Conteúdo

Principais serviços de segurança

- Confidencialidade
- Integridade
- Autenticação de mensagens
- Não-repudiação
  - Não é possível negar autoria

Outros serviços de segurança

- Identificação ou autenticação de entidades
  - Estabelecer e verificar identidades de: pessoa, banco, cartão de crédito
- Controle de acesso
  - Restringir acesso a recursos para entidades com permissão
- Auditabilidade
  - Prover evidências sobre corretude do funcionamento do sistema
- Anonimidade

Introdução

- Como implementar segurança provida por assinaturas autenticadas por cartórios físicos de maneira digital?
- Requisitos
  - Não é possível falsificar (possível com MAC)
  - Não é possível negar a autoria
    - No MAC, todos com chave podem "assinar"
  - Possível transferir uma assinatura para outro
  - Verifiabilidade pública

Importância legal

- Decreto Nº 6.605/2008 (Brasil)
  - `http://www.iti.gov.br/noticias/83-icp-brasil/139-decretos`
- Electronic Signatures in Global and National Commerce Act (ESIGN) (EUA)
- Signaturegesetz (Alemanha)

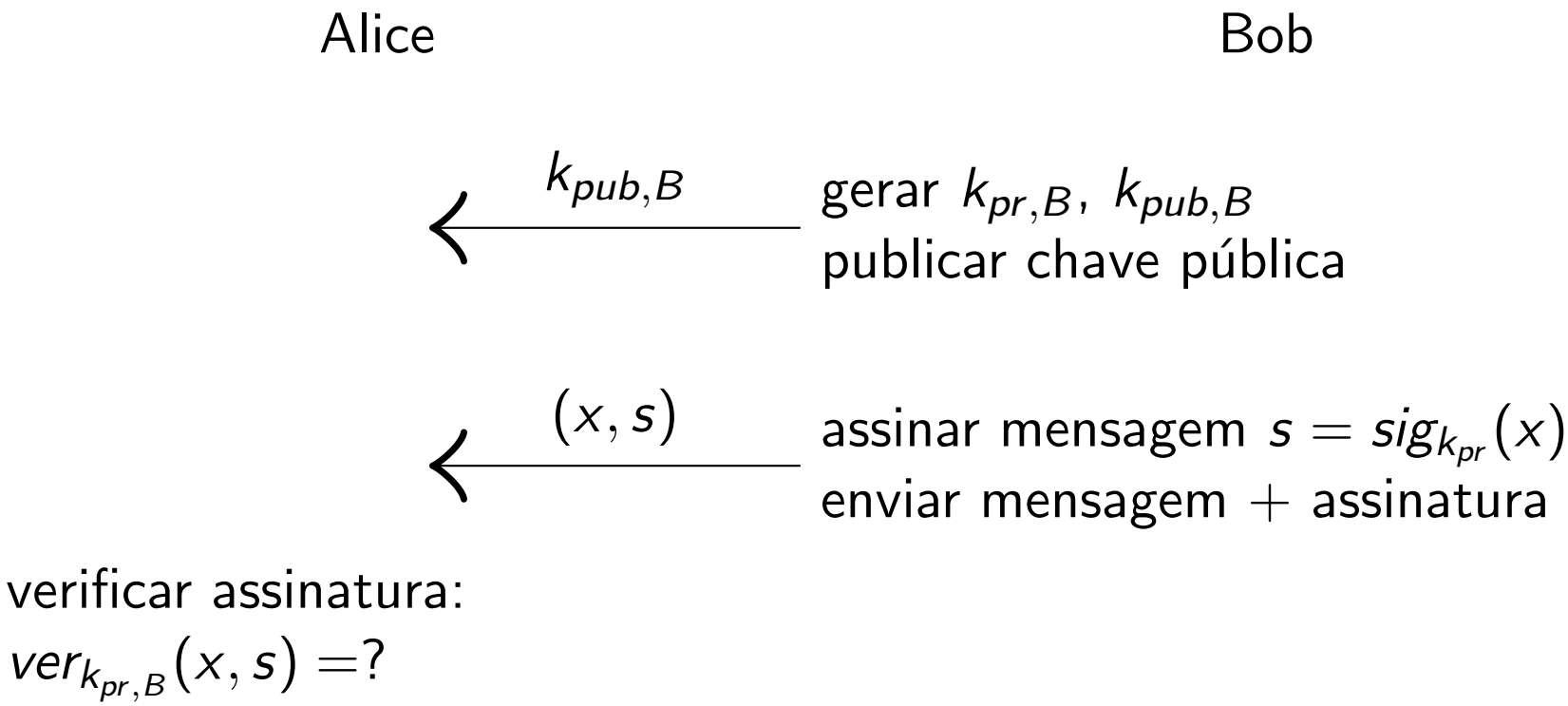
Protocolo genérico

- Dado documento digital d, temos
  - Função de gerar par de chaves para assinar k_a e verificar k_v
    - k_a é privada
    - k_v é pública
  - Função de assinar $sig_{k_a}(d) = x$
  - Função de verificar $ver_{k_v}(d, x) \in \{\text{válida, inválida}\}$

Comparação aos MACs

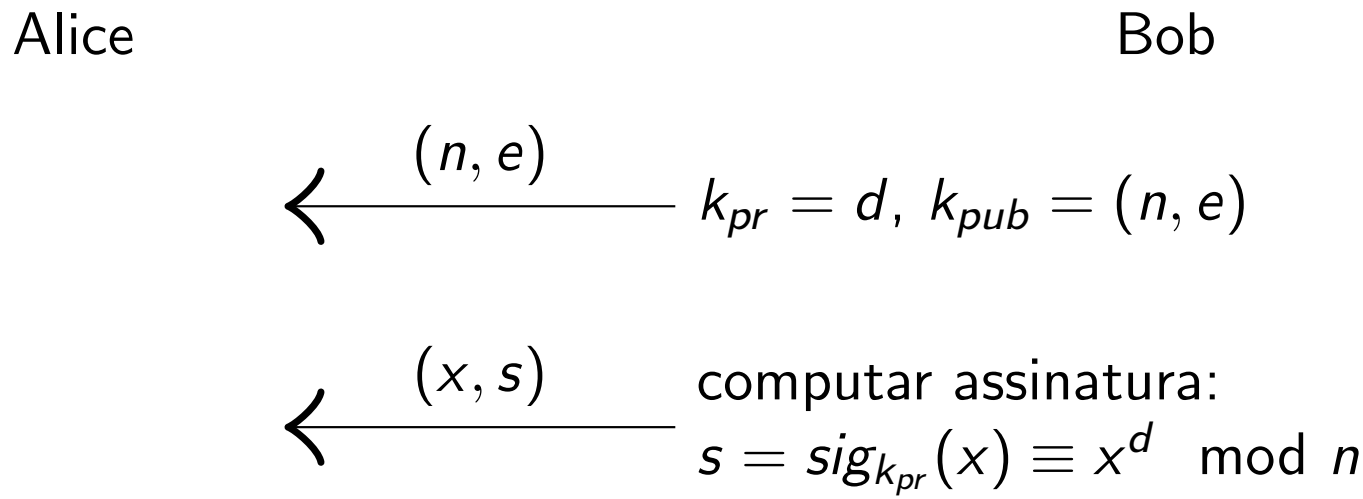
- Verifibilidade pública
  - Qualquer um pode verificar uma assinatura
  - Somente o detentor da chave pode verificar uma etiqueta MAC
- Transferabilidade
  - Possível transferir uma assinatura para outro
- Não repudiação

Protocolo básico de assinatura digital



Assinatura digital RSA simplificada

- Chaves RSA: $k_{pr} = (d)$, $k_{pub} = (n, e)$



verificar com $ver_{k_{pub}}(x, s)$:
$x' \equiv s^e \mod n$
se $x' \equiv x \mod n$, assinatura válida
senão, assinatura inválida

Assinatura digital RSA: exemplo

Alice		Bob
		escolher $p = 3$ e $q = 11$
	$\xleftarrow{(n, e) = (33, 3)}$	$n = p \cdot q = 33$
		$\Phi(n) = 20$
		escolher $e = 3$
		$d \equiv e^{-1} \equiv 7 \mod 20$
	$\xleftarrow{(x, s) = (4, 16)}$	computar assinatura:
		$x = 4$:
		$s = x^d \equiv 4^7 \equiv 16 \mod 33$

verificar:

$x' = s^e \equiv 16^3 \equiv 4 \mod 33$

$x' \equiv x \mod 33$,

assinatura válida!

Ataque 1 contra assinatura RSA simplificada

- Possível assinar algumas mensagens específicas
  - Exemplo: fácil computar a inversa da e-ésima potência para $m = 1$
- Deveria ser difícil para toda mensagem

Ataque 2 contra assinatura RSA simplificada

- Possível assinar mensagens arbitrárias
- Exemplo: fazer de trás para frente
  - Escolher assinatura σ arbitrário; fazer $m = [\sigma^e \mod N]$
- Mas viola exigência de um atacante não conseguir gerar assinaturas válidas!

Ataque 3 contra assinatura RSA simplificada

- Possível combinar duas assinaturas para obter uma terceira
  - Sejam σ_1, σ_2 assinaturas válidas em m_1, m_2 em relação à chave pública N, e
  - Então $\sigma' = [\sigma_1 \cdot \sigma_2 \mod N]$ é uma assinatura válida na mensagem $m' = [m_1 \cdot m_2 \mod N]$
    - $(\sigma_1 \cdot \sigma_2)^e = \sigma_1^e \cdot \sigma_2^e = m_1 \cdot m_2 \mod N$

Paradigma Hash-and-Sign

- Esquema de assinatura $\Pi = (gen, sign, ver)$ para mensagens de comprimento arbitrário:
  - $sign_{sk}(m) = \sigma \leftarrow sign_{sk}(H(m))$
  - $ver_{pk}(m, \sigma) = ver_{pk}(H(m), \sigma)$
- Usando a função hash $H : \{0,1\}^* \rightarrow \{0,1\}^n$

RSA-FDH

- Full Domain Hash (FDH)
- Aplicar uma transformação criptográfica H na mensagem antes de assinar
  - Uma função hash especial
- Chave pública: (N, e), Chave privada: d
- $sign_{sk}(m) = H(m)^d \mod N$
- $ver_{pk}(m, \sigma)$: assinatura válida se e só se $\sigma^e = H(m) \mod N$
- Vantagem de extender uso do RSA para mensagens de tamanho arbitrário maior que o módulo $n = p \cdot q$

Intuição

- Observar que os 3 ataques contra assinatura RSA simplificada não funcionam contra RSA-FDH:
  1. Não é fácil computar a inversa da e-ésima potência de $H(1)$
  2. É fácil escolher σ mas não é fácil achar um m tal que

$$H(m) = \sigma^e \mod N$$

     - Computar inversas de H deve ser difícil
  3. $H(m_1) \cdot H(m_2) = \sigma_1^e \cdot \sigma_2^e = (\sigma_1 \cdot \sigma_2)^e \neq H(m_1 \cdot m_2)$

RSA-FDH na prática

- RSA-FDH é seguro se H for um mapeamento aleatório em $\mathbb{Z}_n^*$
- H deve ter saída com número de bits igual a $|\mathbb{Z}_n^*|$
- O padrão RSA PKCS #1 v2.1 inclue um esquema de assinatura inspirado pelo RSA-FDH
  - Igual a RSA-FDH se assinante não usa aleatoriedade (salt)

Segurança

- Esquema RSA determinístico não é seguro contra CPA
- Segurança de RSA somente refere-se à dificuldade de computar a inversa da potência e para uma m uniforme
  - Como m não é uniforme, RSA simplificado não deve ser usado

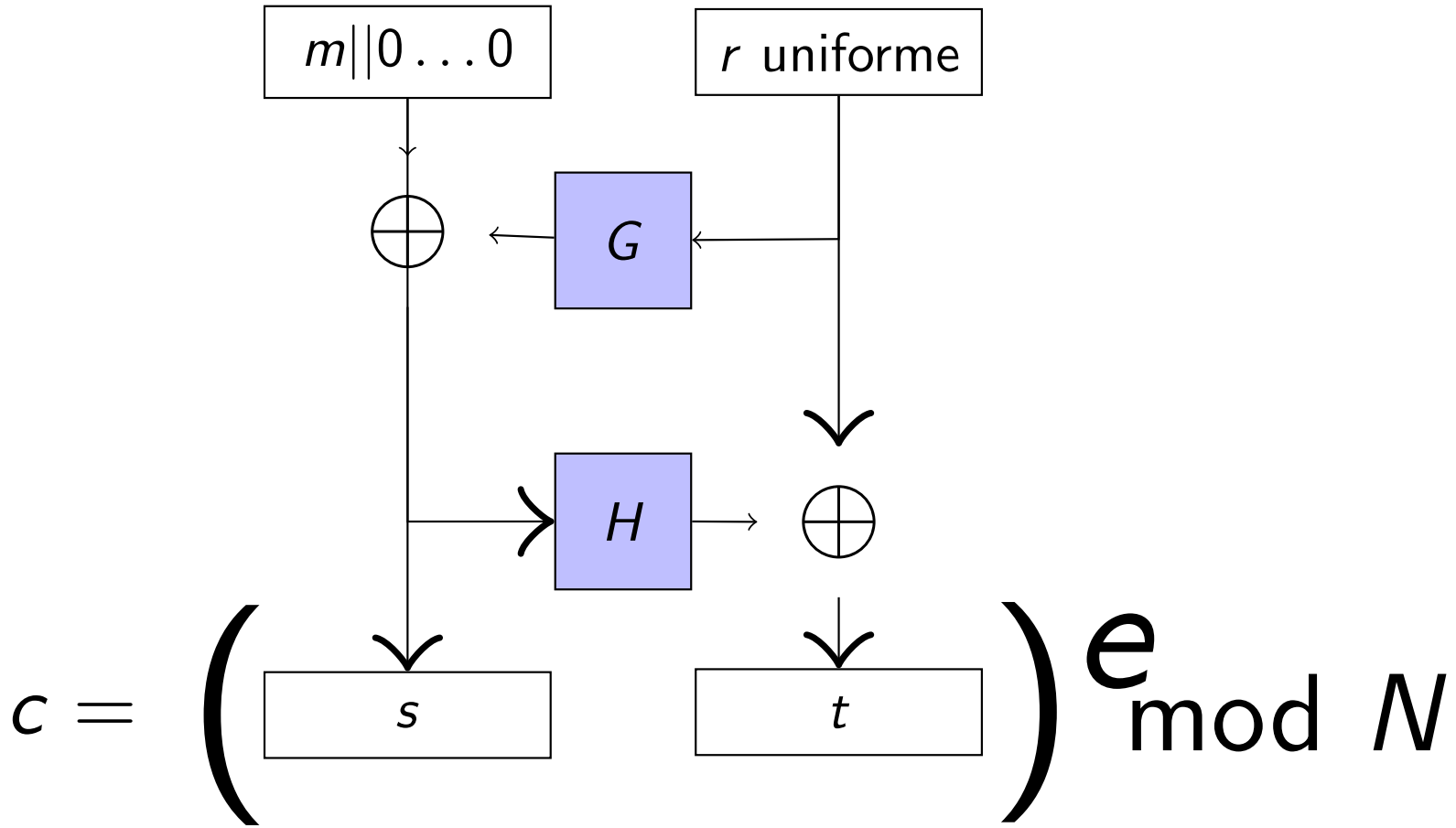
PKCS #1 v1.5

- Public Key Cryptography Standards (PKCS) da RSA Labs (1993)
- Ideia: incluir padding aleatório
  - Para encriptar m, escolher r aleatório
  - $c = [(r|m)^e \mod n]$
- Problemas:
  - Sem prova de segurança contra CPA a não ser para m curtas
  - Existem ataques de texto em claro escolhido para r curto
  - Existem ataques de texto cifrado escolhido

PKCS #1 v2.0

- Optimal asymmetric encryption padding - OAEP (Bellare e Rogaway,1994)
- Esse padding adiciona redundância tal que nem todo $c \in \mathbb{Z}_N^*$ é um texto cifrado válido
  - Necessário checar se formato está adequado ao decriptar
  - Retornar erro se não está no formato adequado

OAEP



- G e H são "oráculos aleatórios" similar a FDH

Segurança

- RSA-OAEP pode ser provada ser seguro contra CCA sob a premissa de RSA, se G e H são modelados como oráculos aleatórios
- Usado amplamente na prática