Instituto de computação

Departamento de Ciência da Computação
Instituto de Computação
Universidade Federal Fluminense

Segurança de Redes

Conceitos, criptografia, segurança em diferentes camadas, *firewalls*

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 em redes de computadores
  - Confidencialidade
  - Integridade
  - Autenticação
  - Não-repúdio
  - Controle de acesso
  - Disponibilidade

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, frequência, 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

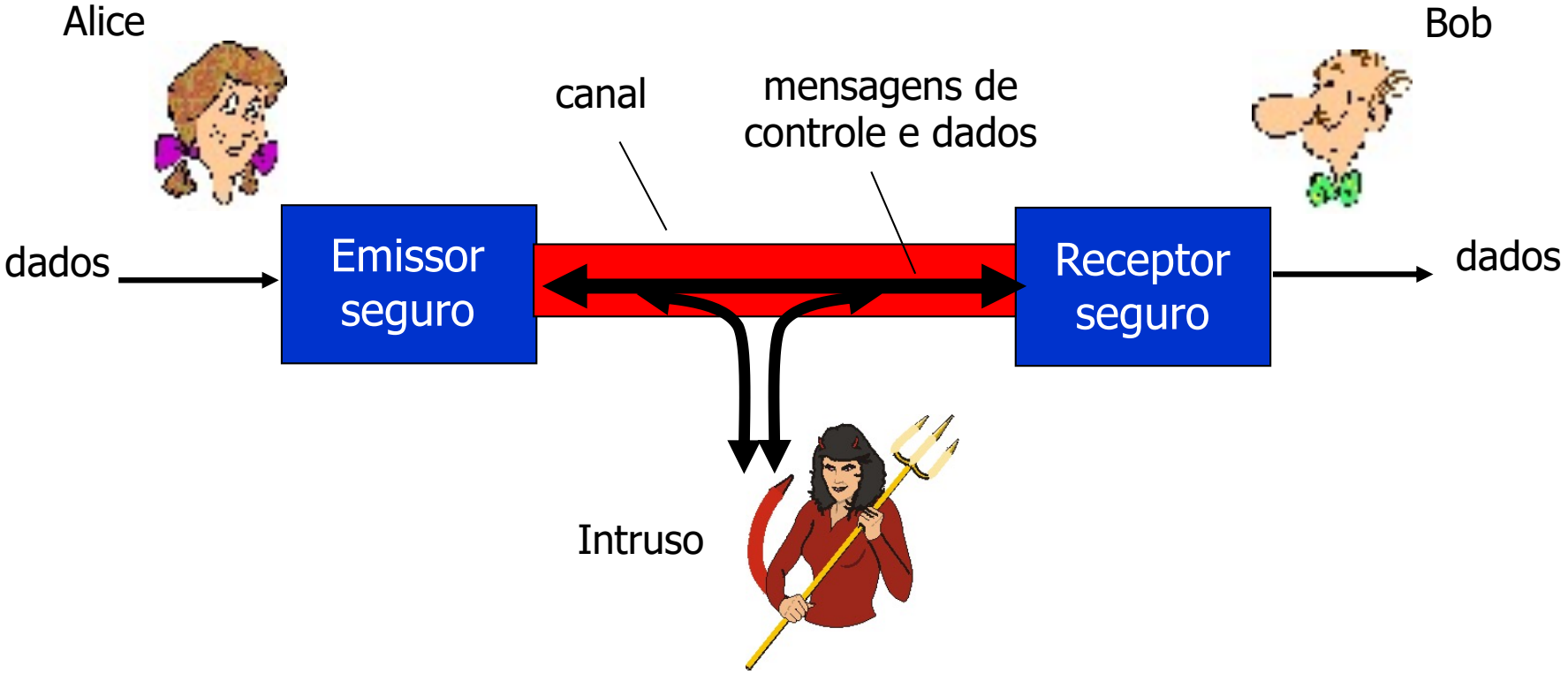
Segurança em Redes

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

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

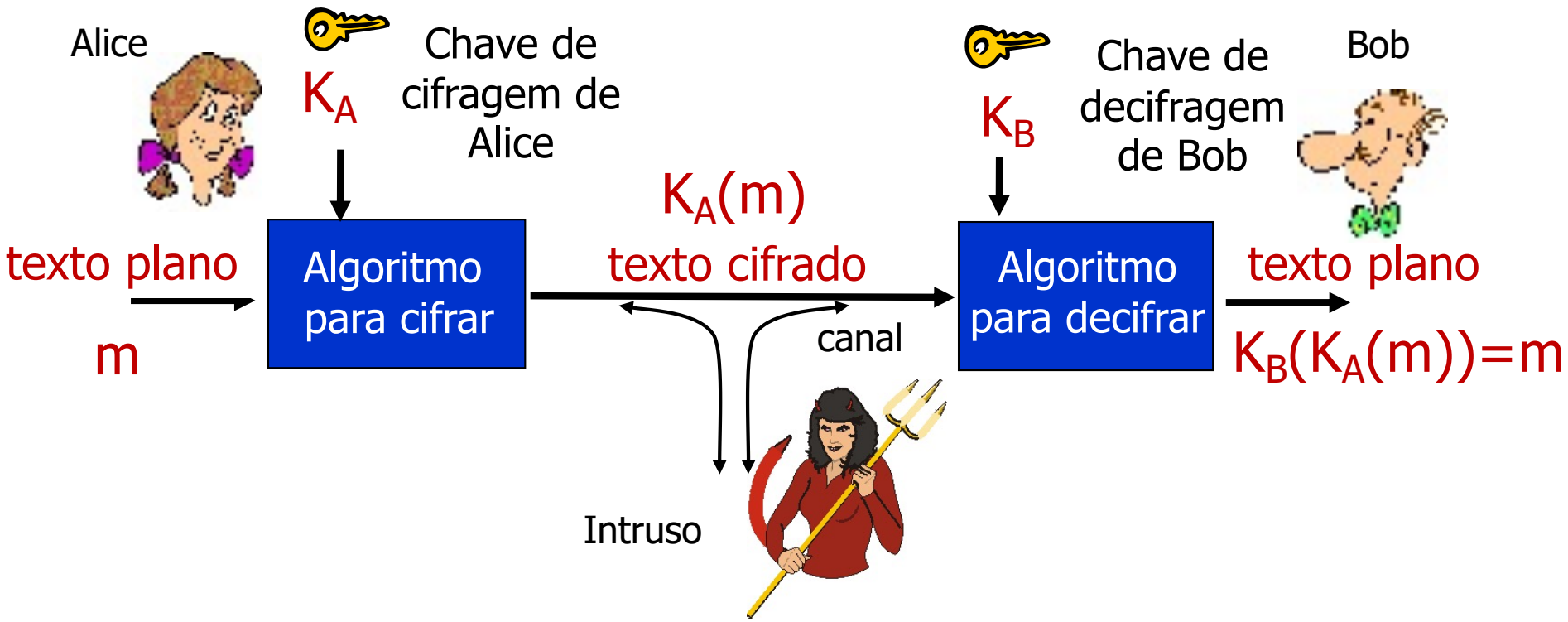
Instituto de computação

- 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

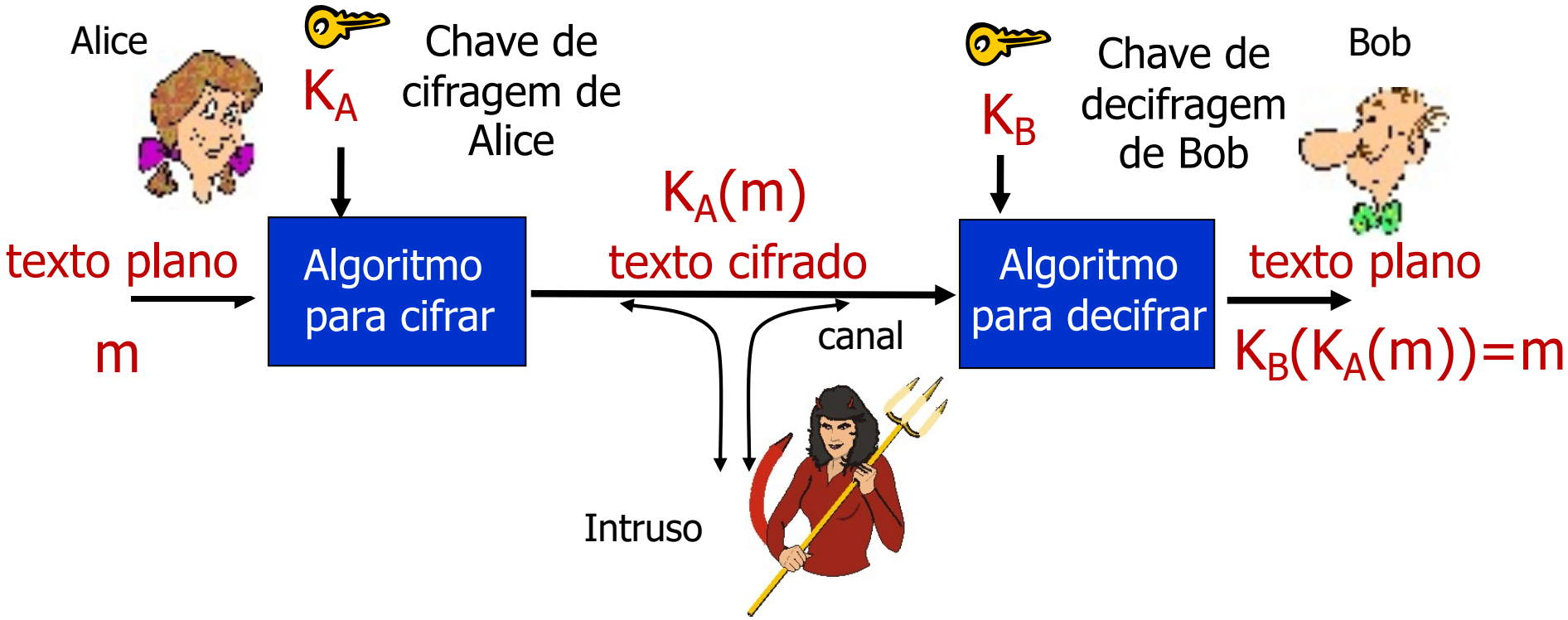


m: texto plano

$K_A(m)$: 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



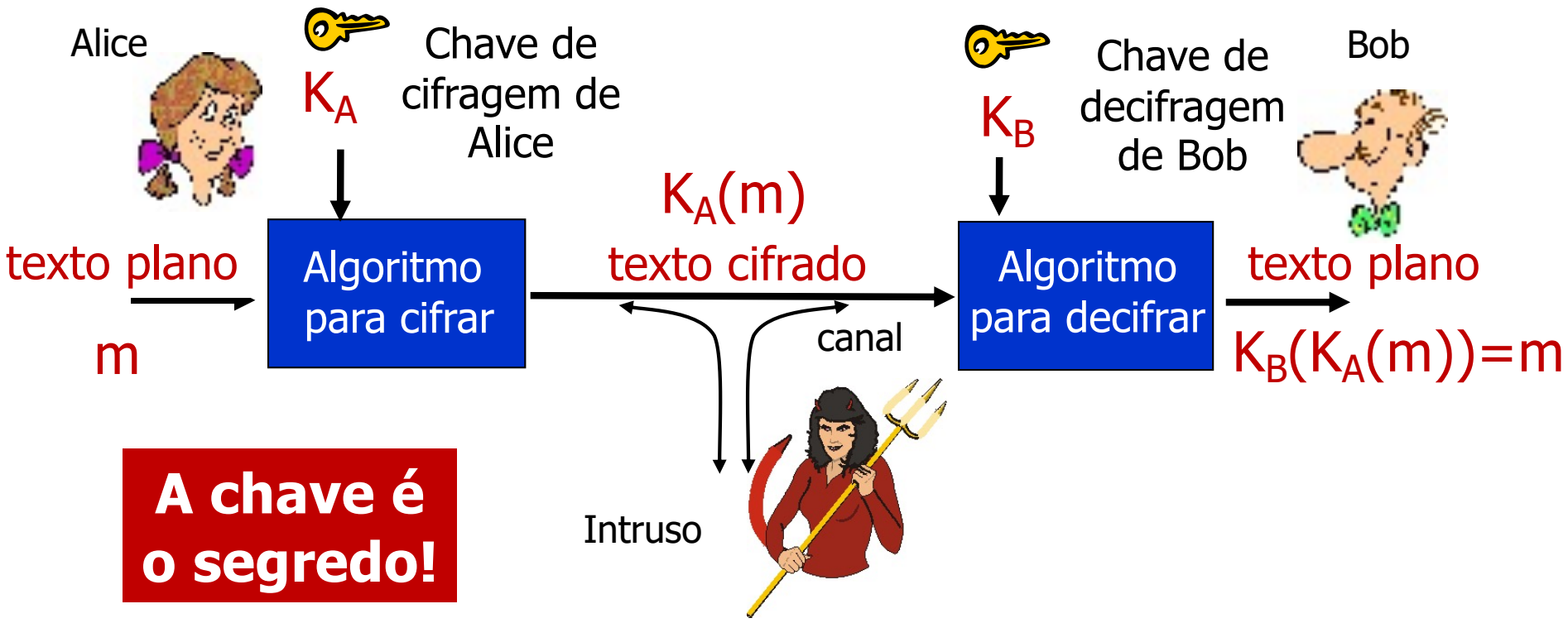
Algoritmos são conhecidos!

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

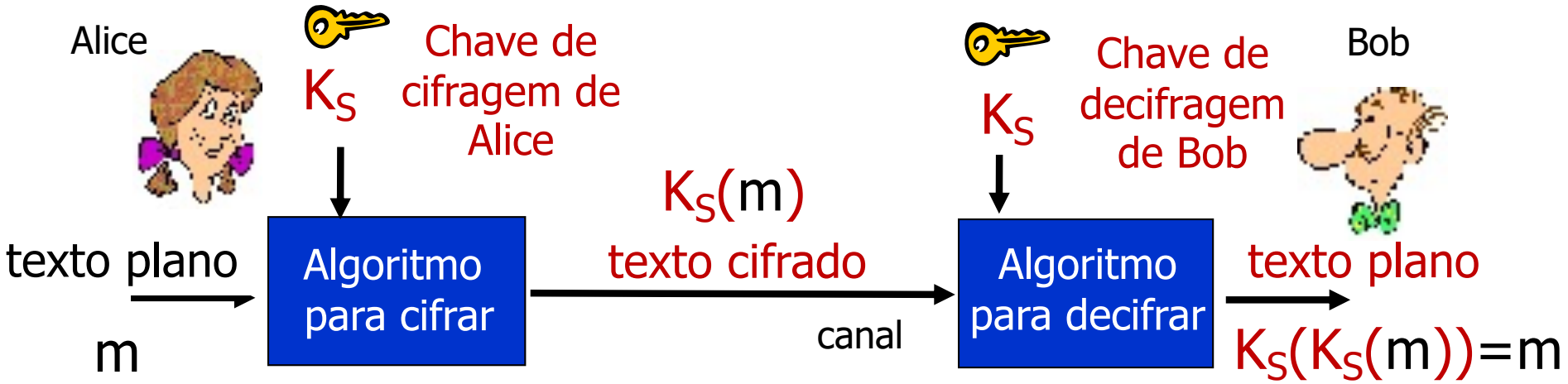
$m = K_B(K_A(m))$: texto plano recuperado, a partir da chave K_B

Tipos de Criptografia

Instituto de computação

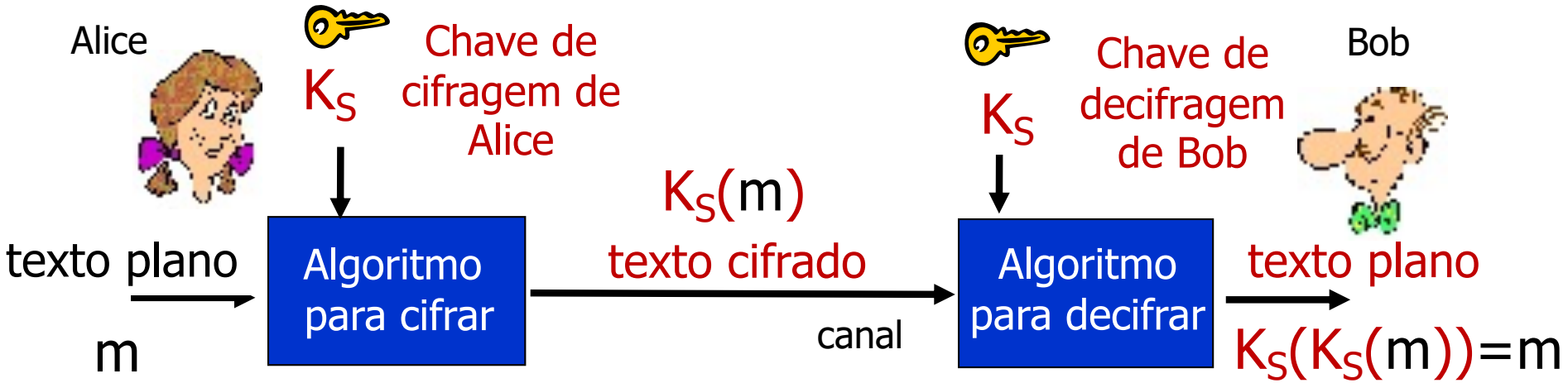
- Em geral, são usadas chaves em criptografia
  - Algoritmo é conhecido por todos e somente as chaves são secretas
- Criptografia de **chaves simétricas**
  - As chaves do remetente e do destinatário são **idênticas**
    - $K_A = K_B$
- Criptografia de **chaves assimétricas ou públicas**
  - As chaves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

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

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

Instituto de computação

- 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
010	101
011	100
100	011
101	010
110	000
111	001

Qual o texto cifrado para 010110001111 ?

101

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 ?

101000

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 ?

101000111

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 ?

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

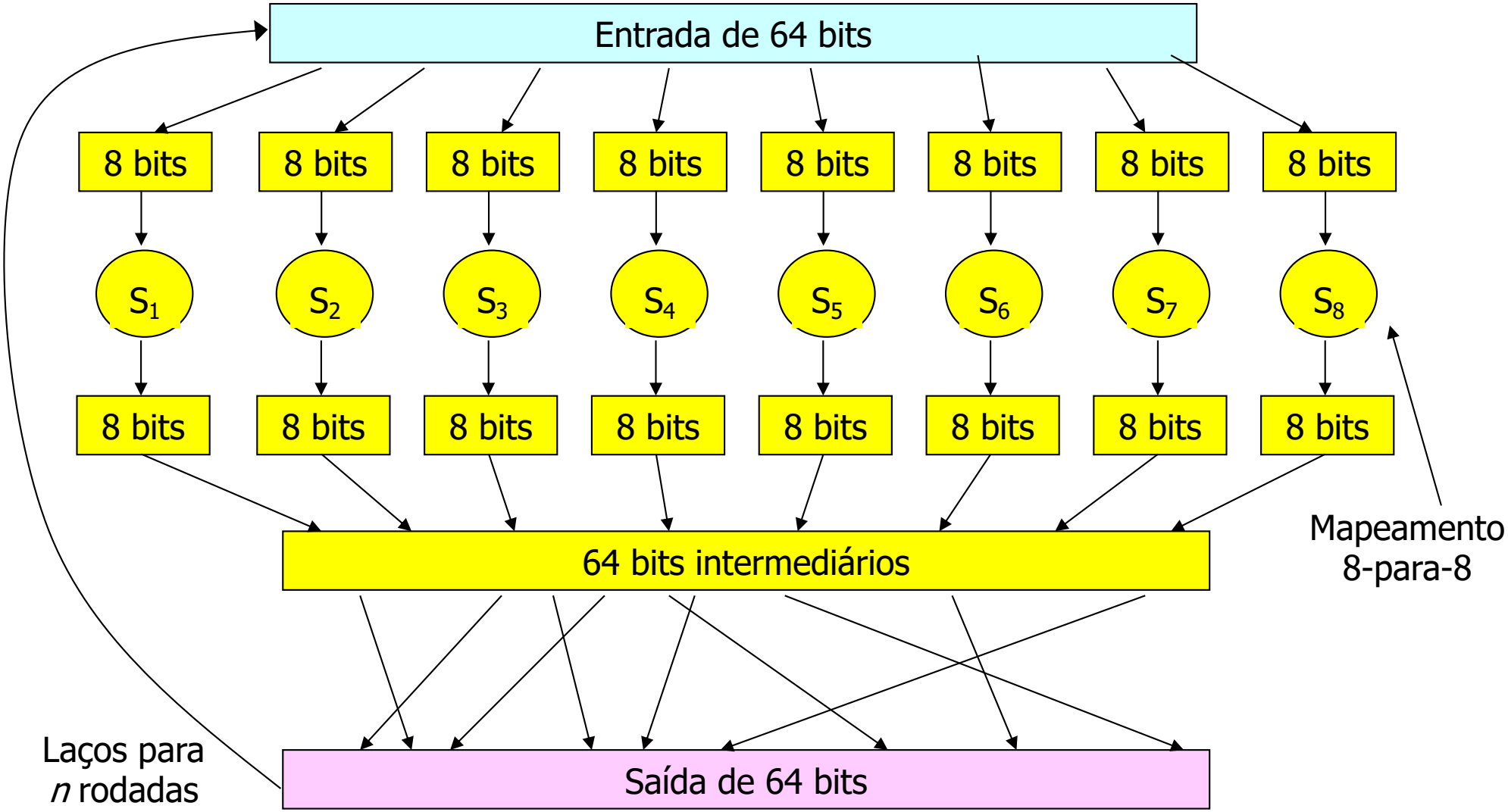
Instituto de computação

- 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

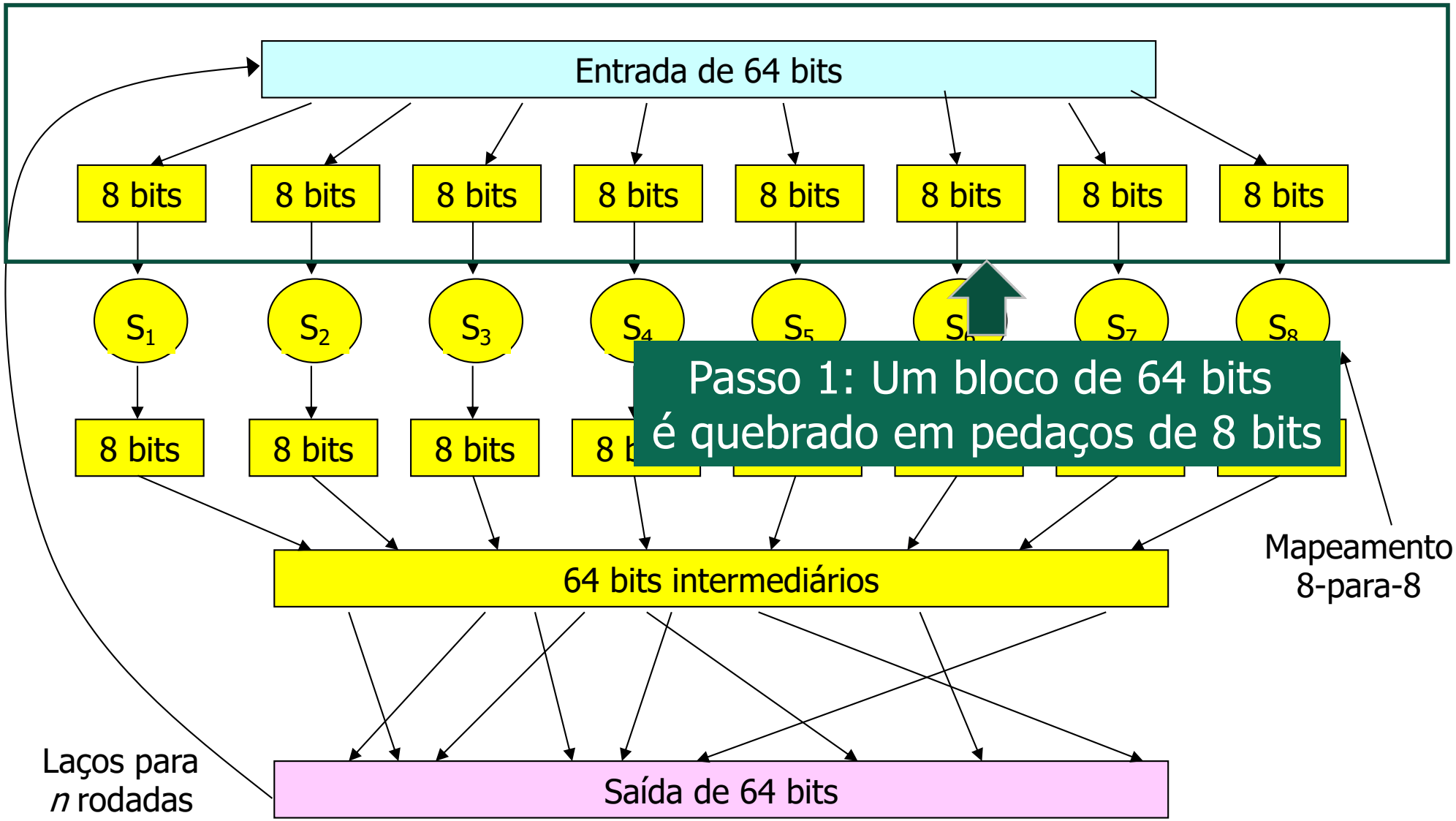
Solução: usar uma função que simula uma **tabela aleatória de permutas** ao invés da completa

Exemplo de uma Função



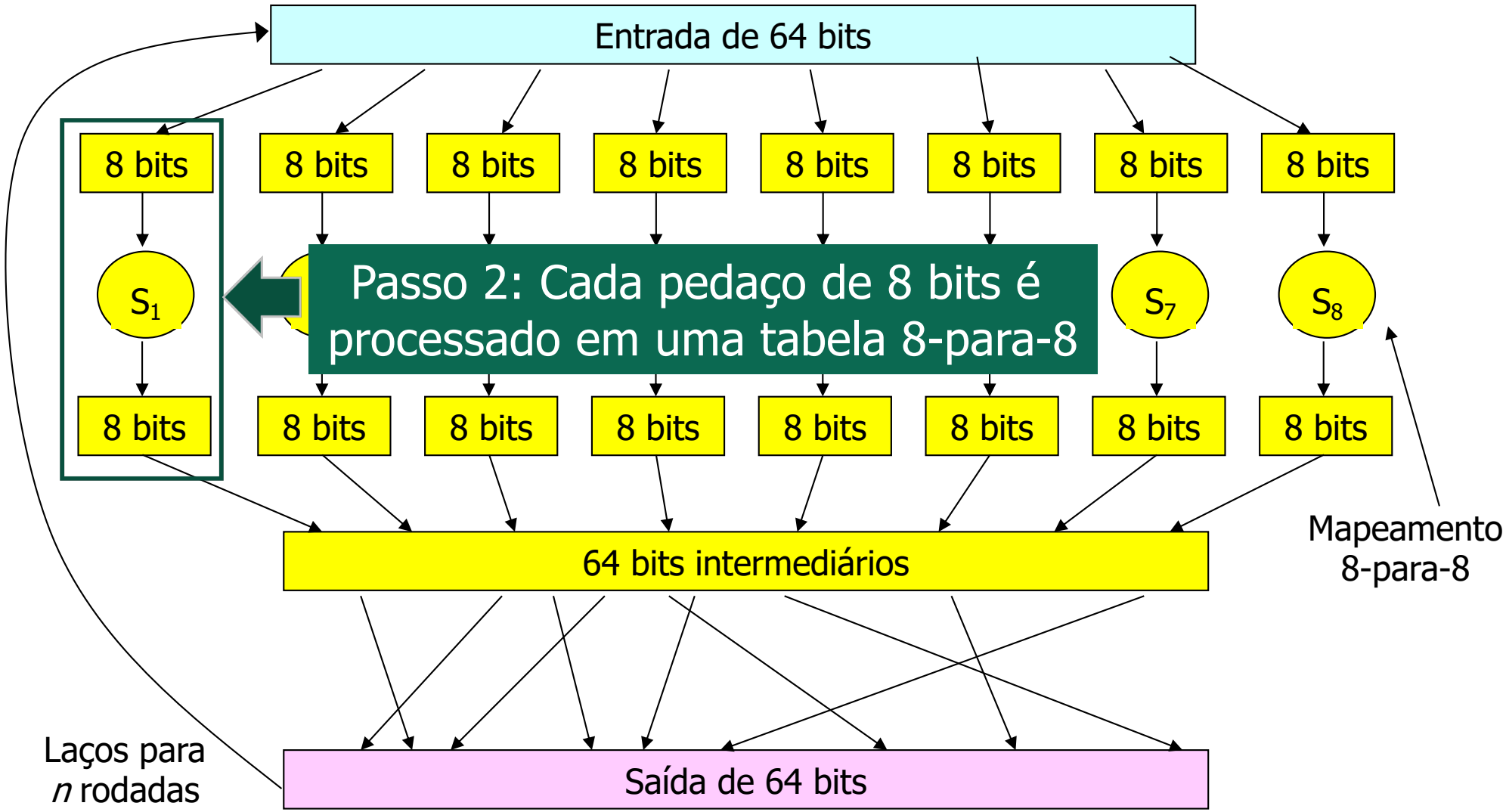
Extraído de Kaufman et al.

Exemplo de uma Função



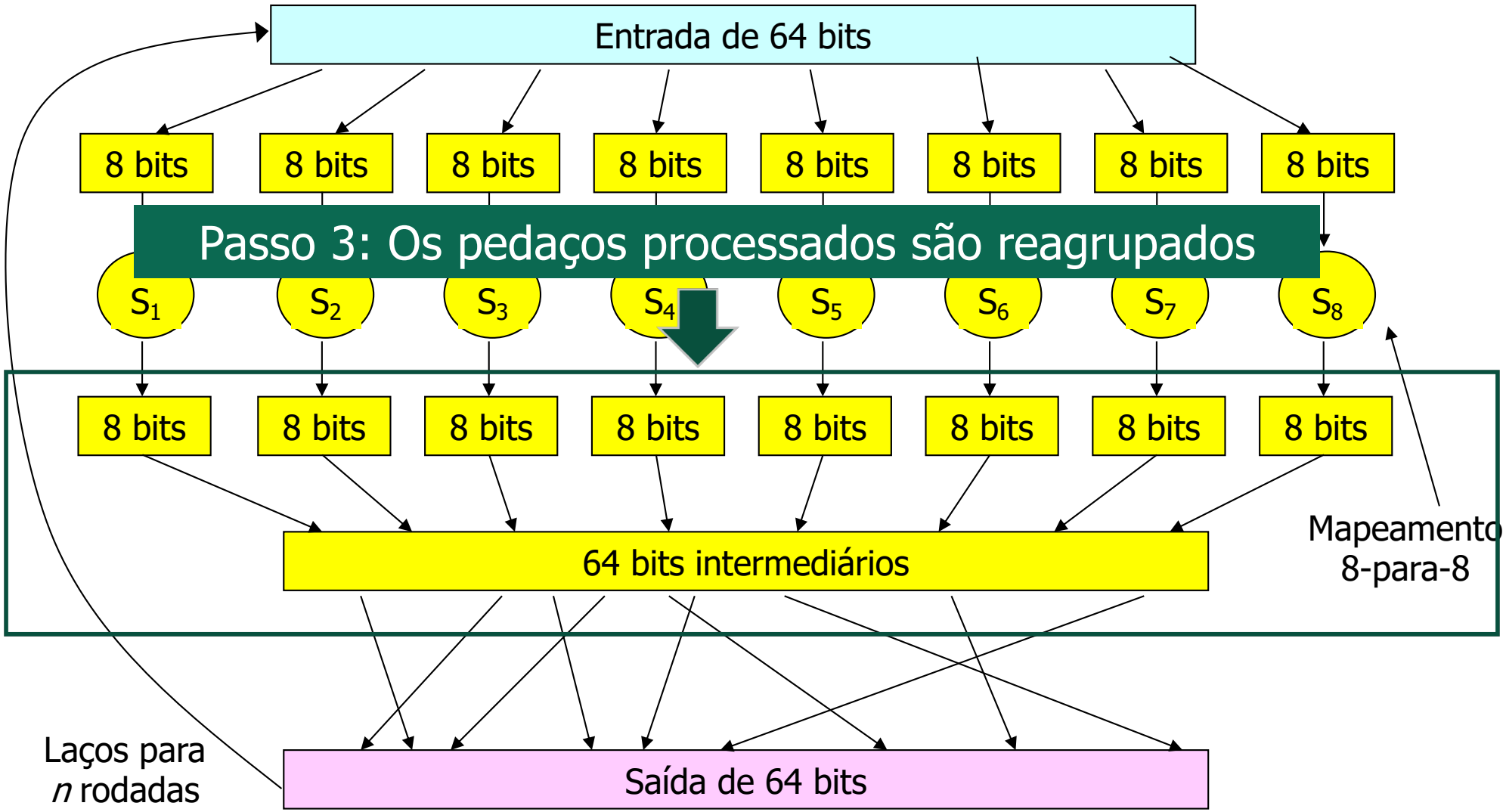
Extraído de Kaufman et al.

Exemplo de uma Função



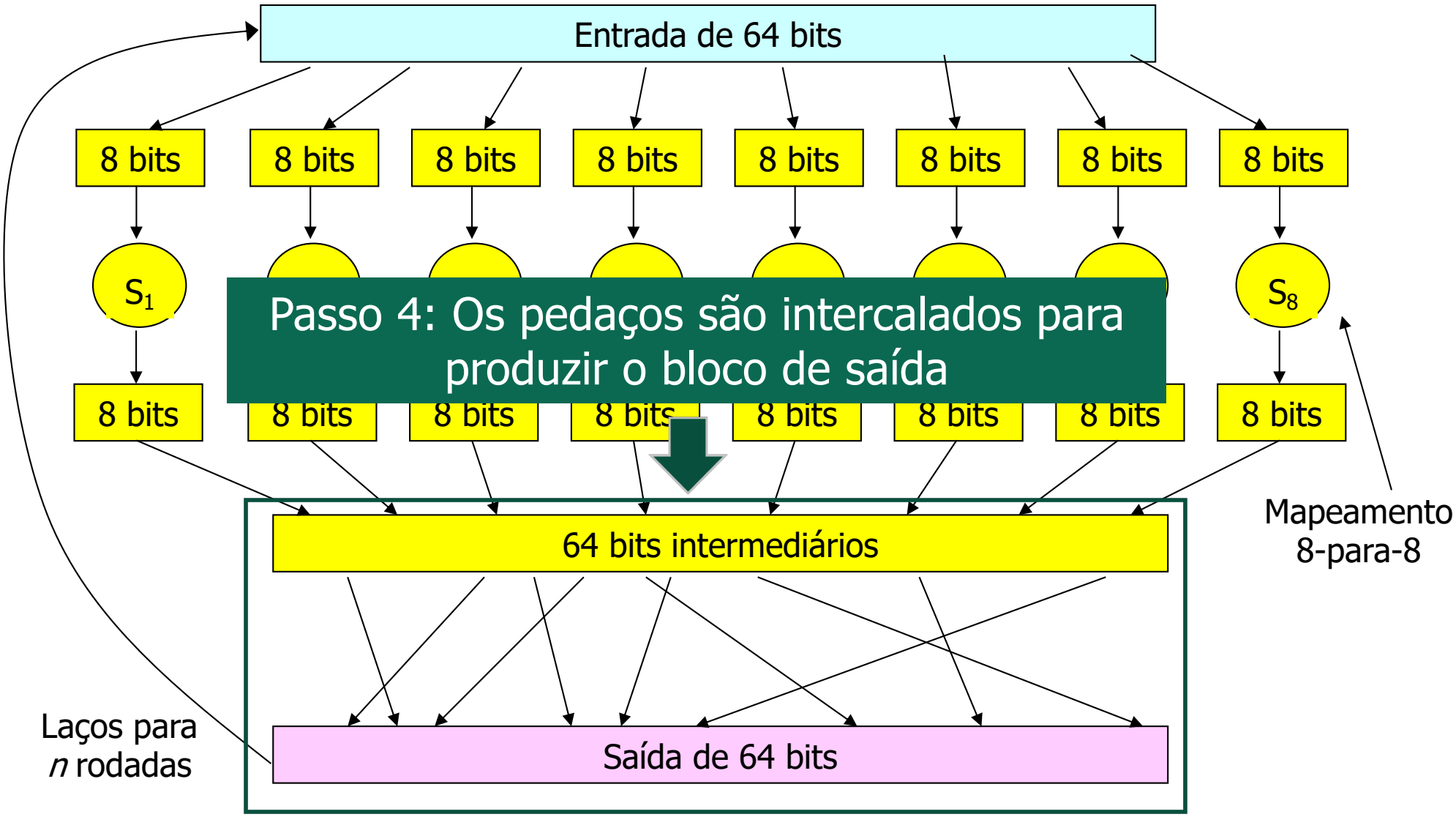
Extraído de Kaufman et al.

Exemplo de uma Função



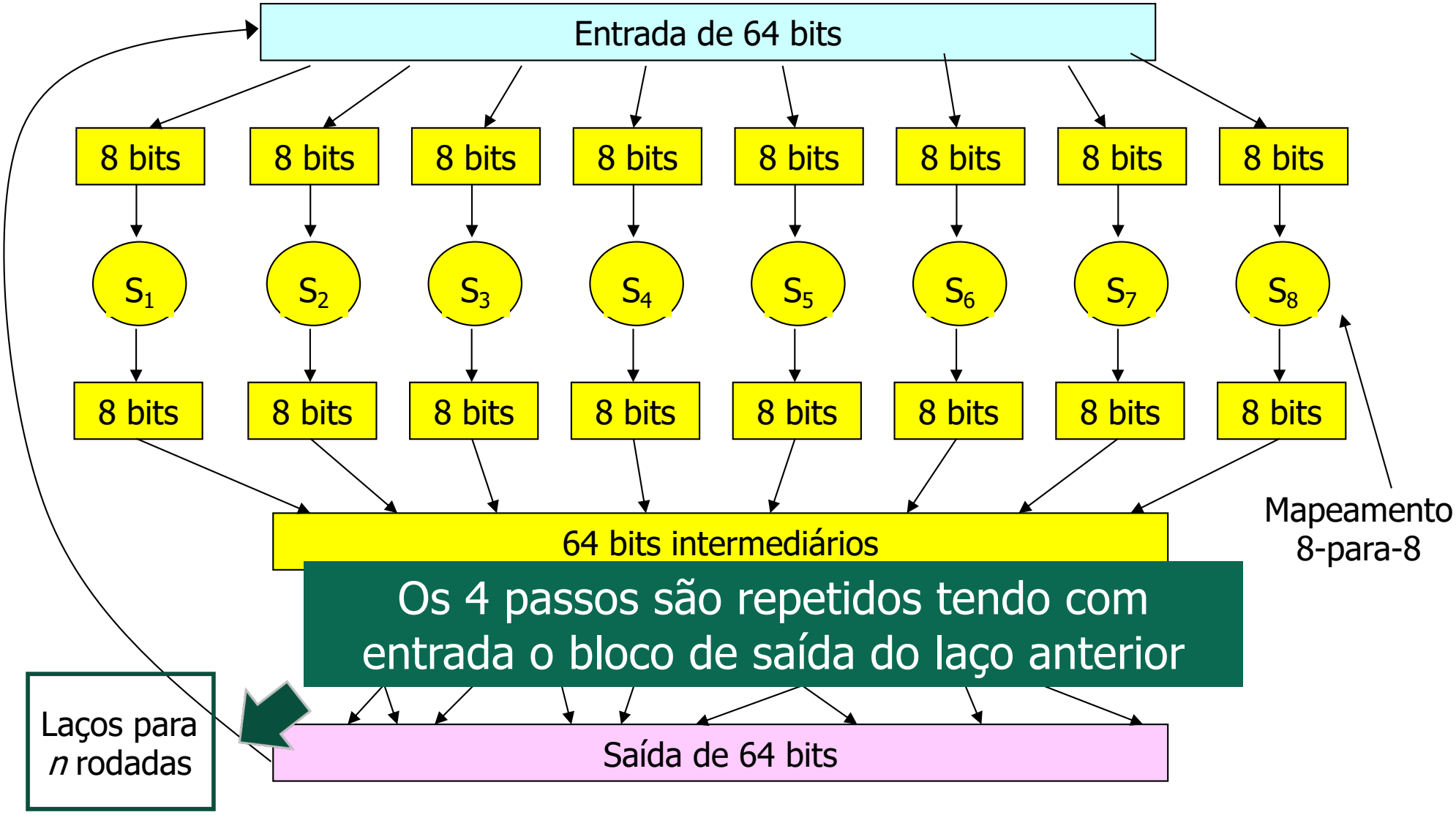
Extraído de Kaufman et al.

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

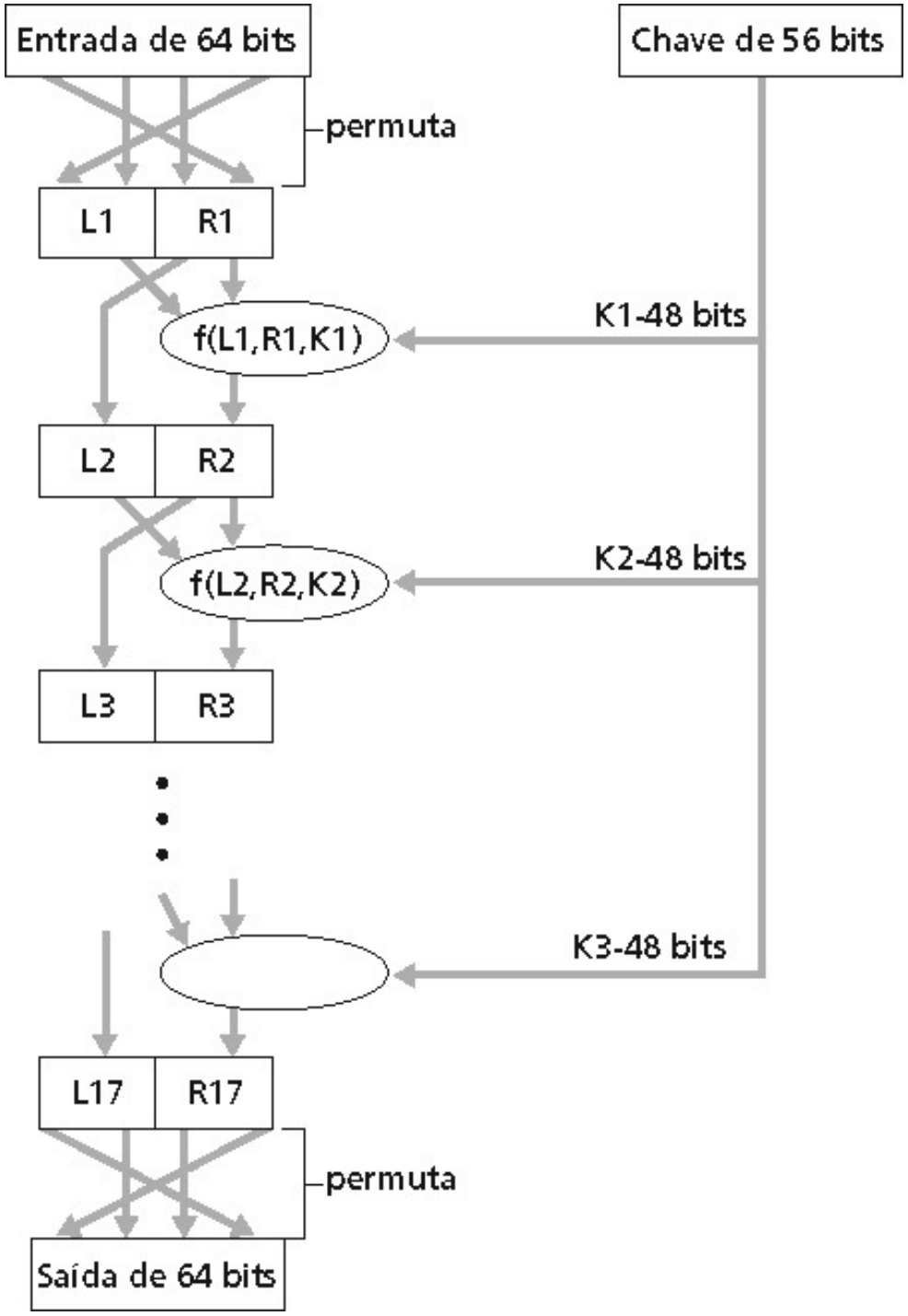
Data Encryption Standard (DES)

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

DES

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



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

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 decifragem 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?

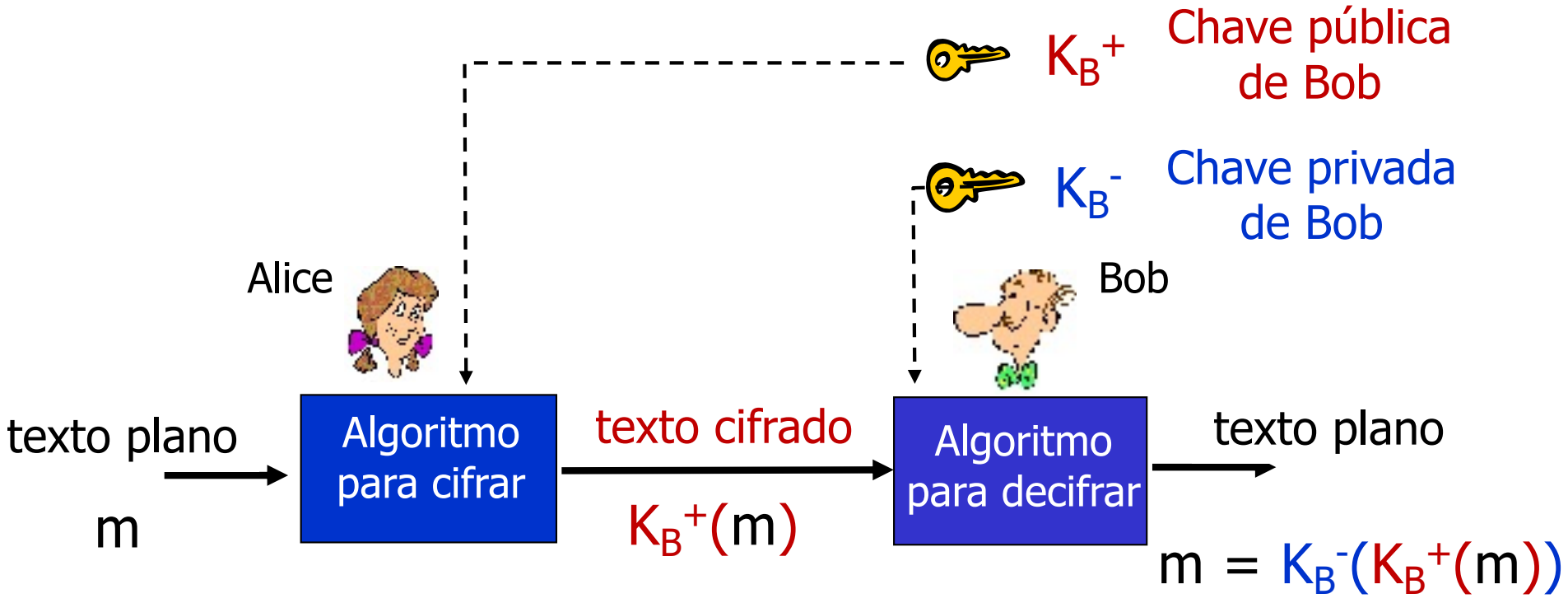
Criptografia de Chaves Assimétricas

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

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

  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

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

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

Algoritmo RSA: Rivest, Shamir e Adleman

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 entre si
4. Escolher d tal que $ed\text{-}1$ é exatamente divísivel por z
   - Ou seja, $ed \bmod z = 1$
5. A chave pública é $\boxed{(n,e)}$ e a chave privada é $\boxed{(n,d)}$

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

Mágica: $$m = \underbrace{(m^e \bmod n)}_{c}{}^d \bmod n$$

RSA: Exemplo

- Bob escolhe
  - $p = 5$ e $q = 7$ → $n = 35$ e $z = 24$
  - $e = 5$ → e e z são primos entre si
  - $d = 29$ → 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 $c = 17$ → $c^d = 17^{29} =$ 481968572106750915091411825223071697
  - $m = c^d \bmod n = 12$ → texto recebido é decifrado

RSA: Outra Propriedade

- Ordem de cifração e decifração não importam

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

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

Usa-se a chave pública, seguida da chave privada

Usa-se a chave privada, seguida da chave pública

O resultado é o mesmo!

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

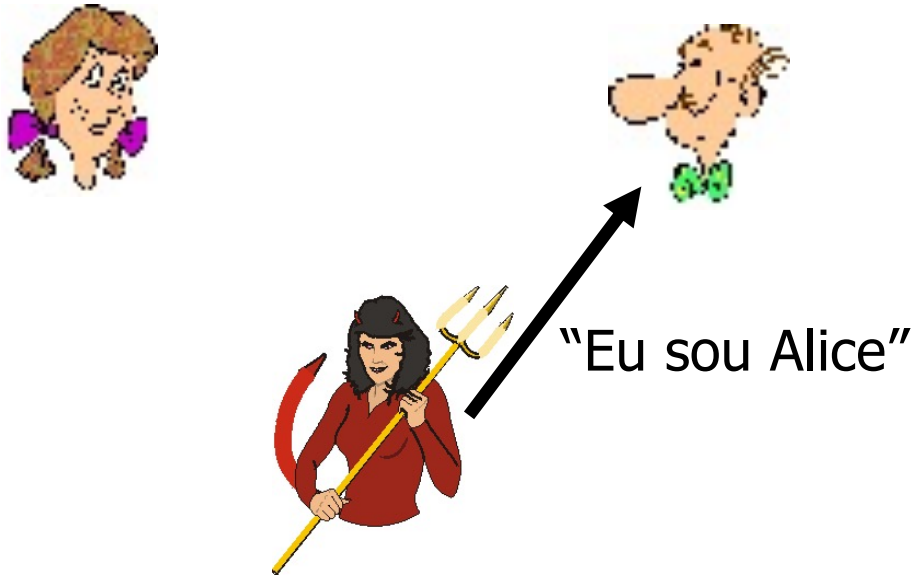
Autenticação

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

Cenário de falha?

Autenticação

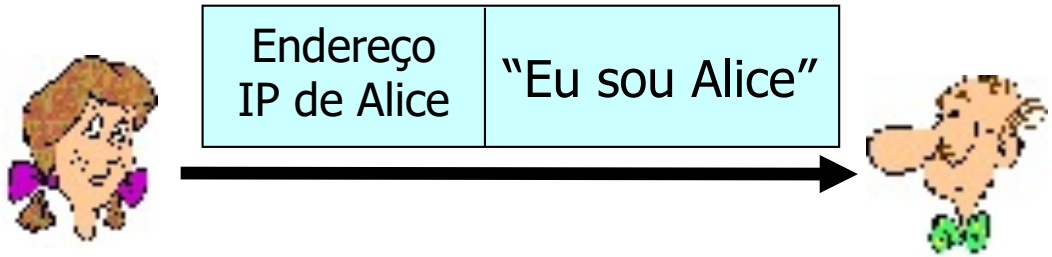
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice"



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

Autenticação

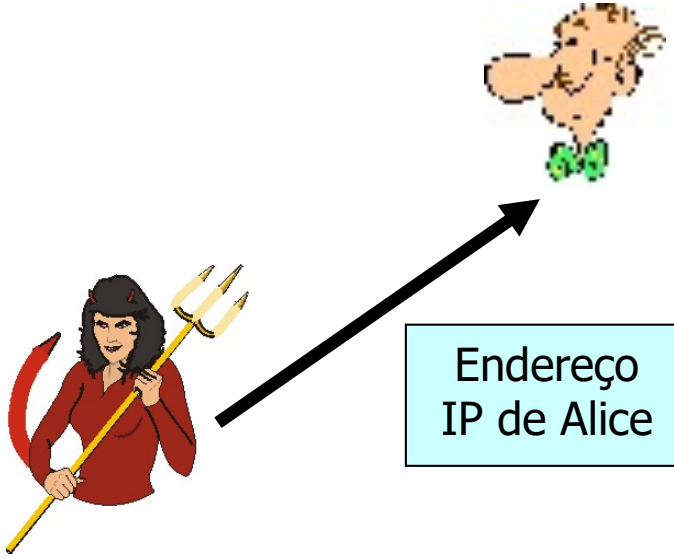
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia junto o seu endereço IP como "prova"



Cenário de falha?

Autenticação

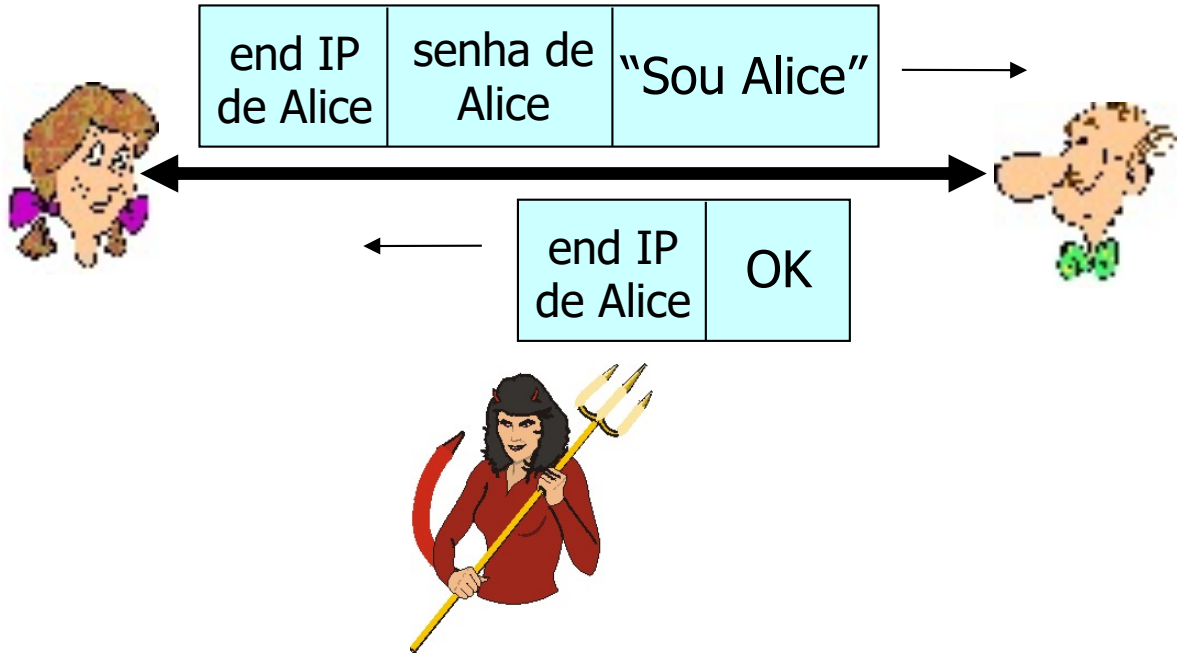
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia junto o seu endereço IP como "prova"



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

Autenticação

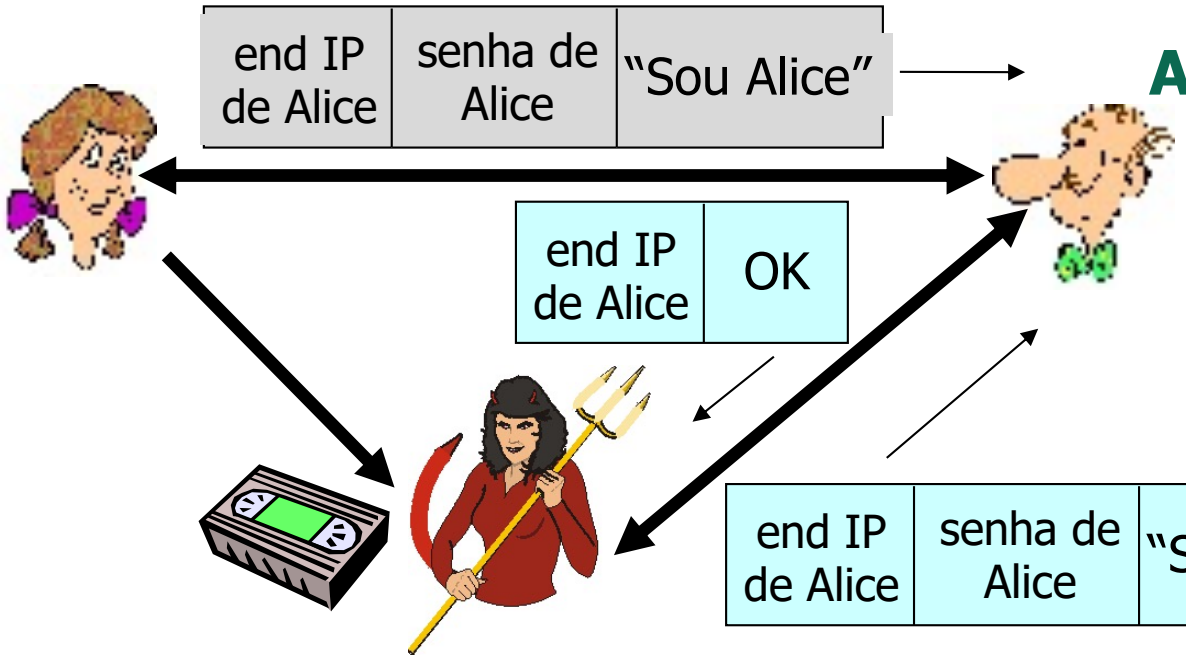
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia a sua senha secreta como "prova"



Cenário de falha?

Autenticação

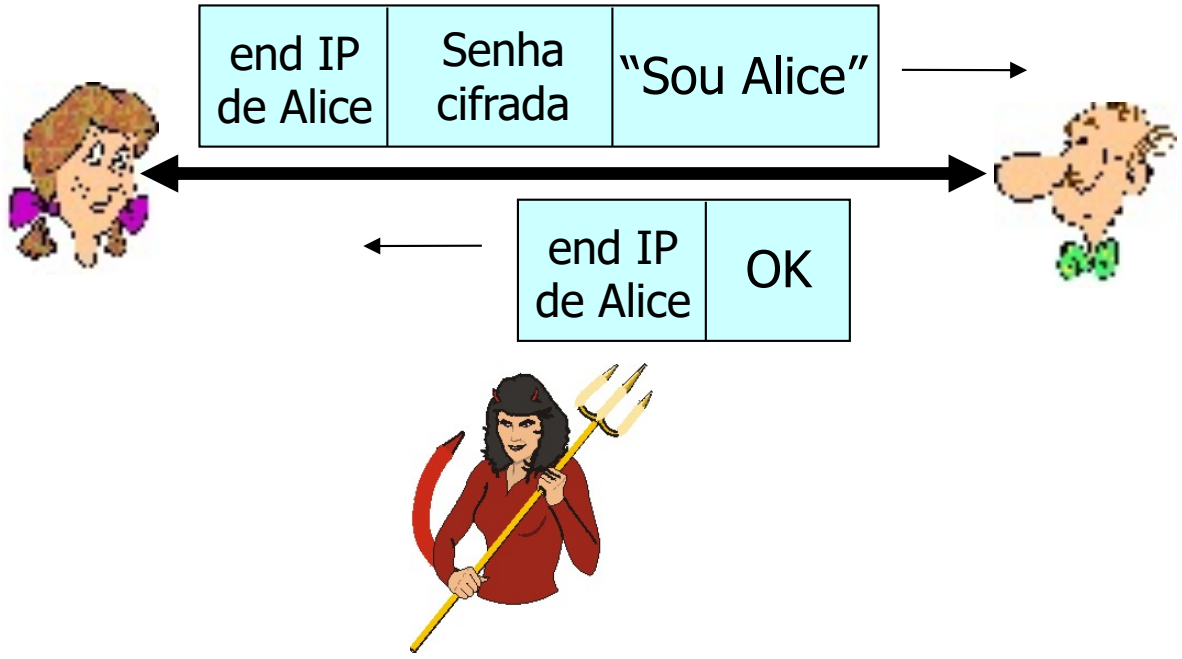
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia a sua senha secreta como "prova"



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

Autenticação

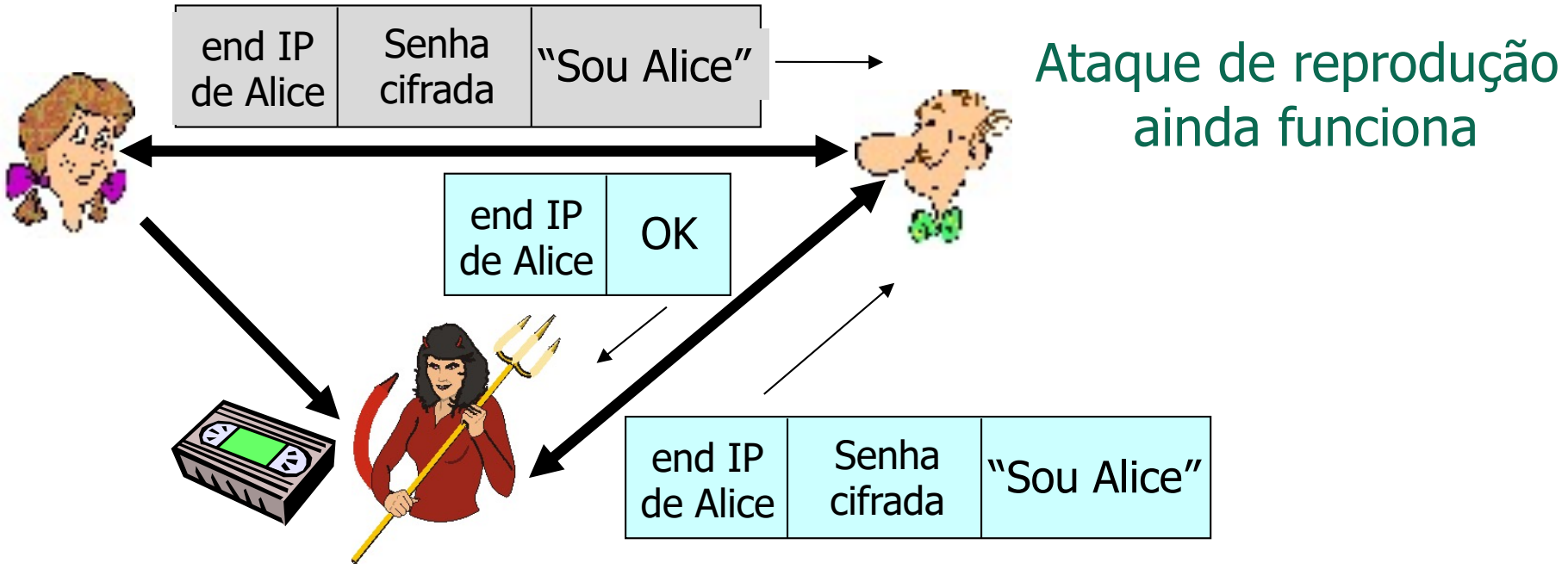
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia a sua senha secreta **cifrada** como "prova"



Cenário de falha?

Autenticação

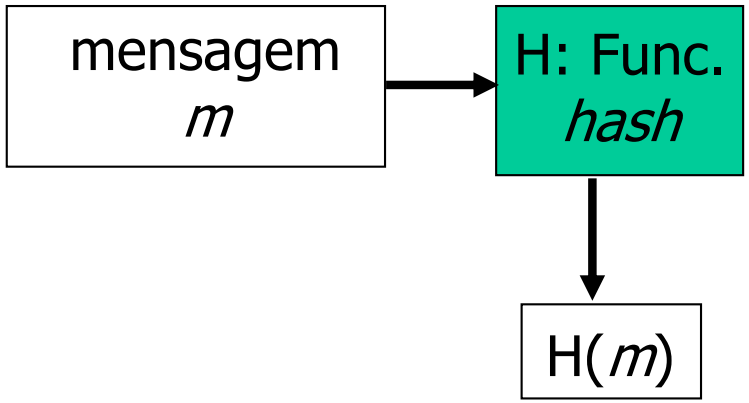
- Objetivo: Bob quer que Alice "prove" a sua identidade
  - Alice diz "Eu sou Alice" e envia a sua senha secreta **cifrada** como "prova"



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

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

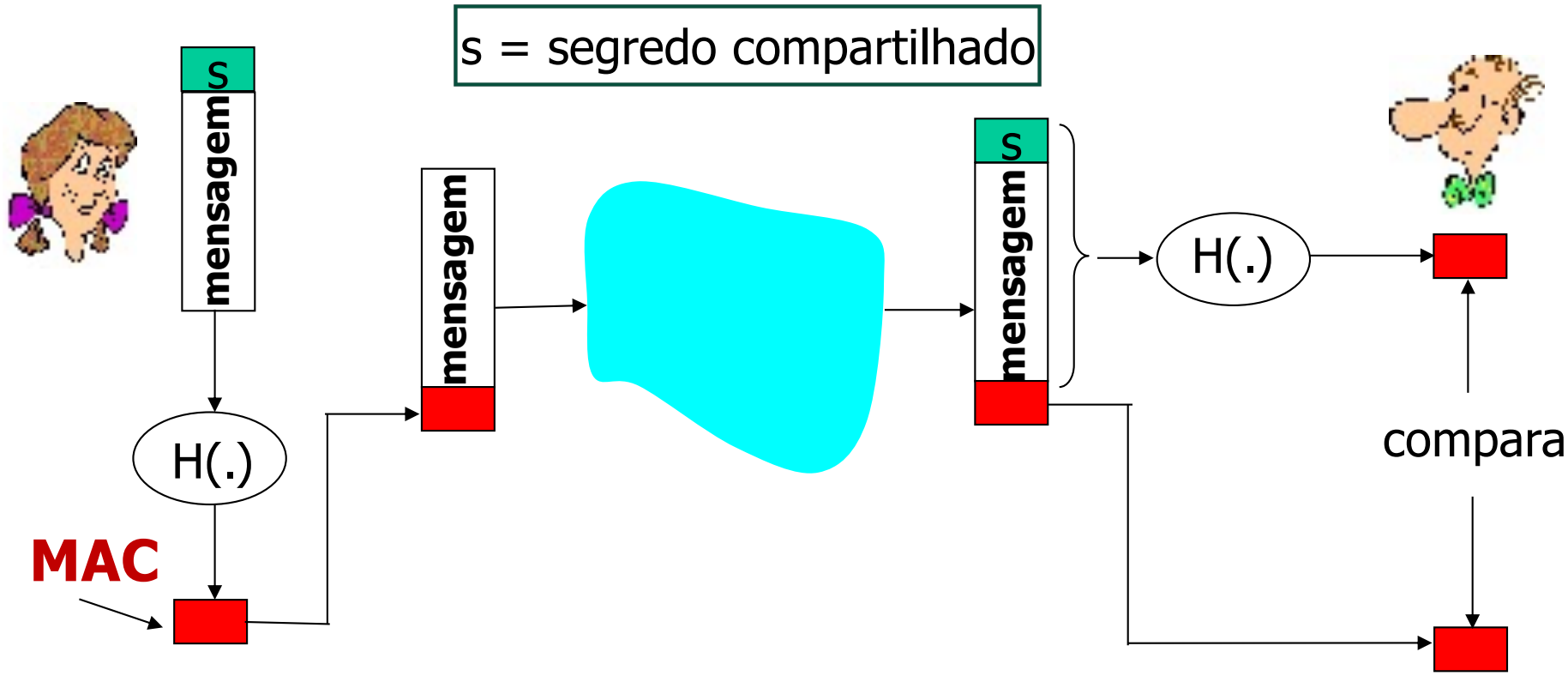
- Função *hash* SHA-1
  - É o padrão atual dos EUA
  - Resumos de mensagens de 160 bits

Algoritmos de Funções *Hash*

- Função *hash* MD5
  - Amplamente usada
    - Autenticação do Linux
  - Definida pela RFC 1321
  - Computa resumos de
    de 4
    ção, mistura
- Função *hash* SHA-1
  - É o padrão atual dos EUA
  - Resumos de mensagens de 160 bits

Como garantir integridade usando funções *hash*?

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 $m||MD_m$

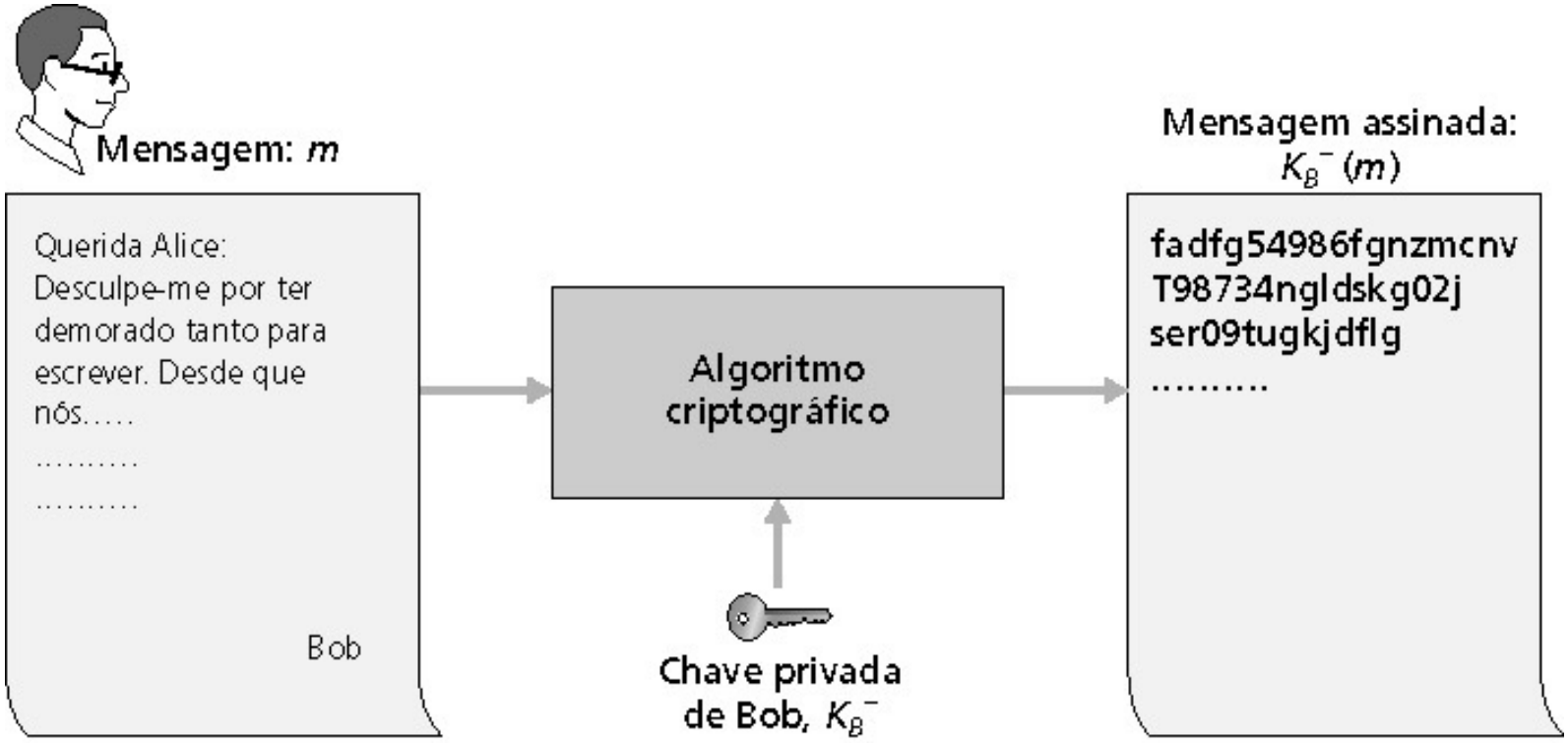
Assinaturas Digitais

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

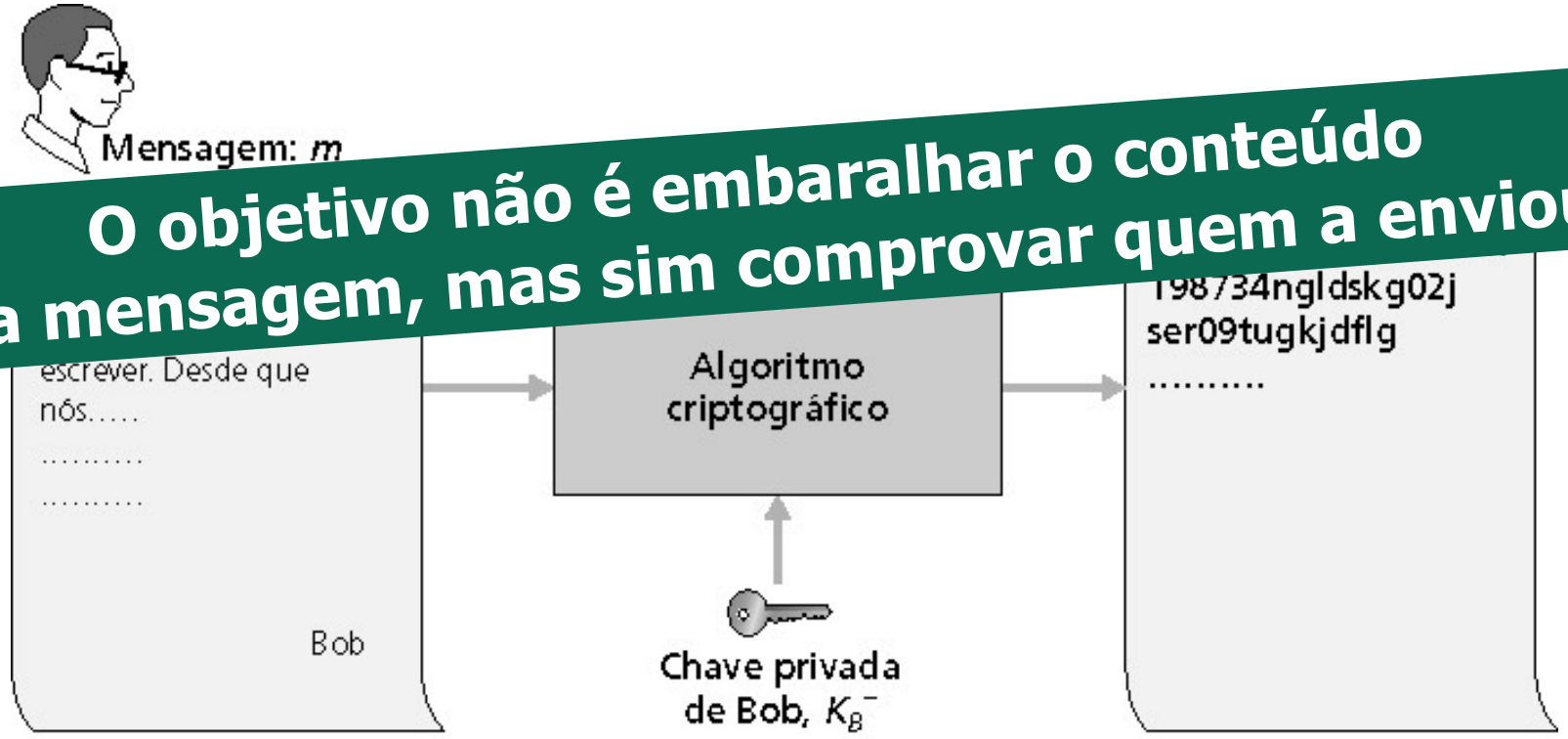
Assinaturas Digitais

- 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

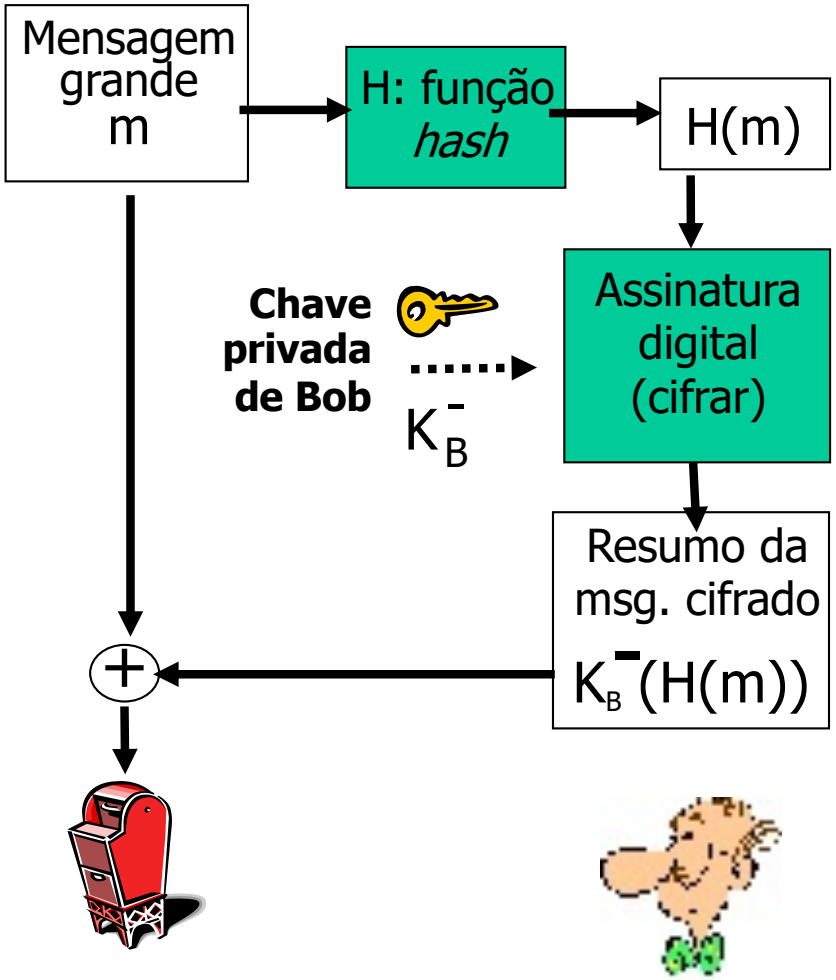
- Assinatura digital simples para a mensagem *m*
  - Bob assina *m* cifrando com a sua chave privada K_B^-, criando mensagem "assinada", $K_B^-(m)$



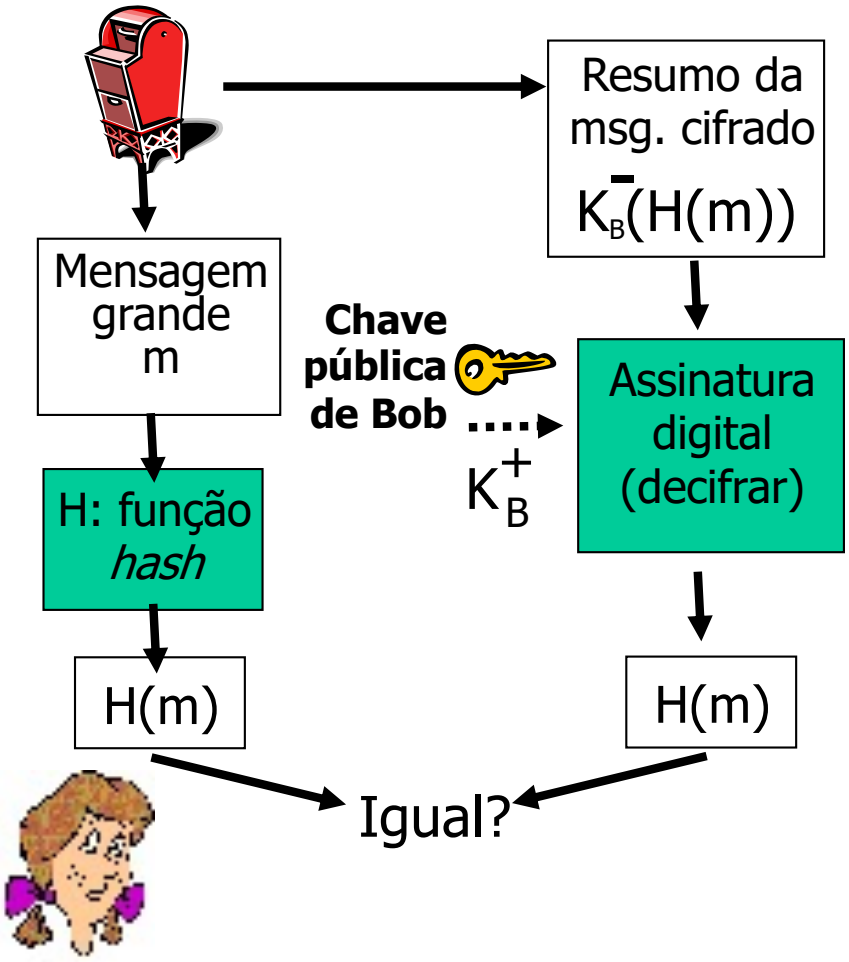
O objetivo não é embaralhar o conteúdo da mensagem, mas sim comprovar quem a enviou

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 o pedido
  - Trudy envia o pedido para a Loja de Pizza
  - Trudy também 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

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)

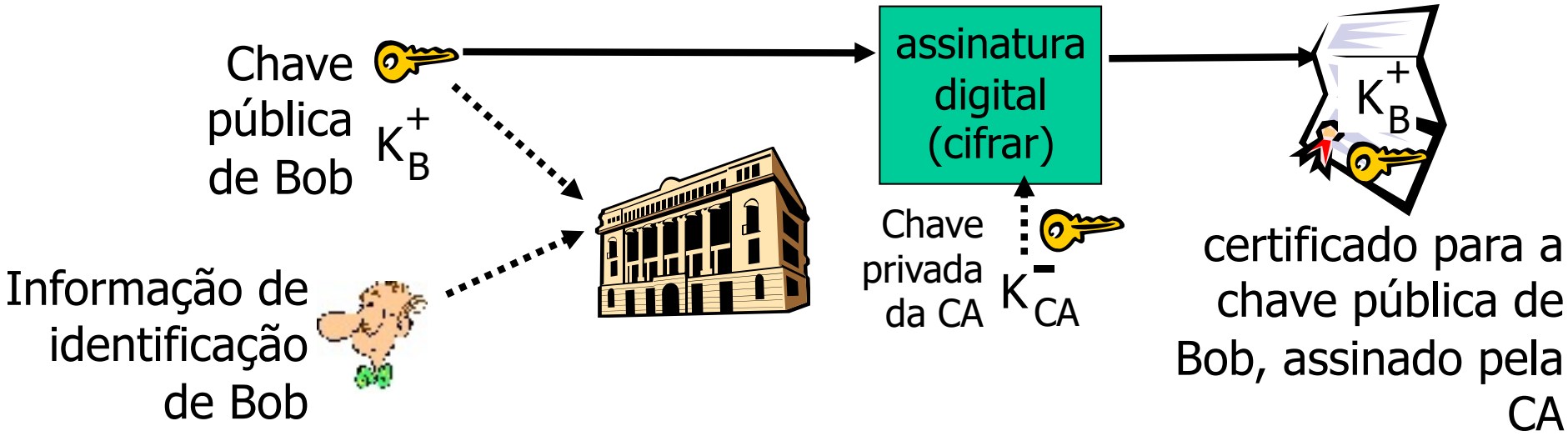
Autoridades Certificadoras (CAs)

Instituto de computação

- 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"

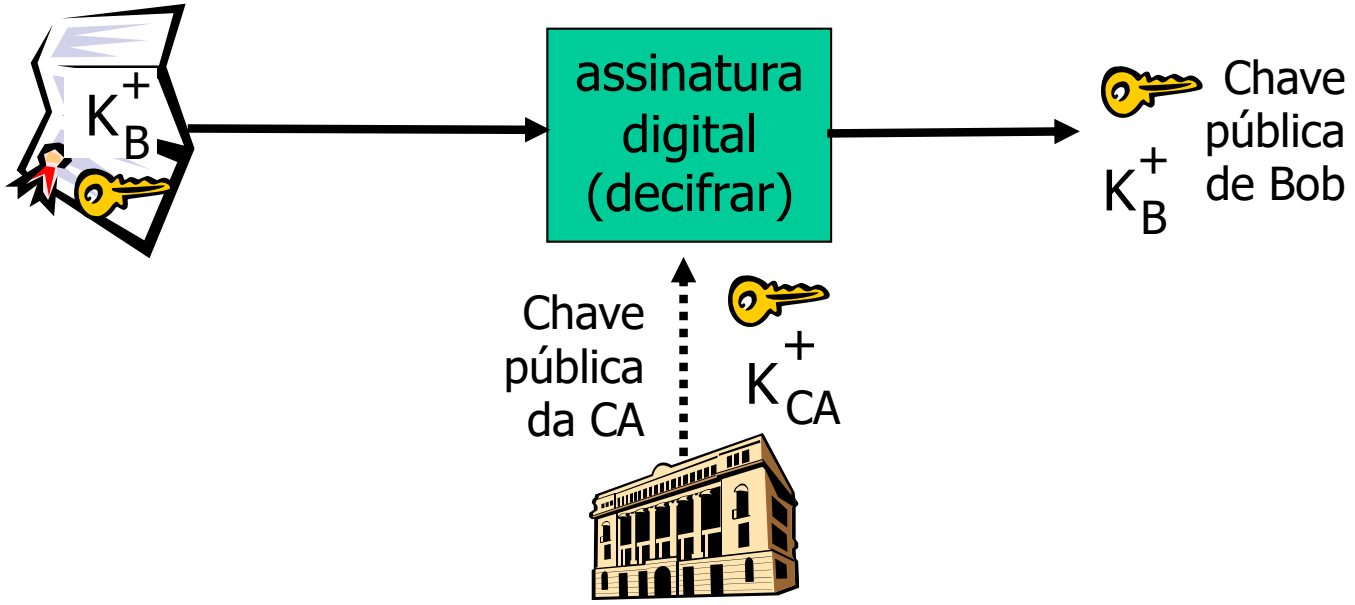
Autoridades Certificadoras (CAs)

- Associam uma chave pública a uma entidade particular, E



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
  - ICP-Brasil
  - ICPEdu - Infraestrutura de Chaves Públicas para Ensino e Pesquisa (AC) - serviço de certificação digital oferecido pela RNP

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

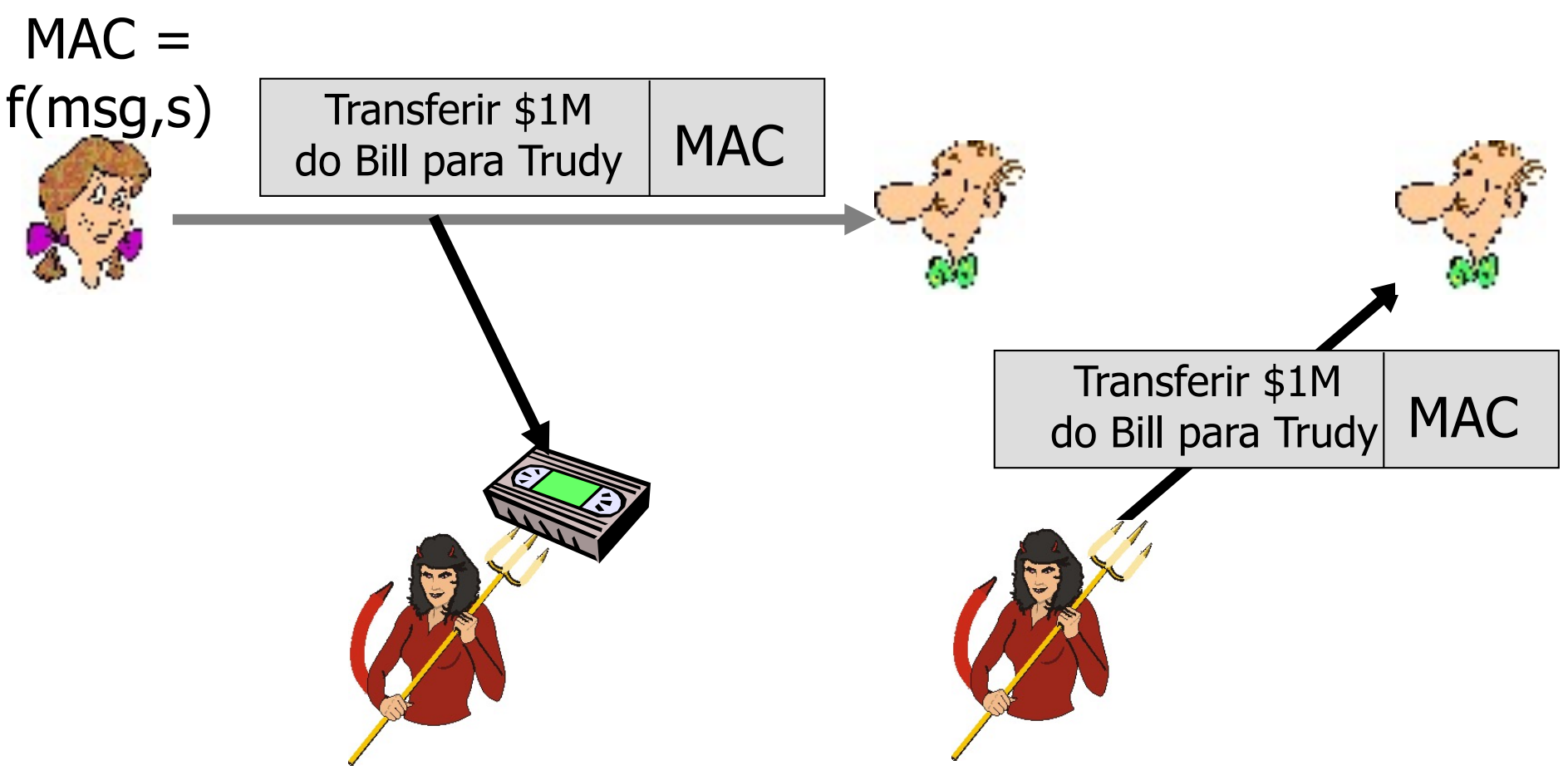
Instituto de computação

- 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? **Sim!**
  - É possível afirmar que Alice **enviou** a mensagem? **Não!**

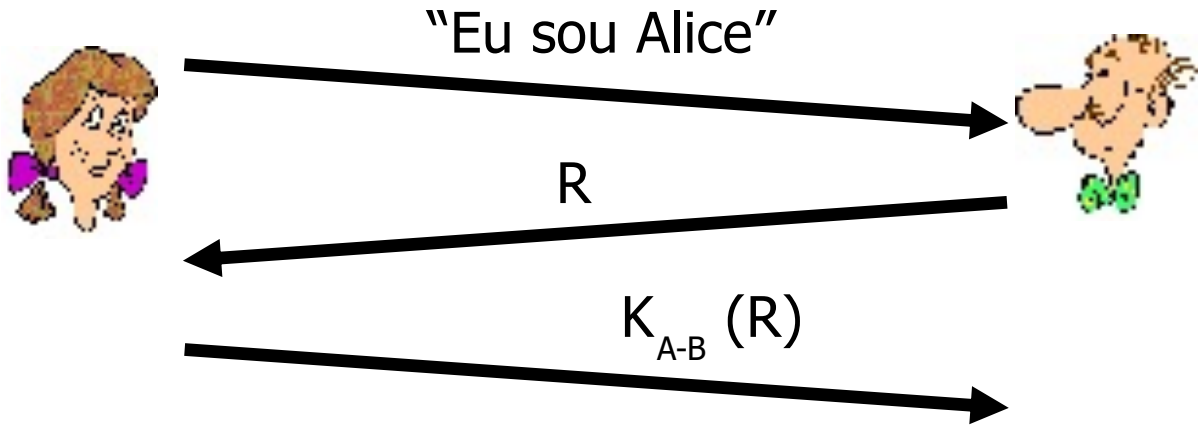
Ataque de Reprodução

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

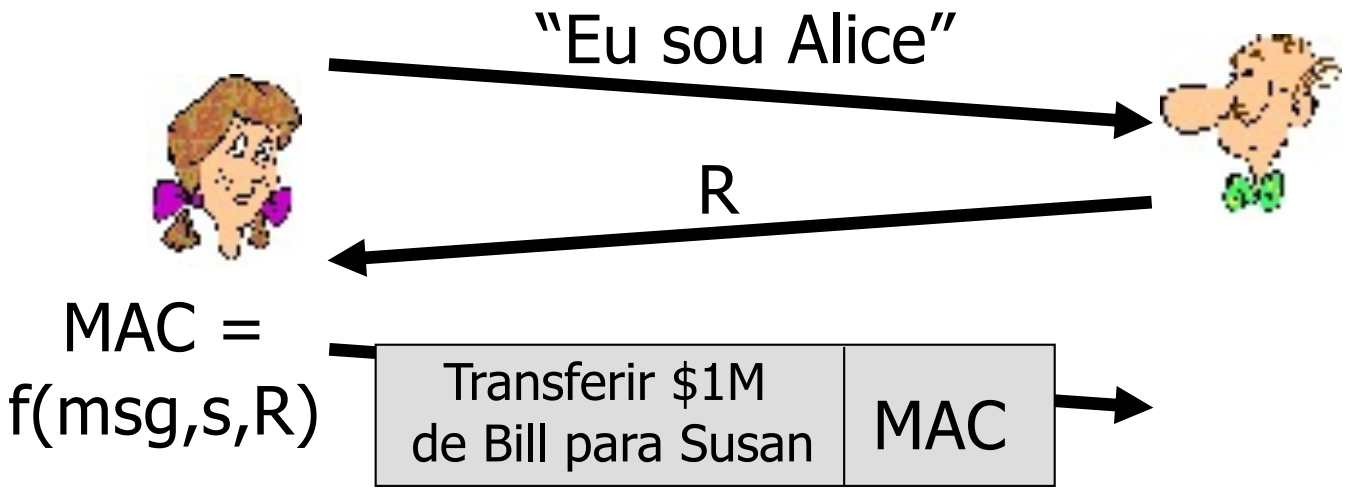


• Para identificar Alice "ao vivo", Bob envia para Alice um *nonce* R em "claro"

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

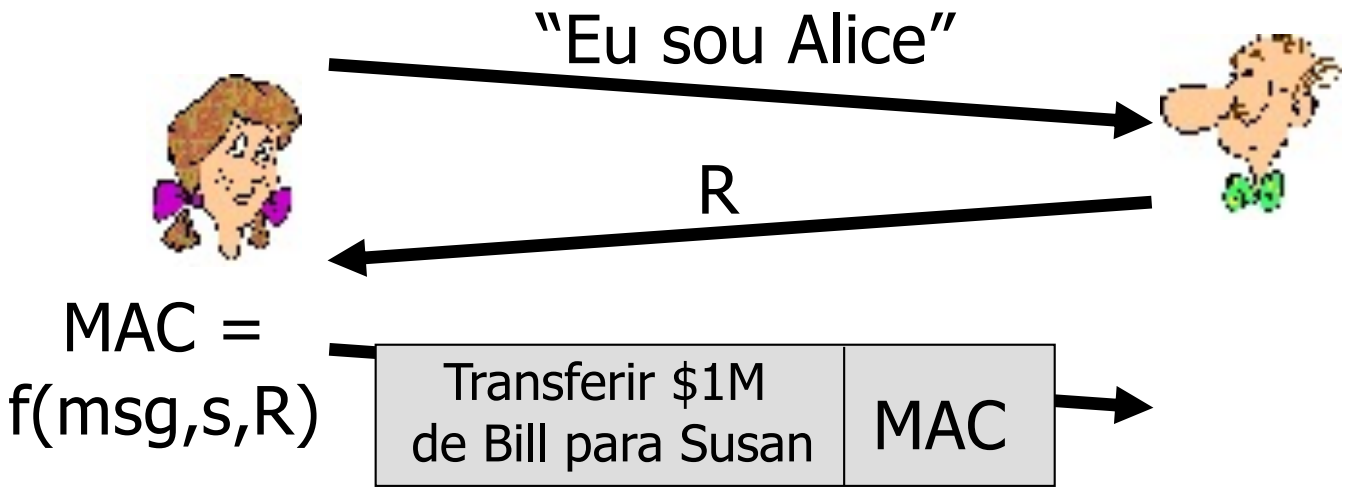
Defesa ao Ataque de Reprodução

- Uso de *nonces* com MAC



Defesa ao Ataque de Reprodução

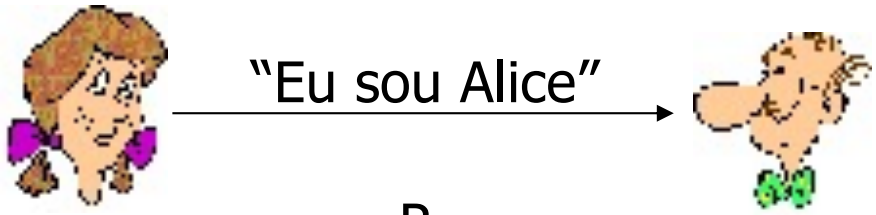
- Uso de *nonces* com MAC



Problema: requer chave secreta compartilhada

Autenticação com Criptografia de Chave Pública

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



R

$K_A^-(R)$

Envie a sua chave pública K_A^+

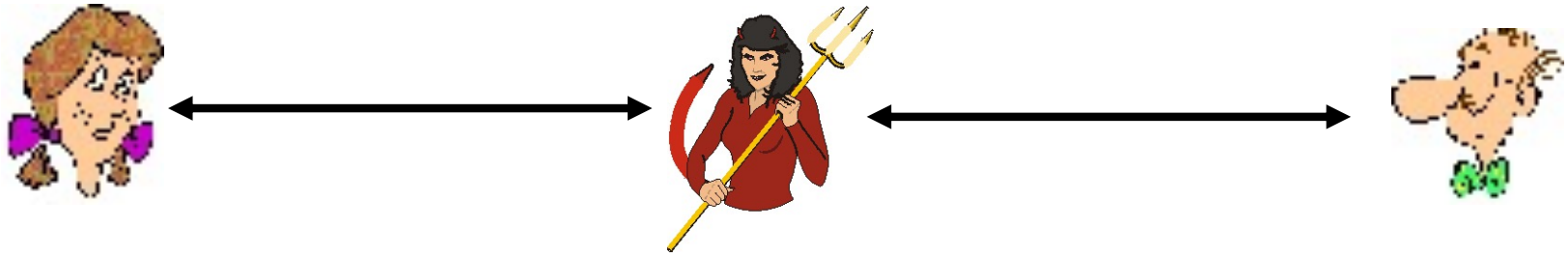
K_A^+

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
  - 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

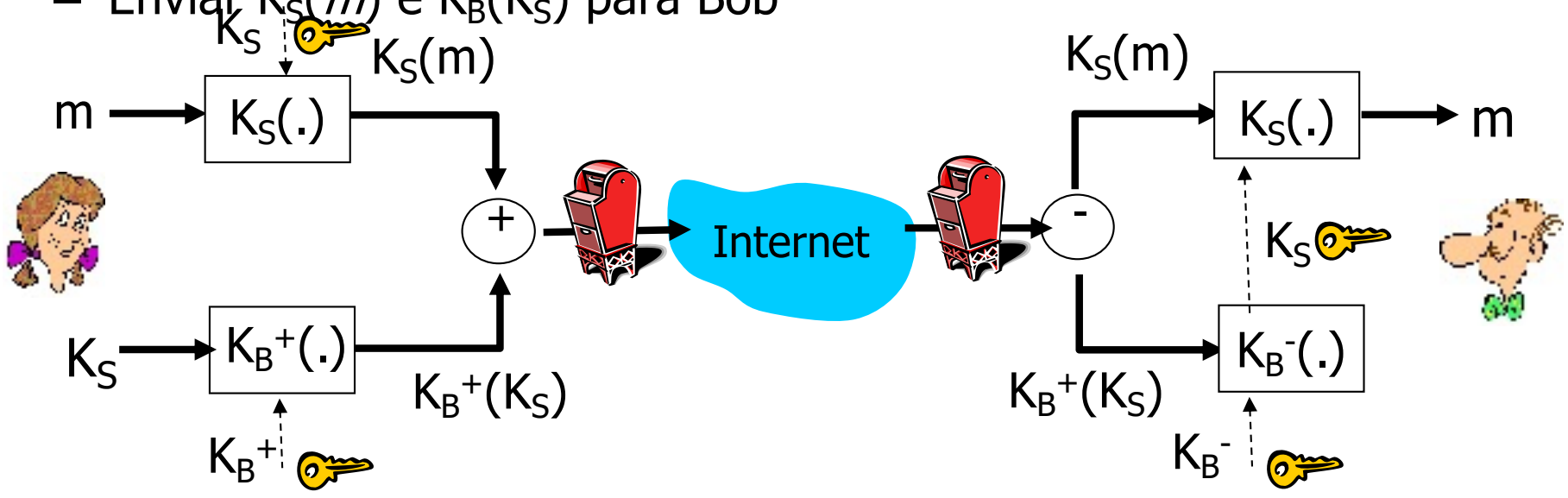
Lembrete: uma camada oferece serviços para as camadas superiores

Instituto de computação

Camada de Aplicação

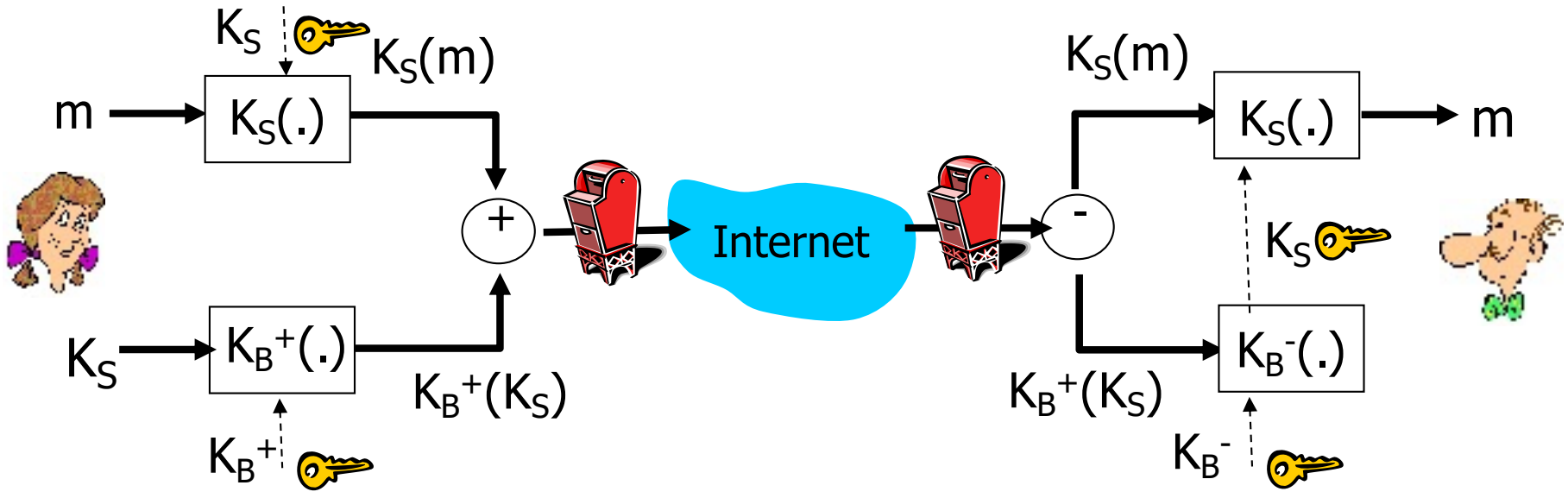
Email Seguro

- 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



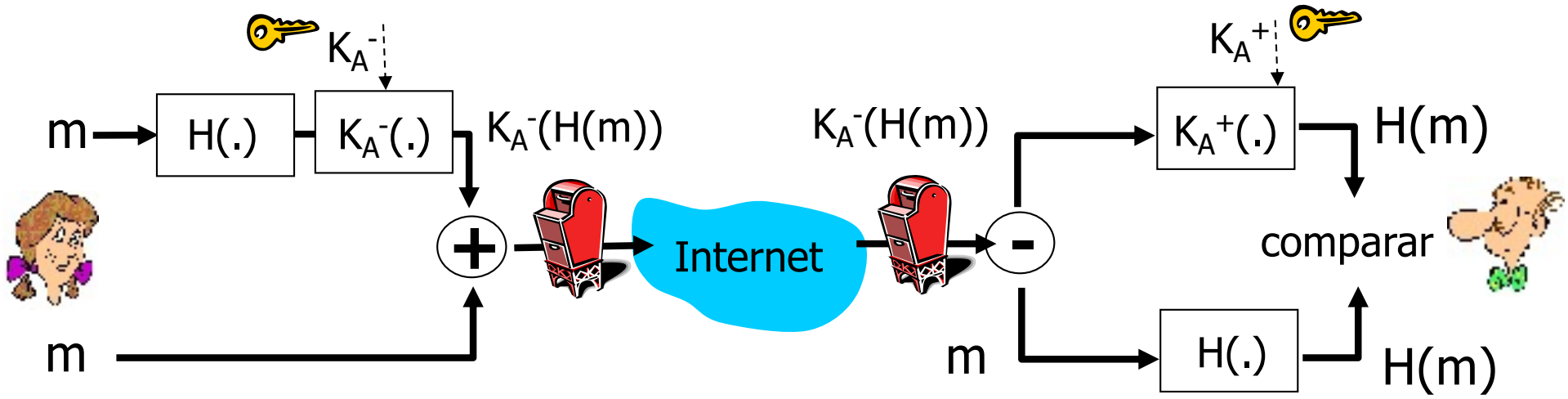
Email Seguro

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



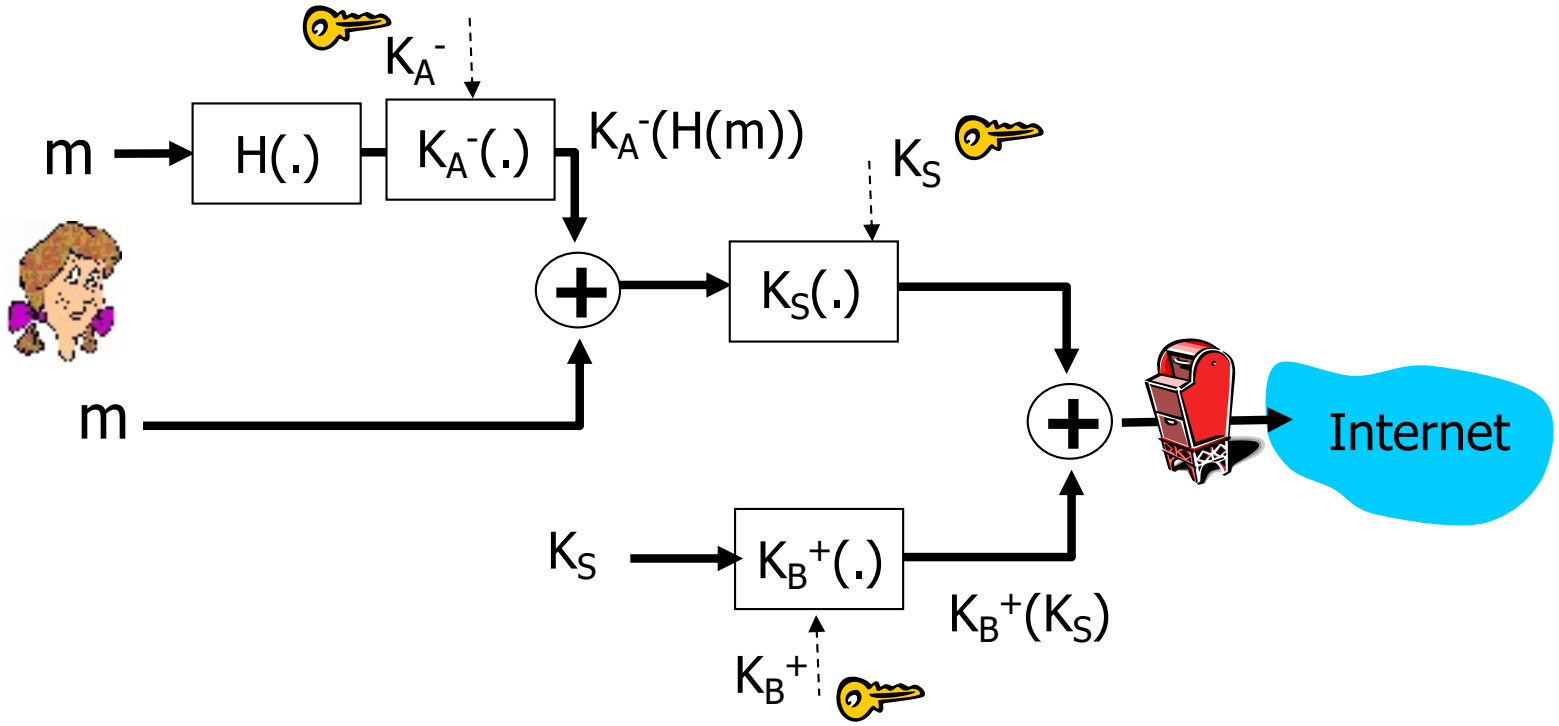
Email Seguro

- 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

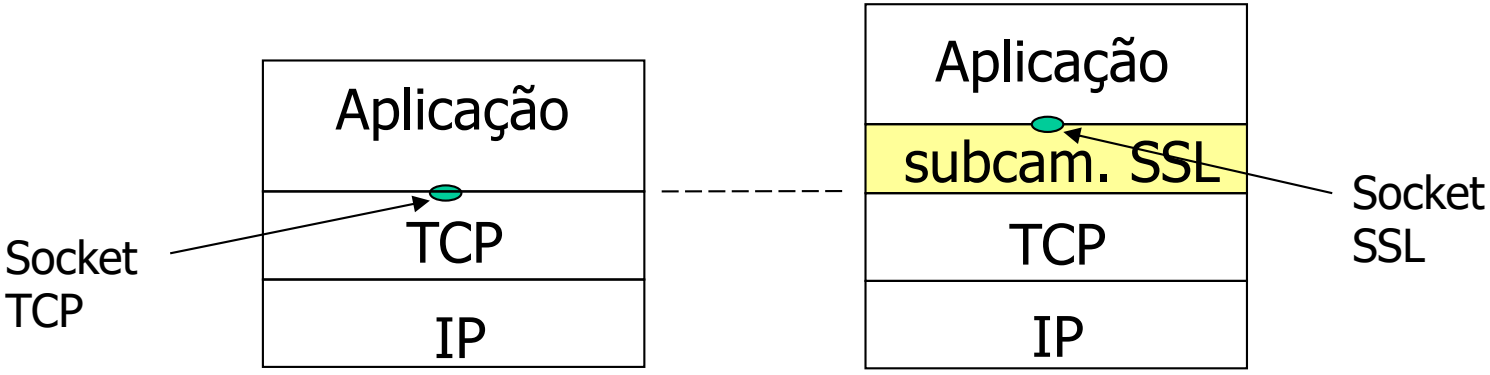
```
---BEGIN PGP SIGNED MESSAGE---
Hash: SHA1

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)

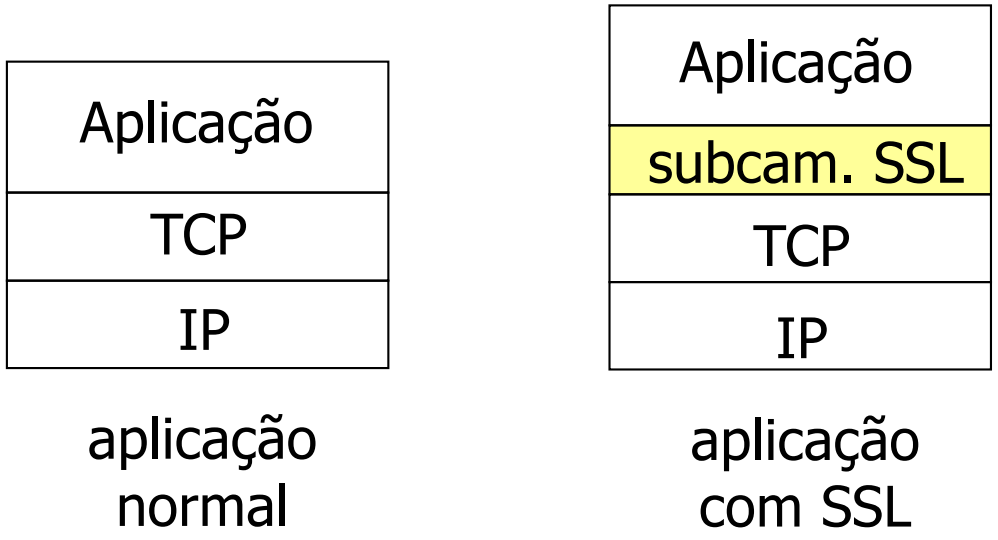
Instituto de computação

- Provê
  - Confidencialidade
  - Integridade
  - Autenticação

SSL e TCP/IP

Instituto de computação

- 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

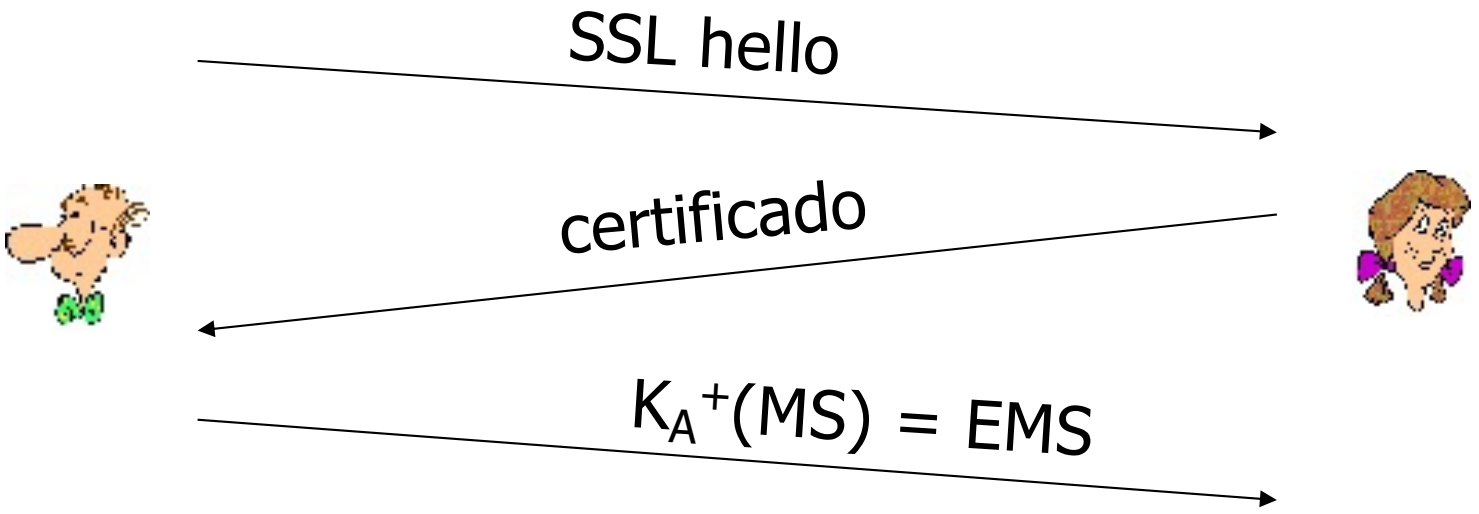
Instituto de computação

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

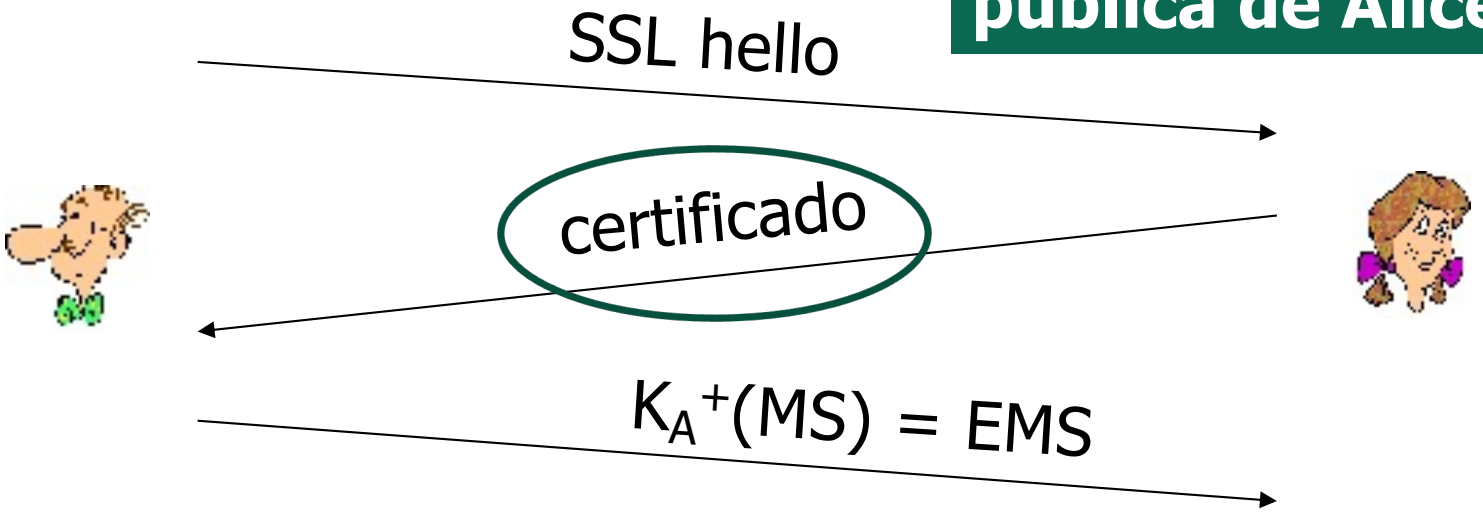
- Primeiro passo: estabelecer uma conexão TCP
  - *Three way handshake* "normal"
- Próximos passos: autenticar e distribuir uma chave secreta
  - MS = segredo mestre
  - EMS = segredo mestre cifrado



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

SSL *hello*

Alice usa sua chave privada para obter o segredo!

certificado

$K_A^+(MS) = EMS$

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

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.
- IPSec

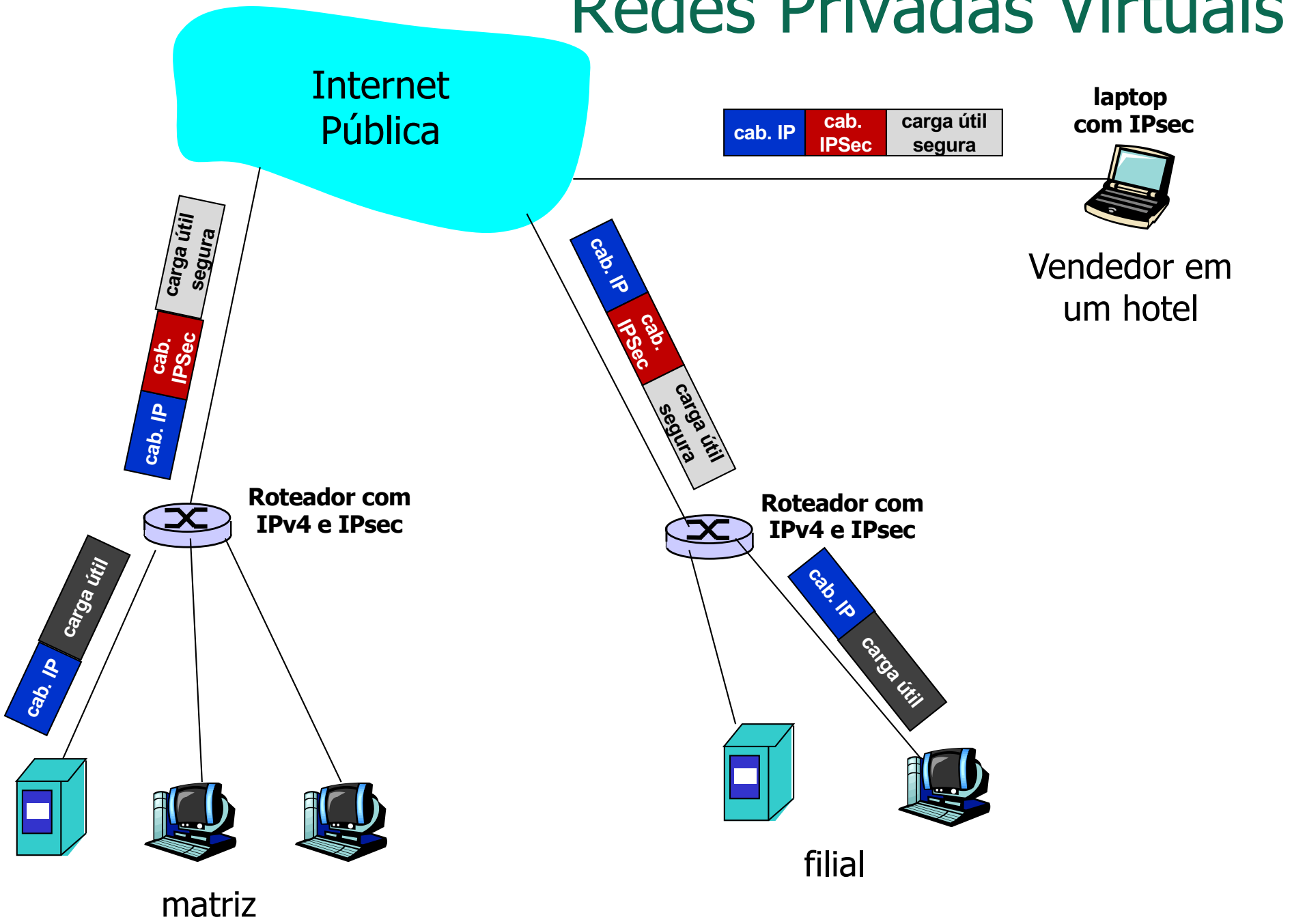
Redes Privadas Virtuais

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

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

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

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

Exemplo de Uso

- Eduroam é um serviço de acesso sem fio seguro desenvolvido para a comunidade internacional de educação e pesquisa.
- Permite que estudantes, pesquisadores e a equipe de instituições participantes obtenham conectividade à Internet, através de conexão sem fio, dentro de seus campi e quando visitam as instituições parceiras.

Funcionamento do Eduroam

- Infraestrutura para implantação do serviço:
  - Estrutura hierárquica de servidores de autenticação RADIUS - *Remote Authentication Dial In User Service* (Padrão IETF);
  - Infraestrutura de pontos de acesso sem fio IEEE 802.11 com suporte a 802.1x e 802.11i.
  - Base de dados LDAP (*Lightweight Directory Access Protocol*) com informações de usuários de cada instituição;
    - **Integração com base de dados da federação CAFe**

RADIUS

Instituto de computação

- Remote Authentication Dial In User Service
  - Padrão IETF – RFC 2865
  - Utiliza UDP porta 1812
- Protocolo AAA (*authentication, authorization and accounting*)
  - Realiza autenticação, controle de acesso e auditoria (*accounting*) a *Network Access Server*, que funciona como cliente RADIUS.

RADIUS

Instituto de computação

- Servidores RADIUS são responsáveis por:
  - Receber solicitações de conexão;
  - Autenticar usuários; e
  - Retornar todas as informações de configuração necessárias para o cliente (NAS) prover conectividade a um usuário.
- Pode atuar como um cliente *proxy* de outros servidores de autenticação.

802.1X

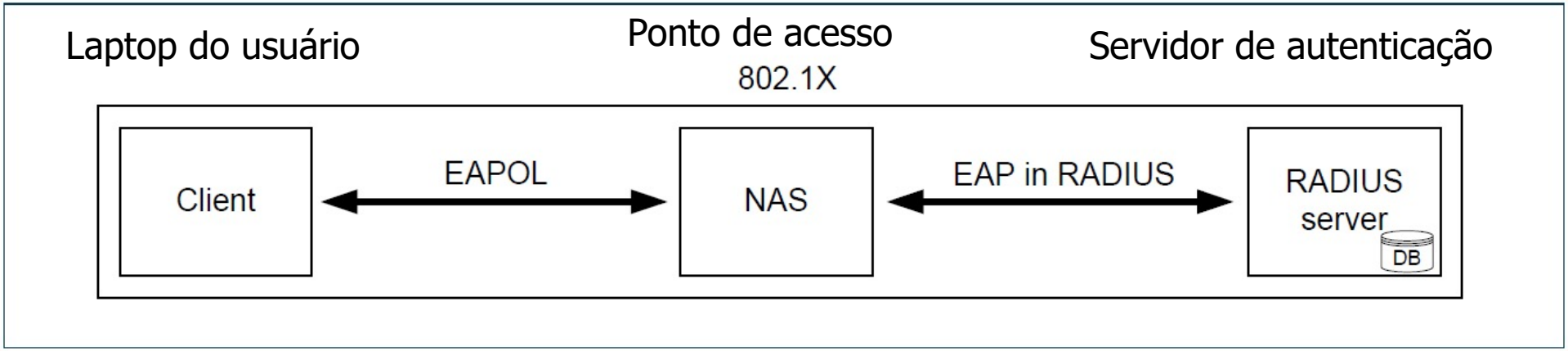
- Port-Based Network Access Control
  - Padrão IEEE para controle de acesso à rede baseado em porta;
- Elementos envolvidos em um processo de autenticação 802.1X:
  - Suplicante – software cliente que solicita acesso através de uma porta;
  - Autenticador – dispositivo de acesso à rede ou um Servidor de Acesso à Rede (NAS)
  - Servidor de Autenticação (AS) – pode ser o servidor RADIUS.

802.1X

Instituto de computação

- Especifica o uso do EAP (*Extensible Authentication Protocol*)
  - Padrão IETF – RFC 3748
  - Framework de autenticação que permite uma variedade de métodos de autenticação
    - WPA2/AES
    - EAP-TTLS/PAP
    - PEAP/MSCHAPv2
  - Funciona diretamente sobre a camada de enlace (PPP ou IEEE 802)

RADIUS + 802.1X



EAP – Extensible Authentication Protocol
EAPOL – EAP over LANs

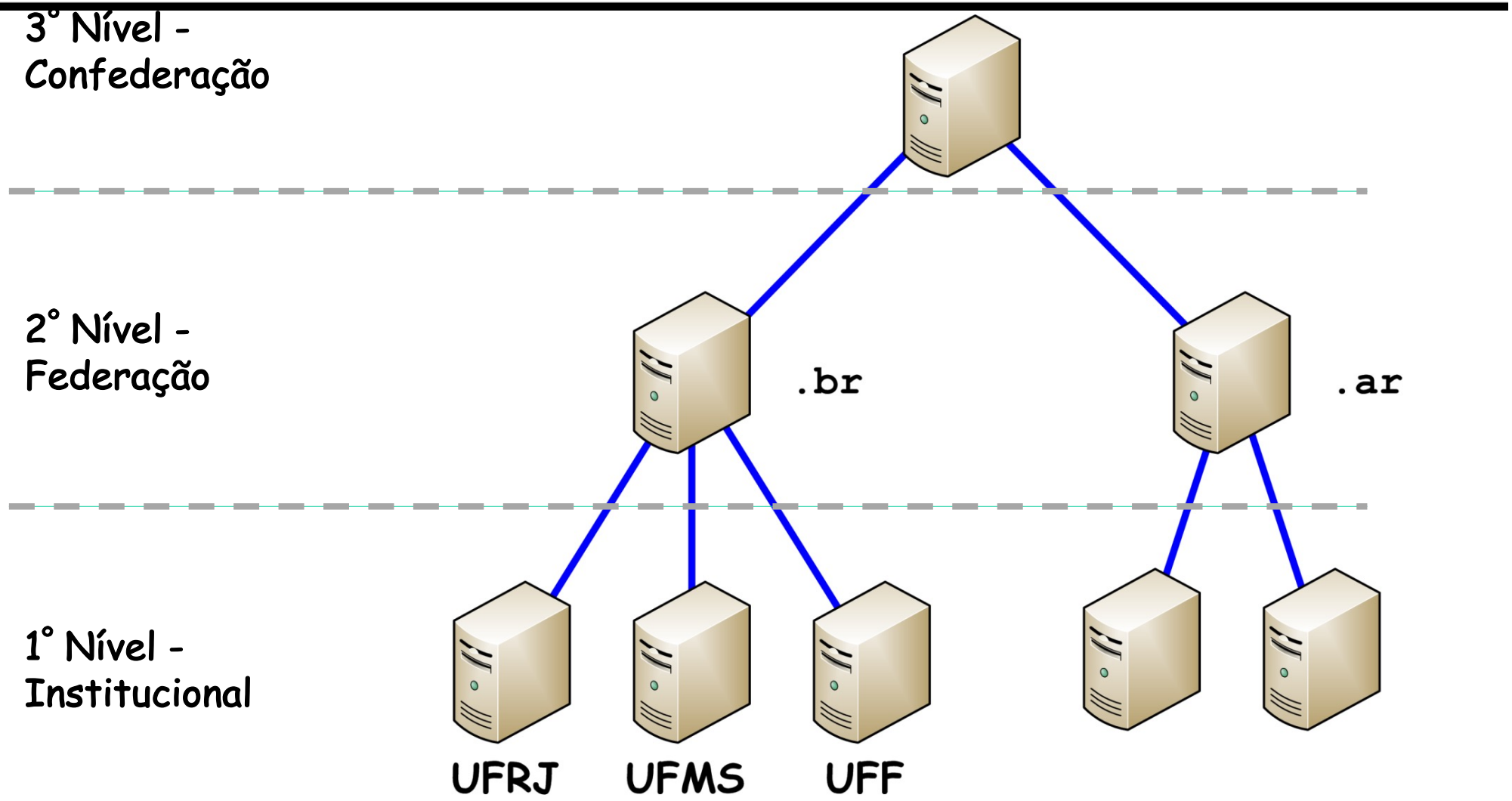
EAP-TTLS/PAP
PEAP-MSCHAPv2

Estrutura Hierárquica

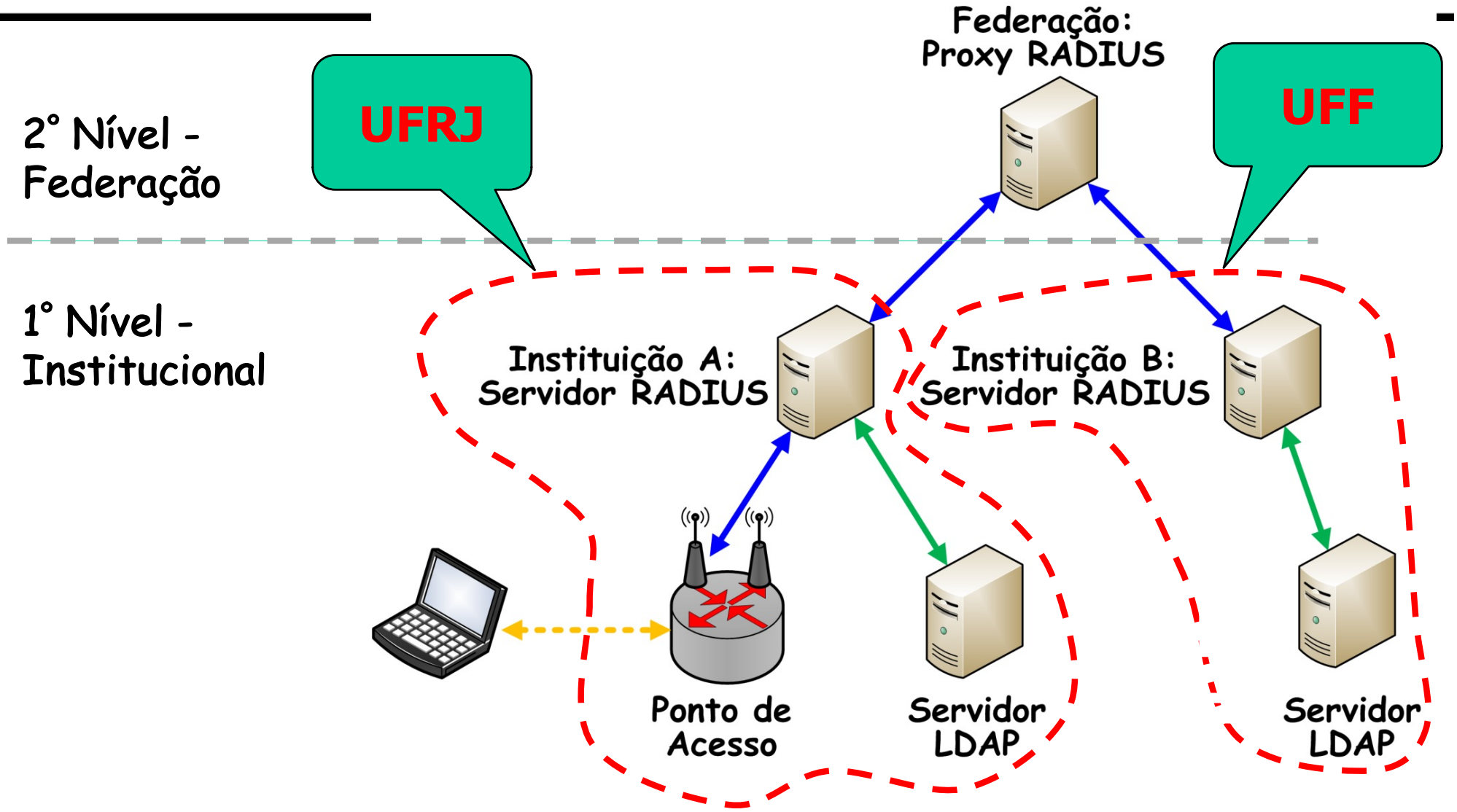
Instituto de computação

- Estrutura de servidores RADIUS em três níveis:
  - Confederação
  - Federação (país)
  - Instituição
- Identificação baseada em domínio
  - Ex.: **`usuario@dominio`**

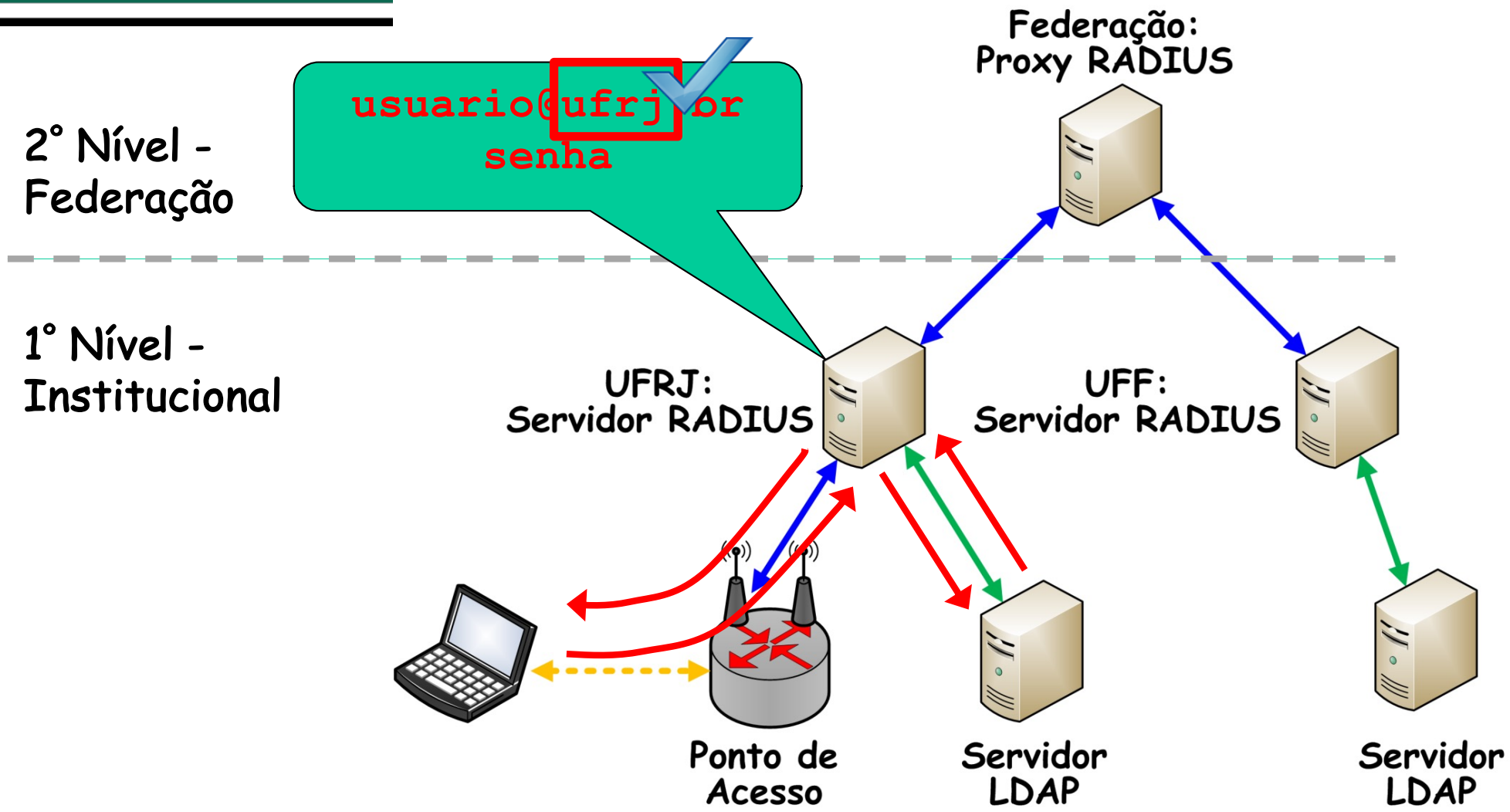
Estrutura Hierárquica



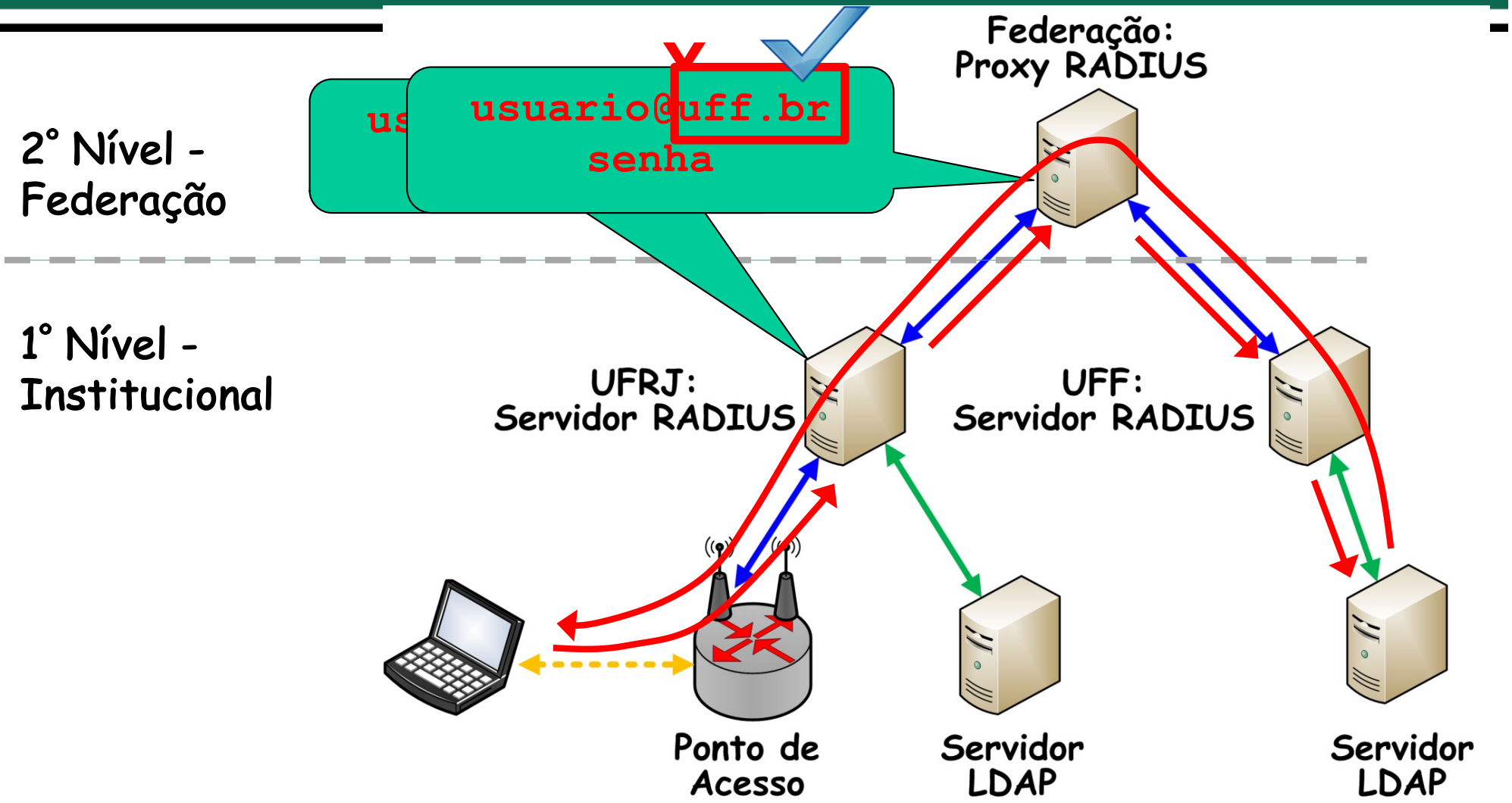
Infraestrutura do Provedor de Acesso



Processo de Autenticação Local



Processo de Autenticação Remota



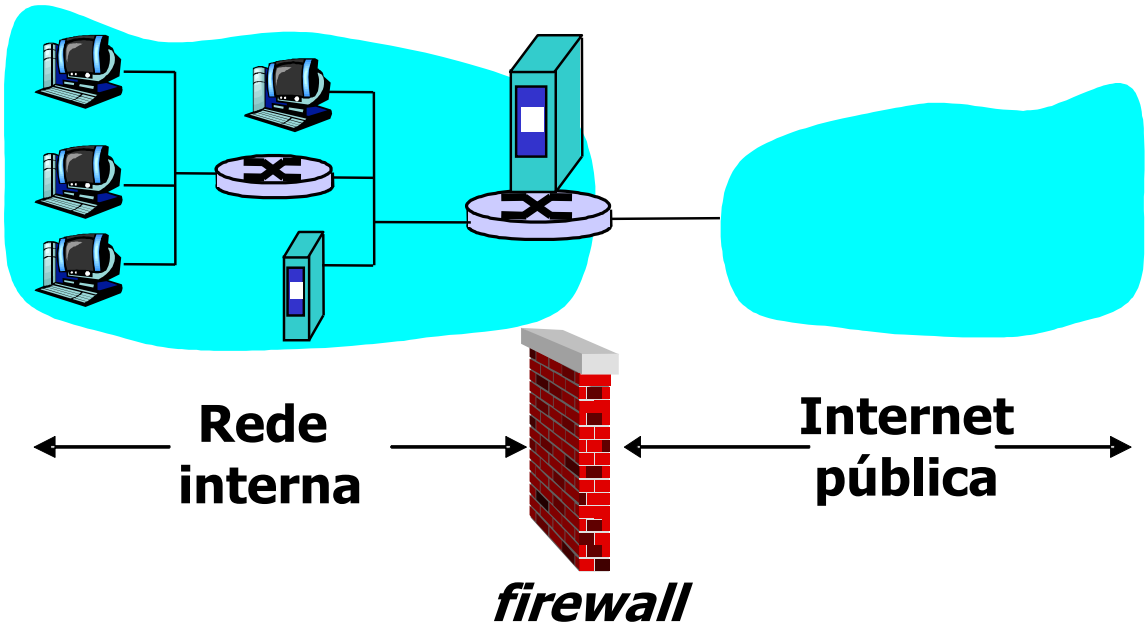
Instituto de Computação

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

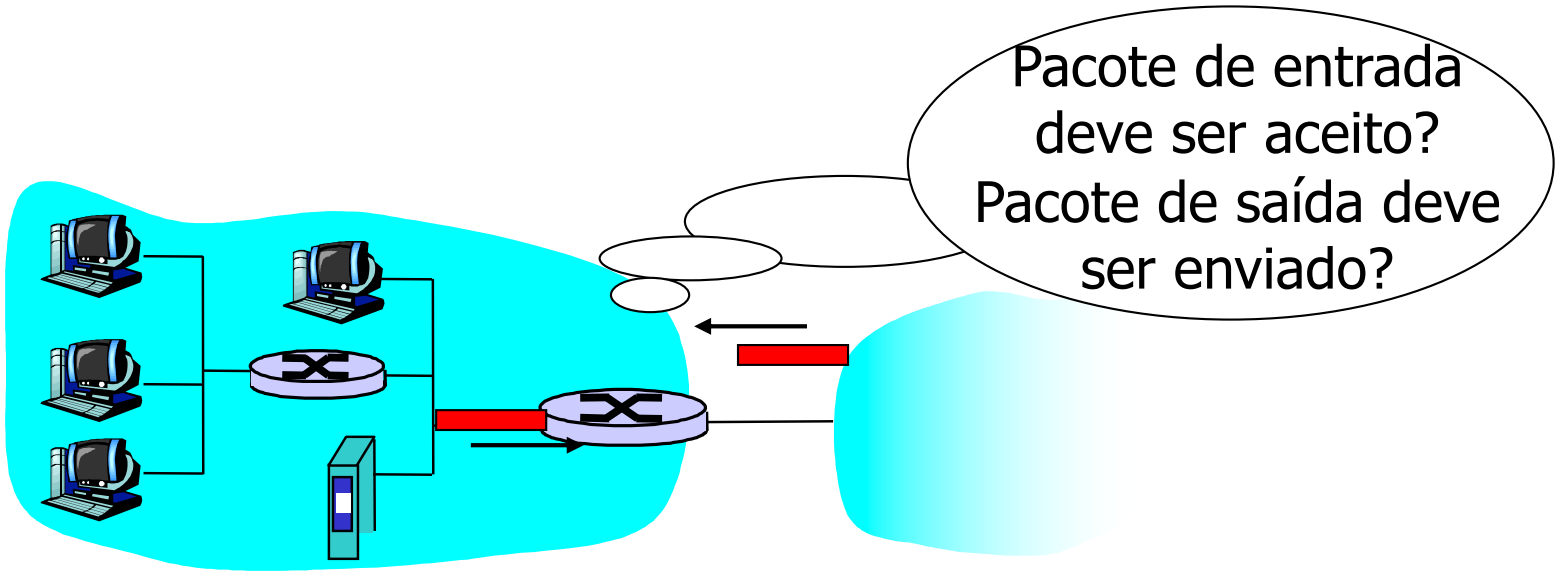
Firewalls

Instituto de computação

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

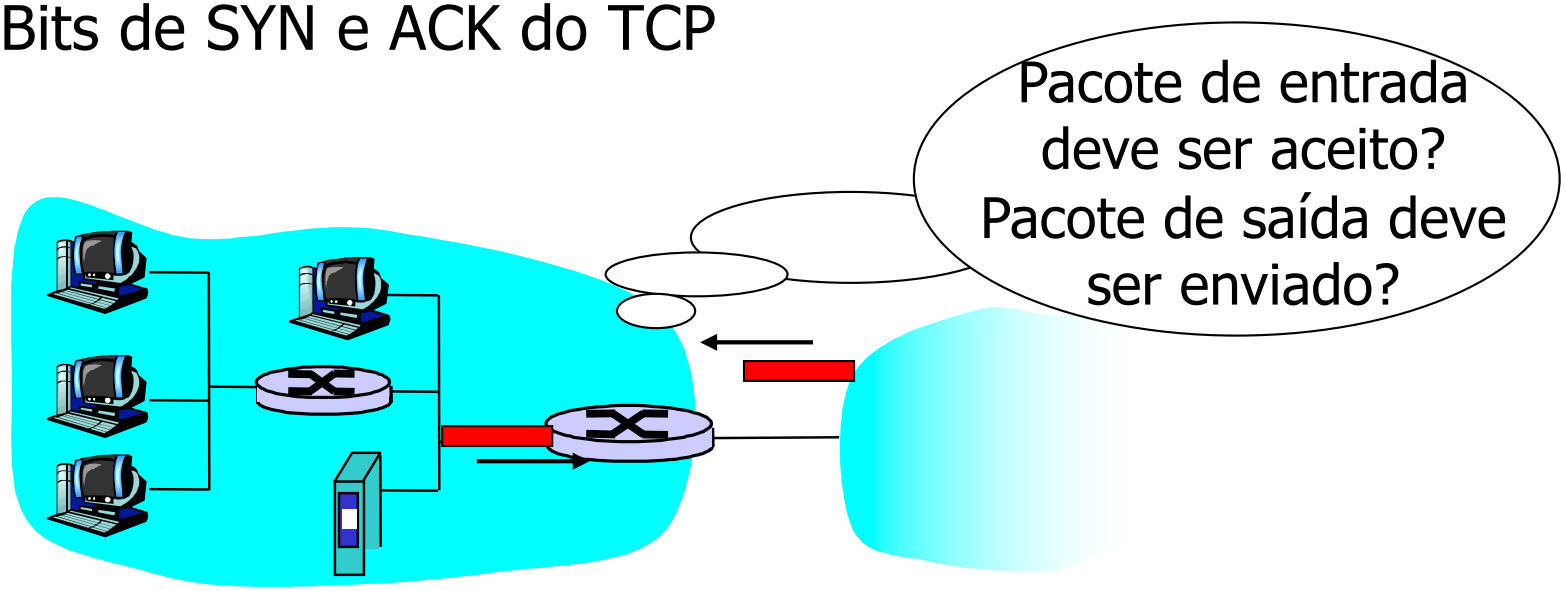
Filtragem de Pacotes Sem Estado

- 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

Instituto de computação

Departamento de Ciência da Computação
Instituto de Computação
Universidade Federal Fluminense

Segurança de Redes

Igor Monteiro Moraes

Redes de Computadores II