Capítulo 8: Segurança em Redes

Objetivos do capítulo:

- Compreender os princípios de segurança em redes:
  - criptografia e os seus *diversos* usos além da "privacidade" (sigilo)
  - autenticação
  - integridade das mensagens
  - distribuição de chaves
- Segurança na prática:
  - Firewalls
  - segurança nas camadas de aplicação, transporte, rede e enlace

Capítulo 8 roteiro

8.1 O que é segurança em redes?

8.2 Princípios de criptografia

8.3 Autenticação

8.4 Integridade

8.5 Distribuição de chaves e certificação

8.6 Controle de acesso: *firewalls*

8.7 Ataques e contra medidas

8.8 Segurança em diversas camadas

O que é Segurança de Redes?

Privacidade (Sigilo): apenas o transmissor e o receptor desejado devem "entender" o conteúdo da mensagem

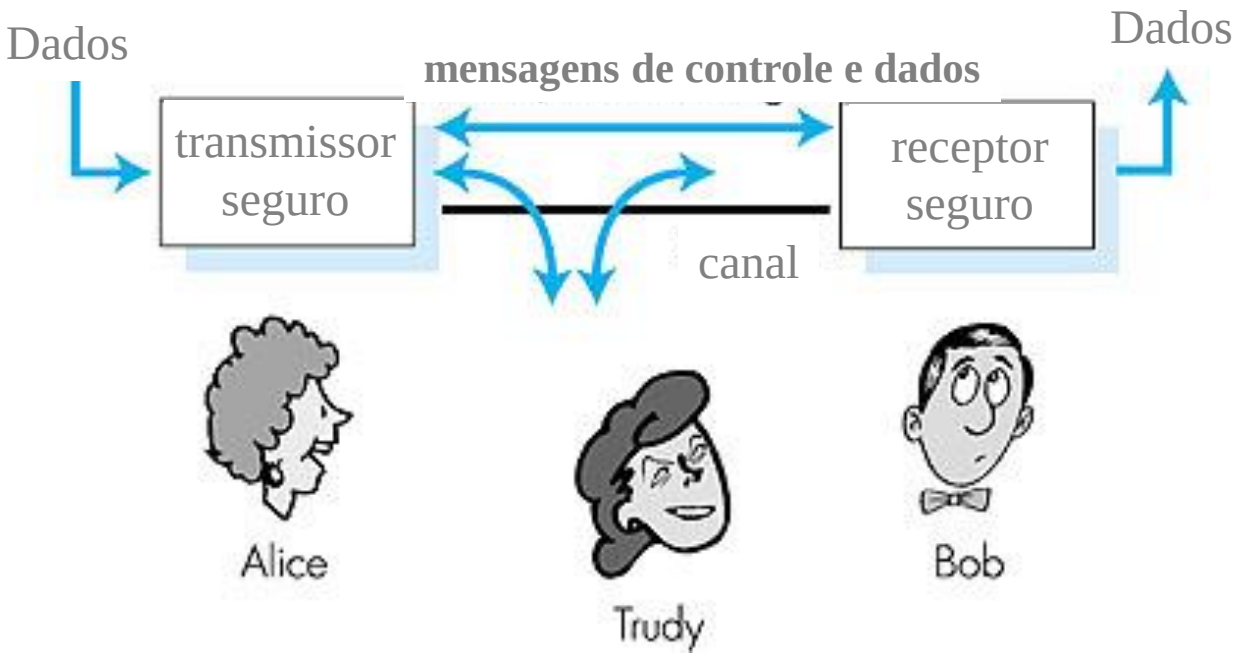
- transmissor cifra (codifica) msg
- receptor decifra (decodifica) msg

Autenticação: transmissor e receptor querem confirmar a identidade um do outro

Integridade da Mensagem: transmissor e receptor querem garantir que a mensagem não seja alterada (em trânsito ou após) sem que isto seja detectado

Acesso e Disponibilidade: os serviços devem estar acessíveis e disponíveis para os usuários

Amigos e Inimigos: Alice, Bob e Trudy



- bem conhecidos no mundo de segurança de redes
- Bob e Alice (amantes!) querem se comunicar de modo "seguro"
- Trudy, a "intrusa" pode interceptar, apagar e/ou acrescentar mensagens

Quem podem ser Bob e Alice?

- ... bem, Bobs e Alices *reais*!
- Browser/servidor web para transações eletrônicas (ex., compras on-line)
- cliente/servidor home banking
- servidores DNS
- roteadores trocando atualizações de tabelas de roteamento
- outros exemplos?

Há muitos vilões por aí!

P: O que um vilão pode fazer?

R: um monte de coisas!

- *grampo:* interceptação de mensagens
- *inserir* ativamente mensagens na conexão
- *falsidade ideológica:* pode imitar/falsificar endereço de origem de um pacote (ou qualquer campo de um pacote)
- *seqüestro:* assumir conexão em andamento removendo o transmissor ou o receptor, colocando-se no lugar
- *negação de serviço:* impede que o serviço seja usado por outros (ex. sobrecarregando os recursos)

mais sobre isto posteriormente...

Capítulo 8 roteiro

8.1 O que é segurança em redes?

8.2 Princípios de criptografia

8.3 Autenticação

8.4 Integridade

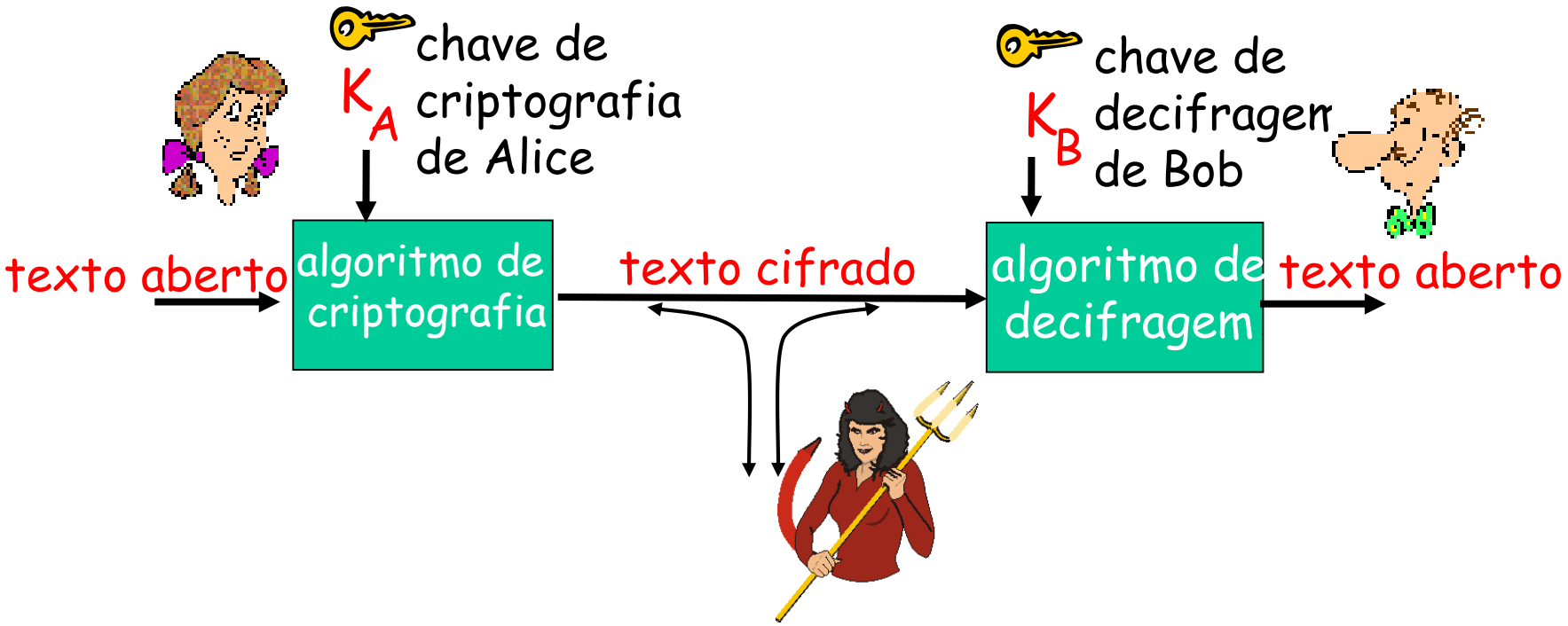
8.5 Distribuição de chaves e certificação

8.6 Controle de acesso: *firewalls*

8.7 Ataques e contra medidas

8.8 Segurança em diversas camadas

A linguagem da criptografia



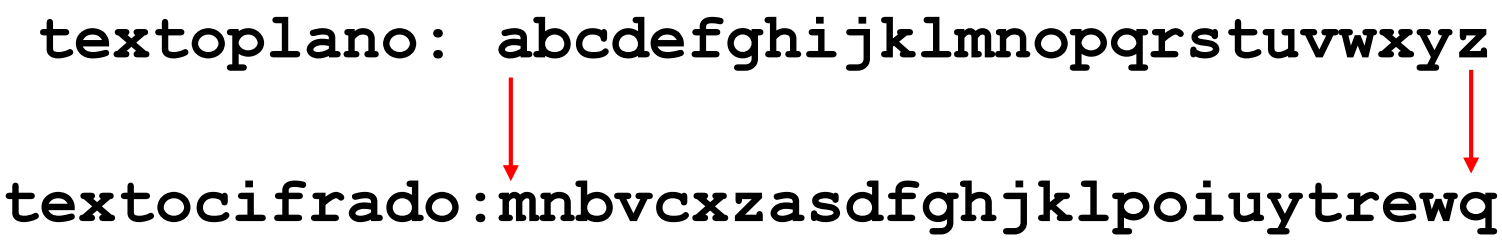
criptografia de chave simétrica: as chaves do transmissor e do receptor são idênticas

criptografia de chave pública: cifra com chave *pública*, decifra com chave *secreta* (privada)

Criptografia de chave simétrica

código de substituição: substitui um caractere por outro

- código monoalfabético: substitui uma letra por outra



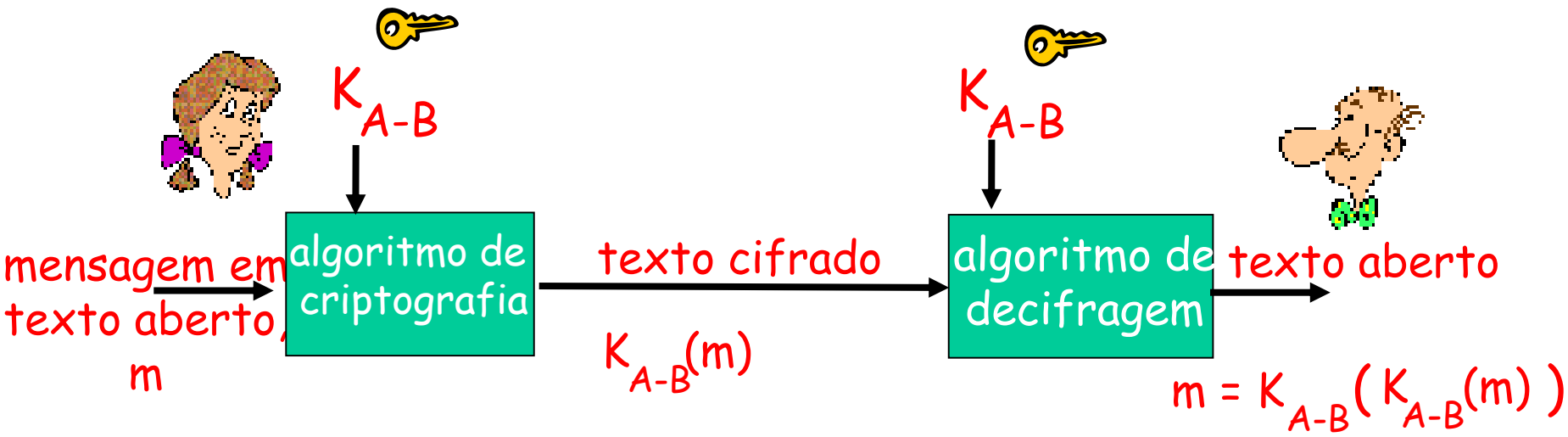
Ex: **Texto aberto: bob. eu te amo. alice**

Texto cifrado: nkn. cy uc mhk. mgsbc

P: Qual a dificuldade em quebrar este código simples?:

- força bruta (qual é a dificuldade?)
- outra estratégia?

Criptografia de chave simétrica



criptografia de chave simétrica: Bob e Alice compartilham a mesma chave (simétrica): K_{A-B}

- ex., a chave é um padrão de substituição conhecido num código de substituição monoalfabético
- P: como Bob e Alice concordam com um valor de chave?

Criptografia de chave simétrica: DES

DES: *Data Encryption Standard*

- padrão americano de cifragem [NIST 1993]
- chave simétrica de 56 bits, entrada do texto em palavras de 64 bits
- Quão seguro é o DES?
  - Desafio DES: frase criptografada com chave de 56 bits ("Strong cryptography makes the world a safer place") foi decifrada (com força bruta) em 4 meses
  - não é conhecida nenhuma porta secreta de decifragem
- tornando o DES mais seguro
  - use três chaves seqüencialmente (3-DES) para cada dado (usado no PPP [RFC 2420]).
  - use cifragem por encadeamento de blocos

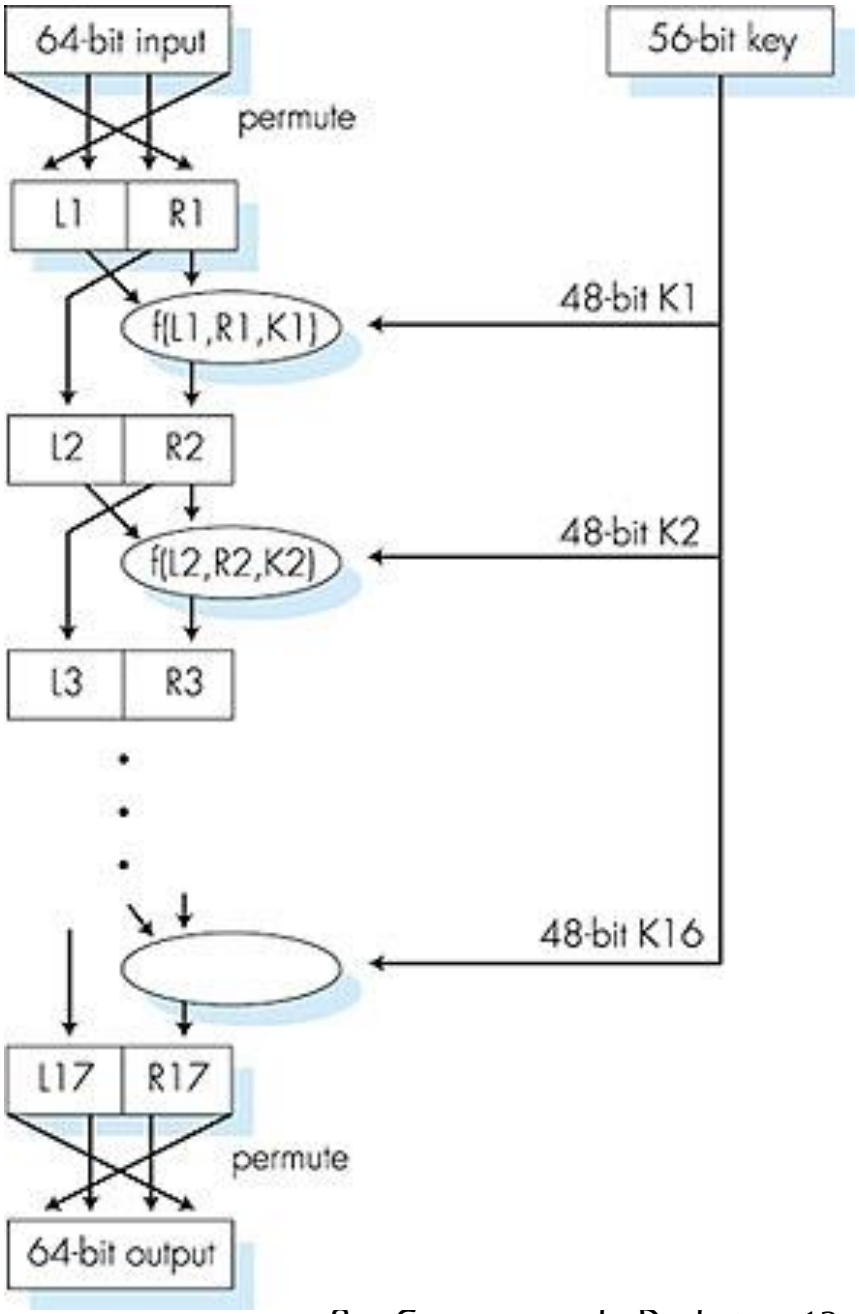
Criptografia de chave simétrica: DES

operação do DES

permutação inicial

16 rodadas idênticas de aplicação de uma função, cada uma usando 48 bits diferentes da chave

permutação final



AES - Advanced Encription Standard

- Novo (Nov. 2001) algoritmo de chave simétrica padronizada pelo NIST, substitui o DES
- processa dados em blocos de 128 bits
- Em uso desde maio de 2002.
- chaves de 128, 192 ou 256 bits
- decifragem em força bruta (tentar cada chave) que leva 1 seg no DES, levaria 149 trilhões de anos no AES

Criptografia de chave pública

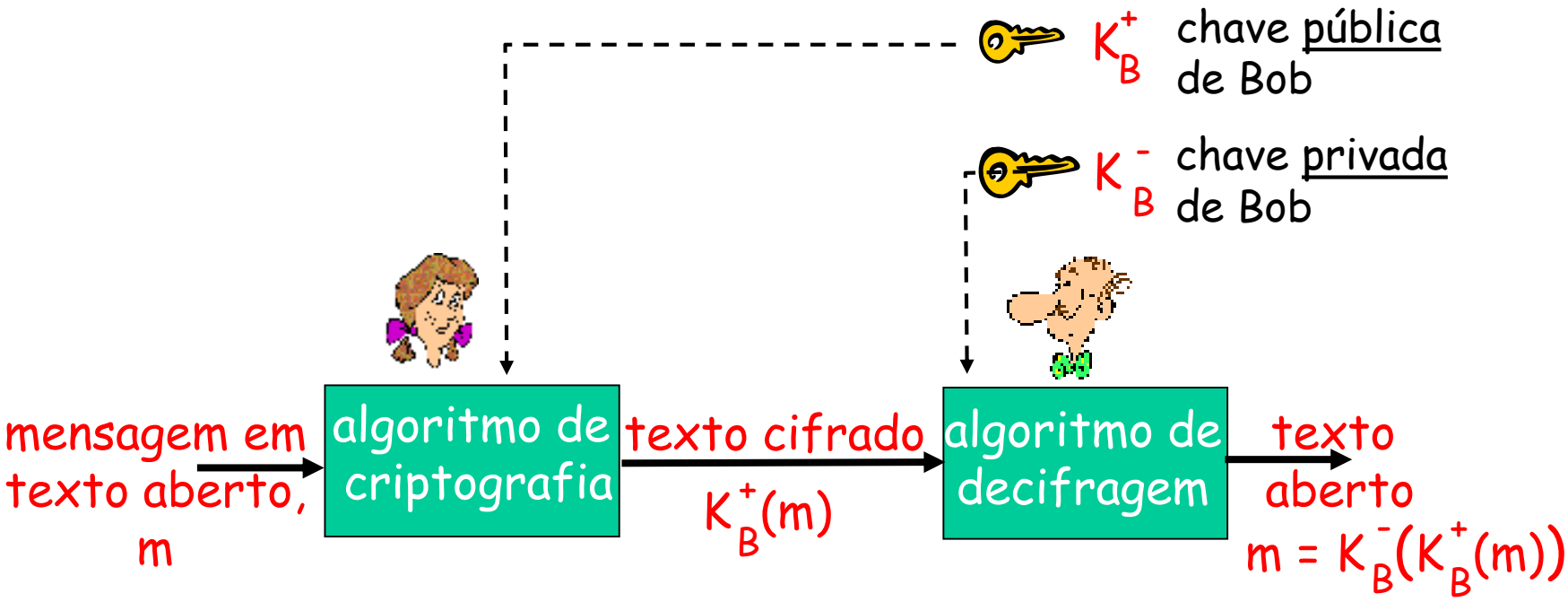
criptografia de chave *simétrica*

- requer que o transmissor e receptor compartilhem a chave secreta
- P: como combinar que chave utilizar (em particular se nunca tiverem se encontrado)?

criptografia de chave *pública*

- abordagem radicalmente diferente [Diffie-Hellman76, RSA78]
- transmissor e receptor *não* compartilham uma chave secreta
- a chave de cifragem é *pública* (conhecida por *todos*)
- a chave de decifragem é privada (conhecida apenas pelo receptor)

Criptografia de chave pública



Algoritmos de cifragem de chave pública

Requisitos:

1. necessita $K_B^+(\ )$ e $K_B^-(\ )$ de modo que

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

2. dada a chave pública K_B^+, deve ser impossível calcular a chave privada K_B^-

RSA: algoritmo de Rivest, Shamir e Adelson

RSA: Escolha das chaves

1. Escolha dois números primos grandes *p*, *q*. (ex., cada um com 1024 bits)

2. Calcule $n = pq$ e $z = (p-1)(q-1)$

3. Escolha *e* (com $e<n$) que não possua nenhum fator comum com z. (*e* e *z* são "primos entre si").

4. Escolha *d* de modo que *ed-1* seja divisível exatamente por *z* (em outras palavras: *ed* mod *z* = 1).

5. A chave *pública* é $\underbrace{(n,e)}_{K_B^+}$. A chave *privada* é $\underbrace{(n,d)}_{K_B^-}$.

RSA: Cifragem e decifragem

0. Dados (n,e) e (n,d) calculados anteriormente

1. Para cifrar o padrão de bits, m, calcule

   $c = m^e \bmod n$ (i.e., resto quando m^e é dividido por n)

2. Para decifrar o padrão de bits recebidos, c, calcule

   $m = c^d \bmod n$ (i.e., resto quando c^d é dividido por n)

Acontece uma mágica!

$$m = (m^e \bmod n)^d \bmod n$$

Exemplo RSA:

Bob escolhe *p=5, q=7*. Então *n=35* e *z=24*.

e=5 (então *e, z* são primos entre si).

d=29 (então *ed-1* é divisível exatamente por z.

cifra:

letra	m	m^e	$c = m^e \bmod n$
l	12	1524832	17

decifra:

c	c^d	$m = c^d \bmod n$	letra
17	481968572106750915091411825223072000	12	l

RSA: Porquê: $m = (m^e \bmod n)^d \bmod n$

Resultado da teoria dos números: Se p,q primos, $n = pq$, então

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

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

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

(usando o resultado acima da teoria dos números)

$$= m^1 \bmod n$$

(dado que escolhemos ed divisível por $(p-1)(q-1)$ com resto 1)

$$= m$$

RSA: outra propriedade importante

A propriedade seguinte será *muito* útil posteriormente:

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

use a chave pública antes, seguida pela chave privada

use a chave privada antes, seguida pela chave pública

O Resultado é o mesmo!

Capítulo 8 roteiro

8.1 O que é segurança em redes?

8.2 Princípios de criptografia

8.3 Autenticação

8.4 Integridade

8.5 Distribuição de chaves e certificação

8.6 Controle de acesso: *firewalls*

8.7 Ataques e contra medidas

8.8 Segurança em diversas camadas

Autenticação

Objetivo: Bob quer que Alice "prove" a sua identidade para ele

Protocolo ap1.0: Alice diz "Eu sou Alice"

Cenário de falha??

Autenticação

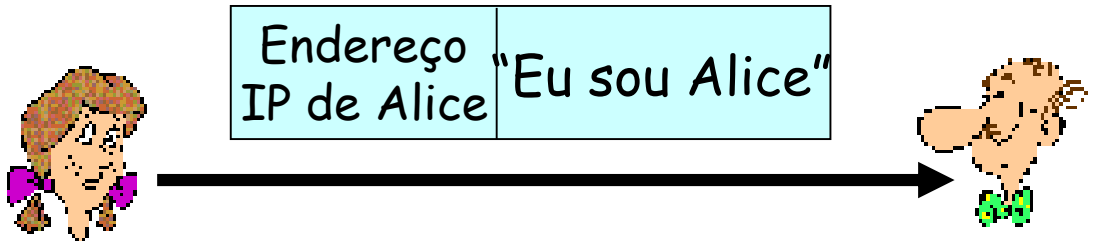
Objetivo: Bob quer que Alice "prove" a sua identidade para ele

Protocolo ap1.0: Alice diz "Eu sou Alice"

Numa rede,
Bob não "vê" Alice,
então Trudy
simplesmente se
declara como sendo
Alice

Autenticação: outra tentativa

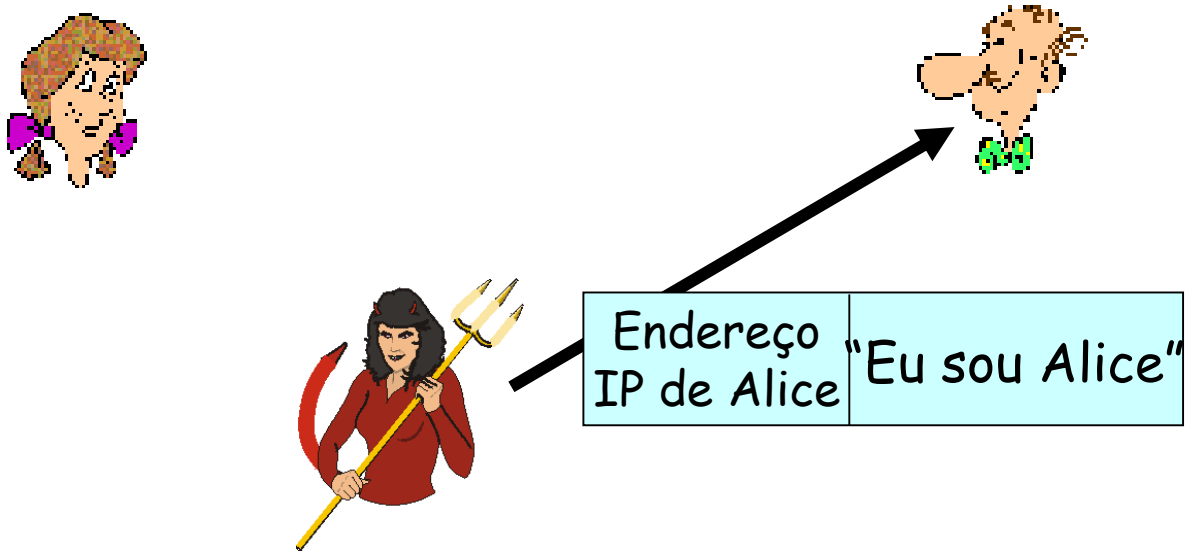
Protocolo ap2.0: Alice diz "Eu sou Alice" e envia junto o seu endereço IP como "prova".



Cenário de falha??

Autenticação: outra tentativa

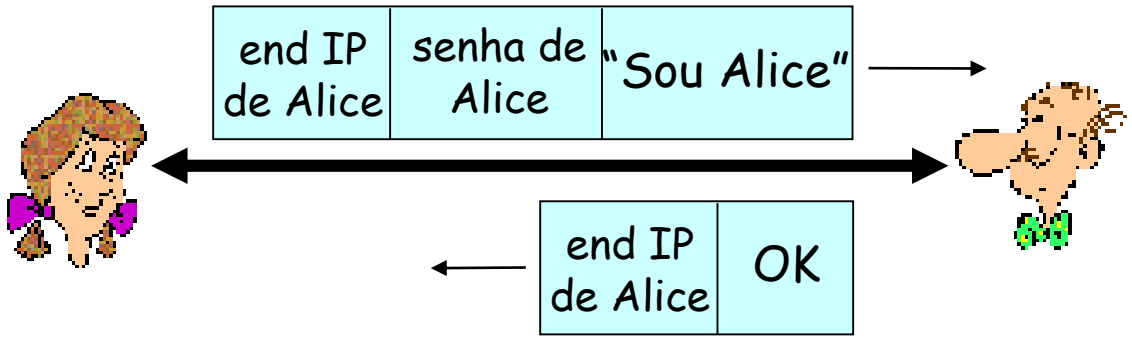
Protocolo ap2.0: Alice diz "Eu sou Alice" e envia junto o seu endereço IP como "prova".



Trudy pode criar um pacote "imitando" o endereço IP de Alice

Autenticação: outra tentativa

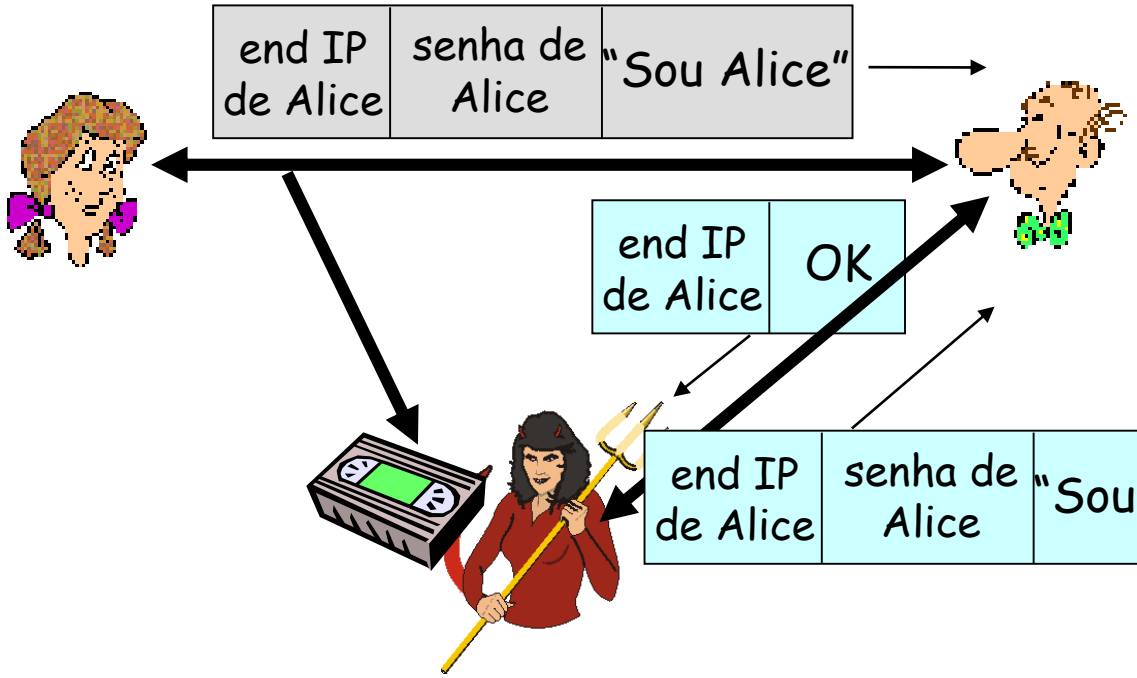
Protocolo ap3.0: Alice diz "Eu sou Alice" e envia a sua senha secreta como "prova".



Cenário de falha?

Autenticação: outra tentativa

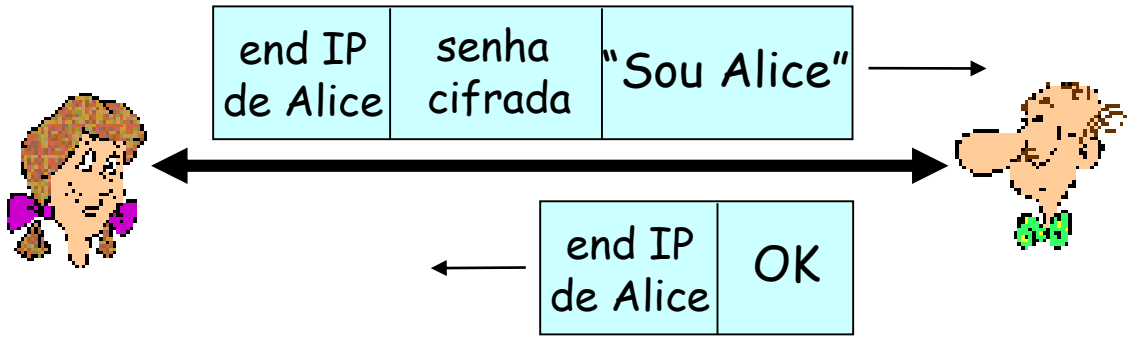
Protocolo ap3.0: Alice diz "Eu sou Alice" e envia a sua senha secreta como "prova".



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

Autenticação: ainda uma outra tentativa

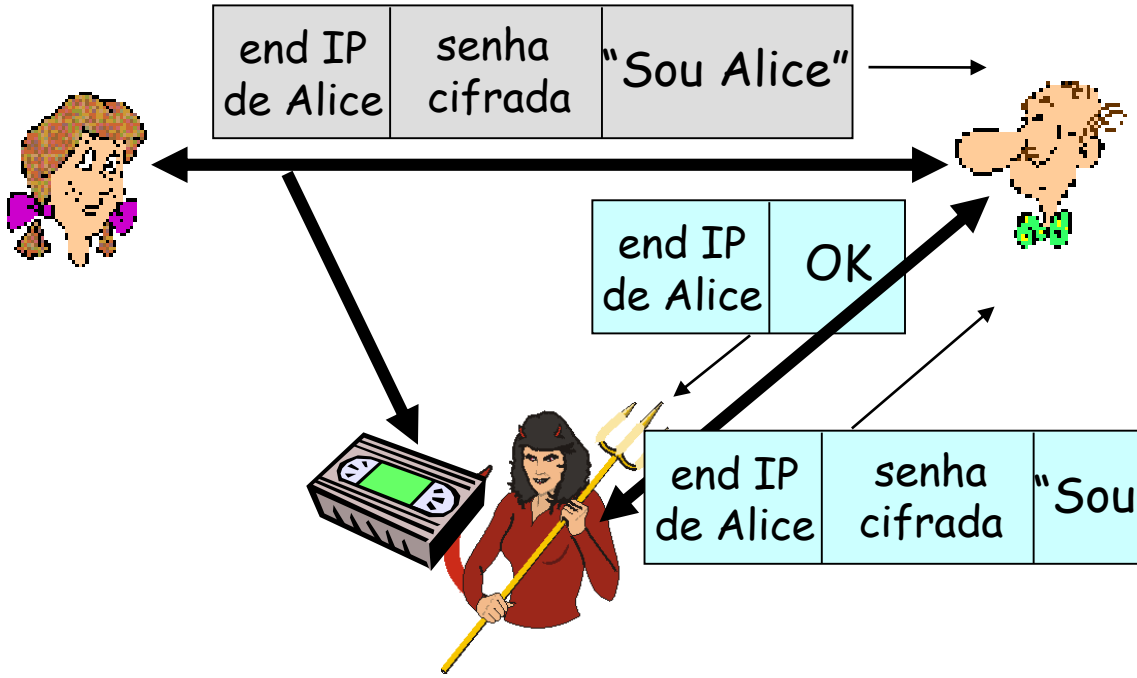
Protocolo ap3.1: Alice diz "Eu sou Alice" e envia a sua senha secreta *cifrada* como "prova".



Cenário de falha?

Autenticação: ainda uma outra tentativa

Protocolo ap3.1: Alice diz "Eu sou Alice" e envia a sua senha secreta *cifrada* como "prova".



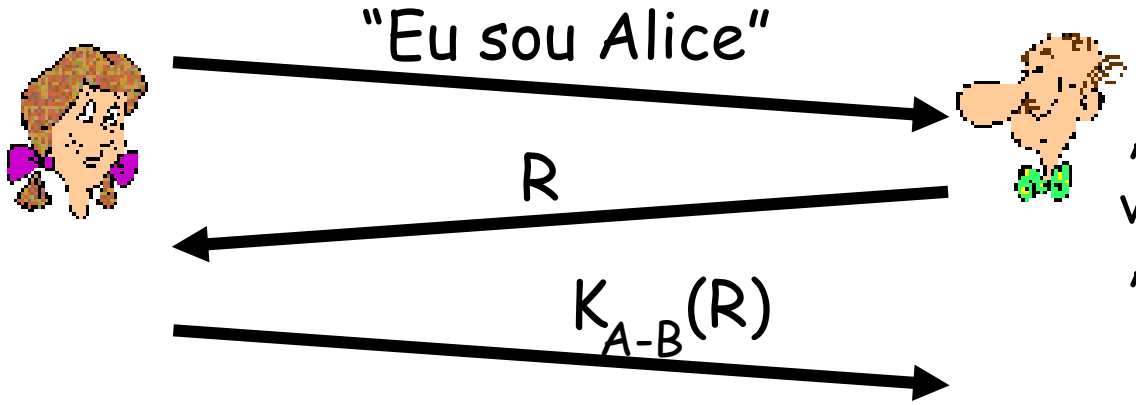
Gravar e reproduzir ainda funciona!

Autenticação: ainda uma outra tentativa

Objetivo: evitar ataque de reprodução (*playback*)

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

ap4.0: de modo a identificar Alice "ao vivo", Bob envia para Alice um nonce R, Alice deve retornar R, cifrado com a chave secreta compartilhada.



Alice está "ao vivo", e apenas Alice conhece chave para criptografar *nonce*, então deve ser Alice!

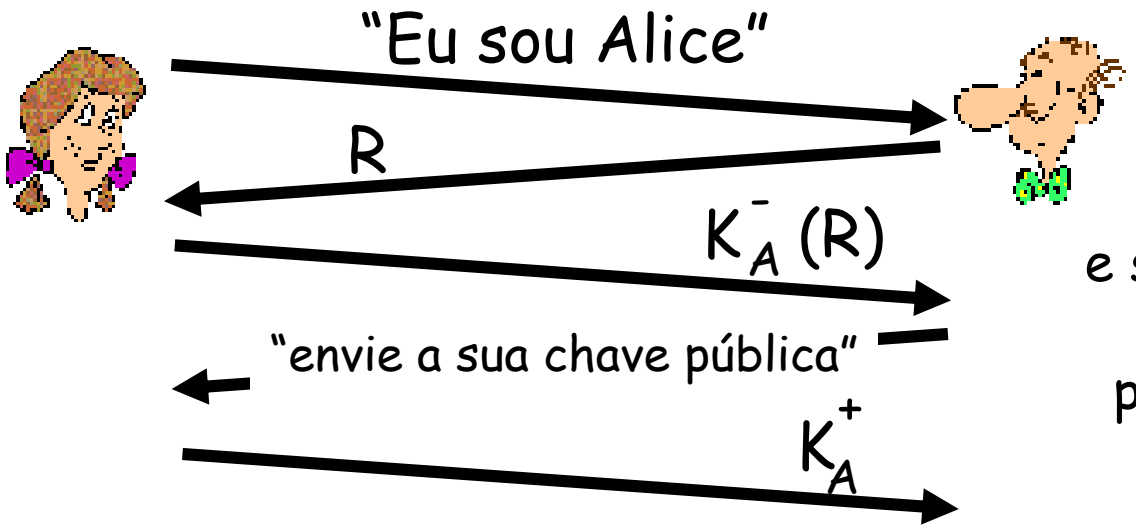
Falhas, desvantagens?

Autenticação: ap5.0

ap4.0 requer chave simétrica compartilhada

- podemos autenticar usando técnicas de chave pública?

ap5.0: use nonce, criptografia de chave pública



Bob calcula

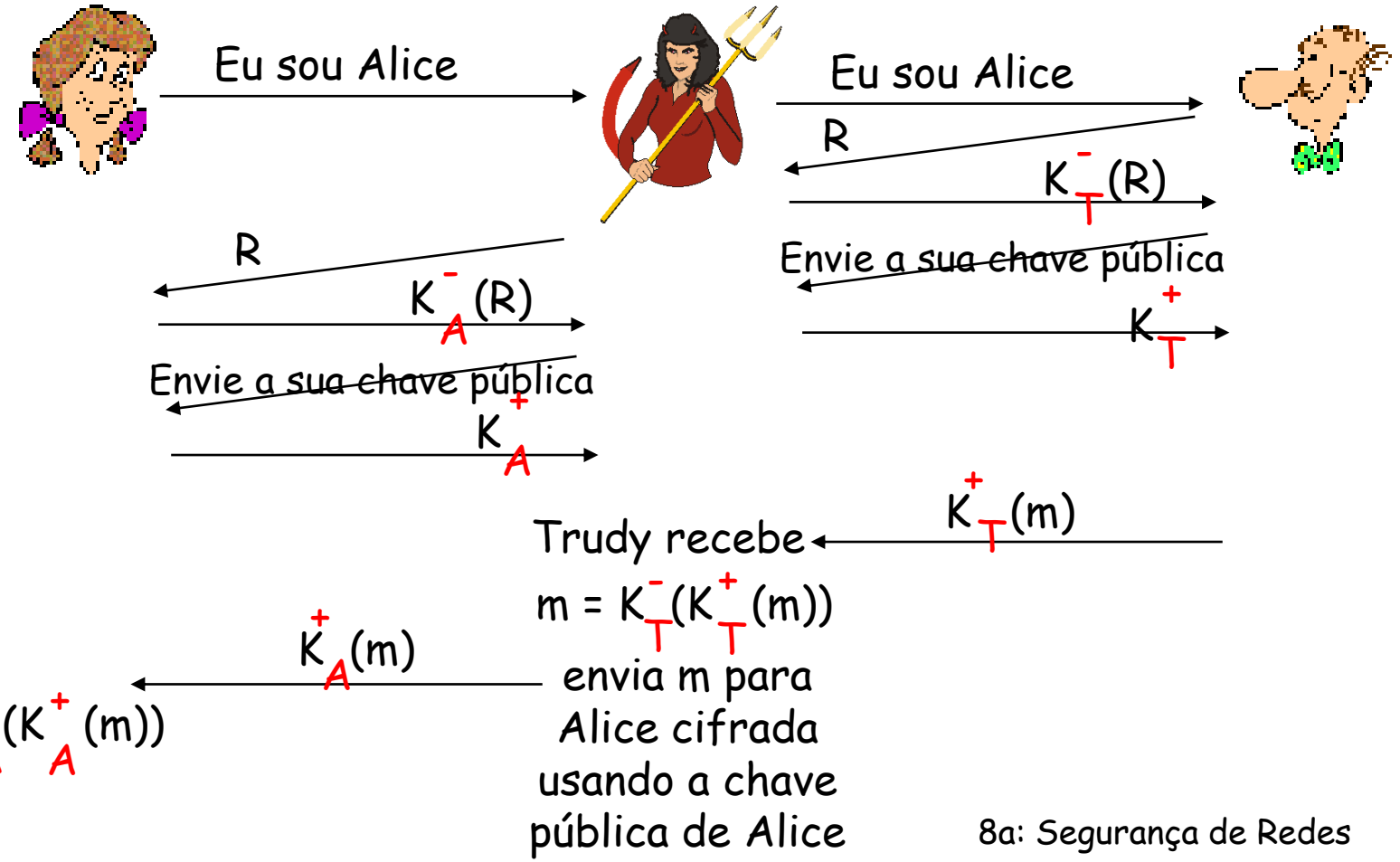
$K_A^+(K_A^-(R)) = R$

e sabe que apenas Alice poderia ter a chave privada, que cifrou R, de modo que

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

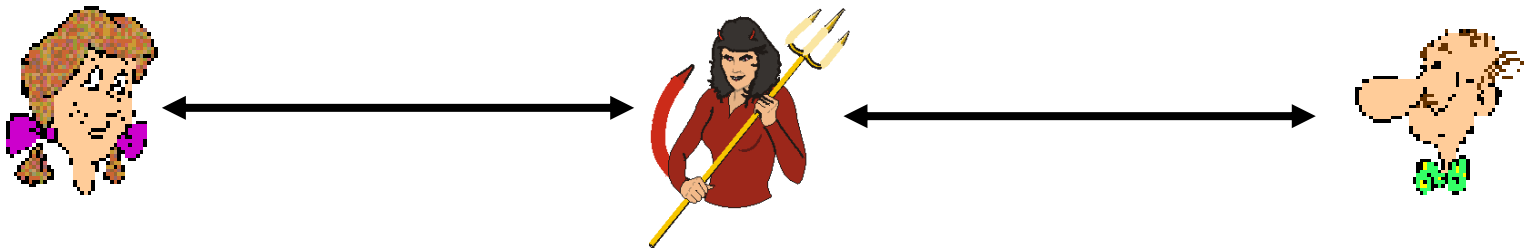
ap5.0: brecha de segurança

Ataque do homem (mulher) no meio: Trudy posa como sendo Alice (para Bob) e como sendo Bob (para Alice)



ap5.0: brecha de segurança

Ataque do homem (mulher) 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., portanto Bob, Alice podem se encontrar uma semana depois e lembrar da conversa)
- o problema é que Trudy também recebe todas as mensagens!

Capítulo 8 roteiro

8.1 O que é segurança em redes?
8.2 Princípios de criptografia
8.3 Autenticação
8.4 Integridade
8.5 Distribuição de chaves e certificação
8.6 Controle de acesso: *firewalls*
8.7 Ataques e contra medidas
8.8 Segurança em diversas camadas

Assinaturas Digitais

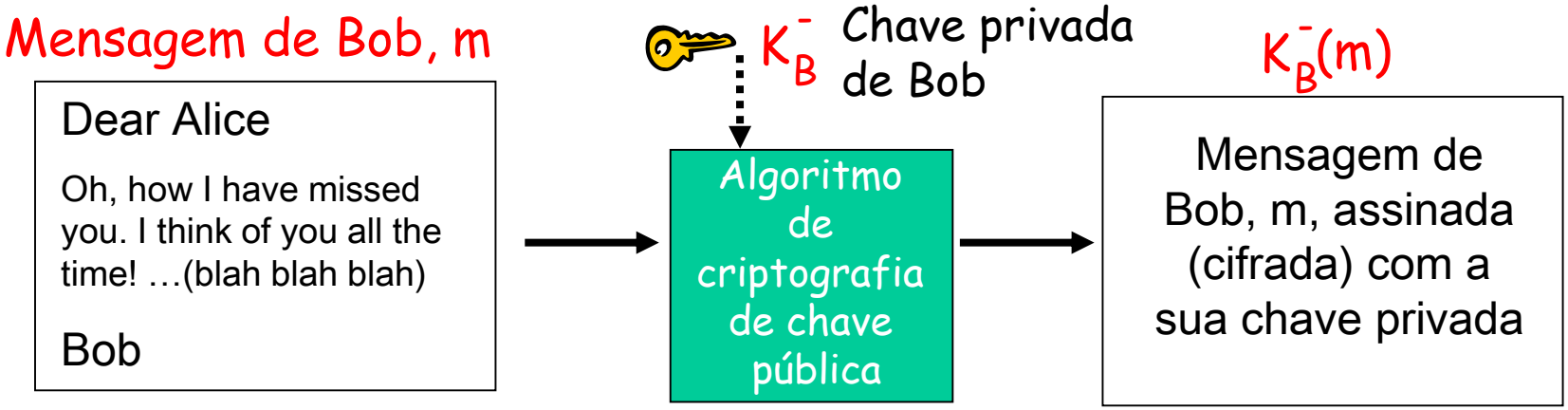
Técnica criptográfica análoga às assinaturas à mão.

- Transmissor (Bob) assina digitalmente o documento, atestando que ele é o dono/criador do documento.
- Verificável, não forjável: destinatário (Alice) pode verificar que Bob, e ninguém mais, 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 (mais)

- Suponha que Alice receba a msg 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.

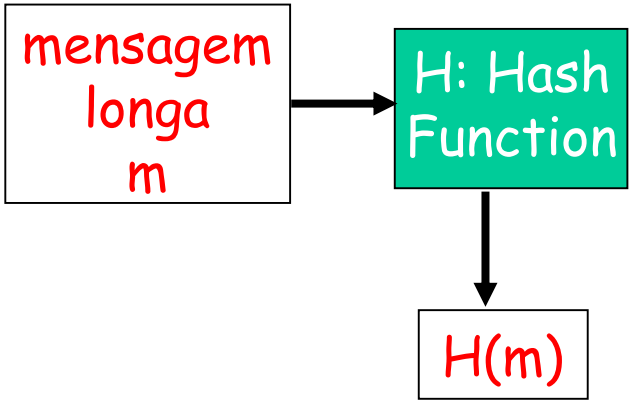
Alice portanto verifica que:

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

Não-repúdio:

- ✓ Alice pode levar m, e a assinatura $K_B^-(m)$ para o tribunal e provar que Bob assinou m.

Resumo (*Digest*) de Mensagens



A codificação com chave pública de mensagens longas é cara computacionalmente

Objetivo: assinatura digital (impressão digital) de comprimento fixo, fácil de ser calculada.

- aplique função de hash H a m, obtém resumo da mensagem de comprimento fixo, *H(m)*.

Propriedades das funções de Hash:

- Muitas-para-1
- Produz resumo da msg de comprimento fixo (impressão digital)
- Dado o resumo da mensagem x, é computacionalmente inviável encontrar m de modo que x = H(m)

checksum Internet: função de hash muito pobre

Checksum Internet possui algumas propriedades das funções de hash:

- ✓ Produz resumos de comprimento fixo (soma de 16-bits) da mensagem
- ✓ é do tipo muitos para um

Mas dada uma mensagem com um dado valor de hash, é fácil encontrar outra mensagem com o mesmo valor de hash:

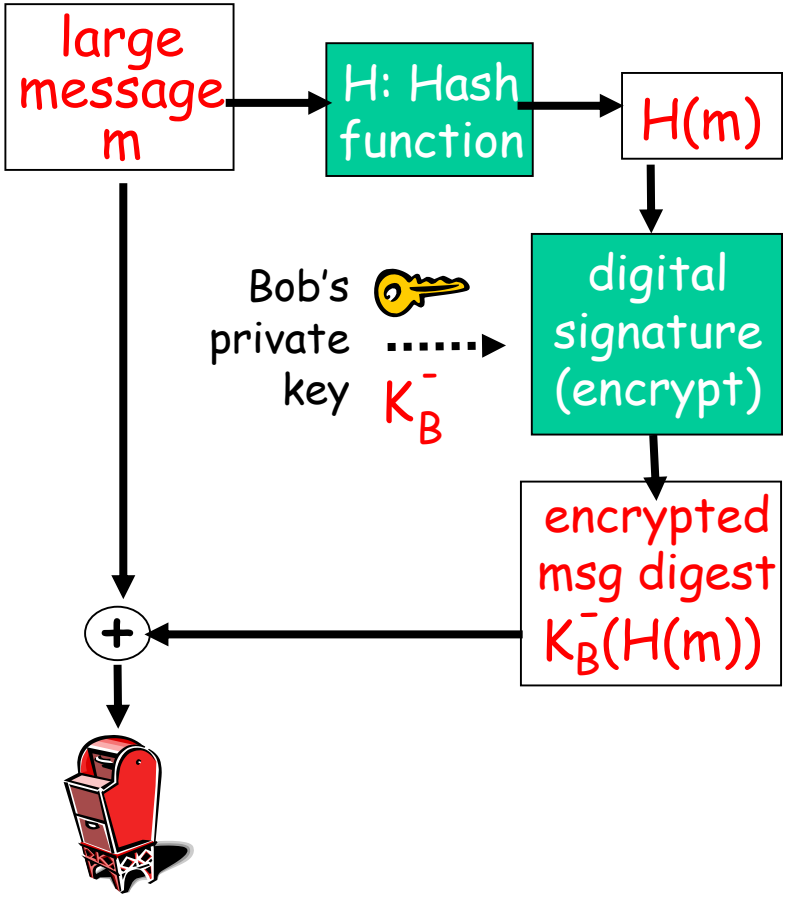
mensagem	Formato ASCII
I O U 1	49 4F 55 31
0 0 . 9	30 30 2E 39
9 B O B	39 42 D2 42
	B2 C1 D2 AC

mensagem	Formato ASCII
I O U 9	49 4F 55 39
0 0 . 1	30 30 2E 31
9 B O B	39 42 D2 42
	B2 C1 D2 AC

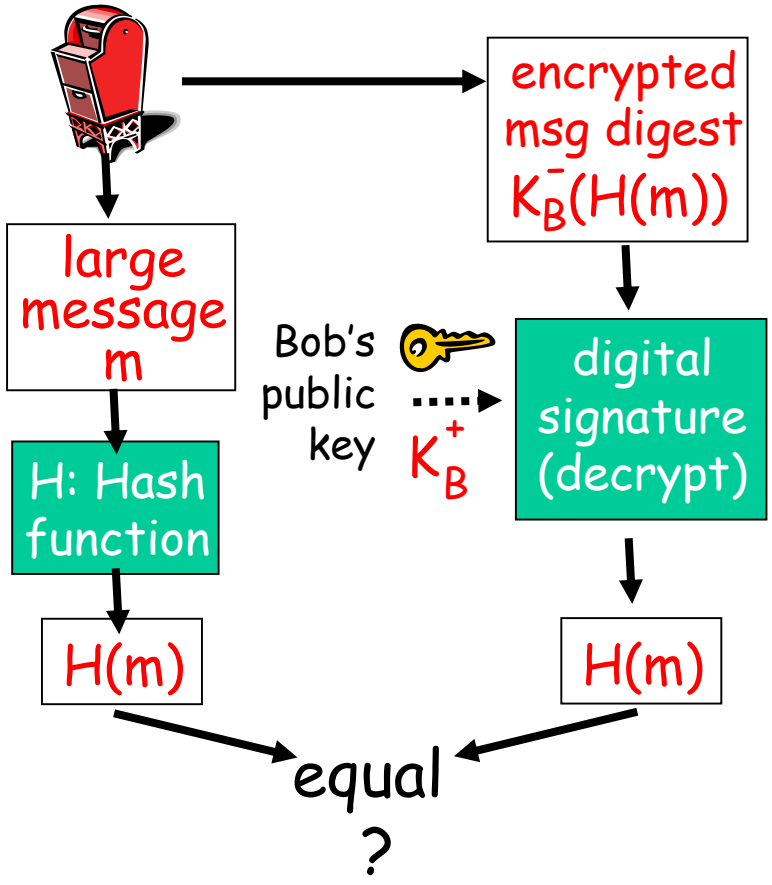
mensagens diferentes com mesmos checksums!

Assinatura digital = Assinatura do resumo da mensagem

Bob envia mensagem assinada digitalmente:



Alice verifica a assinatura e a integridade da mensagem assinada digitalmente:



Algoritmos para a Função de Hash

- A função de hash MD5 é largamente utilizada (RFC 1321)
  - Calcula resumo da mensagem de 128-bits num processo de 4 etapas.
  - dada uma seqüência arbitrária x de 128-bits, parece difícil construir uma msg m cujo hash MD5 seja igual a x.
- Também é usado o SHA-1.
  - padrão americano [NIST, FIPS PUB 180-1]
  - resumo de msg de 160-bits

Capítulo 8 roteiro

8.1 O que é segurança em redes?
8.2 Princípios de criptografia
8.3 Autenticação
8.4 Integridade
8.5 Distribuição de chaves e certificação
8.6 Controle de acesso: *firewalls*
8.7 Ataques e contra medidas
8.8 Segurança em diversas camadas

Intermediários Confiáveis

Problema com chave simétrica:

- Como duas entidades escolhem chave secreta compartilhada pela rede?

Solução:

- centro confiável de distribuição de chaves (KDC) agindo como intermediário entre as entidades

Problema com chave pública:

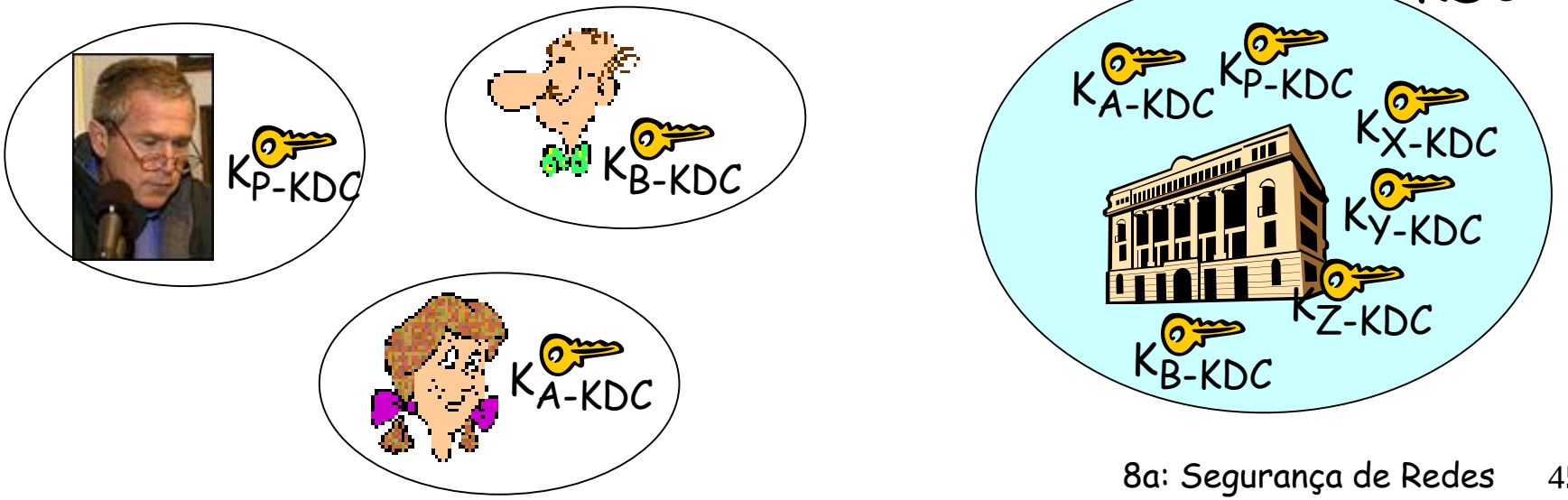
- Quando Alice obtém a chave pública de Bob (da web, e-mail ou disquete), como ela vai saber se a chave pública é mesmo de Bob e não de Trudy?

Solução:

- autoridade certificadora confiável (CA)

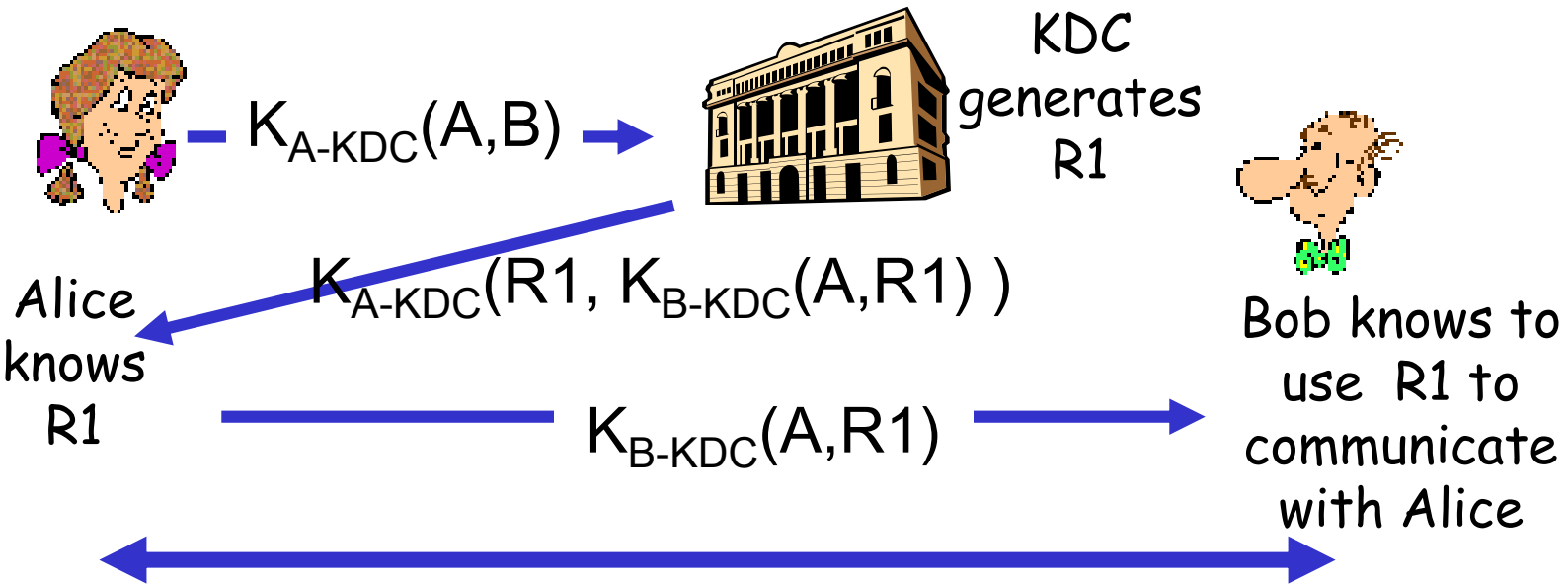
Centro de Distribuição de Chaves (KDC)

- Alice e Bob necessitam de chave simétrica compartilhada.
- KDC: servidor compartilha chaves secretas diferentes com cada usuário registrado.
- Alice e Bob conhecem as próprias chaves simétricas, $K_{A\text{-}KDC}$ e $K_{B\text{-}KDC}$, para se comunicar com o KDC.



Centro de Distribuição de Chaves (KDC)

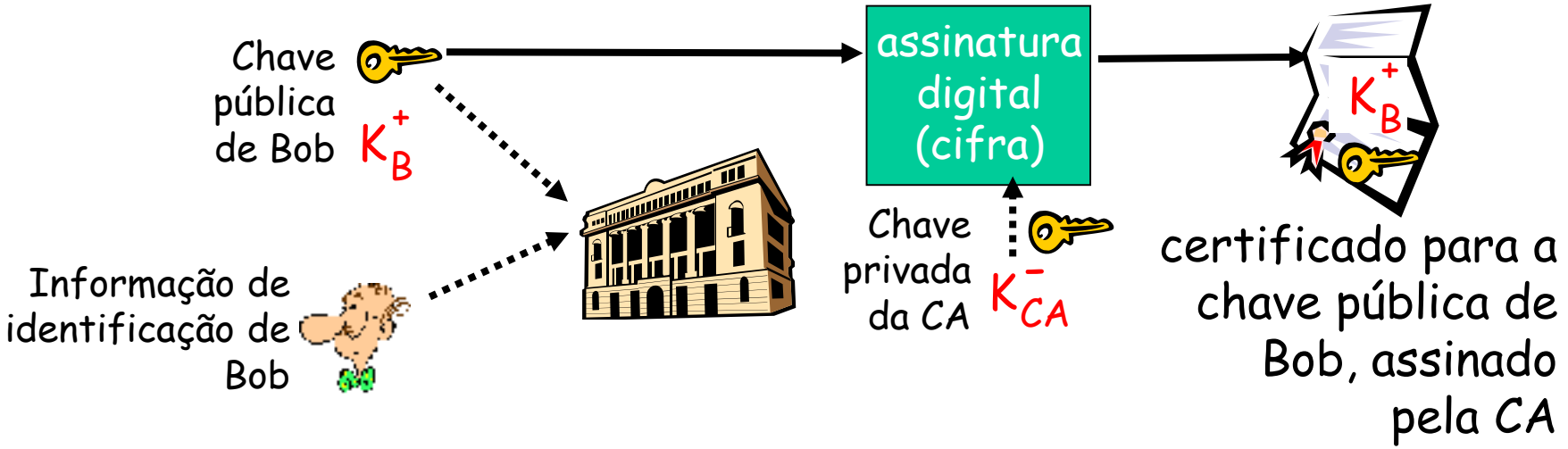
P: Como o KDC permite a Bob, Alice determinar a chave secreta simétrica compartilhada para se comunicarem?



Alice e Bob se comunicam: usando R1 como *chave da sessão* para criptografia simétrica compartilhada

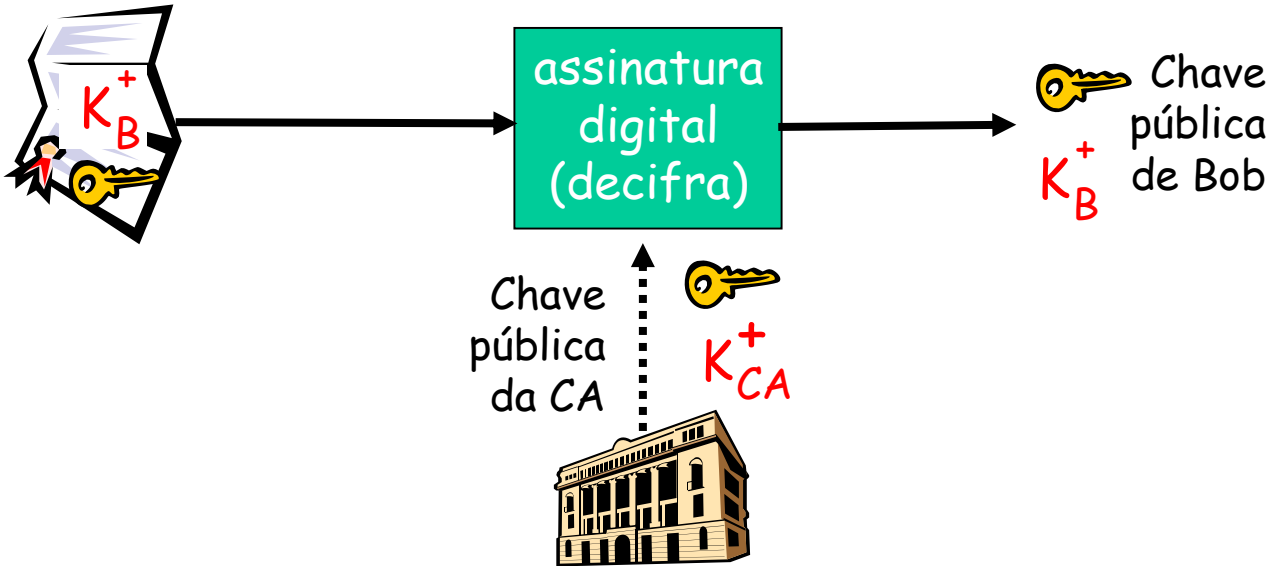
Autoridades Certificadoras

- Autoridade certificadora (CA): associam chave pública a uma entidade particular, E.
- E (pessoa, roteador) registra a sua chave pública com a CA.
  - 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"



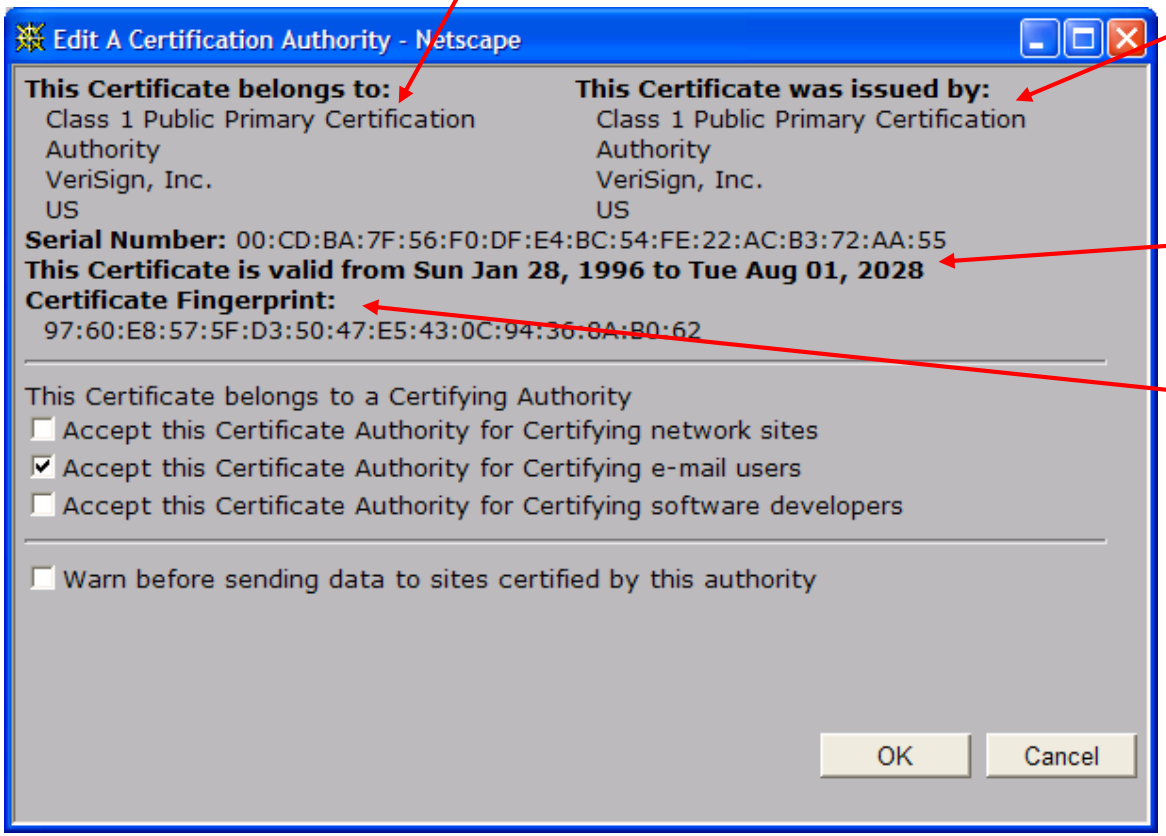
Autoridades Certificadoras

- 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.



Um certificado contém:

- 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

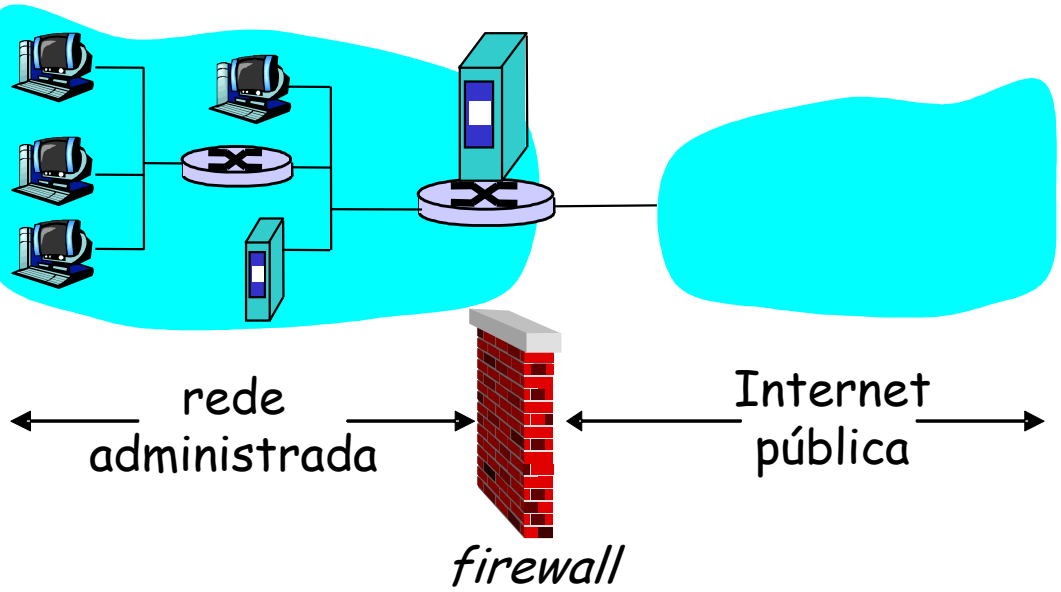
Capítulo 8 roteiro

8.1 O que é segurança em redes?
8.2 Princípios de criptografia
8.3 Autenticação
8.4 Integridade
8.5 Distribuição de chaves e certificação
8.6 Controle de acesso: *firewalls*
8.7 Ataques e contra medidas
8.8 Segurança em diversas camadas

Firewalls

firewall

isolam a rede interna da organização da Internet pública, permitindo que alguns pacotes passem e outros sejam bloqueados.



Firewalls: Para que?

prevenir ataques de negação de serviço:

- inundação de SYNs: atacante estabelece muitas conexões TCP "falsas", não deixa nenhum recurso para as conexões "reais".

prevenir modificação/acesso ilegal aos dados internos.

- ex., o atacante substitui a homepage da CIA com outra coisa.

permitir apenas acessos autorizados ao interior da rede (conjunto de usuários/hosts autenticados)

dois tipos de *firewalls*:

- camada de aplicação
- filtragem de pacotes

Filtragem de Pacotes

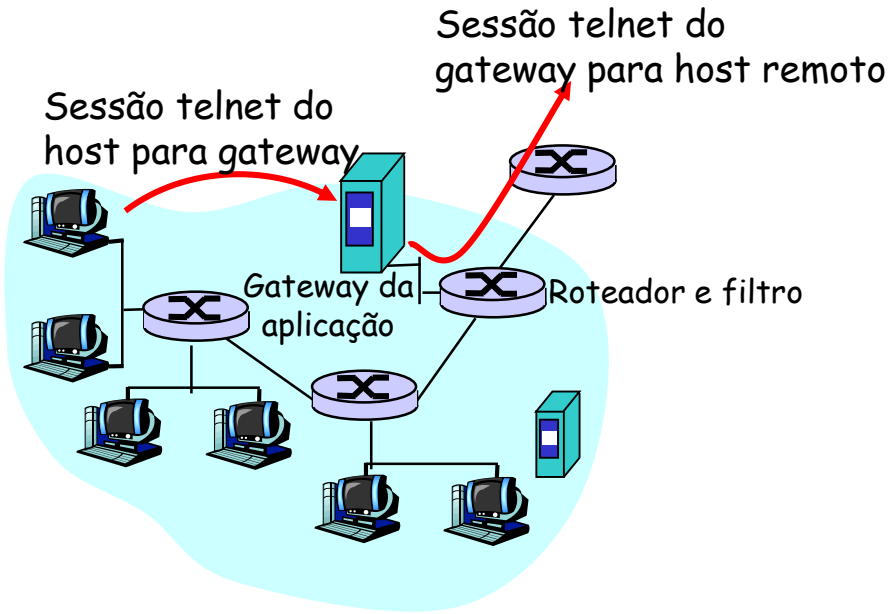
- rede interna conectada à Internet através de um roteador *firewall*
- roteador filtra pacote-a-pacote, decisão de encaminhar/descartar o pacote é baseada em:
  - endereço IP da origem, endereço IP do destino
  - número das portas de origem e destino do TCP/UDP
  - tipo da mensagem ICMP
  - bits de SYN e ACK do TCP

Filtragem de Pacotes

- Exemplo 1: Bloqueia datagramas de entrada e saída com campo de protocolo IP = 17 e com porta de origem ou destino = 23.
  - Todos os fluxos UDP de entrada e saída e conexões telnet são bloqueadas.
- Exemplo 2: Bloqueia segmentos TCP de entrada com ACK=0.
  - Previne que clientes externos estabeleçam conexões TCP com clientes internos, mas permitem que clientes internos se conectem com o exterior.

Gateways de Aplicações

- Filtra os pacotes baseado nos dados das aplicações assim como em campos IP/TCP/UDP.
- Exemplo: permite que usuários internos selecionados façam telnet para o exterior.



1. Requer que todos os usuários telnet façam o telnet através do gateway.
2. Para os usuários autorizados, o gateway estabelece uma conexão telnet com o host destino. O Gateway transfere os dados entre 2 conexões
3. O filtro do roteador bloqueia todas as conexões que não têm origem no gateway.

Limitações dos firewalls e gateways

- 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
- O software do cliente deve saber como contactar o gateway
  - ex., deve setar o endereço IP do proxy no browser
- Para o UDP os filtros normalmente usam uma política de tudo ou nada.
- Compromisso: grau de comunicação com o mundo externo, nível de segurança
- Muitos sítios altamente protegidos ainda sofrem ataques.

Capítulo 8 roteiro

8.1 O que é segurança em redes?
8.2 Princípios de criptografia
8.3 Autenticação
8.4 Integridade
8.5 Distribuição de chaves e certificação
8.6 Controle de acesso: *firewalls*
8.7 Ataques e contra medidas
8.8 Segurança em diversas camadas

Ameaças à segurança na Internet

Mapeamento (Reconhecimento do terreno):

- antes de atacar: descobrir que serviços estão implementados na rede
- Use `ping` para determinar que hosts têm endereços na rede
- Varredura de portas (*Port-scanning*): tenta estabelecer conexões TCP para cada porta em seqüência (para ver o que acontece)
- mapeador nmap (http://www.insecure.org/nmap/): "exploração da rede e auditoria de segurança"

Contramedidas?

Ameaças à segurança na Internet

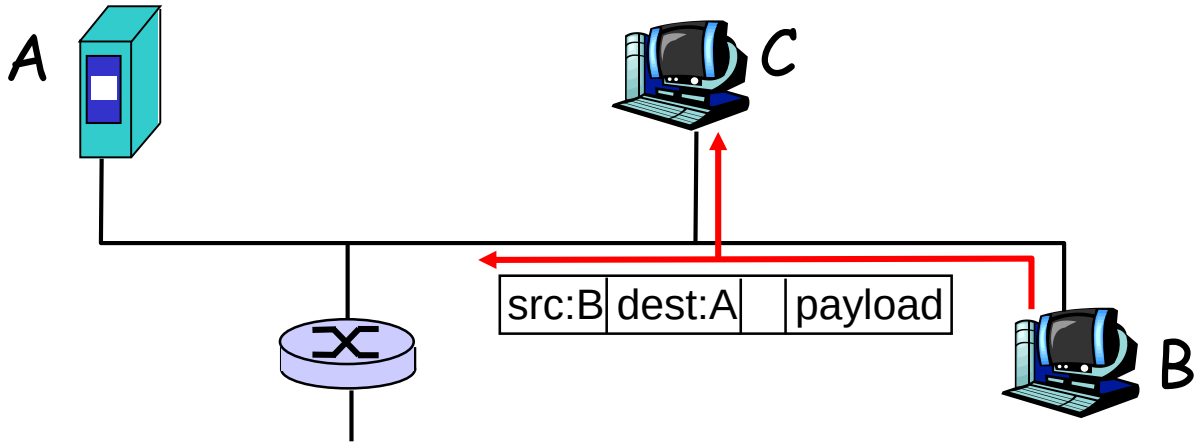
Mapeamento: contramedidas

- registra o tráfego que entra na rede
- procura atividade suspeita (endereços IP, portas sendo varridas seqüencialmente)

Ameaças à segurança na Internet

Bisbilhotar (*Sniffing*) os pacotes:

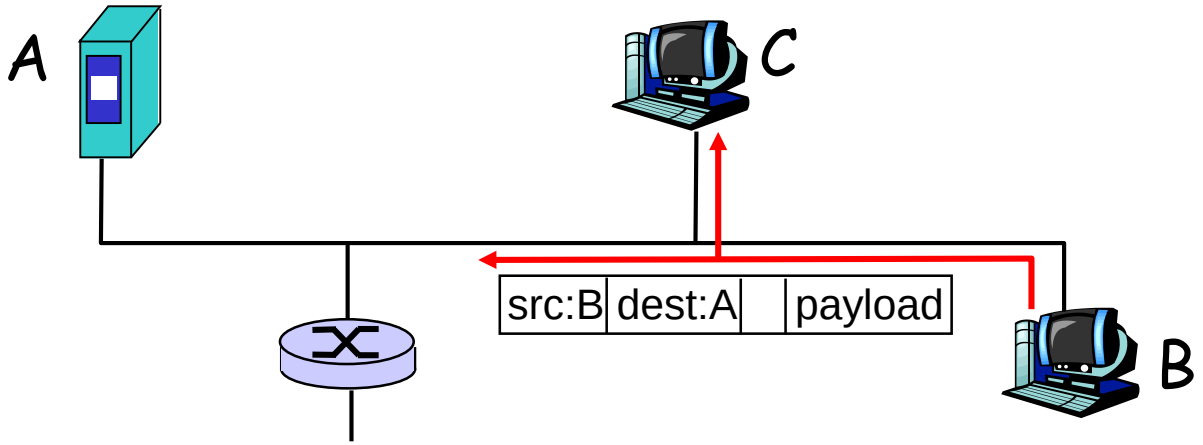
- meios de difusão (*broadcast*)
- NIC promíscuo lê todos os pacotes que passam
- pode ler todos os dados não cifrados (ex. senhas)
- ex.: C bisbilhota os pacotes de B



Ameaças à segurança na Internet

Packet sniffing: contramedidas

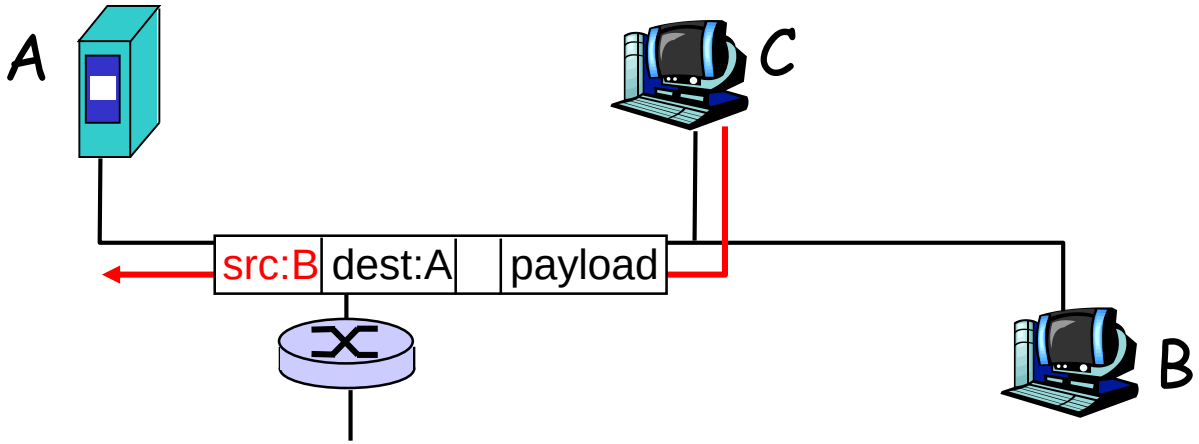
- todos os hosts na organização usam software que verifica periodicamente se a interface do host encontra-se em modo promíscuo.
- Um host por segmento de mídia de difusão (Ethernet comutada no hub)



Ameaças à segurança na Internet

Imitação *(Spoofing)* de IP :

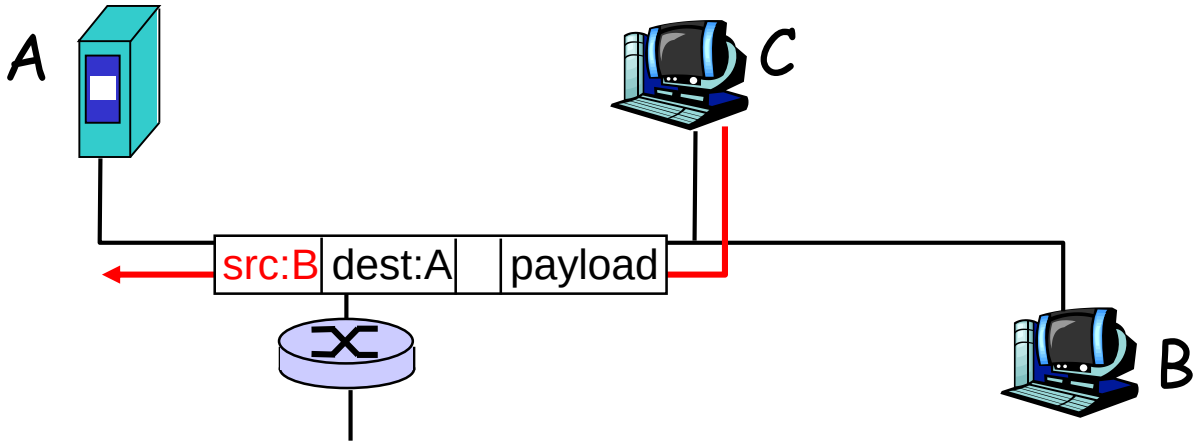
- pode gerar pacotes IP "brutos" diretamente da aplicação, colocando qualquer valor no campo do endereço IP da origem
- o receptor não consegue identificar se a origem foi macaqueada
- ex.: C tenta se passar por B



Ameaças à segurança na Internet

Imitação de IP: filtragem de entrada

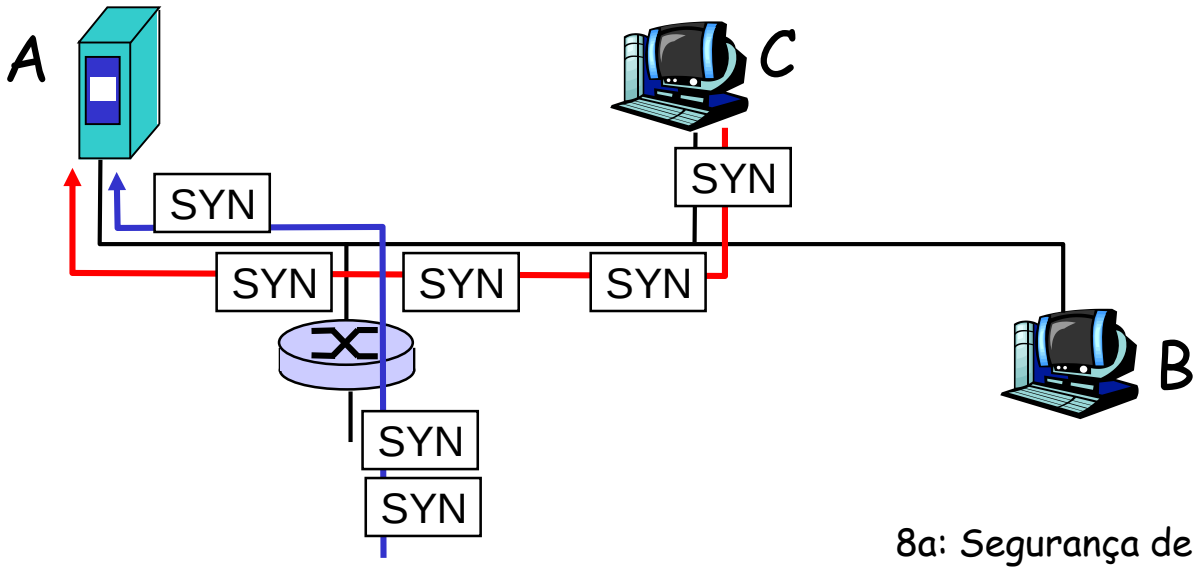
- roteadores não devem encaminhar pacotes com endereços inválidos de origem (ex., endereço de origem do datagrama fora da rede do roteador)
- ótimo, mas a filtragem de entrada não é obrigatória para todas as redes



Ameaças à segurança na Internet

Recusa de serviço (DOS - Denial of service):

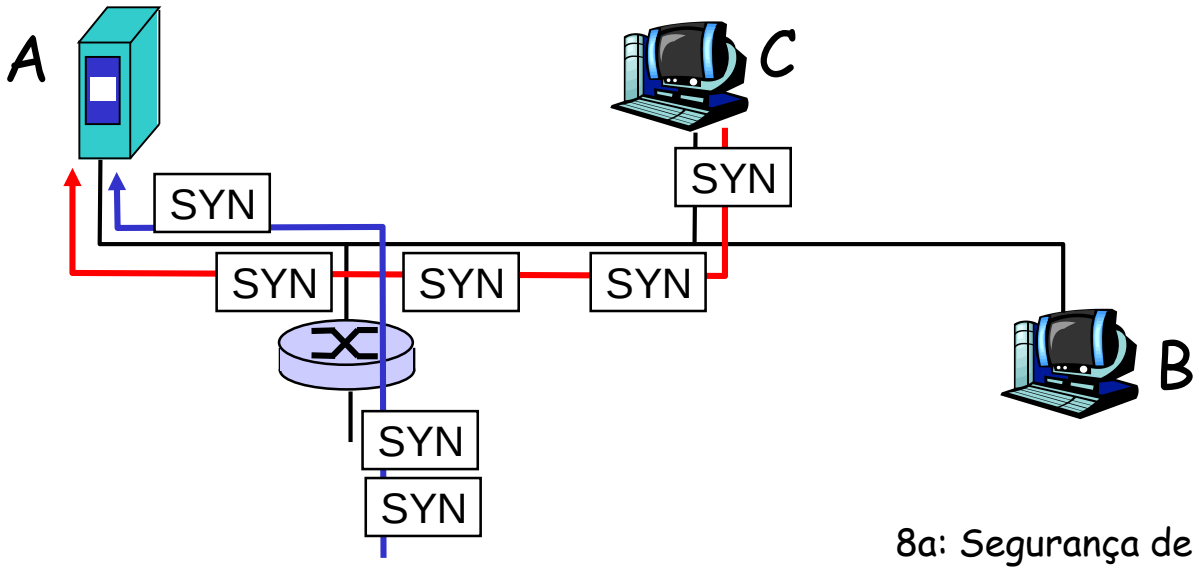
- inundação de pacotes gerados maliciosamente "atolam" o receptor
- DOS distribuído (DDOS): múltiplas fontes de forma coordenada atolam o receptor
- ex., C e host remoto atacam A com SYNs



Ameaças à segurança na Internet

Recusa de serviço (DOS): contramedidas

- filtrar pacotes inundados (ex., SYN) antes de alcançar o host: joga fora os bons com os ruins
- Identificar rastros até a origem das inundações (provavelmente uma máquina inocente, comprometida)

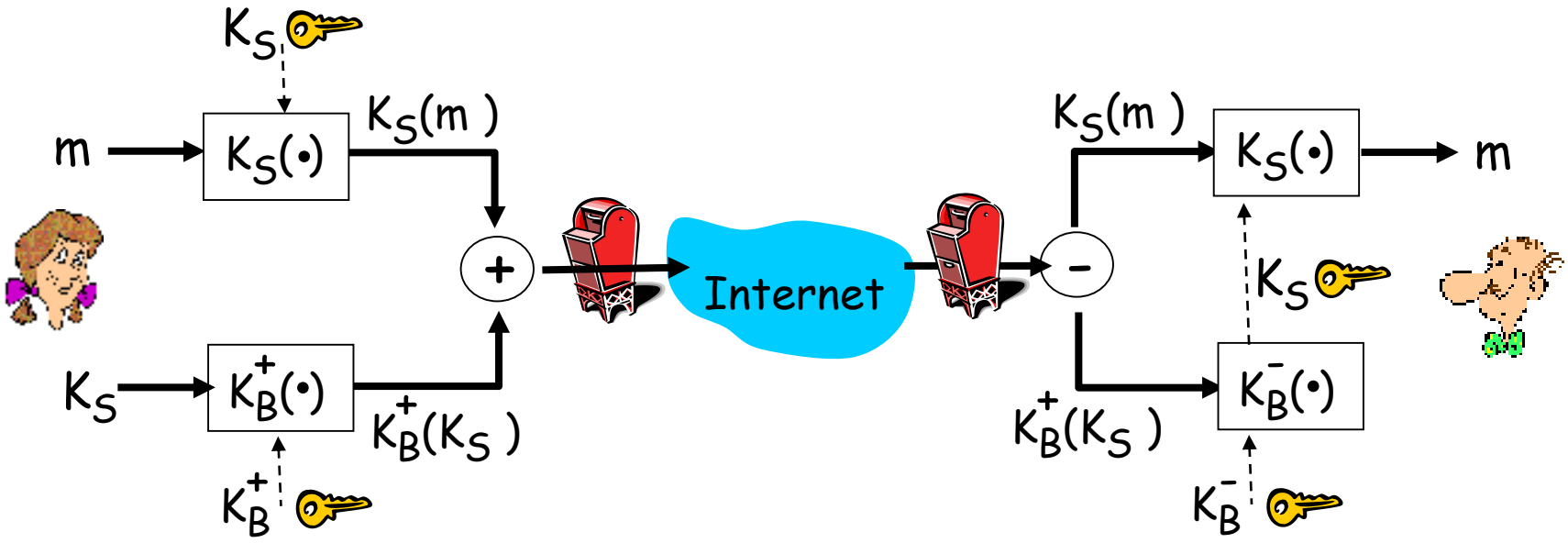


Capítulo 8 roteiro

8.1 O que é segurança em redes?
8.2 Princípios de criptografia
8.3 Autenticação
8.4 Integridade
8.5 Distribuição de chaves e certificação
8.6 Controle de acesso: *firewalls*
8.7 Ataques e contra medidas
8.8 Segurança em diversas camadas
  8.8.1 E-mail Seguro
  8.8.2 Sockets seguros
  8.8.3 IPsec
  8.8.4 Segurança no 802.11

E-mail seguro

- Alice deseja enviar mensagem secreta, m, para Bob.

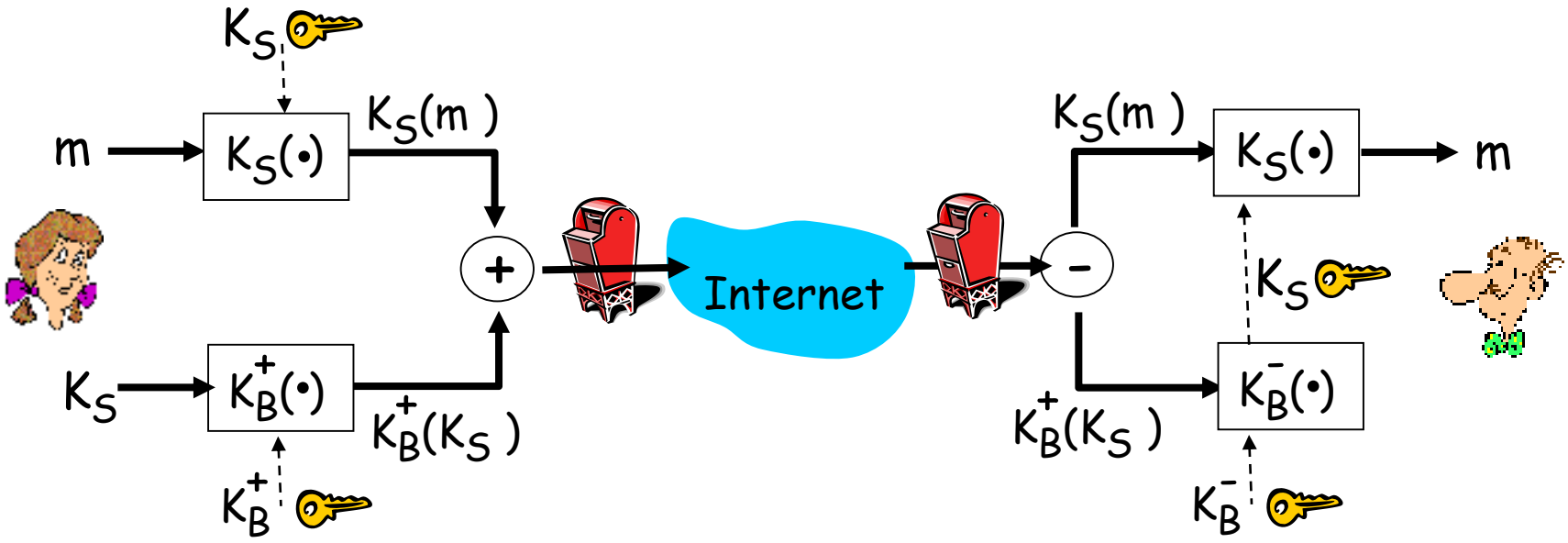


Alice:

- gera aleatoriamente chave *simétrica*, K_S.
- cifra a mensagem comK_S (por questões de eficiência)
- também cifra K_S com a chave pública de Bob.
- envia tanto $K_S(m)$ como $K_B(K_S)$ para Bob.

E-mail seguro

- Alice deseja enviar mensagem secreta, m, para Bob.

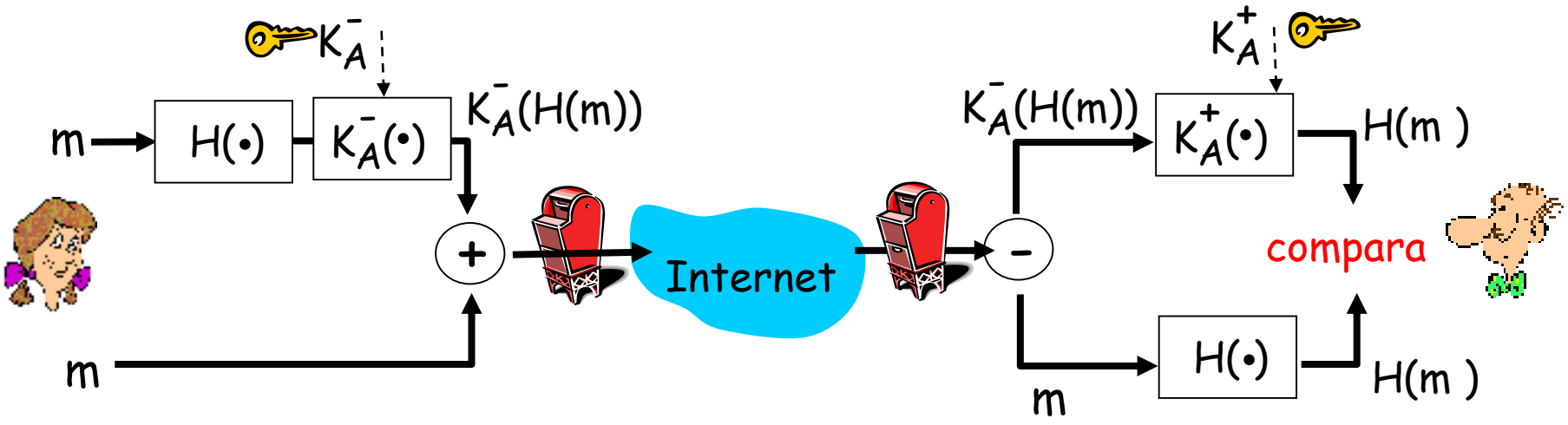


Bob:

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

E-mail seguro (continuação)

• Alice deseja prover autenticação do transmissor e integridade da mensagem.

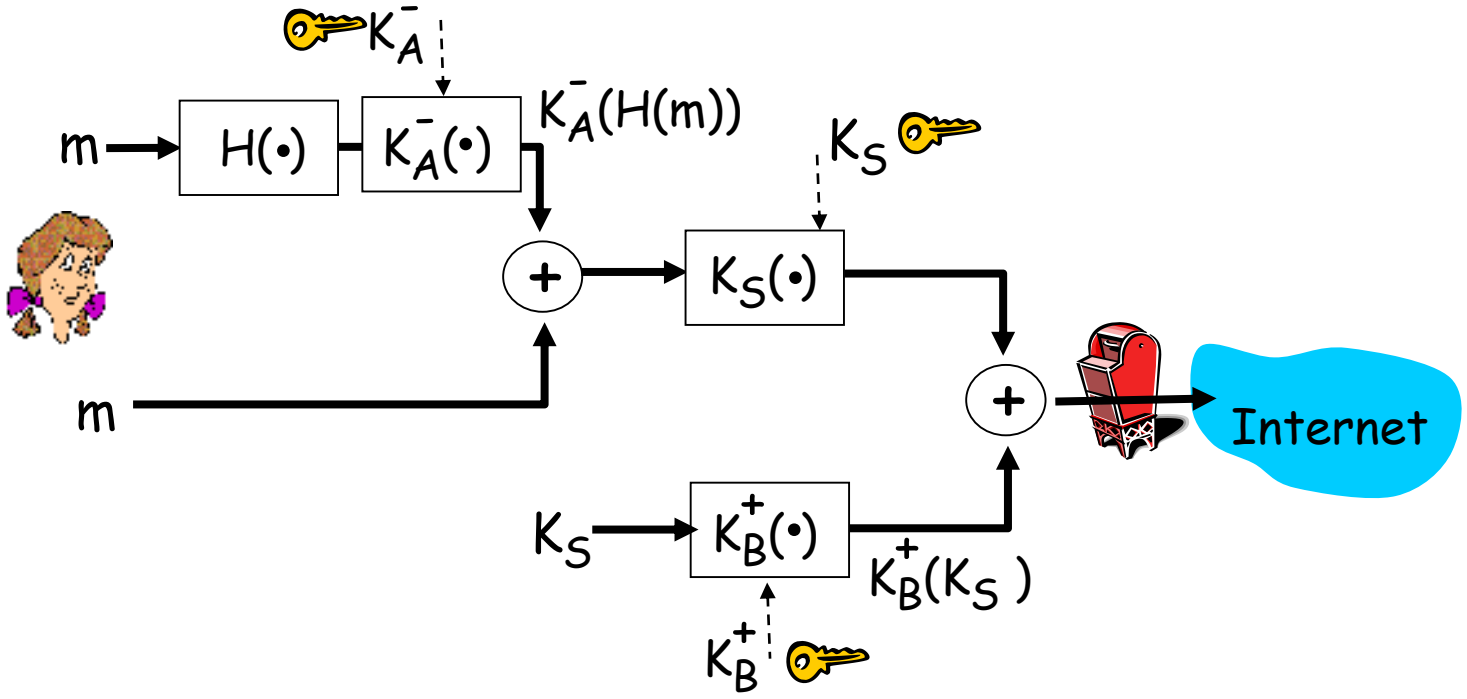


• Alice assina a mensagem digitalmente.

• envia tanto a mensagem (em texto aberto) como a assinatura digital.

E-mail seguro (continuação)

• Alice deseja prover sigilo, autenticação do transmissor e integridade da mensagem.



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

Pretty good privacy (PGP)

- Esquema de criptografia de e-mails para a internet, um padrão de fato.
- Usa criptografia de chave simétrica, criptografia de chave pública, função de hash e assinatura digital como descrito.
- Provê sigilo, autenticação do transmissor, integridade.
- O inventor, Phil Zimmerman, foi alvo de uma investigação federal que durou 3 anos.

Uma mensagem assinada com PGP:

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

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

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

Secure sockets layer (SSL)

- SSL trabalha na camada de transporte. Provê segurança para qualquer aplicação baseada em TCP que use os serviços SSL.
- SSL: usado entre clientes e servidores WWW para comércio eletrônico (https).
- Serviços de segurança SSL:
  - autenticação do servidor
  - codificação dos dados
  - autenticação do cliente (opcional)
- Autenticação do servidor:
  - cliente habilitado com SSL inclui chaves públicas das CAs confiáveis.
  - Cliente solicita certificado do servidor, emitido por CA confiável.
  - Cliente usa a chave pública da CA para extrair a chave pública do servidor a partir do seu certificado.
- Visite o menu de segurança do seu browser para verificar quais são as suas CAs confiáveis.

SSL (continuação)

Sessão SSL criptografada:

- Browser gera chave simétrica para a sessão, cifra-a com a chave pública do servidor, envia a chave cifrada para o servidor.
- O servidor decifra a chave da sessão usando a sua chave privada.
- Browser e servidor concordam que as msgs futuras serão cifradas.
  - Todos os dados enviados para o socket TCP (pelo cliente ou servidor) são cifrados com a chave da sessão.

- SSL: base para a Segurança da Camada de Transporte do IETF (TLS).
- SSL pode ser usado para aplicações não Web, ex., IMAP.
- Autenticação do cliente pode ser realizada com certificados do cliente.

Ipsec: Segurança da Camada de Rede

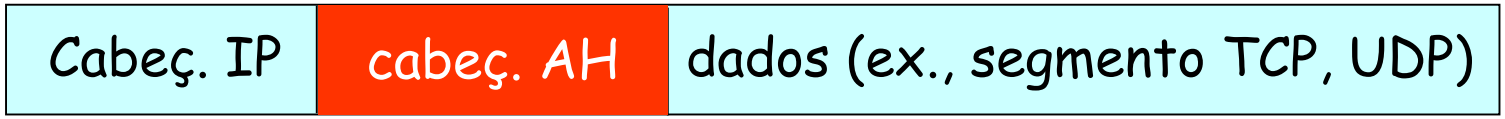
- Sigilo na camada de rede:
  - host transmissor cifra os dados num datagrama IP
  - segmentos TCP e UDP; mensagens ICMP e SNMP.
- Autenticação da camada de rede
  - host destino pode autenticar o endereço IP da origem
- Dois protocolos principais:
  - protocolo de cabeçalho de autenticação (AH)
  - protocolo de encapsulamento de segurança da carga (ESP)
- Tanto para AH como ESP, negociação origem-destino:
  - cria canal lógico de camada de rede chamado de acordo de serviço (AS-*service agreement*)
- Cada SA é unidirecional.
- Determinado univocamente por:
  - protocolo de segurança (AH ou ESP)
  - endereço IP da origem
  - ID da conexão de 32-bits

Protocolo de Autenticação de Cabeçalho (AH)

- Provê autenticação do host de origem e integridade dos dados mas não sigilo.
- Cabeçalho AH inserido entre o cabeçalho IP e o campo de dados do IP
- Protocolo= 51.
- Roteadores intermediários processam os datagramas como usual

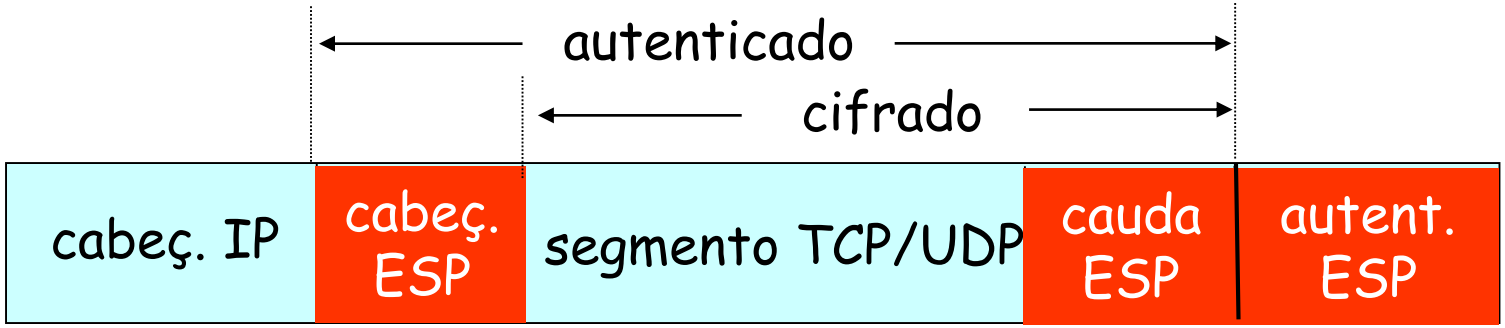
Cabeçalho AH inclui:

- identificador da conexão
- dados de autenticação: resumo assinado da msg, calculado sobre o datagrama original IP, provendo autenticação da origem e integridade dos dados.
- Campo de próximo cabeçalho: especifica o tipo dos dados (TCP, UDP, ICMP, etc.)



Protocolo ESP

- Provê sigilo, autenticação do host e integridade dos dados.
- Dados e cauda do ESP são cifrados.
- O campo de próximo cabeçalho encontra-se na cauda do ESP.
- Campo de autenticação do ESP é semelhante ao do AH.
- Protocol = 50.



Segurança IEEE 802.11

- *War-driving:* dirija em torno da área da Baía (de S. Fco.) verifique quantas redes 802.11 estão disponíveis!
  - Mais do que 9000 acessíveis em vias públicas
  - 85% não usam nenhuma criptografia/autenticação
  - Facilitam a bisbilhotagem de pacotes e diversos ataques!
- Tornando o 802.11 seguro
  - cifragem, autenticação
  - primeira tentativa de segurança 802.11: *Wired Equivalent Privacy* (WEP): um fracasso!
  - tentativa atual: 802.11i

Wired Equivalent Privacy (WEP):

- autenticação como no protocolo *ap4.0*
  - host solicita autenticação do ponto de acesso
  - ponto de acesso envia *nonce* de 128 bits
  - host codifica o *nonce* usando chave simétrica compartilhada
  - ponto de acesso decifra o *nonce*, autentica o host
- não há mecanismo de distribuição de chaves
- autenticação: basta conhecer a chave compartilhada

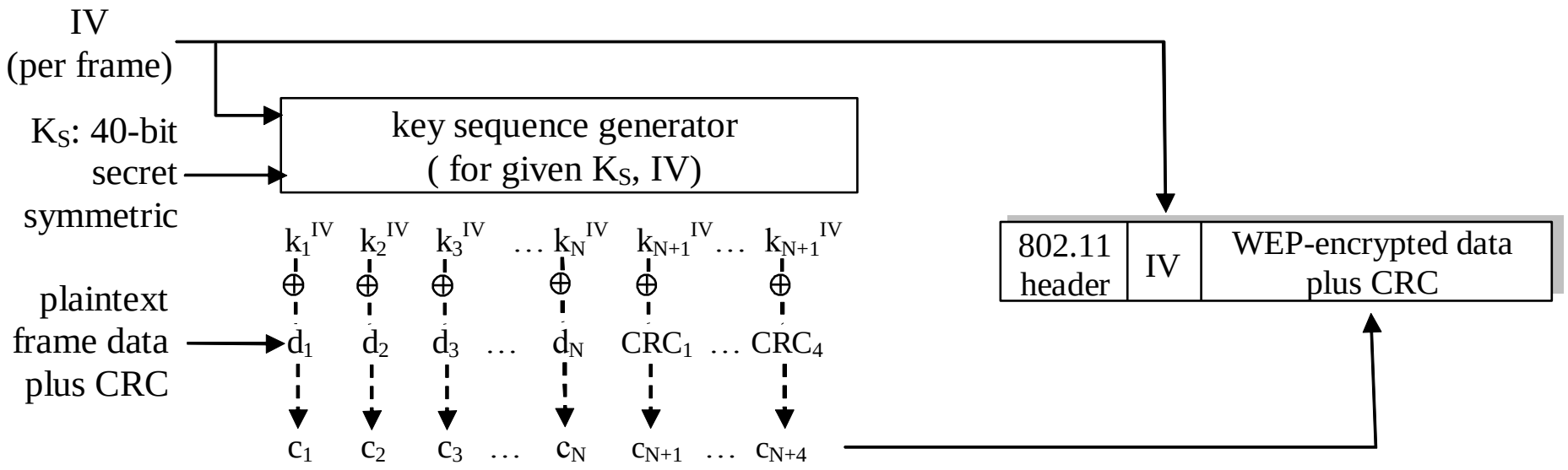
Cifragem de dados com o WEP

- Host/AP compartilham chave simétrica de 40 bits (semi-permanente)
- Host concatena vetor de inicialização (IV) de 24-bits para criar chave de 64-bits
- Chave de 64 bits usada para gerar fluxo de chaves, k_i^{IV}
- k_i^{IV} usado para cifrar o i-ésimo byte, d_i, no quadro:

$$c_i = d_i \text{ XOR } k_i^{IV}$$

- IV e bytes cifrados, c_i são enviados no quadro

Cifragem WEP 802.11



Cifragem WEP do lado do transmissor

Quebrando a cifragem WEP 802.11

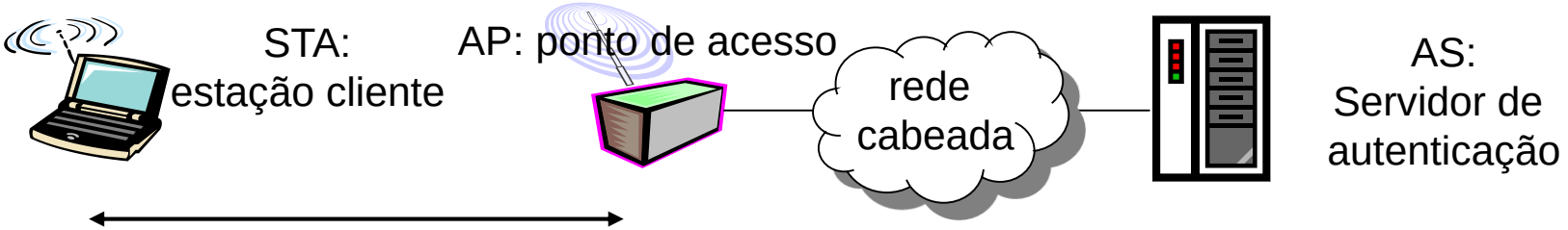
Furo de segurança:

- IV de 24-bits, um IV por quadro, -> IV's são eventualmente reutilizados
- IV transmitido em texto aberto -> reutilização do IV é detectada
- **Ataque:**
  - Trudy faz com que Alice cifre textos abertos conhecidos d_1 d_2 d_3 d_4 …
  - Trudy vê: $c_i = d_i$ XOR k_i^{IV}
  - Trudy conhece c_i d_i, então pode calcular k_i^{IV}
  - Trudy conhece seqüência de chaves de cifragem k_1^{IV} k_2^{IV} k_3^{IV} …
  - Na próxima vez que IV for usado, Trudy poderá decifrar!

802.11i: segurança melhorada

- Diversas formas (mais fortes) de cifragem são possíveis
- Provê distribuição de chaves
- Usa servidor de autenticação separado do ponto de acesso

802.11i: quatro fases da operação



1 Descoberta das capacitações de segurança

2 STA e AS se autenticam mutuamente, juntos
Geram chave mestra (MK - Master Key). AP serve como "passagem"

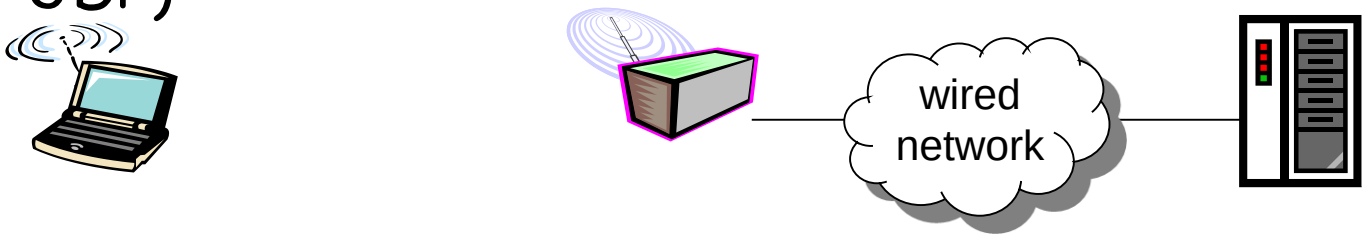
3 STA deriva chave mestra par (PMK)

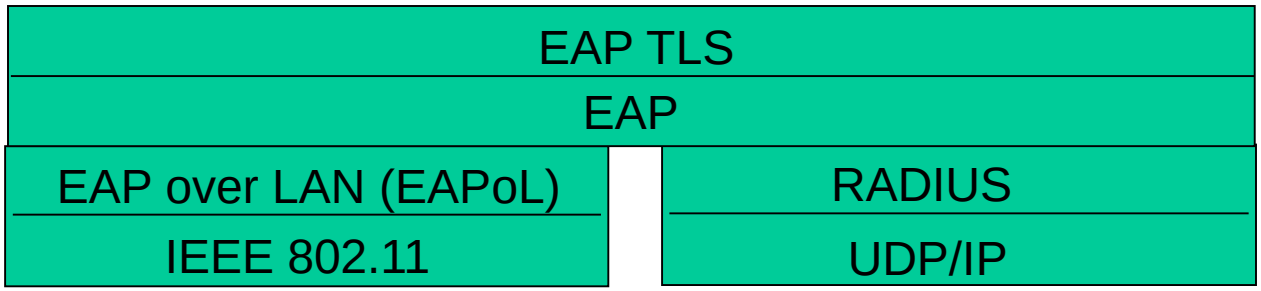
3 AS deriva a mesma PMK, envia para o AP

4 STA, AP usa PMK para derivar chave temporária (TK - Temporal Key) usada para cifragem, integridade das mensagens

EAP: protocolo extensível de autenticação

- 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)





Segurança de Rede (resumo)

Técnicas básicas……

- criptografia (simétrica e pública)
- autenticação
- integridade das mensagens
- distribuição de chaves

…. usadas em muitos cenários de segurança diferentes

- correio eletrônico seguro
- transporte seguro (SSL)
- IP sec
- 802.11