Instituto de computação

Departamento de Ciência da Computação
Instituto de Computação
Universidade Federal Fluminense

Aulas 15, 16, 17 e 18

Segurança de Redes

Conceitos, criptografia, segurança em diferentes camadas, *firewalls* e sistemas de detecção de intrusão

Igor Monteiro Moraes

Redes de Computadores II

ATENÇÃO!

Instituto de computação

- Este apresentação é contém partes baseadas nos seguintes trabalhos
  - Notas de aula do Prof. Marcelo Rubinstein, disponíveis em http://www.lee.eng.uerj.br/~rubi
  - Notas de aula do Prof. José Augusto Suruagy Monteiro, disponíveis em http://www.nuperc.unifacs.br/Members/jose.suruagy/cursos
  - Material complementar do livro Computer Networking: A Top Down Approach, 5th edition, Jim Kurose and Keith Ross, Addison-Wesley, abril de 2009
  - Computer Networks, Andrew S. Tanenbaum, 4a. Edição, Ed. Prentice Hall
  - Vilela, U. C., Cardoso, K. V. e de Rezende, J. F. - "Redes 802.11 em Centros Urbanos: Varredura, Estatísticas e Aplicações" - in VI Workshop em Desempenho de Sistemas Computacionais e de Comunicação - WPerformance'2007 (XXVII Congresso da Sociedade Brasileira de Computação - CSBC 2007), pp. 703-718, junho de 2007.

Ameaças na Internet

Instituto de computação

- Segurança de computadores
  - Intrusos
  - Pragas digitais
- Segurança em redes de computadores
  - Confidencialidade
  - Integridade
  - Autenticação
  - Não-repúdio
  - Controle de acesso
  - Disponibilidade

Segurança de Computadores

Instituto de computação

- Violações por pessoas ou "intrusos"
- Violações por software ou pragas digitais

- Exemplos
  - Pragas digitais
    - Vírus, vermes, cavalos de Tróia, etc.
  - Roubo de informações confidenciais
  - Mensagens não solicitadas (*spams*)
  - Etc.

Segurança em Redes

Instituto de computação

- Confidencialidade
  - Proteção do conteúdo das mensagens
    - Somente o emissor e o receptor pretendido devem "entender" o conteúdo da mensagem
      - Emissor cifra a mensagem
      - Receptor decifra a mensagem
  - Proteção da estatística do tráfego
    - Endereços fonte e destino, freqüência, comprimento etc.

Segurança em Redes

Instituto de computação

- Integridade
  - Proteção do conteúdo das mensagens contra alterações ou destruição
    - Emissor e receptor desejam assegurar que a mensagem não é alterada, em trânsito ou após sua recepção, sem que essa alteração seja detectada
- Autenticação
  - Garantia de que o emissor é, de fato, quem diz ser (autêntico)
    - Emissor e receptor desejam confirmar a identidade de um e de outro

Segurança em Redes

Instituto de computação

- Não-repúdio
  - Impede que emissor negue o envio e o receptor negue o recebimento das mensagens
- Controle de acesso
  - Restrição e controle do acesso a sistemas e aplicações
- Disponibilidade
  - Garantia da manutenção da capacidade de um sistema de realizar suas atividades
    - Serviços devem estar disponíveis para os usuários

Segurança em Redes

Instituto de computação

- Exemplos
  - Roubo de informações confidenciais
  - Mensagens não solicitadas (*spams*)
  - Negação de serviço
  - Etc.

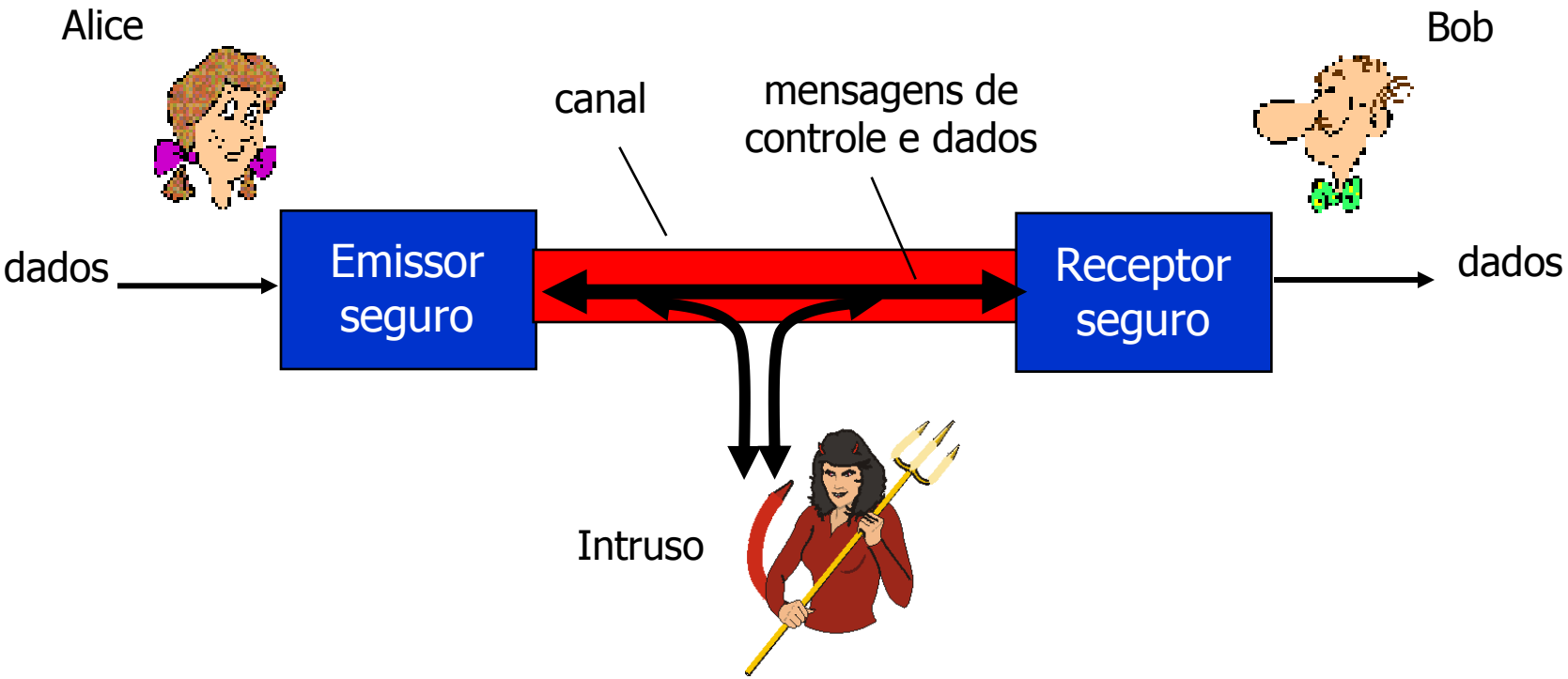
Segurança em Redes

Instituto de computação

- Para cada tipo de ataque existe
  - Um modelo de atacante
  - Uma ou mais soluções para um dado ataque, considerando um modelo de atacante
    - Ex.: Ataques de negação de serviço (*Denial of Service* – DoS)
      - É um ataque à disponibilidade de um serviço

Atacantes e Vítimas

- Cenário na Internet: meio de comunicação compartilhado
  - Dois usuários desejam se comunicar de forma segura
  - Um intermediário (intruso) pode interceptar, apagar e introduzir novas mensagens na comunicação



Atacantes e Vítimas

Instituto de computação

- Na prática, Alice e Bob podem ser
  - Um navegador e um servidor Web usados em transações eletrônicas
    - Ex.: compras *online*, Internet banking etc.
  - Servidores DNS
  - Roteadores que trocam atualizações de tabelas de roteamento
  - Etc.

Atacantes e Vítimas

Instituto de computação

- Na prática, o que um atacante pode fazer
  - Bisbilhotar ou grampear (*eavesdropping*)
    - Interceptar as mensagens
  - Inserir mensagens propositalmente em uma conversação
  - Personificação (*impersonation*)
    - É possível forjar (*spoofing*) o endereço da fonte e outros campos de um pacote IP

Atacantes e Vítimas

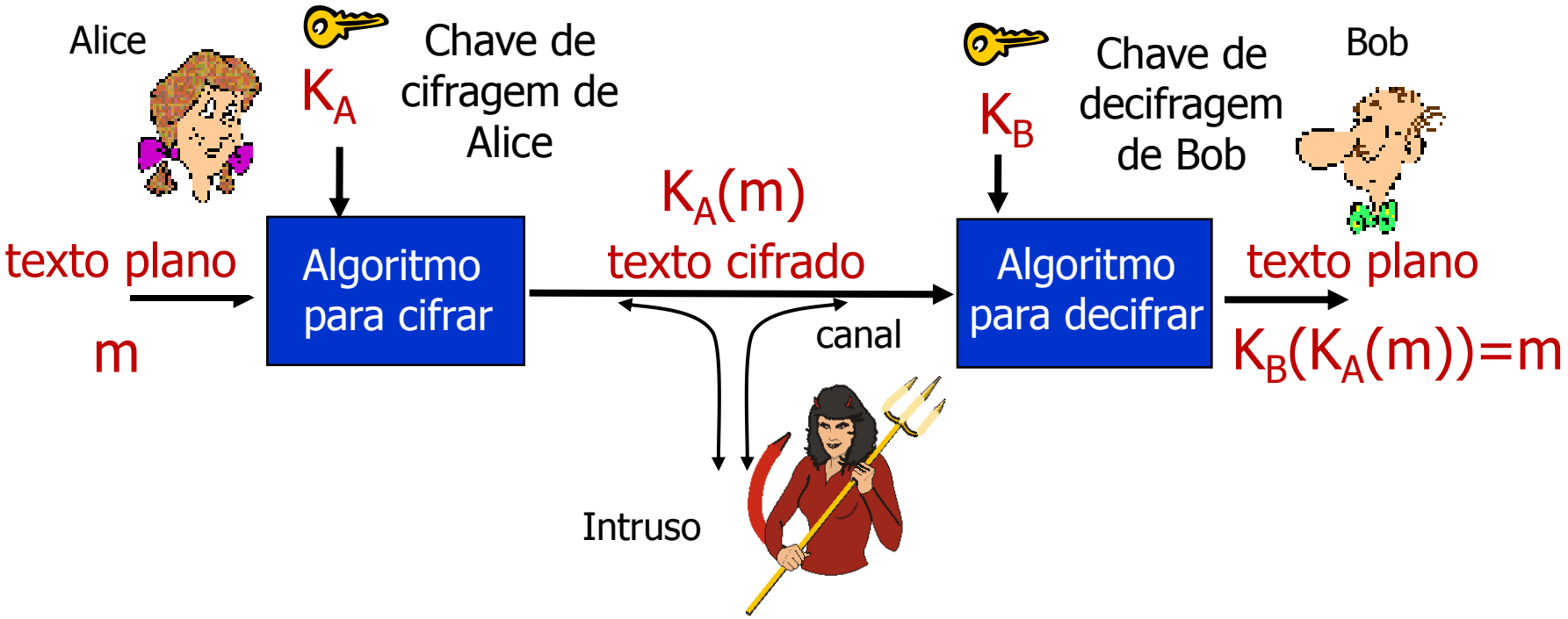
Instituto de computação

- Na prática, o que um atacante pode fazer
  - Sequestro (*hijacking*)
    - "Assumir" uma conversação em curso removendo o remetente ou receptor, inserindo-se no lugar de um deles
  - Negação de serviço (*Denial of Service* – DoS)
    - Tornar um serviço indisponível para usuários legítimos
      - O que é um serviço?
        - Hospedagem de um sítio
        - Buscador de páginas
        - Compra e venda de produtos
        - Troca de mensagens etc.

Instituto de computação

Princípios de Criptografia

Nomenclatura em Criptografia

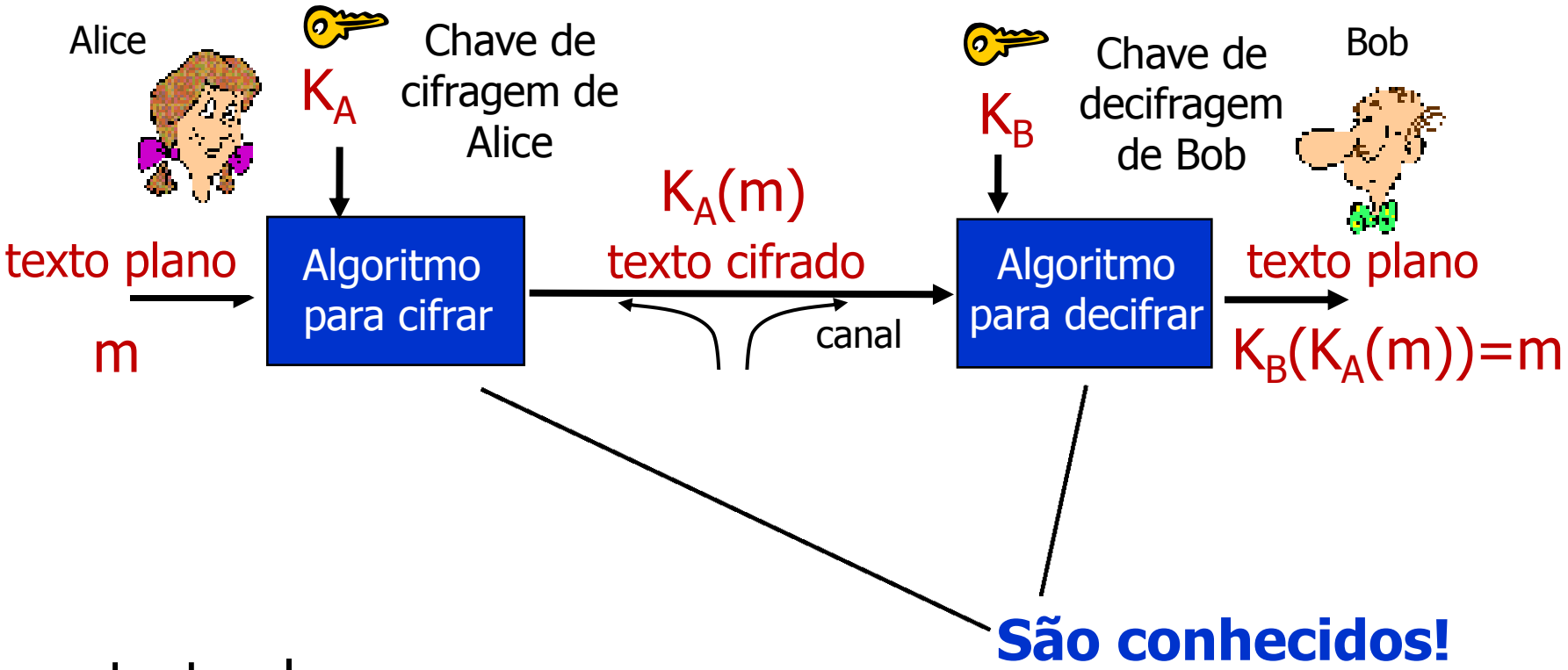


m: texto plano

$K_A(m)$: texto cifrado com a chave K_A

$m = K_B(K_A(m))$: texto plano recuperado, a partir da chave K_B

Nomenclatura em Criptografia

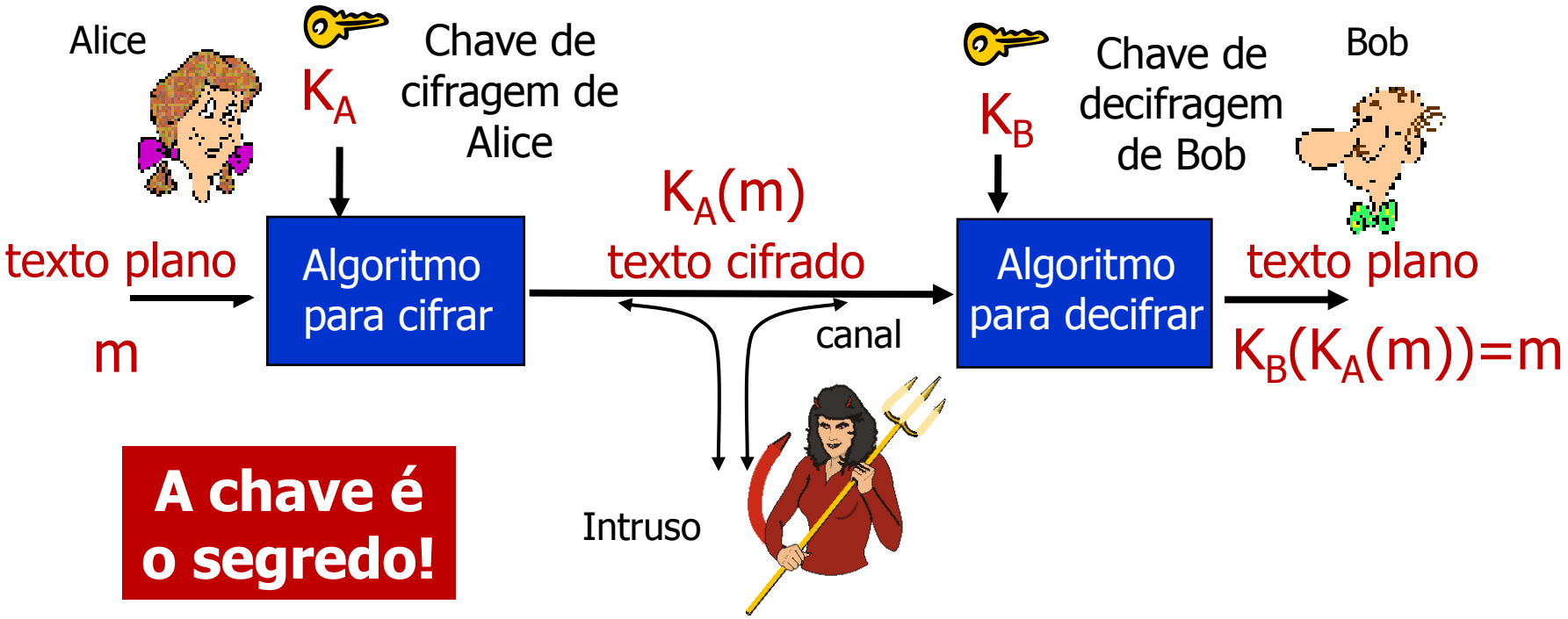


m: texto plano

$K_A(m)$: texto cifrado com a chave K_A

$m = K_B(K_A(m))$: texto plano recuperado, a partir da chave K_B

Nomenclatura em Criptografia

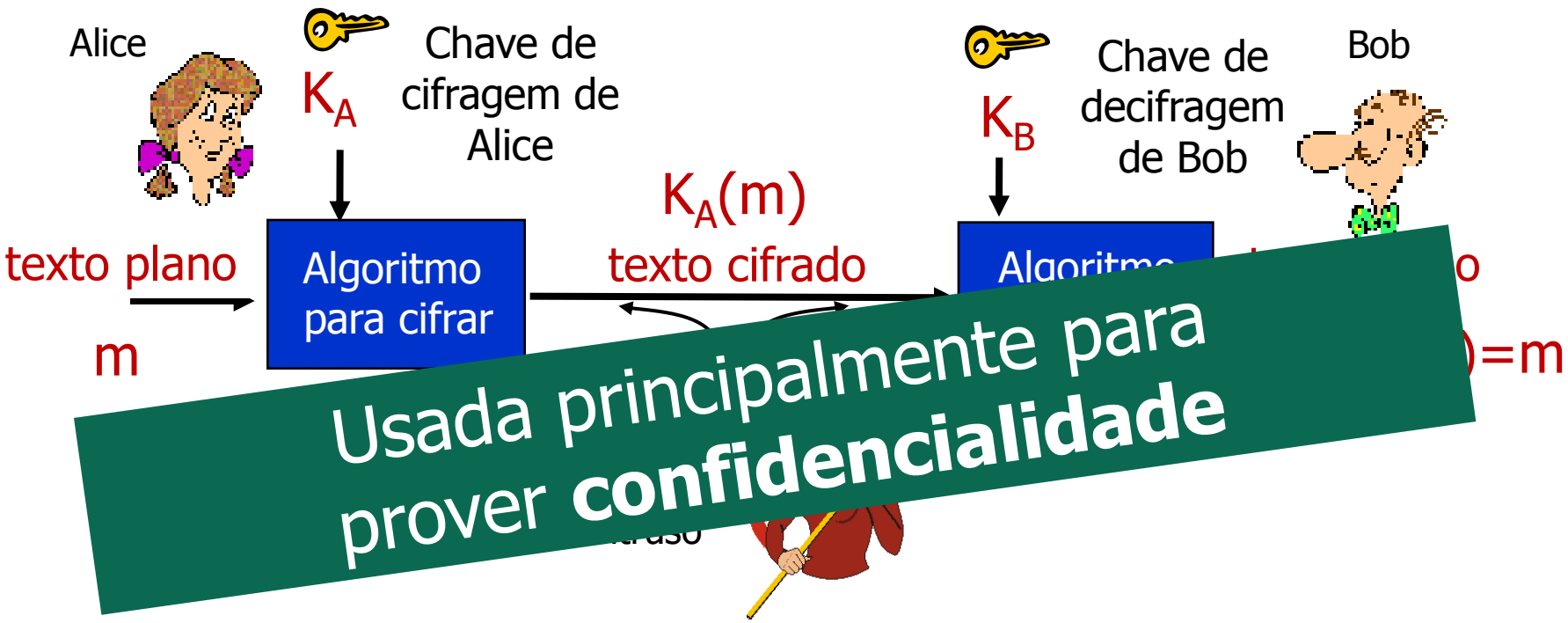


m: texto plano

$K_A(m)$: texto cifrado com a chave K_A

$m = K_B(K_A(m))$: texto plano recuperado, a partir da chave K_B

Nomenclatura em Criptografia



m: texto plano

$K_A(m)$: texto cifrado com a chave K_A

$m = K_B(K_A(m))$: texto plano recuperado, a partir da chave K_B

Tipos de Criptografia

Instituto de computação

- Em geral, são usadas chaves em criptografia
  - Algoritmo é conhecido por todos e somente as chaves são secretas
- Criptografia de **chaves simétricas**
  - As chaves do remetente e do destinatário são **idênticas**
    - $K_A = K_B$
- Criptografia de **chaves assimétricas ou públicas**
  - As chaves do remetente e do destinatário são **diferentes**
    - Cifra com chave pública, decifra com chave secreta (privada)
    - $K_A \neq K_B$

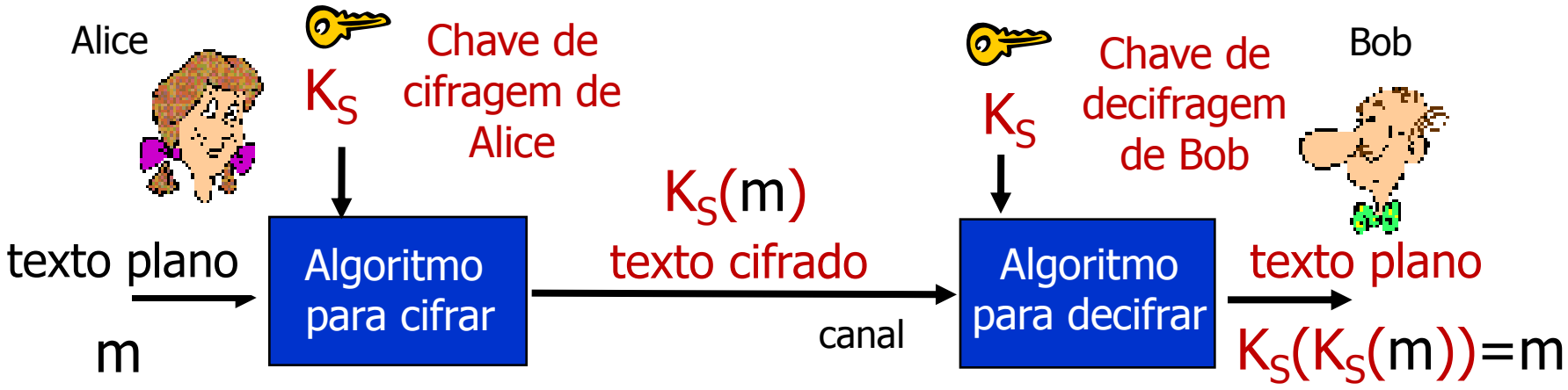
Tipos de Criptografia

Instituto de computação

- Funções *hash*
  - **NÃO** usa chaves
  - **NADA** é secreto

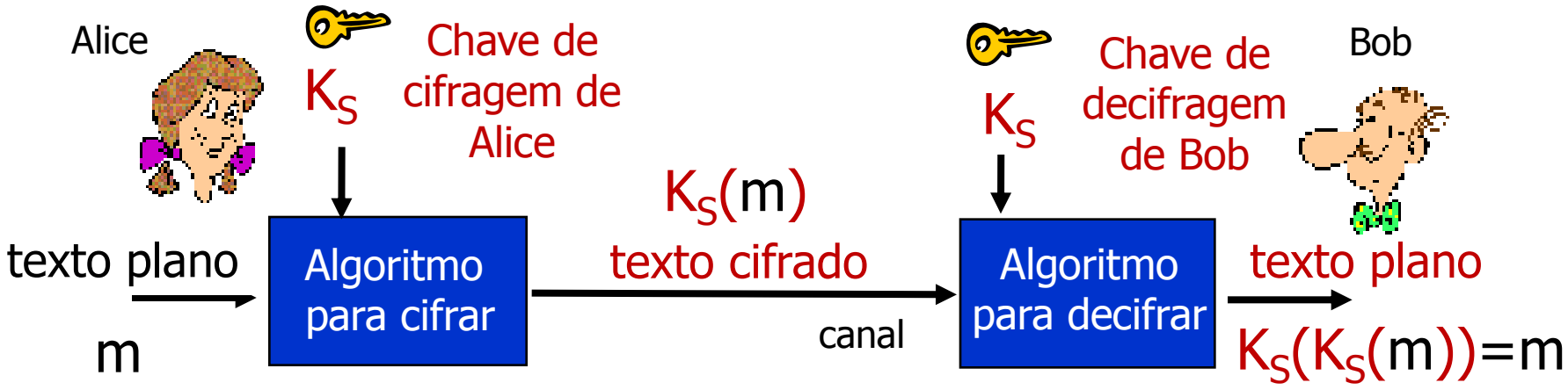
Qual a utilidade?

Criptografia com Chaves Simétricas



- Bob e Alice compartilham a mesma chave (simétrica) K_S
  - Ex.: Compartilhar a mesma chave é ter o mesmo padrão de substituição no algoritmo de substituição monoalfabético

Criptografia com Chaves Simétricas



- Bob e Alice compartilham a mesma chave (simétrica) K_S
  - Ex.: Compartilhar a mesma chave é ter o mesmo padrão de substituição no algoritmo de substituição monoalfabético

Problema: como Alice e Bob definem e trocam as chaves simétricas?

Cifra de César

Instituto de computação

- Deslocar cada letra do alfabeto de *k* de posições
  - *k* é fixo
- Se n=3

Texto plano: **abcdefghijklmnopqrstuvwxyz**

Texto cifrado: **defghijklmnopqrstuvwxyzabc**

Cifra de César

Instituto de computação

- Deslocar cada letra do alfabeto de *k* de posições
  - *k* é fixo
- Se n=3

Texto plano: **`abcdefghijklmnopqrstuvwxyz`**

Texto cifrado: **`defghijklmnopqrstuvwxyzabc`**

A **chave** é o parâmetro *k*

Algoritmos Monoalfabéticos

Instituto de computação

- Baseado na substituição de "uma coisa por outra"
  - Monoalfabético: substituir uma letra por outra

Texto plano: **abcdefghijklmnopqrstuvwxyz**

Texto cifrado: **mnbvcxzasdfghjklpoiuytrewq**

  - Exemplo:
    - Texto plano: **The book is on the table**
    - Texto cifrado: **Uac nkkf si kj uac umngc**

Algoritmos Monoalfabéticos

Instituto de computação

- Baseado na substituição de "uma coisa por outra"
  - Monoalfabético: substituir uma letra por outra

Texto plano: **`abcdefghijklmnopqrstuvwxyz`**

Texto cifrado: **`mnbvcxzasdfghjklpoiuytrewq`**

  - Exemplo:
    - Texto plano: **`The book is on the table`**
    - Texto cifrado: **`Uac nkkf si kj uac umngc`**

A **chave** é o mapeamento do conjunto de 26 caracteres para o outro conjunto de 26 caracteres

Algoritmos Polialfabéticos

Instituto de computação

- *n* cifras monoalfabéticas: M_1, M_2,..., M_n
- Padrão cíclico
  - Ex.: n=4 → M_1, M_3, M_4, M_3, M_2; M_1, M_3, M_4, M_3, M_2;
- Para cada símbolo do texto plano, use a cifra monoalfabética subsequente dada pelo padrão cíclico
- Exemplo
  - sal → s de M_1, a de M_3, l de M_4

Algoritmos Polialfabéticos

Instituto de computação

- *n* cifras monoalfabéticas: M_1, M_2,..., M_n
- Padrão cíclico
  - Ex.: n=4 → M_1, M_3, M_4, M_3, M_2; M_1, M_3, M_4, M_3, M_2;
- Para cada símbolo do texto plano, use a cifra monoalfabética subsequente dada pelo padrão cíclico
- Exemplo
  - sal → s de M_1, a de M_3, l de M_4

A **chave** é o conjunto das *n* cifras monoalfabéticas e do padrão cíclico

Quebrando um Algoritmo Criptográfico

Instituto de computação

- Ataque somente ao texto cifrado
  - O intruso possui o texto cifrado e pode analisá-lo

- Duas abordagens
  - "Força bruta": pesquisar por todas as chaves
    - Deve ser capaz de diferenciar os textos planos resultantes de sequências sem sentido
    - Cifra de César: 25 possíveis valores de *k*
    - Monoalfabética: 26! possíveis mapeamentos
  - Análise estatística
    - Letras mais comuns em palavras de um idioma, por exemplo

Quebrando um Algoritmo Criptográfico

Instituto de computação

- Ataque do texto plano conhecido
  - Intruso possui algum texto plano correspondente a algum texto cifrado
    - Ex.: Em uma cifra monoalfabética, o intruso pode determinar o mapeamento entre os elementos
- Ataque do texto plano escolhido
  - Intruso pode obter o texto cifrado para um dado texto plano escolhido

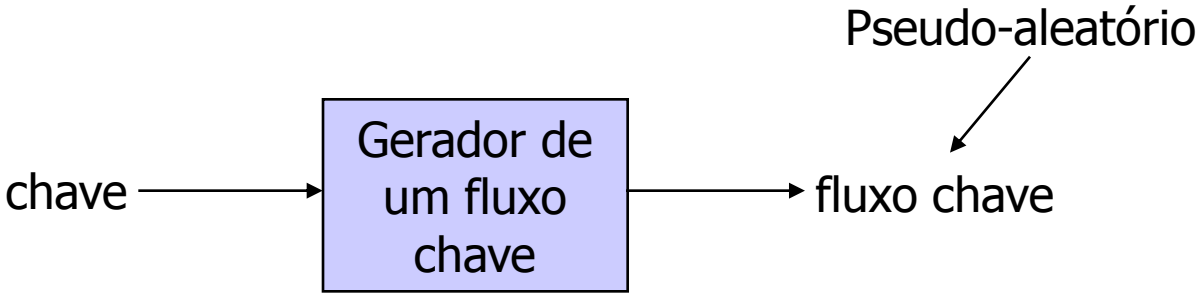
Cifradores Simétricos "Modernos"

Instituto de computação

- Cifradores de fluxo (*stream ciphers*)
  - Cifram um bit por vez
- Cifradores de bloco (*block ciphers*)
  - Quebra a mensagem em texto plano em blocos de mesmo tamanho
  - Cifra cada bloco como uma unidade

Cifradores de Fluxo

- Combinam cada bit de um fluxo chave (*keystream*) com um bit do texto plano para se obter um bit do texto cifrado
  - $m(i)$ = i-ésimo bit da mensagem em texto plano
  - $ks(i)$ = i-ésimo bit do fluxo chave
  - $c(i)$ = i-ésimo bit do texto cifrado
  - $c(i) = ks(i)$ XOR $m(i)$
  - $m(i) = ks(i)$ XOR $c(i)$



Cifrador de Fluxo RC4

Instituto de computação

- É um dos mais populares
- Analisado exaustivamente e considerado “bom”
- Usa chaves de até 256 bytes
- É usado com o WEP no IEEE 802.11
- Também pode ser usado com o SSL

Cifradores de Bloco

Instituto de computação

- Uma mensagem a ser cifrada é **processada em blocos** de k bits
  - Ex.: blocos de 64 bits
- É usado um mapeamento um-para-um para mapear um bloco de k bits de texto plano em um bloco de k bits de texto cifrado

Cifradores de Bloco

- Exemplo: k =3

mapeamento

entrada	saída
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

Qual o texto cifrado para 010110001111 ?

Cifradores de Bloco

- Exemplo: k =3

mapeamento

entrada	saída
000	110
001	111
011	100
100	011
101	010
110	000
111	001

Qual o texto cifrado para [ ]110001111 ?

101

Cifradores de Bloco

- Exemplo: k =3

mapeamento

entrada	saída
000	110
001	111
010	101
011	100
100	011
101	010
111	001

Qual o texto cifrado para 010 001111 ?

101000

Cifradores de Bloco

- Exemplo: k =3

mapeamento

entrada	saída
000	110
010	101
011	100
100	011
101	010
110	000
111	001

Qual o texto cifrado para 01011( )111 ?

101000111

Cifradores de Bloco

- Exemplo: k =3

mapeamento

entrada	saída
000	110
001	111
010	101
011	100
100	011
101	010
110	000

Qual o texto cifrado para 010110001 ?

101000111001

Cifradores de Bloco

Instituto de computação

- Um mapeamento é uma permutação de todas as possíveis entradas
  - Não há repetições

- Quantos mapeamentos são possíveis para k =3?
  - Quantas entradas de 3 bits?
  - Quantas permutações das entradas de 3 bits?

Cifradores de Bloco

Instituto de computação

- Um mapeamento é uma permutação de todas as possíveis entradas
  - Não há repetições

- Quantos mapeamentos são possíveis para k =3?
  - Quantas entradas de 3 bits? **2^3= 8 entradas**
  - Quantas permutações das entradas de 3 bits? **8!= 40320**

Número pequeno → força bruta pode ser usada

Cifradores de Bloco

Instituto de computação

- Em geral: 2^k! mapeamentos

- Se k =64
  - 2^{64}! Possíveis mapeamentos → dificulta a "força-bruta"
  - Mas...
    - Tabela completa mantida pelos usuários é muito grande
      - 2^{64} entradas
      - Cada entrada com 64 bits

Cifradores de Bloco

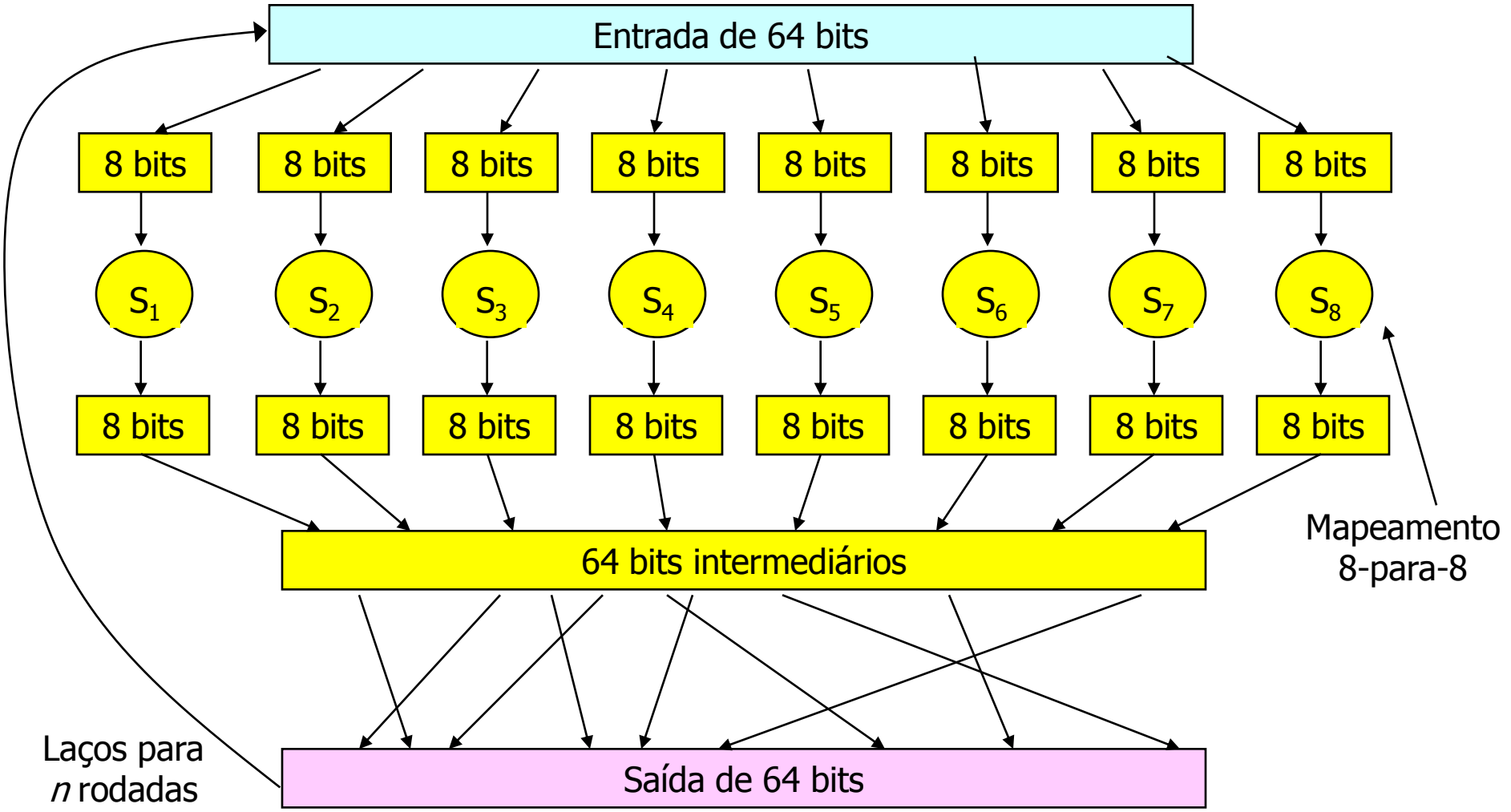
Instituto de computação

- Em geral: 2^k! mapeamentos

- Se k =64
  - 2^{64}! Possíveis mapeamentos → dificulta a "força-bruta"
  - Mas...
    - Tabela completa mantida pelos usuários é muito grande
      - 2^{64} entradas
      - Cada entrada com 64 bits

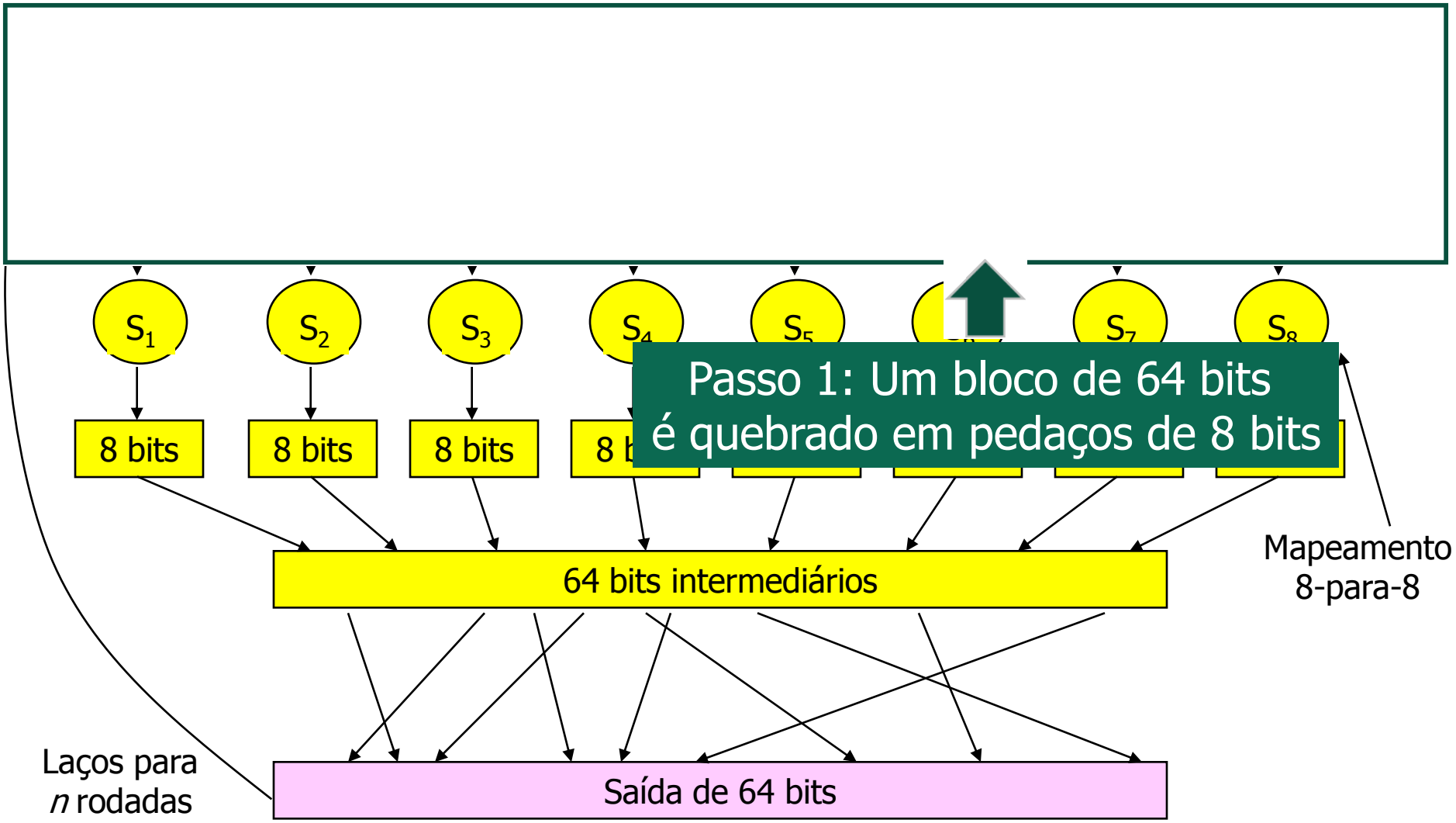
Solução: usar uma função que simula uma **tabela aleatória de permutas** ao invés da completa

Exemplo de uma Função



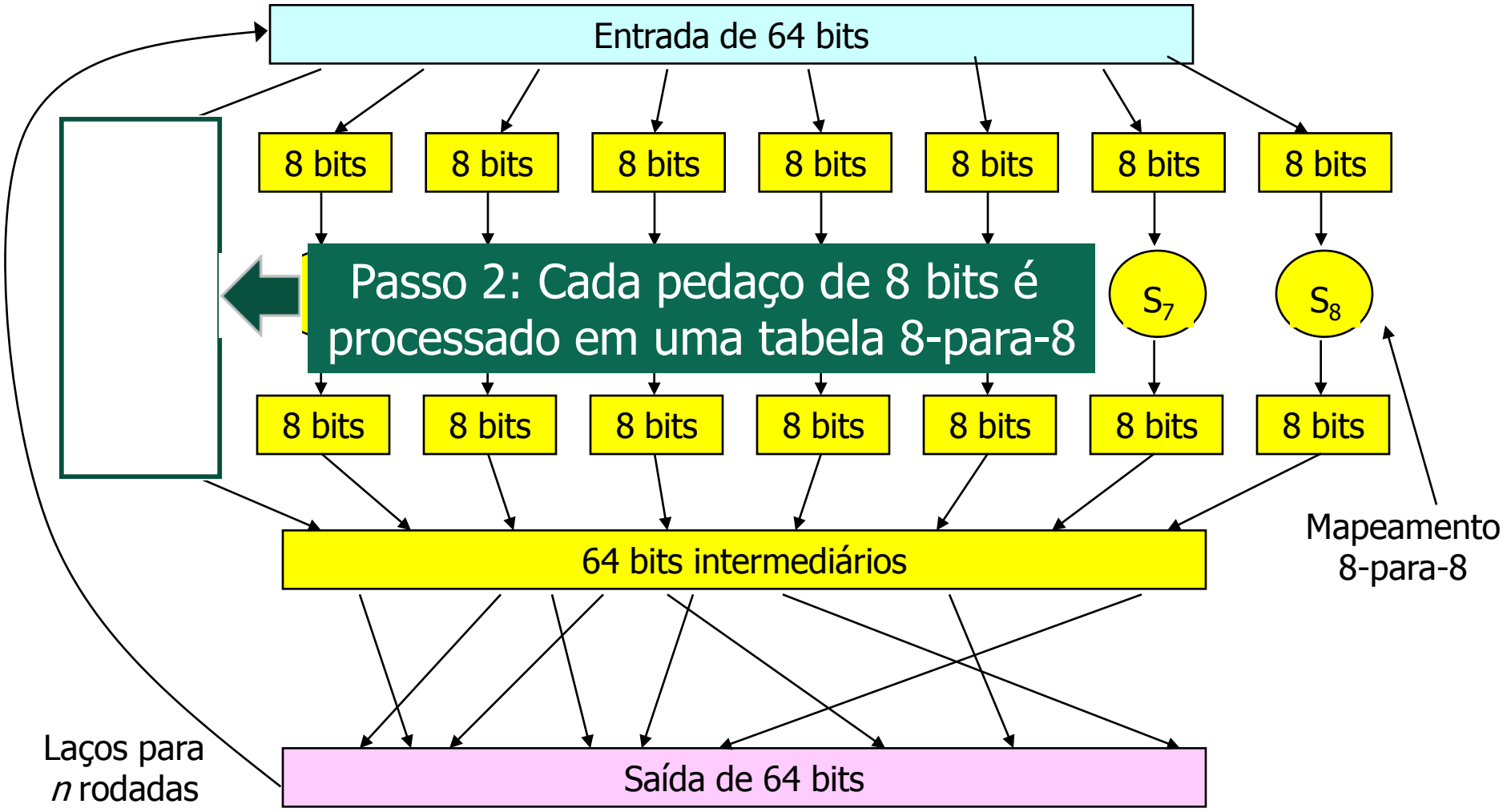
Extraído de Kaufman et al.

Exemplo de uma Função



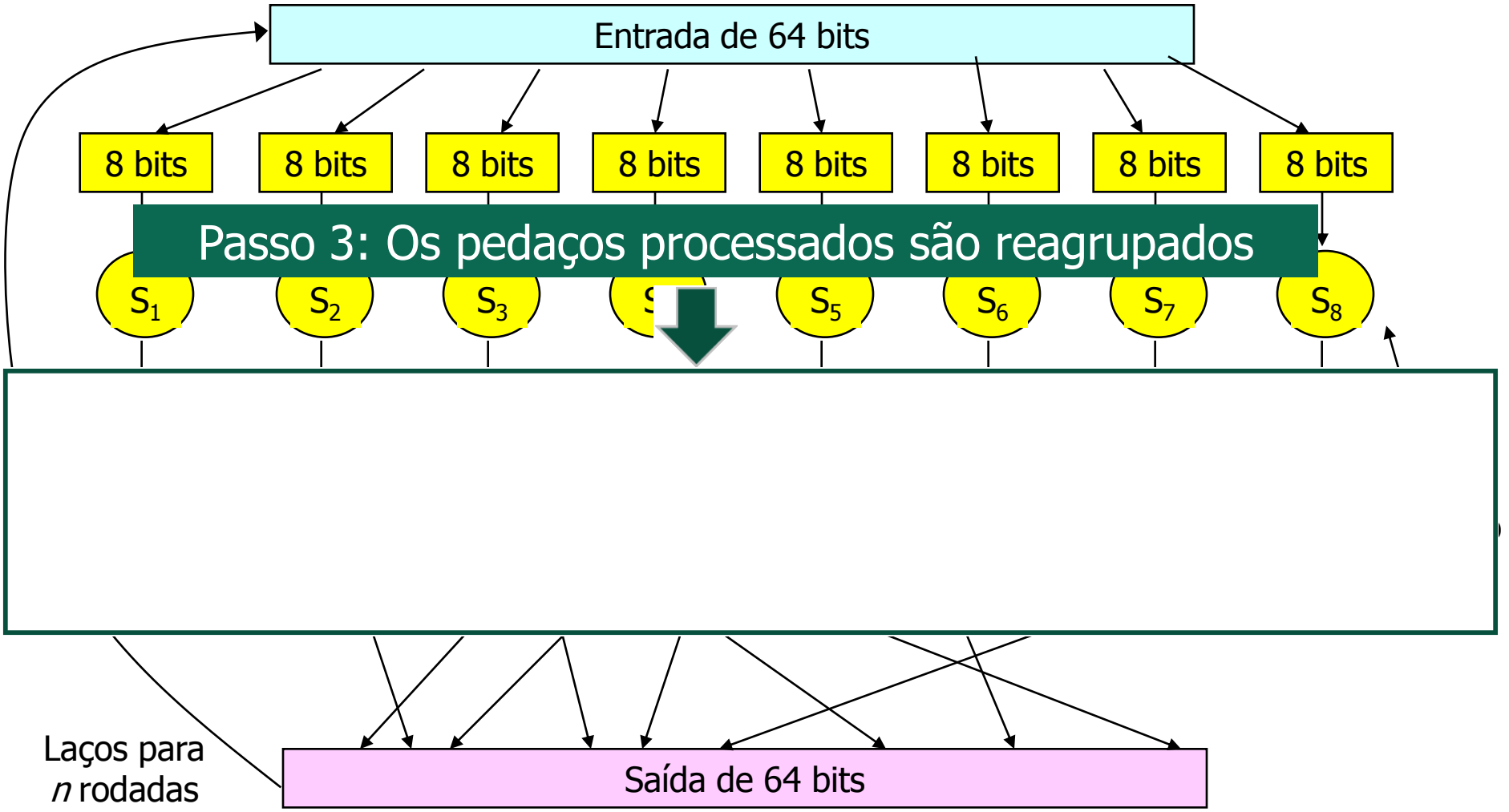
Extraído de Kaufman et al.

Exemplo de uma Função



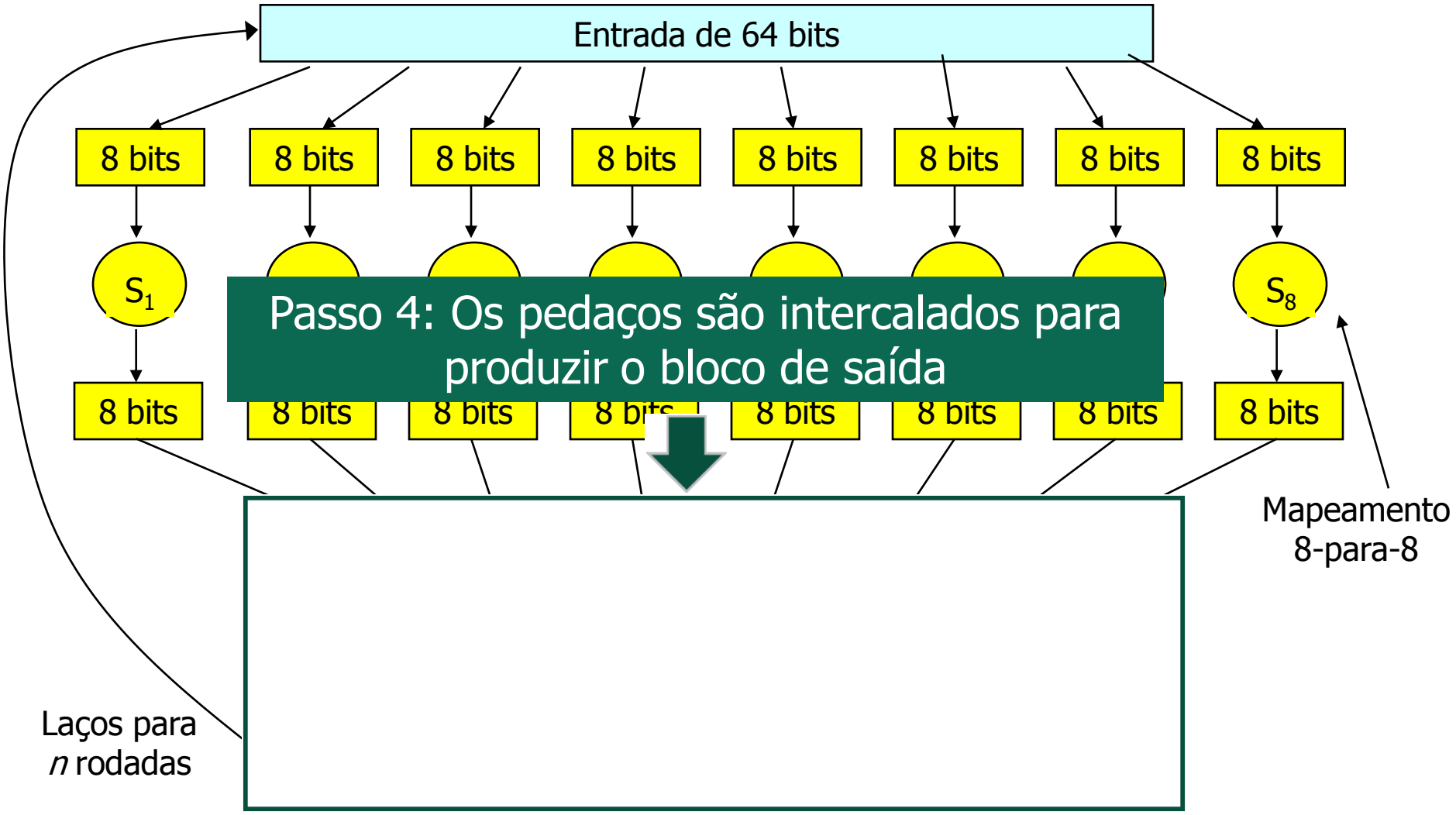
Extraído de Kaufman et al.

Exemplo de uma Função



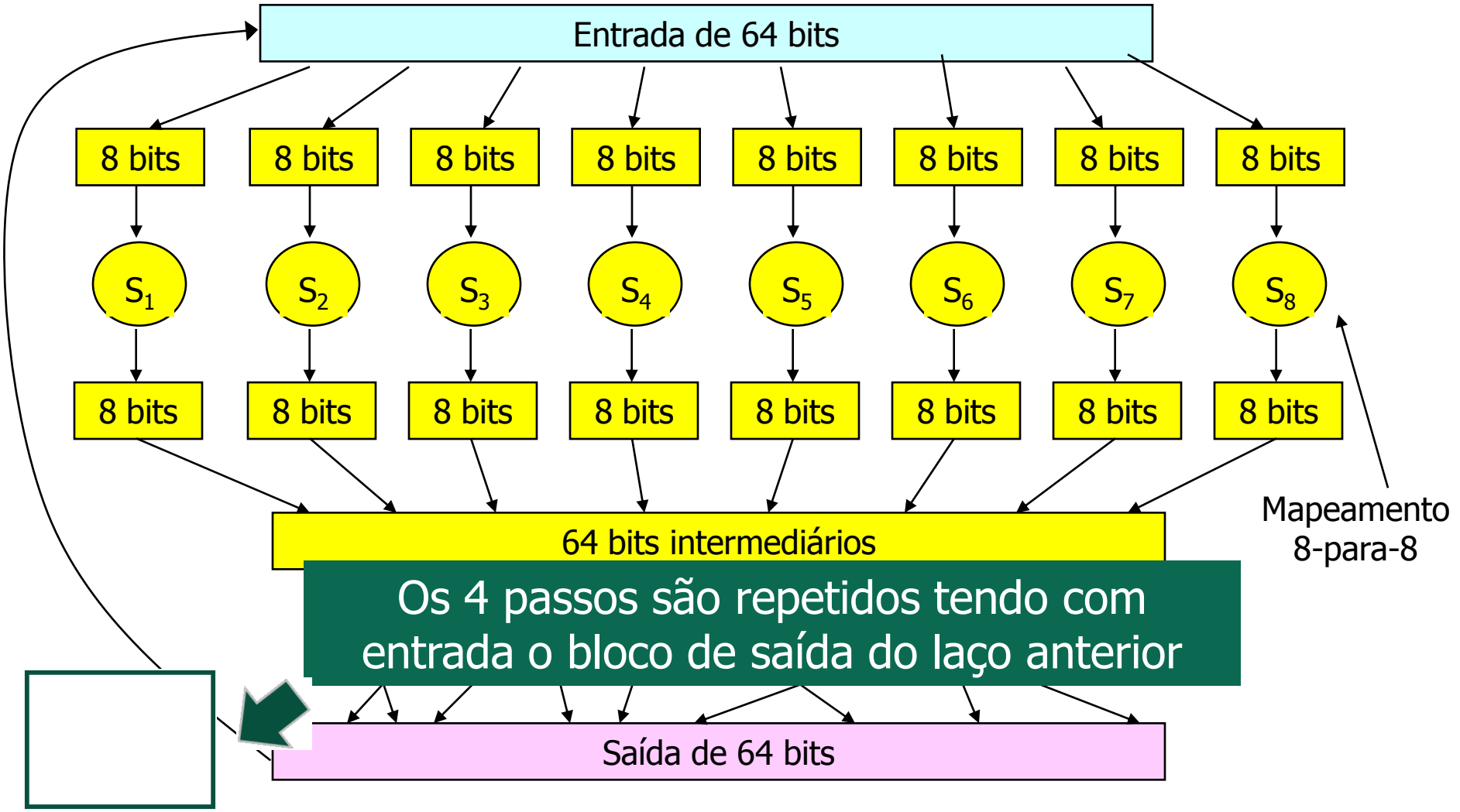
Extraído de Kaufman et al.

Exemplo de uma Função



Extraído de Kaufman et al.

Exemplo de uma Função



Extraído de Kaufman et al.

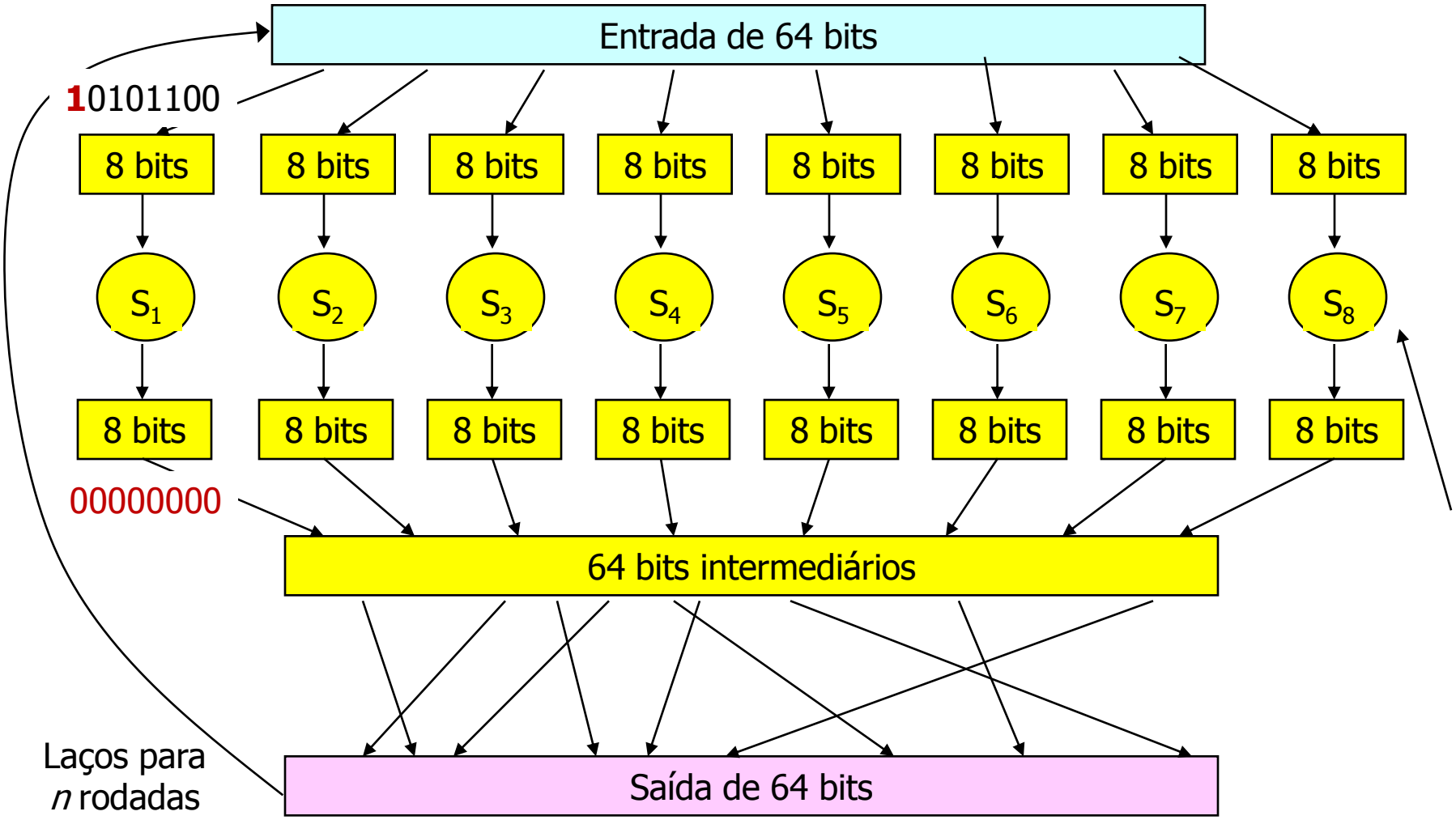
Por que Repetir o Procedimento?

Instituto de computação

- Se o procedimento fosse realizados **apenas uma vez**
  - Um bit de entrada influencia no máximo 8 bits de saída
- Na 2a. rodada
  - Os 8 bits influenciados na primeira rodada se "espalham" e servem de entrada para múltiplas tabelas

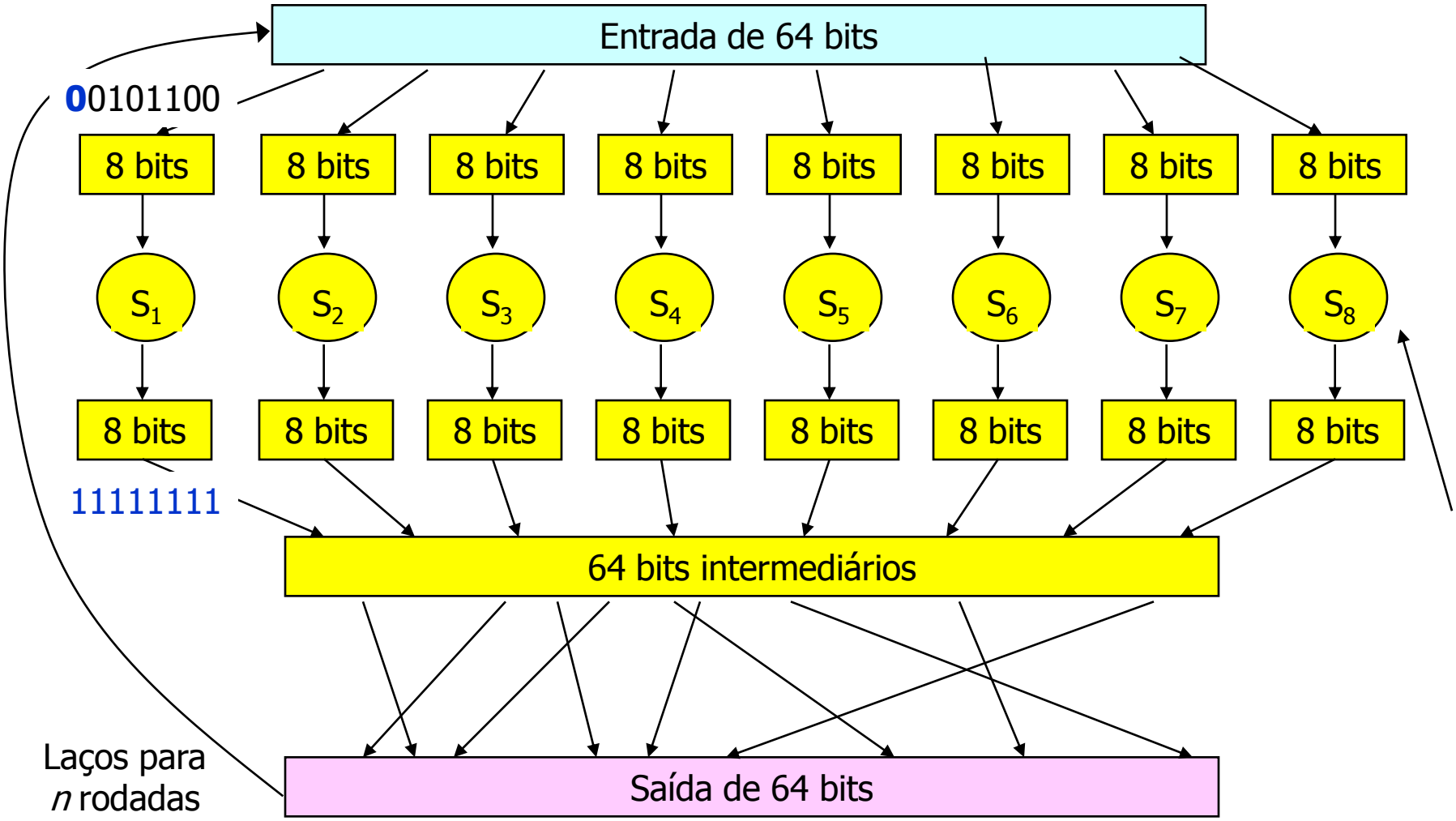
- Compromisso: quantos rodadas?
  - Quantas rodadas forem necessárias para "embaralhar" os bits de entrada
  - Menos eficiente à medida que *n* cresce

Por que Repetir o Procedimento?



Extraído de Kaufman et al.

Por que Repetir o Procedimento?



Extraído de Kaufman et al.

Cifrando uma Mensagem Grande

Instituto de computação

- **NÃO** basta quebrar a mensagem em blocos de 64 bits e cifrar cada bloco separadamente
  - Se o mesmo bloco de texto plano aparece mais de uma vez, o mesmo texto cifrado será produzido para cada um dos blocos

- Consequência
  - Um atacante pode obter o texto plano da mensagem
    - Identifica dois blocos cifrados idênticos
    - Conhece o funcionamento do protocolo
      - Ex.: "HTTP 1.1"

Cifrando uma Mensagem Grande

Instituto de computação

- Possível solução: introduzir um grau de aleatoriedade
  - Gerar um número aleatório de 64 bits $r(i)$ para cada bloco de texto plano $m(i)$
  - Calcular $c(i) = K_S(m(i) \text{ XOR } r(i))$
  - Transmitir $c(i)$, $r(i)$, $i = 1, 2, \cdots$
  - No receptor: $m(i) = K_S(c(i) \text{ XOR } r(i))$

- Problema: é ineficiente
  - É necessário enviar $c(i)$ e $r(i)$

Encadeamento de Cifradores de Bloco

Instituto de computação

- *Cipher Block Chaining* (CBC)

- CBC gera os seus próprios números aleatórios
  - O processo para cifrar o bloco atual depende do resultado do processo para cifrar o bloco anterior
  - $c(i) = K_S(m(i) \text{ XOR } c(i\text{-}1))$
  - $m(i) = K_S(c(i)) \text{ XOR } c(i\text{-}1)$

Encadeamento de Cifradores de Bloco

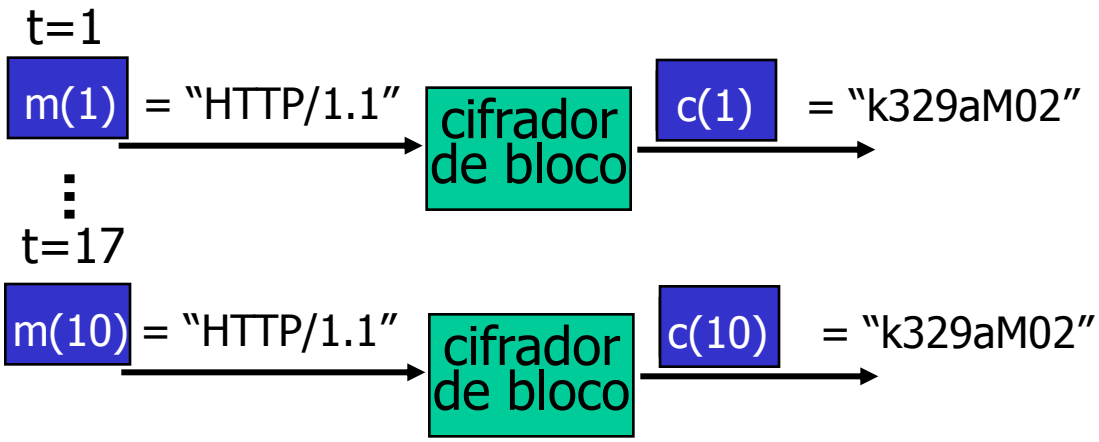
Instituto de computação

- Como cifrar o primeiro bloco?
  - Usar um **vetor de incialização** (VI): bloco aleatório = $c(0)$
  - O vetor VI não precisa ser secreto

- O vetor VI deve ser alterado para cada mensagem (ou sessão)
  - Garantir que mesmo que a mesma mensagem seja enviada repetidamente, o texto cifrado será completada diferente a cada mensagem

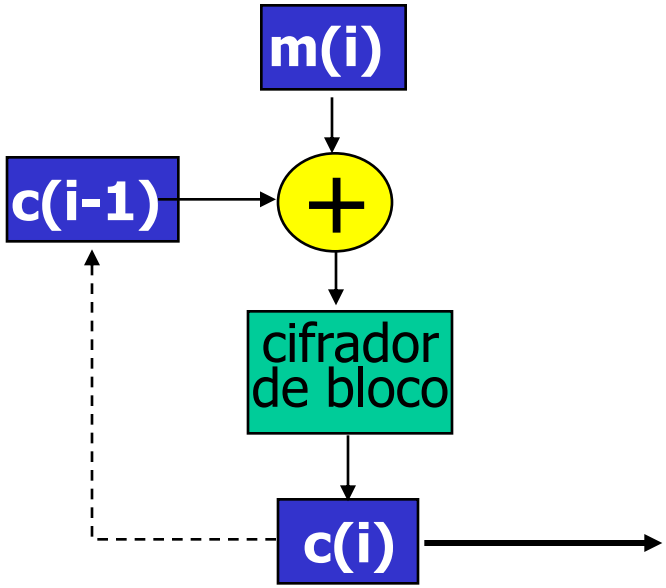
Encadeamento de Cifradores de Bloco

- Cifradores de bloco
  - Se o bloco de entrada é repetido, o mesmo texto cifrado será produzido



Encadeamento de Cifradores de Bloco

- Encadeamento de cifradores de bloco
  - XOR do i-ésimo bloco de entrada, $m(i)$, com o bloco cifrado anteriormente, $c(i\text{-}1)$
  - $c(0)$ é transmitido em claro para o receptor
  - Exemplo: "HTTP/1.1"



Data Encryption Standard (DES)

Instituto de computação

- Usa criptografia de **chaves simétricas**
  - Chaves de 56 bits
  - Entradas em texto plano de 64 bits
- Usa encadeamento de cifradores em bloco
- É o padrão criptográfico dos EUA

Data Encryption Standard (DES)

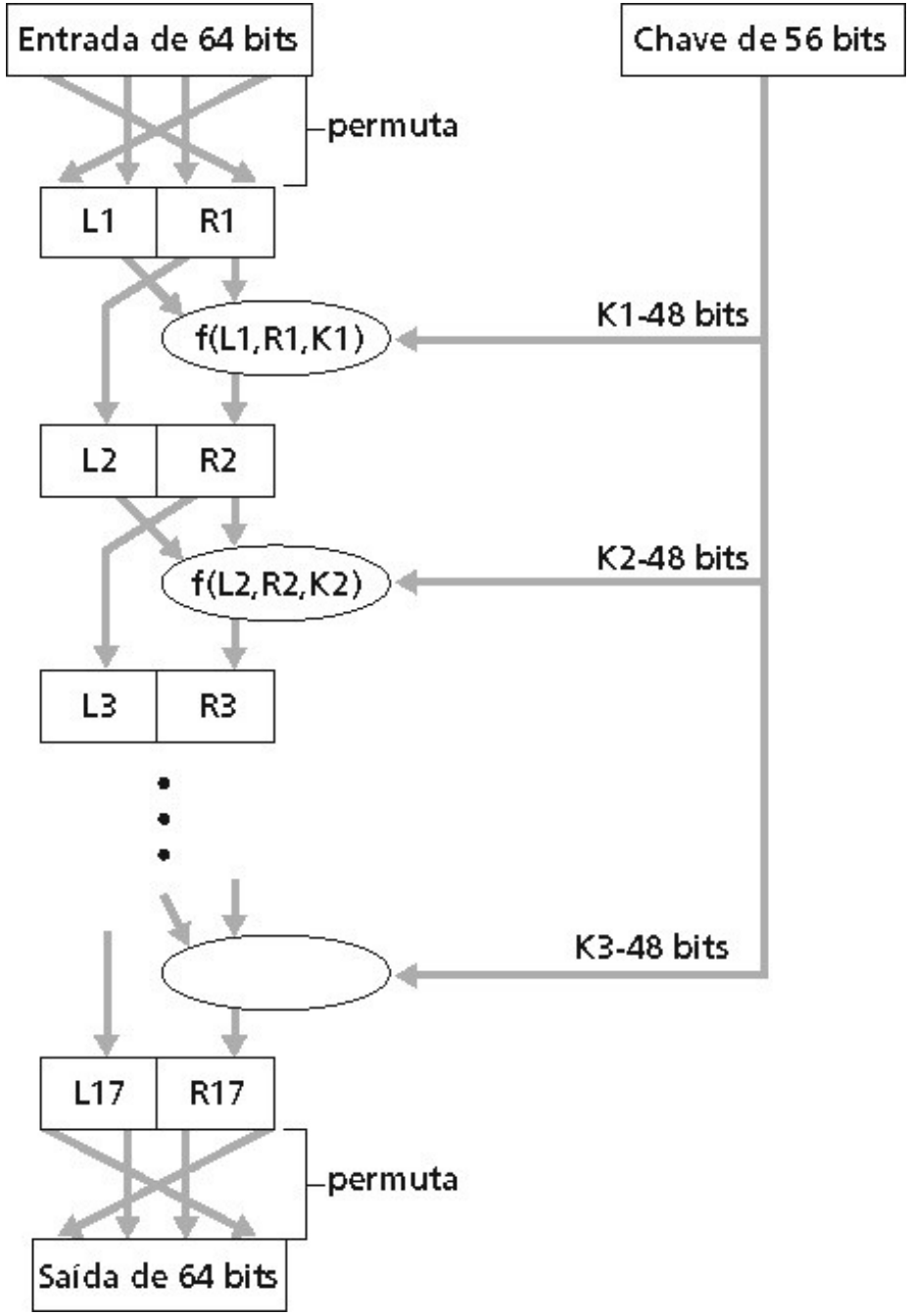
Instituto de computação

- Quão seguro é o DES?
  - Desafio DES
    - Uma frase cifrada com um chave de 56 bits é decifrada em menos de um dia com força bruta
  - Nenhum ataque analítico bem conhecido

- 3DES
  - Tornar o DES mais seguro
  - Cifrar 3 vezes com 3 chaves diferentes
    - Na verdade, cifrar → decifrar → cifrar

DES

- Operação
  - Permutação inicial
  - 16 rodadas idênticas de aplicação da função
    - Usando diferentes chaves de 48 bits
  - Permutação final



Advanced Encryption Standard (AES)

Instituto de computação

- Novo padrão criptográfico de chave simétricas dos EUA
  - Proposto em novembro de 2001 para substituir o DES
  - Em uso desde maio de 2002
- Processa dados em blocos de 128 bits
- Usa chaves de 128, 192, ou 256 bits
- Métrica de segurança
  - Se a decifração usando a força bruta (tentar cada chave) levasse 1 s no DES, levaria 149 trilhões de anos no AES

Criptografia de Chaves Assimétricas

Instituto de computação

- Também chamada de criptografia de chaves públicas

- Problema da criptografia de chaves simétricas
  - Requer que o emissor e o receptor **compartilhem uma chave secreta**

Criptografia de Chaves Assimétricas

Instituto de computação

- Também chamada de criptografia de chaves públicas

- Problema da criptografia de chaves simétricas
  - Requer que o emissor e o receptor **compartilhem uma chave secreta**

Como definir incialmente uma chave?

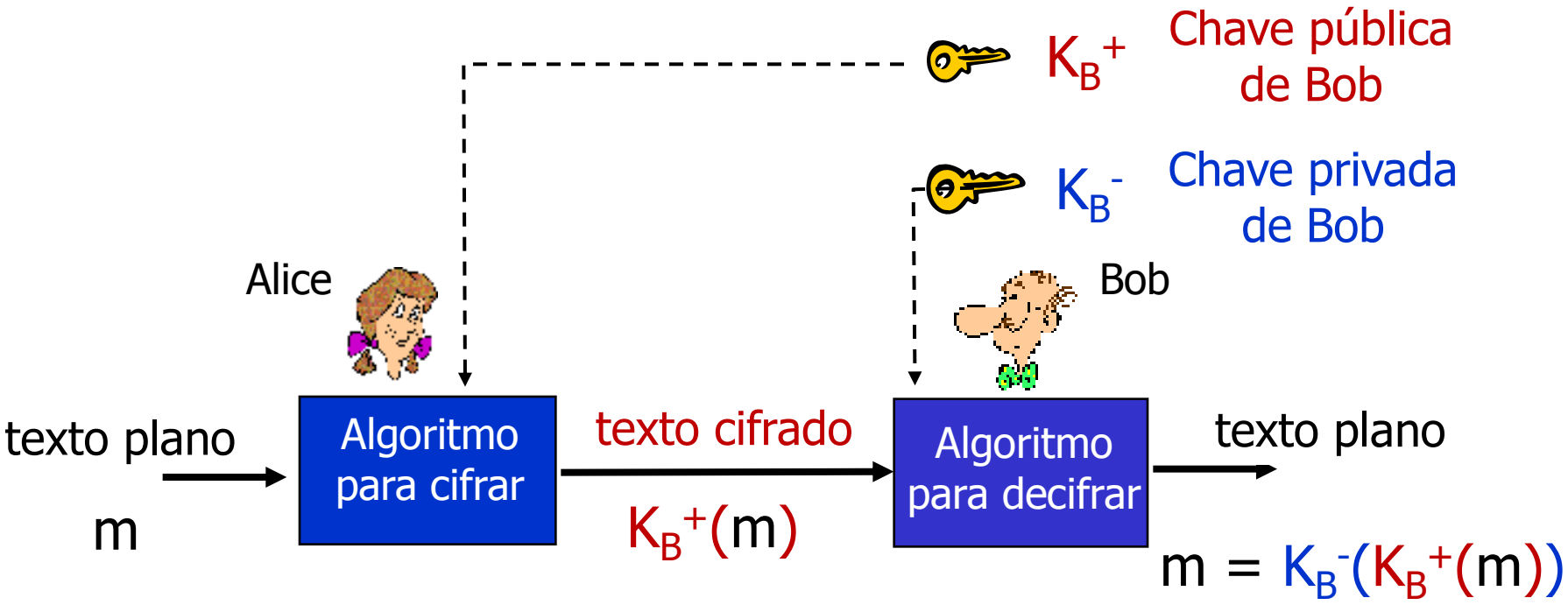
Criptografia de Chaves Assimétricas

Instituto de computação

- Abordagem completamente diferente da criptografia de chaves simétricas

- Emissor e receptor **NÃO** compartilham uma chave secreta
  - A chave para **cifrar** uma mensagem é **pública**
    - Conhecida por todos
  - A chave para **decifrar** uma mensagem é **privada**
    - Só é conhecida pelo receptor

Criptografia de Chaves Assimétricas



Criptografia de Chaves Assimétricas

Instituto de computação

- Possíveis ataques
  - Interceptação de mensagens para descoberta do seu conteúdo
    - Intruso tem posse da mensagem e da chave usada para cifrar
    - Também conhece o algoritmo usado para cifrar
    - Tentar codificar mensagens conhecidas e comparar o resultado
  - Qualquer um pode enviar uma mensagem para Bob usando sua chave pública
    - Alice ou alguém que se passa por ela

Criptografia de Chaves Assimétricas

Instituto de computação

- Requisitos
  1. É necessário $K_B^-(.)$ e $K_B^+(.)$ tais que

  2. Dada a chave pública K_B^+ deve ser impossível computar a chave privada K_B^-

Criptografia de Chaves Assimétricas

Instituto de computação

- Requisitos
  1. É necessário $K_B^-(.)$ e $K_B^+(.)$ tais que

  2. Dada a chave pública K_B^+ deve ser impossível computar a chave privada K_B^-

Algoritmo RSA: Rivest, Shamir e Adelson

Pré-Requisito: Aritmética Modular

Instituto de computação

`x mod n` = resto de **`x`** quando dividido por **`n`**

- Fato

$$[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$$
$$[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$$
$$[(a \bmod n) * (b \bmod n)] \bmod n = (a*b) \bmod n$$

- Assim

$$(a \bmod n)^d \bmod n = a^d \bmod n$$

- Exemplo: x=14, n=10, d=2

$$(x \bmod n)^d \bmod n = (14 \bmod 10)^2 \bmod 10 = 16 \bmod 10 = 6$$
$$14^2 \bmod 10 = 196 \bmod 10 = 6$$

RSA: Passos Iniciais

Instituto de computação

- Uma mensagem é um padrão de bits
- Um padrão de bits pode ser representado unicamente por um número inteiro
  - Junto com o comprimento do padrão
- Cifrar um mensagem, portanto, é equivalente a cifrar um número
- Exemplo: m = 10010001
  - Mensagem é representada unicamente pelo número decimal 145
  - Para cifrar m, é necessário cifrar o número correspondente
    - Um novo número é obtido → **texto cifrado**

RSA: Criação do Par de Chaves

Instituto de computação

1. Escolher dois números primos grandes, p e q
   - Ex.: Cada um com 1024 bits
2. Computar $n = pq$, $z = (p\text{-}1)(q\text{-}1)$
3. Escolher e ($e < n$) que não possui fatores comuns com z
   - e e z são primos relativos
4. Escolher d tal que $ed\text{-}1$ é exatamente divísivel por z
   - Ou seja, $ed \bmod z = 1$
5. A chave pública é [ ] e a chave privada é [ ]

$K_B^+ = (n,e)$ K_B^-

RSA: Cifrar e Decifrar

- Suposição: (n,e) e (n,d) computados como anteriormente

1. Para cifrar a mensagem m ($<n$), computar

$$c = m^e \bmod n$$

só é preciso $K_B^+ = (n,e)$

2. Para decifrar a mensagem o texto cifrado c, computar

$$m = c^d \bmod n$$

só é preciso $K_B^- = (n,d)$

RSA: Por que funciona?

- É necessário mostrar que $c^d \bmod n = m$
  - Onde $c = m^e \bmod n$
- Fato: para quaisquer x e y → $x^y \bmod n = x^{(y \bmod z)} \bmod n$
  - Onde $n = pq$ 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: Por que funciona?

- É necessário mostrar que $c^d \bmod n = m$
  - Onde $c = m^e \bmod n$
- Fato: para quaisquer x e y →
  - Onde $n = pq$ e $z = (p-1)(q-1)$
- Assim

$$c^d \bmod n = (m^e \bmod n)^d \bmod n$$
$$= m^{ed} \bmod n$$
$$= m^1 \bmod n$$
$$= m$$

RSA: Exemplo

Instituto de computação

- Bob escolhe
  - $p = 5$ e $q = 7$ → $n = 35$ e $z = 24$
  - $e = 5$ → e e z são primos relativos
  - $d = 29$ → 5*29-1 é divisível por 24 ($ed - 1$)
- Bob divulga $n = 35$ e $e = 5$
- Bob mantém em segredo $d = 29$

RSA: Exemplo

Instituto de computação

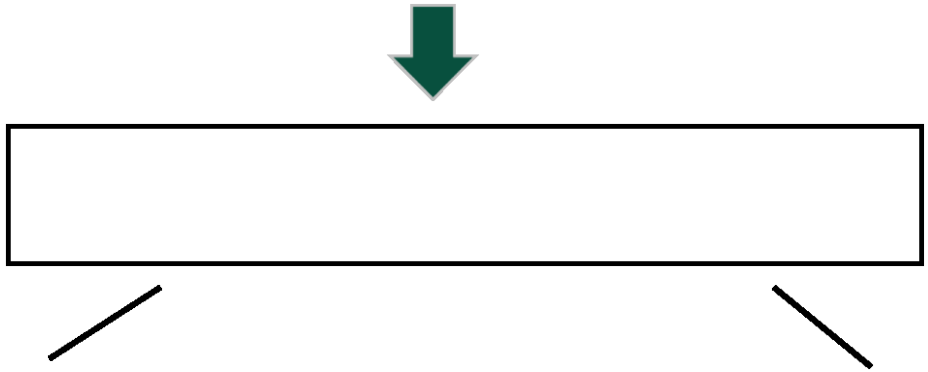
- Alice quer enviar uma mensagem cifrada de 8 bits a Bob
  - Padrão de bits a ser enviado: 00001100
  - mensagem $m = 12$ → $m^e = 12^5 = 248832$
  - $c = m^e \bmod n = 17$ → texto cifrado é enviado

- Ao receber o texto cifrado de Alice, Bob
  - Texto cifrado $c = 17$ → $c^d = 17^{29} =$ 48196857210675091509141182522307169 7
  - $m = c^d \bmod n = 12$ → texto recebido é decifrado

RSA: Outra Propriedade

- Ordem de cifração e decifração não importam

$$(m^d \bmod n)^e \bmod n = m^{de} \bmod n = m^{ed} \bmod n = \\ = (m^e \bmod n)^d \bmod n$$



Usa-se a chave pública, seguida da chave privada

Usa-se a chave privada, seguida da chave pública

O resultado é o mesmo!

RSA: Por que é Seguro?

Instituto de computação

- **NÃO** se conhecem algoritmos para **fatorar rapidamente** um número grande
  - Por isso, números de 1024 bits no RSA
- A chave pública de Bob (n,e) é conhecida
- Qual a dificuldade para se determinar d ?
  - É preciso fatorar o número público n em números primos p e q
  - Com p e q, chegar até a chave secreta d é simples
- Segurança **NÃO** é garantida
  - Não se sabem se existem ou não algoritmos rápidos para fatorar um número grande

Chaves de Sessão

Instituto de computação

- Exponenciação é uma tarefa de alto custo computacional
  - DES é pelo menos 100 vezes mais rápido do que o RSA

- Alternativa: estabelecer uma chave de sessão K_S
  - Bob e Alice usam o RSA para trocar uma chave simétrica K_S
  - Depois que ambos possuem K_S, eles usam criptografia de chave simétrica

Instituto de computação

Integridade e Autenticação

Integridade e Autenticação

Instituto de computação

- Permite que as partes comunicantes possam verificar se as mensagens recebidas não foram alteradas e se são autênticas
  - Conteúdo de uma mensagem não é alterado
  - Fonte da mensagem é quem realmente a enviou
  - Mensagem não pode ser repetida (*replay attack*)
  - Sequência das mensagens é mantida

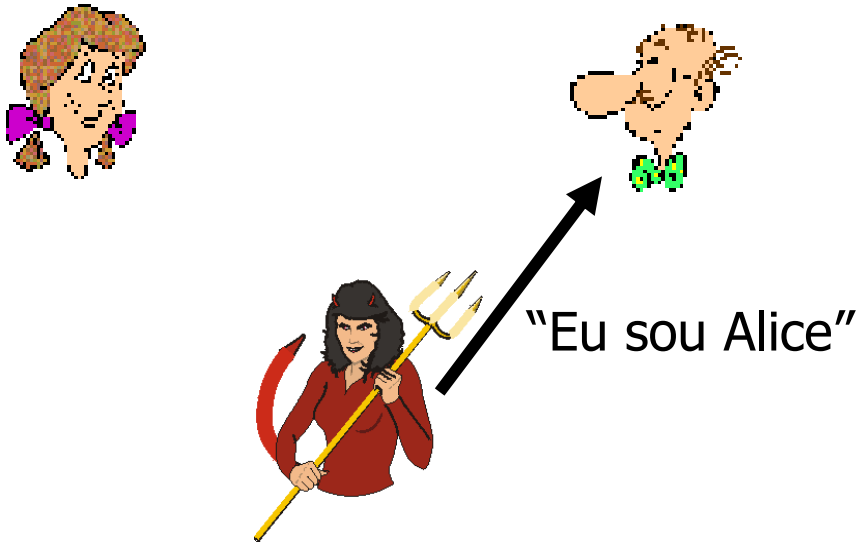
Autenticação

- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice"

Cenário de falha?

Autenticação

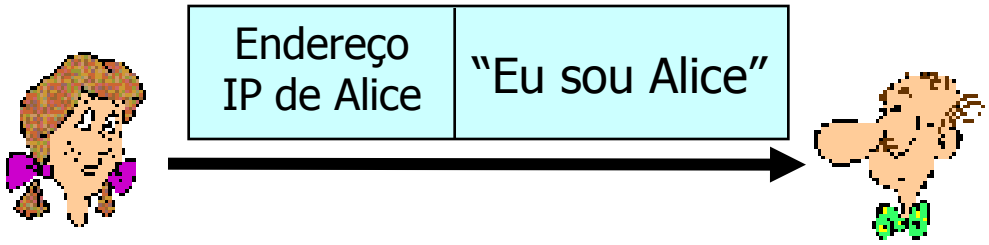
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice"



Em uma rede,
Bob não "vê" Alice, então
Trudy simplesmente se
declara como sendo Alice.

Autenticação

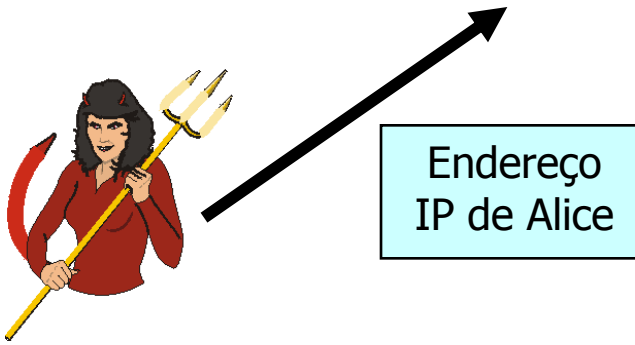
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia junto o seu endereço IP como "prova"



Cenário de falha?

Autenticação

- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia junto o seu endereço IP como "prova"

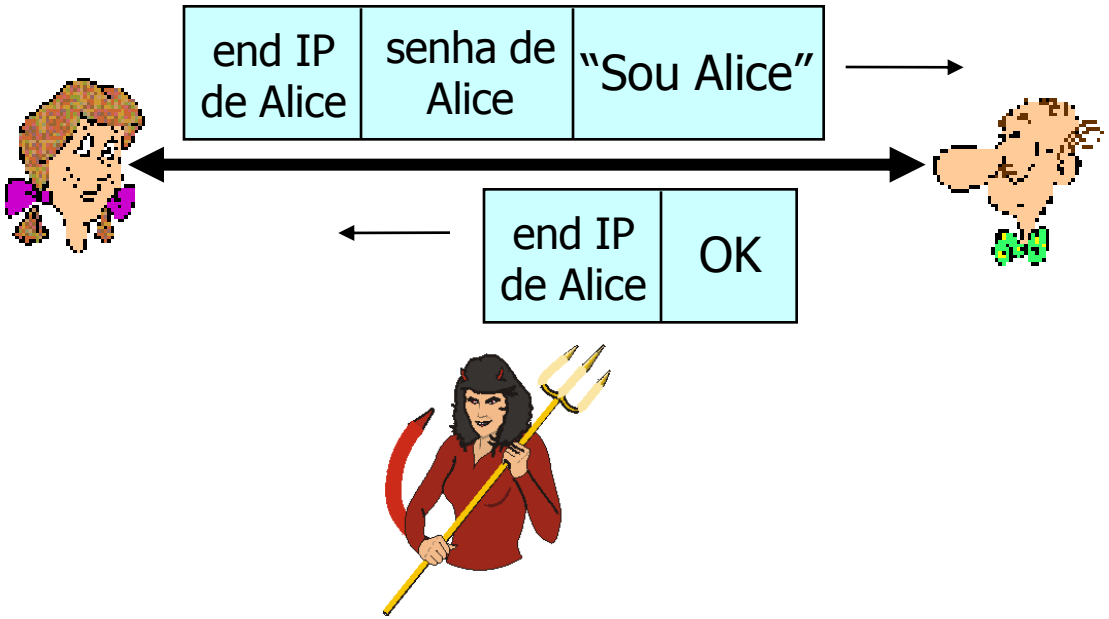


Endereço IP de Alice	"Eu sou Alice"

Trudy pode criar um pacote "imitando" o endereço IP de Alice. Não há verificação do endereço IP da fonte.

Autenticação

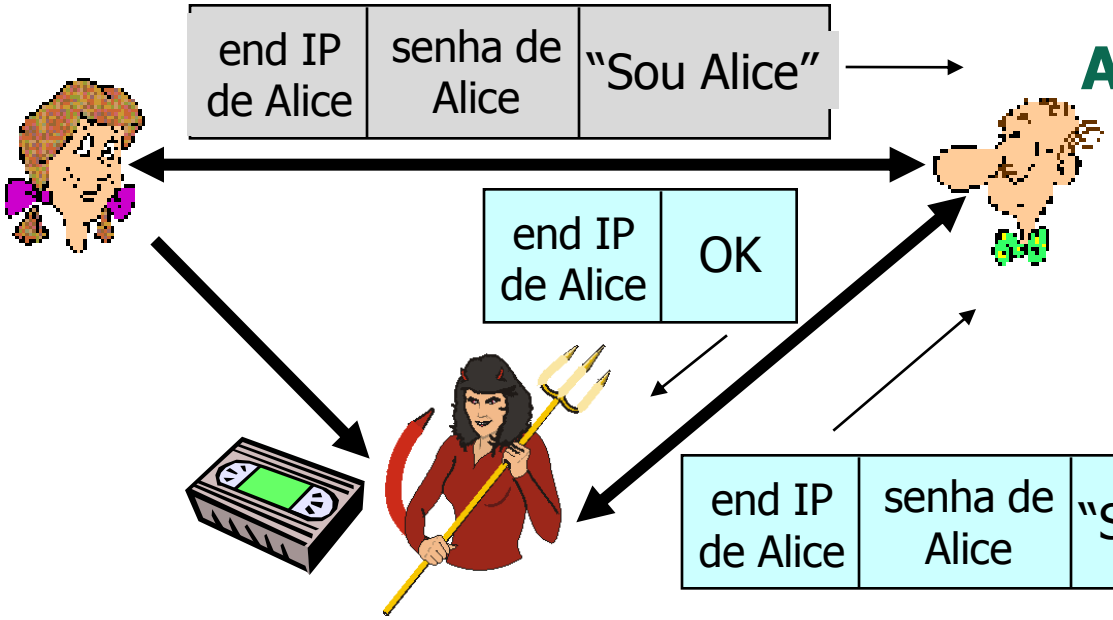
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia a sua senha secreta como "prova"



Cenário de falha?

Autenticação

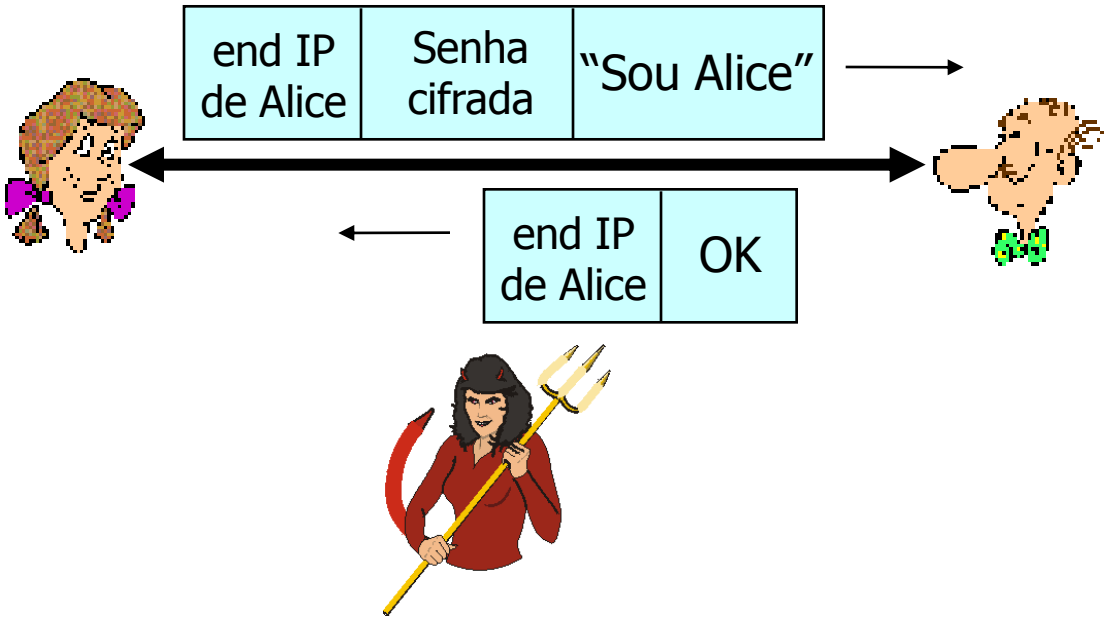
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia a sua senha secreta como "prova"



Ataque de reprodução: Trudy grava o pacote de Alice e depois o envia para Bob

Autenticação

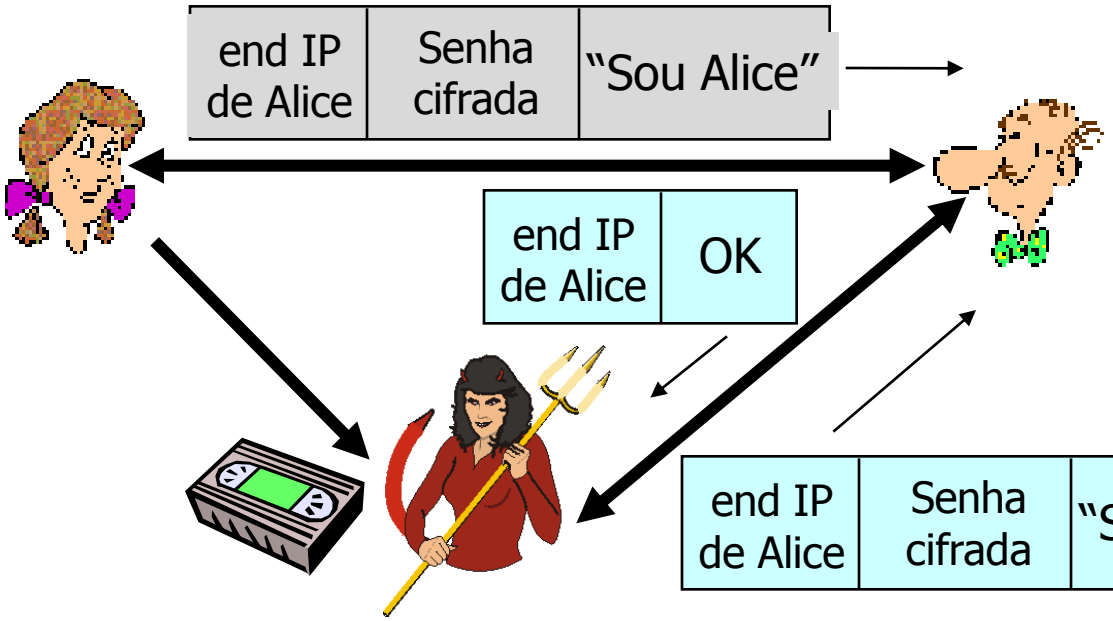
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia a sua senha secreta **cifrada** como "prova"



Cenário de falha?

Autenticação

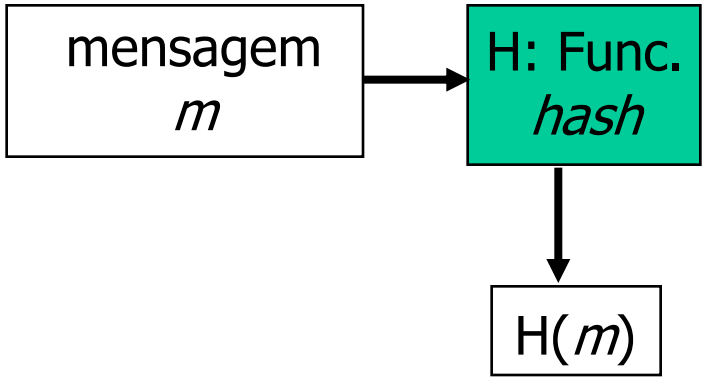
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia a sua senha secreta **cifrada** como "prova"



Ataque de reprodução ainda funciona

Funções *Hash*

- H(.) → função *hash* ou função resumo
- Definição
  - É uma função H(.) que tem como entradas palavras de tamanho arbitrário e fornece como saídas palavras de tamanho fixo → **assinatura da mensagem**
- A função H(.) é muitos-para-um



Funções *Hash*

Instituto de computação

- Propriedades desejadas
  - Fácil de calcular
  - Irreversível
    - É impossível determinar m a partir de H (m)
  - Resistente a colisões
    - Deve ser difícil computacionalmente produzir m e m' tal que H(m) = H(m')
  - Saídas uniformemente distribuídas
    - Aleatórias

Funções *Hash*

Instituto de computação

- Exemplo: Soma de verificação da Internet
  - Possui algumas propriedades de funções *hash*
    - Produz resumo de tamanho fixo da entrada (soma de 16 bits)
    - É muitos-para-um
  - Entretanto…
    - Para uma dada mensagem com um dado valor *hash*, é fácil encontrar uma outra mensagem com o mesmo valor *hash*
      - Colisão!

Funções *Hash*

- Exemplo: soma de verificação da Internet
  - Soma de verificação simplificada
    - Somar pedaços de 4 bytes por 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
	B2 C1 D2 AC

mensagem	Formato ASCII
I O U	49 4F 55
0 0 .	30 30 2E
9 B O B	39 42 D2 42
	B2 C1 D2 AC

Mensagens diferentes, com somas de verificação idênticas

Colisão!

Algoritmos de Funções *Hash*

Instituto de computação

- Função *hash* MD5
  - Amplamente usada
    - Autenticação do Linux
  - Definida pela RFC 1321
  - Computa resumos de mensagens de 128 bits em um processo de 4 passos
    - Enchimento, anexação, acumulação, mistura

- Função *hash* SHA-1
  - É o padrão atual dos EUA
  - Resumos de mensagens de 160 bits

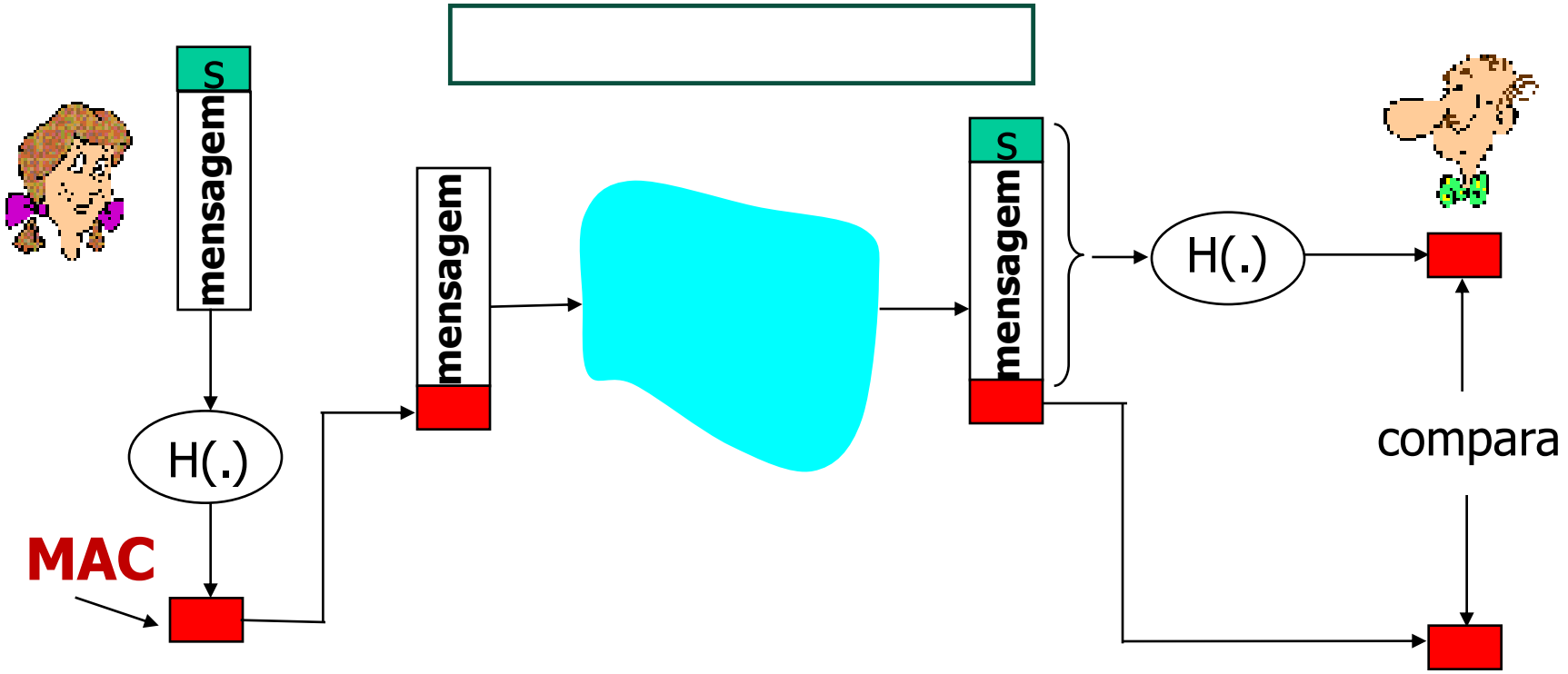
Algoritmos de Funções *Hash*

- Função *hash* MD5
  - Amplamente usada
    - Autenticação do Linux
  - Definida pela RFC 1321
  - Computa resumos de
    de 4
    ção, mistura

Como garantir integridade usando funções *hash*?

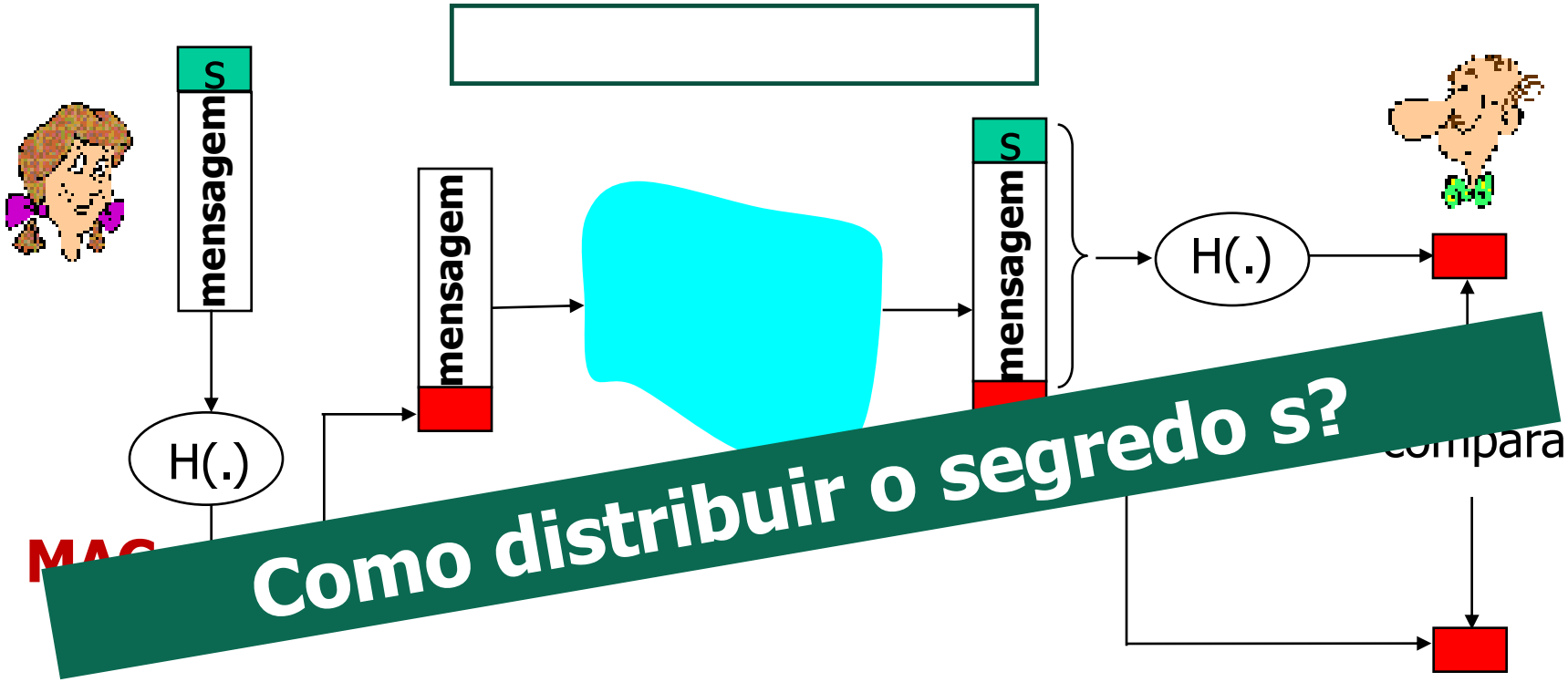
- Função *hash* SHA-1
  - É o padrão atual dos EUA
  - Resumos de mensagens de 160 bits

Message Authentication Code (MAC)



- Autentica o emissor e verifica a integridade da mensagem
- Sem criptografia
- Chamada também de *hash* chaveada (*keyed hash*)
- Notação: $MD_m = H(s||m)$; envia $m||MD_m$

Message Authentication Code (MAC)



- Autentica o emissor e verifica a integridade da mensagem
- Sem criptografia
- Chamada também de *hash* chaveada (*keyed hash*)
- Notação: $MD_m = H(s||m)$; envia $m||MD_m$

HMAC

Instituto de computação

- É um padrão MAC popular
- Aplica a função *hash* duas vezes

1. Concatenar um segredo na frente da mensagem
2. Aplicar a função *hash* à mensagem concatenada
3. Concatenar o segredo na frente do resumo
4. Aplicar a função *hash* novamente à combinação

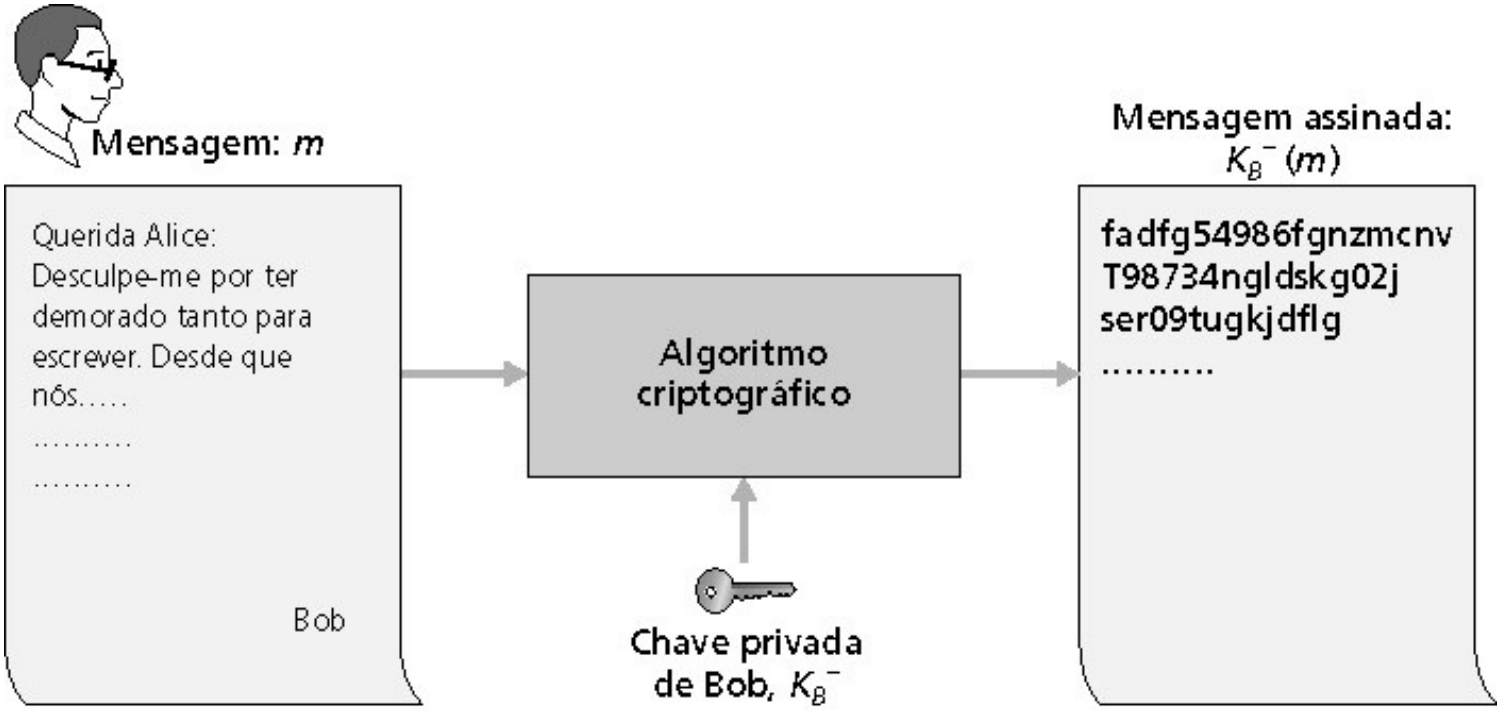
Assinaturas Digitais

Instituto de computação

- Técnica criptográfica análoga às assinaturas à mão
  - Incluir na mensagem algo que seja único e identifique o remetente
- Remetente (Bob) assina digitalmente o documento, atestando que ele é o dono/criador do documento
- Objetivo é **similar ao do MAC**
  - Porém, **usa criptografia de chaves públicas**
- Verificável e não-falsificável
  - Destinatário (Alice) pode provar para alguém que Bob, e ninguém mais (incluindo Alice), assinou o documento

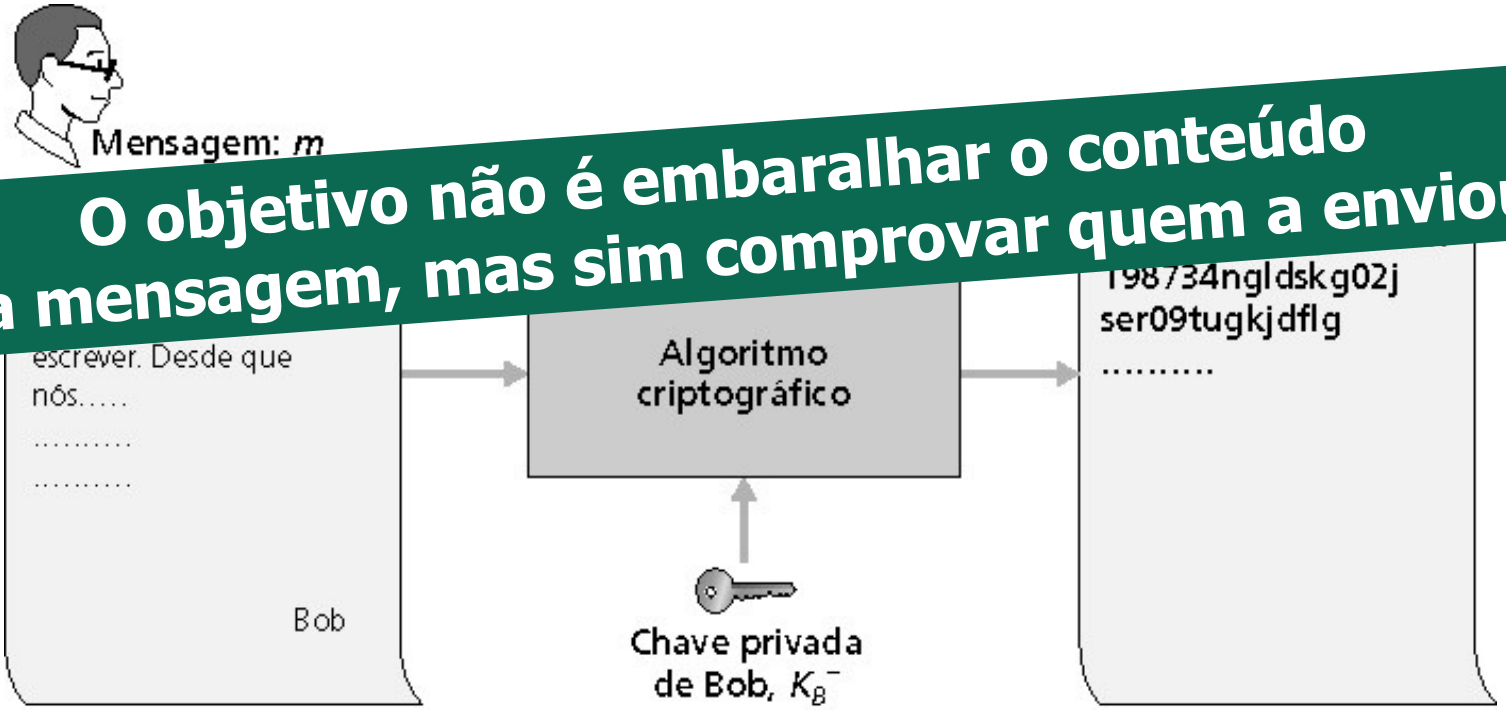
Assinaturas Digitais

- Assinatura digital simples para a mensagem *m*
  - Bob assina *m* cifrando com a sua chave privada K_B^-, criando mensagem "assinada", $K_B^-(m)$



Assinaturas Digitais

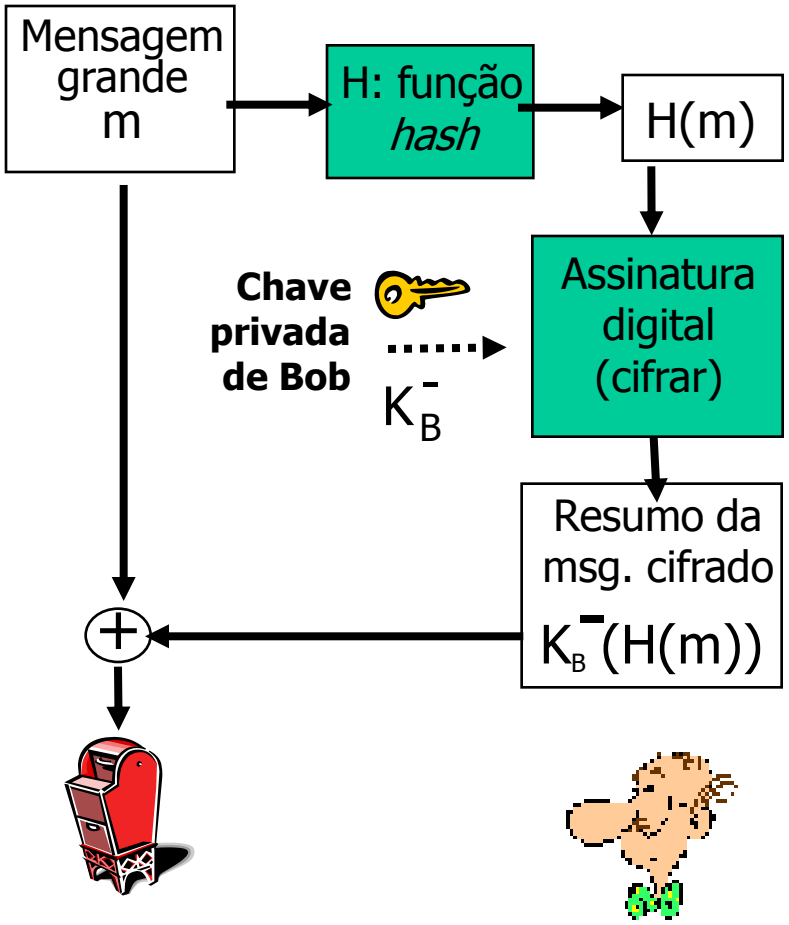
- Assinatura digital simples para a mensagem *m*
  - Bob assina *m* cifrando com a sua chave privada K_B^-, criando mensagem "assinada", $K_B^-(m)$



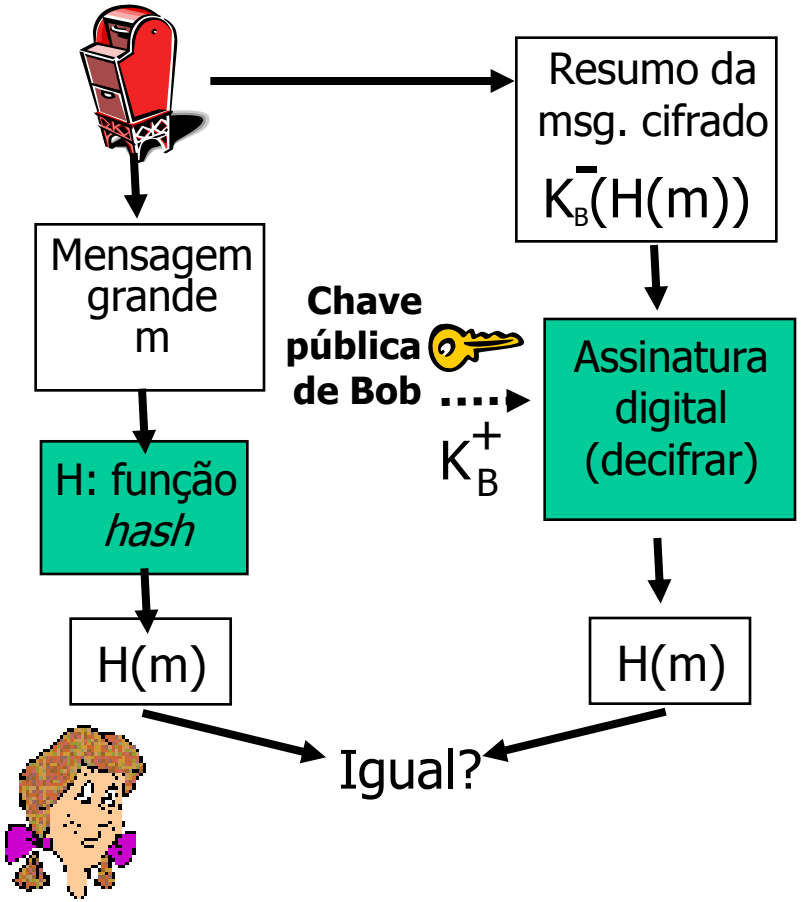
O objetivo não é embaralhar o conteúdo da mensagem, mas sim comprovar quem a enviou

Assinaturas Digitais

Bob envia uma mensagem assinada digitalmente



Alice verifica a **autenticidade** e a **integridade** da mensagem digitalmente assinada



Assinaturas Digitais

Instituto de computação

- Suponha que Alice receba a mensagem *m* e a assinatura digital $K_B(m)$
- Alice verifica que *m* foi assinada por Bob aplicando a chave pública de Bob K_B a $K_B(m)$ depois checa se $K_B(K_B(m)) = m$
- Se $K_B(K_B(m)) = m$, quem quer que tenha assinado *m* deve ter usado a chave privada de Bob

Assinaturas Digitais

Instituto de computação

- Suponha que Alice receba a mensagem *m* e a assinatura digital $K_B(m)$
- Alice verifica que *m* foi assinada por Bob aplicando a chave pública de Bob K_B a $K_B(m)$ depois checa se $K_B(K_B(m)) = m$
- Se $K_B(K_B(m)) = m$, quem quer que tenha assinado *m* deve ter usado a chave privada de Bob

- Alice verifica que
  - Bob assinou *m* → autenticação
  - Ninguém mais assinou *m* → autenticação
  - Bob assinou *m* e não *m'* → integridade

Assinaturas Digitais

- Suponha que Alice receba a mensagem *m* e a assinatura digital $K_B(m)$
- Alice verifica que *m* foi assinada por Bob aplicando a chave pública de Bob K_B a $K_B(m)$ depois checa se $K_B(K_B(m)) = m$
- Se $K_B(K_B(m)) = m$, quem quer que tenha assinado *m* deve ter usado a chave privada de Bob

- Alice verifica que
  - [illegible] assinou *m* → autenticação
  - Bob assinou *m* e não *m′* → integridade

Não-repúdio: Alice pode obter *m* e a assinatura $K_B(m)$ para o tribunal e provar que Bob assinou *m*

Certificação de Chave Pública

Instituto de computação

- Motivação: Trudy passa o trote da pizza para Bob
  - Trudy cria um pedido por email
    - *"Prezada Loja de Pizza, por favor me entregue quatro pizzas de calabresa. Obrigado, Bob"*
  - Trudy assina o pedido com sua chave privada
  - Trudy envia o pedido para a Loja de Pizza
  - Trudy envia para a Loja de Pizza a sua chave pública, mas diz que essa chave é a da Bob
  - A Loja de Pizza verifica a assinatura e entrega as quatro pizzas para Bob

Certificação de Chave Pública

- Motivação: Trudy passa o trote da pizza para Bob
  - Trudy cria um pedido por email
    - *"Prezada Loja de Pizza, por favor me entregue quatro pizzas de calabresa. Obrigado, Bob"*
  - Trudy assina [illegible] Loja de Pizza a sua chave pública, mas diz que essa chave é a da Bob
  - A Loja de Pizza verifica a assinatura e entrega as quatro pizzas para Bob

É preciso verificar se a chave pública é verdadeira!

Intermediários de Confiança

Instituto de computação

- Problema com chave pública
  - Quando Alice obtém a chave pública de Bob (da web, e-mail ou disquete), como ela vai saber se a chave pública é mesmo de Bob e não de Trudy?

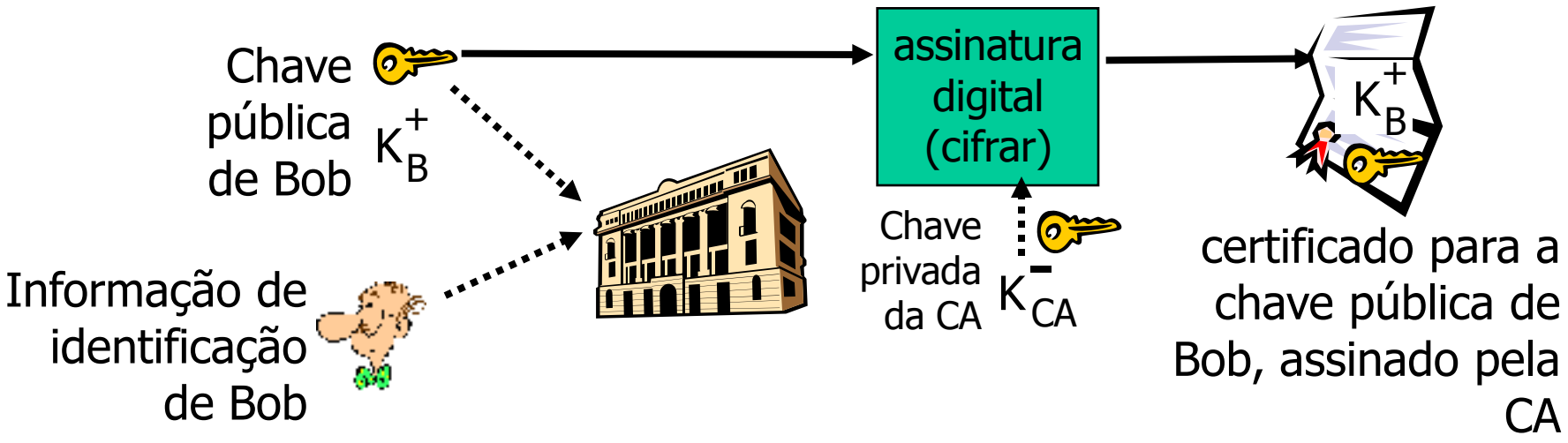
- Solução
  - Autoridade certificadora confiável (CA)

Autoridades Certificadoras (CAs)

- Associam uma chave pública a uma entidade particular, E
  - Entidade E (pessoa, roteador) registra a sua chave pública com a CA
  - Entidade E fornece "prova de identidade" à CA
  - CA cria **certificado** associando E à sua chave pública
  - Certificado contém a chave pública de E assinada digitalmente pela CA
    - CA diz que "esta é a chave pública de E"

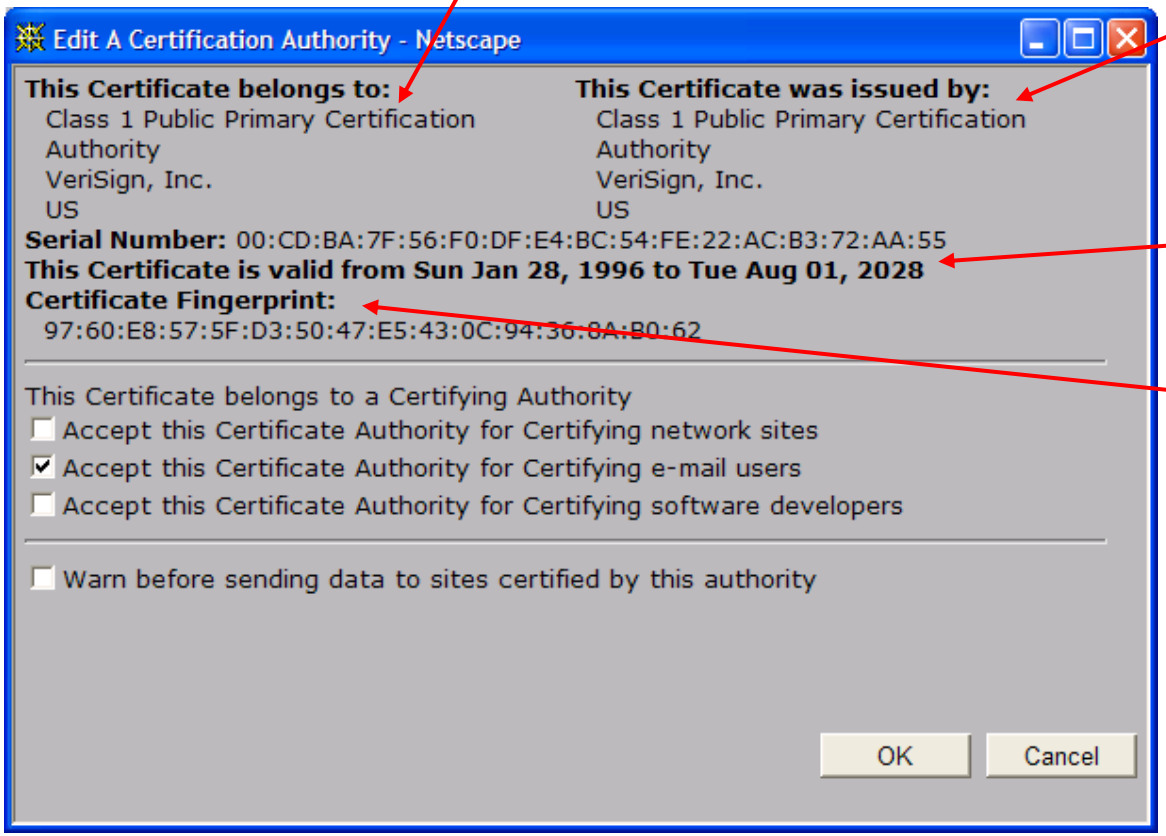
Autoridades Certificadoras (CAs)

- Associam uma chave pública a uma entidade particular, E



Exemplo de Certificado

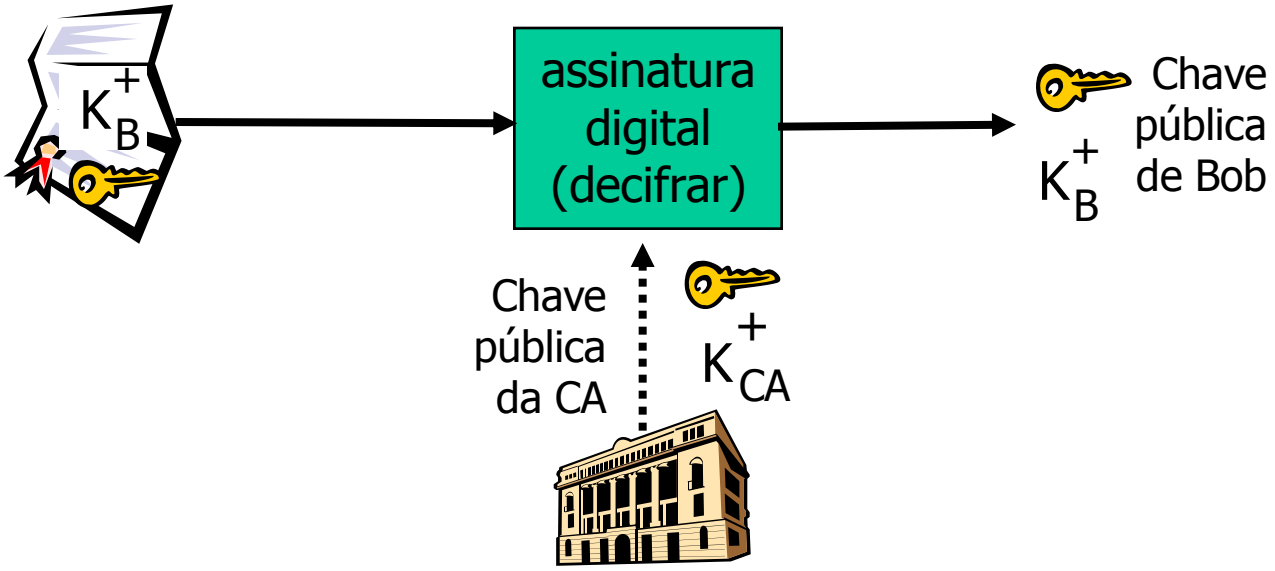
- Número de série (único para cada emissor)
- info sobre o proprietário do certificado, incluindo o algoritmo e o valor da chave propriamente dita (não apresentada)



- info sobre o emissor do certificado
- datas de validade
- assinatura digital do emissor

Autoridades Certificadoras (CAs)

- Quando Alice precisa da chave pública de Bob
  - Obtém o certificado de Bob (de Bob ou de outro lugar)
  - Aplica a chave pública da CA ao certificado de Bob, obtém a chave pública de Bob



Certificados: Resumo

Instituto de computação

- Padrão primário X.509 (RFC 2459)
- Um certificado contém
  - Nome do emissor
  - Nome, endereço, domínio etc. da entidade
  - Chave pública da entidade
  - Assinatura digital (assinada com a chave privada do emissor)
- Infraestrutura de chaves públicas (*Public-Key Infrastructure* - PKI
  - Composta por certificados e autoridades certificadoras
  - Considerada "pesada" computacionalmente

Autenticação do Ponto Final

Instituto de computação

- Deseja-se ter certeza do emissor da mensagem

- MAC provê autenticação do ponto final?
  - Assumindo que Alice e Bob possuem um **segredo compartilhado**
  - É possível afirmar que Alice **criou** a mensagem?
  - É possível afirmar que Alice **enviou** a mensagem?

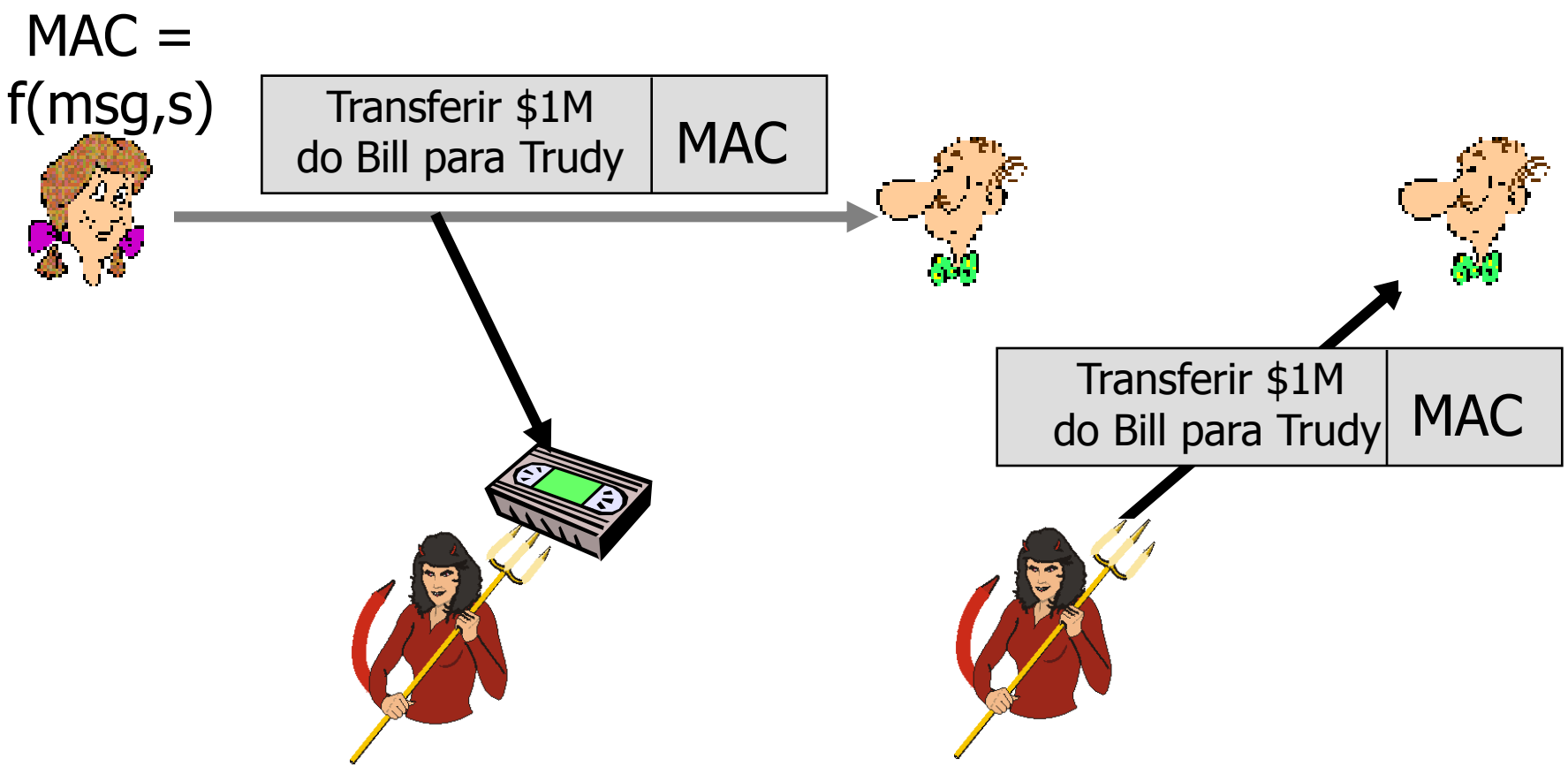
Autenticação do Ponto Final

- Deseja-se ter certeza do emissor da mensagem

- MAC provê autenticação do ponto final?
  - Assumindo que Alice e Bob possuem um **segredo compartilhado**
  - É possível afirmar que Alice **criou** a mensagem? **Sim!**
  - É possível afirmar que Alice **enviou** a mensagem? **Não!**

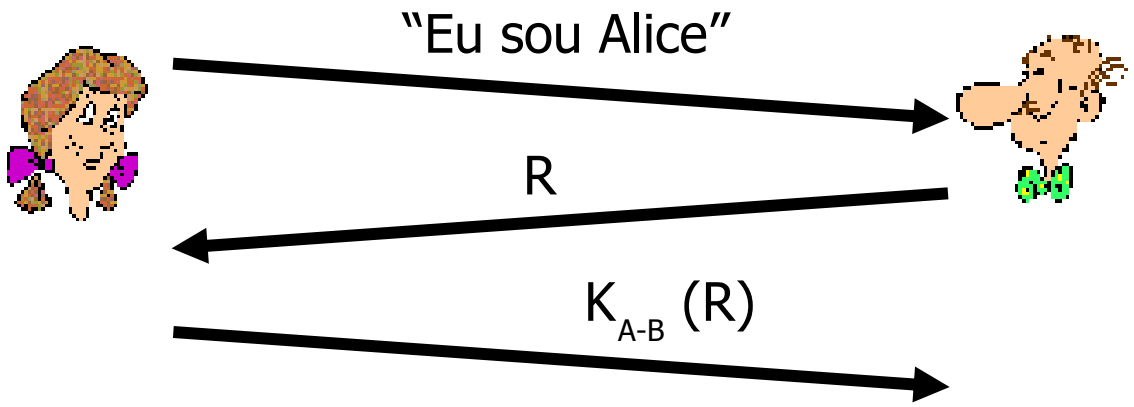
Ataque de Reprodução

Ataque de Reprodução



Defesa ao Ataque de Reprodução

- Objetivo: evitar o ataque de reprodução
  - Solução: uso de *nonces*
    - É um número (R) usado apenas uma vez na vida

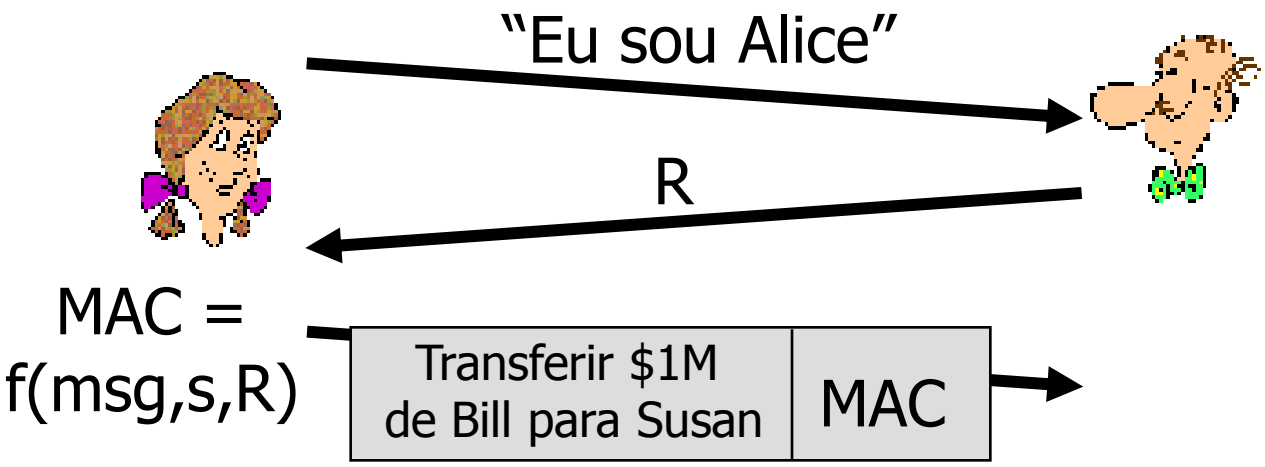


• Para identificar Alice "ao vivo", Bob envia para Alice um *nonce* R em "claro"

• Alice deve retornar R, cifrado com a chave secreta compartilhada que **apenas** eles conhecem

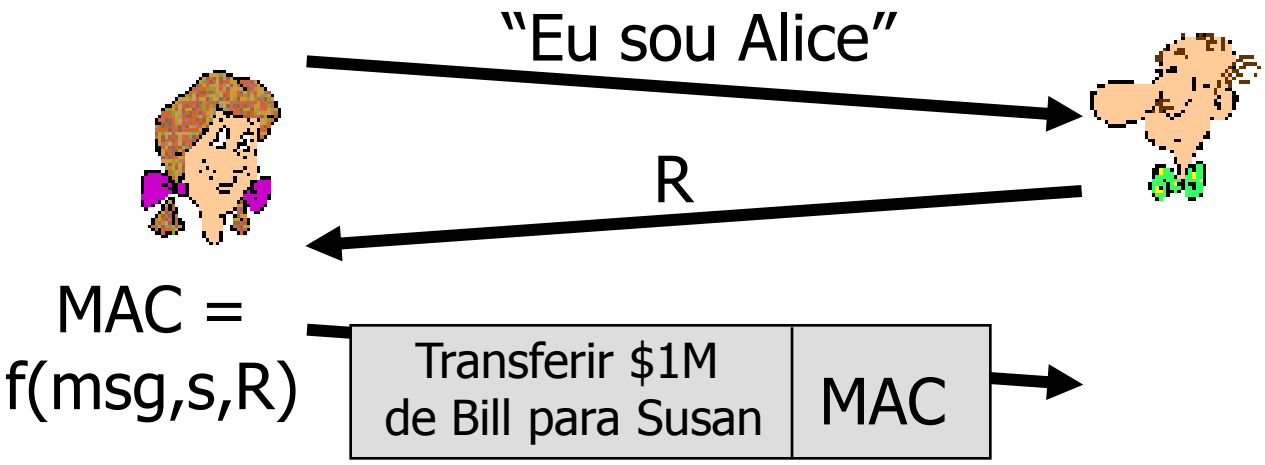
Defesa ao Ataque de Reprodução

- Uso de *nonces* com MAC



Defesa ao Ataque de Reprodução

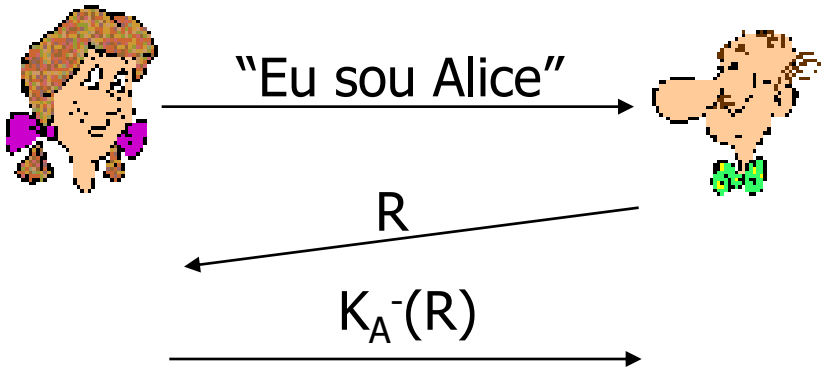
- Uso de *nonces* com MAC



Problema: requer chave secreta compartilhada

Autenticação com Criptografia de Chave Pública

- *Nonces* + chave pública? → evitar o segredo compartilhado



Envie a sua chave pública K_A^+

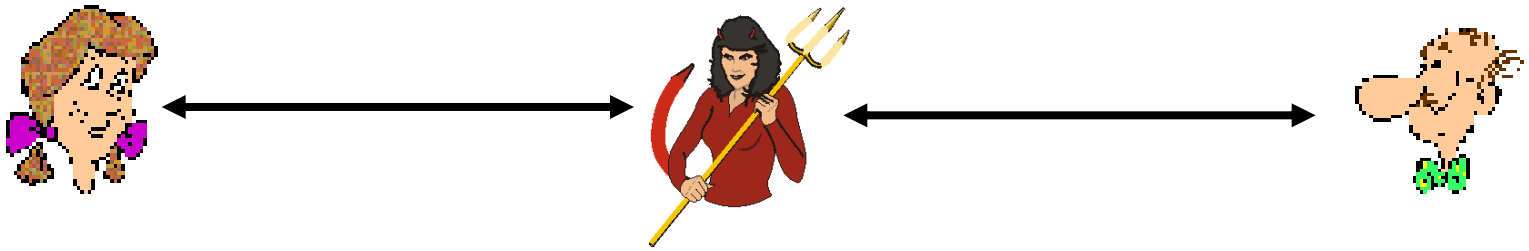
K_A^+

Bob processa
$K_A^+(K_A^-(R))=R$
autenticando Alice

Só Alice conhece K_A^-

Ataque do Homem-no-Meio

- Trudy posa como sendo Alice (para Bob) e como sendo Bob (para Alice)



- Difícil de detectar
  - Bob recebe tudo o que Alice envia, e vice versa
    - Ex.: Alice podem se encontrar uma semana depois e lembrar da conversa
  - O problema é que Trudy também recebe todas as mensagens!

Ataque do Homem-no-Meio

"Eu sou Alice"

"Eu sou Alice"

R

$K_T^-(R)$

R

$K_A^-(R)$

Envie a sua chave pública K_A^+

Envie a sua chave pública K_A^+

K_T^+

K_A^+

$K_T^+(m)$

$K_A^+(m)$

Trudy recebe
$m = K_T^-(K_T^+(m))$
envia m para Alice
cifrada com a chave
pública de Alice

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

Instituto de computação

Seguranças nas diferentes camadas

Segurança nas Diferentes Camadas

Instituto de computação

- Camada de aplicação
  - Ex.: correio eletrônico
- Camada de transporte
  - SSL
- Camada de rede
  - IPSec e redes virtuais privadas
- Camada de enlace
  - Redes sem-fio IEEE 802.11

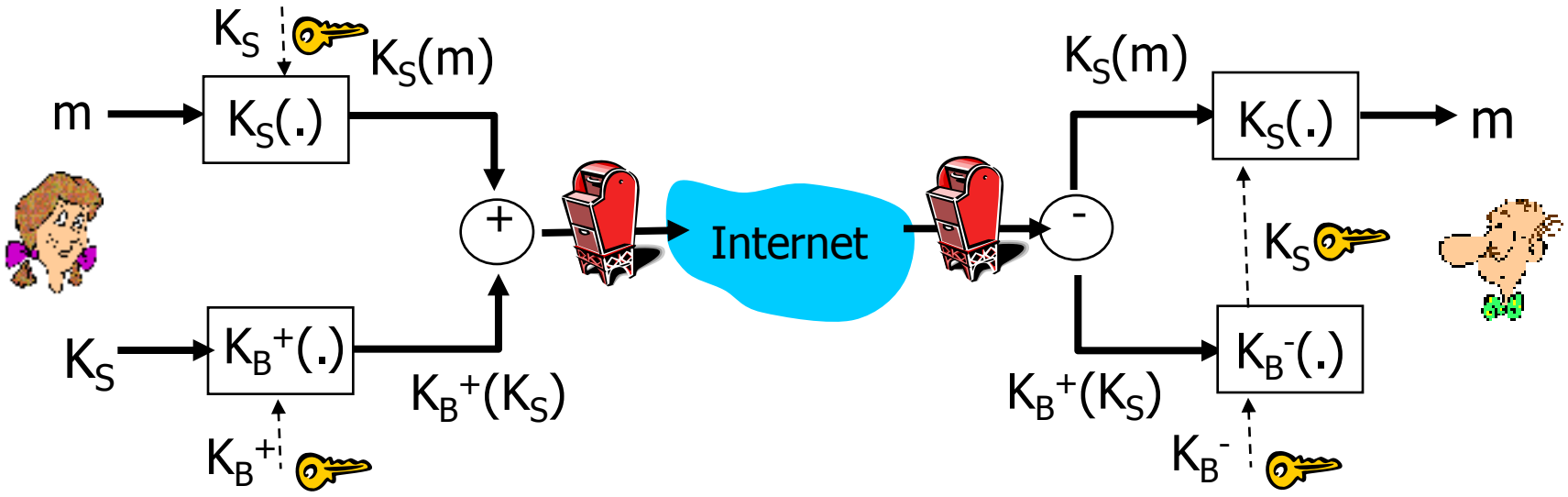
Lembrete: uma camada oferece serviços para as camadas superiores

Instituto de computação

Camada de Aplicação

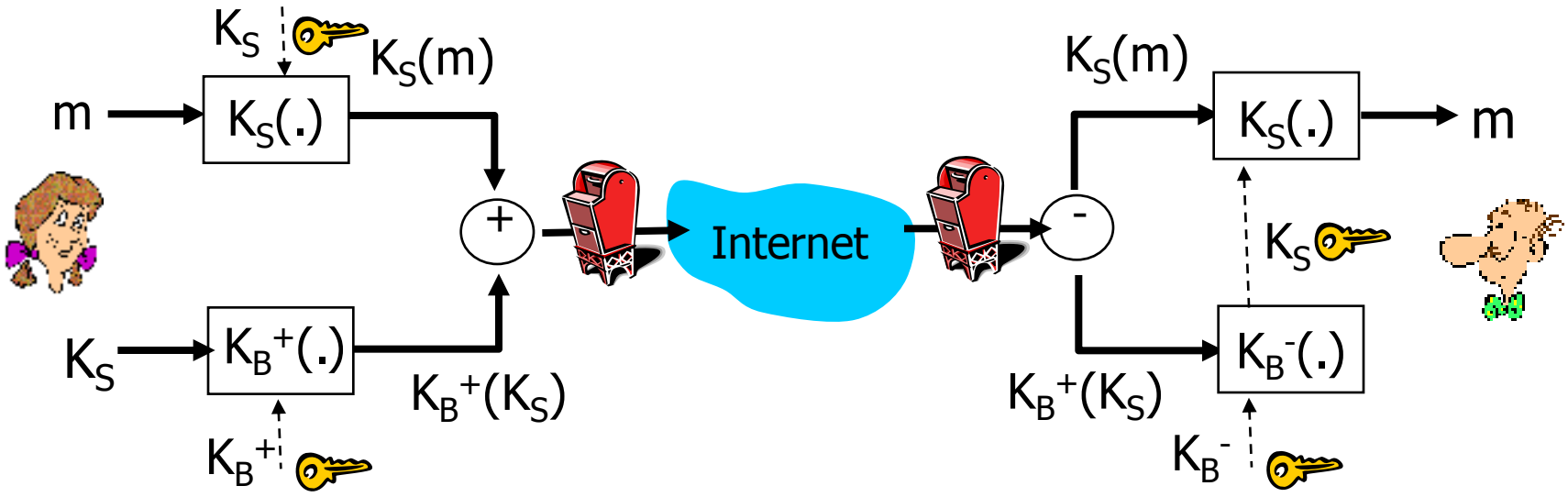
Email Seguro

- Alice quer enviar um email **confidencial** *m* para Bob, ela deve
  - Gerar uma chave de sessão simétrica privada aleatória K_S
  - Cifrar a mensagem com K_S
  - Cifrar K_S com a chave pública de Bob K_B
  - Enviar $K_S(m)$ e $K_B(K_S)$ para Bob



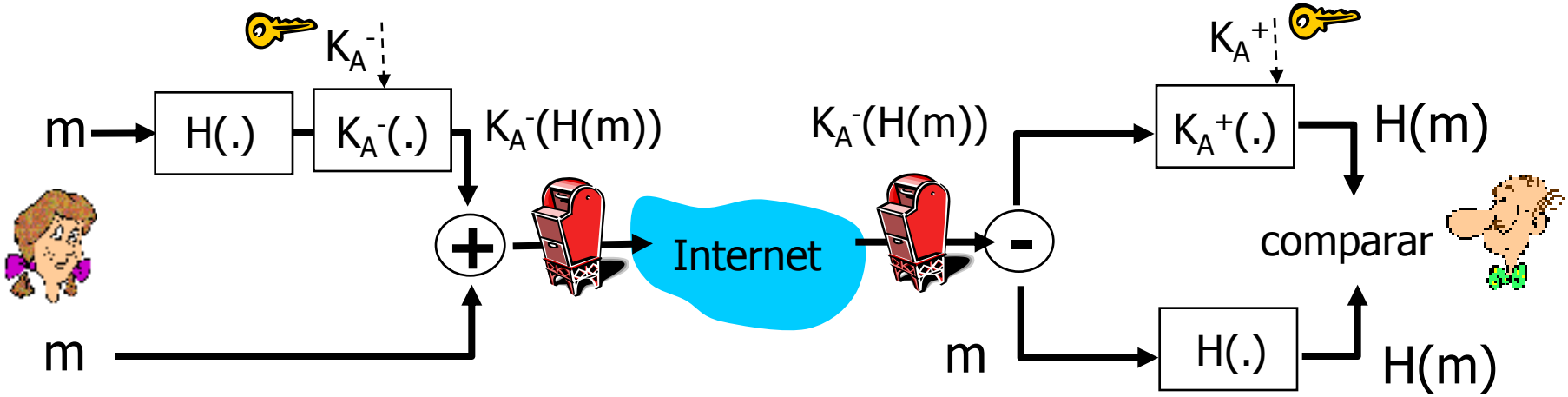
Email Seguro

- Bob
  - Usa a sua chave privada para decifrar e recuperar K_S
  - Usa K_S para decifrar $K_S(m)$ e recuperar m



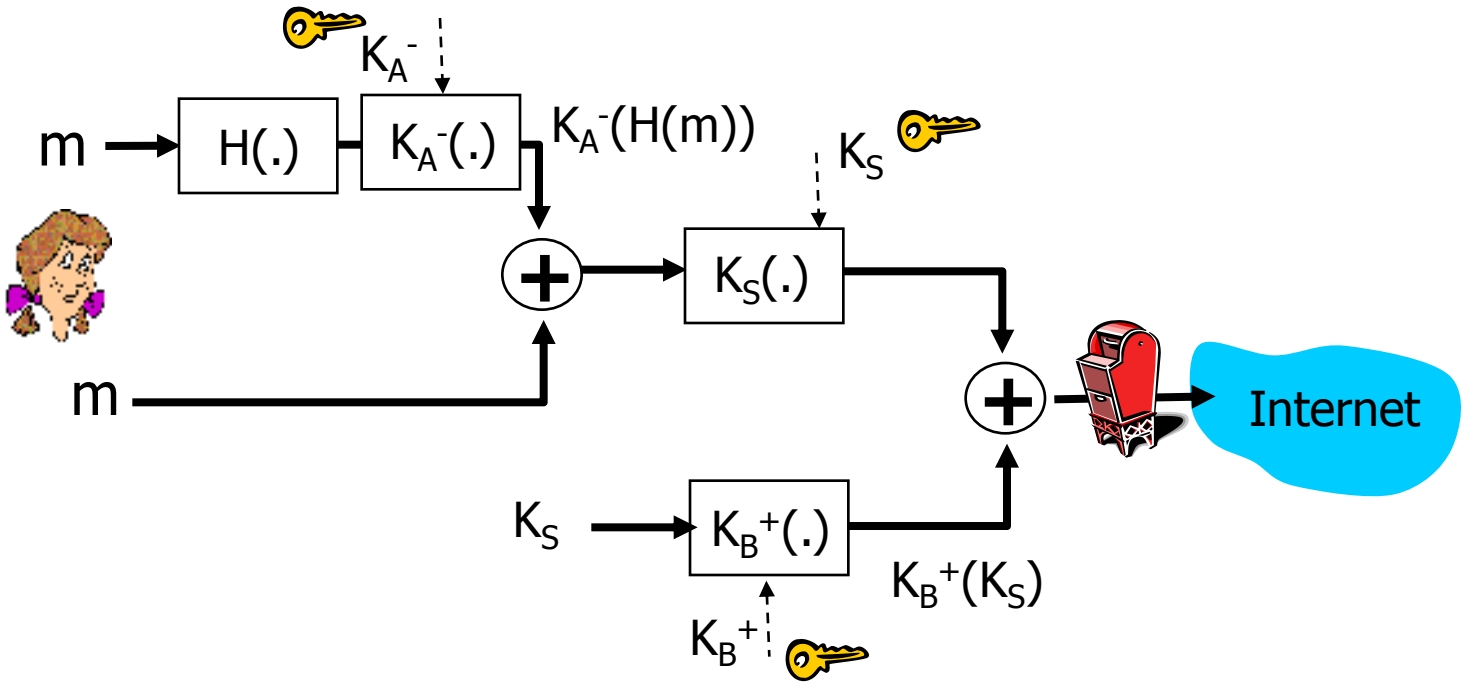
Email Seguro

- Alice quer prover **autenticação** do emissor e **integridade** da mensagem
- Alice assina digitalmente a mensagem
  - Envia a mensagem em claro e a assinatura digital



Email Seguro

- Alice quer prover **confidencialidade, autenticação** do emissor e **integridade** da mensagem
  - Deve usar três chaves: sua chave privada, a chave pública de Bob e uma chave simétrica recém-criada(sessão)



Pretty good privacy (PGP)

Instituto de computação

- Esquema de criptografia de e-mails para a Internet
  - É um padrão de fato
- Usa
  - Criptografia de chave simétrica
  - Criptografia de chave pública
  - Função *hash*
  - Assinatura digital
- Provê
  - Confidencialidade
  - Autenticação do transmissor
  - Integridade

Pretty good privacy (PGP)

Instituto de computação

- Exemplo

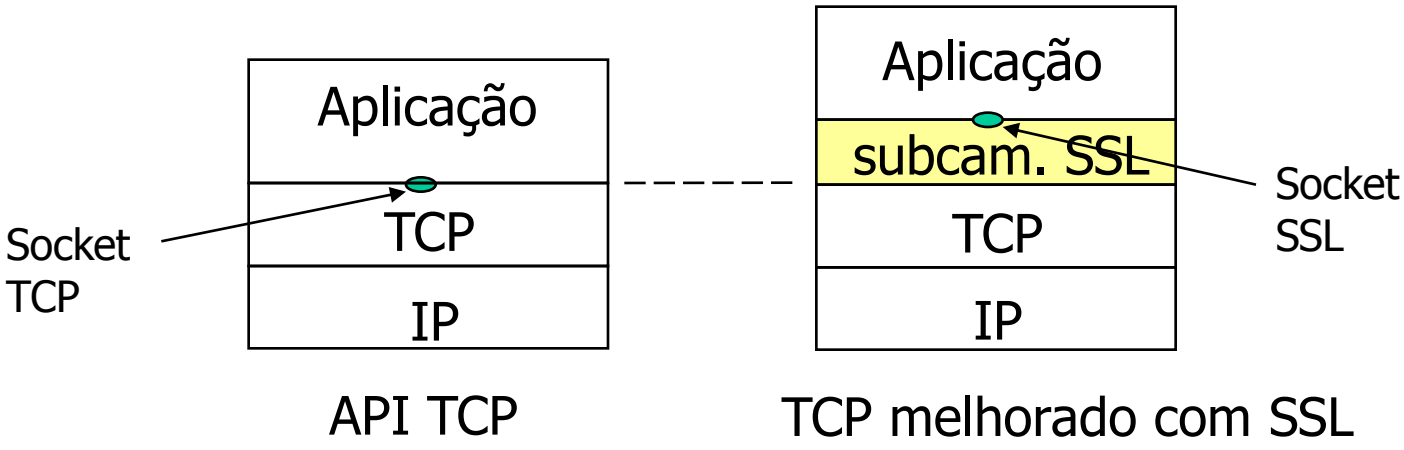
```
---BEGIN PGP SIGNED MESSAGE---
Hash: SHA1

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

---BEGIN PGP SIGNATURE---
Version: PGP 5.0
Charset: noconv
yhHJRHhGJGhgg/12EpJ+lo8gE4vB3mqJ
   hFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

Instituto de computação

Camada de Transporte

Secure Sockets Layer (SSL)

- SSL trabalha na camada de transporte
  - Provê segurança para qualquer aplicação baseada em TCP que use os serviços SSL
- Serviços de segurança SSL
  - Autenticação do servidor, codificação dos dados, autenticação do cliente (opcional)



Secure Sockets Layer (SSL)

Instituto de computação

- Protocolo de segurança muito usado
  - Suportado por quase todos os navegadores e servidores Web
    - Desenvolvido pela Netscape em 1993
  - Usado para implementar o **`https`**
  - Dezenas de bilhões de dólares gastos anualmente no seu desenvolvimento

- SSLv3 é a base do TLS (*Transport Layer Security)*
  - Padrão do IETF, RFC 2246

Secure Sockets Layer (SSL)

Instituto de computação

- Provê
  - Confidencialidade
  - Integridade
  - Autenticação

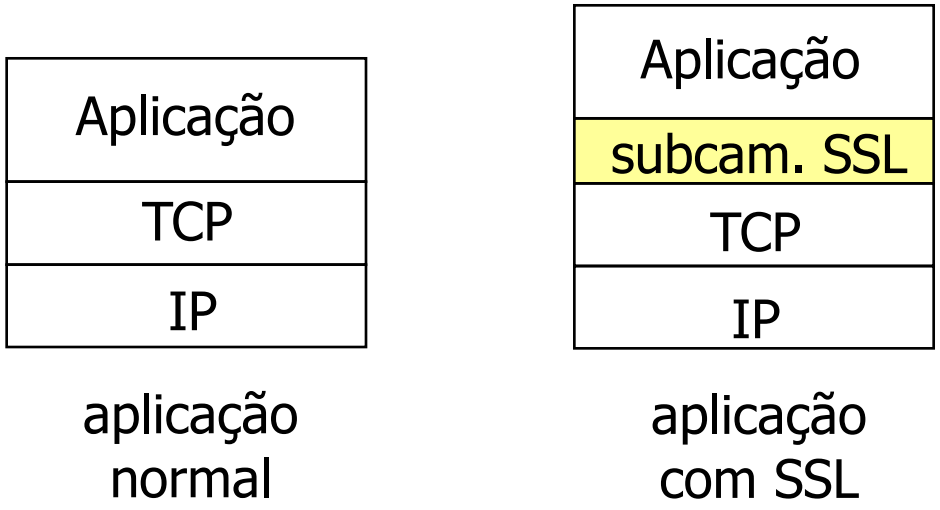
Secure Sockets Layer (SSL)

Instituto de computação

- Objetivos originais
  - Permitir transações de comércio eletrônico na Web
  - Cifragem (ex.: números de cartões de créditos e senhas)
  - Autenticação o servidor Web
  - Autenticação opcional do cliente
  - Mínimo incômodo em fazer negócios com novo parceiro
  - Disponível para todas as aplicações TCP
  - Interface de *socket* seguro

SSL e TCP/IP

- SSL provê para as aplicações uma API (*Application Programming Interface*)
  - Bibliotecas/classes SSL em C e Java estão disponíveis



Quase-SSL: Canal Seguro Simples

Instituto de computação

- Versão simplificada: quatro fases

1. Inicialização ou Apresentação (*handshake*)
   - Bob quer estabelecer uma conexão TCP com Alice
   - Alice e Bob usam seus certificados e chaves privadas para autenticarem um ao outro e trocar a chave secreta compartilhada

2. Derivação das chaves
   - Alice e Bob usam o segredo compartilhado para derivar um conjunto de chaves

Quase-SSL: Canal Seguro Simples

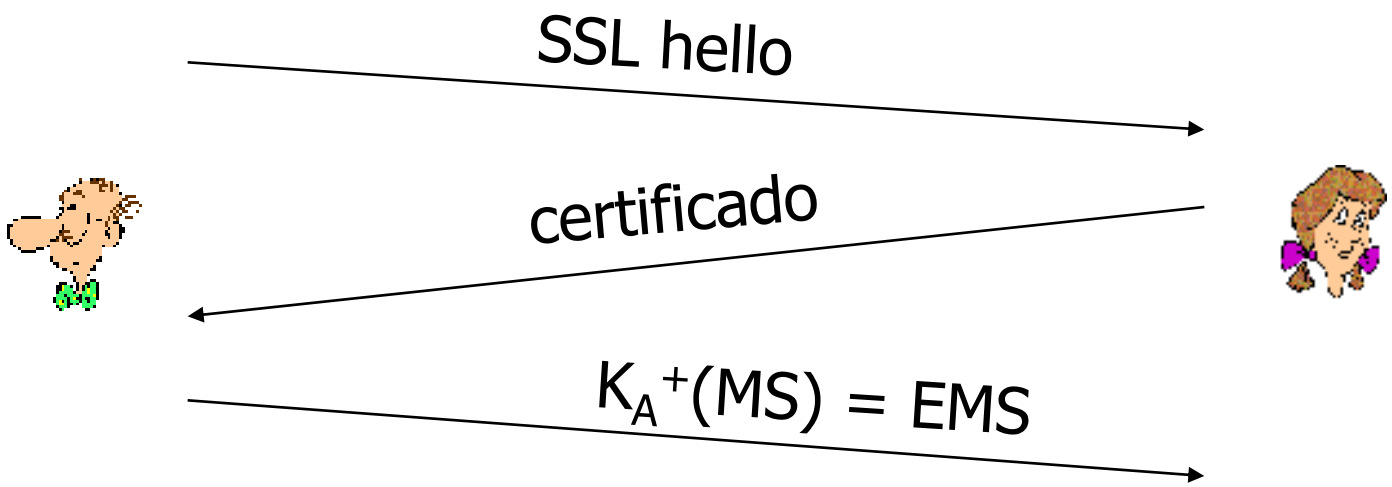
Instituto de computação

3. Transferência de dados
   - Dados a serem transferidos são divididos em uma série de registros

4. Encerramento de conexão
   - Mensagens especiais são trocadas para encerrar uma conexão de forma segura

Quase-SSL: Incialização Simples

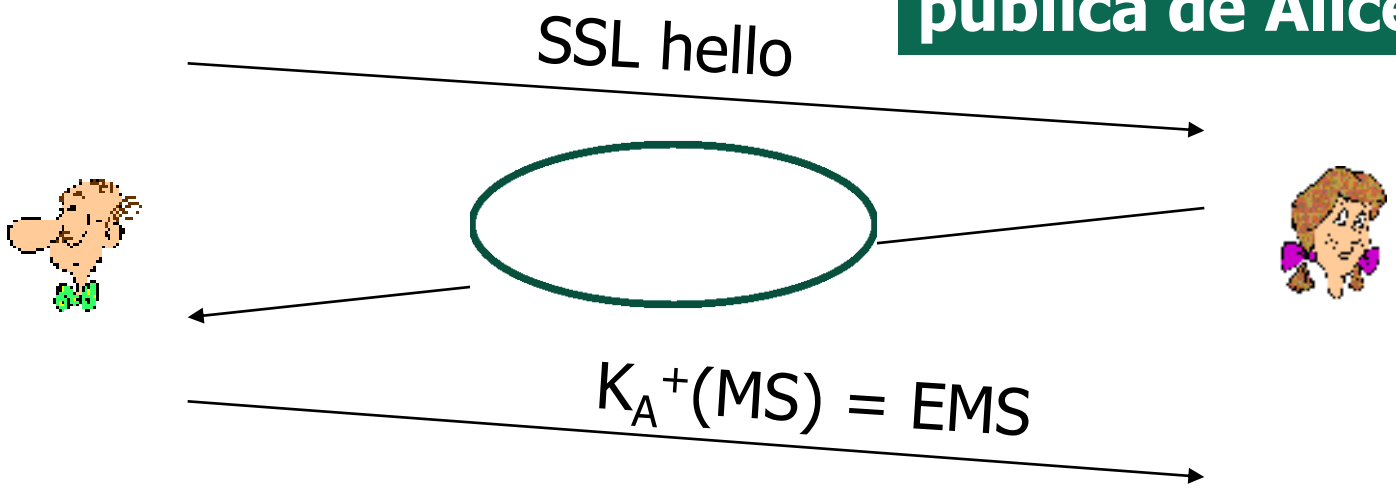
- Primeiro passo: estabelecer uma conexão TCP
  - *Three way handshake* "normal"
- Próximos passos: autenticar e distribuir uma chave secreta
  - MS = segredo mestre
  - EMS = segredo mestre cifrado



Quase-SSL: Incialização Simples

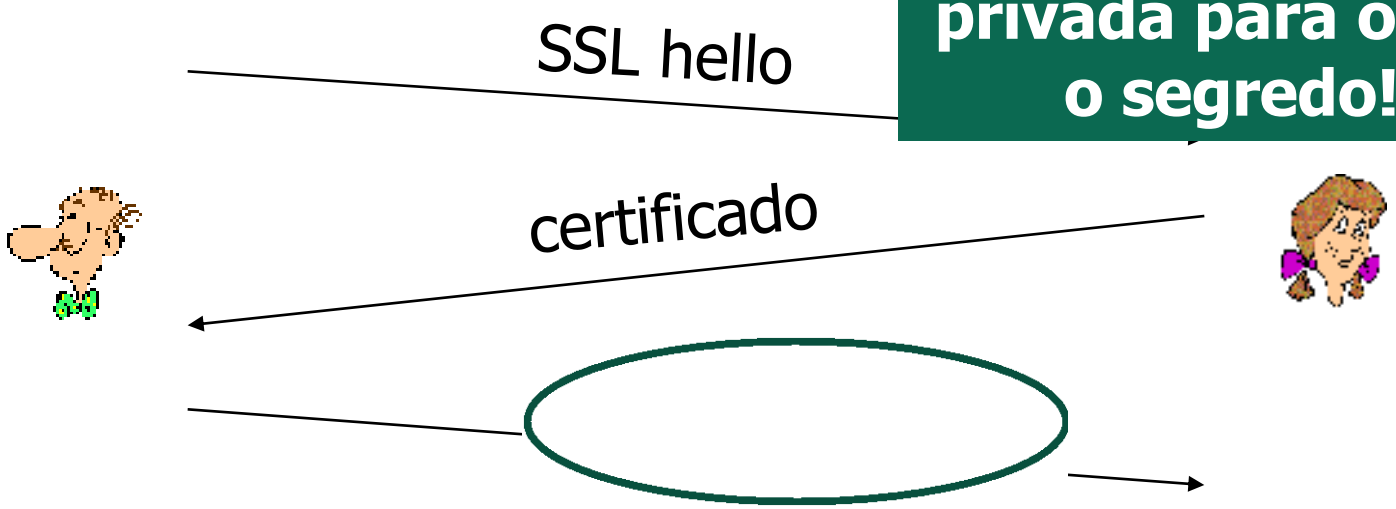
- Primeiro passo: estabelecer uma conexão TCP
  - *Three way handshake* "normal"
- Próximos passos: autenticar e distribuir uma chave secreta
  - MS = segredo mestre
  - EMS = segredo mestre cifrado

Contém a chave pública de Alice!



Quase-SSL: Incialização Simples

- Primeiro passo: estabelecer uma conexão TCP
  - *Three way handshake* "normal"
- Próximos passos: autenticar e distribuir uma chave secreta
  - MS = segredo mestre
  - EMS = segredo mestre cifrado



Quase-SSL: Derivação das Chaves

Instituto de computação

- Recomenda-se não usar a mesma chave para mais de uma operação criptográfica
  - Deve-se usar chaves diferentes para cifrar e gerar códigos de autenticação de mensagens (MAC)
- Quatro chaves
  - K_C = chave para cifrar os dados enviados do cliente para o servidor
  - M_C = chave MAC para os dados enviados do cliente para o servidor
  - K_S = chave para cifrar os dados enviados do servidor para o cliente
  - M_S = chave MAC para os dados enviados do servidor para o cliente

Quase-SSL: Derivação das Chaves

Instituto de computação

- Chaves são derivadas a partir de uma função derivadora de chaves (*Key Derivation Function* - KDF)
  - Usa o segredo mestre e (possivelmente) mais dados aleatórios adicionais e cria as chaves

- Mais detalhes adiante...

Quase-SSL: Registros de Dados

Instituto de computação

- Por que não cifrar os dados em um fluxo constante à medida que são criados os segmentos TCP?
  - Onde colocar o MAC?
    - Se colocá-lo no fim, não há como verificar a integridade da mensagem até que ele seja toda processada
    - Exemplo
      - Em um serviço de mensagens instantâneas, como se pode verificar a integridade de todos os bytes enviados antes da exibição?

Quase-SSL: Registros de Dados

Instituto de computação

- Ao invés disso, dividir o fluxo em uma série de **registros**
  - Cada registro possui um MAC
  - Registros são cifrados e enviados pelo receptor
  - Receptor pode agir sobre cada registro assim que ele o recebe

- Problema: registros tem tamanho variável
  - Em um registro, o receptor deve distinguir o MAC dos dados

Quase-SSL: Registros de Dados

Instituto de computação

- Ao invés disso, dividir o fluxo em uma série de **registros**
  - Cada registro possui um MAC
  - Registros são cifrados e enviados pelo receptor
  - Receptor pode agir sobre cada registro assim que ele o recebe

- Problema: registros tem tamanho variável
  - Em um registro, o receptor deve distinguir o MAC dos dados
  - Adicionar campo de comprimento

Quase-SSL: Números de Sequência

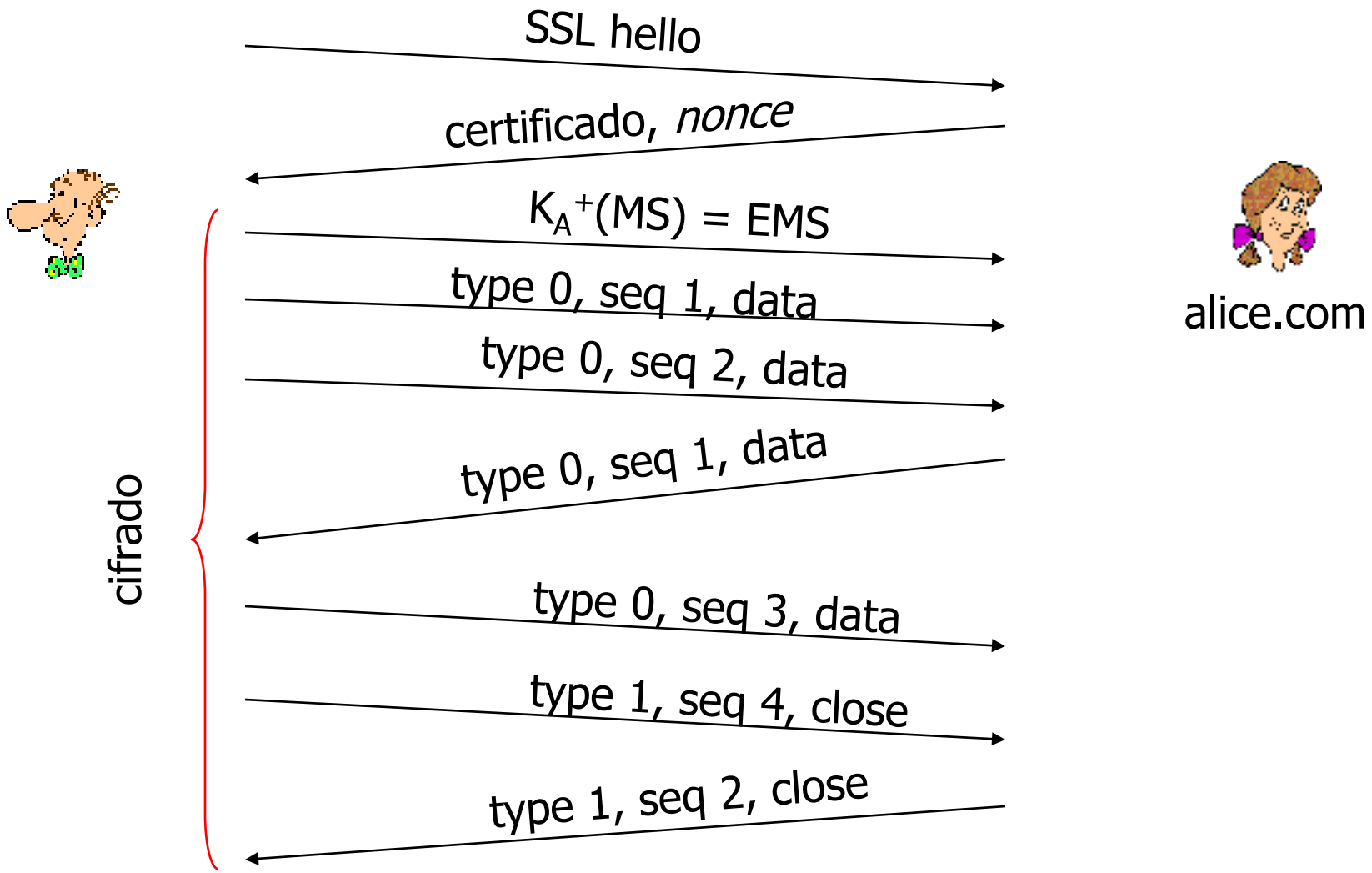
Instituto de computação

- Um atacante pode capturar e reproduzir um registro ou reordenar os registros
  - Lembrete: números de sequência do TCP não serão cifrados
- Solução: adicionar um número de sequência ao MAC
  - MAC = MAC(M_X, sequência||dados)
  - Não há um campo para o número de sequência em cada registro
  - Contadores no cliente e no servidor
- Um atacante ainda pode reproduzir todos os registros
  - Solução: usar *nonces* aleatórios

Quase-SSL: Controle de Informação

- Ataque de Encerramento Antecipado (*trunking*)
  - Atacante forja o segmento TCP de encerramento de conexão
  - Um ou os dois lados pensam que existe menos dados do que realmente há
- Solução: tipos de registros, com um tipo para encerramento
  - Tipo 0 para dados; Tipo 1 para encerramento
- MAC = MAC(M_X, sequência||tipo||dados)

Quase-SSL: Resumo



Quase-SSL não está completo

Instituto de computação

- Qual o comprimento dos campos?
- Quais são os protocolos criptográficos?
- Sem negociação?
  - Permite que cliente e servidor suportem diferentes algoritmos de cifração
  - Permite que cliente e servidor escolham juntos algoritmos específicos antes da transferência de dados

Conjunto de Cifradores

Instituto de computação

- Composto por
  - Algoritmos de chave pública
  - Algoritmos de cifração simétrica
  - Algoritmos MAC
- SSL suporta diversos conjuntos de cifradores
- Negociação
  - Cliente e servidor devem concordar com o conjunto a ser usado
- Cliente oferece as alternativas; servidor escolhe uma

Conjunto de Cifradores

Instituto de computação

- Cifradores simétricos mais usados com o SSL
  - DES – Data Encryption Standard: bloco
  - 3DES – Triple strength: bloco
  - RC2 – Rivest Cipher 2: bloco
  - RC4 – Rivest Cipher 4: fluxo
- Algoritmos de criptografia de chaves públicas
  - RSA

SSL Real: Inicialização

Instituto de computação

- Propósito
  1. Autenticação do servidor
  2. Negociação: acordo sobre os algoritmos criptográficos
  3. Estabelecimento de chaves
  4. Autenticação do cliente (opcional)

SSL Real: Inicialização

Instituto de computação

1. Cliente envia uma lista de algoritmos que ele suporta juntamente do seu *nonce*
2. Servidor escolhe os algoritmos da lista; envia de volta: escolha + certificado + *nonce* do servidor
3. Cliente verifica o certificado, extrai a chave pública do servidor, gera segredo pré-mestre, cifra com a chave pública do servidor; envia para o servidor
4. Cliente e servidor independentemente computam as chaves de cifração e MAC keys a partir do segredo pré-mestre e dos *nonces*

SSL Real: Inicialização

Instituto de computação

5. Cliente envia um MAC de todas as mensagens de incialização
6. Servidor envia um MAC de todas as mensagens de incialização

SSL Real: Inicialização

Instituto de computação

- **Problema**: Homem-no-meio pode apagar os mais fortes da lista

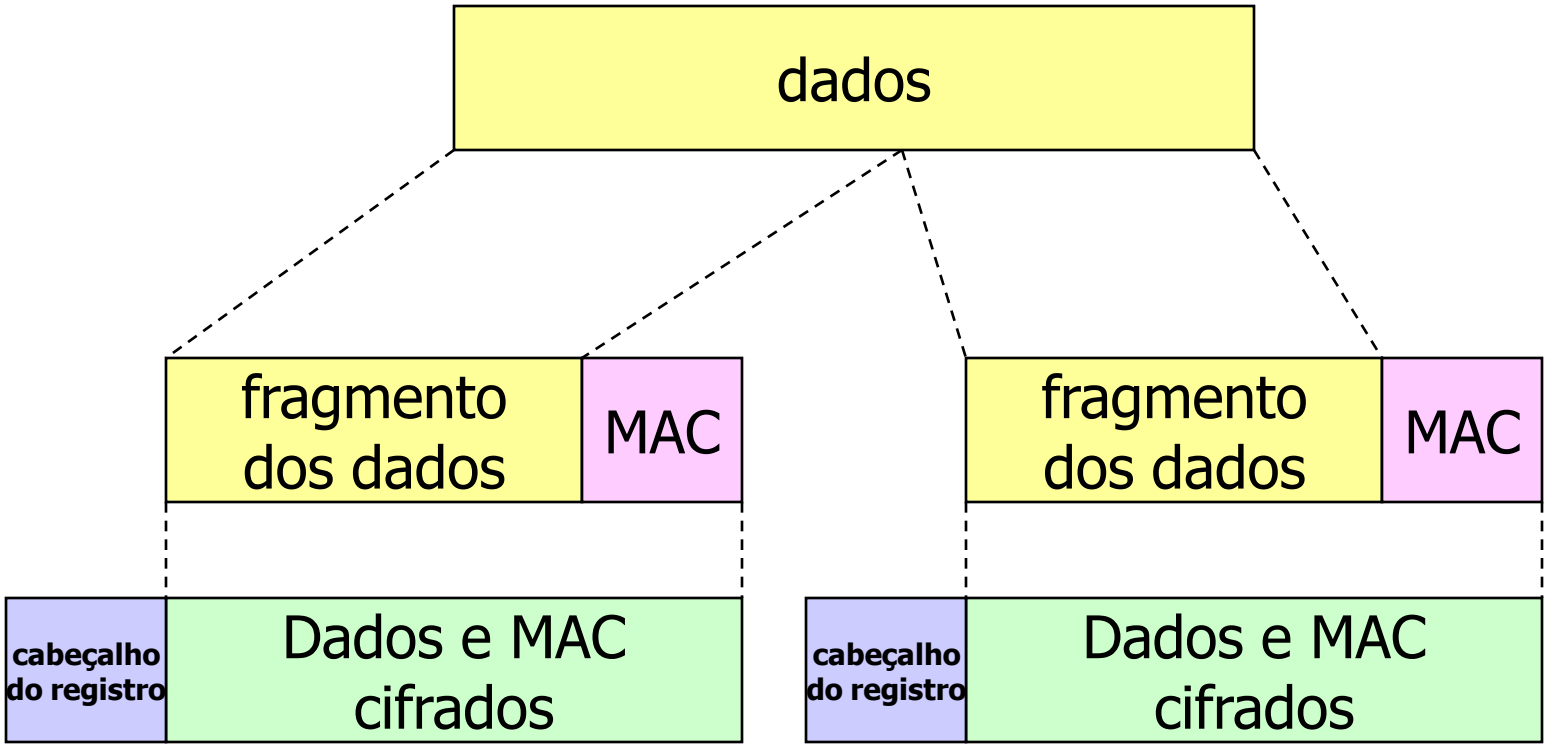
- **Solução**: duas últimas etapas para proteger a inicialização de ataques
  - Cliente tipicamente oferece uma lista de algoritmos: um mais "fortes", outros mais "fracos"
    - Últimas duas mensagens são cifradas

SSL Real: Proteção de Registros

Instituto de computação

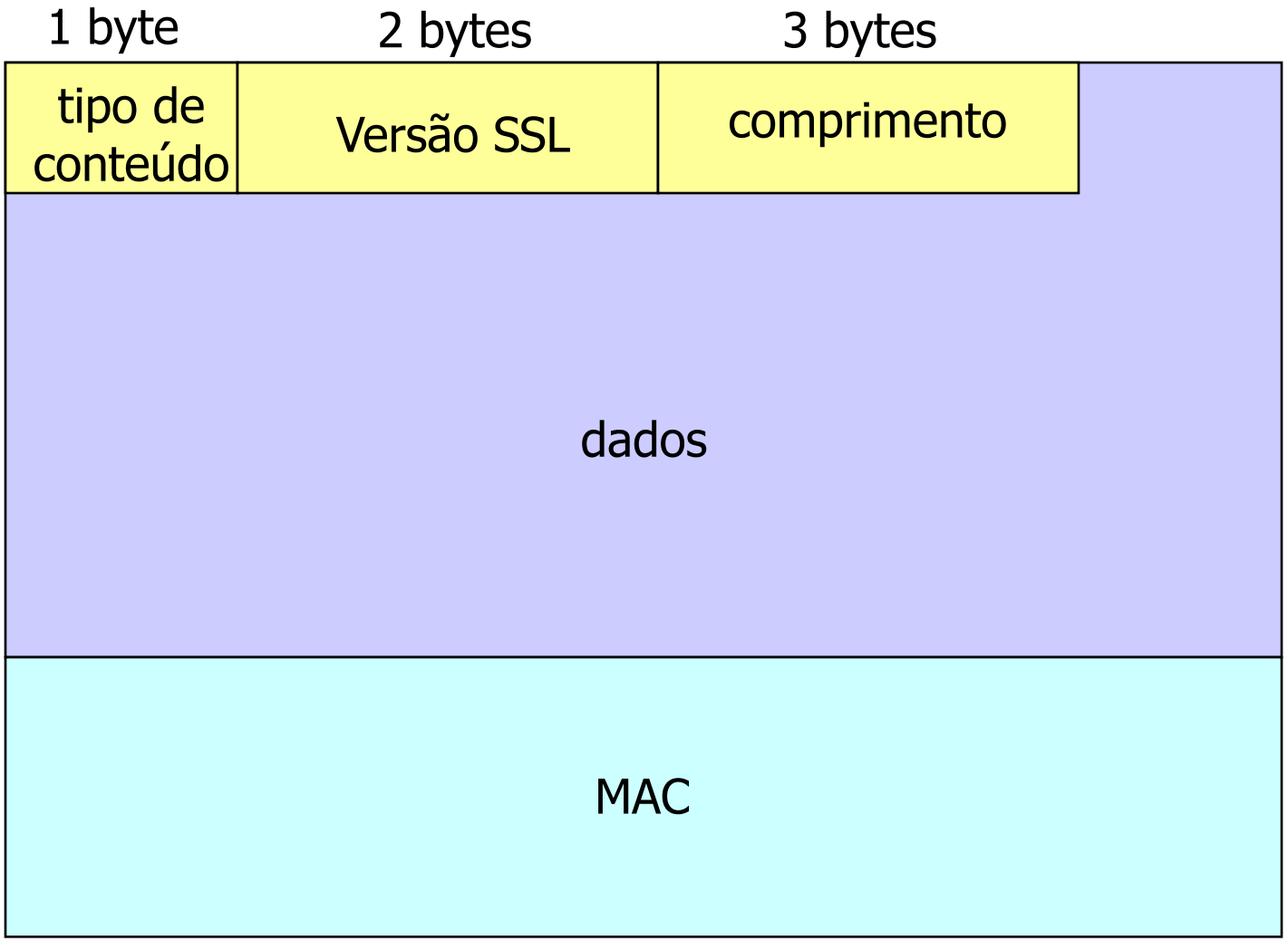
- Por que dois *nonces* aleatórios?
- Suposição: Trudy fareja todas as mensagens entre Alice e Bob
- No próximo dia, Trudy estabelece uma conexão com Alice e envia a mesma sequência de registros
  - Alice (Amazon) pensa que Bob fez separadamente dois pedidos do mesmo produto
  - Solução: Alice envia diferentes *nonces* aleatórios para cada conexão
    - Assim, chaves de cifração serão diferentes nos dois dias
  - Mensagens de Trudy não irão passar no teste de integridade de Alice

SSL Real: Protocolo de Registros



- Cabeçalho do registro: tipo do conteúdo, versão, comprimento
- MAC: inclui o número de sequência, chave MAC M_X
- Fragmento: cada fragmento SSL é de 2^{14} bytes (~ 16 KB)

SSL Real: Formato dos Registros



Dados e MAC cifrados (algoritmo de cripto. Simétrica)

Conexão Real

handshake: ClientHello

handshake: ServerHello

handshake: Certificate

handshake: ServerHelloDone

handshake: ClientKeyExchange

ChangeCipherSpec

handshake: Finished

ChangeCipherSpec

handshake: Finished

application_data

application_data

Alert: warning, close_notify

Deste ponto em diante tudo está cifrado

TCP FIN a seguir

SSL Real: Derivação das Chaves

Instituto de computação

- *Nonces* do cliente e do servidor e o segredo pré-mestre são entradas de um gerador de números pseudo-aleatório
  - Gerar o segredo mestre
- Segredo mestre e novos *nonces* são entradas de um outro gerador de números pseudo-aleatório: bloco de chaves
- Bloco de chaves é "fatiado e picado"
  - Chaves MAC do cliente e do servidor
  - Chaves para cifrar do cliente e do servidor
  - Vetores de inicialização (VI) do cliente e do servidor

Instituto de computação

Camada de Rede

Confidencialidade na Camada de Rede

Instituto de computação

- Confidencialidade entre duas entidades de rede

- A entidade transmissora **cifra a carga útil** dos datagramas
  - Carga útil pode ser um segmento TCP, um segmento UDP, uma mensagem ICMP, uma mensagem OSPF etc.
- Todos os dados enviados de uma entidade para outra estão **escondidos**
  - Páginas Web, e-mail, transferência de arquivos P2P, pacotes TCP SYN etc.

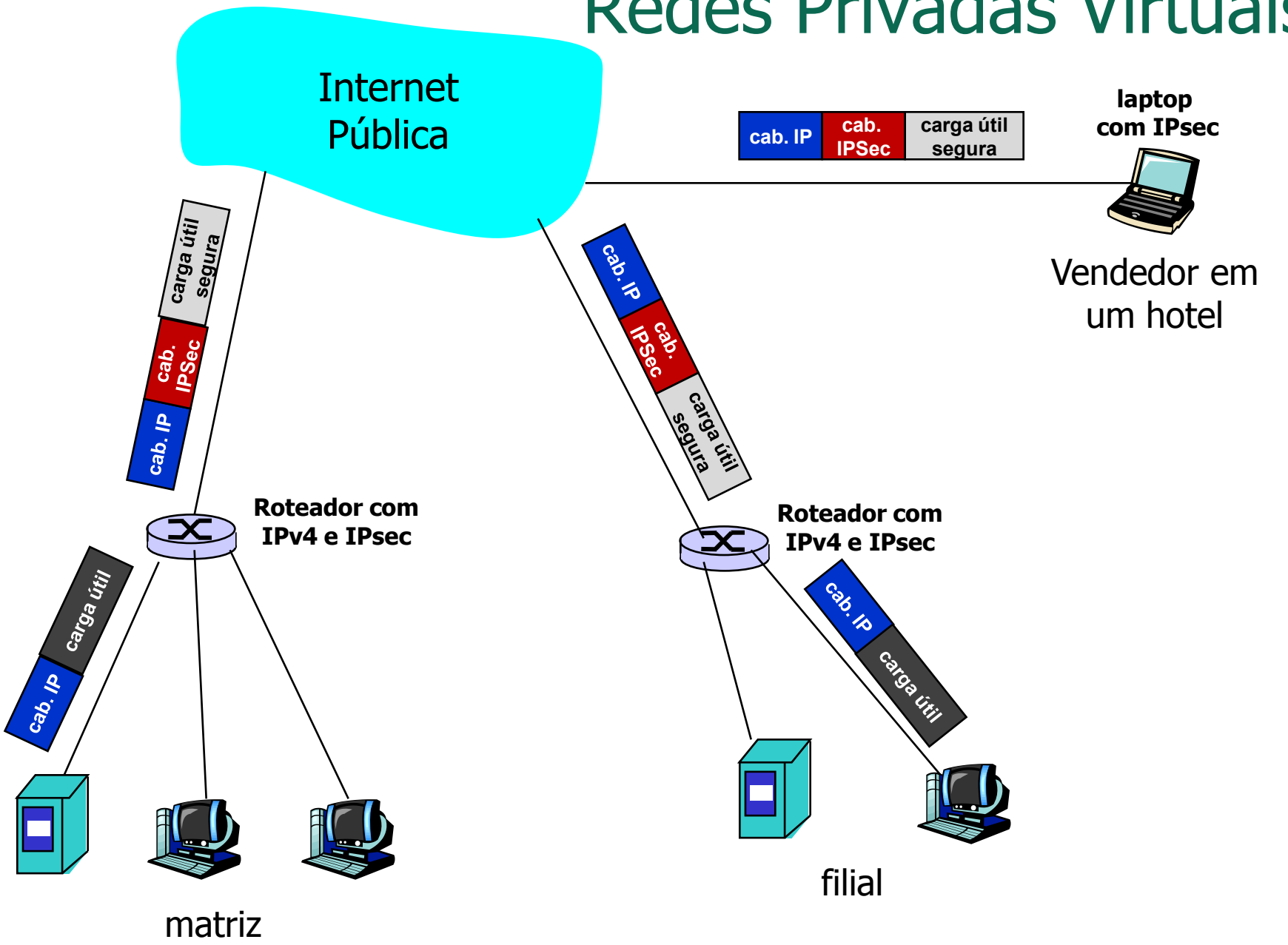
Redes Privadas Virtuais

Instituto de computação

- Mais conhecidas como VPNs (*Virtual Private Networks*)

- Instituições querem redes privadas por segurança
  - **Alto custo** → roteadores separados, enlaces dedicados, infraestrutura de DNS etc.
- Com uma VPN, o tráfego entre os escritórios de uma instituição são enviados usando a Internet pública
  - Porém, o tráfego é cifrado antes de entrar na Internet pública

Redes Privadas Virtuais



Serviços IPsec

Instituto de computação

- Integridades dos dados
- Autenticação da origem
- Prevenção contra o ataque de repetição
- Confidencialidade

- Dois protocolos oferecem modelos de serviço diferentes
  - Protocolo de autenticação de cabeçalho (*Authentication Header* – AH)
  - Protocolo de encapsulamento seguro (*Encapsulation Security Protocol* – ESP)

IPsec: Dois Protocolos

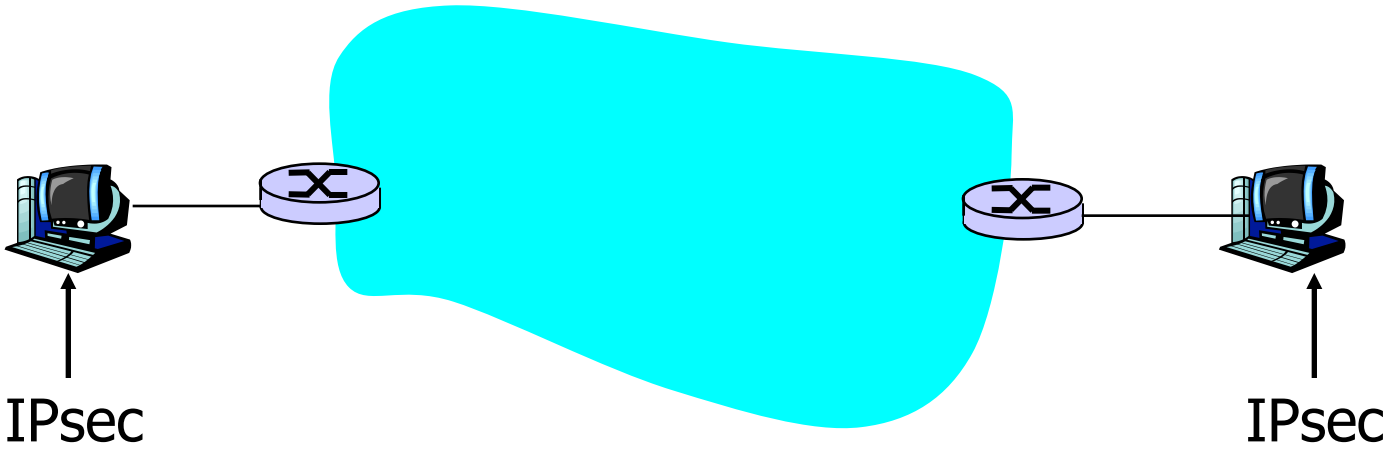
Instituto de computação

- Protocolo de autenticação de cabeçalho (*Authentication Header* – AH)
  - Provê **autenticação** da fonte e **integridade** dos dados
  - **NÃO** provê **confidencialidade**

- Protocolo de encapsulamento seguro (*Encapsulation Security Protocol* – ESP)
  - Provê **autenticação** da fonte, **integridade** dos dados e **confidencialidade**
  - Mais usado do que o AH

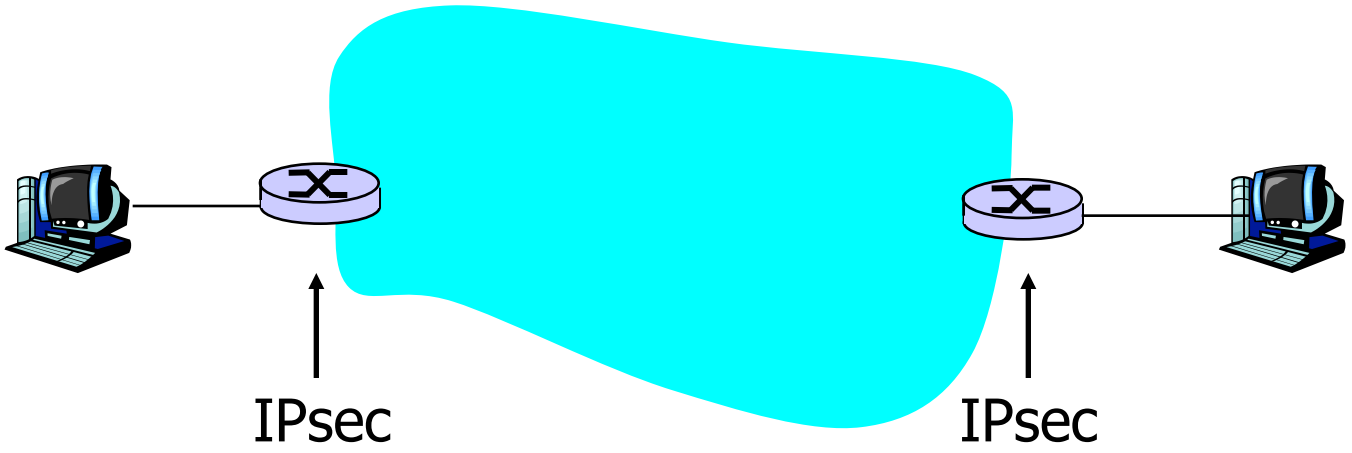
Modos de Transporte do IPsec

- Modo estação
  - Datagramas IPsec são transmitidos e recebidos pelos sistemas finais
  - Proteção para os protocolos das camadas superiores



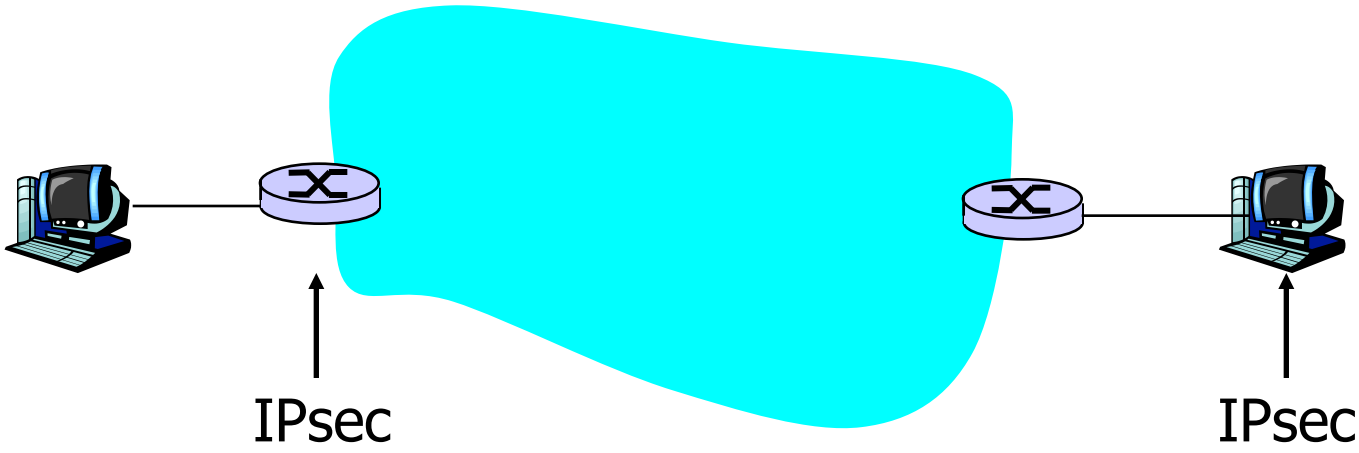
Modos de Transporte do IPsec

- Modo tunelamento
  - Roteadores de borda rodam IPsec
  - Estações não precisam



Modos de Transporte do IPsec

- Modo tunelamento
  - Exemplo abaixo: também é um modo de tunelamento



Possíveis Combinações

Modo estação com AH	Modo estação com ESP
Modo tunelamento com AH	

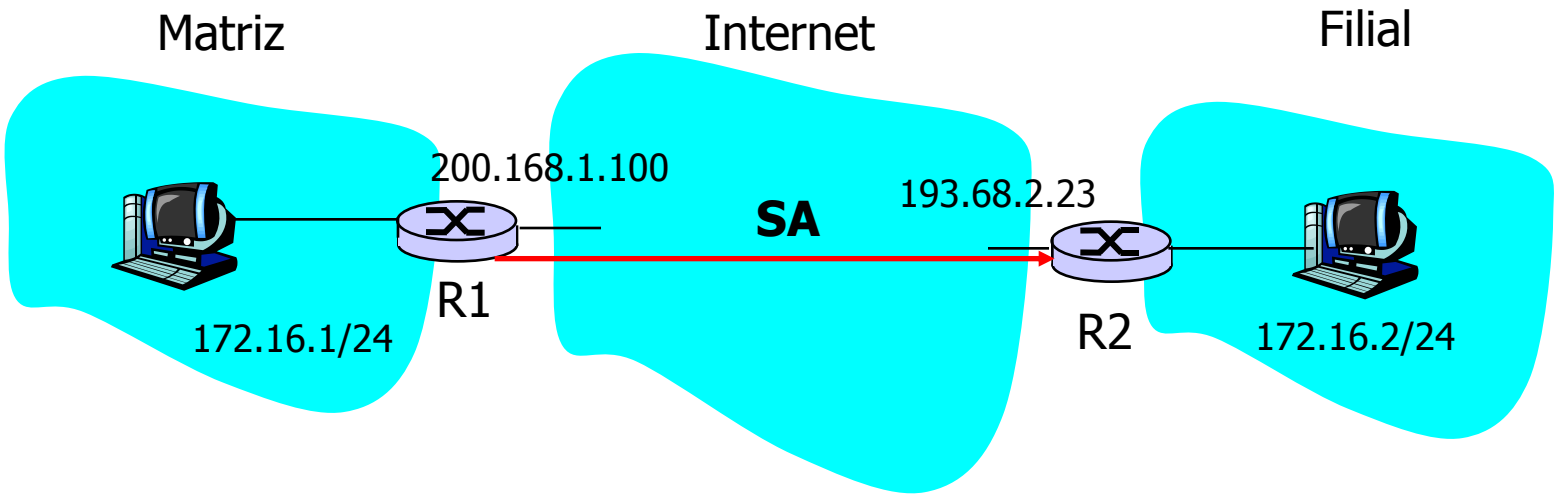
Mais comum e mais importante

Associações de Segurança (SAs)

Instituto de computação

- Antes de enviar os dados, uma **conexão virtual** é estabelecida entre as entidades transmissora e receptora
  - Chamada de **associação de segurança** (*Security Association* - SA)
    - São **unidirecionais**
- Entidades transmissora e receptora mantêm informações de estado sobre a SA
  - Terminações TCP também mantêm informações de estado
  - IP é sem conexão → **IPsec é orientado à conexão**

Exemplo: SA de R1 para R2



- R1 armazena para SA
  - Identificador de 32 bits da SA: *Security Parameter Index* (SPI)
  - Interface de origem da SA (200.168.1.100)
  - Interface de destino da SA (193.68.2.23)
  - Tipo de algoritmo para cifrar a ser usado (ex.: 3DES com CBC)
  - Chave para cifrar
  - Tipo de verificação de integridade (ex.: HMAC com MD5)
  - Chave de autenticação

Base de Dados de Associações de Segurança

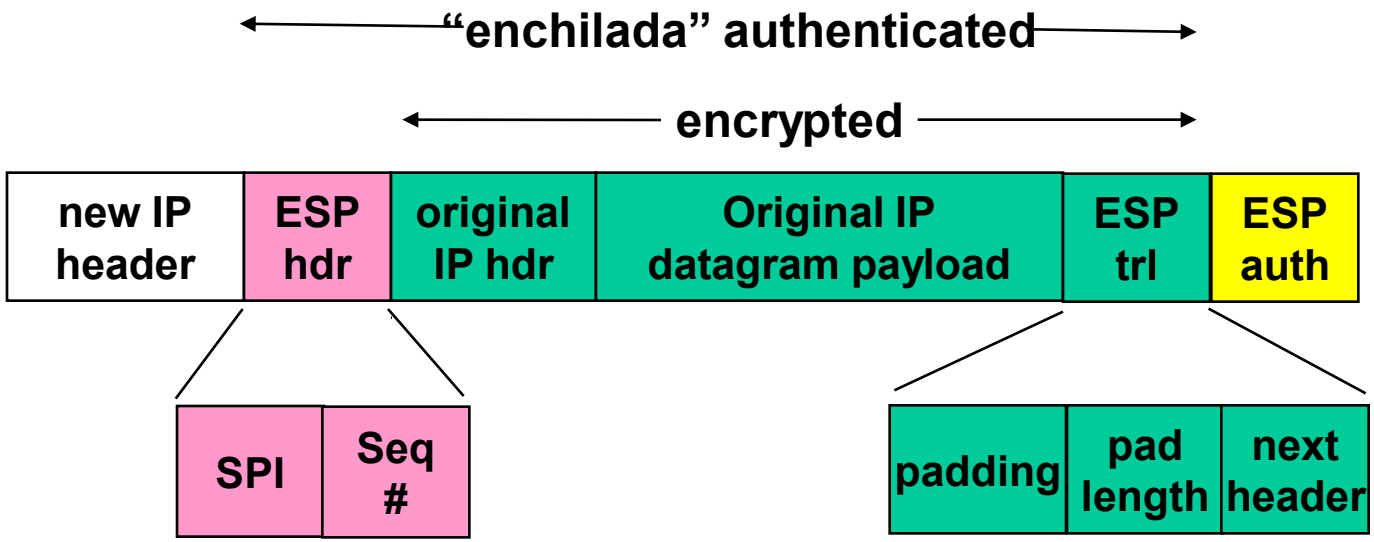
Instituto de computação

- *Security Association Database* (SAD)

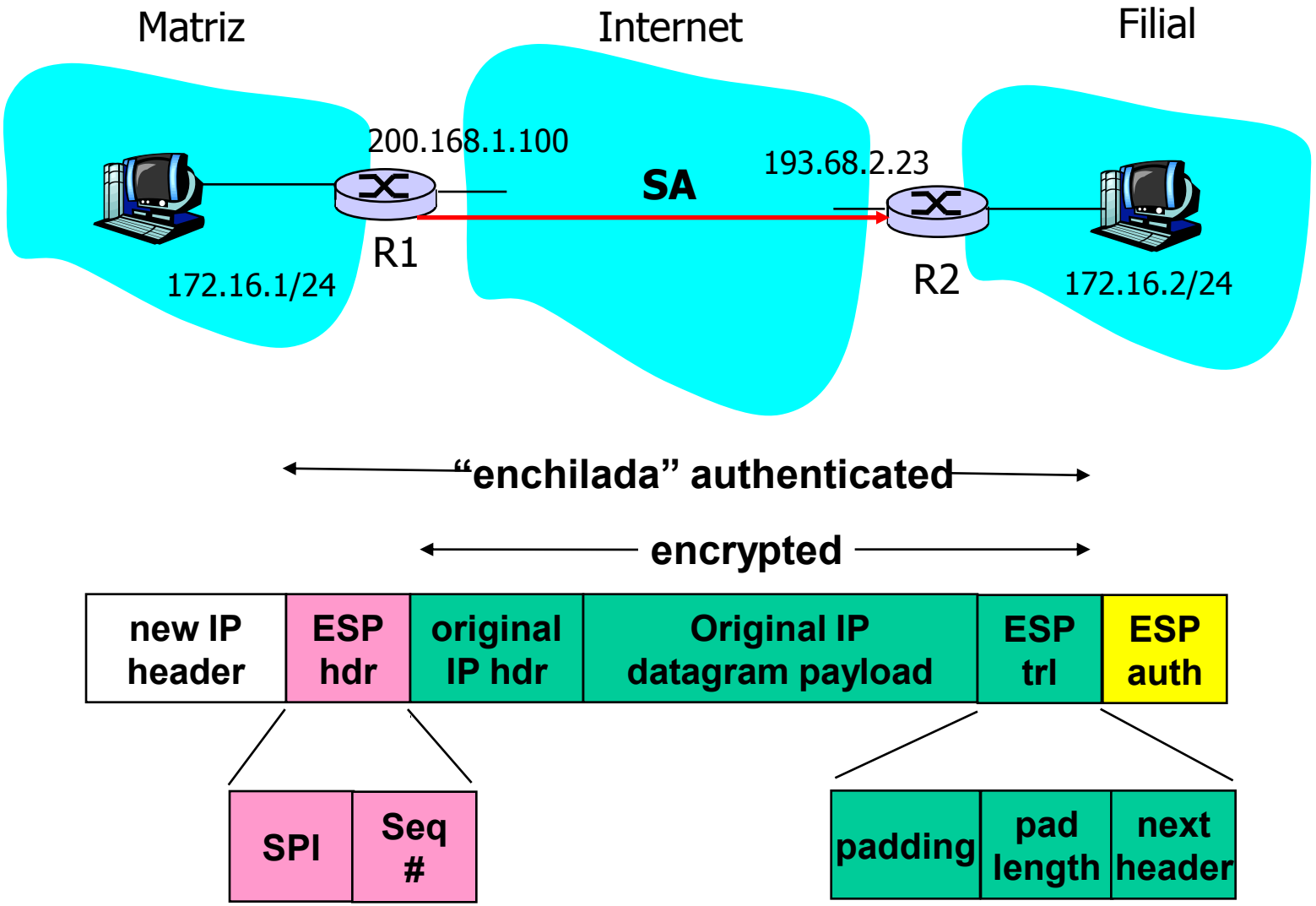
- Pontos finais armazenam estados de suas SAs na SAD
  - Local que verificam durante o processamento
- Com *n* vendedores, 2+2*n* SAs na SAD de R1
- Ao enviar um datagrama IPsec, R1 acessa sua SAD para determinar como processar o datagrama
- Ao receber um datagrama IPsec, R2 examina o SPI no datagrama, indexa a SAD com SPI e processa o datagrama como especificado

Datagrama IPSec

- Foco: modo túnel + ESM



O que acontece?



Conversão Datagrama Original em Datagrama IPsec em R1

Instituto de computação

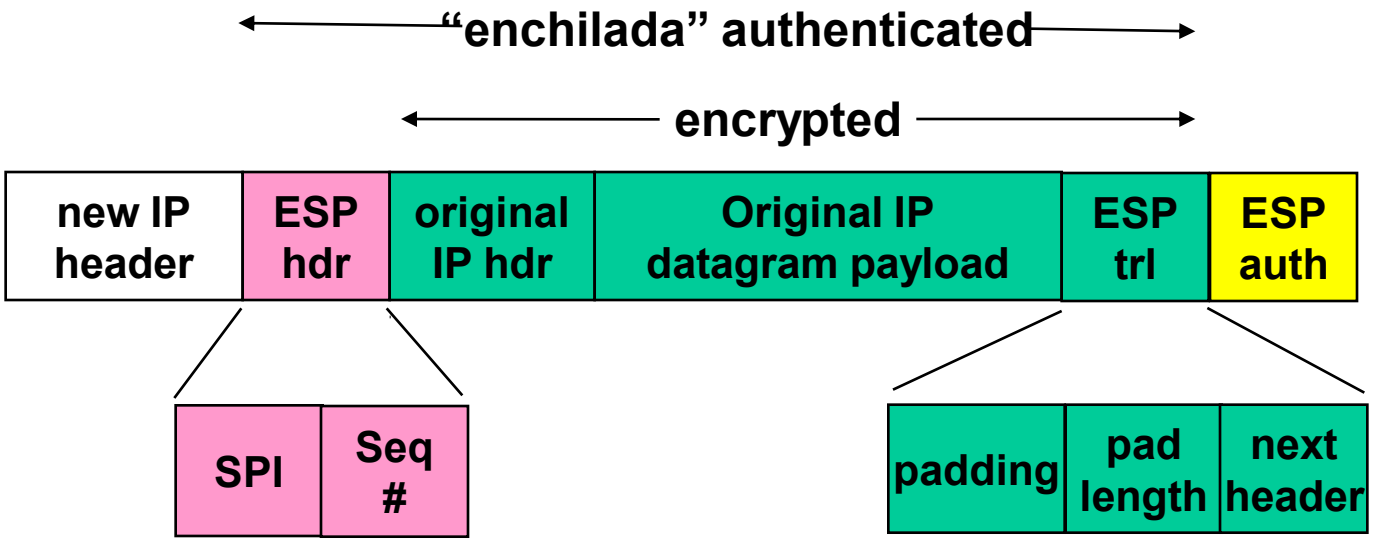
- Adicionar ao fim do datagrama original (incluindo os campos do cabeçalho original) um campo "ESP trailer"
- Cifrar o resultado usando o algoritmo e a chave especificados pela SA
- Adicionar ao início do resultado da cifragem o cabeçalho ESP → criar a "enchilada"
- Criar o MAC de autenticação para toda a enchilada usando o algoritmo e a chaves especificados pela SA

Conversão Datagrama Original em Datagrama IPsec em R1

Instituto de computação

- Adicionar o MAC ao fim da enchilada → formar a carga útil
- Criar um novo cabeçalho IP, com os campos tradicionais do cabeçalho IPv4, e adicioná-lo antes da carga útil

Campos da "Enchilada"



- ESP trailer: enchimento para cifradores de bloco
- Cabeçalho ESP
  - SPI: entidades receptoras sabem o que fazer ("id. da conexão")
  - Número de sequência para combater ataques de reprodução
- MAC no campo de autenticação do ESP é criado com uma chave secreta compartilhada

IPsec: Números de Sequência

Instituto de computação

- Emissor incializa o número de sequência com 0
  - Para um nova SA
- Para cada datagrama enviado na SA, o emissor
  - Incrementa o contador do número de sequência
  - Preenche o campo do número de sequência do cabeçalho ESP com esse valor

IPsec: Números de Sequência

Instituto de computação

- Objetivo
  - Evitar que um atacante fareje e reproduza um pacote
    - Recepção de pacotes IP autenticados e duplicados pode interromper o serviço
- Método
  - Destino verifica se existem pacotes duplicados
  - Porém, **NÃO** guarda informações de TODOS os pacotes recebidos
    - Usa uma janela

Security Policy Database (SPD)

Instituto de computação

- Política: para um dado datagrama, a entidade transmissora precisa saber se ele deve usar IPsec
- Também precisa saber qual SA usar
  - Pode usar: endereços IP da fonte e do destino, número do protocolo
- Informações na **SPD** indicam **o que fazer** com um datagrama recebido
- Informações na **SAD** indicam **como fazer** isso

Serviços IPsec: Resumo

Instituto de computação

- Assuma que há um intruso entre R1 e R2 e que ele não conheça as chaves usadas

- O intruso é capaz de
  - Ver o conteúdo do datagrama original?
  - Descobrir os endereços IP da fonte e dos destino, o protocolo de transporte e a porta usada pela aplicação?
  - Inverter bits sem ser detectado?
  - Se passar por R1 usando o endereço IP de R1?
  - Reproduzir um datagrama?

Serviços IPsec: Resumo

Instituto de computação

- Assuma que há um intruso entre R1 e R2 e que ele não conheça as chaves usadas

- O intruso é capaz de
  - Ver o conteúdo do datagrama original?
  - Descobrir os endereços IP da fonte e dos destino, o protocolo de transporte e a porta usada pela aplicação?
  - Inverter bits sem ser detectado?
  - Se passar por R1 usando o endereço IP de R1?
  - Reproduzir um datagrama?

Não!

Internet Key Exchange

Instituto de computação

- Exemplos anteriores: SAs IPsec estabelecidas manualmente nos pontos finais
- Exemplo de SA
  - SPI: 12345
  - IP da fonte: 200.168.1.100
  - IP do destino: 193.68.2.23
  - Protocolo: ESP
  - Algoritmo para cifrar: 3DES-cbc
  - Algoritmo HMAC: MD5
  - Chave para cifrar: 0x7aeaca…
  - Chave HMAC:0xc0291f…

Internet Key Exchange

Instituto de computação

- Exemplos anteriores: SAs IPsec estabelecidas manualmente nos pontos finais
- Exemplo de SA
  - SPI: 12345
  - I
  - I
  - Protocolo: ESP
  - Algoritmo para cifrar: 3DES-cbc
  - Algoritmo HMAC: MD5
  - Chave para cifrar: 0x7aeaca…
  - Chave HMAC:0xc0291f…

Configuração manual se torna impraticável para VPNs com centenas de usuários

Internet Key Exchange

- Exemplos anteriores: SAs IPsec estabelecidas manualmente nos pontos finais
- Exemplo de SA
  - SPI: 12345
  - I
  - I
  - Protocolo: ESP
  - Algoritmo para cifrar: 3DES-cbc
  - Algoritmo HMAC: MD5
  - Chave para cifrar: 0x7aeaca…
  - Chave HMAC:0xc0291f…

Solução: usar IKE

IKE: PSK e PKI

Instituto de computação

- Autenticação pode ser feita de duas formas
  - Chave secreta previamente compartilhada (*pre-shared secret* – PSK)
  - Infraestrutura de chaves públicas (PKI)

- Com PSK, ambos os lados começam com um segredo
  - Executam a IKE para
    - Autenticarem um ao outro
    - Gerar as SAs IPsec
      - Uma em cada direção
      - Incluindo as chaves para cifrar e autenticar

IKE: PSK e PKI

Instituto de computação

- Com PKI, ambos os lados começam com o par chave pública/privada e o certificado
  - Executam a IKE para
    - Autenticarem um ao outro
    - Gerar as SAs IPsec
      - Uma em cada direção
  - Similar a inicialização no SSL

IKE: Fases

Instituto de computação

- IKE possui duas fases
  - Fase 1: Estabelecer uma SA IKE bidirecional
    - Atenção: SA IKE é diferente de uma SA IPsec
    - Também chamada de associação de segurança ISAKMP
  - Fase 2: ISAKMP é usada para negociar de forma segura o para de SAs IPsec

- Fase 1 possui dois modos: agressivo e principal
  - Modo agressivo usa menos mensagens
  - Modo principal provê proteção de identidade e é mais flexível

IPsec: Resumo

Instituto de computação

- Troca de mensagens com IKE para algoritmos, chaves secretas e números SPI
- Pode-se usar os protocolos AH ou o ESP (ou ambos)
- Protocolo AH provê integridade e autenticação da fonte
- Protocolo ESP provê o mesmo que o AH e mais confidencialidade
- Pares IPsec podem ser dois sistemas finais, dois roteadores/*firewalls* ou um roteador/*firewall* e um sistema final

Instituto de computação

Camada de Enlace

Segurança em Redes Sem-Fio

Instituto de computação

- É um grande desafio
  - Meio de transmissão compartilhado e em difusão
    - Basta estar no raio de alcance de uma estação para receber tudo que ela transmite
  - Em redes cabeadas um atacante tem maior dificuldade para obter um acesso ao meio físico
    - São consideradas mais seguras
- Rede sem-fio susceptível a diversos ataques
  - Escuta clandestina (espionagem) passiva das mensagens
  - Interferências ativas: criação, modificação e destruição das mensagens.

Segurança no IEEE 802.11

- *War-driving*
  - Dirigir em uma determinada área dotado de um dispositivo com interface de rede sem-fio IEEE 802.11 e um GPS
  - Varredura do espectro à medida que se desloca

Extraído de Vilela et al., 2007

Segurança no IEEE 802.11

Instituto de computação

- Dados
  - Baía de São Francisco, CA, EUA
  - Mais do que 9000 redes acessíveis em vias públicas
  - 85% não usam nenhuma criptografia/autenticação
  - Facilitam a bisbilhotagem de pacotes e outros ataques

Segurança no IEEE 802.11

65% com criptografia
35% sem criptografia

Extraído de Vilela et al., 2007

Segurança no IEEE 802.11

Instituto de computação

- Para tornar o IEEE 802.11 mais seguro
  - Autenticação e criptografia dos dados
  - Primeira tentativa de segurança 802.11: *Wired Equivalent Privacy* (WEP): fracasso
  - Tentativa atual: 802.11i

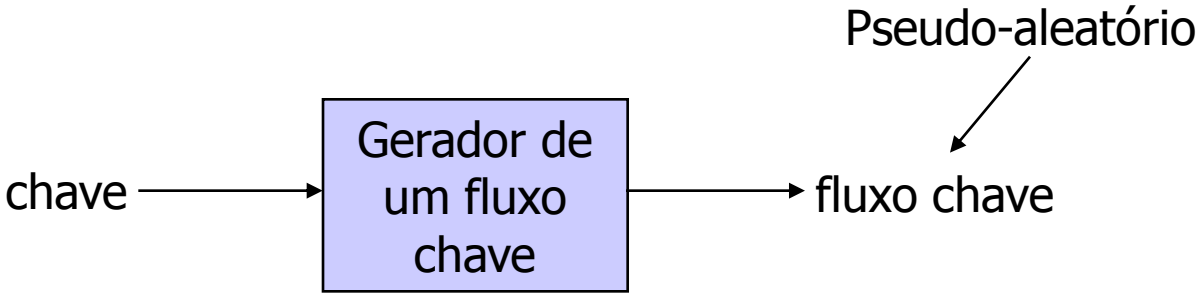
WEP: Objetivos de Projeto

Instituto de computação

- Usar criptografia de chaves simétricas
  - Confidencialidade
  - Autorização de estações
  - Integridade dos dados
- Ser autossincronizável
  - Cada pacote é cifrado separadamente
    - Dado um pacote cifrado e a chave, é possível decifrá-lo
    - Pode-se continuar a decifrar pacotes mesmo quando pacotes precedentes são perdidos
      - Diferente dos cifradores de bloco
- Ser eficiente
  - Pode ser implementado em hardware ou software

Cifradores de Fluxo Simétrico

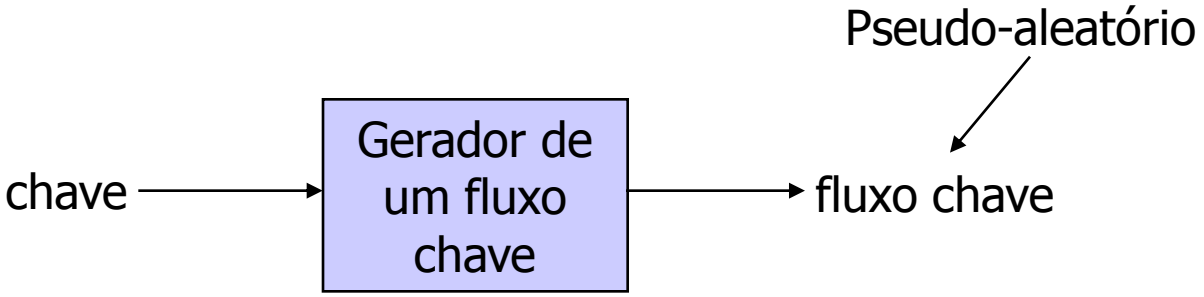
- Combinam cada bit de um fluxo chave (*keystream*) com um bit do texto plano para se obter um bit do texto cifrado
  - $m(i)$ = i-ésimo bit da mensagem em texto plano
  - $ks(i)$ = i-ésimo bit do fluxo chave
  - $c(i)$ = i-ésimo bit do texto cifrado
  - $c(i) = ks(i)$ XOR $m(i)$
  - $m(i) = ks(i)$ XOR $c(i)$



Cifradores de Fluxo Simétrico

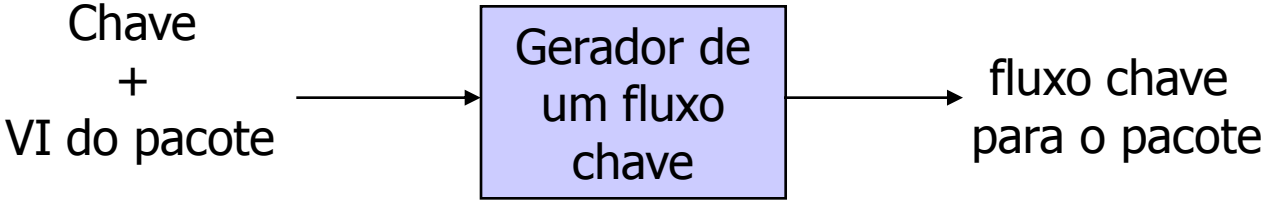
- Combinam cada bit de um fluxo chave (*keystream*) com um bit do texto plano para se obter um bit do texto cifrado
  - $m(i)$ = i-ésimo bit da mensagem em texto plano
  - $ks(i)$ = i-ésimo bit do fluxo chave
  - $c(i)$ = i-ésimo bit do texto cifrado
  - $c(i) =$
  - $m(i) =$

WEP usa o RC-4



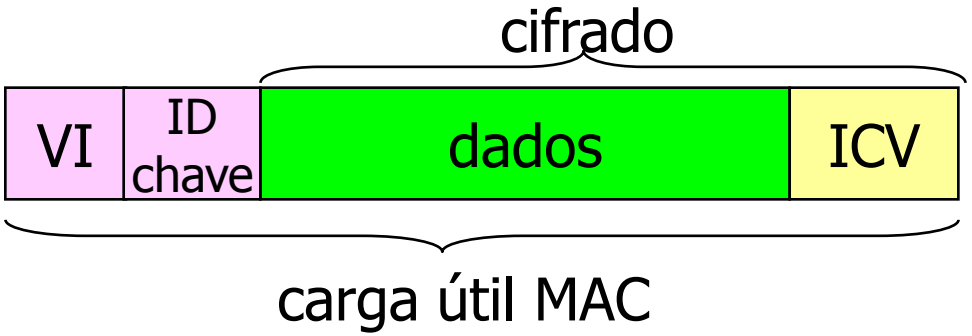
Cifradores de Fluxo e Independência dos Pacotes

- Cada pacote é cifrado **separadamente**
- Se para o quadro n+1, o fluxo chave for usado do ponto de parada para o quadro n → quadro não serão cifrados separadamente
- É preciso saber onde terminou para o quadro n
- Abordagem WEP
  - Inicializar o fluxo chave com a chave + um novo vetor de incialização (VI) para cada pacote



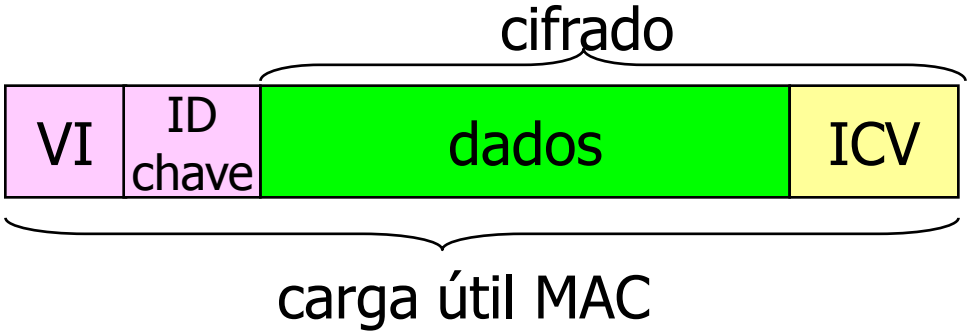
Cifração com WEP

- Emissor calcula o valor do teste de integridade (*Integrity Check Value* - ICV) para os dados
  - *Hash*/CRC de 4 bytes
- Emissor e receptor possuem um chave compartilhada de 104 bits
- Emissor cria um vetor de inicialização (VI) de 24 bits e o adiciona à chave
  - Resultado: chave de 128 bits



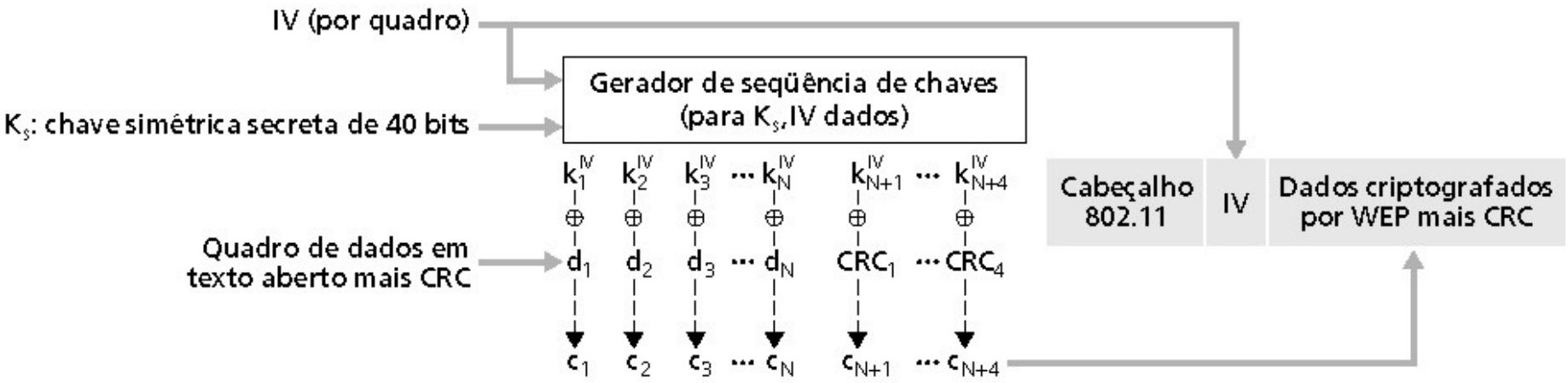
Cifração com WEP

- Emissor também adiciona o identificador da chave (8 bits)
- Chave de 128 bits é a entrada de um gerador de números pseudo aleatórios e a saída é o fluxo chave
- Dados do quadro + ICV são cifrados com RC4
  - XOR do fluxo chave com os dados + ICV
  - VI e id. da chave são adicionados aos dados cifrados para gerar a carga útil
  - Carga útil é "enquadrada" em um quadro 802.11



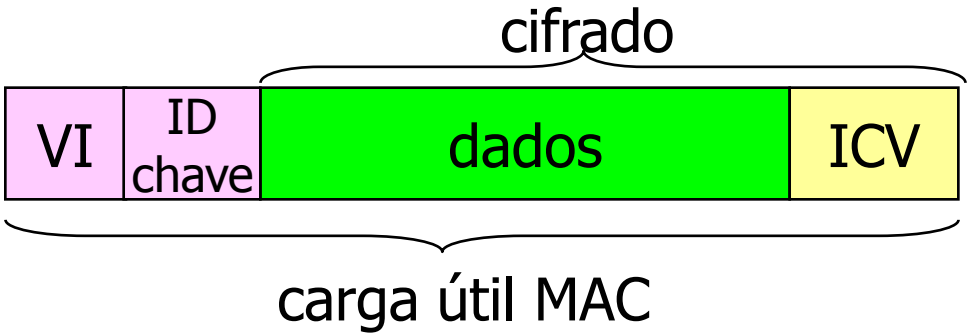
Cifração com WEP

- Lado emissor
- Um VI para cada quadro



Decifração com WEP

- Receptor extrai VI
- VI e a chave secreta compartilhada são a entrada de um gerador de números pseudo-aleatórios e a saída é o fluxo chave
- XOR do fluxo chave com os dados cifrados para decifrar dados + ICV
  - Teste de integridade é diferente do MAC e da assinatura digital



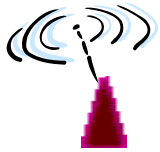
Autenticação com WEP

Instituto de computação

- Estação solicita autenticação do ponto de acesso
- Ponto de acesso envia *nonce* de 128 bits
- Estação codifica o *nonce* usando a chave simétrica compartilhada
- Ponto de acesso decifra o *nonce* e autentica a estação

- Não há mecanismo de distribuição de chaves
- Para se autenticar, basta conhecer a chave compartilhada

Autenticação com WEP



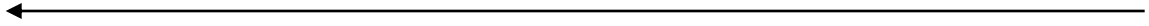
AP

pedido de autenticação

***nonce* (128 bytes)**

***nonce* cifrado com a chave secreta compartilhada**

Successo se o valor decifrado é igual ao *nonce*



Autenticação com WEP

Instituto de computação

- Nem todos os pontos de acesso exigem autenticação
  - Mesmo com o WEP habilitado
- Pontos de acesso indicam se a autenticação é necessária no quadro de *beacon*
- Autenticação é feita após a associação

Quebrando a Cifração WEP

- Falha de segurança
  - Vetor de inicialização (VI) de 24 bits
  - Um VI por quadro
  - VIs são enviados em texto plano

VIs podem ser **reusados**

Reuso pode ser **detectado**

Quebrando a Cifração WEP

Instituto de computação

- Ataque
  - Atacante faz com que Alice cifre textos planos conhecidos: d_1, d_2, d_3, ...
  - Atacante observa $c_i = d_i \text{ XOR } k_i^{\text{VI}}$
  - Atacante conhece c_i e d_i e pode computar k_i^{VI}
  - Atacante descobre a sequência de chaves para cifrar: k_1^{VI}, k_2^{VI}, k_3^{VI}, ...
  - Da próxima vez que o VI for usado, o atacante pode decifrar os dados enviados

Quebrando a Cifração WEP

Instituto de computação

- Ataque
  - RC4 → necessita que o mesmo valor de chave de 64 bits nunca seja usado mais de uma vez
  - WEP muda a chave quadro-a-quadro
  - 2^{24} chaves únicas
  - 99% de chance de usar a mesma chave depois de 12000 quadros
  - Se quadros tem 1 KB e enviados a 11 Mb/s
    - Repetição em poucos segundos

problema

- Aircrack
  - Injetar pacotes para acelerar a repetição do VI

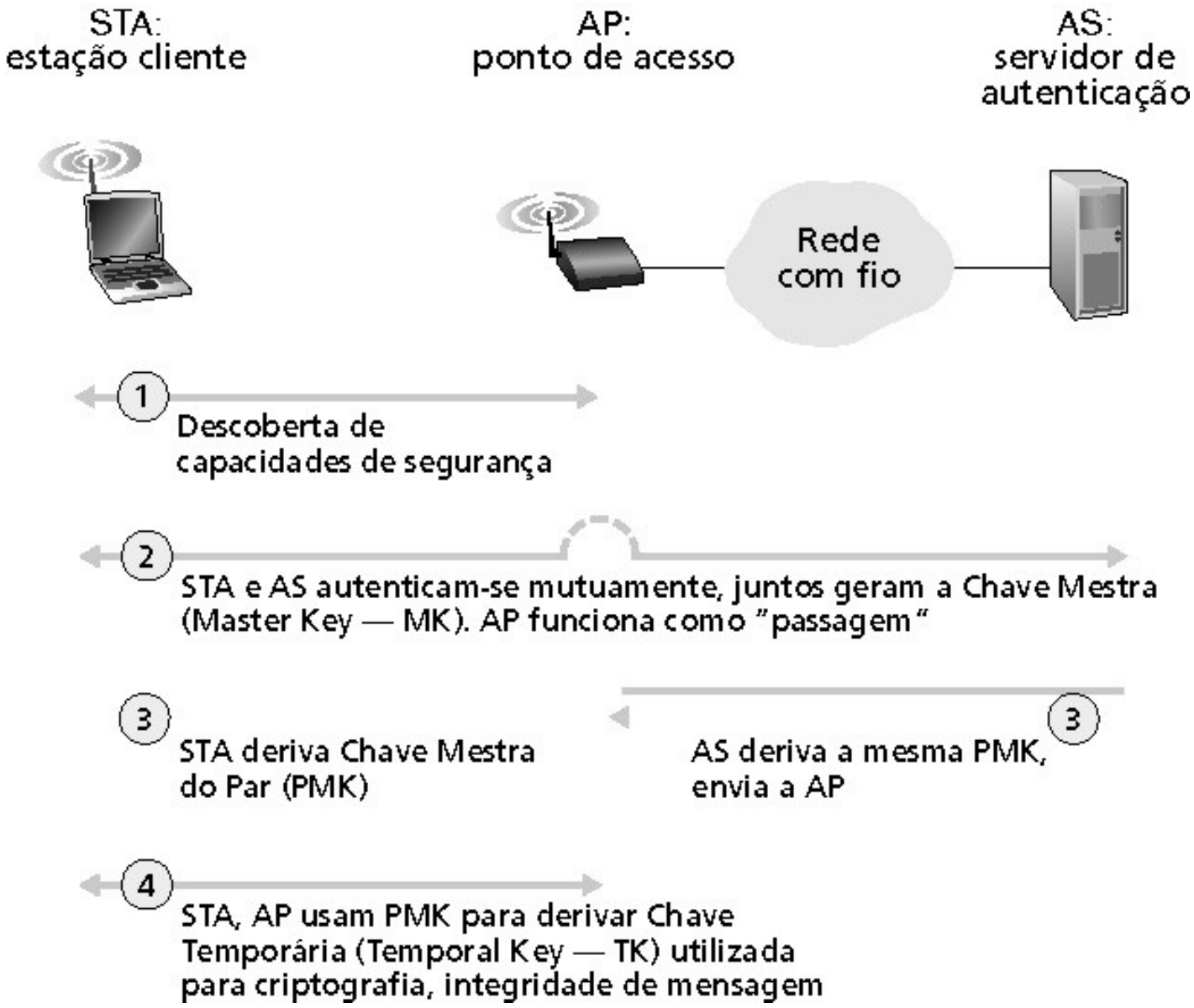
IEEE 802.11i

Instituto de computação

- Objetivo
  - Aumentar a segurança das redes sem-fio IEEE 802.11

- Diferentes formas de cifração são possíveis
  - Mais "fortes" que a do WEP
- Provê distribuição de chaves
- Usa um servidor de autenticação separado do ponto de acesso

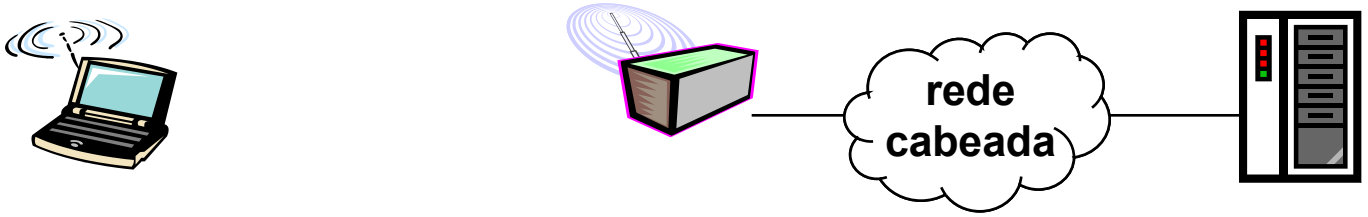
IEEE 802.11i: Quatro Fases

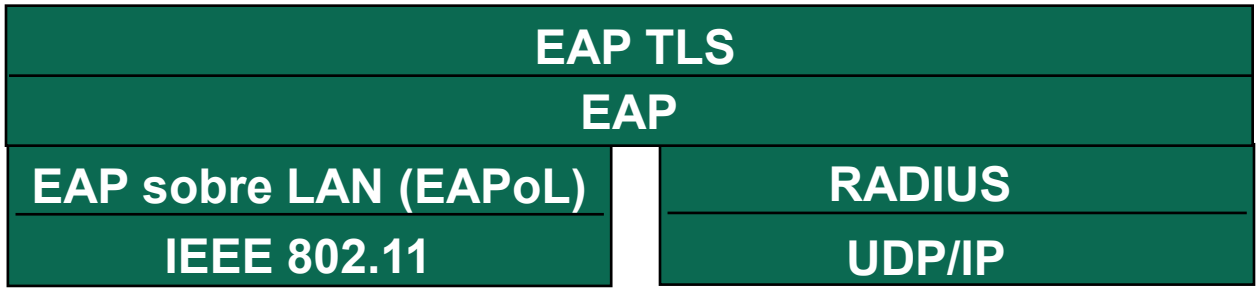


Extensible Authentication Protocol

Instituto de computação

- EAP: fim-a-fim (móvel) para protocolo do servidor de autenticação
- EAP enviado sobre "enlaces" separados
  - Móvel-para-AP (EAP sobre LAN)
  - AP para servidor de autenticação (RADIUS sobre UDP)





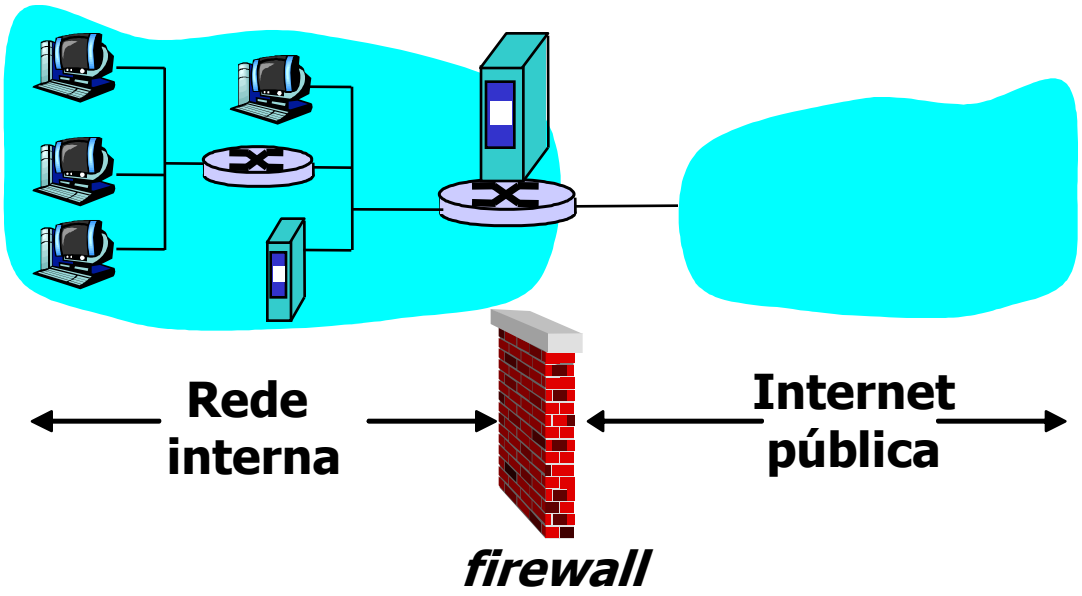
Instituto de computação

Segurança Operacional

Firewalls

Instituto de computação

- Objetivo: isolar a rede interna de uma organização da Internet
  - **Permitir** a passagem de alguns pacote e **bloquear** outros



Firewalls

Instituto de computação

- São necessários para
  - Prevenir ataques de negação de serviço
    - Inundação SYN: atacante estabelece várias conexões TCP "falsas" e esgota os recursos para conexões "reais"
  - Prevenir modificação/acesso não autorizado a dados da rede interna
    - Ex.: atacante troca a página do IC/UFF por outra qualquer
- Permite somente o acesso autorizado à rede interna
  - Conjunto de usuários e estações autenticados

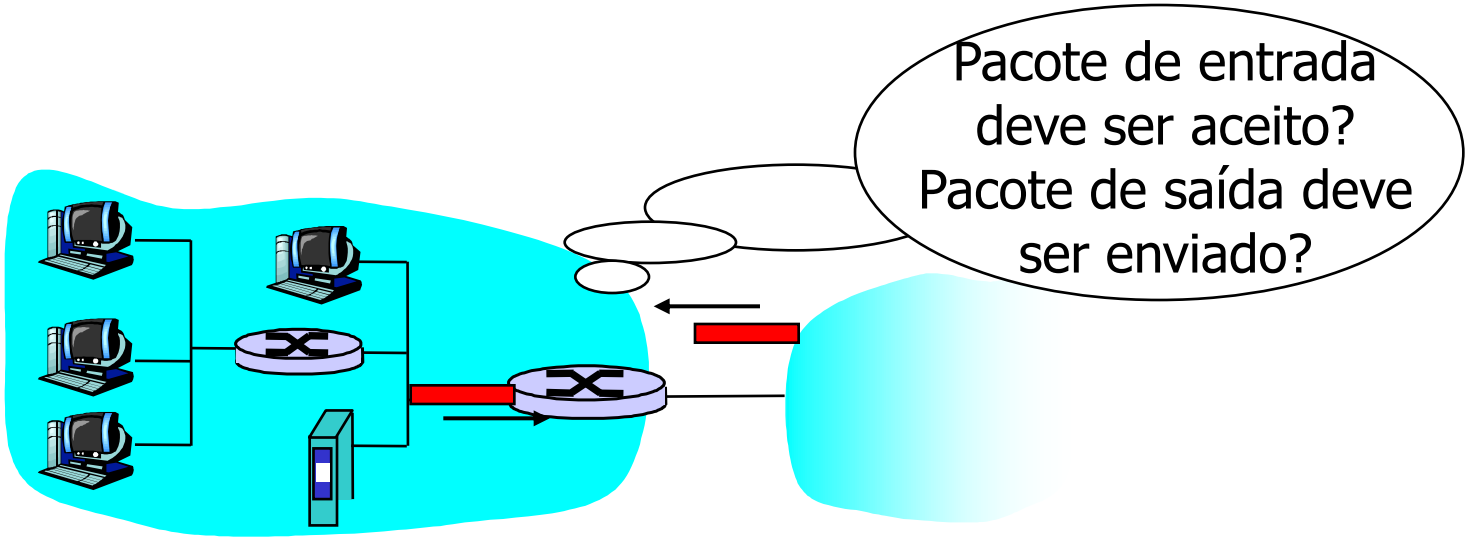
Firewalls

Instituto de computação

- Três tipos
  - Filtros de pacotes sem estado
  - Filtros de pacote com estados
  - *Gateways* de aplicação

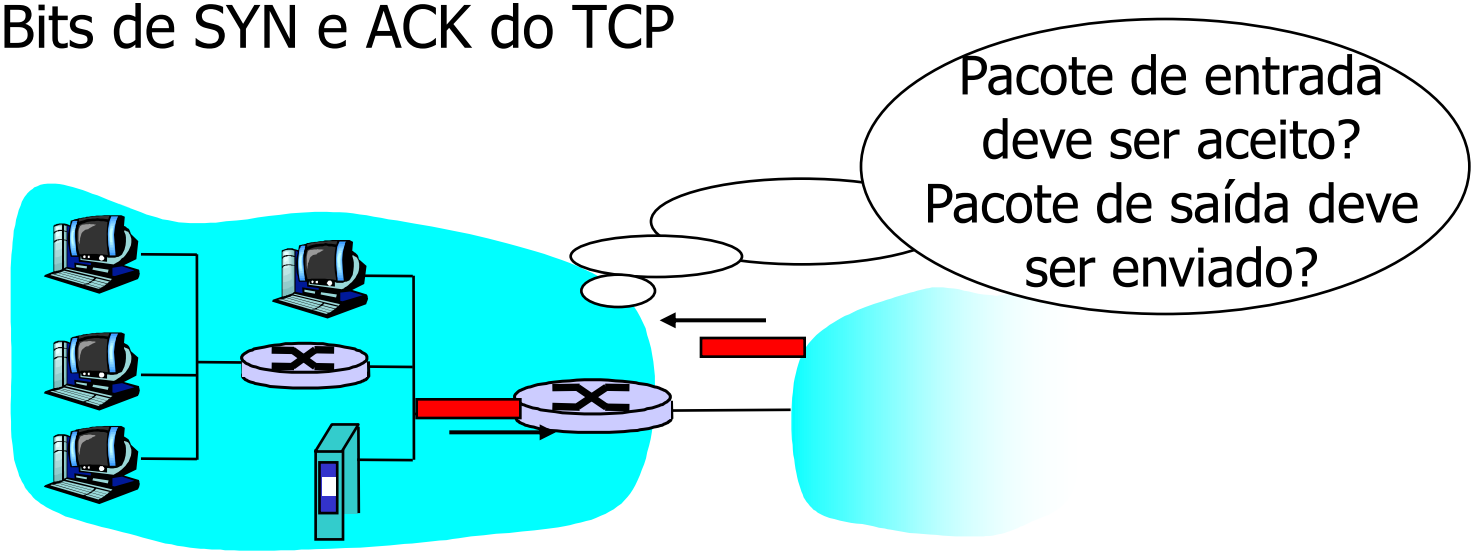
Filtragem de Pacotes Sem Estado

- Rede interna conectada à Internet através de um roteador com *firewall*



Filtragem de Pacotes Sem Estado

- Roteador filtra pacote-a-pacote,
- Decisão de encaminhar/descartar o pacote é baseada no(s)
  - Endereço IP da origem, endereço IP do destino
  - Número das portas de origem e destino do TCP/UDP
  - Tipo da mensagem ICMP
  - Bits de SYN e ACK do TCP



Filtragem de Pacotes Sem Estado

Instituto de computação

- Exemplo
  - Ação
    - Bloquear datagramas IP de entrada e saída com o campo protocolo = 17 e com a porta de origem ou destino = 23
  - Consequência
    - Todos os pacotes de fluxos UDP de entrada e saída e de conexões telnet são bloqueados

Filtragem de Pacotes Sem Estado

Instituto de computação

- Mais um exemplo
  - Ação
    - Bloquear segmentos TCP de entrada com ACK=0
  - Consequência
    - Previne que clientes externos abram conexões TCP com clientes internos
    - Mas clientes internos podem se conectar a clientes externos

Mais Exemplos

Instituto de computação

Política	Definição do *firewall*
Sem acesso Web externo	Descartar todos os pacotes de saída para qualquer endereço IP, porta 80
Sem conexões TCP de entrada, exceto somente para o servidor Web público	Descartar todos os segmentos TCP SYN para qualquer endereço IP, exceto 200.20.15.38, porta 80
Evitar que rádios Web ocupem a banda passante disponível	Descartar todos os segmentos UDP de entrada, exceto DNS e *broadcast* de roteadores
Evitar que a rede interna seja usada como parte de um ataque de negação de serviço (DoS) do tipo Smurf	Descartar todos os pacotes ICMP destinados a um endereço de *broadcast*
Evitar que a rede interna seja "rastreada" com traceroute	Descartar pacotes ICMP de saída que informam que o TTL expirou

Listas de Controle de Acesso (*Access Control Lists* - ACLs)

Instituto de computação

- É uma tabela de regras aplicada de cima para baixo aos pacotes de entrada: pares (ação, condição)

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

Instituto de computação

- Filtragem de pacotes sem estado → "mão pesada"
  - Admite pacotes que "não fazem sentido"
    - Ex.: porta de destino = 80, bit de ACK marcado, mesmo que não existam conexões TCP estabelecidas

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

Filtragem de Pacotes Com Estado

Instituto de computação

- Verifica o estado de cada conexão TCP
- Armazena estado com base na abertura (SYN) e encerramento (FIN) de um conexão
  - É possível determinar quando pacotes de entrada e saída "fazem sentido"
- Se o temporizador estoura → conexões inativas
  - *Firewall* não admite mais pacotes dessa conexão

Filtragem de Pacotes Com Estado

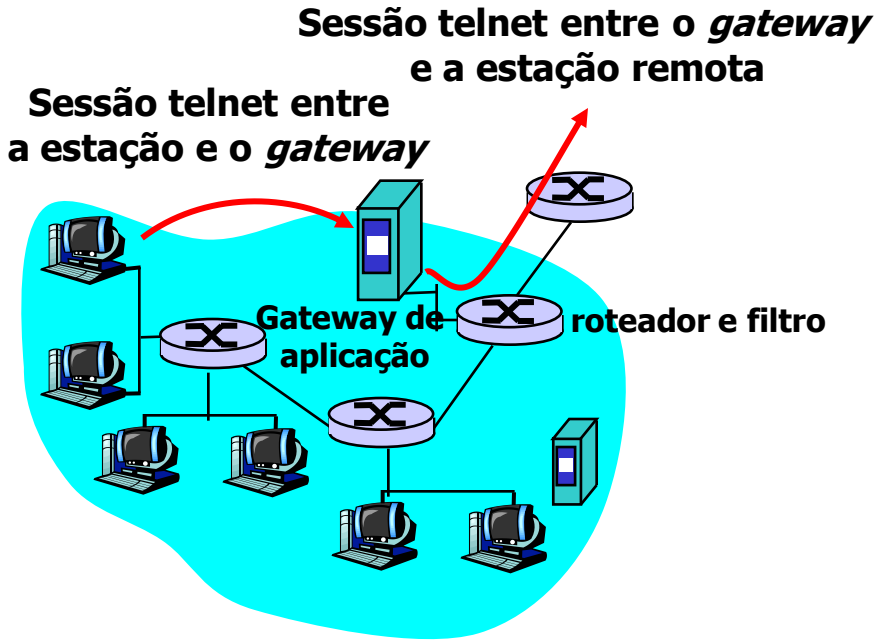
Instituto de computação

- ACL incrementada para indicar que é necessário verificar o estado da conexão na tabela antes de admitir o pacote

ação	endereço de origem	endereço de destino	proto	porta de origem	porta de destino	*bit* de flag	verificar conexão
permitir	222.22/16	fora de 222.22/16	TCP	> 1023	80	qualquer	
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	

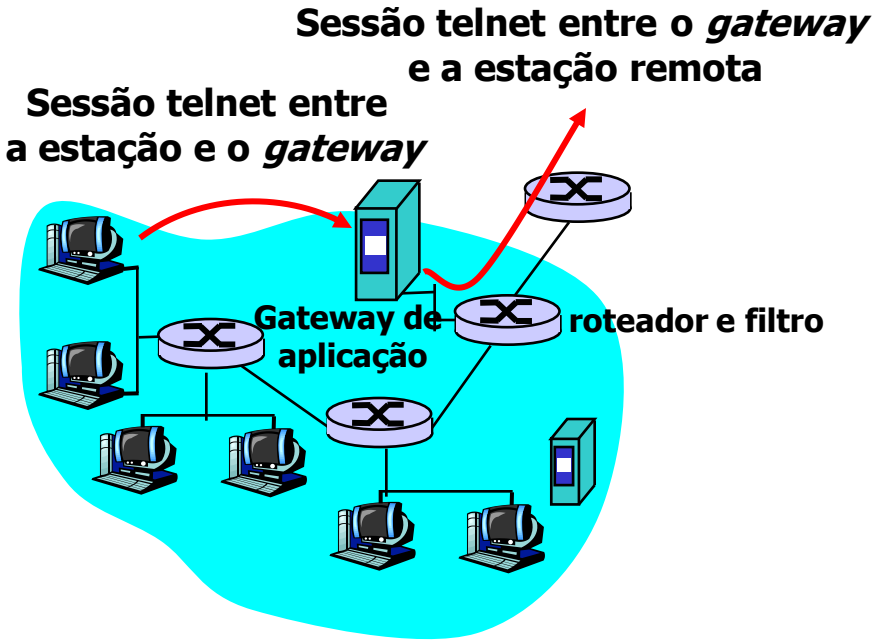
Gateways de Aplicação

- Filtrar os pacotes com base nos dados das aplicações assim como em campos IP/TCP/UDP
- Exemplo: permitir que determinados usuários da rede interna façam conexões telnet com redes externas



Gateways de Aplicação

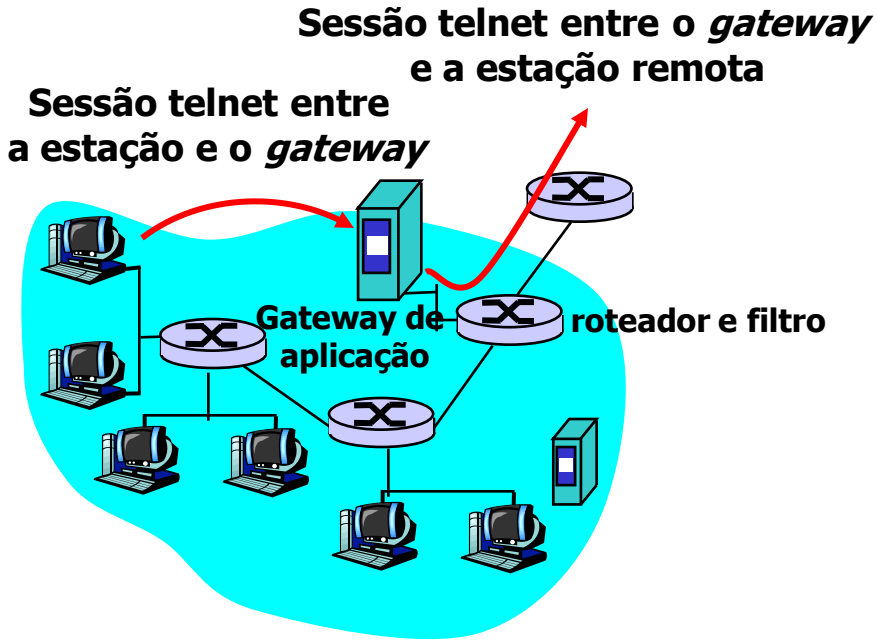
- Filtrar os pacotes com base nos dados das aplicações assim como em campos IP/TCP/UDP
- Exemplo: permitir que determinados usuários da rede interna façam conexões telnet com redes externas



1. Requer que todos os usuários telnet façam o telnet através do *gateway*

Gateways de Aplicação

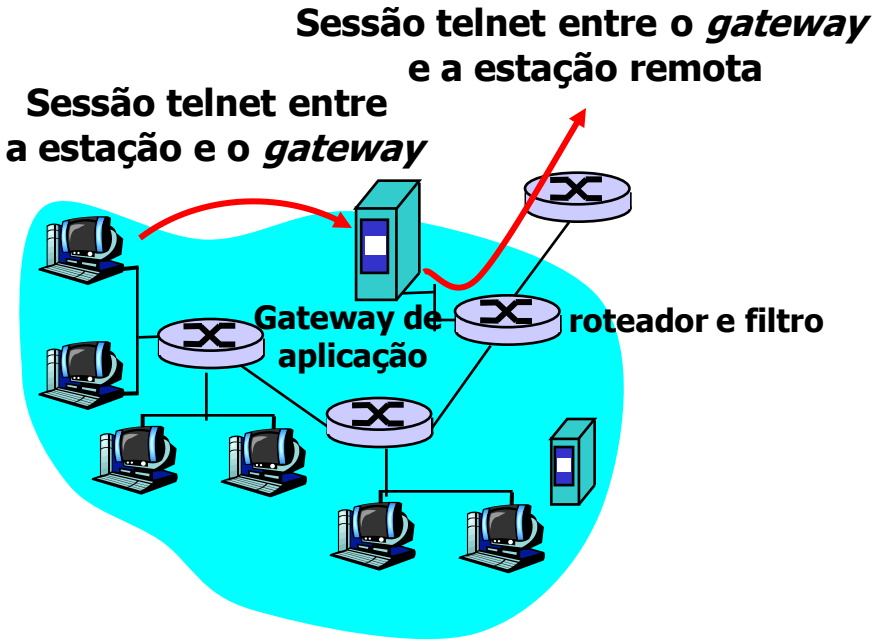
- Filtrar os pacotes com base nos dados das aplicações assim como em campos IP/TCP/UDP
- Exemplo: permitir que determinados usuários da rede interna façam conexões telnet com redes externas



2. Para os usuários autorizados, o *gateway* estabelece uma conexão telnet com a estação destino.
O *gateway* transfere os dados entre duas conexões

Gateways de Aplicação

- Filtrar os pacotes com base nos dados das aplicações assim como em campos IP/TCP/UDP
- Exemplo: permitir que determinados usuários da rede interna façam conexões telnet com redes externas



3. O filtro do roteador bloqueia todas as conexões que não têm origem no *gateway*

Firewalls

Instituto de computação

- IPTABLES

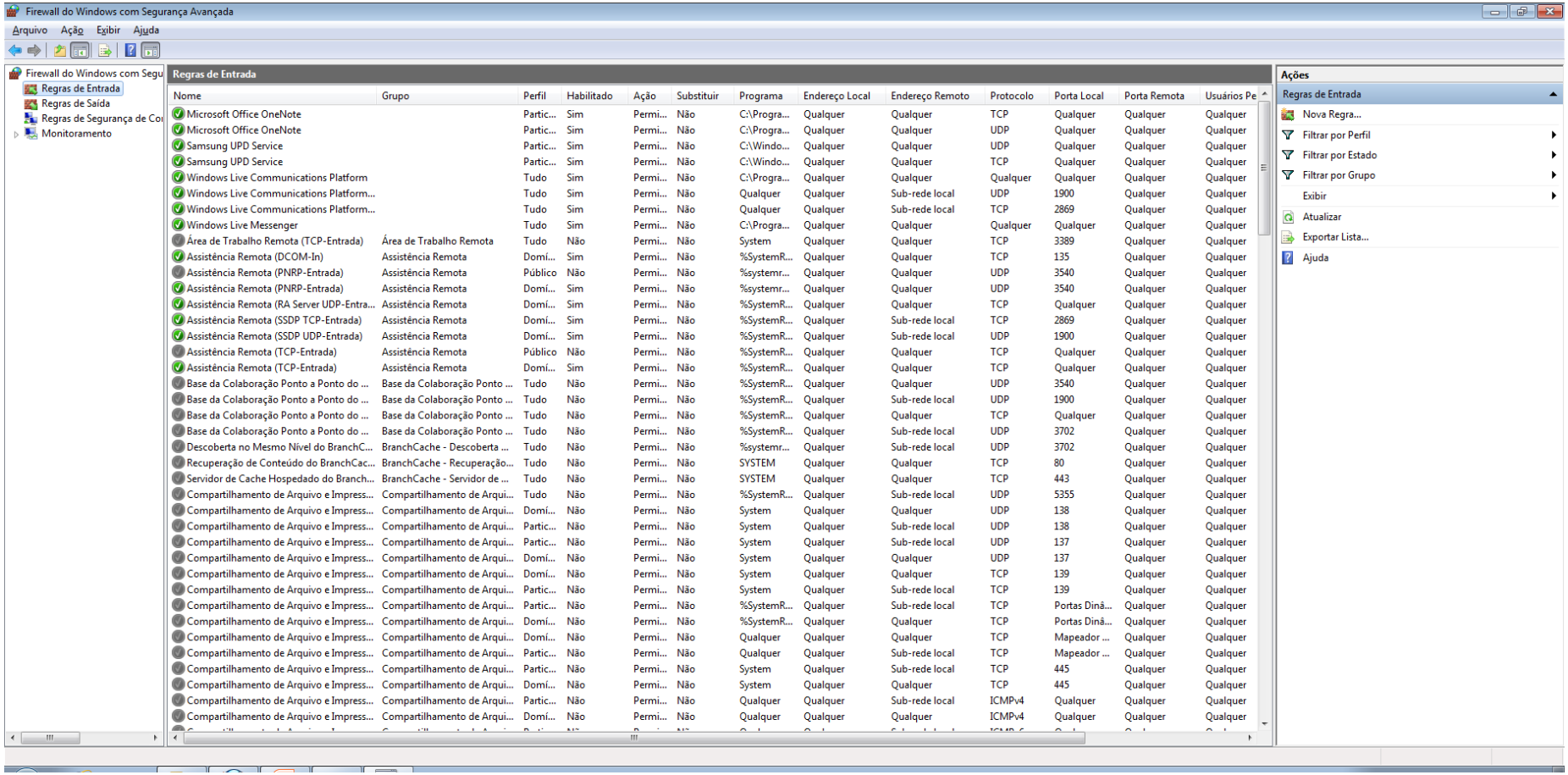
```
[root@redhat9 sysconfig]# /sbin/iptables --list
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABL
ISHED
REJECT     tcp  --  anywhere             anywhere            tcp option=!2 reject
-with tcp-reset
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
ACCEPT     udp  --  anywhere             anywhere            udp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:http
ACCEPT     udp  --  anywhere             anywhere            udp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
[root@redhat9 sysconfig]#
```

Extraído de http://xenocafe.com

Firewalls

Instituto de computação

- *Firewall* do Windows 7



Limitações de *Firewalls* e *Gateways*

Instituto de computação

- Endereços IP de origem podem ser forjados (IP *spoofing*)
  - Roteador não tem como saber se os dados "realmente" vêm da fonte alegada
- Se múltiplas aplicações necessitam tratamento especial, cada uma deve ter o próprio *gateway*
- Aplicativo do cliente deve saber como contatar o *gateway*
  - Ex.: Deve definir o endereço IP do *proxy* no navegador
- Para o UDP, os filtros normalmente usam uma política de "tudo ou nada"

Limitações de *Firewalls* e *Gateways*

Instituto de computação

- Compromisso
  - Grau de comunicação com o mundo externo x nível de segurança

- Muitos sítios altamente protegidos ainda sofrem ataques

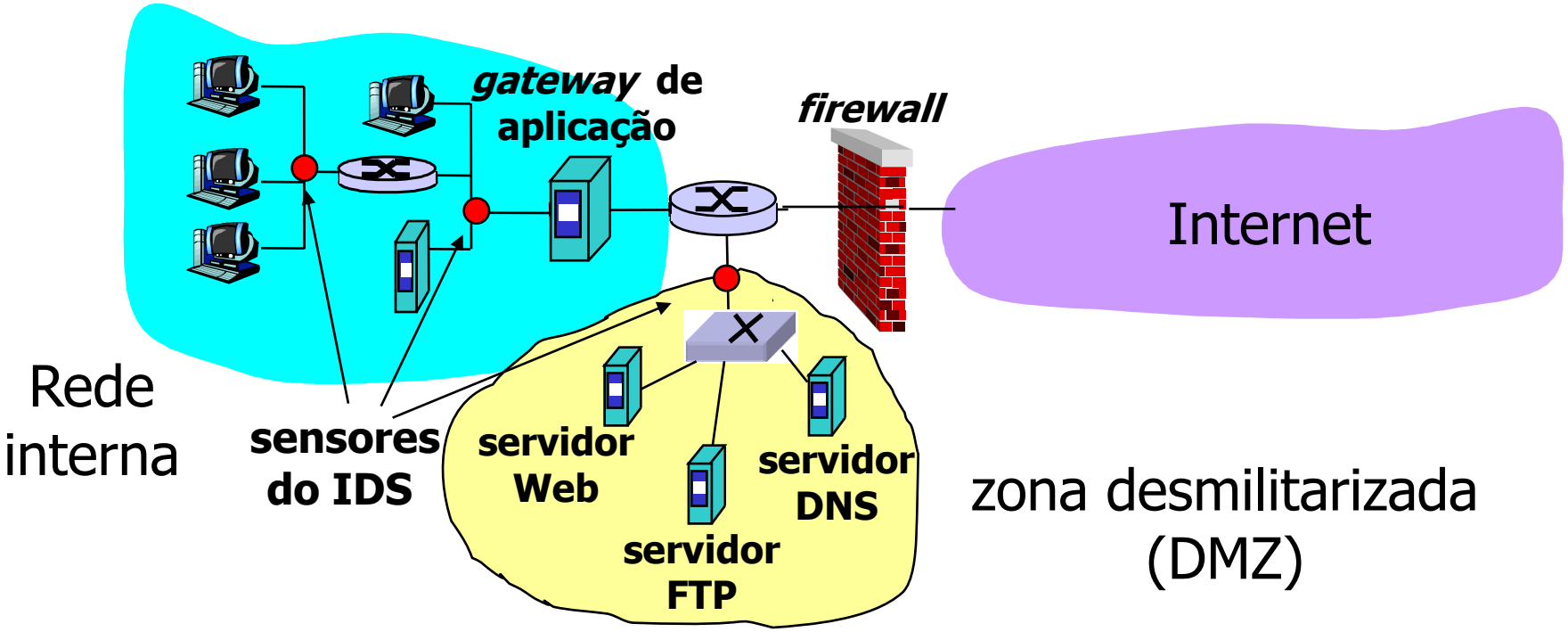
Sistemas de Detecção de Intrusão

Instituto de computação

- Filtragem de pacotes
  - Baseada apenas nos cabeçalhos TCP/IP
  - Não há testes de correlação entre as sessões
- Sistema de Detecção de Intrusão (*Intrusion Detection System* –IDS)
  - Verificação detalhada do **conteúdo dos pacotes**
    - Ex.: verificar seq. de caracteres em pacotes e comparar com bases de dados de vírus
  - Examinar a correlação entre múltiplos pacotes
    - Escaneamento de portas, mapeamento de rede, ataque de negação de serviço etc.

Sistemas de Detecção de Intrusão

- Múltiplos IDSes
  - Diferentes tipos de verificação em diferentes locais



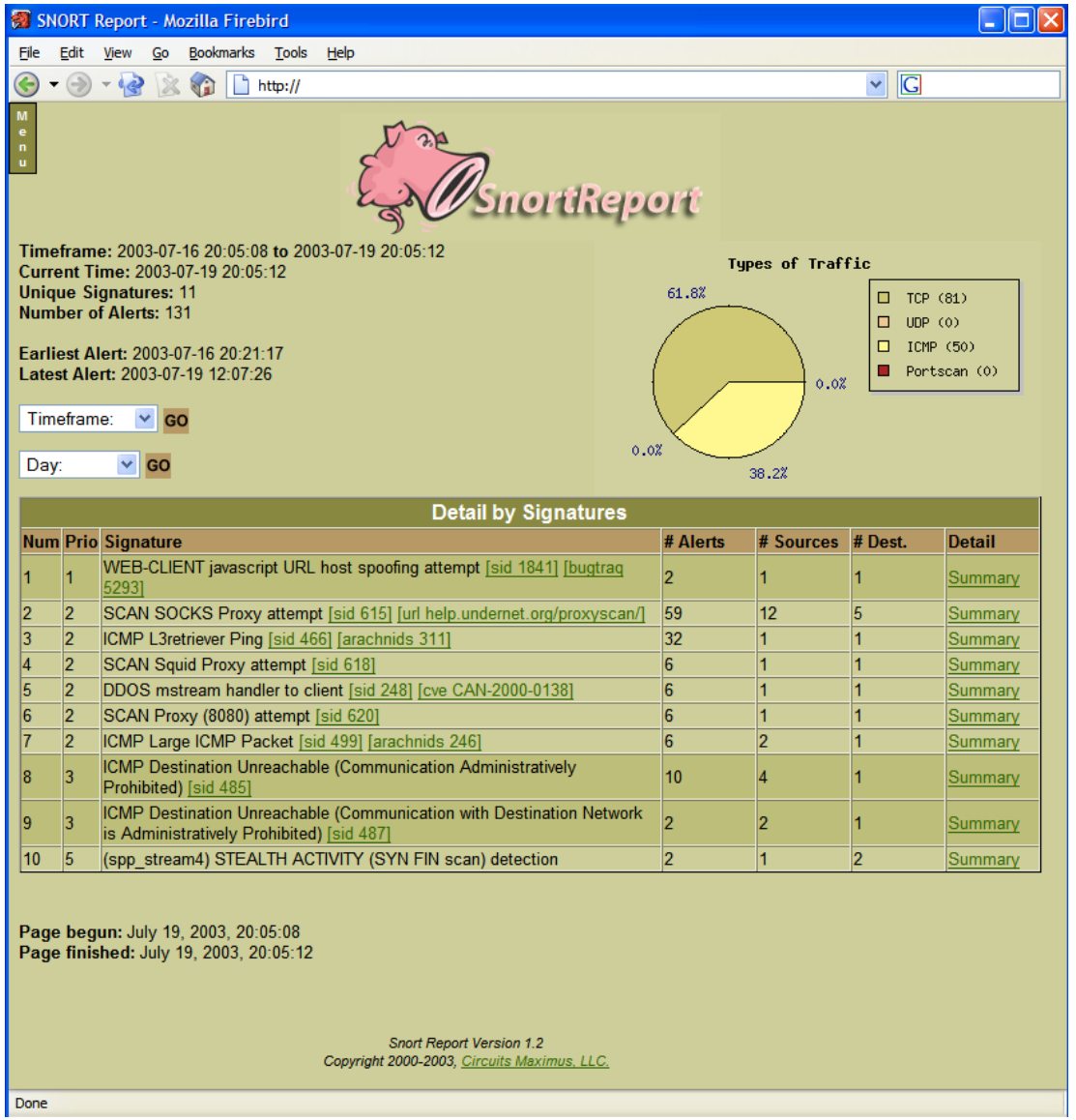
Sistemas de Detecção de Intrusão

Instituto de computação

- Dois tipos
  - Baseados em assinaturas
    - Mantém um banco de dados de assinaturas de ataques conhecidos
    - Compara cada pacote com as assinaturas conhecidas
      - Se um pacote é classificado → alerta para o administrador
    - Exigem conhecimento prévio do ataque
  - Baseados em anomalias
    - Cria um perfil de tráfego "normal"
    - Observa o tráfego para observar anomalias
    - Não exige conhecimento prévio do ataque
    - Porém, é difícil distinguir anomalias e tráfego normal

Exemplo de Sistema de Detecção de Intrusão

- Snort (**`snort.org`**)
  - Baseado em assinaturas
  - Base de dados atualizada por uma grande comunidade de usuários



Extraído de http://symmetrixtech.com

Instituto de computação

Departamento de Ciência da Computação
Instituto de Computação
Universidade Federal Fluminense

Aulas 15, 16, 17 e 18

Segurança de Redes

Conceitos, criptografia, segurança em diferentes camadas, *firewalls* e sistemas de detecção de intrusão

Igor Monteiro Moraes

Redes de Computadores II