Esquemas assimétricos

- Cifra
- Assinatura digital

ISEL – Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1 | 1959-007 Lisboa

Revisão

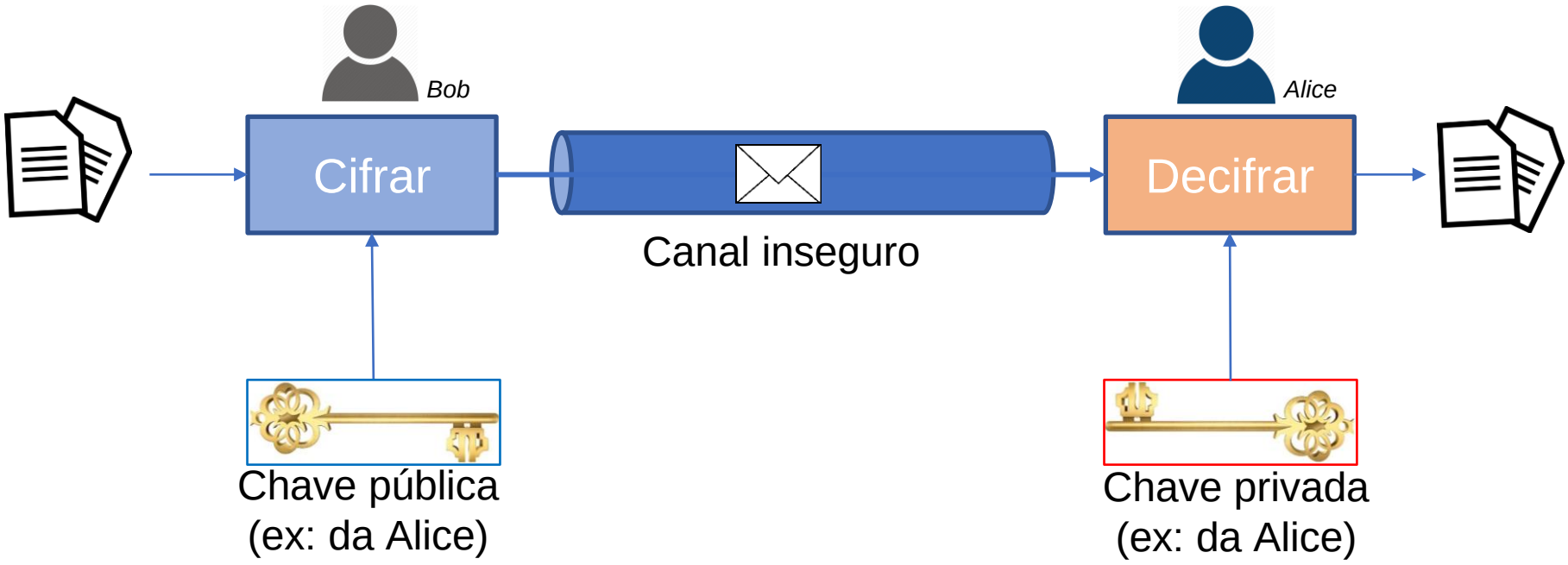
- Esquemas simétricos
  - Cifra e autenticidade
- Esquemas assimétricos
  - Cifra e assinatura digital

	Simétrico	Assimétrico
Confidencialidade	Cifra simétrica	Cifra assimétrica
Autenticidade	MAC	Assinatura Digital

Esquemas Assimétricos

- Esquemas simétricos
  - A mesma chave é utilizada na cifra e na decifra
  - A mesma chave é utilizada na geração da marca e na verificação da marca
- Esquemas assimétricos
  - Esquemas de cifra – qual a operação privada?
    - "Todos podem cifrar, apenas o receptor autorizado pode decifrar"
  - Esquemas de autenticação – qual a operação privada?
    - "Todos podem verificar, apenas o emissor autorizado pode assinar (gerar a marca)"
- Utilização
  - Transporte seguro de chaves simétricas
  - Assinatura digital

Crifra assimétrica – visão geral



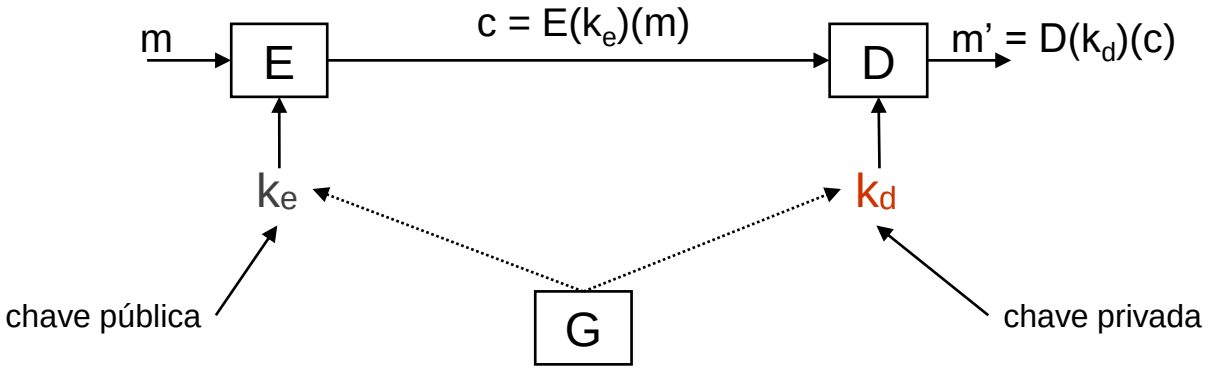
Esquema de cifra assimétrica

- Esquema de cifra assimétrica – algoritmos **(G,E,D)**
  - **G** – função (probabilística) de geração de pares de chaves

    G: $\rightarrow$ **KeyPairs** , onde **KeyPairs** $\subseteq$ **PublicKeys** $\times$ **PrivateKeys**
  - **E** – função (probabilística) de cifra

    E: **PublicKeys** $\rightarrow$**PlainTexts** $\rightarrow$ **CipherTexts**
  - **D** – função (determinística) de decifra

    D: **PrivateKeys** $\rightarrow$**CipherTexts** $\rightarrow$ **PlainTexts**



Notas

- Propriedade da correcção
  - $\forall$ m $\in$ M, $\forall$ (k_e,k_d) $\in$ **KeyPairs**: D(k_d)(E(k_e)(m)) = m
- Propriedades de segurança
  - É *computacionalmente infazível* obter **m** a partir de **c**, sem o conhecimento de $\mathbf{k_d}$
- Esquema assimétrico
  - utilização de chaves diferentes para os algoritmos **E** e **D**
- O espaço de mensagens, denotado por **PlainTexts**, é definido por todas as sequências de bits com dimensão menor do que o limite definido pelo esquema
- O espaço de *criptogramas*, denotado por **CipherTexts** , é definido como um sub-conjunto das sequências de *bits* com dimensão menor do que o limite definido pelo esquema
- Não garante integridade

Notas (2)

- Custo computacional significativamente maior do que os esquemas simétricos (maior do que duas ordens de grandeza)

- Limitações na dimensão da informação cifrada
  - Note-se que a entrada de **E** é **PlainTexts** e não **{0,1}***

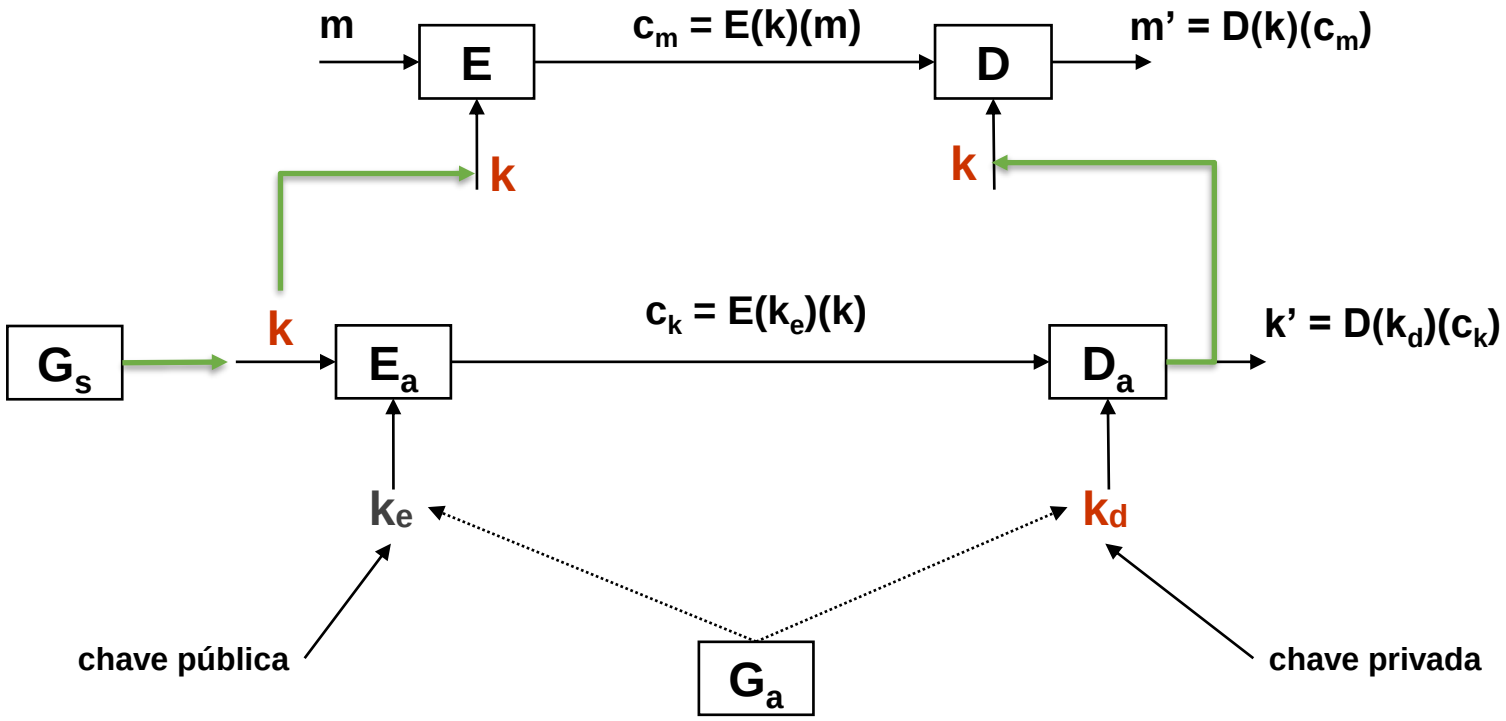
- Utilização em esquemas híbridos
  - Esquema assimétrico usado para cifrar uma chave simétrica – transporte de chaves
  - Esquema simétrico usado para cifrar a informação

Princípios da primitiva RSA

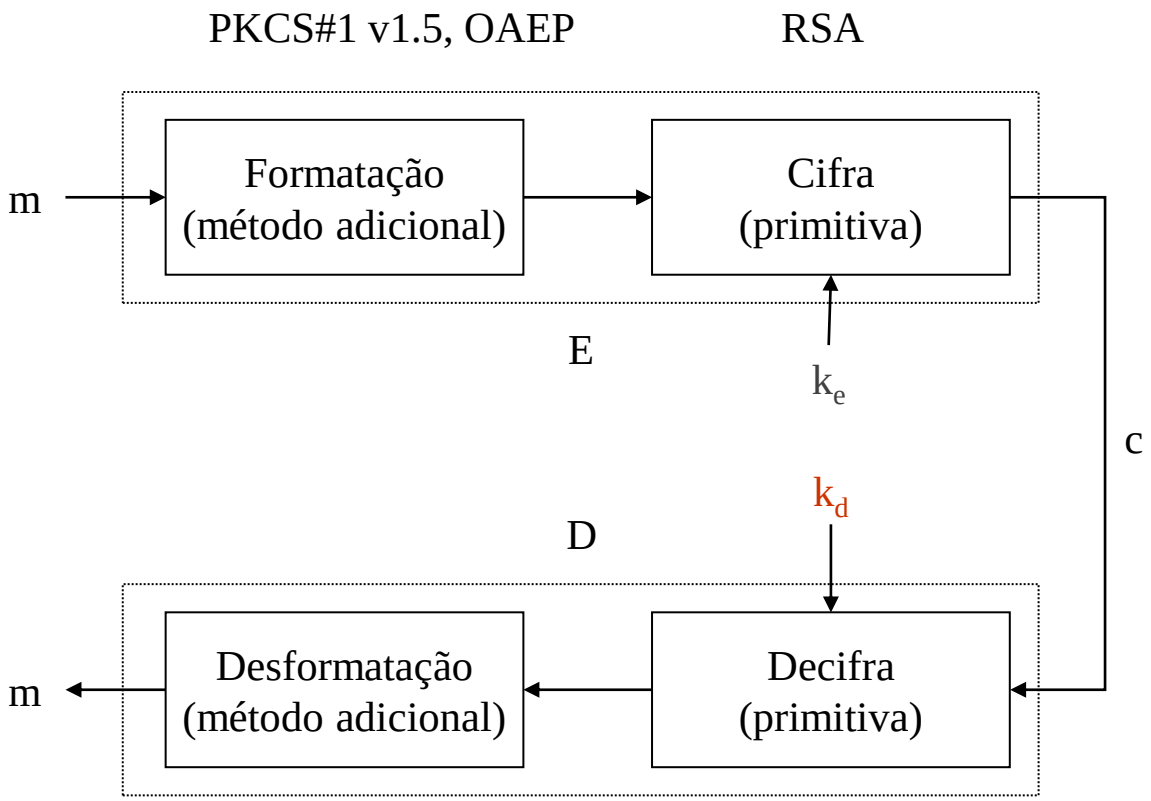
- Sejam P e Q dois primos distintos e N = PQ
  - Dimensões típicas: $2^{1023} \leq N \leq 2^{4095}$
- Sejam E e D tal que ED mod (P-1)(Q-1) = 1
- Par de chaves
  - Chave pública: (E, N)
  - Chave privada: (D, N)
- Operação pública (utilizada na cifra)
  - $C = M^E \bmod N$
- Operação privada (utilizada na decifra)
  - $M = C^D \bmod N$
- A factorização de números primos é o problema que suporta a primitiva RSA

Esquema híbrido

- Devido ao seu elevado custo computacional, a cifra assimétrica é na prática usada para proteger chaves simétricas



Cifra assimétrica: arquitectura interna



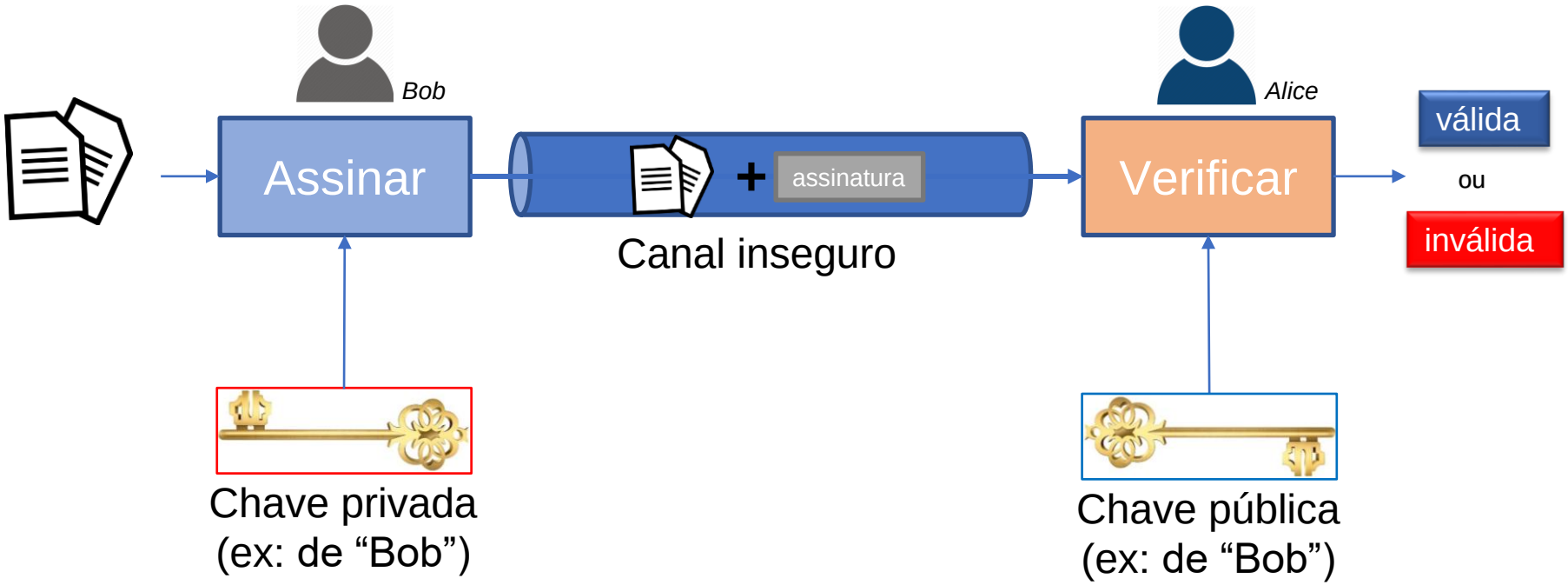
Resumo da cifra assimétrica

- A arquitectura típica dos algoritmos de cifra e decifra dos esquemas de cifra assimétrica é constituída por:
  - Primitiva de cifra assimétrica – ex. RSA
  - Método de formatação ou *padding* – ex. PKCS#1 v1.5, OAEP
- A mesma primitiva pode ser usada com vários tipos de formatação
- A função da formatação é
  - Adequar a entrada do algoritmo (**PlainTexts**) à entrada da primitiva
  - Evitar casos especiais
  - Introduzir informação aleatória
- As chaves são usadas apenas pela primitiva
  - Exemplo: chaves da primitiva RSA podem ser usada nos esquemas RSA+PKCS#1 v1.5 ou RSA+OAEP

Assinatura digital

- Cada interveniente tem 1 par de chaves por cada identidade digital
- Processo de assinatura usa chave privada
  - Ex: Só “Bob” pode assinar
- Processo de verificação usa chave pública
  - Ex: todos podem verificar
- Par de chaves usadas durante um largo período de tempo
- Chave pública difundida através de *certificados digitais*

Assinatura digital – visão geral



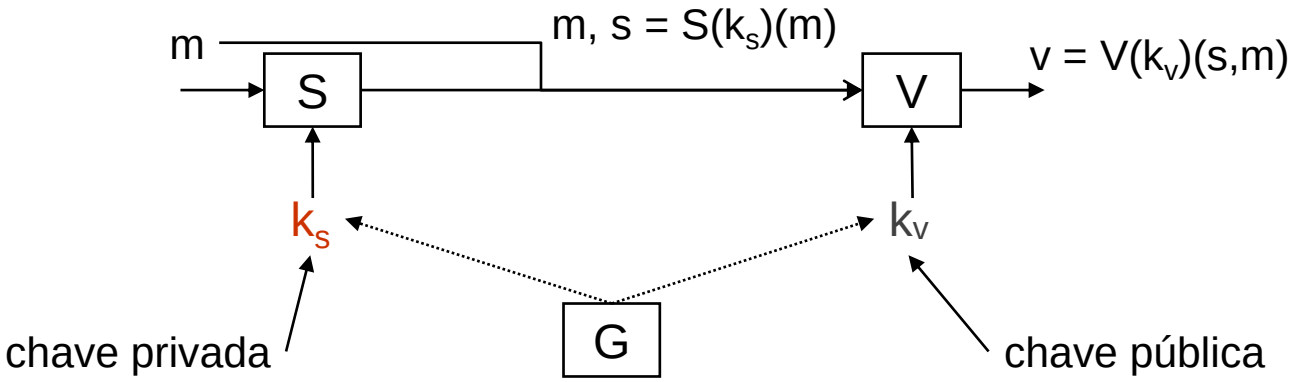
Esquema de assinatura digital

- Esquema de assinatura digital – algoritmos **(G,S,V)**
  - **G** – função (probabilística) de geração de pares de chaves

    G: $\rightarrow$ **KeyPairs** , onde **KeyPairs** $\subseteq$ **PublicKeys** $\times$ **PrivateKeys**
  - **S** – função (probabilística) de assinatura

    S: **PrivateKeys** $\rightarrow$ {0,1}* $\rightarrow$ **Signatures**
  - **V** – função (determinística) de verificação

    V: **PublicKeys** $\rightarrow$ (**Signatures** $\times$ {0,1}*) $\rightarrow$ {true,false}



Notas

- Propriedade da correcção
  - $\forall$ m $\in \{0,1\}^*$, $\forall \in (k_s,k_v) \in$ **KeyPairs**: $V(k_v)(S(k_s)(m),m) = true$
- Propriedades de segurança
  - Sem o conhecimento de $\mathbf{k_s}$ é *computacionalmente infazível*
    - falsificação selectiva – dado **m**, encontrar **s** tal que **$V(k_v)(s, m) = true$**
    - falsificação existencial – encontrar o par **(m,s)** tal que **$V(k_v)(s,m) = true$**

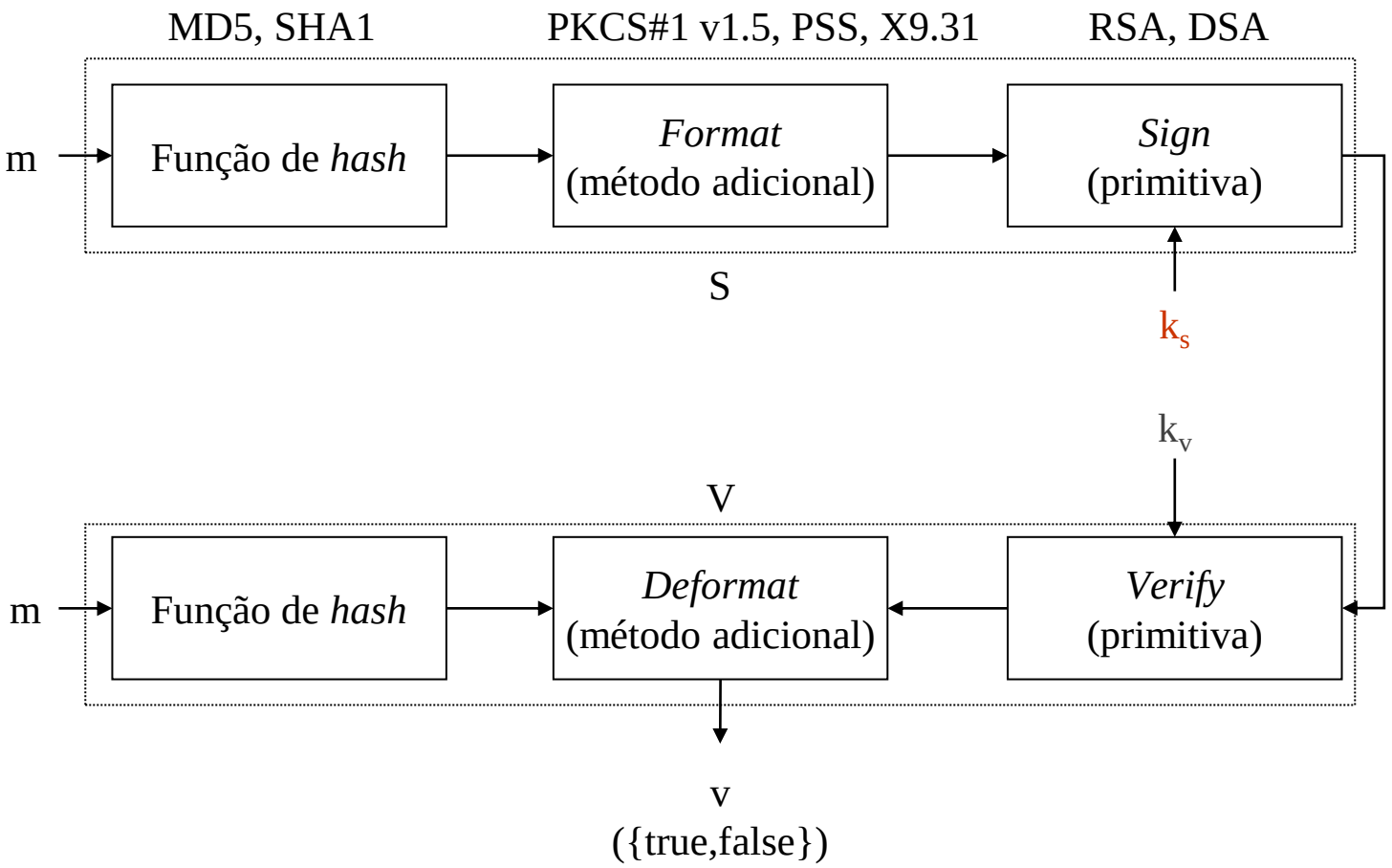
  note-se que $\mathbf{k_v}$ é conhecido

- Assinatura **s** (pertencente ao conjunto **Signatures**) tem tipicamente dimensão fixa
  - Ex.: 160, 1024, 2048 *bits*
- Custo computacional significativamente maior do que os esquemas simétricos

Notas (2)

- Assimétrico
  - utilização de chaves diferentes para os algoritmos **S** e **V**
- Mensagem **m** é uma sequência de *bytes* de dimensão variável
- assinar ≠ decifrar; verificar ≠ cifrar

Assinatura digital: arquitectura interna



Assinatura digital: arquitectura interna

- A arquitectura típica dos algoritmos de assinatura e verificação dos esquemas de assinatura digital é constituída por:
  - Primitiva de assinatura/verificação assimétrica – ex. RSA
  - Método de formatação ou *padding* – ex. PKCS#1 v1.5, PSS
  - Função de *hash*

- A mesma primitiva pode ser usada com vários tipos de formatação e funções de *hash*

- As chaves são usadas apenas pela primitiva
  - Exemplo: chaves da primitiva RSA podem ser usada nos esquemas RSA+PKCS#1 v1.5 ou RSA+PSS