo
    - $q = 353, \alpha = 3$
    - Chave secreta de Alice  $X_A = 97$
    - Chave secreta de Bob  $X_B = 233$
  - Cálculos
    - Chave pública de Alice  $Y_A = 3^{97} \text{ mod } 353 = 40$
    - Chave pública de Bob  $Y_B = 3^{233} \text{ mod } 353 = 248$ 
      - *Alice envia  $Y_A$  para Bob e Bob envia  $Y_B$  para Alice*
    - Alice calcula:  $K = (Y_B)^{X_A} \text{ mod } 353 = 248^{97} \text{ mod } 353 = 160$
    - Bob calcula:  $K = (Y_A)^{X_B} \text{ mod } 353 = 40^{233} \text{ mod } 353 = 160$ 
      - Ambos encontram o valor 160 que será usado como chave de sessão
-

# Sumário

- Fundamentos
  - Criptografia Simétrica
  - Criptografia Assimétrica
  - Integridade
  - Autenticação
  - Assinatura Digital e Certificação Digital
  - Distribuição de Chaves
  - Serviços de Rede
-

# E-mail seguro

- Alice quer enviar e-mail confidencial,  $m$ , para Bob.



Alice envia uma mensagem de e-mail,  $m$

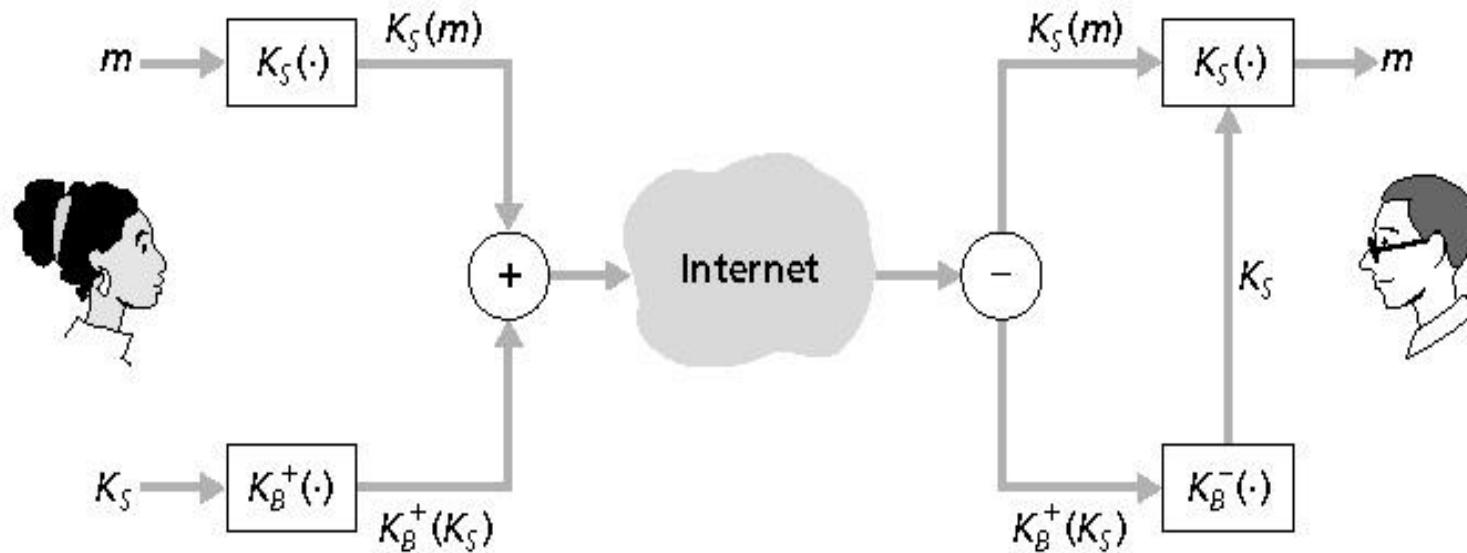
Bob recebe uma mensagem de e-mail,  $m$

**Alice:**

- Gera uma chave privada *simétrica*,  $K_S$
- Codifica mensagem com  $K_S$  (por eficiência)
- Também codifica  $K_S$  com a chave pública de Bob
- Envia tanto  $K_S(m)$  como  $K_B(K_S)$  para Bob

# E-mail seguro (cont.)

- Alice quer enviar e-mail confidencial,  $m$ , para Bob.



Alice envia uma mensagem de e-mail,  $m$

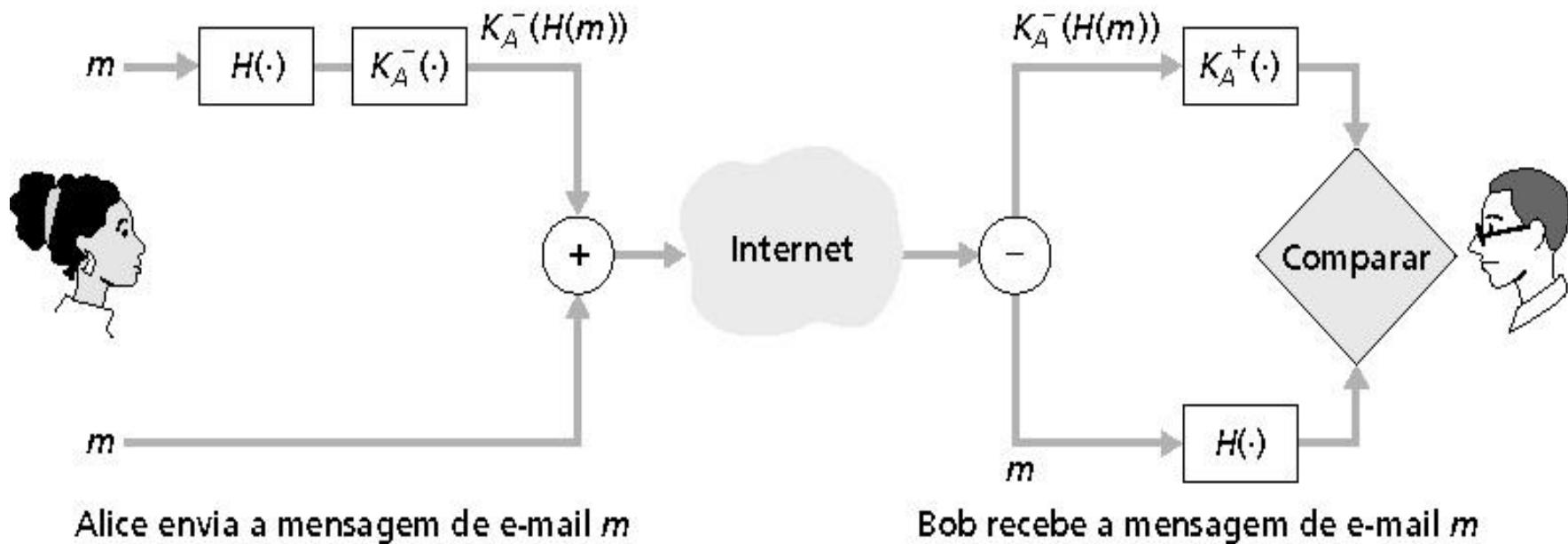
Bob recebe uma mensagem de e-mail,  $m$

## Bob:

- Usa sua chave privada para decodificar e recuperar  $K_S$
- Usa  $K_S$  para decodificar  $K_S(m)$  e recuperar  $m$

# E-mail seguro (cont.)

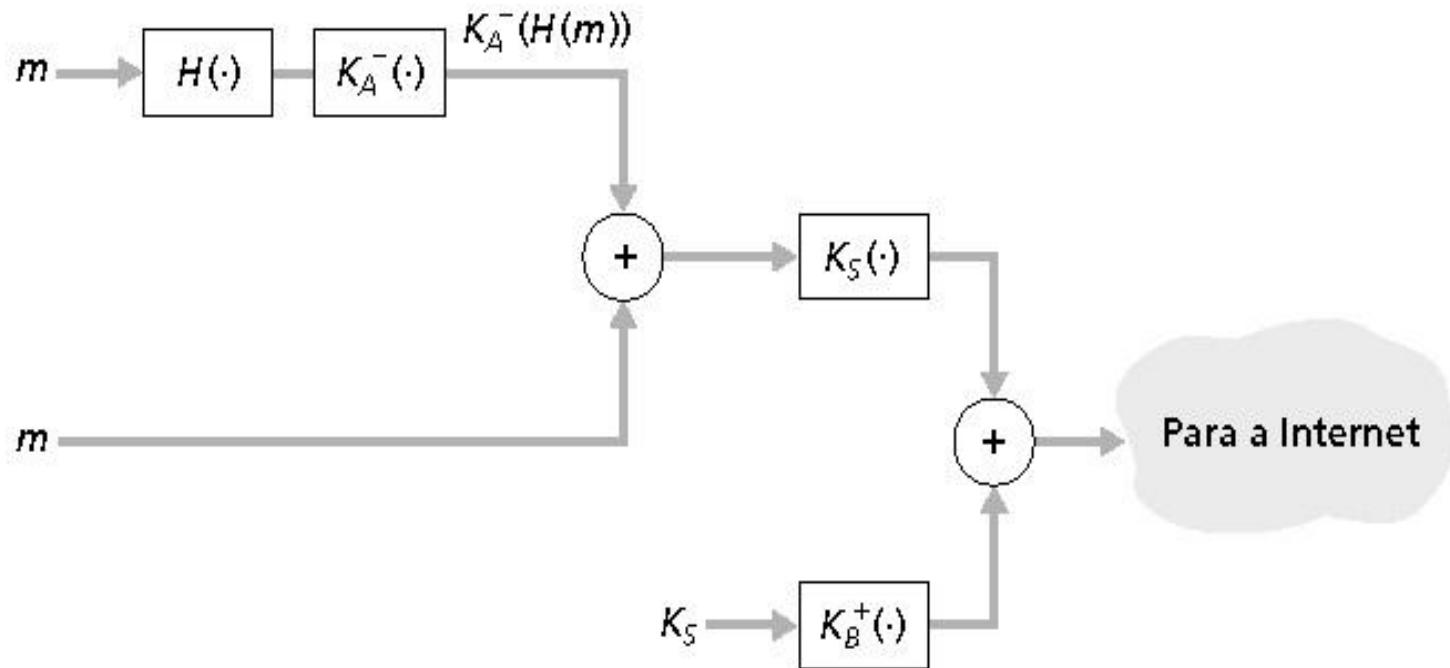
- Alice quer fornecer autenticação de emissor e integridade de mensagem.



- Alice assina digitalmente a mensagem
- Envia tanto a mensagem (aberta) quanto a assinatura digital

# E-mail seguro (cont.)

- Alice quer fornecer confidencialidade, autenticação de emissor e integridade de mensagem



Alice usa três chaves: sua chave privada, a chave pública de Bob e uma nova chave simétrica

# Pretty good privacy (PGP)

- Esquema de codificação de e-mail da Internet, padrão de fato
- Usa criptografia de chave simétrica, criptografia de chave pública, função de hash e assinatura digital, como descrito
- Fornece confidencialidade, autenticação do emissor, integridade
- Seu inventor, Phil Zimmermann, foi alvo durante 3 anos de uma investigação federal

## Uma mensagem PGP:

```
---BEGIN PGP SIGNED MESSAGE---
```

Hash: SHA1

Bob: My husband is out of town  
tonight. Passionately yours,  
Alice

```
---BEGIN PGP SIGNATURE---
```

Version: PGP 5.0

Charset: noconv

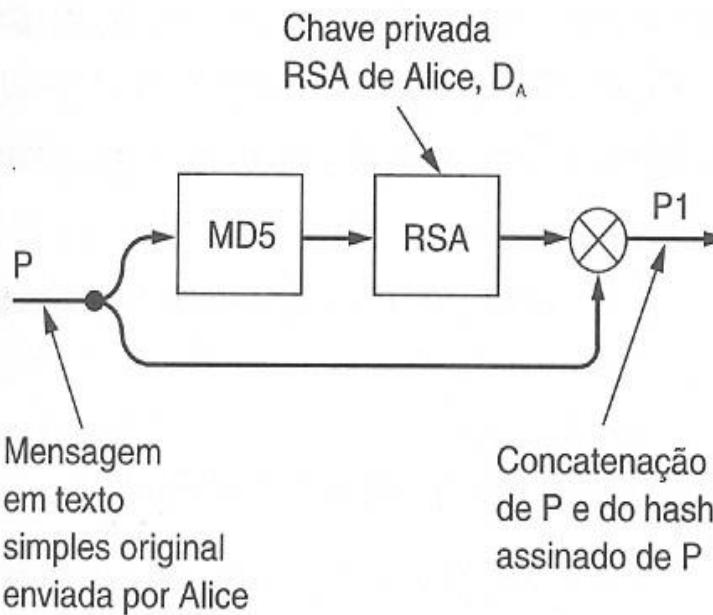
yhHJRHHGJGhgg/12EpJ+lo8gE4vB3mqJh  
FEvZP9t6n7G6m5Gw2

```
---END PGP SIGNATURE---
```

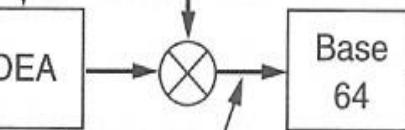
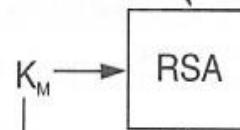
# Pretty good privacy (PGP)

KM : Chave de mensagem de uma única vez para IDEA

$\bigcirc \times$  : Concatenação



Chave pública RSA de Bob,  $E_B$



Concatenação de  $P_1.Z$  criptografada com IDEA e  $K_M$  criptografada com  $E_B$

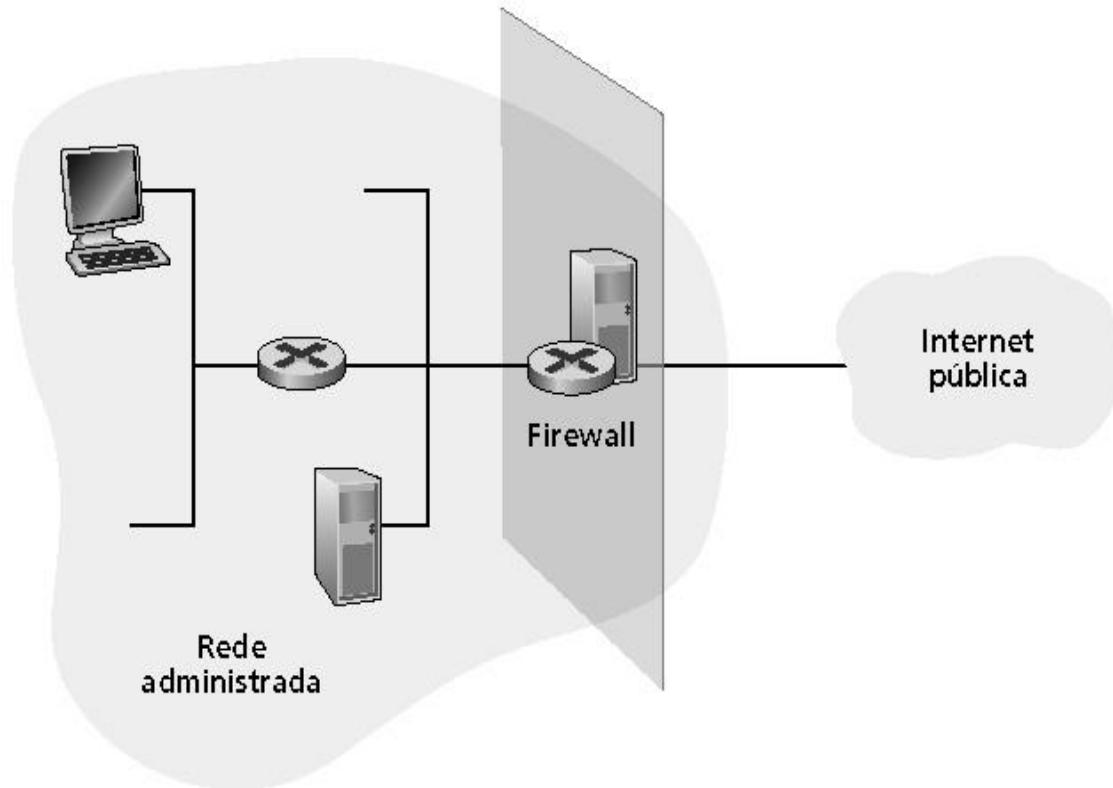
# Tamanhos de Chave - PGP

- 1. Casual (384 bits): pode ser decifrado com facilidade atualmente.
  - 2. Comercial (512 bits): pode ser decifrado por empresas de informatica.
  - 3. Militar (1024 bits): ninguem no planeta consegue decifrar.
  - 4. Alienigena (2.048 bits): nao pode ser decifrado por ninguem de outros planetas.
-

# Firewalls

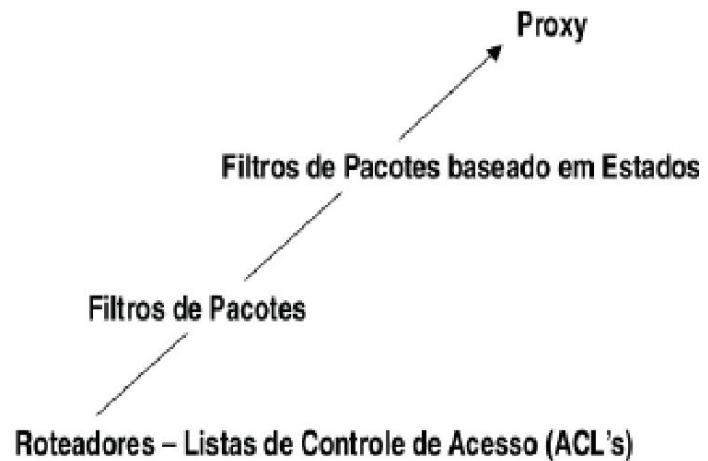
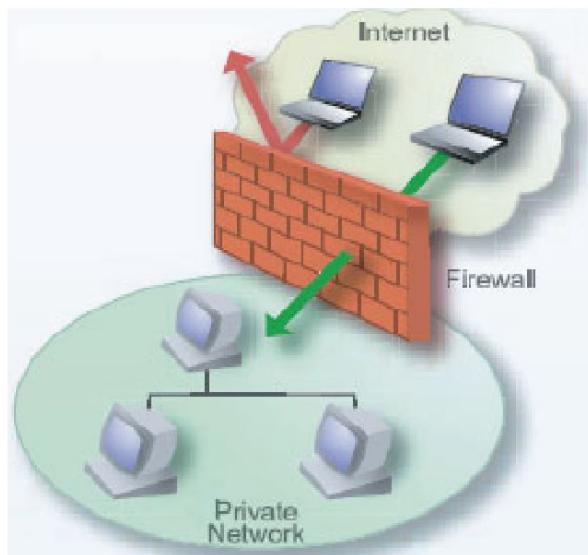
## Firewall

Isola a rede interna da organização da área pública da Internet, permitindo que alguns pacotes passem e outros não.



# Firewalls

- É um mecanismo de proteção que controla a passagem de pacotes entre redes, tanto locais como externas
- É um dispositivo que possui um conjunto de regras especificando que tráfego ele permitirá ou negará



# Firewalls: por quê?

Previne ataques de negação de serviço:

- Inundação de SYN: atacante estabelece muitas conexões TCP falsas, esgota os recursos para as conexões “reais”

Previne modificações e acessos ilegais aos dados internos

- Ex.: o atacante substitui a página da CIA por alguma outra coisa

Permite apenas acesso autorizado à rede interna (conjunto de usuários e hospedeiros autenticados)

Três tipos de firewalls:

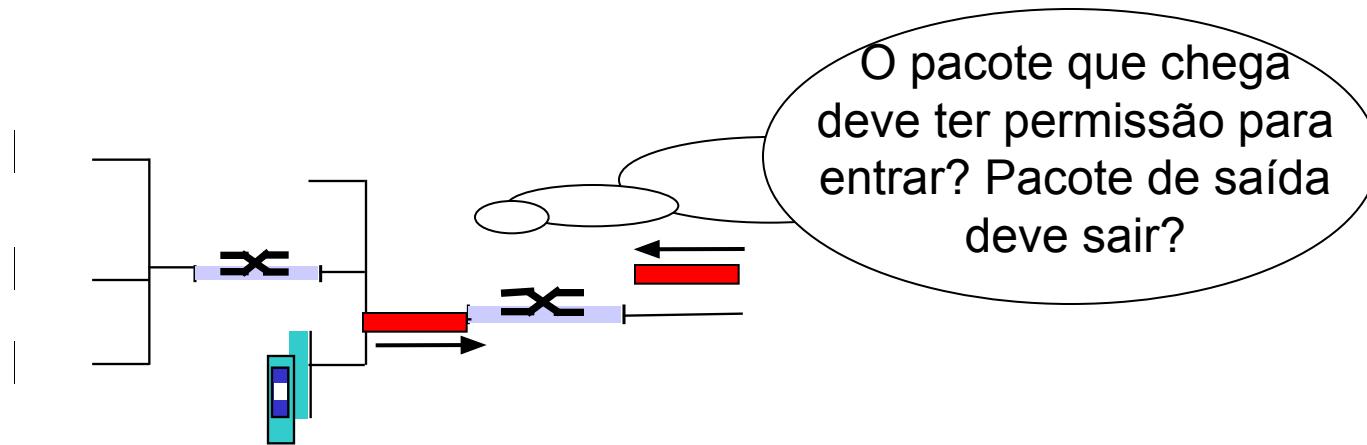
- Nível de aplicação
- Filtro de pacotes sem estado
- Filtro de pacotes com estado



# Firewalls

- **Política Padrão**
    - Tudo é permitido, exceto o que for expressamente proibido
    - Tudo é proibido, exceto o que for expressamente permitido
  - **Política**
    - [ACCEPT | DROP | REJECT] Protocolo [TCP | UDP | ICMP ]  
Endereço Origem [SA=ipaddr/msk] Porta Origem [SP] Endereço  
Destino [DA=ipaddr/msk] Porta Destino [DP]
  - **Filtros**
    - IP de origem e de destino
    - Protocolo TCP, UDP e ICMP
    - Portas de origem e de destino
    - Flags SYN ou ACK
-

# Filtragem de pacotes sem estado



- rede interna conectada à Internet via **firewall do roteador**
- roteador **filtra pacote-por-pacote**, decisão de repassar/descartar pacote com base em:
  - endereço IP de origem, endereço IP de destino
  - números de porta de origem e destino do TCP/UDP
  - tipo de mensagem ICMP
  - bits SYN e ACK do TCP

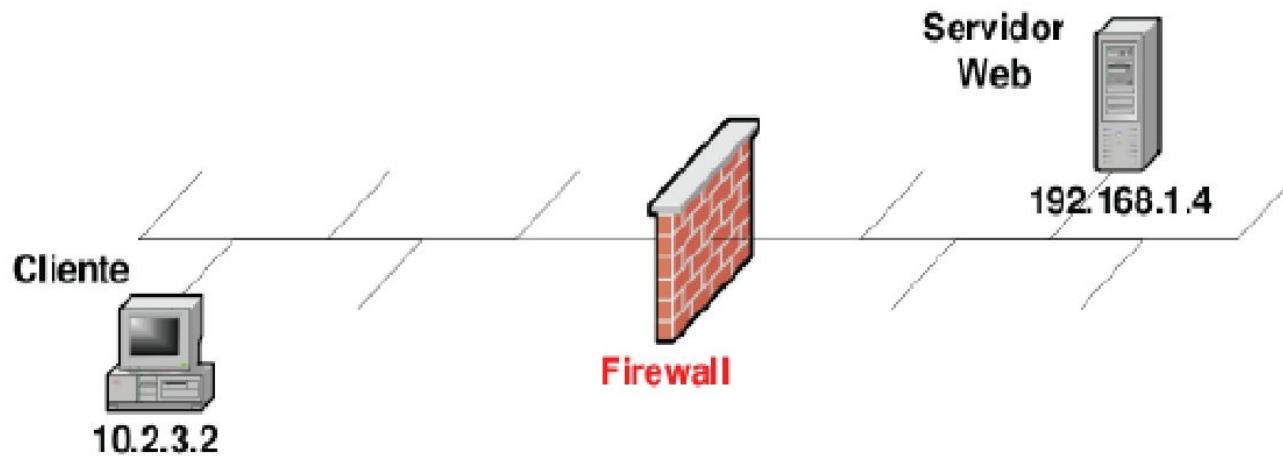
# Filtro de pacotes (cont.)

- Exemplo 1: bloqueia datagramas que chegam e que saem com campo de protocolo = 17 e com porta de destino ou de origem = 23
  - Todos os fluxos UDP que entram e que saem e as conexões Telnet são bloqueadas
- Exemplo 2: bloqueia segmentos TCP entrantes com ACK = 0
  - Evita que os clientes externos façam conexões com clientes internos, mas permite que os clientes internos se conectem para fora

Iptables: Implementação de *firewall* via *software* (Kenel Linux >= 2.4): <http://www.netfilter.org/projects/iptables/index.html>  
<http://focalinux.cipsqa.org.br/quia/avancado/ch-fw-iptables.htm>

---

# Filtro de Pacotes



Regras de Filtagem de

Pacotes:

**ACCEPT TCP SA=10.2.3.2 SP>1024 DA=192.168.1.4**

**DP=80**

**ACCEPT TCP SA=192.168.1.4 SP=80 DA=10.2.3.2 DP>1024**

**DROP ALL SA=0.0.0.0 ALL DA =0.0.0.0 ALL**

**SA - Source Address**

**SP - Source Port**

**DA - Destination Address**

**SA - Source Address**

# Filtragem de pacotes sem estado: mais exemplos

<u>Política</u>	<u>configuração de firewall</u>
sem acesso externo à Web	descarta todos os pacotes que saem para qualquer endereço IP, porta 80
sem conexões TCP entrando, exceto aquelas apenas para o servidor Web público da instituição	descarta todos pacotes TCP SYN que chegam a qualquer IP, exceto 130.207.244.203, porta 80
impedir que Web-radios devorem a largura de banda disponível	descarta todos os pacotes UDP que chegam - exceto DNS e broadcasts do roteador
impedir que sua rede seja usada para um ataque DoS	descarta todos os pacotes ICMP indo para um endereço de "broadcast" (p. e., 130.207.255.255)
impedir que sua rede interaja com o programa Traceroute	descarta todo tráfego expirado ICMP TTL de saída

# Filtragem de pacotes sem estado

- **Vantagens:**
    - Barato, simples e flexível;
    - Alto desempenho da rede;
    - Transparente para o usuário.
  - **Desvantagens:**
    - Permite a conexão direta para hosts internos de clientes externos;
    - Difícil de gerenciar em ambientes complexos;
    - Não oferece autenticação de usuários.
-

# Listas de controle de acesso

- **ACL:** tabela de regras, aplicadas de cima para baixo aos pacotes que chegam: pares (ação, condição)

Ação	Endereço de origem	Endereço de destino	Protocolo	Porta de origem	Porta de destino	Flag bit
Permitir	222.22/16	Fora de 222.22/16	TCP	>1023	80	Qualquer um
Permitir	Fora de 222.22/16	222.22/16	TCP	80	>1023	ACK
Permitir	222.22/16	Fora de 222.22/16	UDP	>1023	53	_
Permitir	Fora de 222.22/16	222.22/16	UDP	53	>1023	_
Negar	Todos	Todos	Todos	Todos	Todos	Todos

# Filtragem de pacotes com estado

- filtro de pacotes sem estado:
  - admite pacotes que “não fazem sentido”, p. e., porta destino = 80, bit ACK marcado, mesmo sem conexão TCP estabelecida:

ação	endereço de origem	endereço de destino	protocolo	porta de origem	porta de destino	bit de flag
permitir	fora de 222.22/16	222.22/16	TCP	80	> 1023	ACK

- *filtro de pacotes com estado:* rastreia status de cada conexão TCP
  - rastrear configuração de conexão (SYN), encerramento (FIN): pode determinar se pacotes de entrada e saída “fazem sentido”
  - timeout de conexões inativas no firewall: não admite mais pacotes

# Tabela de Conexão para Filtro de Estado

Endereço de Origem	Endereço de Destino	Porta de Origem	Porta de Destino
222.22.1.7	37.96.87.123	12699	80
222.22.93.2	199.1.205.43	37654	80
222.22.65.143	203.77.240.43	48712	80

---

- ACL aumentada para indicar necessidade de verificar tabela de estado da conexão antes de admitir pacote

Ação	Endereço de origem	Endereço de destino	Protocolo	Porta de origem	Porta de destino	Flag bit	Conexão de checagem
Permitir	222.22/16	Fora de 222.22/16	TCP	>1023	80	Qualquer um	
Permitir	Fora de 222.22/16	222.22/16	TCP	80	>1023	ACK	X
Permitir	222.22/16	Fora de 222.22/16	UDP	>1023	53	—	
Permitir	Fora de 222.22/16	222.22/16	UDP	53	>1023	—	X
Negar	Todos	Todos	Todos	Todos	Todos	Todos	

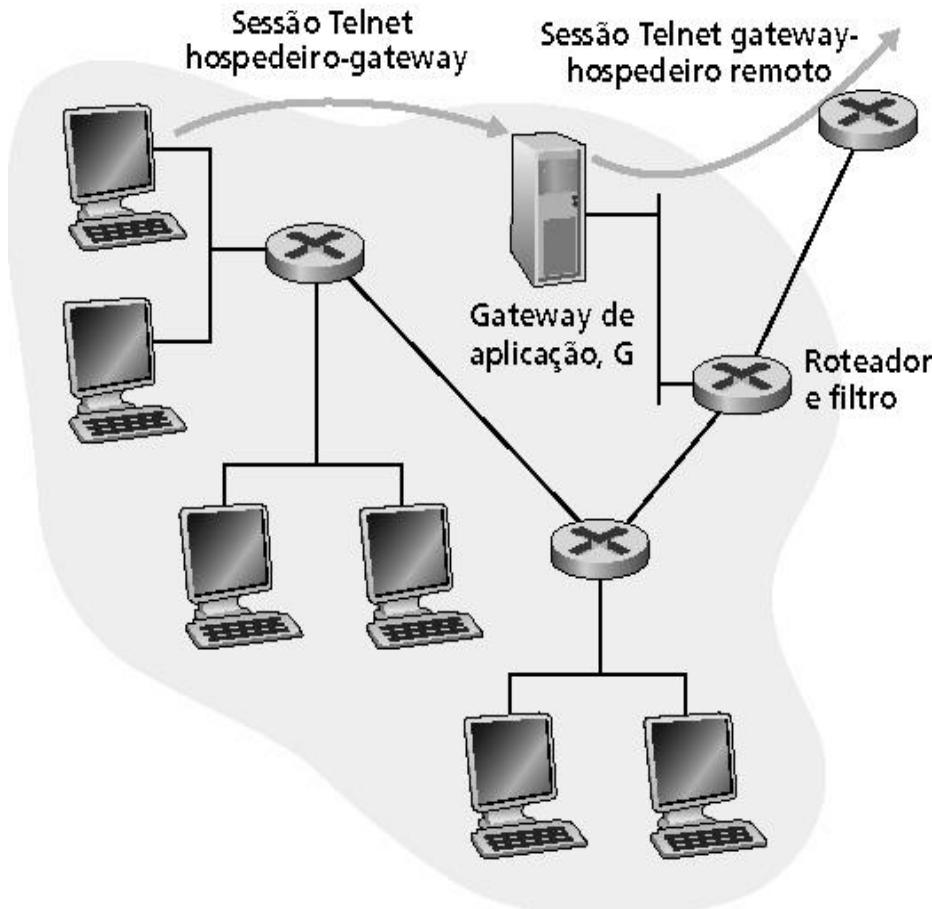
O *firewall* verificará a Tabela de Conexões TCP e somente permitirá a entrada de pacotes com bit ACK = 1 se forem provenientes dos endereços externos contidos na tabela e destinados aos respectivos endereços internos.

# Filtro de Estado das Conexões

---

- **Vantagens:**
  - Alto desempenho da rede;
  - Aceita quase todos os tipos serviços;
  - Transparente para o usuário.
- **Desvantagens:**
  - Permite a conexão direta para hosts internos de clientes externos,
  - Não oferece autenticação de usuários.

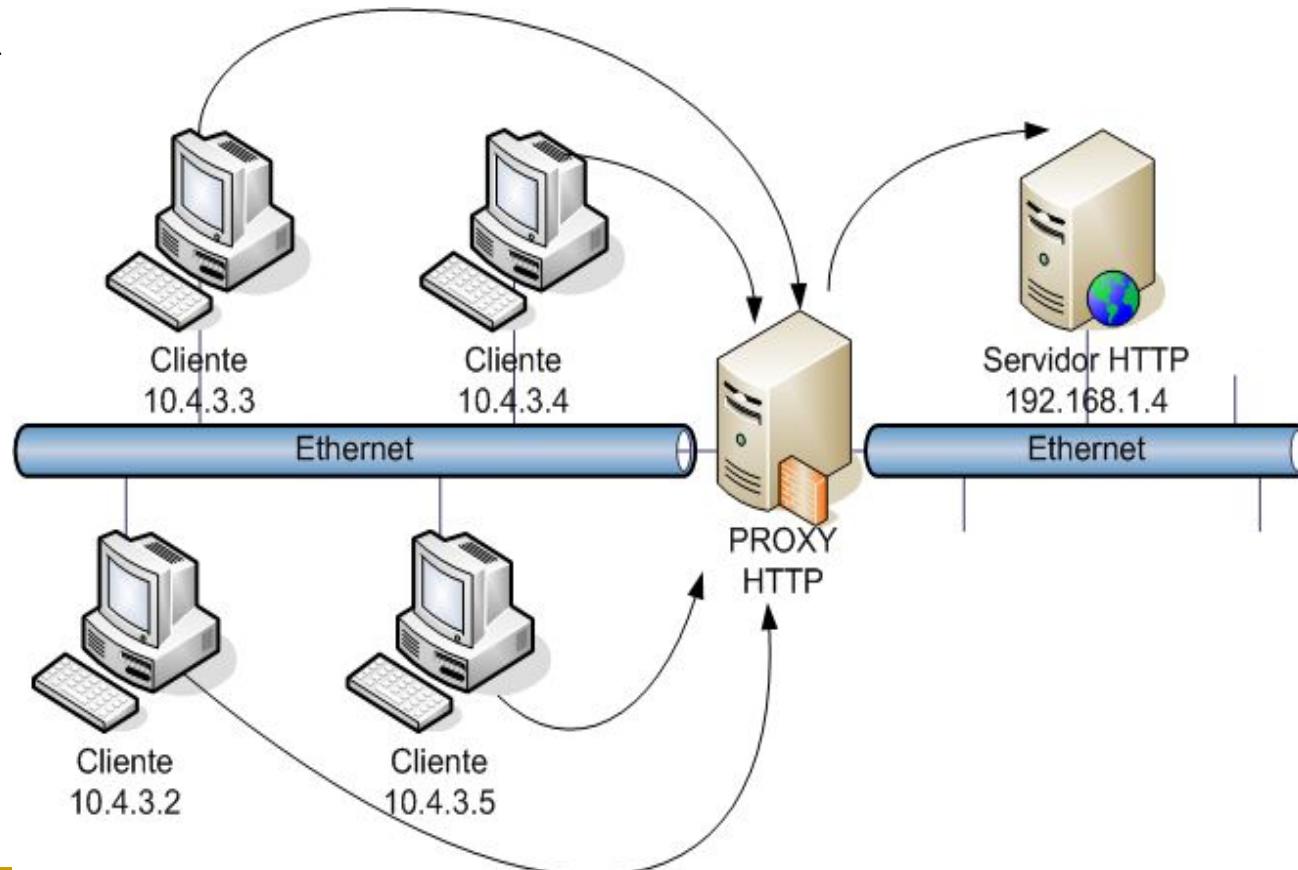
# Gateways de aplicação



- Filtra pacotes em função de dados de aplicação, assim como de campos do IP/TCP/UDP
  - **Exemplo:** permite selecionar usuários internos que podem usar o Telnet
1. Exige que todos os usuários Telnet se comuniquem através do gateway
  2. Para os usuários autorizados, o gateway estabelece conexões Telnet com o hospedeiro de destino. O gateway repassa os dados entre as duas conexões
  3. O filtro do roteador bloqueia todas as sessões Telnet que não se originam no gateway

# Gateway de Aplicação - Proxy

Os servidores *proxy* recebem as solicitações dos usuários e as encaminha para os respectivos servidores web. Analogamente, as respostas do servidores web são repassadas pelo proxy aos respectivos clientes



Exemplo: *Squid* - <http://www.squid-cache.org/>

# Proxy de Serviços Funcionalidades

---

- Prover Anônimo;
- Autenticação;
- Armazena dados em forma de cache;
- Filtrar conteúdo (Camada de Aplicação).

# Proxy não é Filtro de Pacotes

---

- Neste contexto, o proxy de serviço roda em uma máquina com duas interfaces de rede;
- Entretanto, diferentemente do filtro de pacotes, o proxy não realiza roteamento dos datagramas IP.
- Desta forma, não é necessário criar regras de filtragem dentro do proxy de serviços pois as duas redes conectadas ao proxy não são visíveis entre si.
- Geralmente, se configura um proxy por aplicação.

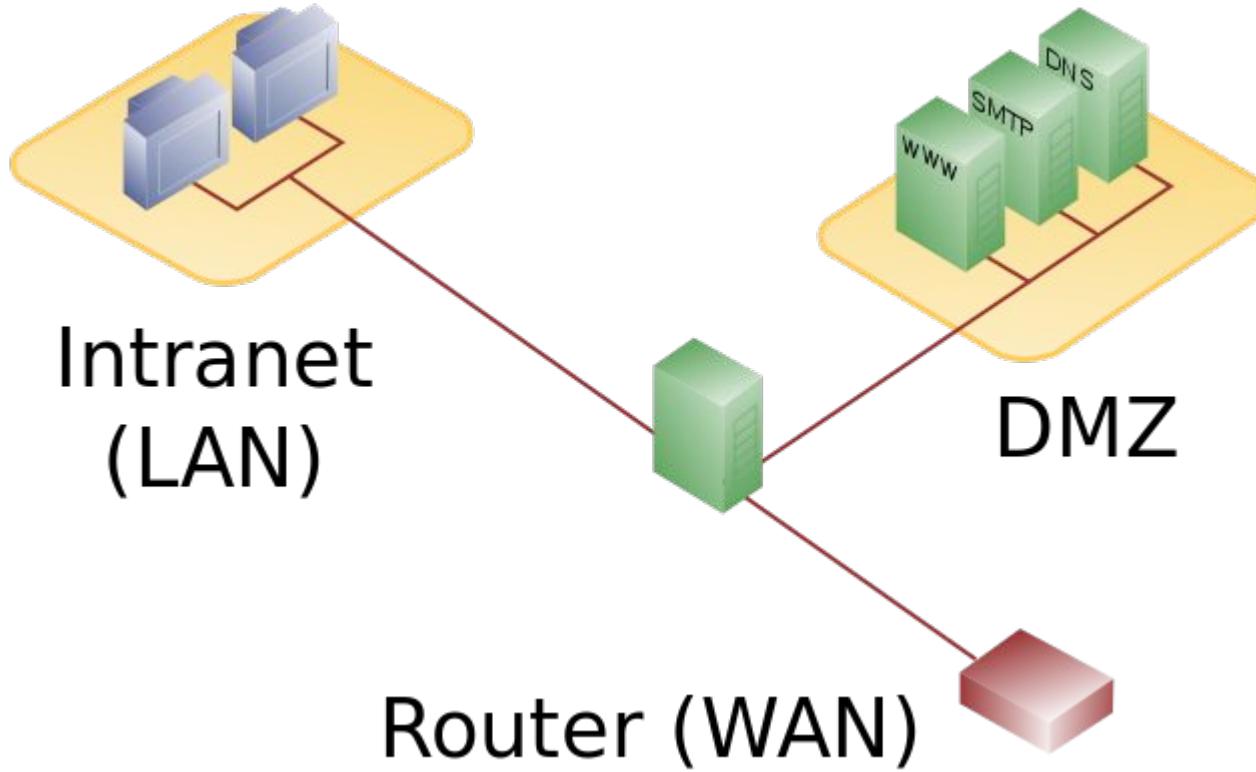
# Proxy de Serviços

---

- **Vantagens:**
  - Não permite conexões diretas entre hosts internos e hosts externos;
  - Aceita autenticação do usuário;
  - Analisa comandos da aplicação no payload dos pacotes de dados, ao contrário do filtro de pacotes.
- **Desvantagens:**
  - Mais lento do que os filtros de pacotes;
  - Requer um proxy específico pra cada aplicação;
  - Não trata pacotes ICMP;
  - Não possui transparência.

# DMZ – *De Militarized Zone*

---



# DMZ – *De Militarized Zone*

---

- Manter todos os serviços que possuem acesso externo (HTTP, FTP, de correio eletrônico, etc) separados da rede local
- Limitar danos potencial após comprometimento de alguns desses serviços
- Computadores da DMZ não podem conter acesso à rede local
- O *firewall* implementa a separação entre rede local, a *Internet* e a DMZ

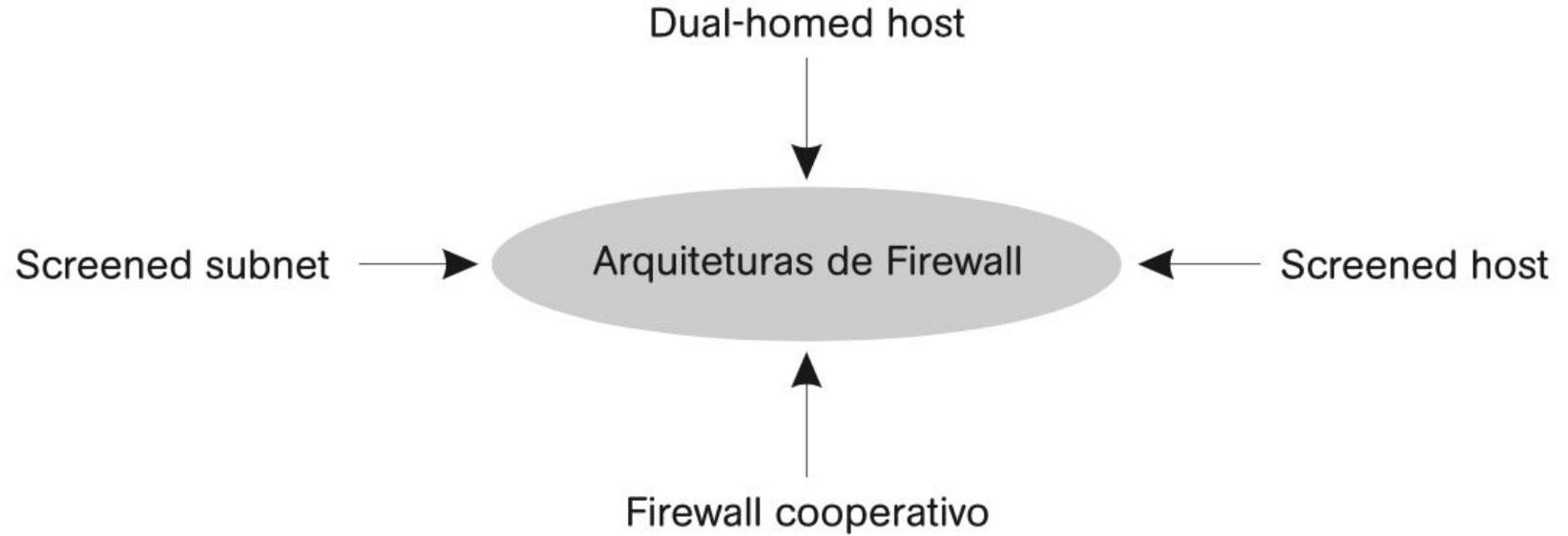
# DMZ – *De Militarized Zone*

---

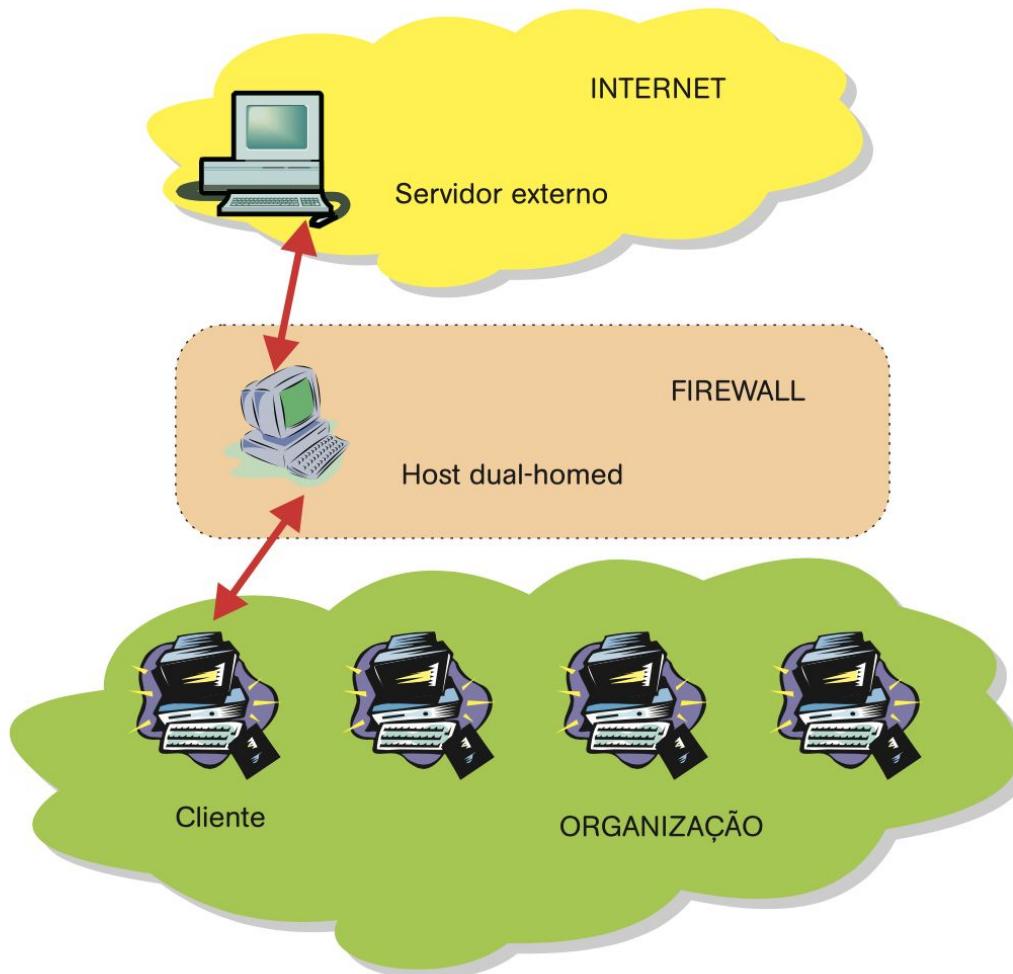
- A rede interna pode iniciar conexões a qualquer uma das outras redes, mas nenhuma das outras redes pode iniciar conexões nesta.
- A rede pública (internet) não pode iniciar conexões na rede interna, mas pode na DMZ.
- A DMZ não pode fazer conexões à rede interna mas pode na rede pública.

## Arquiteturas

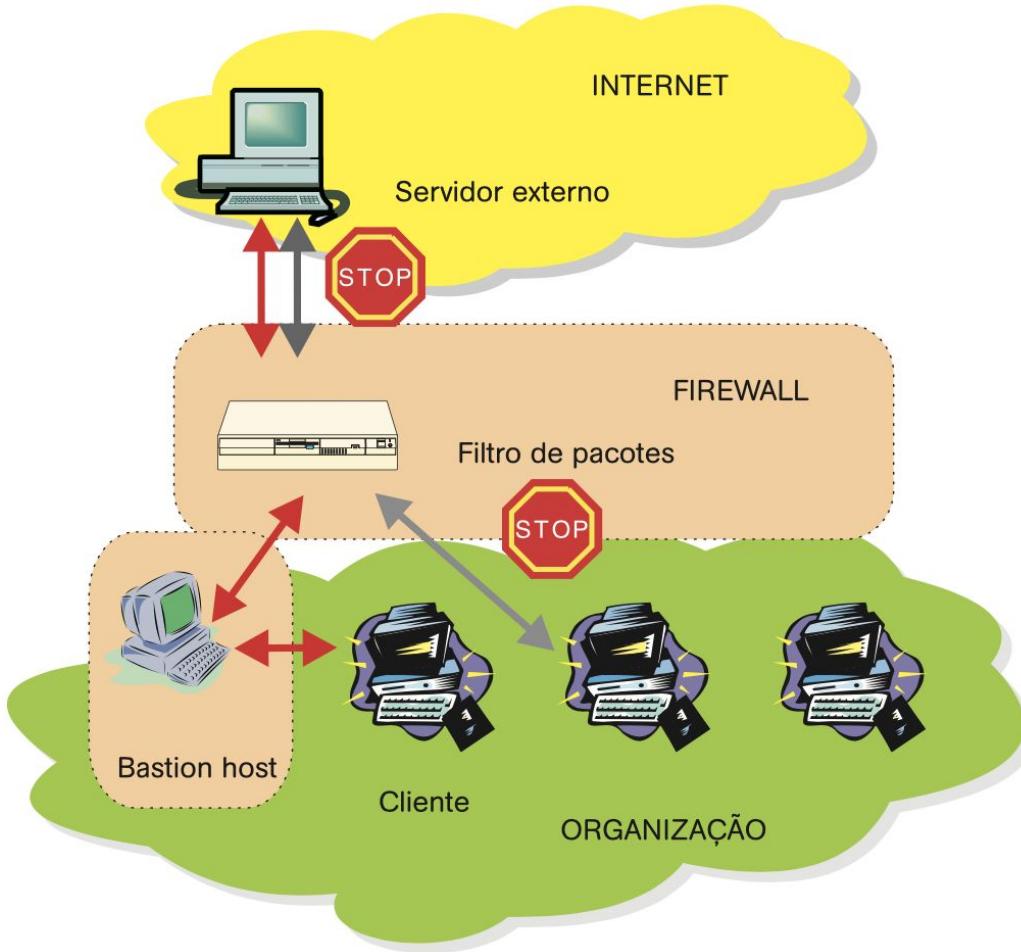
---



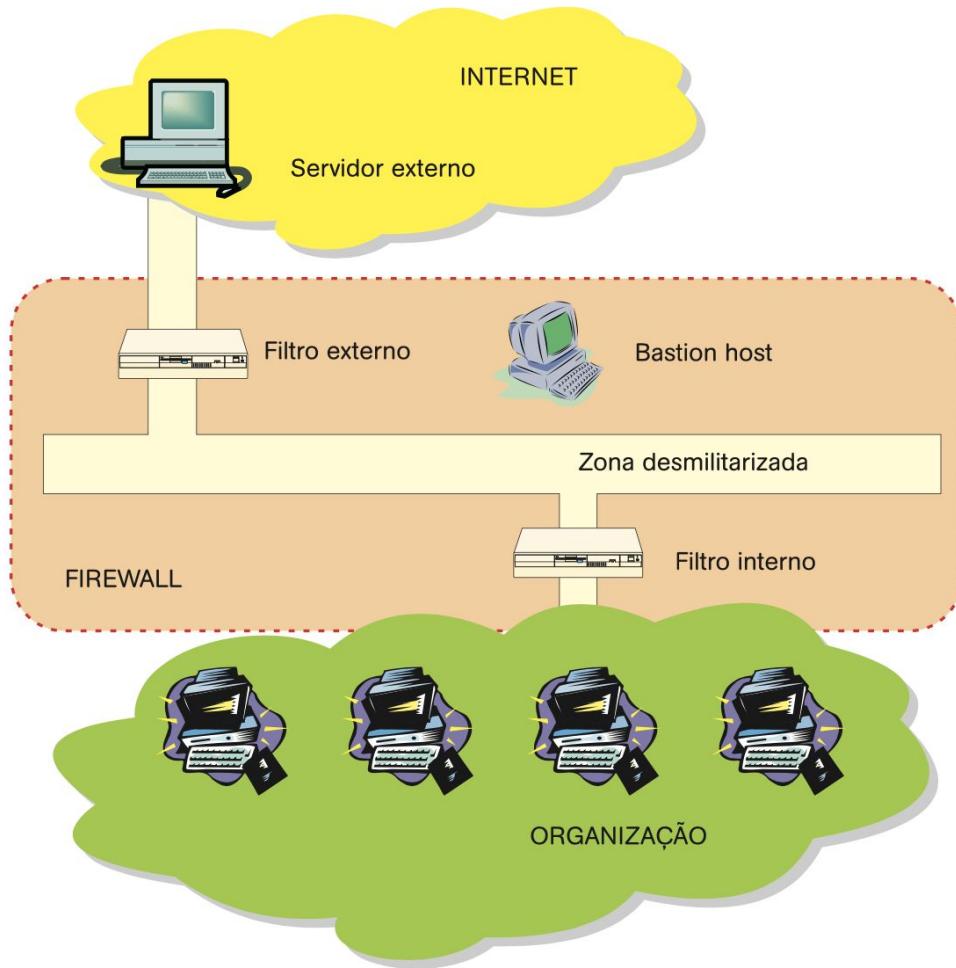
## Host dual-homed



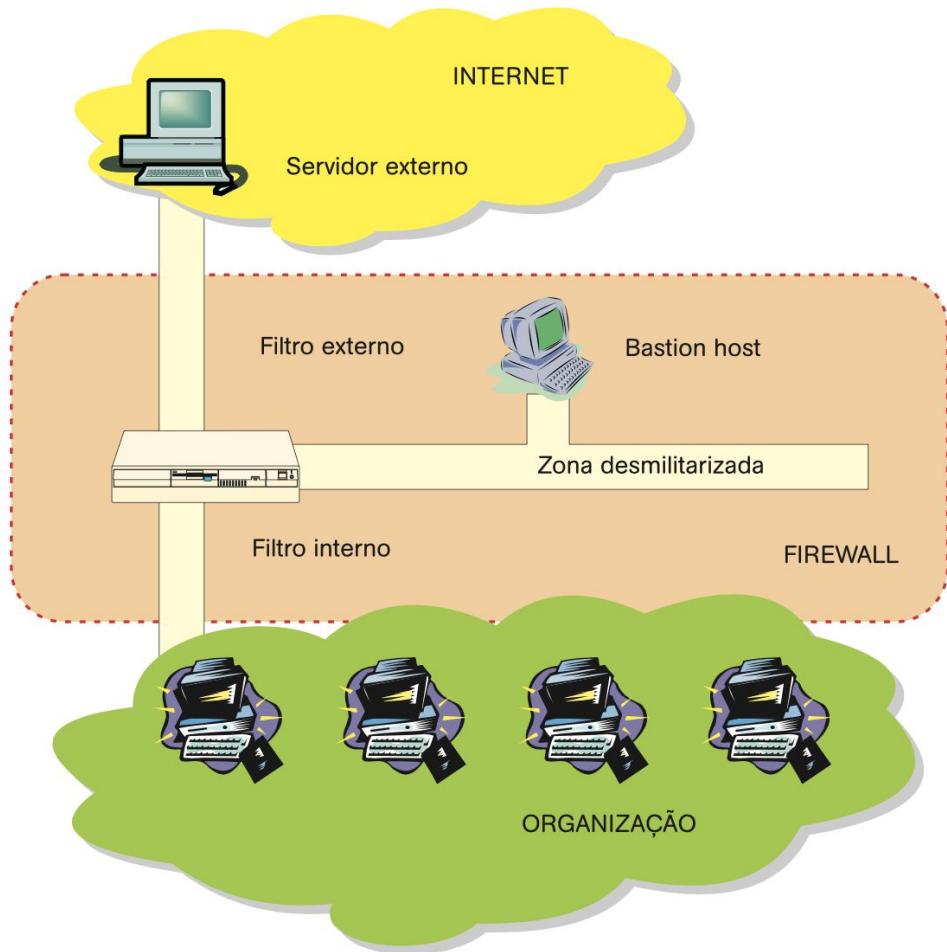
## Screened host architecture



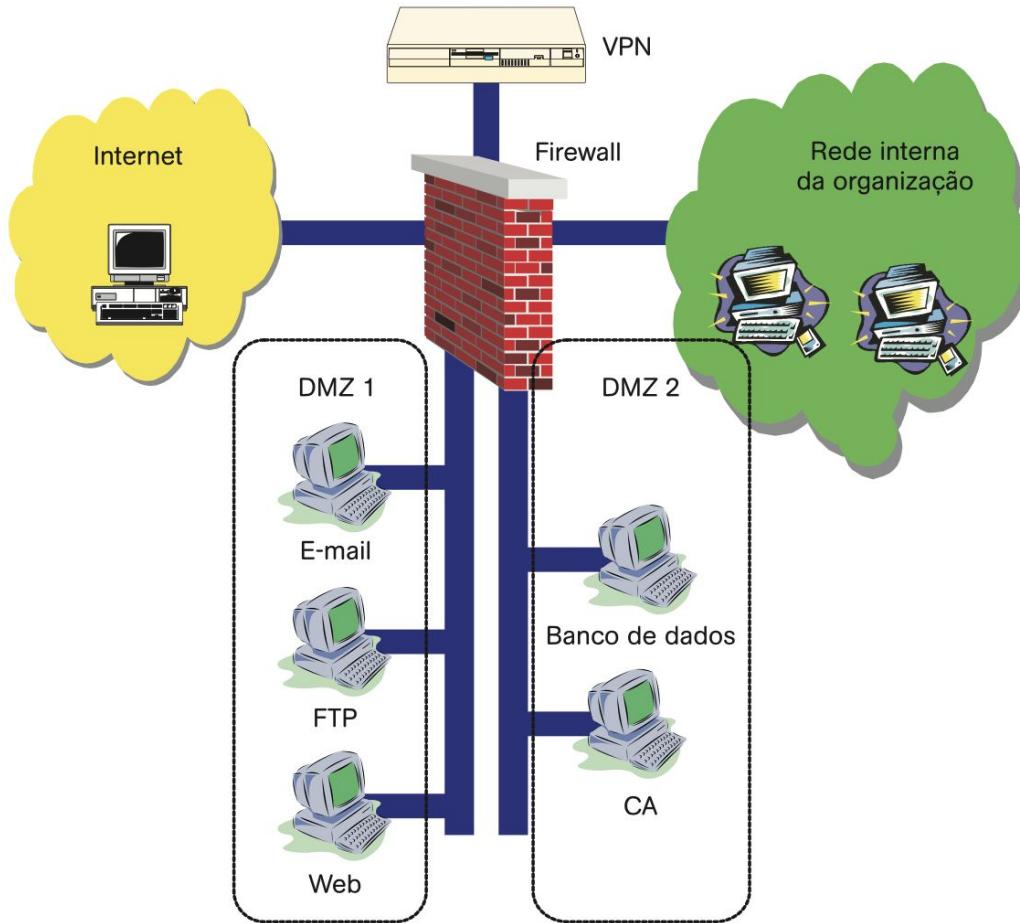
## Screened subnet architecture



## Screened subnet architecture



# Firewall Corporativo



# Zona Desmilitarizada (DMZ)

---

- A utilização de firewall para se conectar uma rede à Internet possibilitou uma topologia de acesso interessante e segura.
- Na figura a seguir, tem-se uma rede composta por máquinas clientes, servidores de serviços de Intranet (acessados pelos clientes internos) e servidores de serviços Internet (acessados pela Internet).

# Zona Desmilitarizada (DMZ)

---

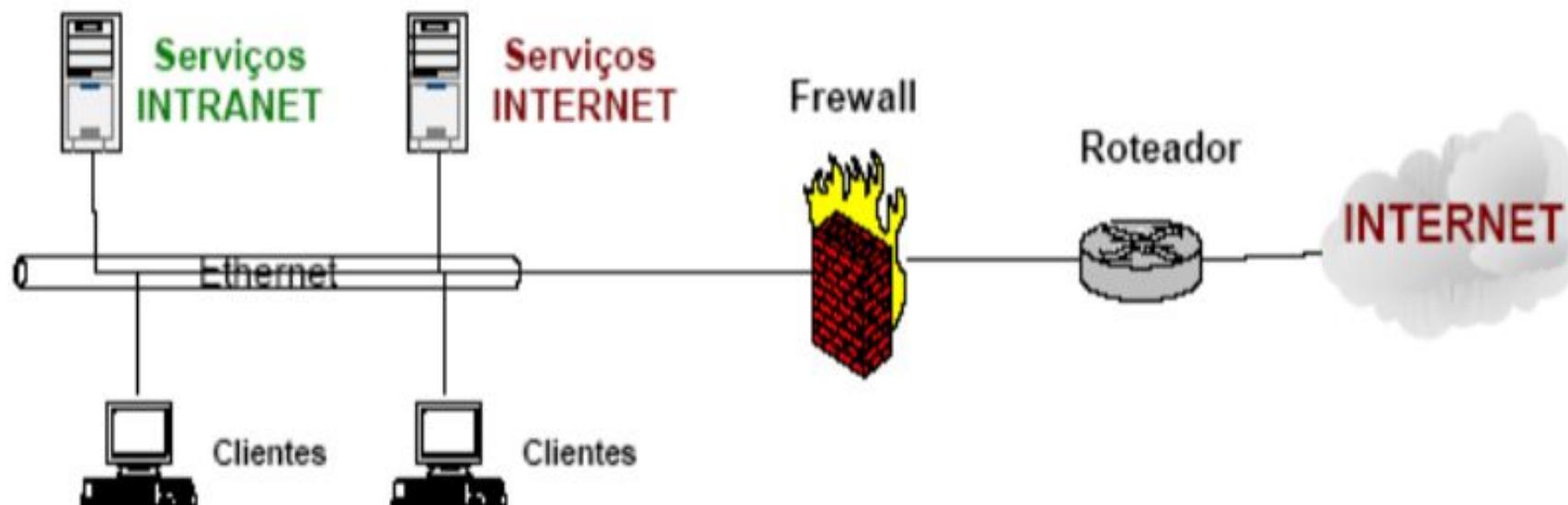
- A utilização de firewall para se conectar uma rede à Internet possibilitou uma topologia de acesso interessante e segura.
- Na figura a seguir, tem-se uma rede composta por máquinas clientes, servidores de serviços de Intranet (acessados pelos clientes internos) e servidores de serviços Internet (acessados pela Internet).

# Zona Desmilitarizada (DMZ)

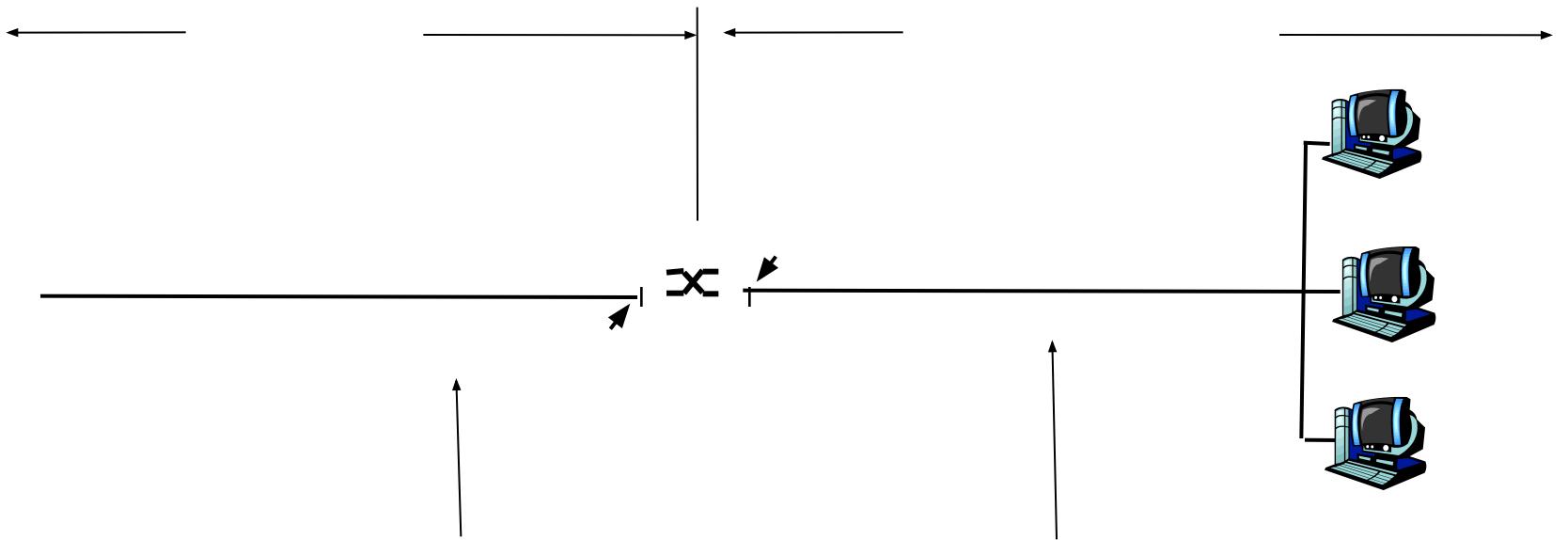
---

- A utilização de firewall para se conectar uma rede à Internet possibilitou uma topologia de acesso interessante e segura.
- Na figura a seguir, tem-se uma rede composta por máquinas clientes, servidores de serviços de Intranet (acessados pelos clientes internos) e servidores de serviços Internet (acessados pela Internet).

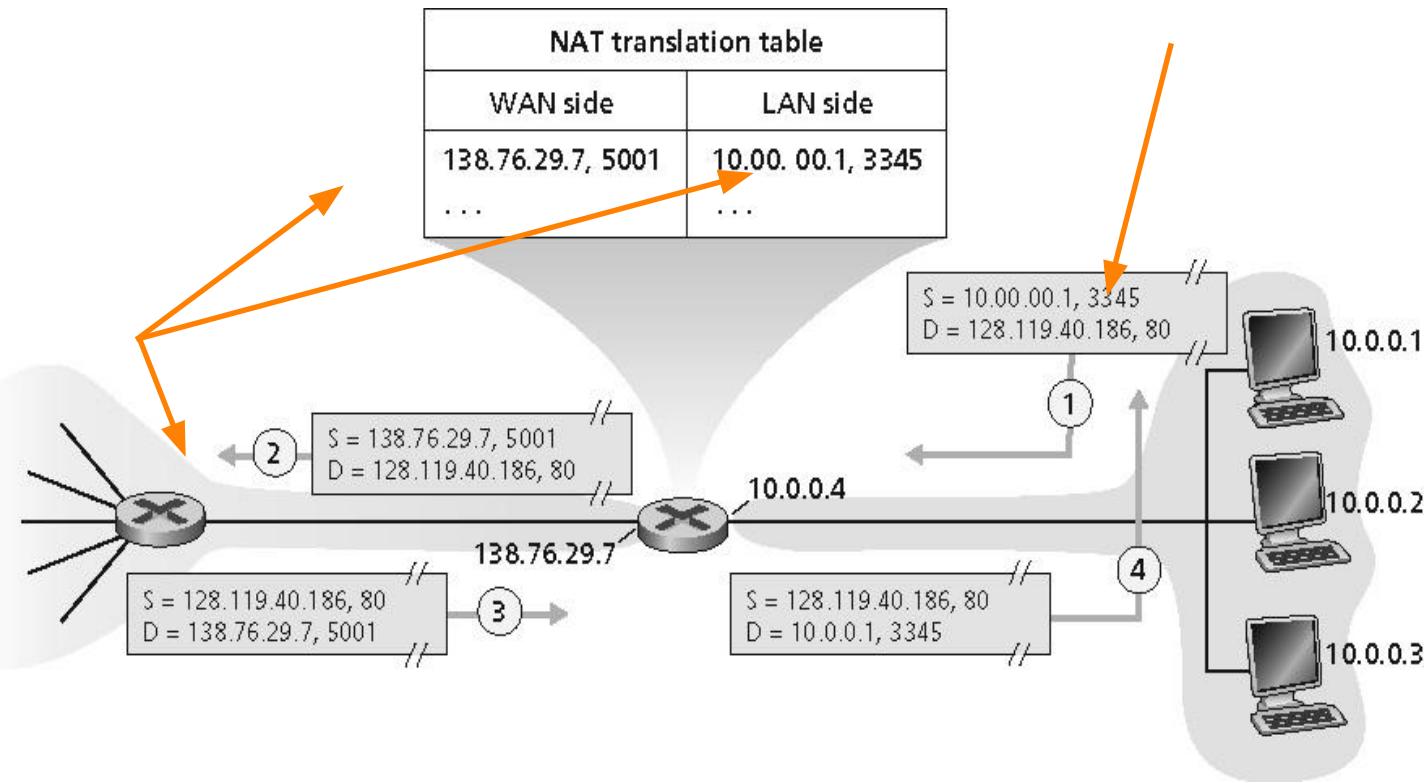
## REDE DA EMPRESA











# Limitações de firewalls e gateways

- **falsificação de IP:** roteador não sabe se os dados “realmente” vêm de fonte alegada
- se múltiplas aplicações precisam de tratamento especial, cada uma tem gateway próprio.
- software cliente deve saber como contatar gateway.
  - p. e., deve definir endereço IP do proxy no servidor Web



# Limitações de firewalls e gateways

- **falsificação de IP:** roteador não sabe se os dados “realmente” vêm de fonte alegada
- se múltiplas aplicações precisam de tratamento especial, cada uma tem gateway próprio.
- software cliente deve saber como contatar gateway.
  - p. e., deve definir endereço IP do proxy no servidor Web



# Sistemas de detecção de invasão

- IDS – deixa o tráfego suspeito passar, mas envia alertas ao administrador para investigação futura
  - IPS (Sistema de Prevenção de Intrusão/*Intrusion Prevention System*) – barra todo o tráfego suspeito
-

# Tipos de IDS

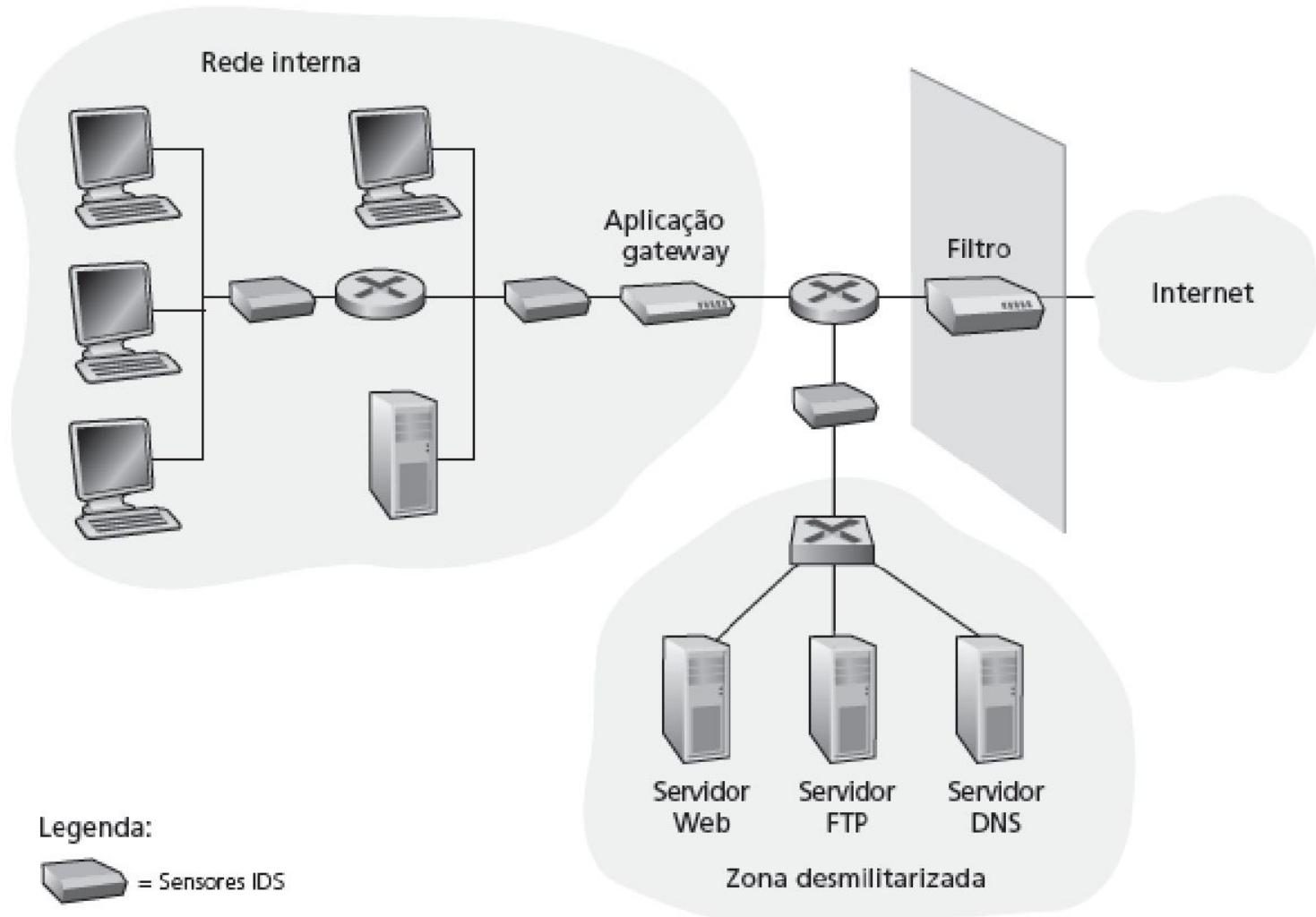
- Baseados em Assinatura
    - Mantém um banco de dados de assinaturas de ataques
      - Trechos de códigos usados por virus
      - Números de porta de origem/destino
      - Sequência de bits em um pacote
    - Vantagem
      - Gera poucos alertas falsos
    - Desvantagem
      - Conhecimento prévio – cego a novos ataques
      - Baixo desempenho pois tem que comparar cada pacote com uma grande variedade de assinaturas
-

# Tipos de IDS

## ■ Baseados em Anomalia

- Assume que toda a atividade intrusa é necessariamente anômala.
- Se pudermos estabelecer um perfil de atividade normal para um sistema, poderemos teoricamente, comparar todos os demais perfis com o estabelecido, traçando um quadro de atividade que foge do padrão
- Ex: Porcentagem irregular de pacotes ICMP, Tráfego muito pesado para o horário, tipos de protocolos em uso etc.
- Vantagem
  - Pode descobrir ataques novos
- Desvantagem
  - Geração de Falsos positivos e Falsos Negativos
  - Difícil determinar a normalidade

- múltiplos IDSs: diferentes tipos de verificação em diferentes locais



# Exemplo de IDS: Snort

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any  
(msg: "ICMP PING MAP"; dsize: 0; itype: 8;)
```

<http://www.snort.org/>

<http://www.snort.com.br/>



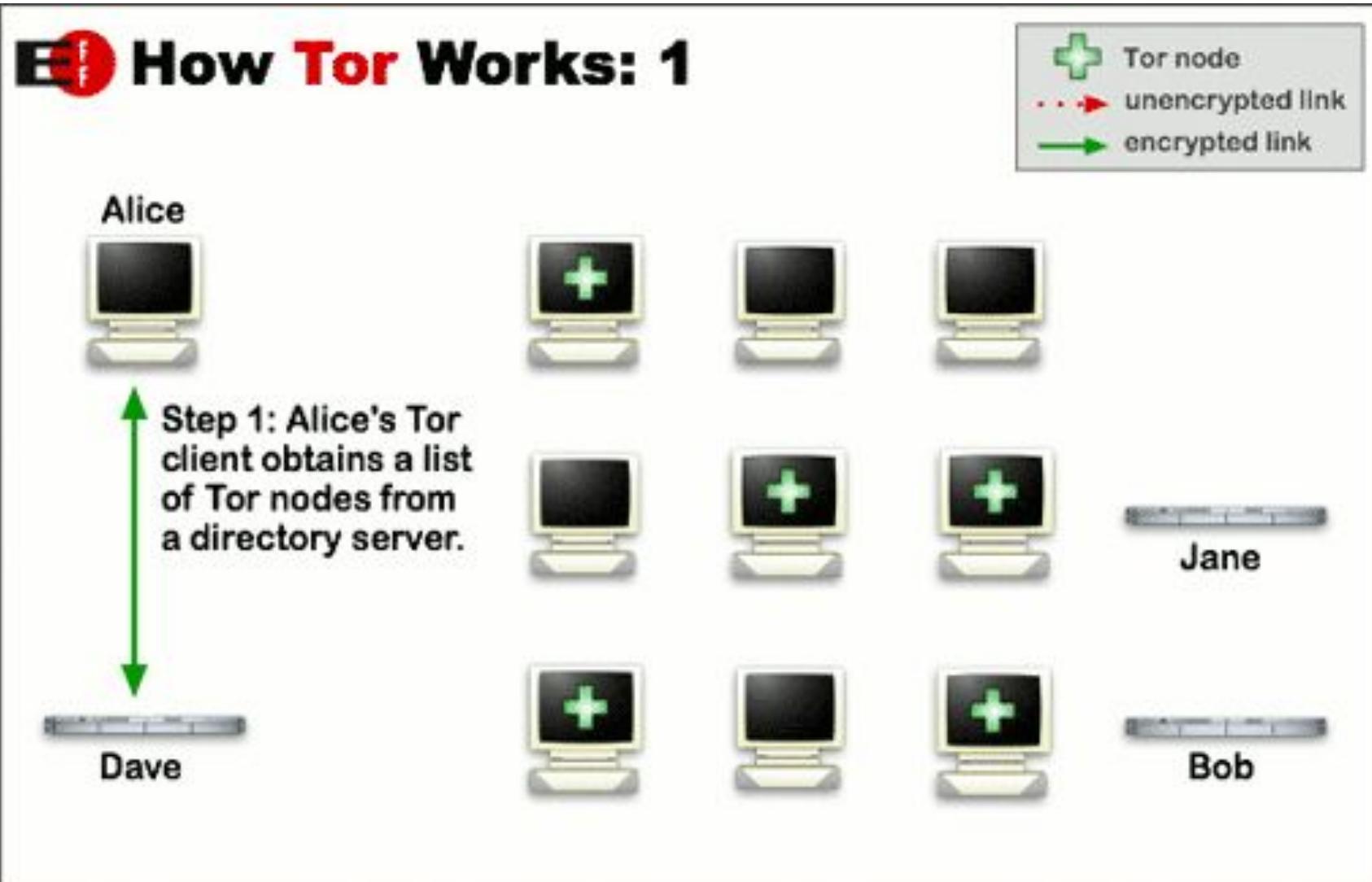
# Pacotes - ICMP

Tipo	Cód,	Descrição
0	0	resposta de eco (ping)
3	0	rede de destino inalcançável
3	1	hosp. de destino inalcançável
3	2	protocolo de destino inalcançável
3	3	porta de destino inalcançável
3	6	rede de destino desconhecida
3	7	hosp. de destino desconhecido
4	0	redução da fonte (controle de congestionamento – não usado)
8	0	solicitação de eco (ping)
9	0	anúncio de rota
10	0	descoberta do roteador
11	0	TTL expirado
12	0	cabeçalho IP inválido

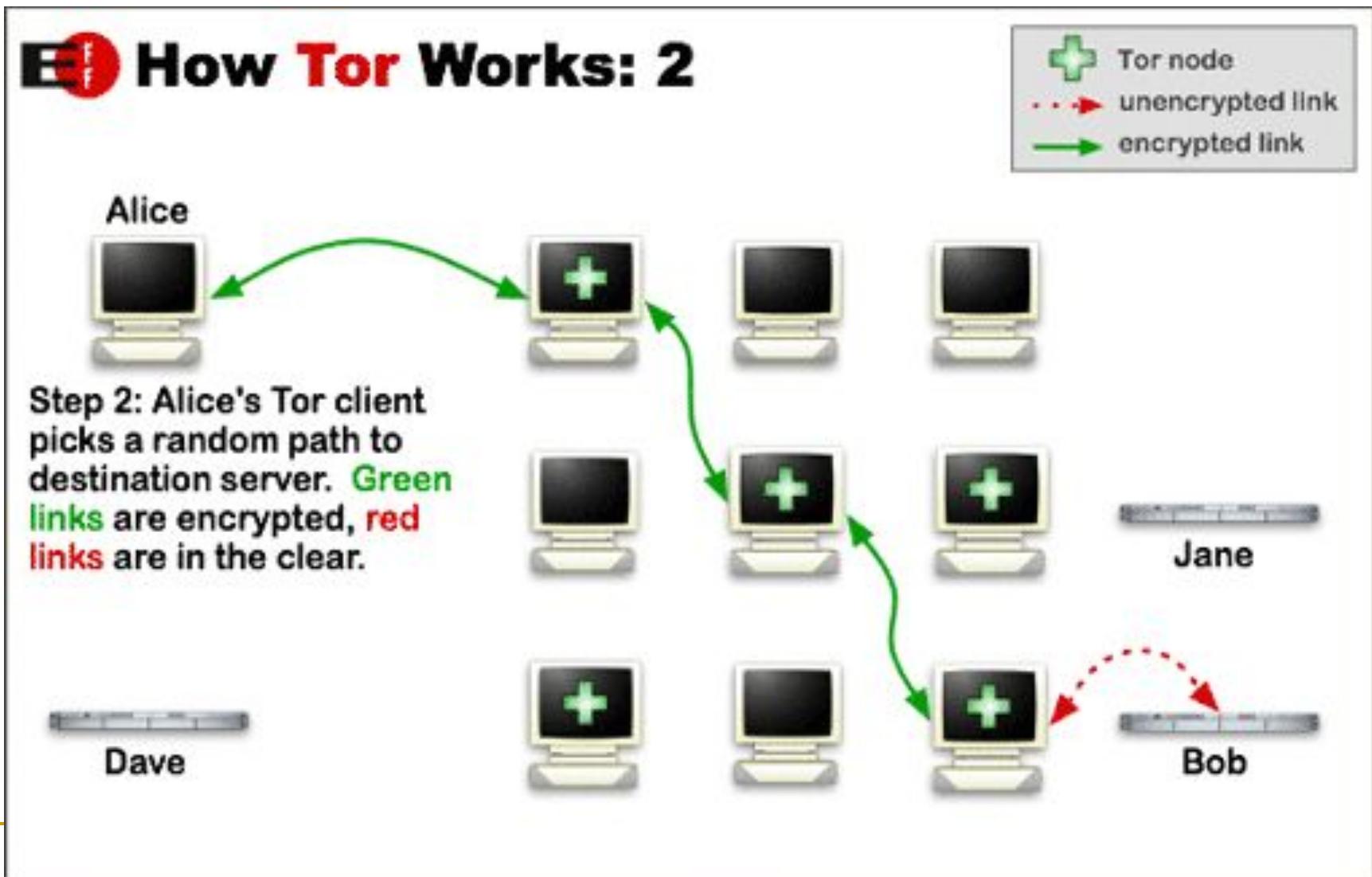
# Anonimato

- TOR - <http://www.torproject.org/>
    - Cadeia de proxies que esconde o cliente do servidor
    - Como os proxies não compartilham informações entre si, o ultimo não sabe o endereço do cliente e o primeiro proxy não sabe o endereço do servidor
  - + Privacidade
    - Utilizar conexões SSL entre o cliente e a cadeia de proxies
-

# TOR



# TOR



# TOR

