# PTC 2550 - Aula 22

5.4 Autenticação do ponto final
5.5 Exemplo de aplicação: tornando o e-mail seguro

(Kurose, p. 587 - 626)
(Peterson, p. 444-454)

23/06/2017

# Capítulo 5 - Sumário

*5.1 O que é segurança de rede?*

5.2 Princípios de criptografia

5.3 Integridade de mensagem, autenticação

5.4 Tornando o e-mail seguro

# Autenticação

- *Objetivo:* Bob quer que Alice "prove" sua identidade para ele
- Exemplos: usuário identificando-se a servidor de email, roteadores identificando-se antes de trocar informações de rota, etc.

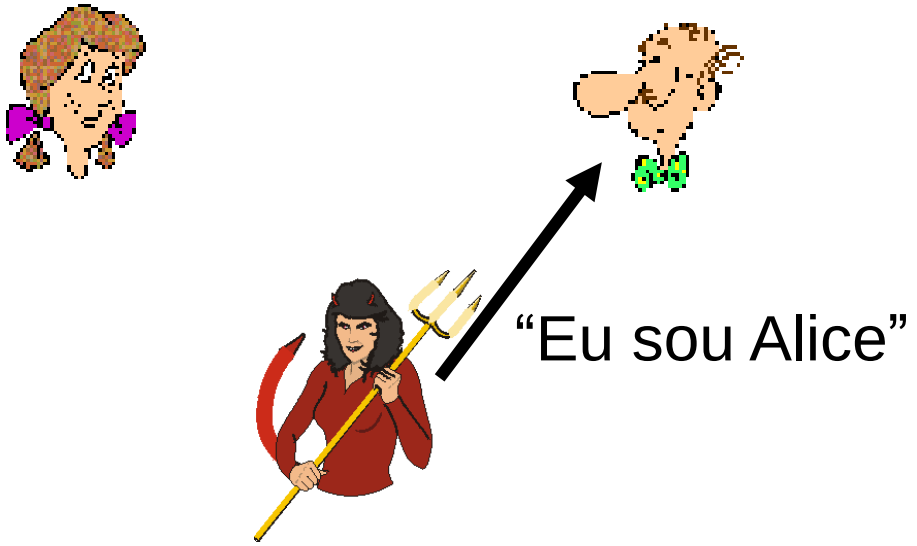
*Protocolo ap1.0:* Alice envia "Eu sou Alice"


Cenário de falha??

# Autenticação

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

*Protocolo ap1.0:* Alice envia "Eu sou Alice"

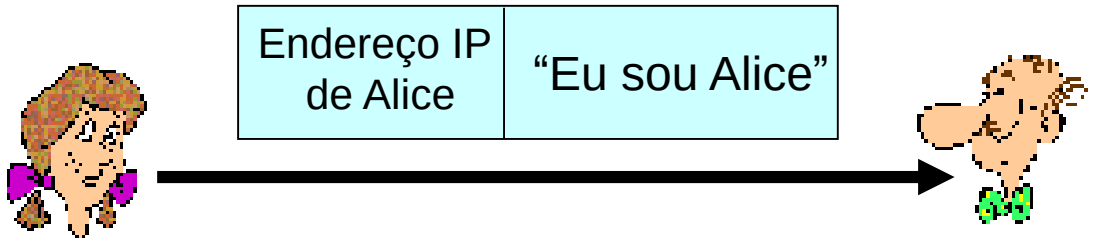



em uma rede,
Bob não pode "ver"
Alice, entãoTrudy
declara simplesmente
ser ela Alice

# Autenticação: outra tentativa

*Protocolo ap2.0:* Alice diz "Eu sou Alice" em um pacote IP contendo seu endereço IP fonte

*Hipótese: Alice tem um IP fixo "bem conhecido"*

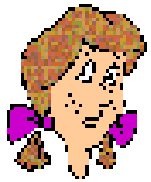


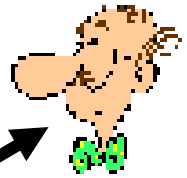

Cenário de falha??

# Autenticação: outra tentativa

*Protocolo ap2.0:* Alice diz "Eu sou Alice" em um pacote IP contendo seu endereço IP fonte

*Hipótese: Alice tem um IP fixo "bem conhecido"*





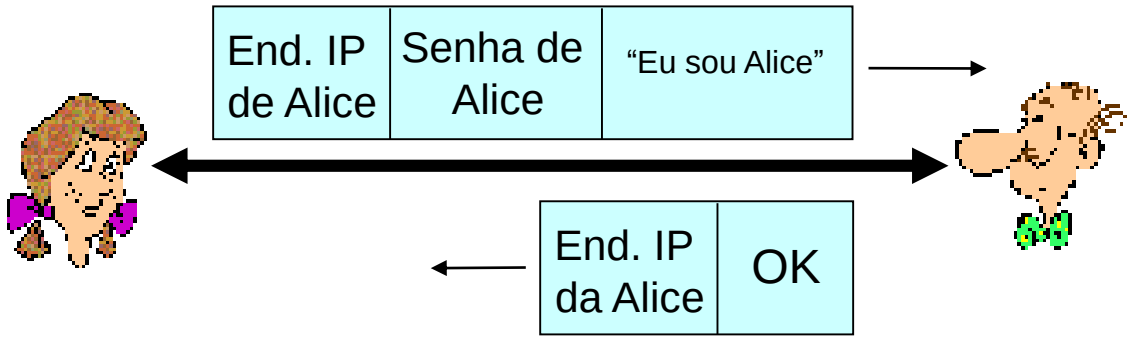
| Endereço IP de Alice | "Eu sou Alice" |
|---|---|

Trudy pode criar pacote "*falsificando*" seu endereço de IP *(IP spoofing)*, se roteador de primeiro salto não filtrar pacote [RFC 2827]

# Autenticação: outra tentativa

*Protocolo ap3.0:* Alice envia "Eu sou Alice" e envia sua senha secreta para "prová-lo".

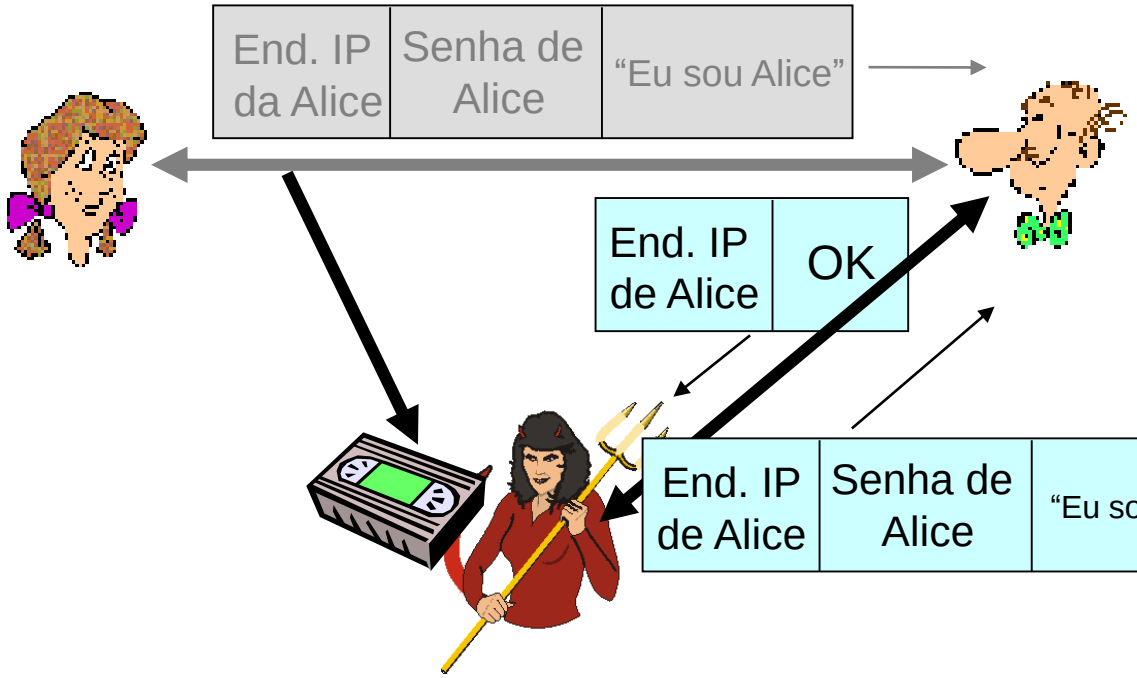
Abordagem clássica usada em servidor de e-mail, Facebook, Netflix, FTP, telnet, e muitos outros serviços que usam autenticação por senha.




Seguro??
Cenário de falha??

# Autenticação: outra tentativa

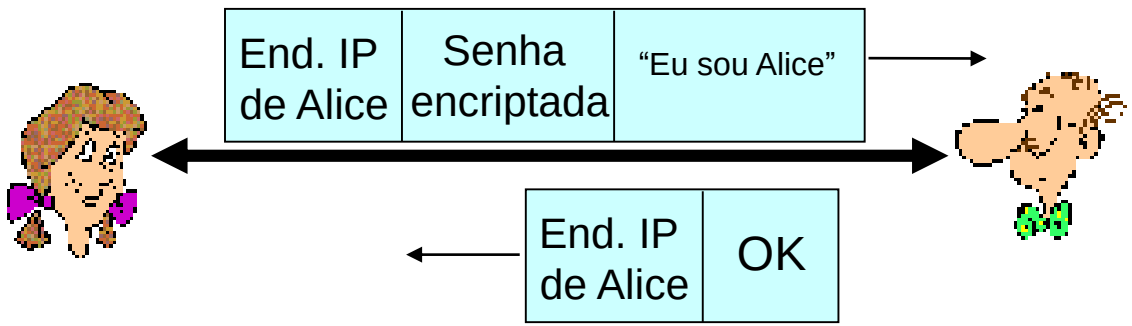
- *Protocolo ap3.0:* Alice envia "Eu sou Alice" e envia sua senha secreta para provar sua identidade.




*ataque playback:* Trudy grava pacote de Alice e depois "toca" novamente para Bob

# Autenticação: mais uma tentativa

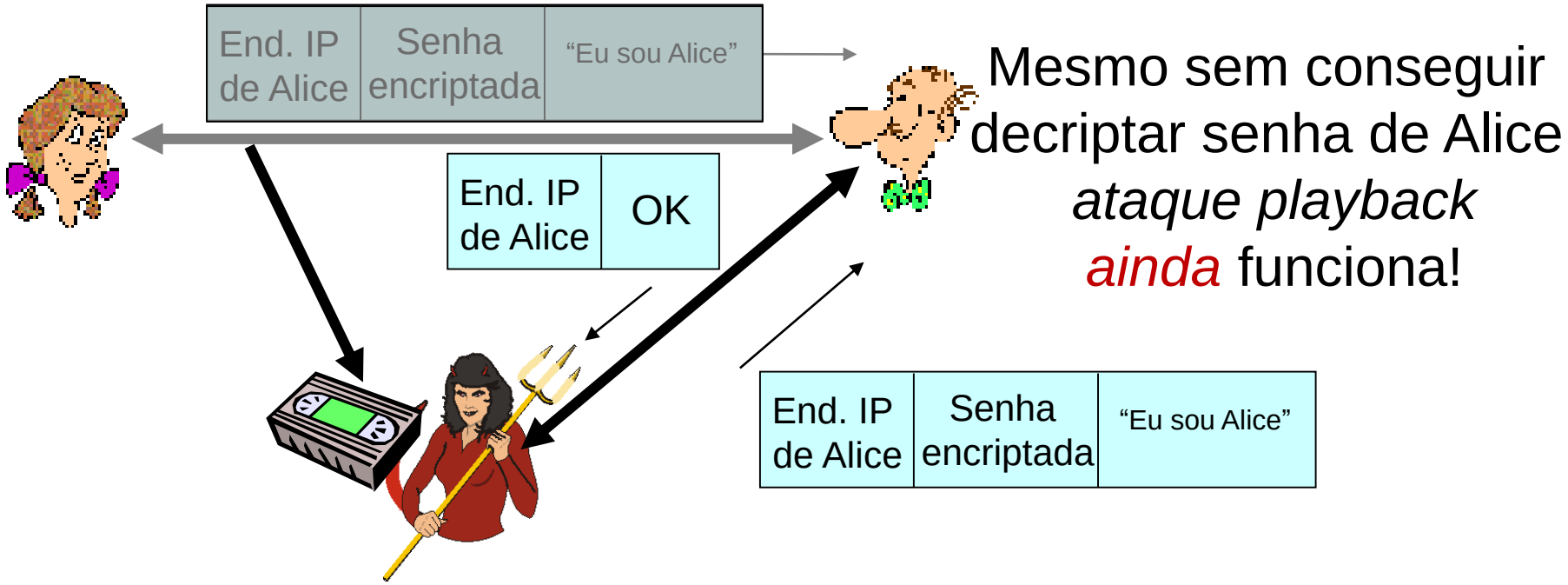
*Protocolo ap3.1:* Alice envia "Eu sou Alice" e envia sua senha secreta *encriptada* para provar sua identidade.




Cenário de falha??

# Autenticação: mais uma tentativa

*Protocolo ap3.1:* Alice envia "Eu sou Alice" e envia sua senha secreta *encriptada* para provar sua identidade.

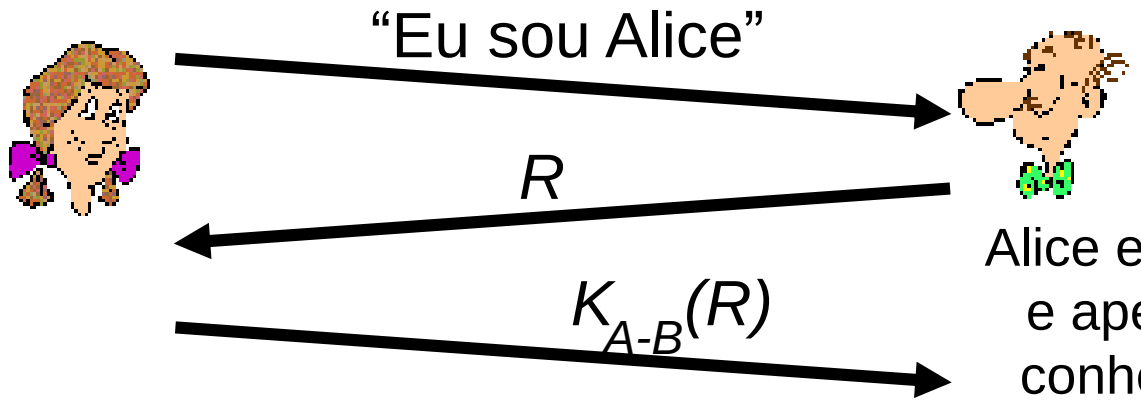


# Autenticação: ainda tentando

*Objetivo:* evitar ataque *playback*

*nonce:* número (*R*) usado apenas "*once-in-a-lifetime*"

- *ap4.0:* para provar que Alice está "ao vivo", Bob envia a Alice *nonce*, *R*. Alice precisa então retornar *R*, encriptado com a chave secreta compartilhada




Falhas, problemas?

# Autenticação: ap5.0

ap4.0 requer chave simétrica compartilhada

- Podemos autenticar usando técnicas de chave pública?

*ap5.0:* usa *nonce* e criptografia de chave pública


Bob computa

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

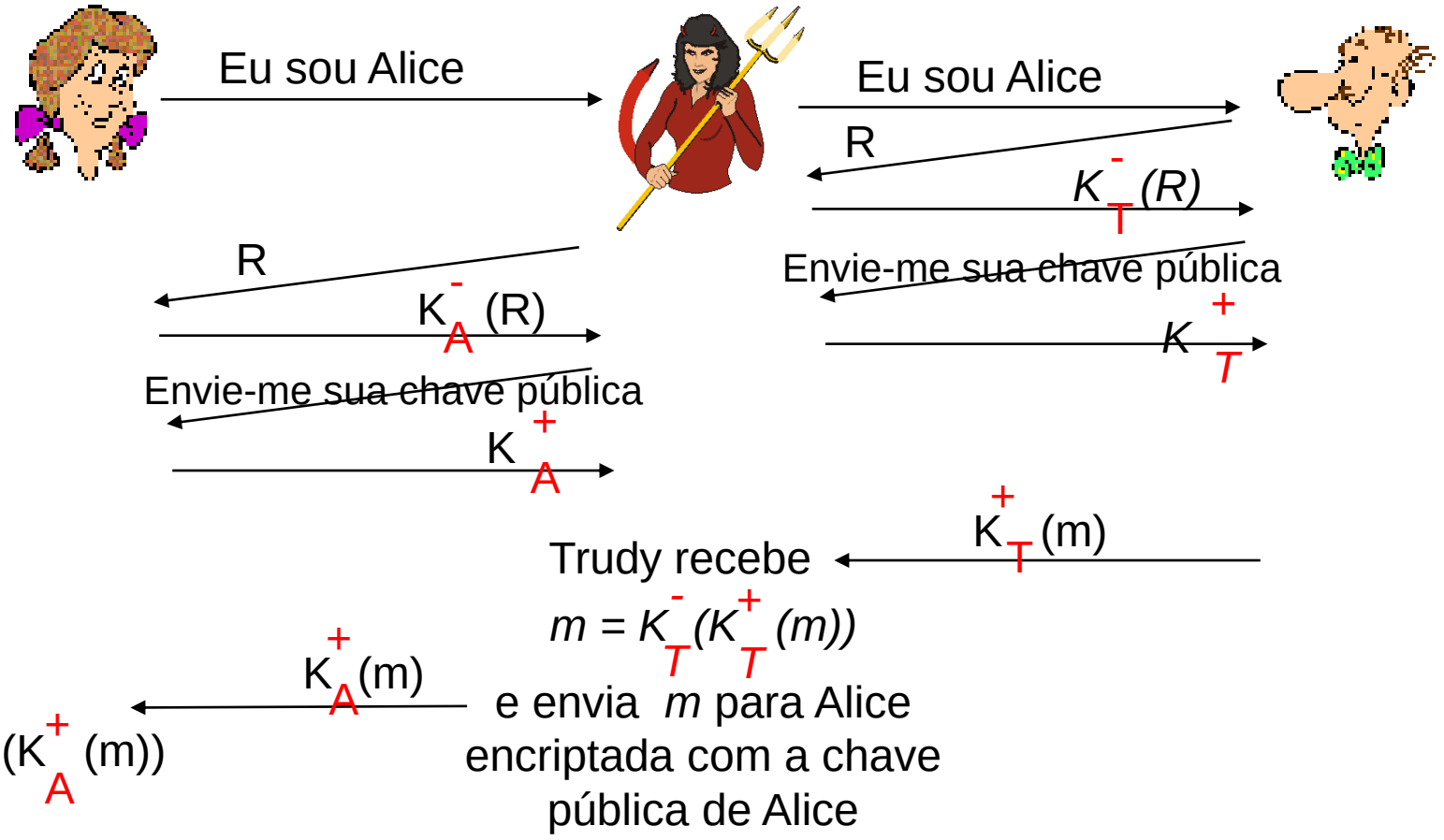
e sabe que apenas Alice poderia ter a chave privada que encriptou *R* de modo que

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

Problemas?

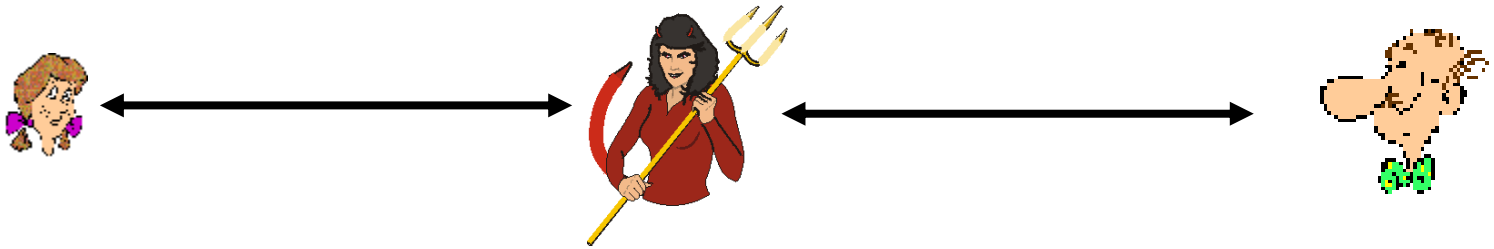
# ap5.0: falha de segurança

*ataque man-in-the middle:* Trudy se passa por Alice (para Bob) e por Bob (para Alice)



# ap5.0: falha de segurança

*ataque man-in-the middle:* Trudy se passa por Alice (para Bob) e como Bob (para Alice)



difícil de detectar:

- Bob recebe tudo que Alice envia e vice-versa (assim, Bob pode confirmar conversa com Alice pessoalmente depois!)
- Problema é que Trudy recebe todas as mensagens também!

❖ Como resolver? Já vimos na aula anterior...

❖ Autoridade certificadora!

# Aplicações

A seguir, ver como conceitos vistos:

- criptografia de chave simétrica
- criptografia de chave pública
- autenticação de ponto final
- distribuição de chaves
- integridade da mensagem
- assinatura digital

são usados para prover segurança na Internet

- Questão de segurança presente em todas as camadas, que serão exemplificadas
- Aplicação (e-mail), Transporte (SSL), Rede (IPsec), Enlace (segurança em IEEE 802,11)

# Capítulo 5 - Sumário

*5.1 O que é segurança de rede?*

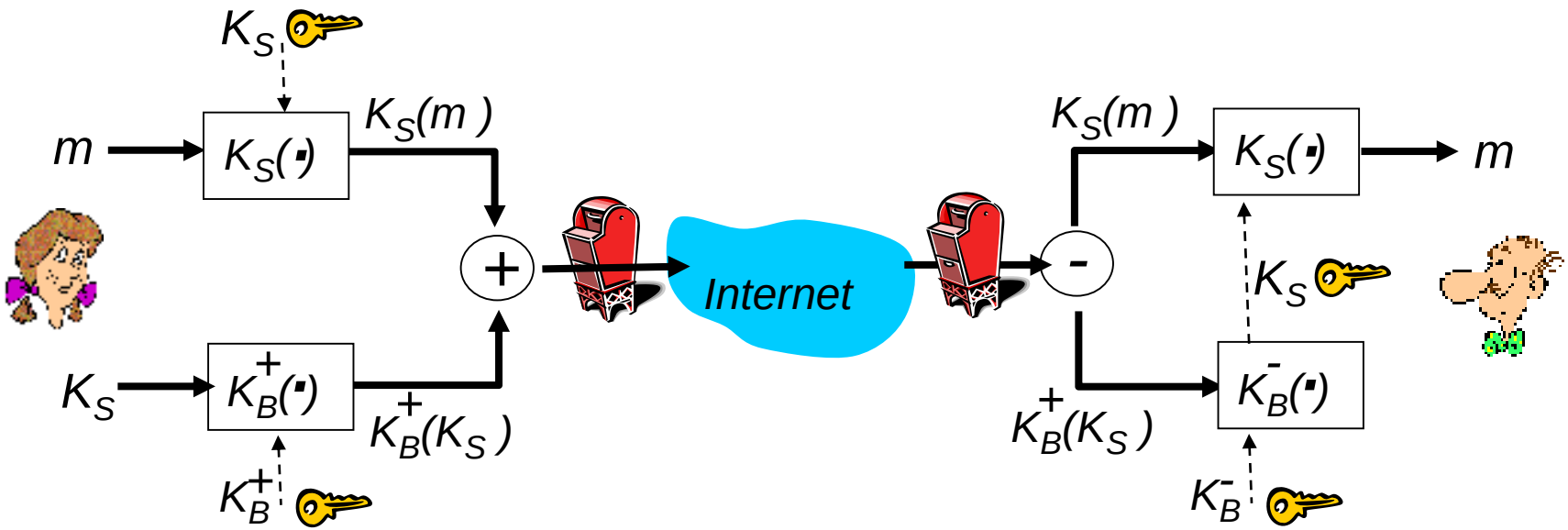
5.2 Princípios de criptografia

5.3 Integridade de mensagem, autenticação

5.4 Tornando o e-mail seguro

# E-mail seguro: *confidencialidade*

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

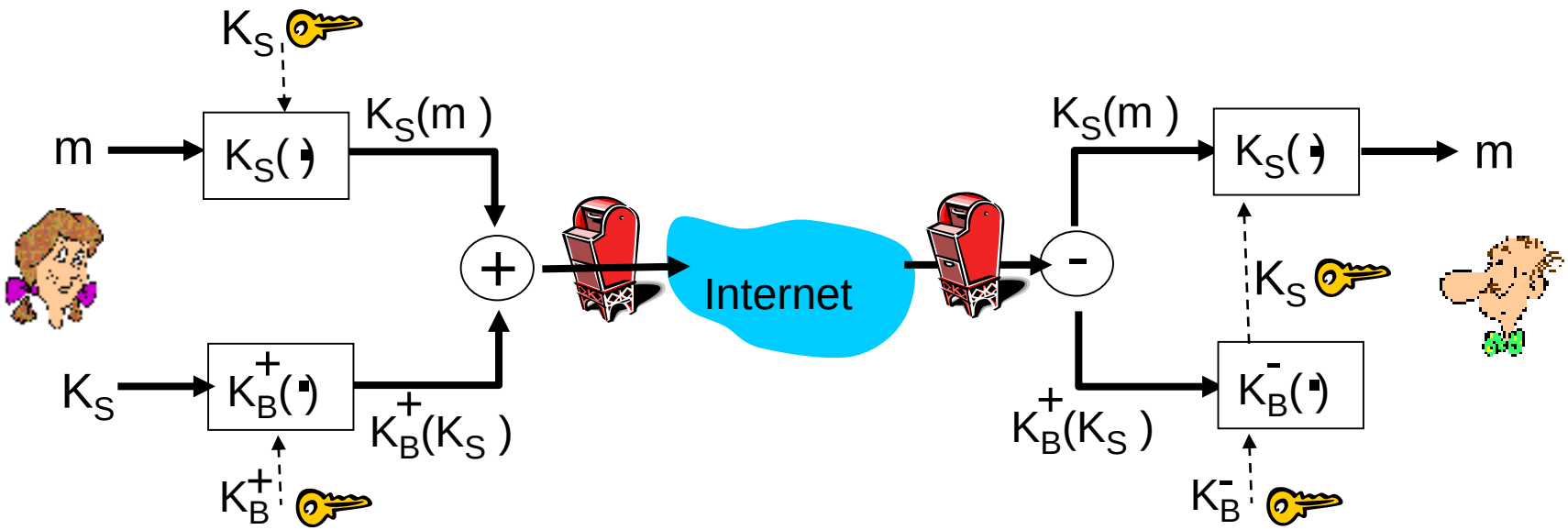



*Alice:*

- gera chave simétrica aleatória, $K_S$
- encripta mensagem com $K_S$ (por eficiência)
- também encripta $K_S$ com a chave pública de Bob
- envia $K_S(m)$ e $K_B(K_S)$ para Bob

# E-mail seguro: *confidencialidade*

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

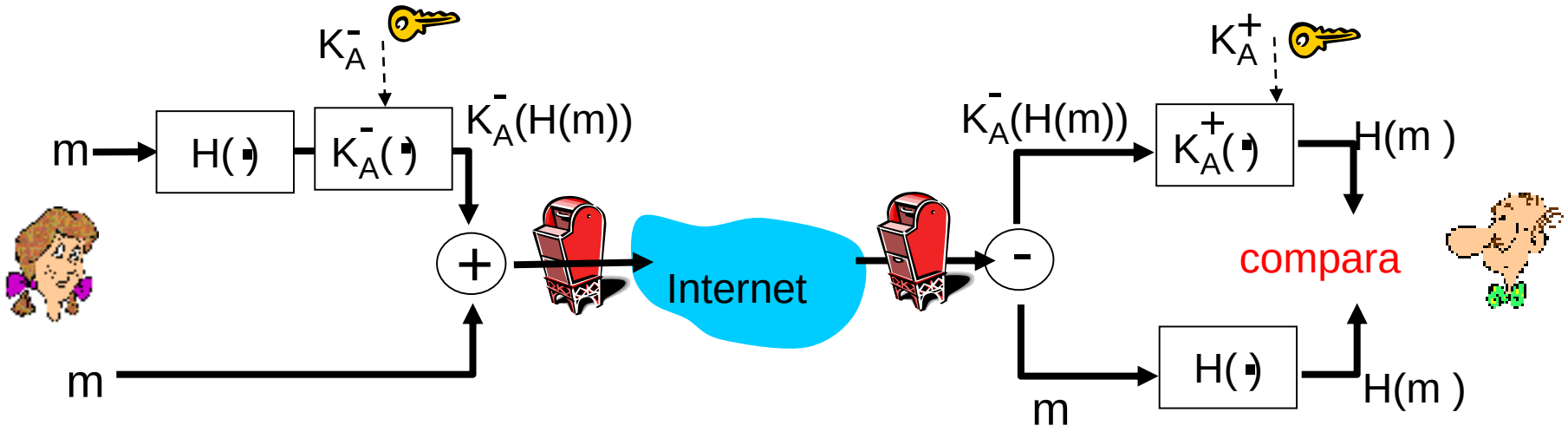



*Bob:*

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

# E-mail seguro: *autenticação e integridade*

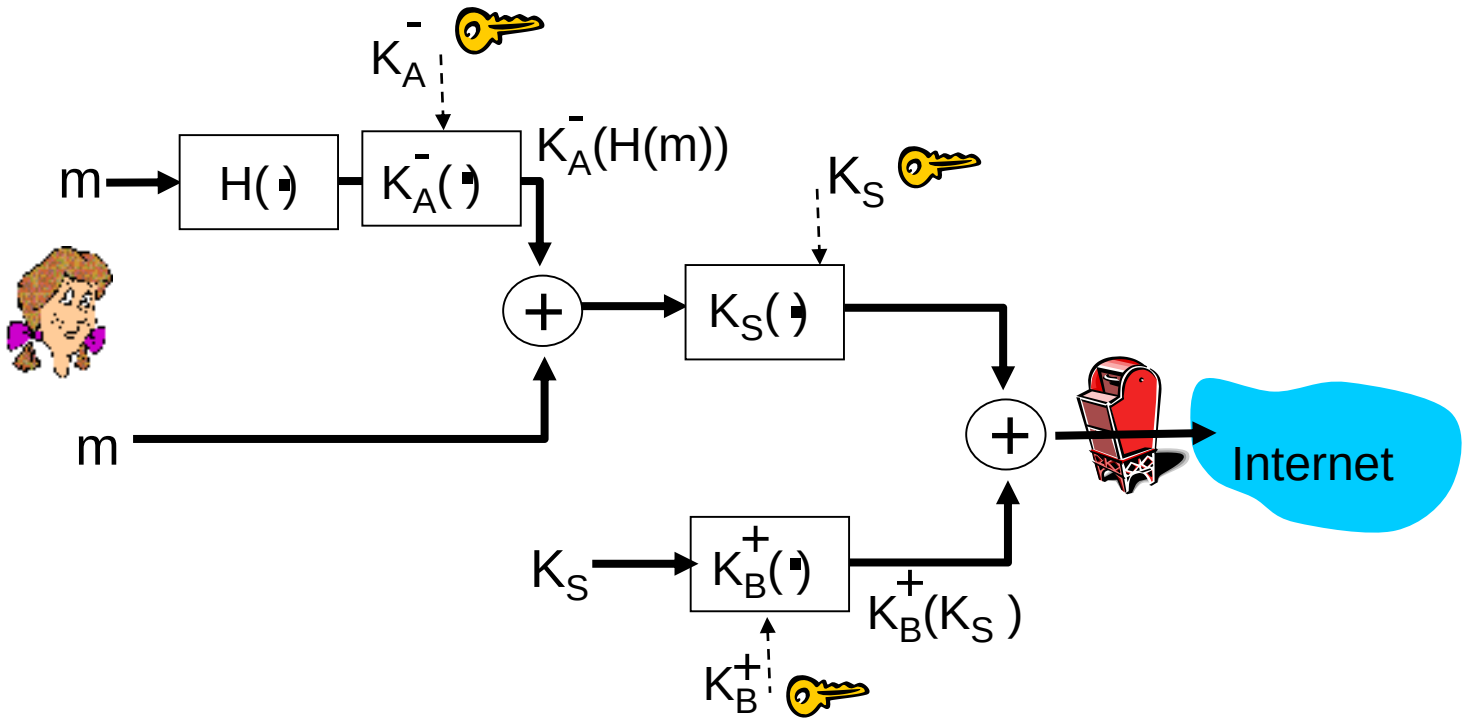
Alice quer *autenticação do remetente e integridade da mensagem*




- Alice assina digitalmente a mensagem
- envia a mensagem (em texto aberto) e a assinatura digital

# E-mail seguro: *completo*

Alice quer confidencialidade, autenticação do remetente e integralidade da mensagem.




*Alice usa 3 chaves:* sua chave privada, a chave pública de Bob, chave simétrica recém criada

# Exemplo: PGP (Pretty Good Privacy)

- Escrito por Phil Zimmermann em 1991
- Usa MD5 ou SHA para calcular o resumo da mensagem
- CAST, 3-DES ou IDEA para encriptação de chave simétrica
- RSA para encriptação de chave pública
- Opção de assinar digitalmente, encriptar ou ambos.

- Extensão gratuita para o Gmail: CryptUp

# Exemplos PGP

- -----BEGIN PGP SIGNED MESSAGE-----
  Hash: SHA256

  Essa mensagem está apenas assinada.
  -----BEGIN PGP SIGNATURE-----
  Version: CryptUp 4.1.8 Gmail Encryption https://cryptup.org
  Comment: Seamlessly send, receive and search encrypted email

  wsFcBAEBCAAQBQJZStEaCRA2GNn0mCypHgAAMQAP/1xh848HIGWeEzejPS+e
  KDDZo+8yF8692yu3RiC0RINUetB2WoJVXknQLYM4XLpZQiRO2wyFtBywKBvT
  ZNlqgf9q6pngcGEzeBZPgmv63VHVbS9T7Iif6cuQ7wCboEu0bdC1sWSphpQk
  /al0B44hQxw/XU+jQKCzr6YduFygdm+RzEQVTx5594E2I9tghsstdPtVgjWt
  Jz/9h9vg0Z9XH4609yOZQaj3aPXaK/5kePCJdX4otQWcYkc0mjGxwfxP6tQa
  NoqzKSpumiioIgwoQ1HPePGdyAMvJltGsAjPowsExijwUpdF7U5p+61gdW9d
  tEGIyxaaMp/h1OmLidMqhpCDYbPoaCNAHWWz4TS7cRfQlqfZVwSCJx15bm4f
  aDqwaLKii3vlYT704wWFVRP2RfmNT8Qys6BSax1TuRQPRn9RVpQf0lCcuLHD
  eOUhV6nXYJsSZgTYEtUKnUzY0Uh2j7Qnn/cYWGqLG/ERYeXNBPsYKEJYVbfm
  AmoOjP4vAtrU8a+qmGBO4MDnBDjOjNwSt043bv28xKvqy2BuCwP3iozOud9l
  PSA/zgqTn4YEFLitFkrGQ478tQP5kSJU3fFP82v1NP+IUWehi0NIjdAS5qyt
  wk1EW9Y+mxIoQ0lf4gJ46c+SvyXVE6BGk858p7VjaEOfM8Y71g4NHLVFHcOt
  aBOl
  =PVM2
  -----END PGP SIGNATURE-----

# Exemplos PGP

```
-----BEGIN PGP MESSAGE-----
Version: CryptUp 4.1.8 Gmail Encryption https://cryptup.org
Comment: Seamlessly send, receive and search encrypted email

wcFMA/bPCi5wYqxPARAAmTT3R21zfxdOxy+c07VoIgt+eZLTryBaBt5PlGQP
ku7hiiJnProgNJZLnymKdZUB55nvjX3AFdlqMoY5Sz7+SOCbaDiX1T3yURaa
RqiEIjpWycQNOdcjtOQHNOc73Ud11uRMAy31CXdMZZ1TZm0kvApj0kjD85KW
1ki/z41a7qeUZO3UCSzcVNO1bmNVfaVsuVCAABb/IllgEUR36xa881ynWc+k
440Nx0jJ8OxwdFF4LJPlto3tAgi0Ov6HBB9lYigNddJ85Ot9uF6cdBu6pCs2
SD0LyKQhL95HApID8n0MEtUu3n/CcOkgj2OAQ7l0rLj81uW7fgf5nMYcrxEZ
QMqUB42LRmeosNeqznECkpyp8n/AdX8Zp+siLjCY35mEGyXuFE5X4Evyddy2
q8fEOoRekQmq7fEaN7ocOD+7c6tfTT0TY+KCNHwxZhLaKFy7LvqRpvRe+4Ke
teUFEth8H1HWeUF5zOfIur277NKpiFrjTvWm7K//PTUYMVVXidd9hKTpS1BM
X6COmK9GdWL9FcDYPm4BwJI883eb4Mlfa9gXfCP+gtSoCKKSTFXvMxgV4CLn
ZfwYetaZwqleNmITkKjauNjyuO4FFCrs2yTVU6CiieOzLHaroRu7VodvR8s6
2CcT5dKmH8KfChMjMe5CIHIowEJW8vsQkdUBVDo1+sDBwUwD9s8KLnBirE8B
D/9JqgA8d0scNRacVo/ZGH5g5CipyyzmM25DjApiNPbCMqEkiDgLMUz4Dz71
kTBKYVRDhmPRHElqzQRNMxpRvMGGbieItlxTdcKdEEoWBfeWIAHRzCk64yCB
zd9UtIZo/iF8COPTACjJKiMtw078ouN64Rn03qe8DDcz9HqX9E5o1fjqBvWW
Zull3KT+XRHLGzrYI9QOUglcQ9KKm+jxxe2/5cATHhsRnBBlZEEZ78ppuxdF
tj0sDC74U0WD2obDXsZorz6yWfPt3b5yfnkWLWWxD69tdjWGNLGCbmwzeDtu
zLvnG0BhL+CKjzzDxamA90j3BCVdVMKHafbY2joX5Qqq2kLwlekHLuXybIB2
UnkKQcxDhhUUPnmvvlAS7dYupch/PQ5WapGu4PW4ewQvZ0DPVsIFt2NUqycp
QQgb3hhj3MXM212s2ZyEh0ijzsJug4pbqK/nuFxIJCbVOGuhj+4L8MAbwr8a
puv0ZUoqRuKj59iVUPyytTgHYabQWTNH09o9NBtZHHAuXIdHTfMnh3Sai1XC
MMAg4YQt7eAonas4evIWvPZDfqlMvCdEqn1g1A6fxApRY3d1RiDhFHf9mmZA
zoynMF68nd8b7IFdHL2ckXDLnfzn7ZnTQBKJdv8Iaiq6ITyLnspnmCVQPAHX
CjC6uw/nHt4GdWyo5QkK8Aar99JlAWQpCmKAjoAzzQLaqZkmRjUCebIcglcE
qkDyU/tlcqqcXHj2d9iH37ZMMCUj0hfr7z40RvSF+B3a7atw+JPU/HbX3gFF
Q/kInH4nK+y9a69f7MfPgCXsMvXFsm1BAApje5x49hc=
=tNRm
-----END PGP MESSAGE-----
```