# PTC 2550 - Aula 21

## 5.3 Integridade de mensagem e assinaturas digitais

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

21/06/2017

# 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
5.5 Tornando conexões TCP seguras: SSL
5.6 Segurança na camada de rede: IPsec
5.7 Tornando LANs sem fio seguras
5.8 Segurança operacional: *firewalls* e IDS

# Assinaturas digitais

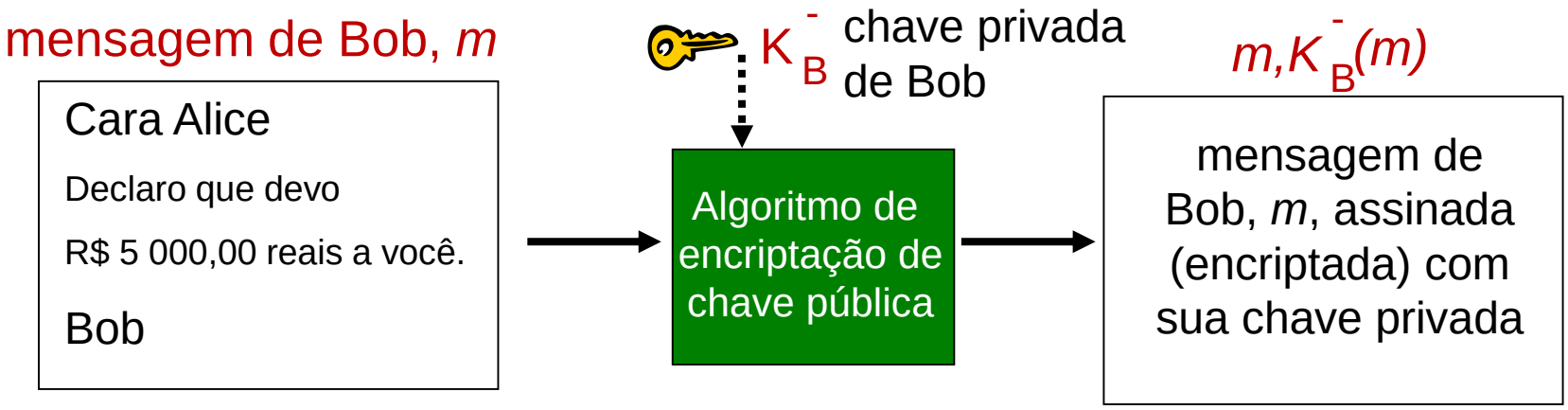
Técnica criptográfica análoga à assinatura manuscrita:

- remetente (Bob) assina digitalmente o documento, estabelecendo que ele é o dono/criador do documento.
- *verificável, não falsificável:* destinatário (Alice) pode provar para alguém que Bob, e ninguém mais (incluindo a Alice), deve ter assinado o documento

# Assinaturas digitais

## assinatura digital simples para mensagem *m*:

- Bob assina *m* encriptando com sua chave privada $K_B^-$, criando mensagem "assinada", $K_B^-(m)$



# Assinaturas digitais

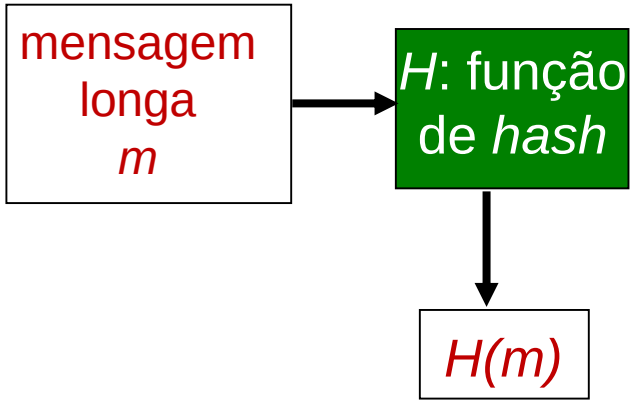
- Suponha que Alice recebe a mensagem *m*, com assinatura: *m*, $K_B^-(m)$
- Alice verifica *m* assinada por Bob aplicando a chave pública de Bob $K_B^+$ a $K_B^-$(m) e então verifica se $K_B^+(K_B^-(m)) = m$.
- Se $K_B^+(K_B^-(m)) = m$, quem assinou *m* precisa ter usado a chave privada de Bob.

  Alice então verifica que:
  - Bob assinou *m*
  - mais ninguém pode ter assinado *m*
  - Bob assinou *m* e não *m'*

  Não repúdio:
  - ✓ Alice pode levar *m*, e a assinatura $K_B^-(m)$ à justiça e provar que Bob assinou *m*

- ✓ Problema: algoritmo muito pesado para *m* grande

# Resumo de mensagens




É computacionalmente caro encriptar com chave pública longas mensagens

*objetivo:* "impressão digital" fácil de computar e de comprimento fixo

- aplicar função de *hash H* a *m*, obtendo resumo de mensagem de tamanho fixo, *H(m)*.

Propriedades de funções *hash*:

- muitos-para-1
- produz resumo de mensagem de tamanho fixo (impressão digital)
- é computacionalmente impossível encontrar duas mensagens diferentes *x* e *y* tais que *H(x)=H(y)*
  - Ou seja, é computacionalmente impossível um intruso substituir uma mensagem protegida por *hash* por outra

# *Checksum* da Internet : exemplo (pobre) de função de *hash* criptográfica

*Checksum* da Internet tem algumas das propriedades das funções de *hash*:

- produz resumo de comprimento fixo (soma de 16-bit) da mensagem
- é muitos-para-1

Mas dada mensagem com certo valor de *hash*, é fácil encontrar outra mensagem com 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 |

diferentes mensagens
mas *checksums idênticos*!

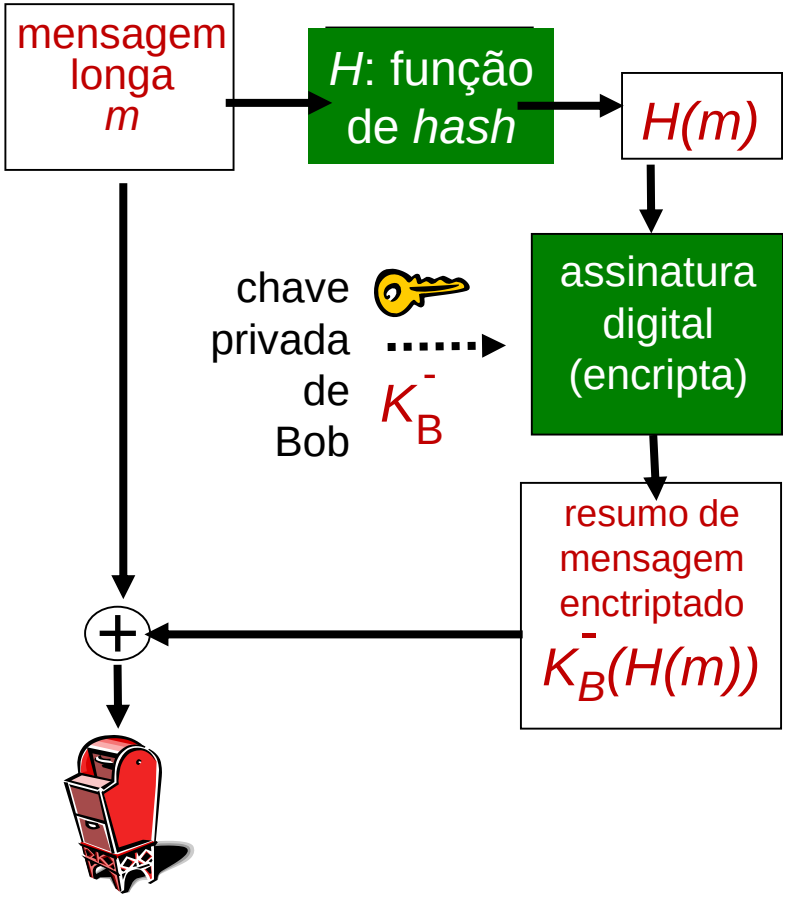
# Algotitmos para funções de *hash*

- MD5 – função de *hash* muito utilizada (RFC 1321)
  - computa resumo de mensagem (*hash)* de 128 bits em processo de 4 etapas.
  - dada uma sequência arbitrária de 128 bits, é "difícil" construir mensagem *m* cujo *hash* MD5 seja igual a *x*

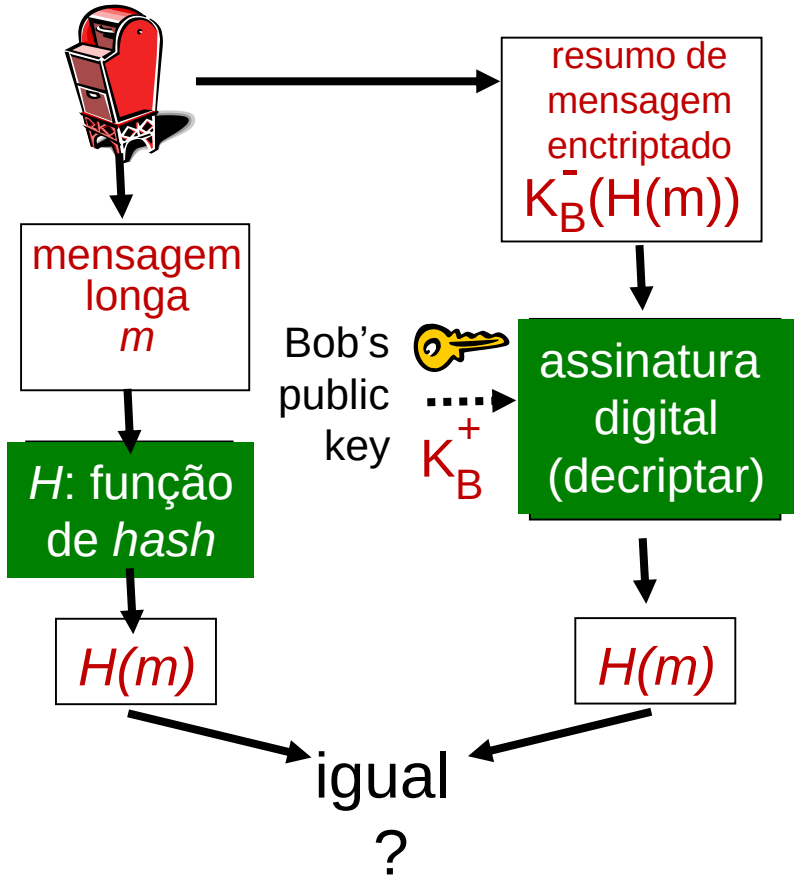
- SHA-3 também utilizada
  - padrão americano [NIST, FIPS PUB 180-14] - Obrigatório para aplicações do governo americano
  - resumo de mensagem de 160 bits

# Assinatura digital – resumo de mensagem assinada

Bob envia mensagem digitalmente assinada:




Alice verifica a assinatura e integridade da mensagem assinada digitalmente:



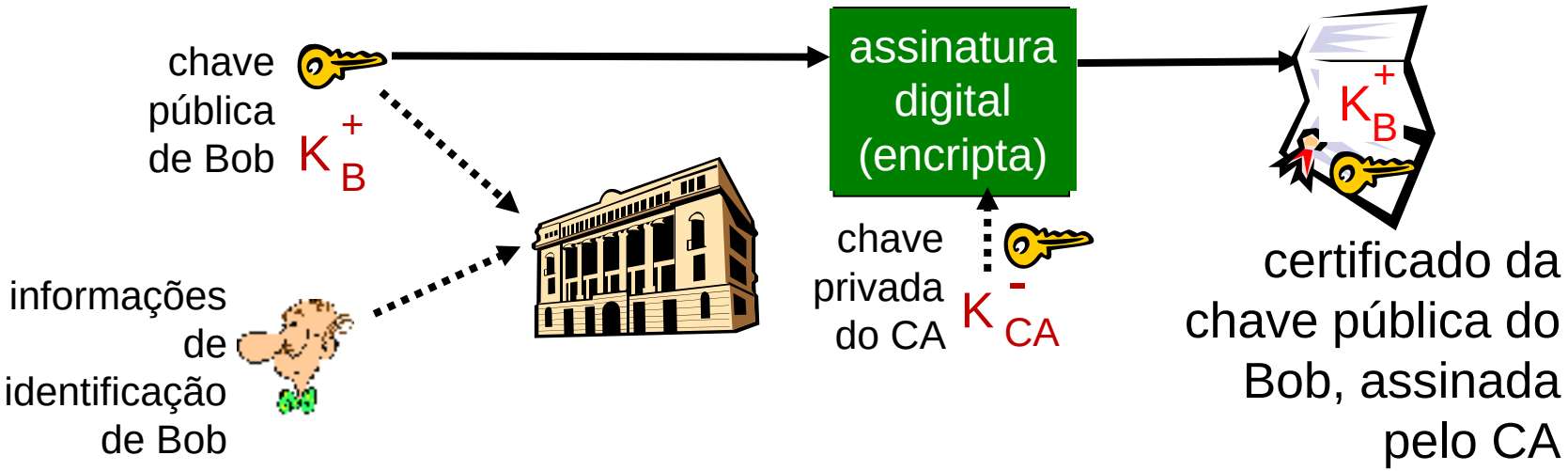

Problema: como saber se chave pública é do Bob mesmo?

# Certificação de chave pública

- motivação: Trudy prega trote da pizza no Bob
  - Trudy cria um pedido por e-mail:
    *Cara Pizza Store, Por favor envie-me 4 pizzas de atum.*
    *Obrigado, Bob*
  - Trudy assina seu pedido com sua chave privada
  - Trudy envia o pedido para a *Pizza Store*
  - Trudy envia para a *Pizza Store* sua chave pública, mas diz que é a do Bob
  - *Pizza Store* verifica a assinatura; entrega 4 pizzas de atum para o Bob
  - Bob odeia atum!! ☹

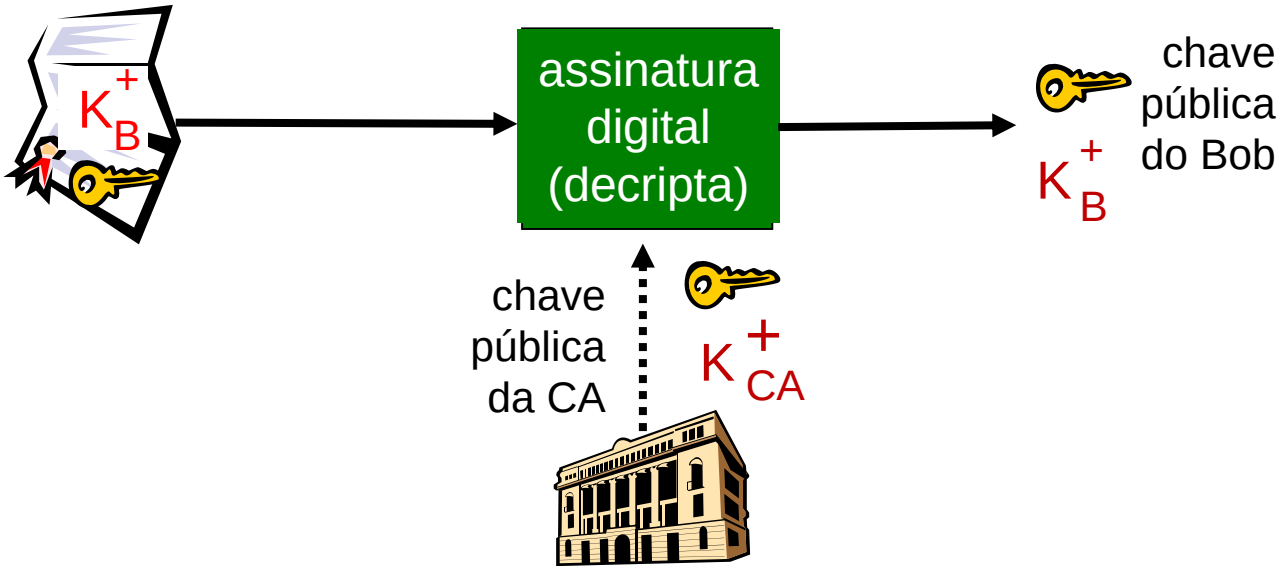
# Autoridades de certificação

- *autoridade de certificação (CA):* Vincula chave pública a determinada entidade. Exemplos: Comodo, Symantec, GoDaddy
- Entidade *E* registra sua chave pública com CA.
  - *E* provê "prova de identidade" para CA (não há procedimento padrão).
  - CA cria certificado vinculando a entidade a usa chave pública.
  - certificado contém chave pública da entidade digitalmente assinada por CA – CA diz "essa é a chave pública de *E*"



# Autoridades de certificação

- quando Alice quer a chave pública de Bob:
  - obtém certificado de Bob (de Bob ou outro lugar).
  - applica a chave pública da CA ao certificado de Bob, obtém a chave pública de Bob




Formato do certificado dado pela RFC1422