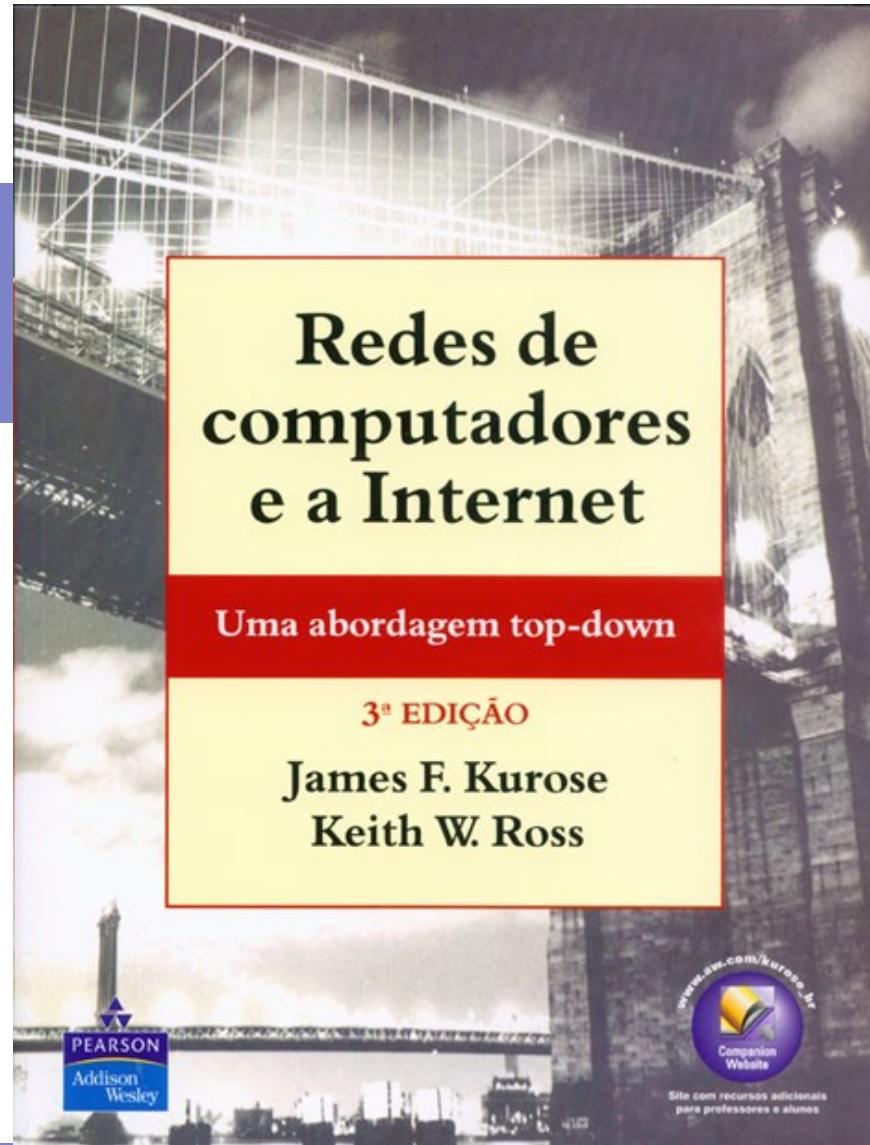


Redes de computadores e a Internet

Capítulo 8

Segurança em redes de computadores



Objetivos do capítulo:

- Compreender princípios de segurança de redes:
 - Criptografia e seus *muitos* usos além da “confidencialidade”
 - Autenticação
 - Integridade de mensagem
 - Distribuição de chave
- Segurança na prática:
 - Firewalls
 - Segurança nas camadas de aplicação, transporte, rede e enlace



- 8.1 O que é segurança?
- 8.2 Princípios da criptografia
- 8.3 Autenticação
- 8.4 Integridade
- 8.5 Distribuição de chaves e certificação
- 8.6 Controle de acesso: firewalls
- 8.7 Ataques e medidas de defesa
- 8.8 Segurança em muitas camadas



8 O que é segurança de rede?

Confidencialidade: apenas o remetente e o destinatário pretendido deveriam “entender” o conteúdo da mensagem

- Remetente cifra (encripta) a mensagem
- Destinatário decifra (decripta) a mensagem

Autenticação: remetente e destinatário querem confirmar a identidade um do outro

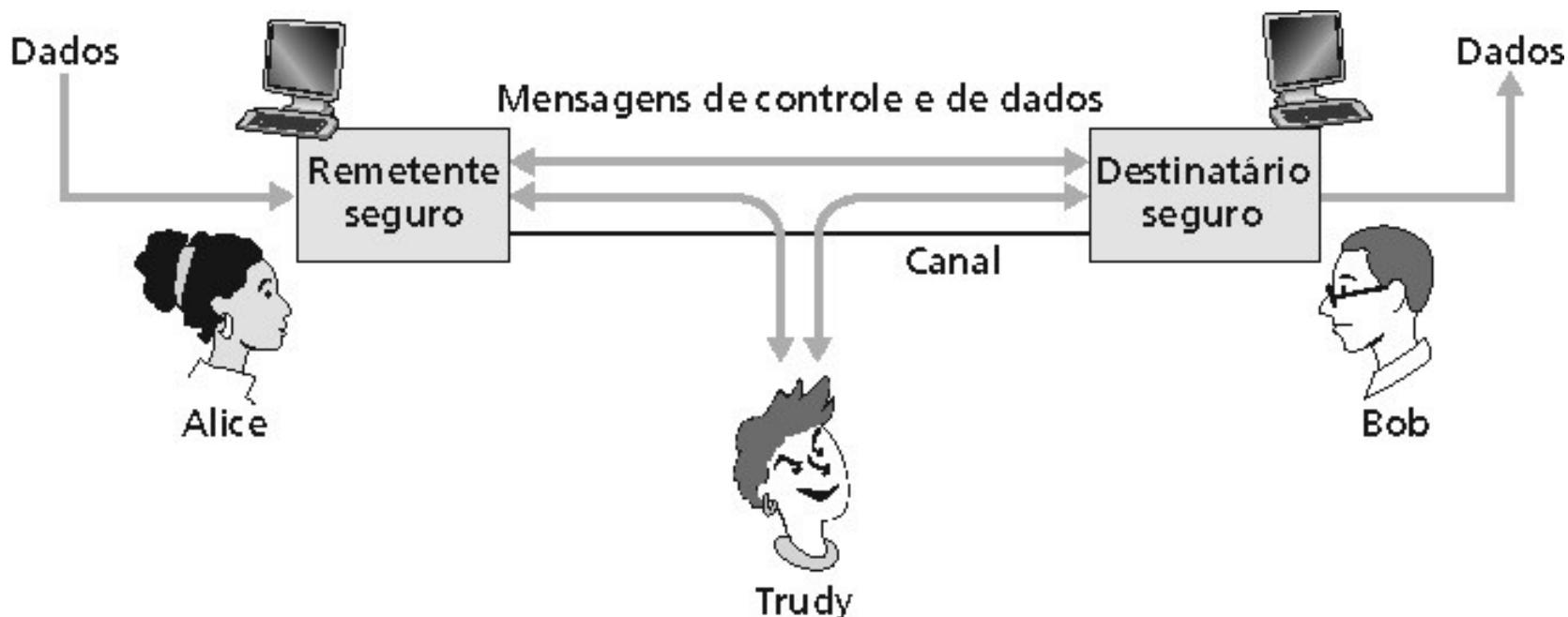
Integridade de mensagens: remetente e destinatário querem assegurar que as mensagens não foram alteradas, (em trânsito, ou depois) sem detecção

Acesso e disponibilidade: serviços devem ser acessíveis e disponíveis para os usuários



Amigos e inimigos: Alice, Bob, Trudy

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



Quem poderiam ser Bob e Alice?

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



Existem pessoas más por aí!

P.: O que uma “pessoa má” pode fazer?

R.: Muito!

Interceptação de mensagens

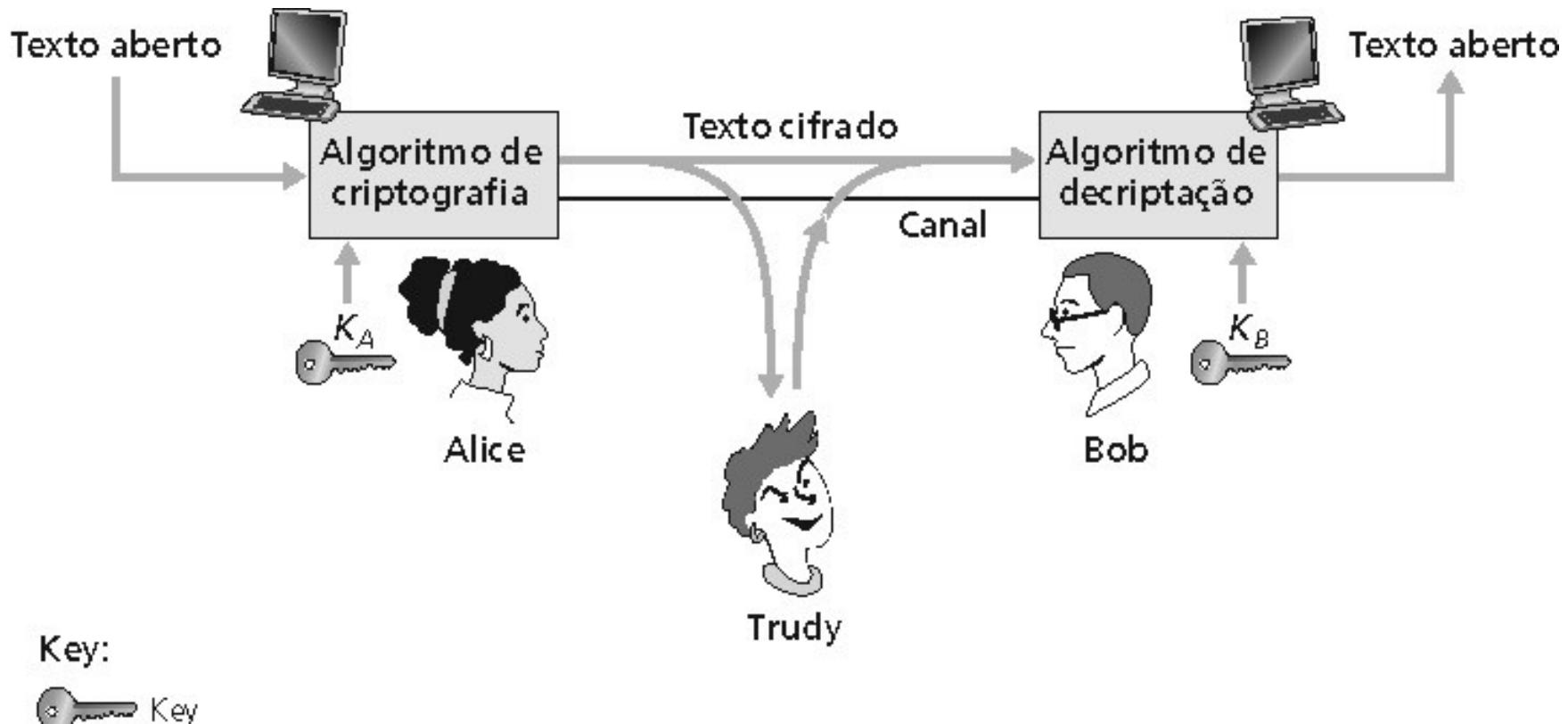
- *Inserção* ativa de mensagens na conexão
- *Personificação*: falsificar (spoof) endereço de origem no pacote (ou qualquer campo no pacote)
- *Hijacking*: assume a conexão removendo o transmissor ou receptor e se inserindo no lugar
- *Negação de serviço*: impede que um serviço seja usado pelos outros (ex., por sobrecarga de recursos)

mais sobre isso depois...



- 8.1 O que é segurança?
- 8.2 Princípios da criptografia
- 8.3 Autenticação
- 8.4 Integridade
- 8.5 Distribuição de chaves e certificação
- 8.6 Controle de acesso: firewalls
- 8.7 Ataques e medidas de defesa
- 8.8 Segurança em muitas camadas





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

Chave pública de criptografia: criptografa com chave pública, decriptografa com chave secreta (privada)

Código de substituição: substituindo uma coisa por outra

- Código monoalfabético: substituir uma letra por outra

texto aberto: abcdefghijklmnopqrstuvwxyz
texto cifrado: mnbvcxzasdfghjklpoiuytrewq

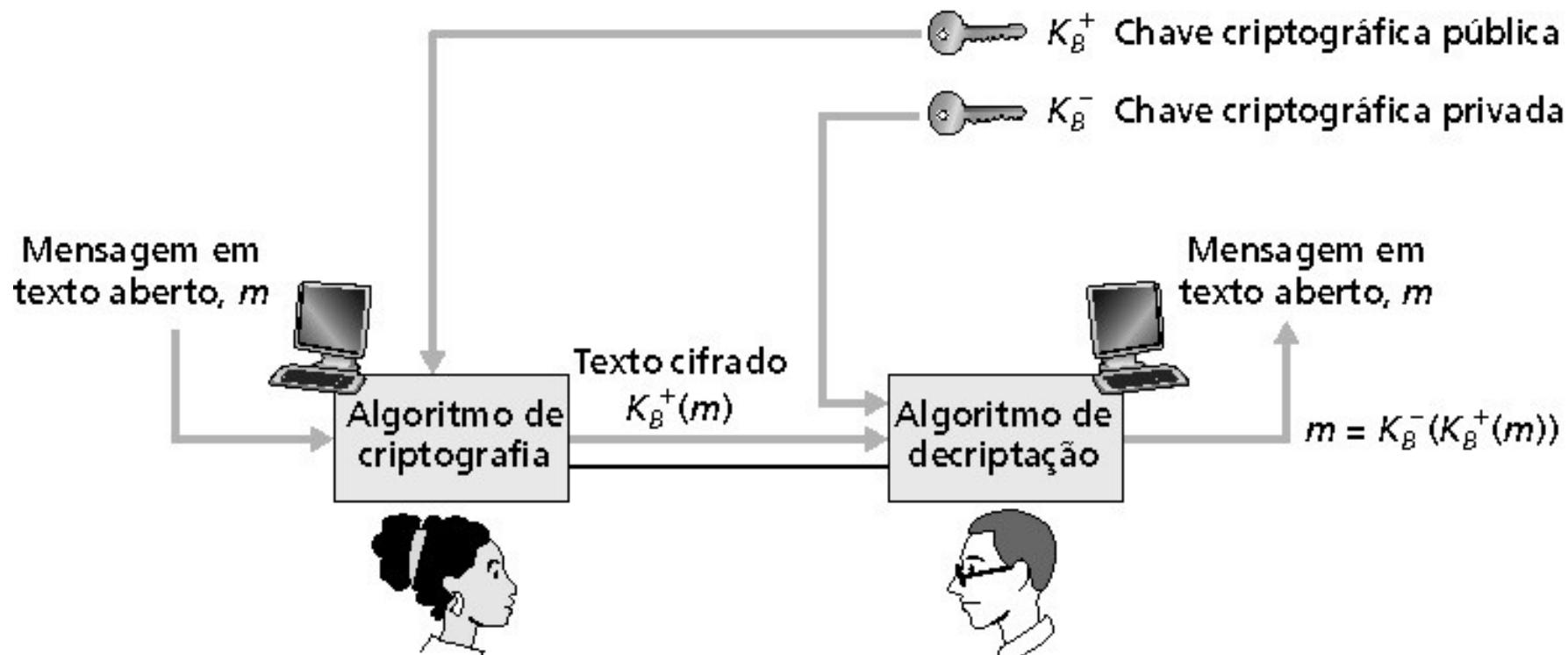


Ex.:

texto aberto: bob. i love you. alice
texto cifrado: nkn. s gktc wky. mgsbc

P.: Quão difícil é quebrar esse código simples?

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



(simétrica) conhecida: K

- Ex.: sabe que a chave corresponde ao padrão de substituição num código substituição mono alfábético
- P.: Como Bob e Alice combinam o tamanho da chave?



DES: Data encryption standard

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

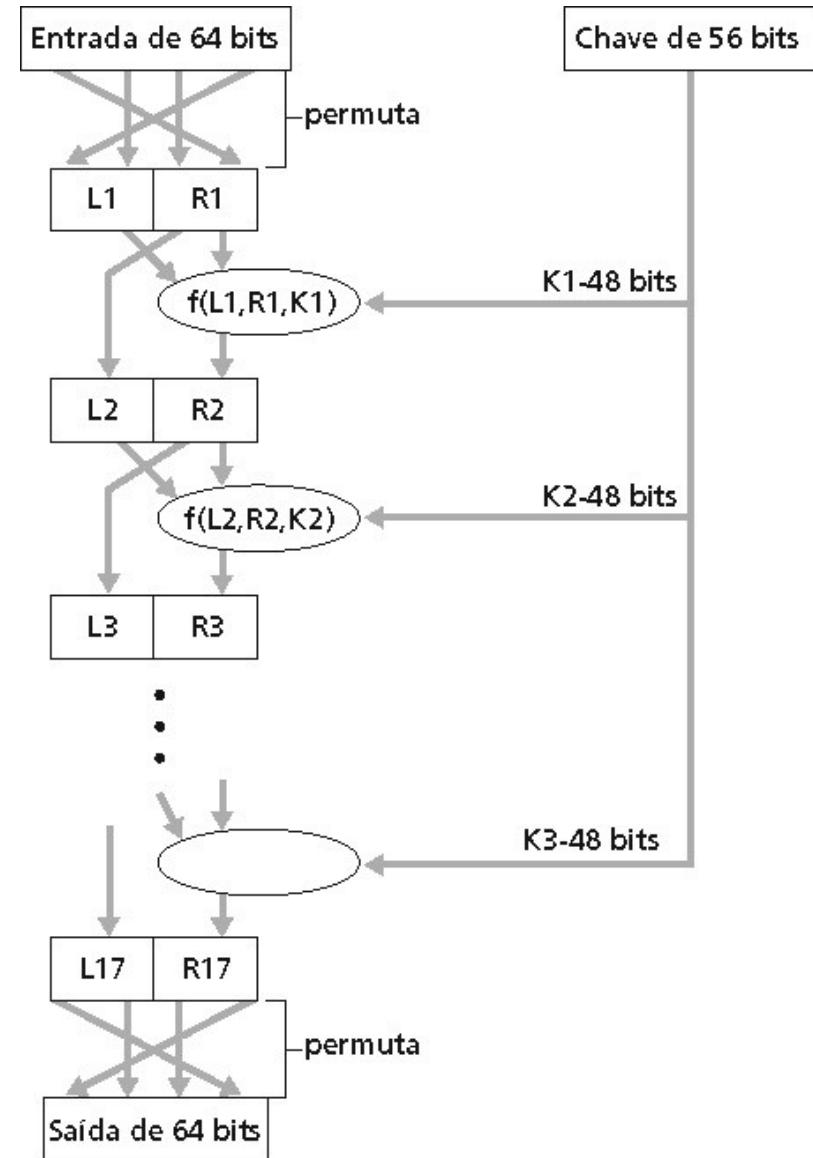


Operação do DES

permutação inicial

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

permutação final



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

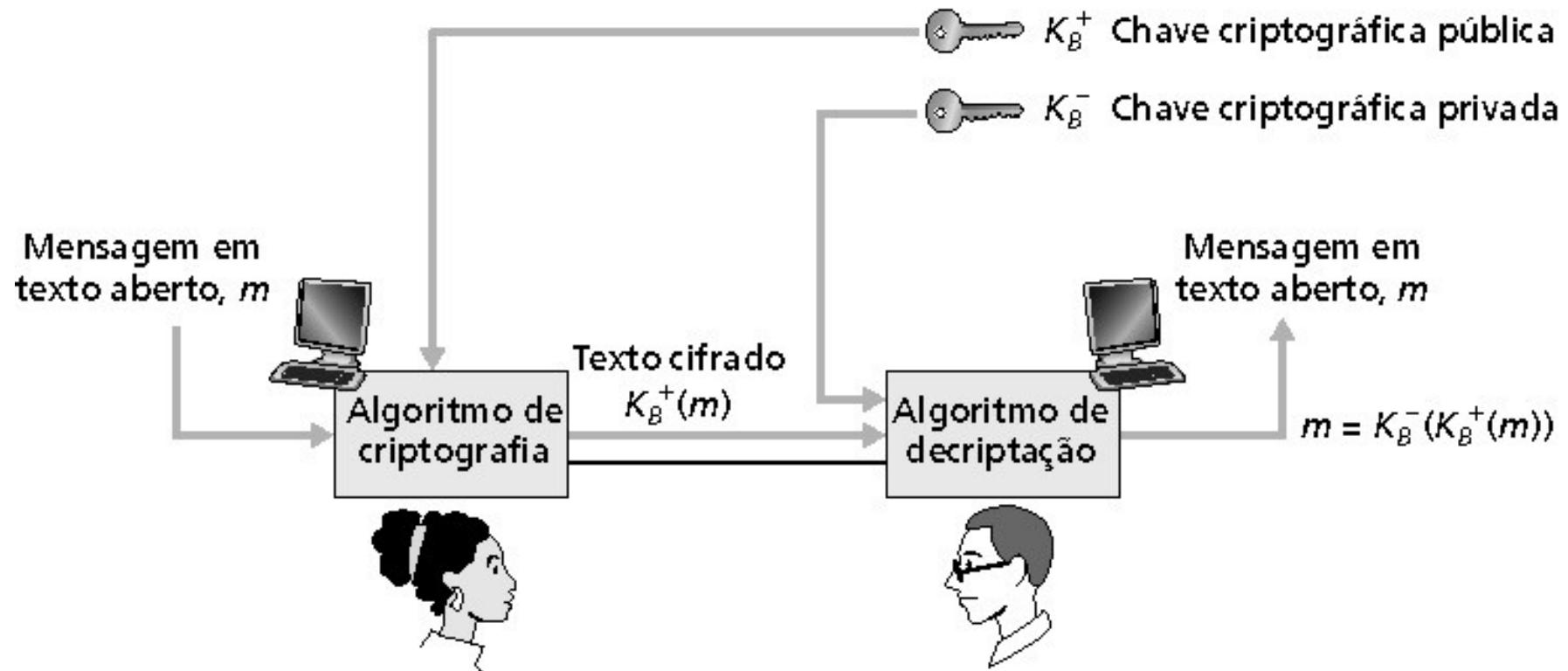


Chave simétrica

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

*Chave pública*

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



Duas exigências correlatas:

- 1 necessita $d_B(\cdot)$ e $e_B(\cdot)$ tal que

$$d_B(e_B(m)) = m$$

- 2 necessita chaves pública e privada para $d_B(\cdot)$ e $e_B(\cdot)$

RSA: Algoritmo de Rivest, Shamir, Adleman

8 RSA: Escolhendo as chaves

1. Encontre dois números primos grandes p, q .
(ex., 1.024 bits cada um)
2. Calcule $n = pq$, $z = (p - 1)(q - 1)$
3. Escolha e (com $e < n$) que não tenha fatores primos em comum com z. (e, z são “primos entre si”).
4. Escolha d tal que $ed - 1$ seja exatamente divisível por z.
(em outras palavras: $ed \bmod z = 1$).
5. Chave pública é (n, e) . Chave privada é (n, d) .

 K_B^+ K_B^- 

0. Dado (n, e) e (n, d) como calculados antes.

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

$$c = m^e \text{ mod } n \quad (\text{i.e., resto quando } m^e \text{ é dividido por } n).$$

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

$$m = c^d \text{ mod } n \quad (\text{i.e., resto quando } c^d \text{ é dividido por } n).$$

Mágica
acontece!

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

c

RSA exemplo:

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

$e = 5$ (assim e , z são primos entre si).

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

criptografia:

<u>letra</u>	<u>m</u>	<u>m^e</u>	<u>$c = m^e \bmod n$</u>
l	12	1524832	17

decriptografia:

<u>c</u>	<u>c^d</u>	<u>$m = c^d \bmod n$</u>	<u>letra</u>
17	481968572106750915091411825223072000	12	l

RSA: Por que $m = (m \bmod n)^d \bmod n$

Resultado da teoria dos números: Se p, q são 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 teorema apresentado acima)

$$= m^1 \bmod n$$

(pois nós escolhemos ed divisível por $(p-1)(q-1)$ com resto 1)

$$= m$$



RSA: outra propriedade importante

A propriedade a seguir será *muito* útil mais tarde:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{usa chave pública primeiro,}} = \underbrace{K_B^+(K_B^-(m))}_{\text{seguida pela chave privada}}$$

usa chave pública primeiro,
seguida pela chave privada

usa chave privada primeiro,
seguida pela chave pública

O resultado é o mesmo!



- 8.1 O que é segurança?
- 8.2 Princípios da criptografia
- 8.3 Autenticação
- 8.4 Integridade
- 8.5 Distribuição de chaves e certificação
- 8.6 Controle de acesso: firewalls
- 8.7 Ataques e medidas de defesa
- 8.8 Segurança em muitas camadas



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

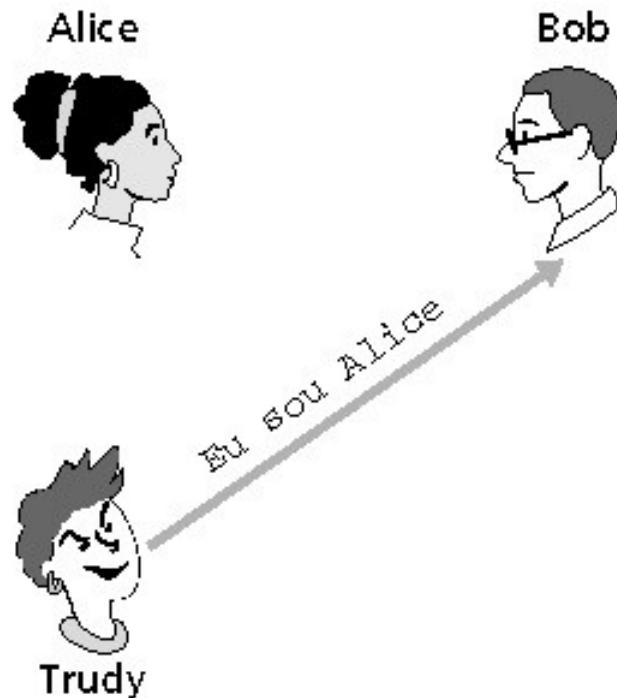
Protocolo ap1.0: Alice diz “Eu sou Alice”



Cenário de falha??

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

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



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

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



Cenário de falha??



Autenticação: outra tentativa (cont.)

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



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

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

Cenário de falha??



Legenda:



Gravador

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

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



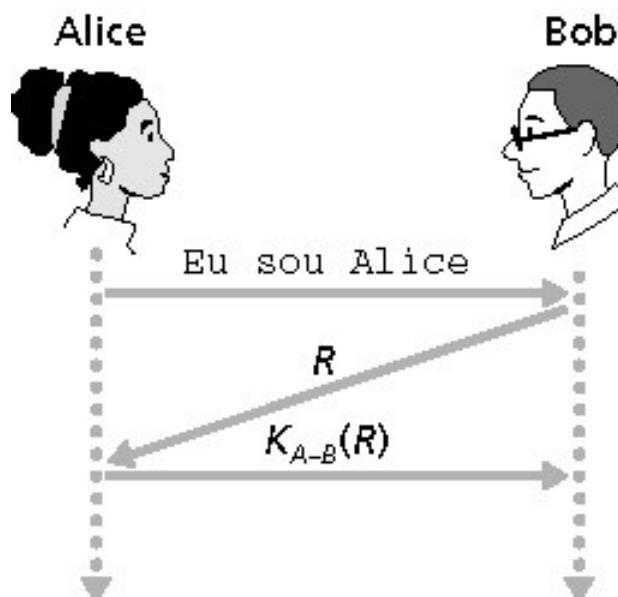
Protocolo ap3.1: Alice diz “Eu sou Alice” e envia sua senha secreta *criptografada* para prová-lo.



Meta: evitar ataque de reprodução (playback).

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

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



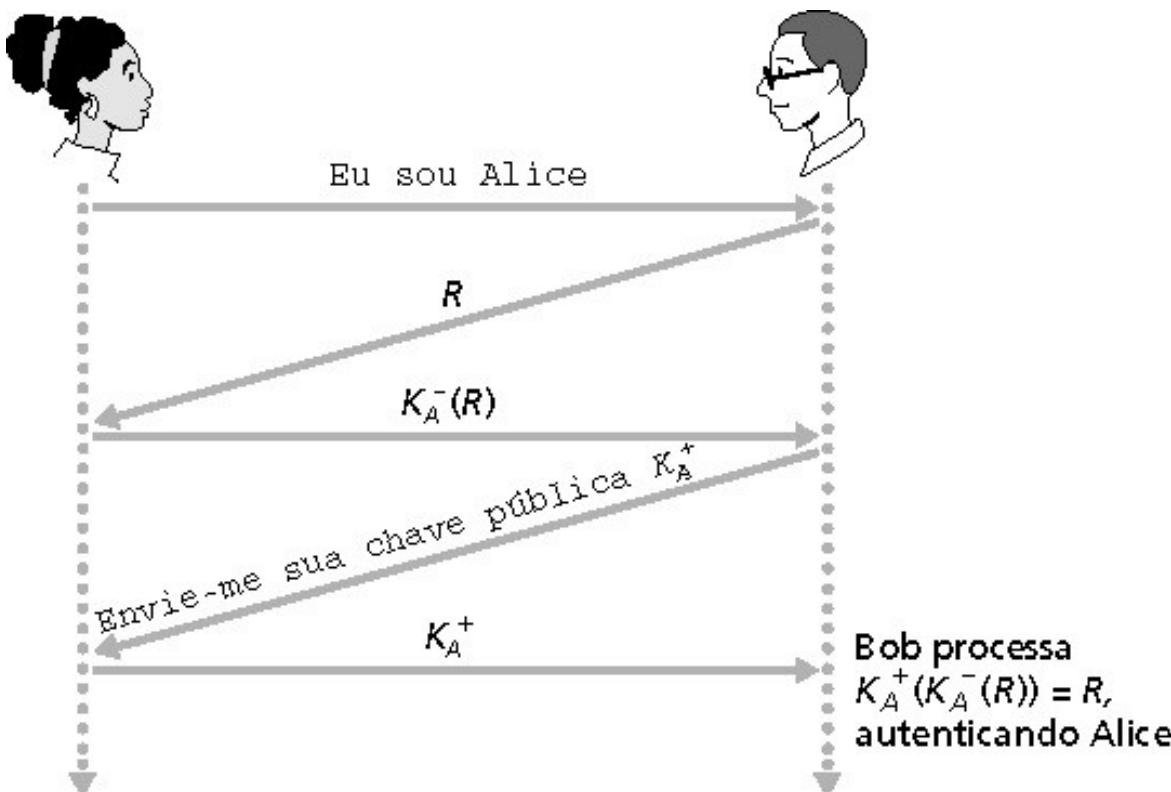
Falhas, problemas?

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

ap4.0 exige chave secreta compartilhada.

- é possível autenticar usando técnicas de chave pública?

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



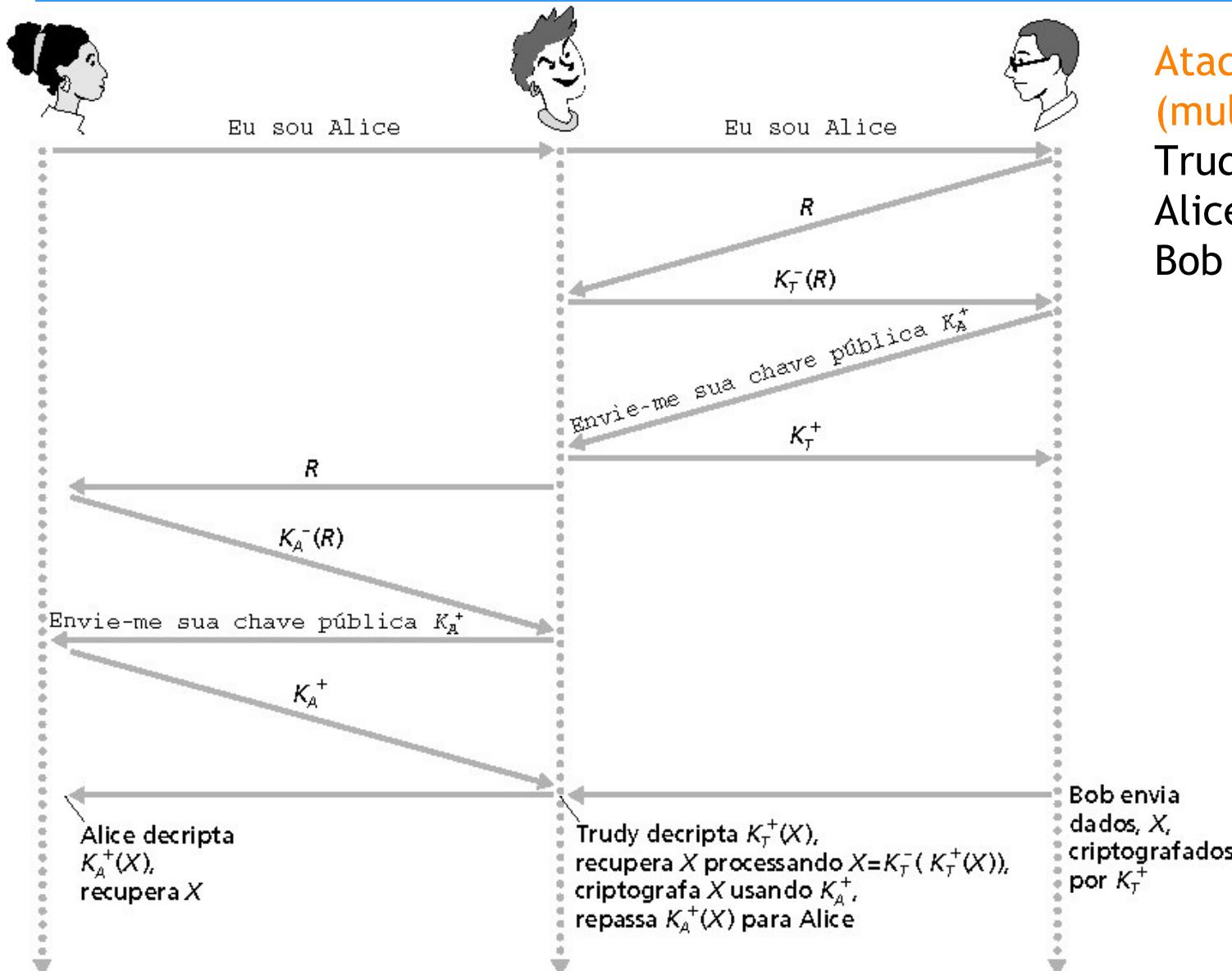
Bob calcula

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

e sabe que apenas Alice poderia ter a chave privada, que criptografou R desta maneira

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

ap5.0: falha de segurança



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

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

Difícil de detectar:

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



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



Técnica criptográfica análoga às assinaturas manuais.

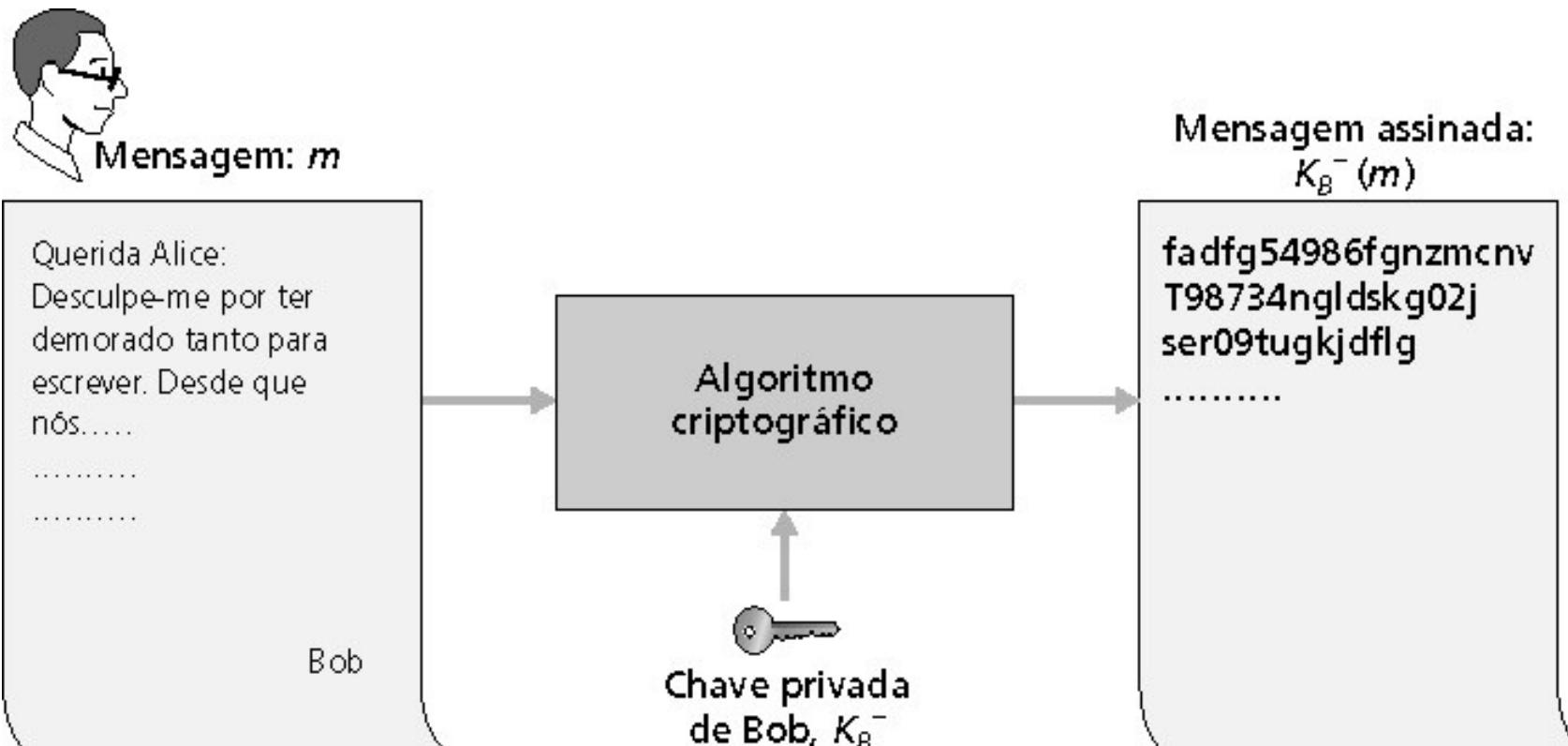
- Transmissor (Bob) assina digitalmente um documento, estabelecendo que ele é o autor/criador.
- **Verificável, não forjável:** receptor (Alice) pode verificar que Bob, e ninguém mais, assinou o documento.



8 Assinaturas digitais (cont.)

Assinatura digital simples para mensagem m :

- Bob assina m criptografado com sua chave privada K_B , criando a mensagem “assinada”, $K_B^{-1}(m)$



- 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 para $K_B(m)$ e então verifica que $K_B(K_B(m)) = m$. + -
- Se $K_B(K_B(\tilde{m})) = m$, quem quer que tenha assinado m deve possuir a chave privada de Bob.

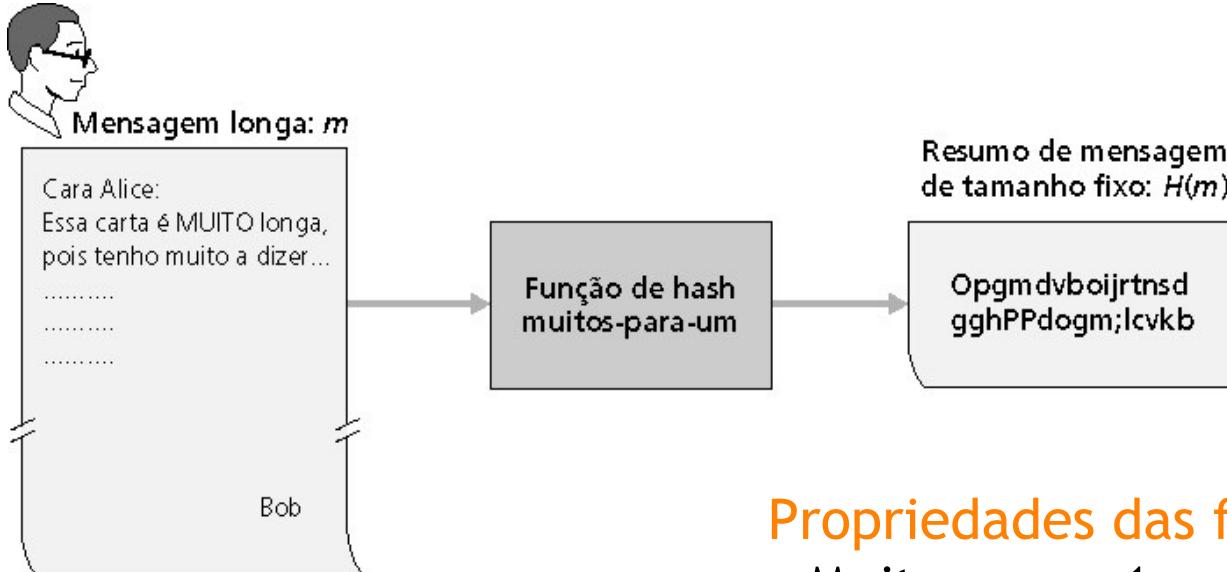
Alice verifica então que:

- Bob assinou m .
- ninguém mais assinou m .
- Bob assinou m e não m' .

Não-repúdio:

- Alice pode levar m e a assinatura $K_B(m)$ a um tribunal para provar que Bob assinou m .





Computacionalmente caro
criptografar com chave pública
mensagens longas

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

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

Propriedades das funções de Hash:

- Muitas-para-1
- Produz um resumo da mensagem
de tamanho fixo (impressão
digital)
- Dado um resumo da mensagem
 x , é computacionalmente
impraticável encontrar m
tal que $x = H(m)$

Soma de verificação da Internet: função de Hash criptográfico pobre

Verificação da Internet possui algumas propriedades de função de hash:

- Produz resumo de tamanho fixo (soma de 16 bits) de mensagem
- É muitos-para-um

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

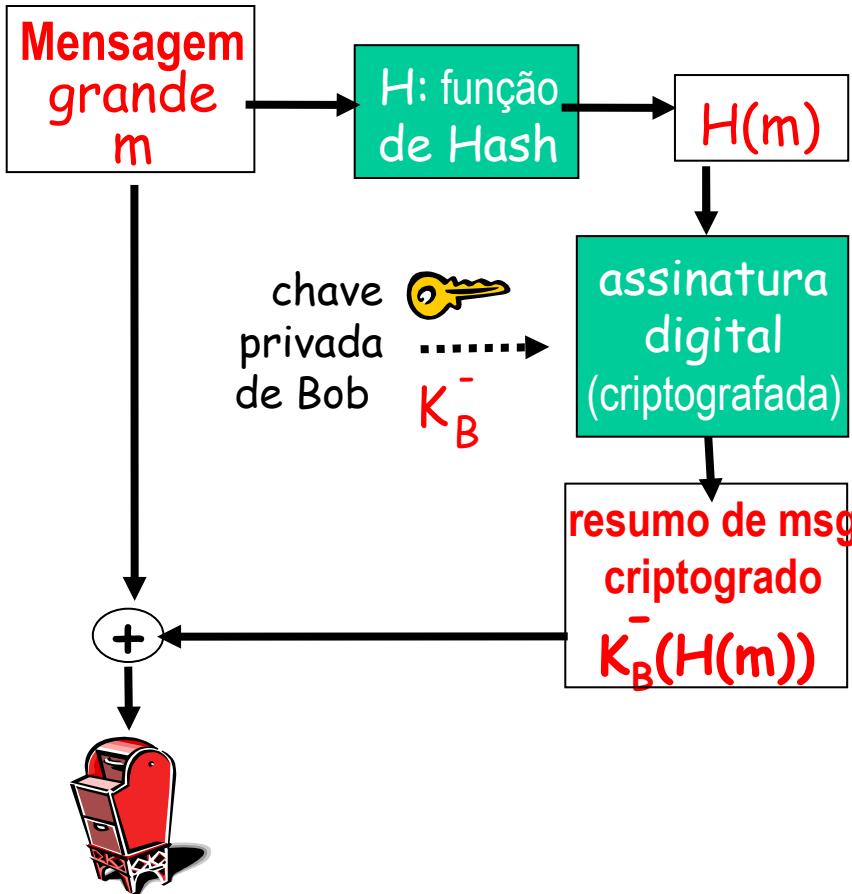
mensagem	formato ASCII
I O U 1	49 4F 55 31
0 0 . 9	30 30 2E 39
9 B O B	39 42 D2 42
	<hr/>
	B2 C1 D2 AC

mensagem	formato ASCII
I O U 9	49 4F 55 <u>39</u>
0 0 . 1	30 30 2E <u>31</u>
9 B O B	39 42 D2 42
	<hr/>
	B2 C1 D2 AC

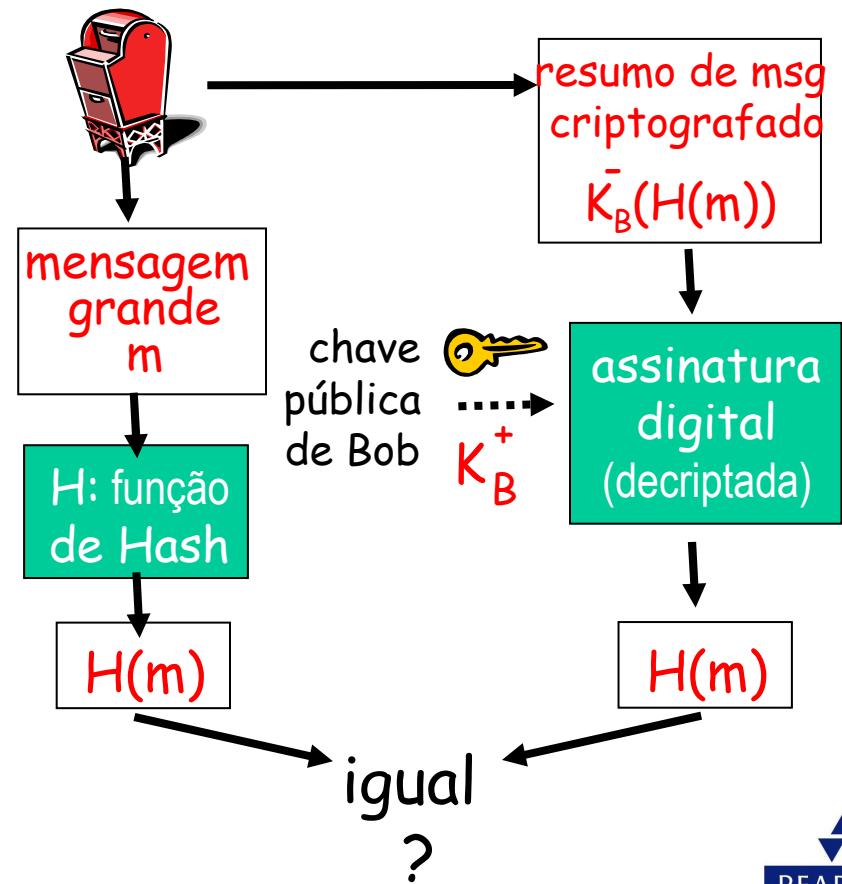
mensagens diferentes
mas resumos idênticos!

Assinatura digital = resumo assinado de mensagem

Bob envia mensagem digitalmente assinada:



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



- MD5 é a função de hash mais usada (RFC 1321)
 - Calcula resumo de 128 bits da mensagem num processo de 4 etapas
 - Uma cadeia arbitrária X de 128 bits parece difícil de construir uma mensagem m cujo hash MD5 é igual ao hash de um cadeia X.
- SHA-1 também é usado.
 - Padrão dos EUA [NIST, FIPS PUB 180-1]
 - Resumo de mensagem de 160 bits



- 8.1 O que é segurança?
- 8.2 Princípios da criptografia
- 8.3 Autenticação
- 8.4 Integridade
- 8.5 Distribuição de chaves e certificação
- 8.6 Controle de acesso: firewalls
- 8.7 Ataques e medidas de defesa
- 8.8 Segurança em muitas camadas



Problema da chave simétrica:

- Como duas entidades estabelecem um segredo mútuo sobre a rede?

Solução:

- Centro de distribuição de chaves confiável (KDC) atuando como intermediário entre entidades

Problema da chave pública:

- Quando Alice obtém a chave pública de Bob (de um site web site, e-mail, diskette), como ela sabe que é a chave pública de Bob e não de Trudy?

Solução:

- Autoridade de certificação confiável (CA)



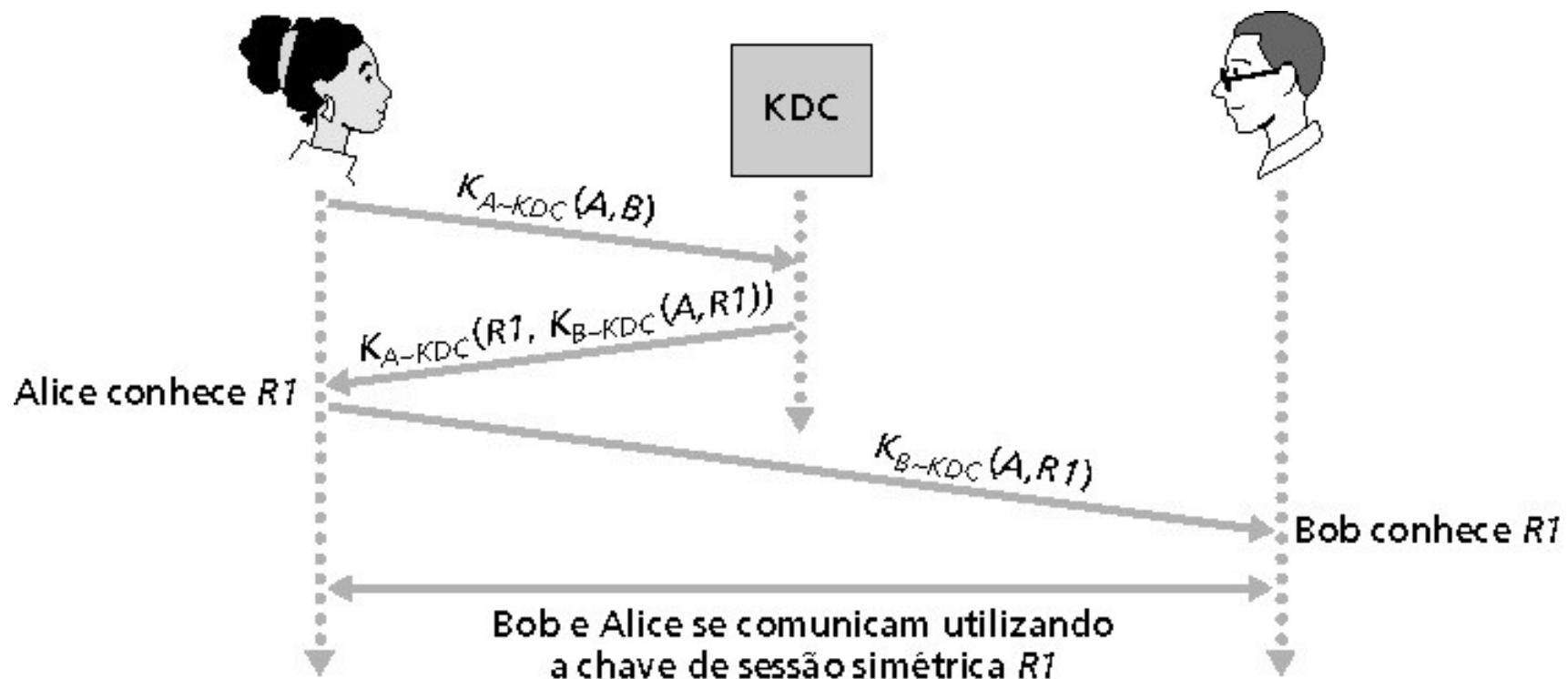
Centro de distribuição de chave (KDC)

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



Centro de distribuição de chave (KDC)

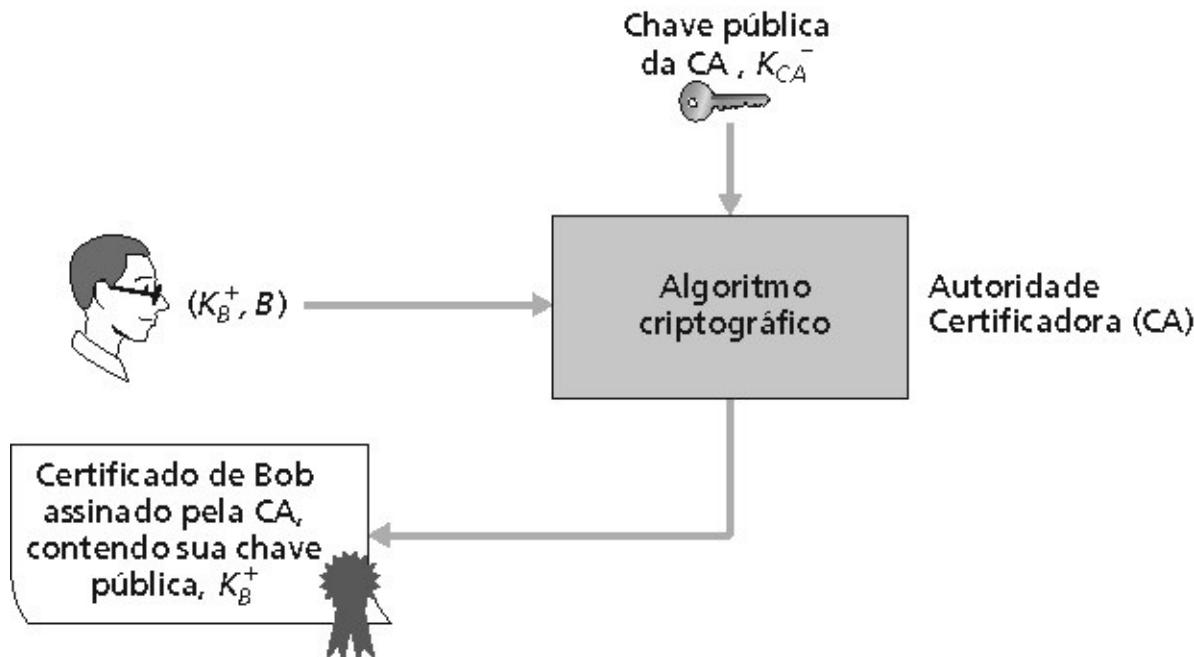
P.: Como o KDC permite que Bob e Alice determinem uma chave simétrica comum para comunicarem-se entre si?



Autoridades certificadoras

Autoridade certificadora (CA): associa uma chave pública a uma entidade em particular, E

- E (pessoa, roteador) registra sua chave pública com CA
 - E fornece “prova de identidade” ao CA
 - CA cria um certificado associando E a sua chave pública
 - Certificado contendo a chave pública de E digitalmente assinada pela CA
 - CA diz “esta é a chave pública de E”

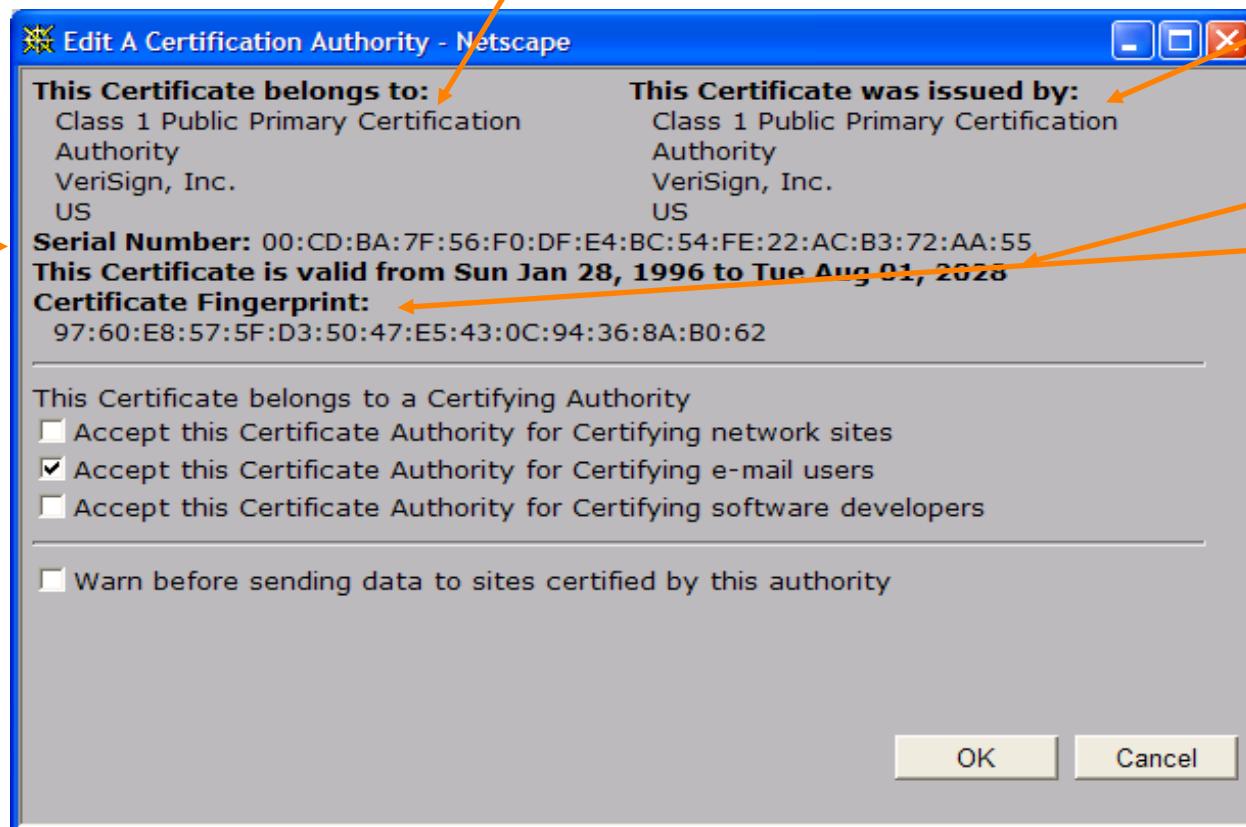


- Quando Alice quer a chave publica de Bob:
- Obtém o certificado de Bob (de Bob ou em outro lugar).
- Aplica a chave pública da CA ao certificado de Bob, obtém a chave pública de Bob



Um certificado contém:

- Número serial (único para o emissor)
- Informação sobre o dono do certificado, incluindo o algoritmo e o valor da própria chave (não mostrada)



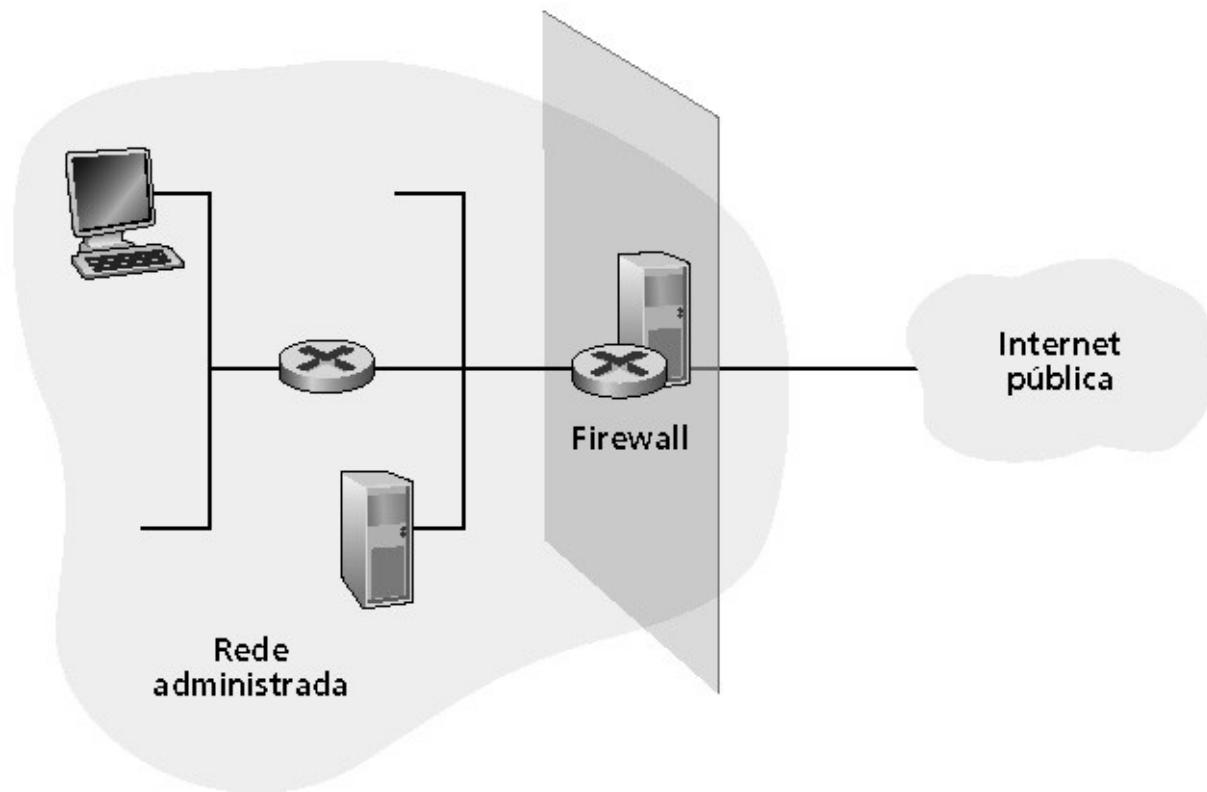
- Informação sobre o emissor do certificado
- Data de validade
- Assinatura digital do emissor

- 8.1 O que é segurança?
- 8.2 Princípios da criptografia
- 8.3 Autenticação
- 8.4 Integridade
- 8.5 Distribuição de chaves e certificação
- 8.6 Controle de acesso: firewalls
- 8.7 Ataques e medidas de defesa
- 8.8 Segurança em muitas camadas



Firewall

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



Previne ataques de negação de serviço:

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

Previne modificações e acessos ilegais aos dados internos.

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

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

Dois tipos de firewalls:

- Nível de aplicação
- Filtro de pacotes

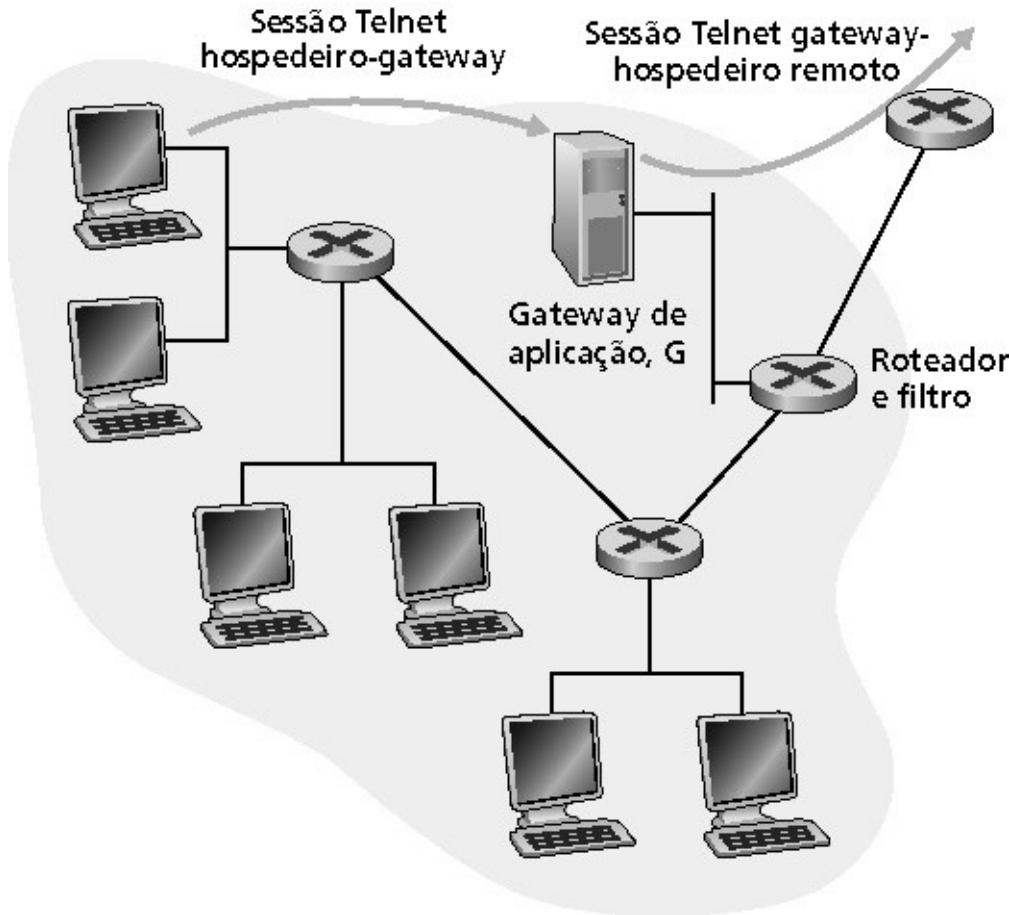


- Rede interna conectada à Internet via **roteador firewall**
- Roteador **filtre pacotes**; decisão de enviar ou descartar pacotes baseia-se em:
 - Endereço IP de origem, endereço IP de destino
 - Número de portas TCP/UDP de origem e de destino
 - Tipo de mensagem ICMP
 - Bits TCP SYN e ACK



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





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

- **IP spoofing:** roteador não pode saber se os dados realmente vêm da fonte declarada
- Se múltiplas aplicações requerem um tratamento especial, cada uma deve ter seu próprio gateway de aplicação
- O software cliente deve saber como contatar o gateway
Ex., deve configurar o endereço IP do proxy no browser Web
- Filtros muitas vezes usam uma regra radical para UDP: bloqueiam tudo ou deixam passar tudo
- Compromisso: **grau de comunicação com mundo exterior versus nível de segurança**
- Muitos sites altamente protegidos mesmo assim sofrem ataques



- 8.1 O que é segurança?
- 8.2 Princípios da criptografia
- 8.3 Autenticação
- 8.4 Integridade
- 8.5 Distribuição de chaves e certificação
- 8.6 Controle de acesso: firewalls
- 8.7 Ataques e medidas de defesa
- 8.8 Segurança em muitas camadas



Mapeamento:

- Antes do ataque: “teste a fechadura” - descubra quais serviços estão implementados na rede
- Use ping para determinar quais hospedeiros têm endereços acessíveis na rede
- Varredura de portas: tente estabelecer conexões TCP com cada porta em seqüência (veja o que acontece)
vnmap (<http://www.insecure.org/nmap/>) mapeador: “exploração de rede e auditoria de segurança”

Contramedidas?



PEARSON
Addison Wesley

Mapeamento: contramedidas

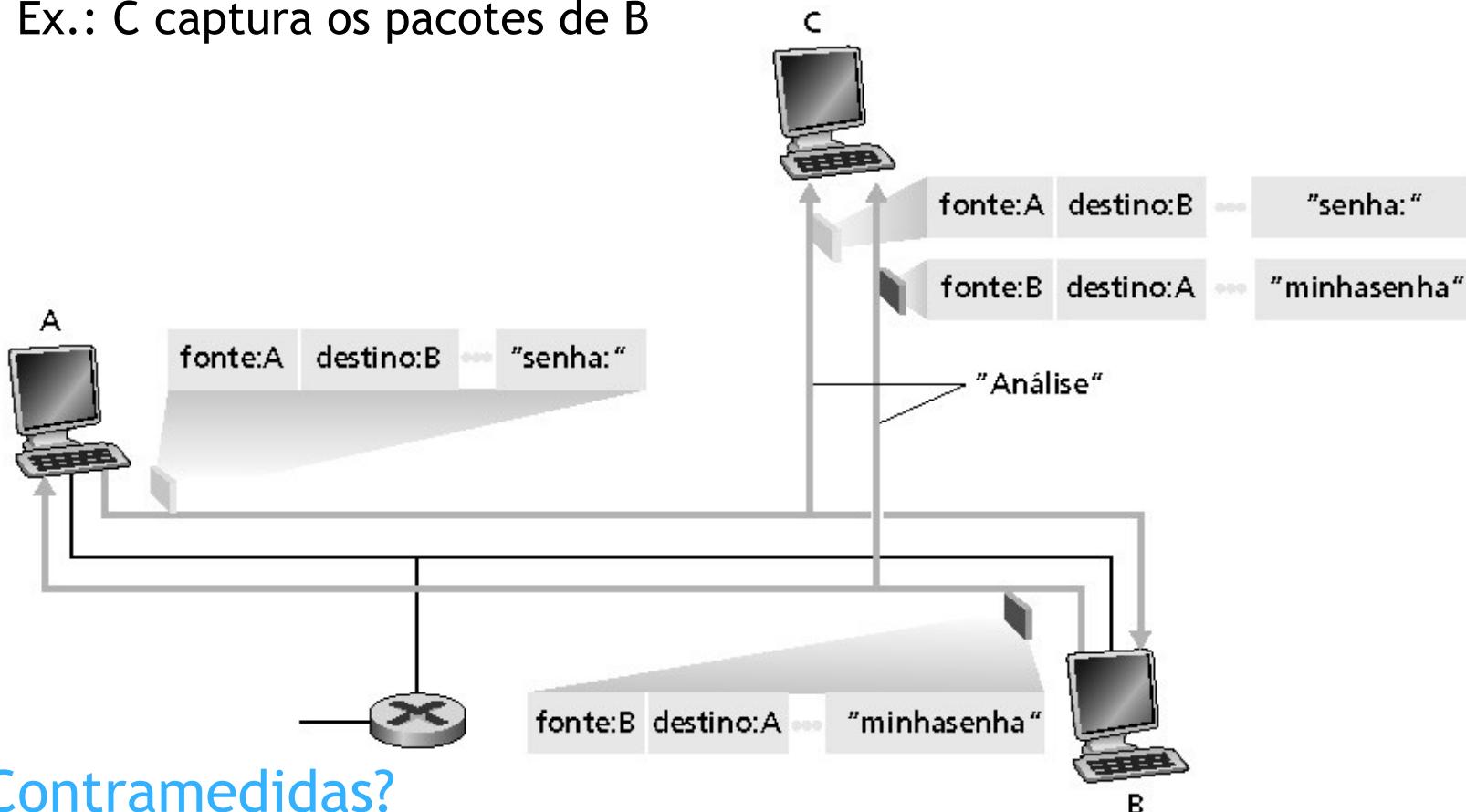
- Grave o tráfego entrando na rede
- Examine atividades suspeitas (endereços IP e portas sendo varridas seqüencialmente)



Ameaças de segurança na Internet (cont.)

Packet sniffing:

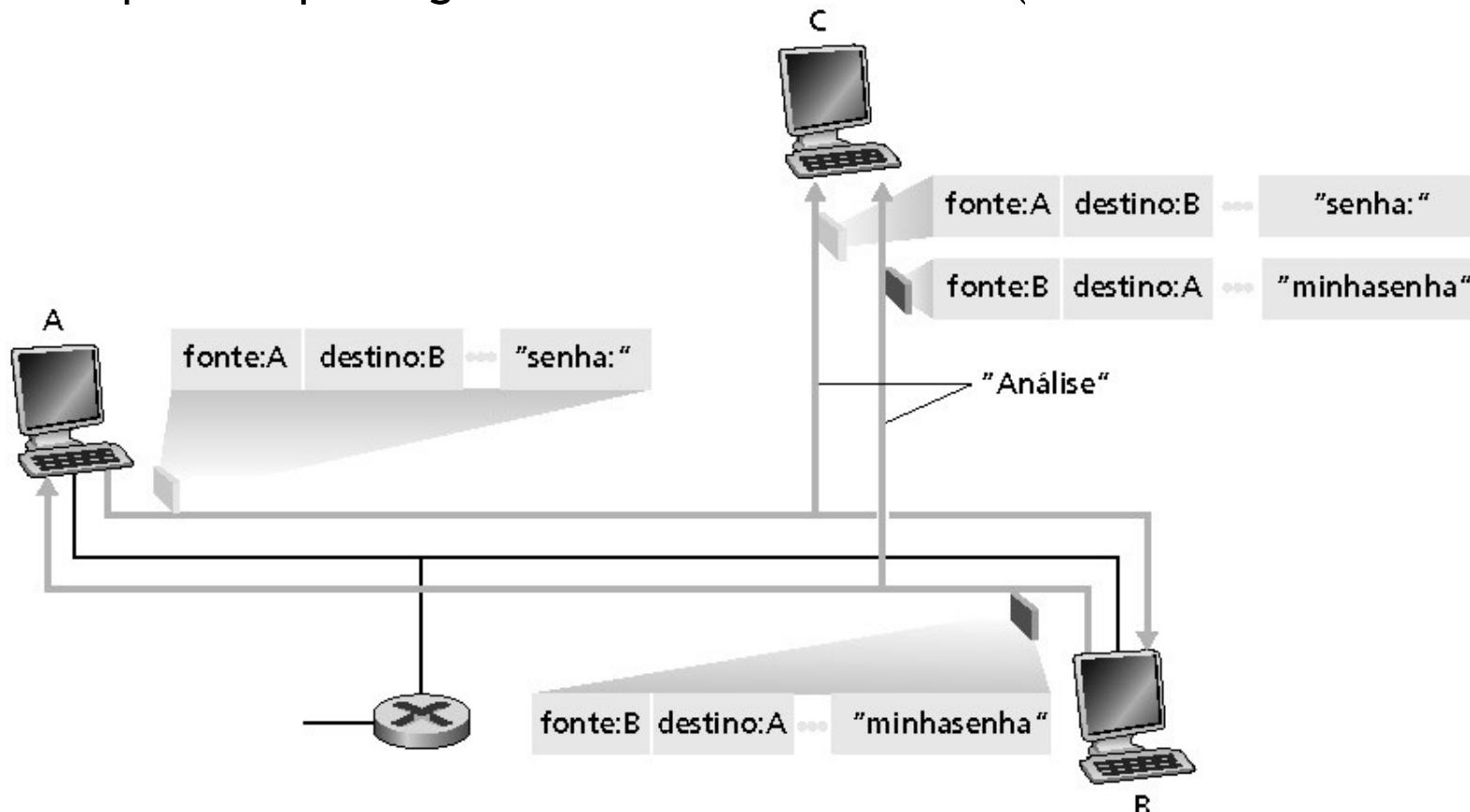
- Meio broadcast
- NIC em modo promíscuo lêem todos os pacotes que passam
- Pode ler todos os dados não criptografados (ex., senhas)
- Ex.: C captura os pacotes de B



Contramedidas?

Packet sniffing: contramedidas

- Todos os hospedeiros na organização executam software que examina periodicamente se a interface do hospedeiro está operando em modo promíscuo
- Um hospedeiro por segmento de meio broadcast (Ethernet comutada no hub)



IP Spoofing:

- Pode gerar pacotes IP “puros” diretamente da aplicação, colocando qualquer valor do endereço IP no campo de endereço de origem
- Receptor não sabe se a fonte é verdadeira ou se foi forjada
Ex.: C finge ser B



IP Spoofing: filtro de entrada

- Roteadores não devem repassar pacotes para a saída com endereço de origem inválido (ex., endereço de fonte do datagrama fora do endereço da rede local)
- Grande, mas filtros de entrada não podem ser obrigatórios para todas as redes



Negação de serviço (DoS):

- Inundação de pacotes maliciosamente gerados invade o receptor receiver
- DoS Distribuído (DDoS): múltiplas fontes coordenadas atacam simultaneamente o receptor
ex.: C é um hospedeiro remoto ataca A com inundação de SYN



Negação de serviço (DoS): contramedidas

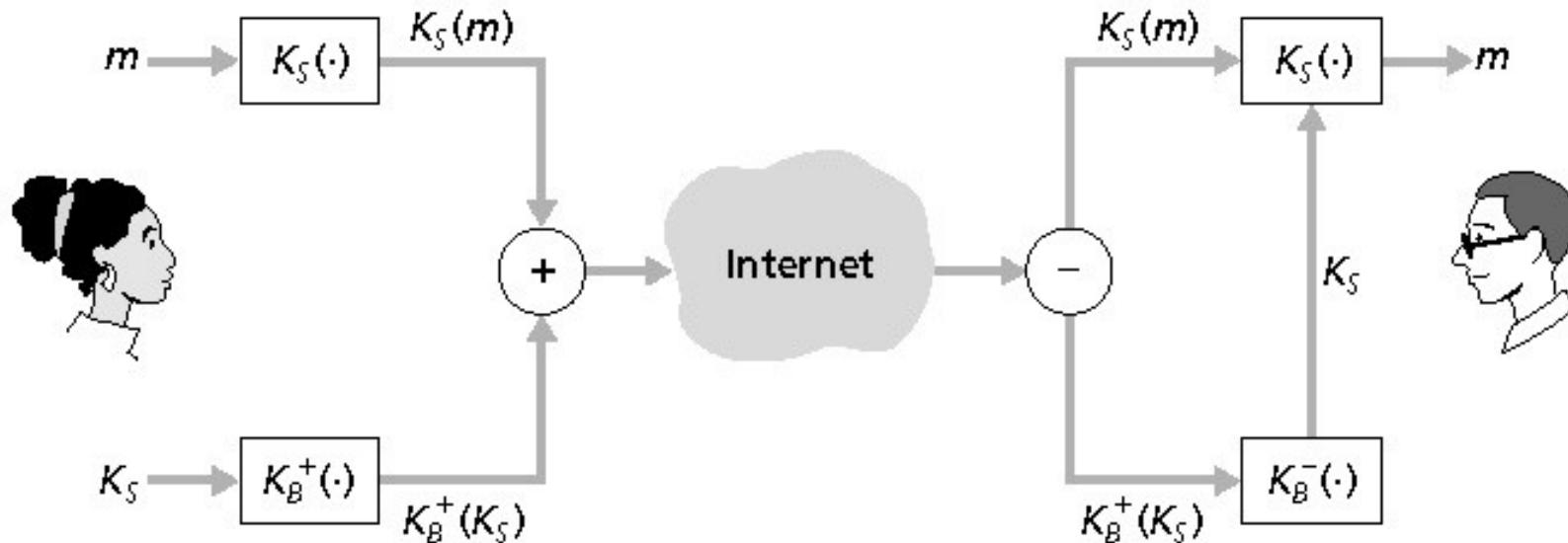
- **Filtragem** de pacotes de inundação (ex., SYN) antes de atingirem o alvo: corta os pacotes bons e os maus
- **Rastrear** em busca da fonte da inundação (mais provavelmente uma máquina inocente que foi invadida)



- 8.1 O que é segurança?
- 8.2 Princípios da criptografia
- 8.3 Autenticação
- 8.4 Integridade
- 8.5 Distribuição de chaves e certificação
- 8.6 Controle de acesso: firewalls
- 8.7 Ataques e medidas de defesa
- 8.8 Segurança em muitas camadas
 - 8.8.1 e-mail seguro
 - 8.8.2 sockets seguros
 - 8.8.3 IPsec
 - 8.8.4 segurança em 802.11



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



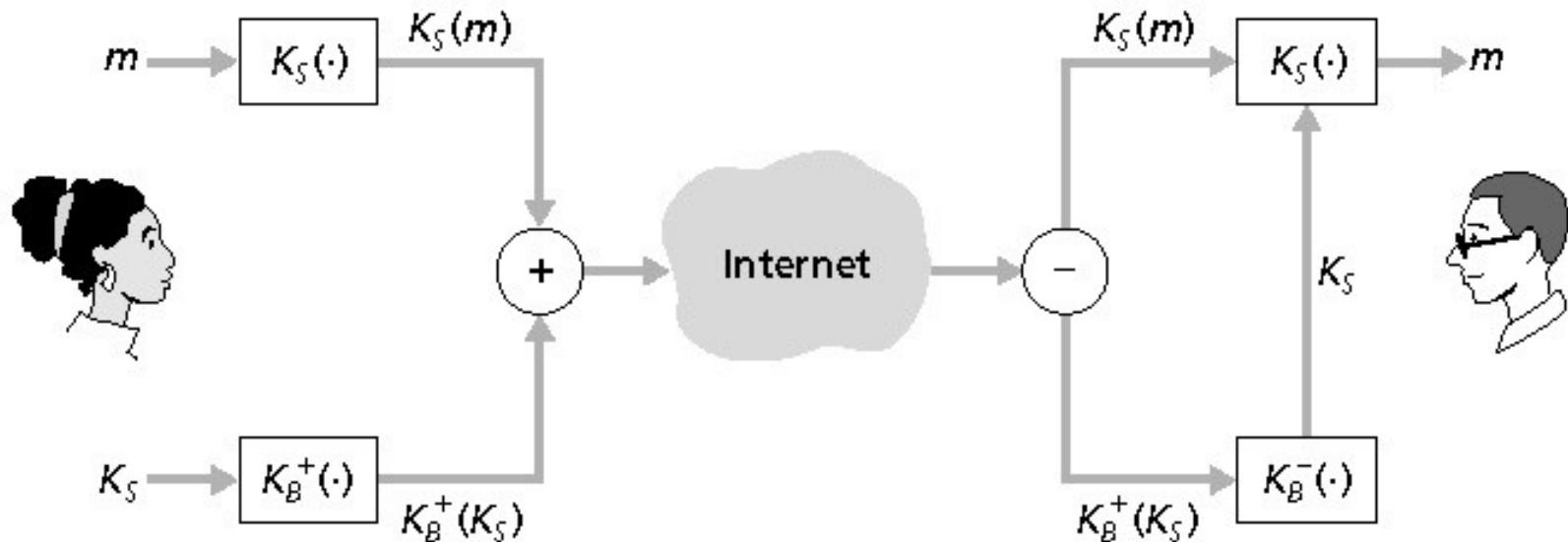
Alice envia uma mensagem de e-mail, m

Alice:

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

Bob recebe uma mensagem de e-mail, m

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



Alice envia uma mensagem de e-mail, m

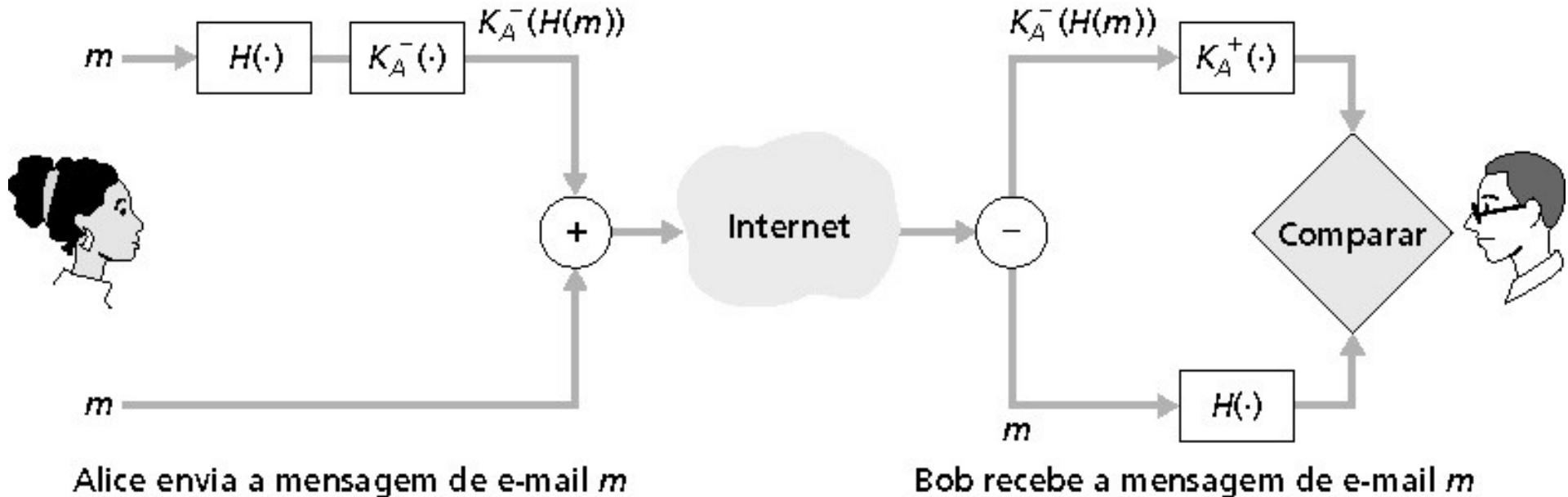
Bob recebe uma mensagem de e-mail, m

Bob:

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

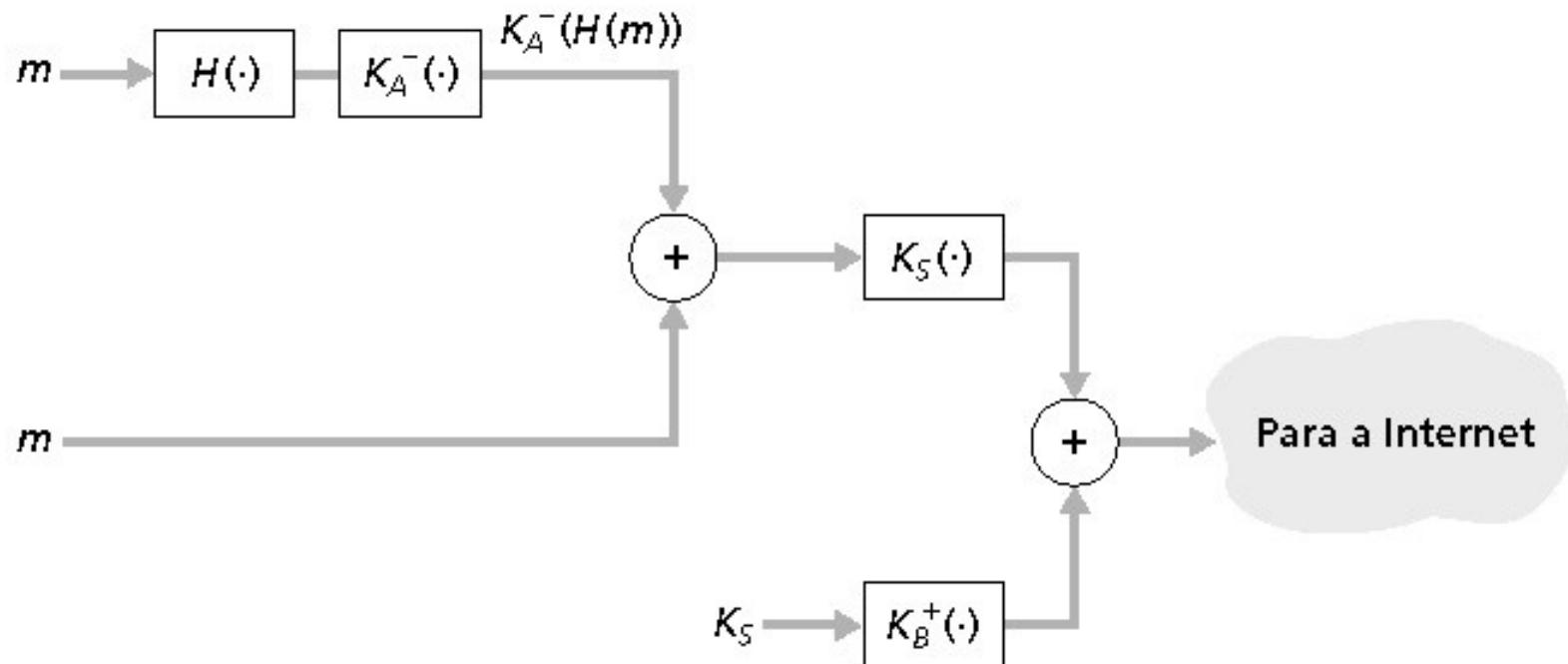


- Alice quer fornecer autenticação de emissor e integridade de mensagem.



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

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



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



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

Uma mensagem PGP:

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

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

---BEGIN PGP SIGNATURE---
Version: PGP 5.0
Charset: noconv
yHJRHHGJGhgg/12EpJ+lo8gE4vB3mqJhFE
vZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```



- Segurança de camada de transporte para qualquer aplicação baseada no TCP usando serviços SSL
- Usado entre browsers Web e servidores para comércio eletrônico (shttp)

Serviços de segurança:

- Autenticação de servidor
- Criptografia de dados
- Autenticação de cliente (opcional)
- Servidor de autenticação:
 - Browser com SSL habilitado inclui chaves públicas para CA confiáveis
 - Browser pede certificado do servidor, emitido pela CA confiável
 - Browser usa chave pública da CA para extrair a chave pública do servidor do certificado
- Verifique o menu de segurança do seu browser para ver suas CAs confiáveis



- Sessão SSL criptografada:
- Browser gera *chave de sessão simétrica*, criptografa essa chave com a chave pública do servidor e a envia para o servidor
- Usando a chave privada, o servidor recupera a chave de sessão
- Browser e servidor conhecem agora a chave de sessão
 - Todos os dados são enviados para o socket TCP (pelo cliente e pelo servidor) criptografados com a chave de sessão
- SSL: base do padrão transport layer security (TLS) do IETF
- SSL pode ser usado por aplicações fora da Web; ex., IMAP.
- Autenticação do cliente pode ser feita com certificados do cliente

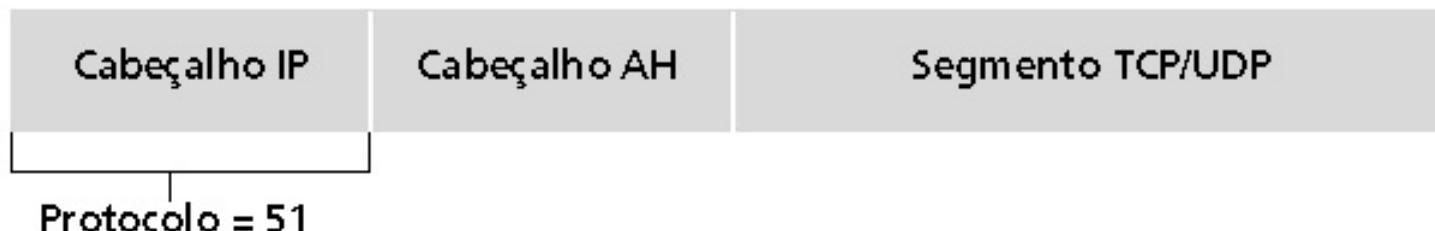


- Confidencialidade na camada de rede:
 - Hospedeiro transmissor criptografa os dados no datagrama IP
 - Segmentos TCP e UDP; mensagens ICMP e SNMP
- Autenticação na camada de rede
 - Hospedeiro de destino pode autenticar o endereço IP da origem
- Dois protocolos principais:
 - Protocolo de autenticação de cabeçalho (AH)
 - Protocolo de encapsulamento seguro dos dados (ESP)
- Tanto o AH quanto o ESP realizam uma associação da fonte e do destino:
 - Cria um canal lógico de camada de rede denominado associação de segurança (SA - Security association)
- Cada SA é unidirecional
- Unicamente determinado por:
 - Protocolo de segurança (AH ou ESP)
 - Endereço IP de origem
 - ID de conexão de 32 bits

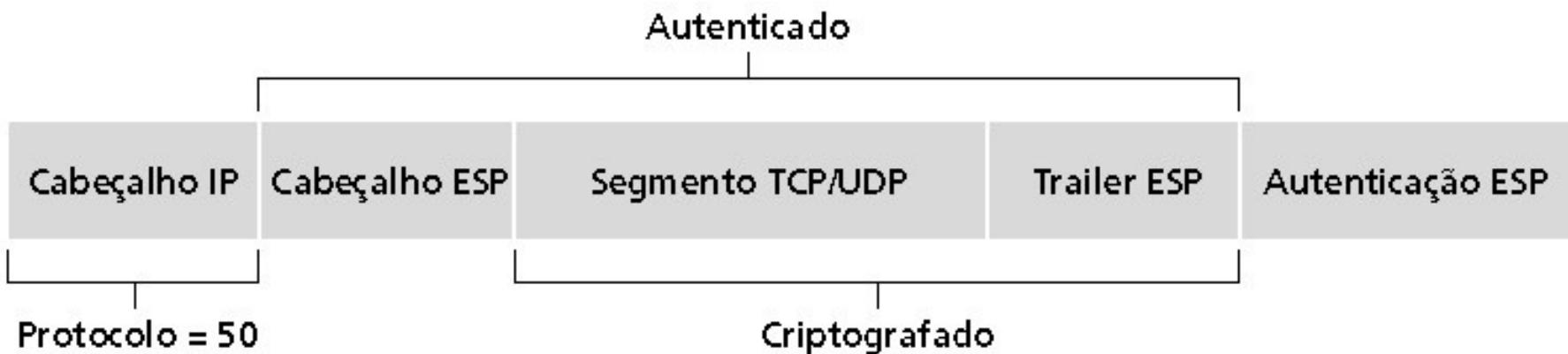
- Oferece autenticação de fonte, integridade dos dados, mas não confidencialidade
- Cabeçalho AH é inserido entre o cabeçalho IP e o campo de dados
- Campo de protocolo 51
- Roteadores intermediários processam o pacote na forma usual

Cabeçalho AH inclui:

- Identificador de conexão
- Dados de autenticação de dados: resumo da mensagem assinado pela fonte calculado sobre o datagrama IP original.
- Campo de próximo cabeçalho: especifica tipo de dado (ex.: TCP, UDP, ICMP)



- Oferece confidencialidade, autenticação de hospedeiro e integridade dos dados
- Dados e trailer ESP são criptografados
- Campo de próximo cabeçalho vai no trailer ESP
- Campo de autenticação do ESP é similar ao campo de autenticação do AH
- Protocolo = 50



- **Guerra:** uma pesquisa na área da Baía de San Francisco procurou encontrar redes 802.11 acessíveis
 - Mais de 9.000 acessíveis a partir de áreas públicas
 - 85% não usam criptografia nem autenticação
 - Packet-sniffing e vários outros ataques são fáceis!
- **Tornando 802.11 seguro**
 - Criptografia, autenticação
 - Primeira tentativa no padrão 802.11: Wired Equivalent Privacy (WEP): um fracasso
 - Tentativa atual: 802.11i

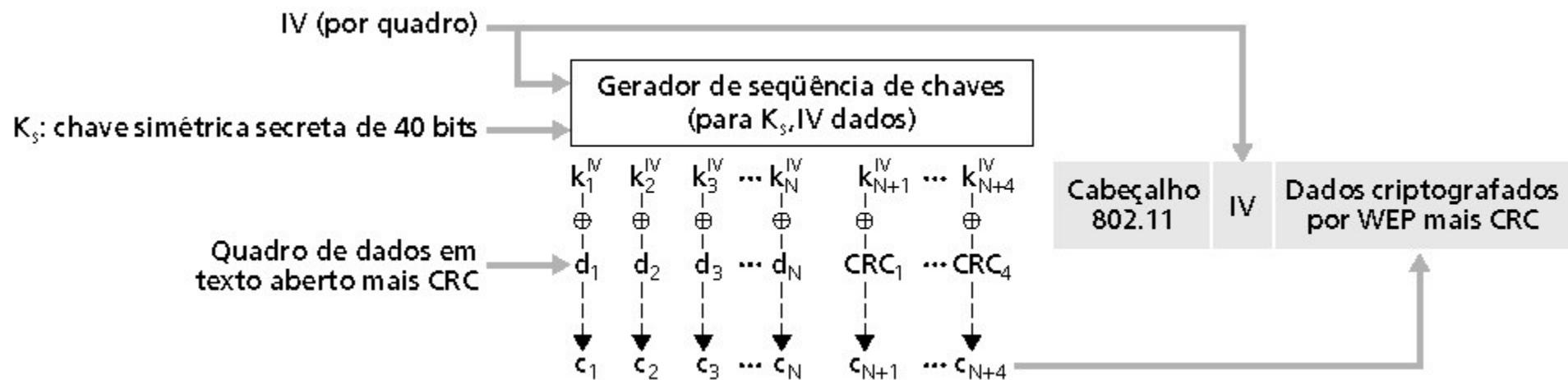


- Autenticação como no protocolo *ap4.0*
 - Hospedeiro solicita autenticação do ponto de acesso
 - Ponto de acesso envia um nonce de 128 bits
 - Hospedeiro criptografa o nonce usando uma chave simétrica compartilhada
 - Ponto de acesso decodifica o nonce, autentica o hospedeiro
- Faltam mecanismos de distribuição de chaves
- Autenticação: conhecer a chave compartilhada é o bastante



- Hospedeiro e AP compartilham uma chave simétrica de 40 bits (semipermanente)
- Hospedeiro acrescenta vetor de inicialização de 24 bits (IV) para criar uma chave de 64 bits
- A chave de 64 bits é usada para gerar uma seqüência de chaves, k_i^{IV}
- k_i^{IV} é usada para criptografar o i -ésimo byte, d_i , no quadro:
$$c_i = d_i \text{ XOR } k_i^{IV}$$
- IV e bytes criptografados, c_i , são enviados no quadro





Furo de segurança:

- 24 bits IV, um IV por quadro, -> IV's são reusados eventualmente
- IV é transmitido aberto -> reuso do IV é detectado

Ataque:

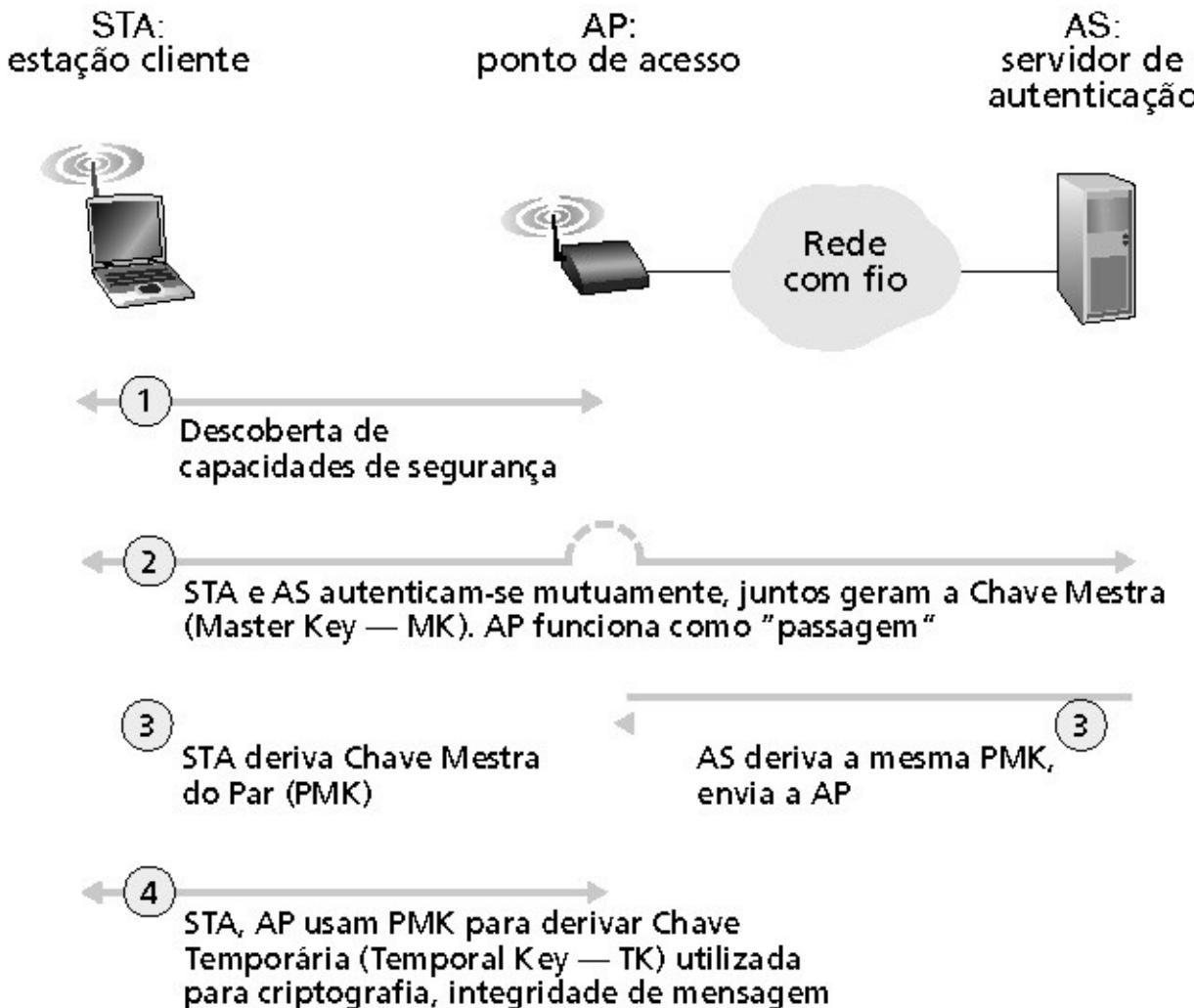
- Trudy provoca Alice para criptografar um texto conhecido $d_1 d_2 d_3 d_4 \dots$
- Trudy vê: $c_i = d_i \text{ XOR } k_i^{\text{IV}}$
- Trudy conhece c_i e d_i , logo pode calcular k_i^{IV}
- Trudy sabe a seqüência de chaves criptográfica $k_1^{\text{IV}} k_2^{\text{IV}} k_3^{\text{IV}} \dots$
- Próxima vez que IV for usado, Trudy pode decodificar!



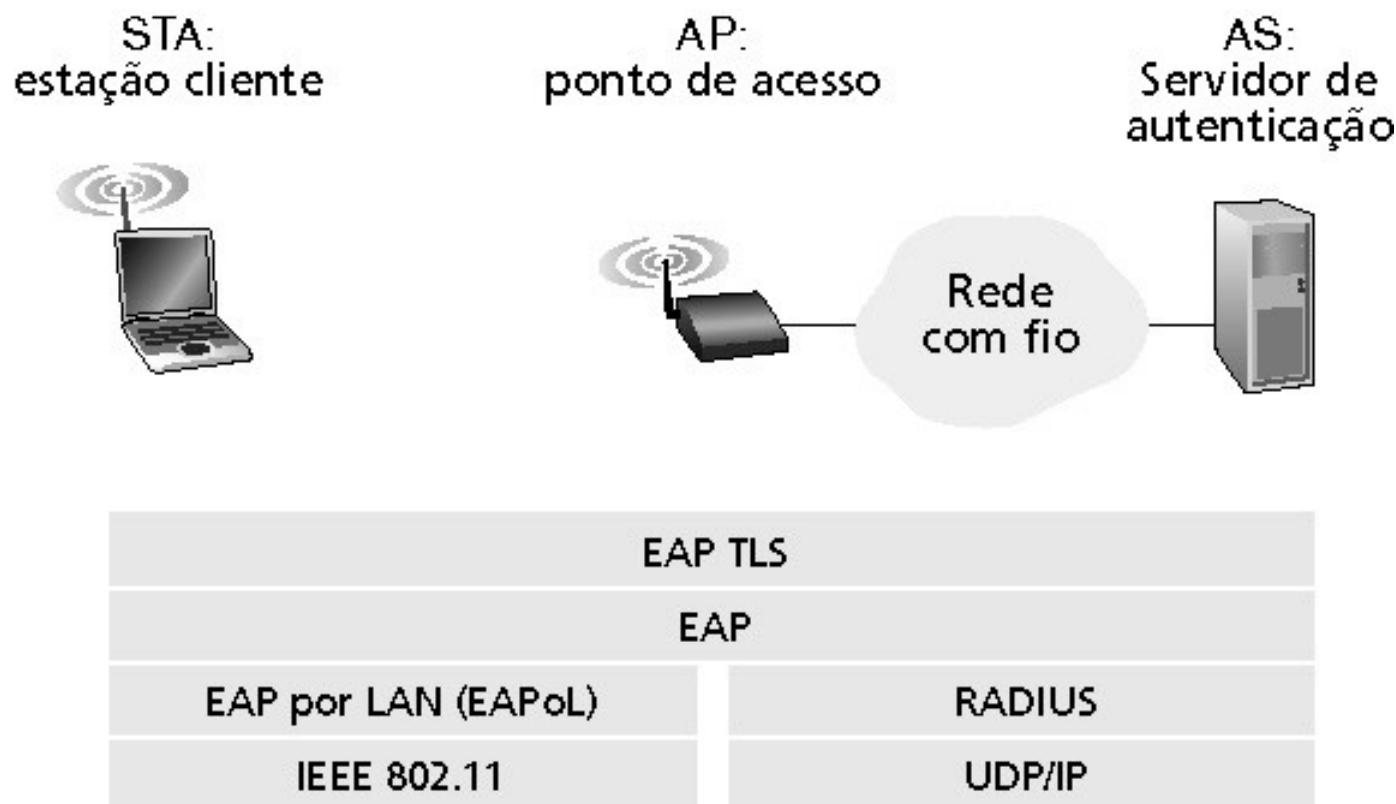
- Numerosas (e mais fortes) forma de criptografia são possíveis
- Oferece distribuição de chave
- Usa autenticação de servidor separada do ponto de acesso



802.11i: quatro fases de operação



- EAP: protocolo fim-a-fim entre o cliente (móvel) e o servidor de autenticação
 - EAP envia sobre “enlaces” separados
 - Móvel para AP (EAP sobre LAN)
 - AP para servidor de autenticação (RADIUS sobre UDP)



Técnicas básicas.....

- Criptografia (simétrica e pública)
- Autenticação
- Integridade de mensagens
- Distribuição de chaves

.... usadas em muitos cenários diferentes de segurança

- E-mail seguro
- Transporte seguro (SSL)
- IP sec
- 802.11

