

---

# Mecanismos Criptográficos Esquemas

---

Notas para a UC de “Segurança Informática”  
Inverno de 10/11

Pedro Félix ([pedrofelix em cc.isel.ipl.pt](mailto:pedrofelix@cc.isel.ipl.pt))

José Simão ([jsimao em cc.isel.ipl.pt](mailto:jsimao@cc.isel.ipl.pt))

[Instituto Superior de Engenharia de Lisboa](#)

# Sumário

---

- Hierarquia de mecanismos criptográficos
- Funcionalidade de esquemas criptográficos
  - Esquemas simétricos e assimétricos de cifra
  - Esquemas MAC e esquemas de assinatura digital
  - Funções de *hash*
- Arquitectura interna dos esquemas assimétricos

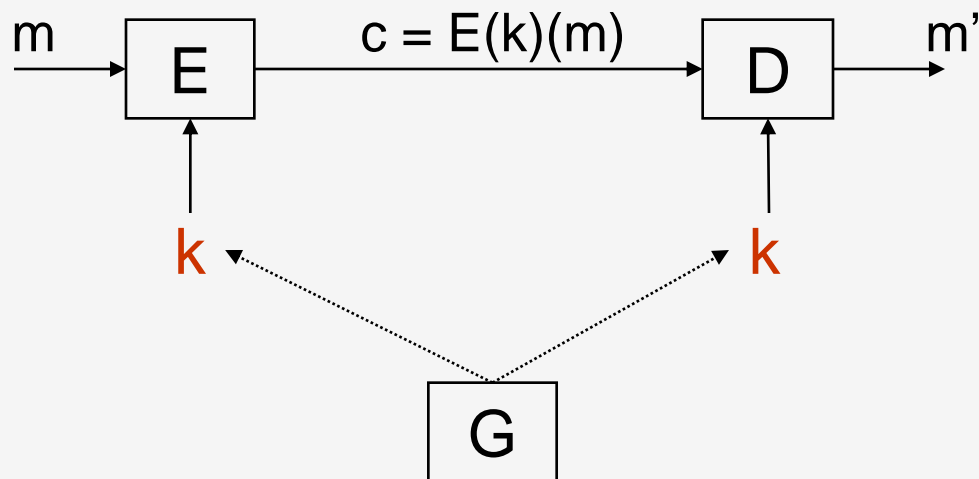
# Classificação dos mecanismos criptográficos

---

- Primitivas – operações matemáticas, usadas como blocos construtores na realização de esquemas; a sua caracterização depende dos problemas matemáticos que sustentam a sua utilização criptográfica
  - ex: DES, RSA
- Esquemas – combinação de primitivas e métodos adicionais para a realização de tarefas criptográficas como a cifra e a assinatura digital
  - ex: DES-CBC-PKCS5Padding; RSA-OAEP-MGF1-SHA1
- Protocolos – sequências de operações, a realizar por duas ou mais entidades, envolvendo esquemas e primitivas, com o propósito de dotar uma aplicação com características seguras
  - ex: TLS com TLS\_RSA\_WITH\_DES\_CBC\_SHA

# Esquema de cifra simétrica

- Esquema de cifra simétrica – algoritmos **(G,E,D)**
  - **G** – função (probabilística) de geração de chaves  
**G:  $\rightarrow$  Keys**
  - **E** – função (probabilística) de cifra  
**E: Keys  $\rightarrow \{0,1\}^* \rightarrow \{0,1\}^*$**
  - **D** – função (determinística) de decifra  
**D: Keys  $\rightarrow \{0,1\}^* \rightarrow \{0,1\}^*$**



- Propriedade da correcção
  - $\forall m \in \{0,1\}^*, \forall k \in \mathbf{Keys}: D(k)(E(k)(m)) = m$ 
    - **Keys** é o conjunto de chaves geradas por G
- Propriedades de segurança
  - É *computacionalmente infazível* obter **m** a partir de **c**, sem o conhecimento de **k**
- Esquema simétrico
  - utilização da mesma chave **k** nas funções E e D
- Mensagem **m** e *criptograma* **c** são sequências de *bytes* com dimensão variável ( $\{0,1\}^*$ )
- Não garante integridade
- Exemplos:
  - DES-CBC-PKCS5Padding

# Esquemas MAC

- Esquema MAC (*Message Authentication Codes*) – algoritmos (**G,T,V**)

- **G** – função (probabilística) de geração de chaves

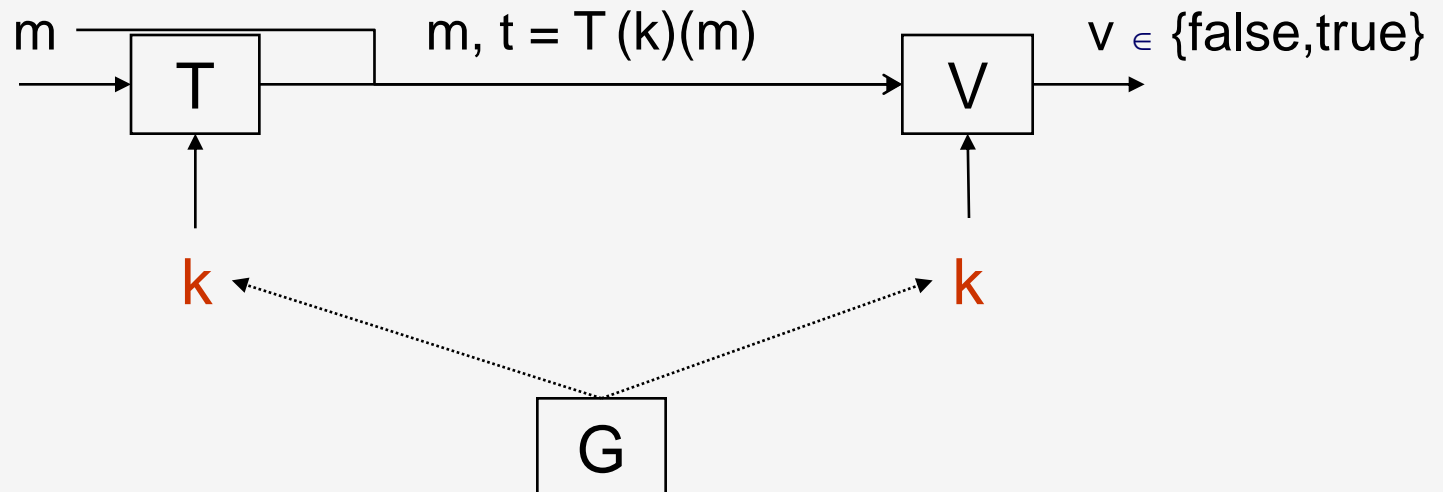
**G**:  $\rightarrow \mathbf{Keys}$

- **T** – função (probabilística) de geração de marcas

**T**:  $\mathbf{Keys} \rightarrow \{0,1\}^* \rightarrow \mathbf{Tags}$

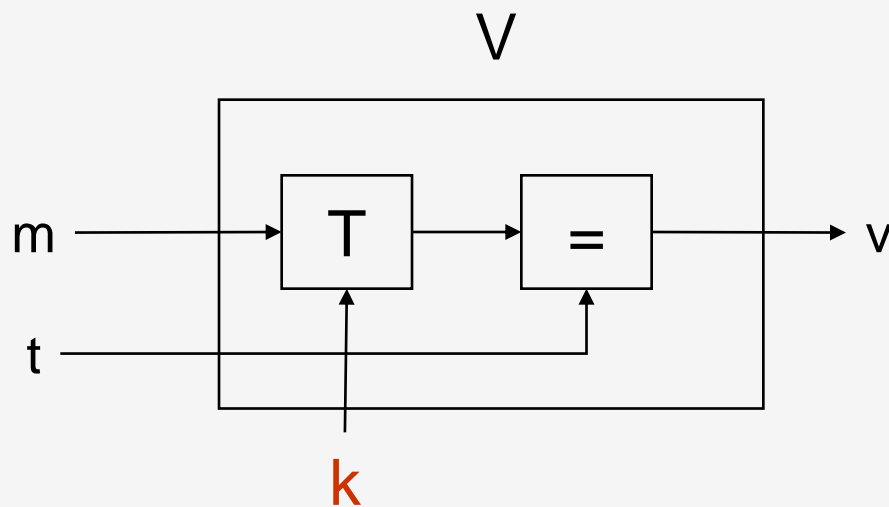
- **V** – função (determinística) de verificação de marcas

**V**:  $\mathbf{Keys} \rightarrow (\mathbf{Tags} \times \{0,1\}^*) \rightarrow \{\text{true}, \text{false}\}$



# Esquemas MAC: verificação

- Esquema usual para o algoritmo de verificação
  - Algoritmo **T** é determinístico
  - Algoritmo **V** usa **T**
  - **V(k)(t, m): T(k)(m) = t**



- Propriedade da correcção
  - $\forall \mathbf{m} \in \{0,1\}^*, \forall \mathbf{k} \in \mathbf{Keys}: \mathbf{V}(\mathbf{k})(\mathbf{T}(\mathbf{k})(\mathbf{m}), \mathbf{m}) = \text{true}$
- Propriedades de segurança
  - Sem o conhecimento de  $\mathbf{k}$ , é *computacionalmente infazível*
    - falsificação selectiva – dado  $\mathbf{m}$ , encontrar  $\mathbf{t}$  tal que  $\mathbf{V}(\mathbf{k})(\mathbf{t}, \mathbf{m}) = \text{true}$
    - falsificação existencial – encontrar o par  $(\mathbf{m}, \mathbf{t})$  tal que  $\mathbf{V}(\mathbf{k})(\mathbf{t}, \mathbf{m}) = \text{true}$
- Esquema simétrico
  - utilização da mesma chave  $\mathbf{k}$  nos algoritmos  $\mathbf{T}$  e  $\mathbf{V}$
- Mensagem  $\mathbf{m}$  é uma sequência de *bytes* de dimensão variável
- Marca  $\mathbf{t}$  (*tag*) tem tipicamente dimensão fixa
  - 128, 160, 256 *bits*
- Códigos detectores e correctores de erros não servem para esquemas de MAC
- Exemplos:
  - HMAC-SHA1



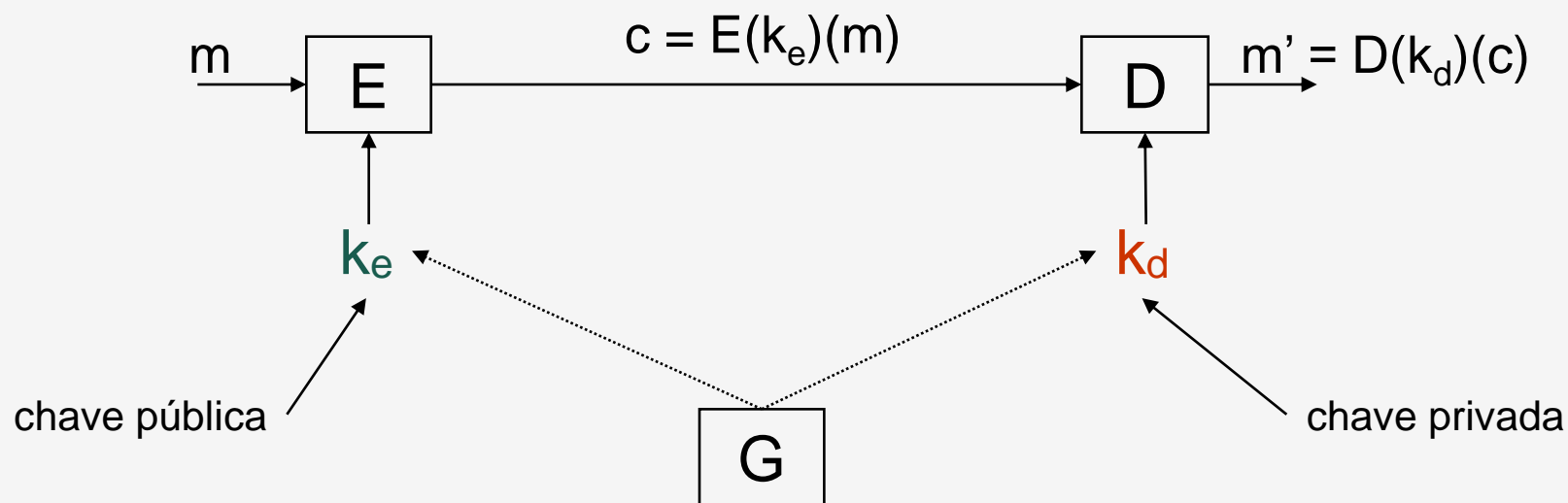
# Esquemas Assimétricos

---

- 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
- A cifra é uma operação pública?
- A verificação é uma operação pública?
- Esquemas assimétricos
  - Esquemas de cifra – qual a operação privada?
    - “Todos podem cifrar, apenas o receptor autorizado pode decifrar”
  - Esquemas MAC – qual a operação privada?
    - “Todos podem verificar, apenas o emissor autorizado pode assinar (gerar a marca)”
- Utilização
  - Transporte de chaves simétricas
  - Assinatura digital

# Esquema de cifra assimétrica

- Esquema de cifra assimétrica – algoritmos **(G,E,D)**
  - **G** – função (probabilística) de geração de pares de chaves  
 $G: \rightarrow \mathbf{KeyPairs}$  , onde  $\mathbf{KeyPairs} \subseteq \mathbf{PublicKeys} \times \mathbf{PrivateKeys}$
  - **E** – função (probabilística) de cifra  
 $E: \mathbf{PublicKeys} \rightarrow \mathbf{PlainTexts} \rightarrow \mathbf{CipherTexts}$
  - **D** – função (determinística) de decifra  
 $D: \mathbf{PrivateKeys} \rightarrow \mathbf{CipherTexts} \rightarrow \mathbf{PlainTexts}$



- Propriedade da correcção
  - $\forall m \in M, \forall (k_e, k_d) \in \mathbf{KeyPairs}: D(k_d)(E(k_e)(m)) = m$
- Propriedades de segurança
  - É *computacionalmente infazível* obter **m** a partir de **c**, sem o conhecimento de **k<sub>d</sub>**
- 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
  - Os esquemas de cifra assimétrica são utilizados para cifrar chaves
- 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

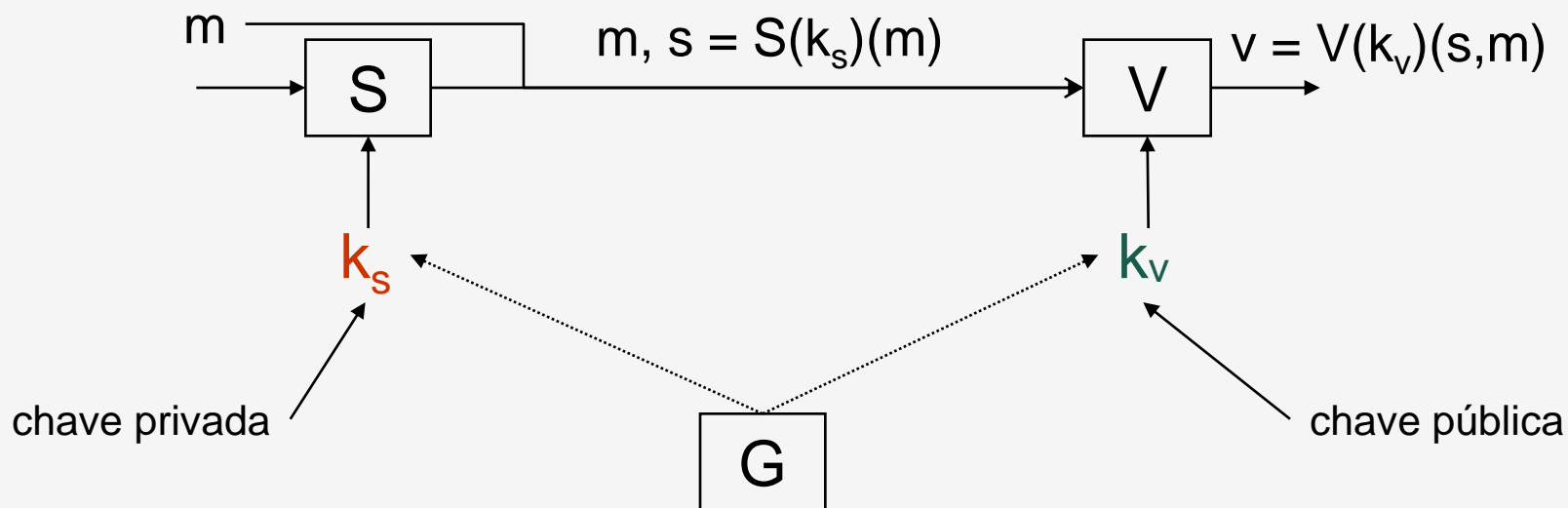
## Notas (2)

---

- 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

# Esquema de assinatura digital

- Esquema de assinatura digital – algoritmos **(G,S,V)**
  - **G** – função (probabilística) de geração de pares de chaves  
 $G: \rightarrow \mathbf{KeyPairs}$  , onde  $\mathbf{KeyPairs} \subseteq \mathbf{PublicKeys} \times \mathbf{PrivateKeys}$
  - **S** – função (probabilística) de assinatura  
 $S: \mathbf{PrivateKeys} \rightarrow \{0,1\}^* \rightarrow \mathbf{Signatures}$
  - **V** – função (determinística) de verificação  
 $V: \mathbf{PublicKeys} \rightarrow (\mathbf{Signatures} \times \{0,1\}^*) \rightarrow \{\text{true}, \text{false}\}$



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

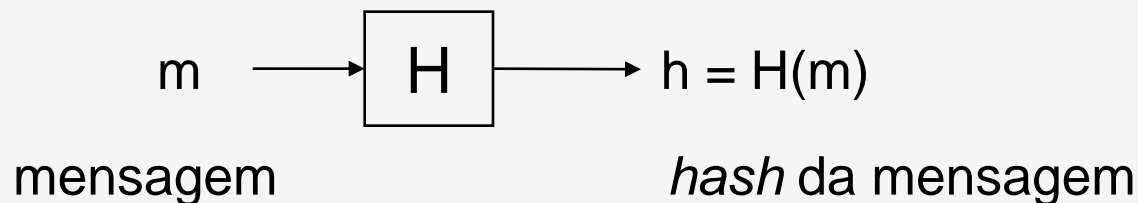
---

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

# Funções de *hash*

---

- Função de *hash* criptográfica
  - **H**:  $\{0,1\}^* \rightarrow \{0,1\}^n$ , onde **n** é a dimensão do *hash*
  - Entrada:
    - Sequências binárias de dimensão finita
  - Saída:
    - Sequência binária de dimensão fixa (**n**)
  - **n** é a dimensão do *hash*

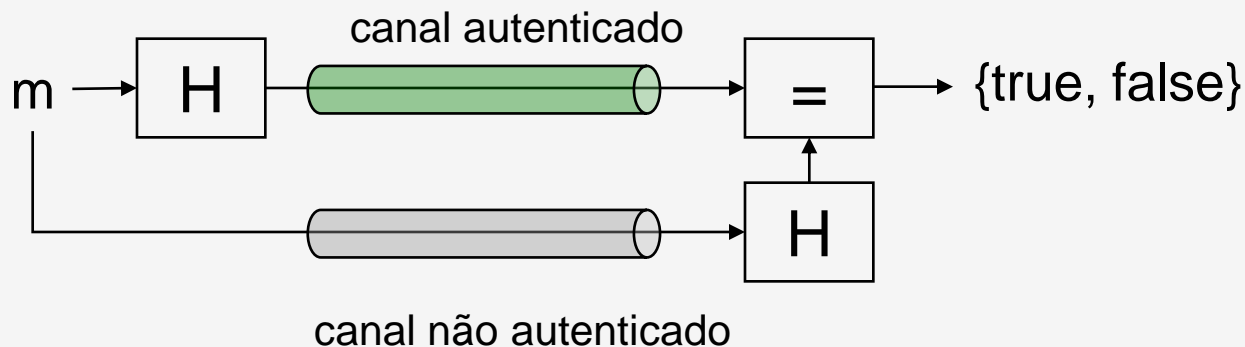




- Propriedades de segurança
  - É *computacionalmente fácil* obter  $H(x)$  dado  $x$
  - É *computacionalmente difícil*, dado  $x$ , obter  $x' \neq x$  tal que  $H(x') = H(x)$
  - É *computacionalmente difícil* obter  $(x, x')$ , com  $x' \neq x$ , tal que  $H(x) = H(x')$
- O *hash* de  $m$  serve como representante (“*impressão digital*”) de  $m$
- Exemplos de dimensões: MD5 ( $n=128$ ) e SHA-1 ( $n=160$ )
- Baseiam-se em operações *booleanas* e aritméticas sobre palavras de pequena dimensão (16, 32, 64 *bit*)

# Exemplo de utilização: integridade

- Exemplo: Distribuição de *software*
  - Produtor calcula o *hash* da distribuição (ex. sources.tar.gz)
  - Cliente obtêm, de forma autenticada, o *hash* da distribuição
  - Cliente obtêm a distribuição (ex. através dum *mirror* não autenticado)
  - Cliente compara o *hash* da distribuição recebida com o *hash* obtido em 2

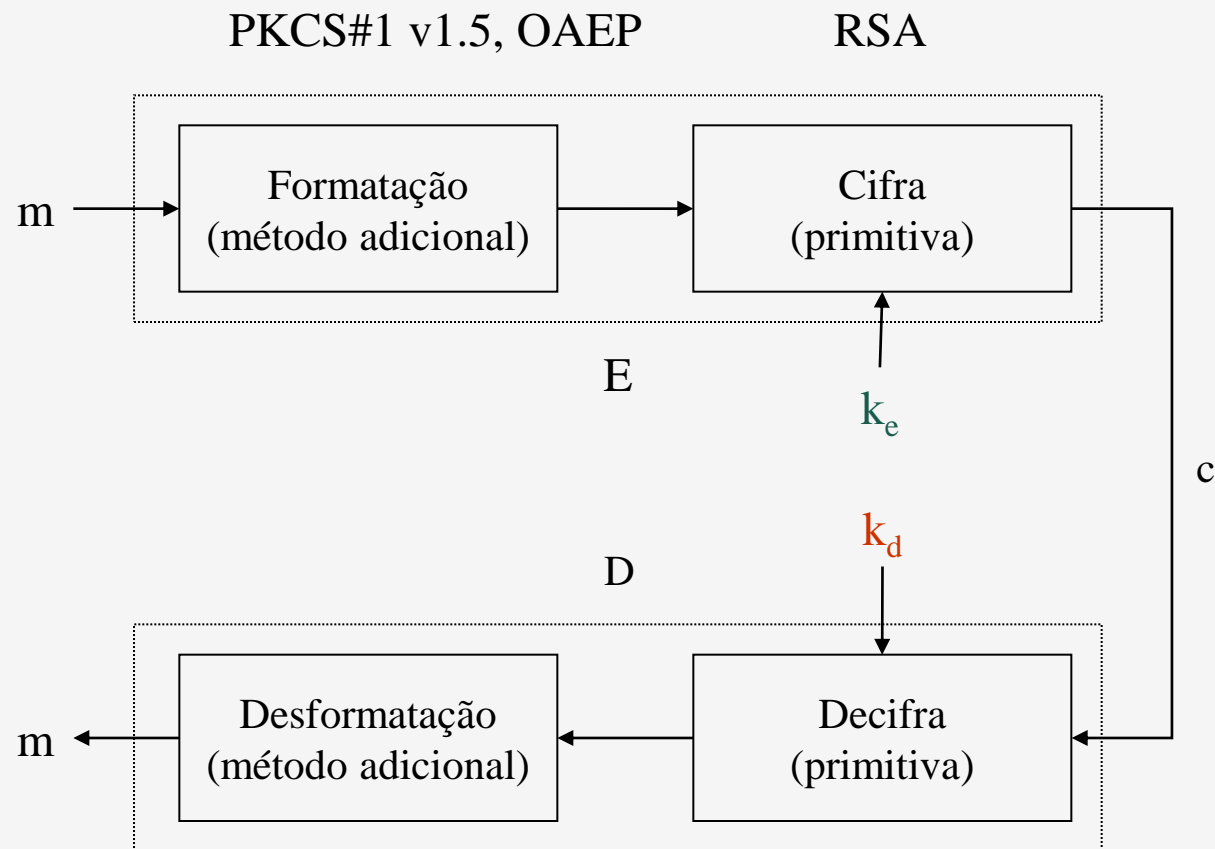


## Funções de *hash* com chave

---

- É usual designar-se um esquema de MAC, com algoritmo **T** determinístico, como *função de hash com chave* (*Keyed Hash Function*)
- Em alguns contextos, as funções de *hash* são designadas por *Manipulation Detection Codes* (MDC)

# Cifra assimétrica: arquitetura interna

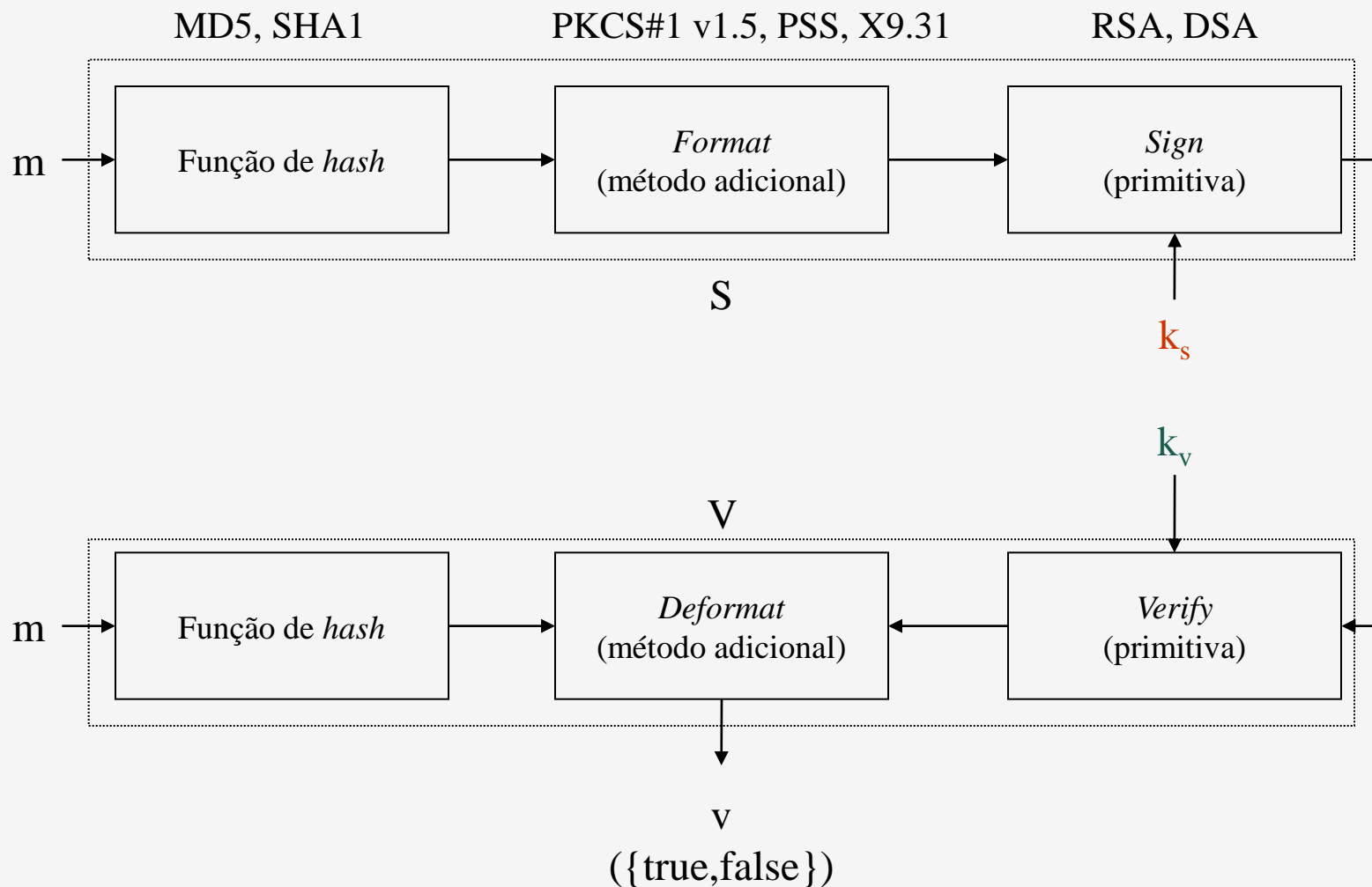


# Cifra assimétrica

---

- 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: arquitetura interna



# Assinatura digital

---

- 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