Mecanismos Criptográficos
Esquemas

Notas para a UC de “Segurança Informática”

Pedro Félix (pedrofelix *em* cc.isel.ipl.pt)
José Simão (jsimao *em* 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

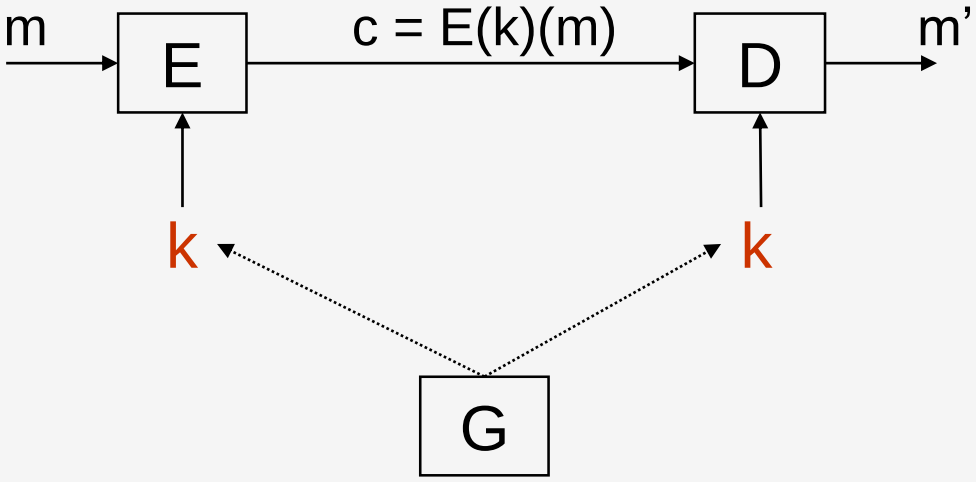
- Primitivas – operações matemáticas, usadas como blocos construtores na realização de esquemas; a sua caracterização depende dos problemas matemáticos que sustentam a sua utilização criptográfica
  - ex: DES, RSA

- 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}*

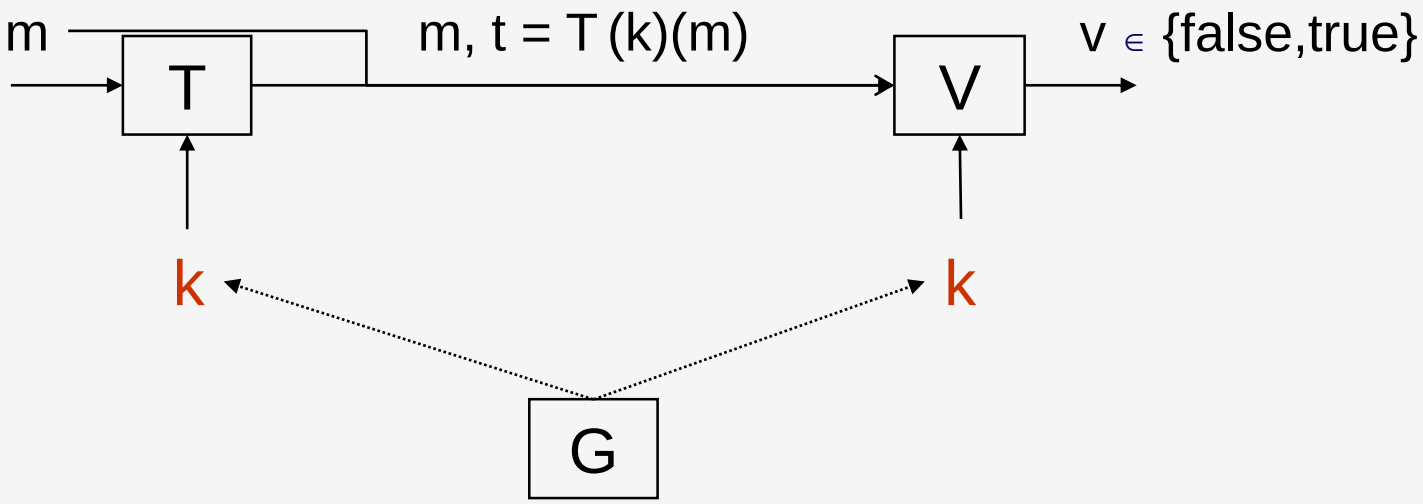


Notas

- Propriedade da correcção
  - $\forall$ m $\in$ {0,1}*, $\forall$ k $\in$ **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

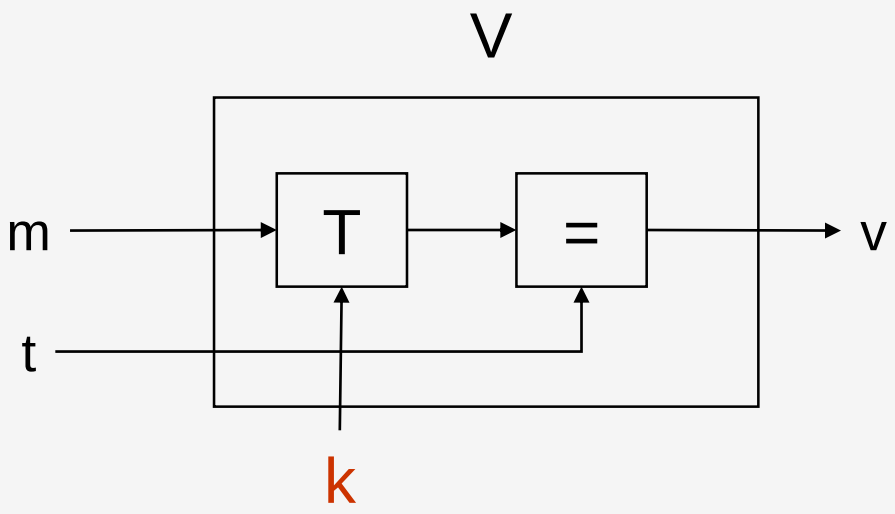
Esquemas MAC

- Esquema MAC (*Message Authentication Codes*) – algoritmos **(G,T,V)**
  - **G** – função (probabilística) de geração de chaves
    G: $\rightarrow$ **Keys**
  - **T** – função (probabilística) de geração de marcas
    T: **Keys** $\rightarrow$ {0,1}* $\rightarrow$ **Tags**
  - **V** – função (determinística) de verificação de marcas
    V: **Keys** $\rightarrow$ (**Tags** × {0,1}*) $\rightarrow$ {true, 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**



Notas

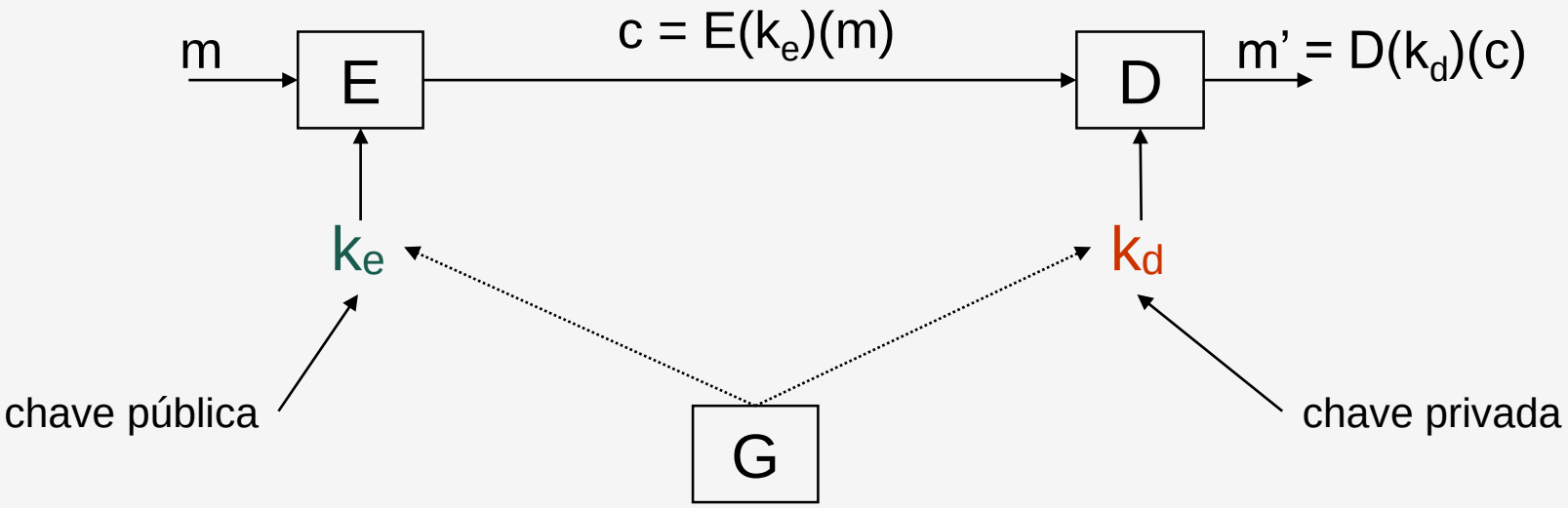
- Propriedade da correcção
  - $\forall$ **m** $\in \{0,1\}^*$, $\forall$ **k** $\in$ **Keys**: **V(k)(T(k)(m),m) = true**
- Propriedades de segurança
  - Sem o conhecimento de **k**, é *computacionalmente infazível*
    - falsificação selectiva – dado **m**, encontrar **t** tal que **V(k)(t, m) = true**
    - falsificação existencial – encontrar o par **(m, t)** tal que **V(k)(t,m) = true**
- Esquema simétrico
  - utilização da mesma chave **k** nos algoritmos **T** e **V**
- Mensagem **m** é uma sequência de *bytes* de dimensão variável
- Marca **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: $\rightarrow$ **KeyPairs** , onde **KeyPairs** $\subseteq$ **PublicKeys** × **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
  - 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

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

Exemplo de utilização da primitiva RSA

- Exemplo com número primos pequenos

P	Q	N	(P-1)(Q-1)	E	D
17	11	187	160	23	7

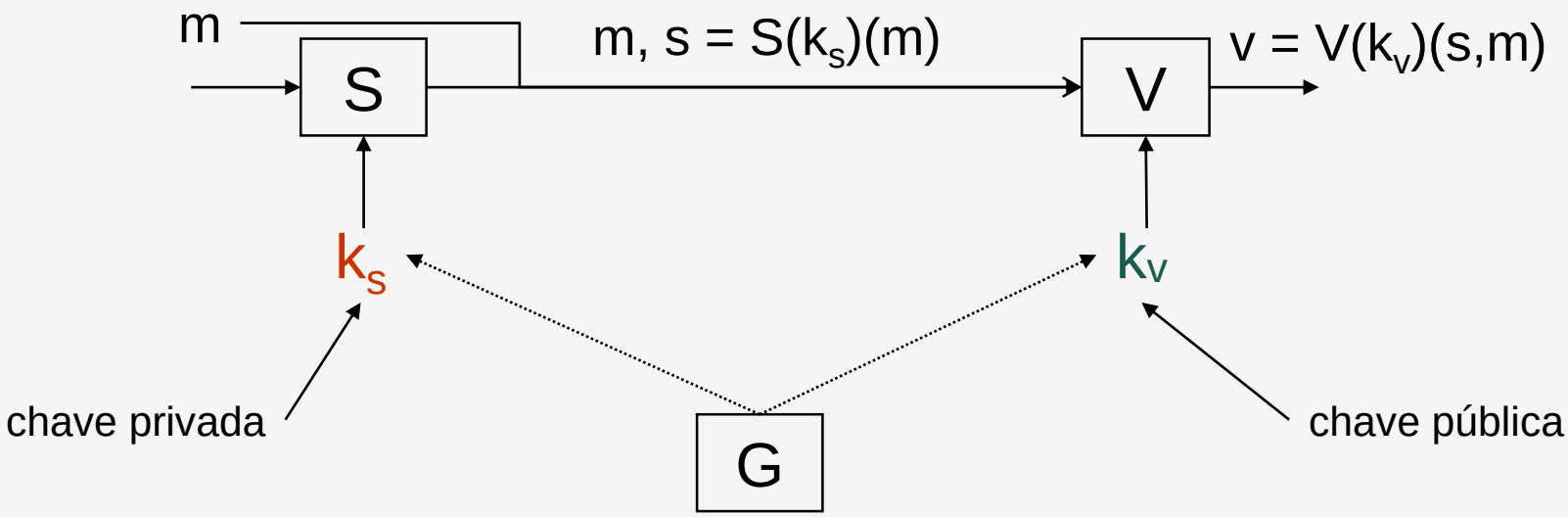
$$c = m^e \bmod n$$

Mensagem (m)	I	S	E	L
Criptograma (c)	183	125	137	80

- A utilização da primitiva por si só não está imune aos seguintes ataques:
  - Reordenação dos criptogramas
  - Pré-computação de mensagens e comparação com criptogramas

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** × **PrivateKeys**
  - **S** – função (probabilística) de assinatura
    S: **PrivateKeys** $\rightarrow$ {0,1}* $\rightarrow$ **Signatures**
  - **V** – função (determinística) de verificação
    V: **PublicKeys** $\rightarrow$ (**Signatures** × {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 $\mathbf{V(k_v)(s, m) = true}$
    - falsificação existencial – encontrar o par **(m,s)** tal que $\mathbf{V(k_v)(s,m) = true}$

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

- Assinatura **s** (pertencente ao conjunto **Signatures**) tem tipicamente dimensão fixa
  - Ex.: 160, 1024, 2048 *bits*

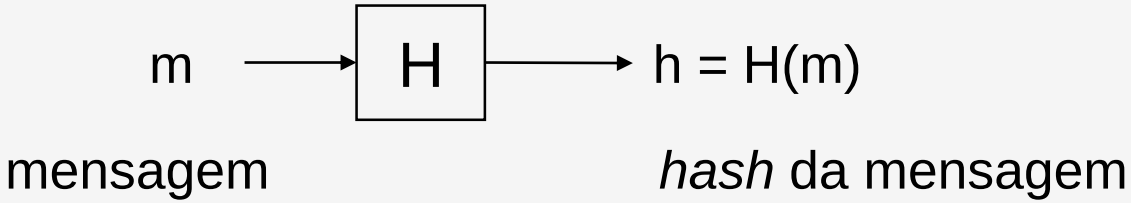
- Custo computacional significativamente maior do que os esquemas simétricos

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

Funções de *hash*

- Função de *hash* criptográfica
  - **H: {0,1}* → {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*



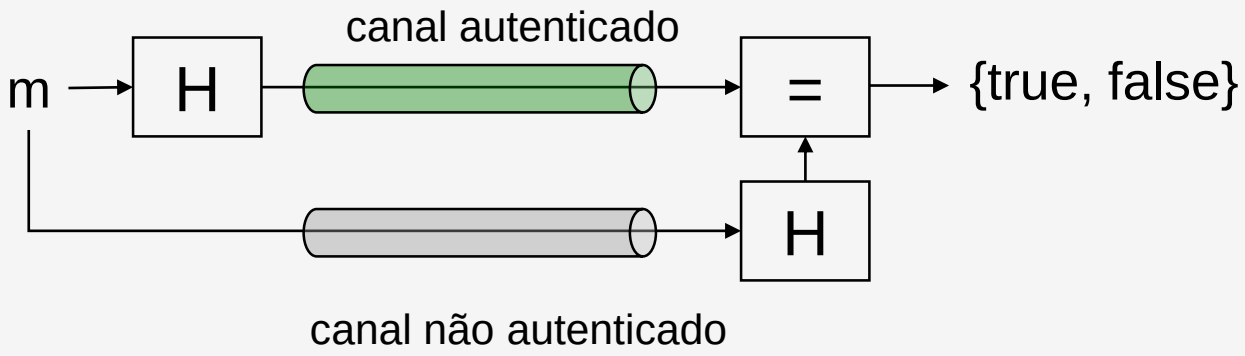
Notas

- Propriedades de segurança
  - É *computacionalmente fácil* obter **H(x)** dado **x**
  - É *computacionalmente difícil*, dado **x**, obter **x'≠x** tal que **H(x') = H(x)**
    - *Segunda pré-imagem*
  - É *computacionalmente difícil* obter **(x, x')**, com **x'≠x**, tal que **H(x) = H(x')**
    - colisão

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

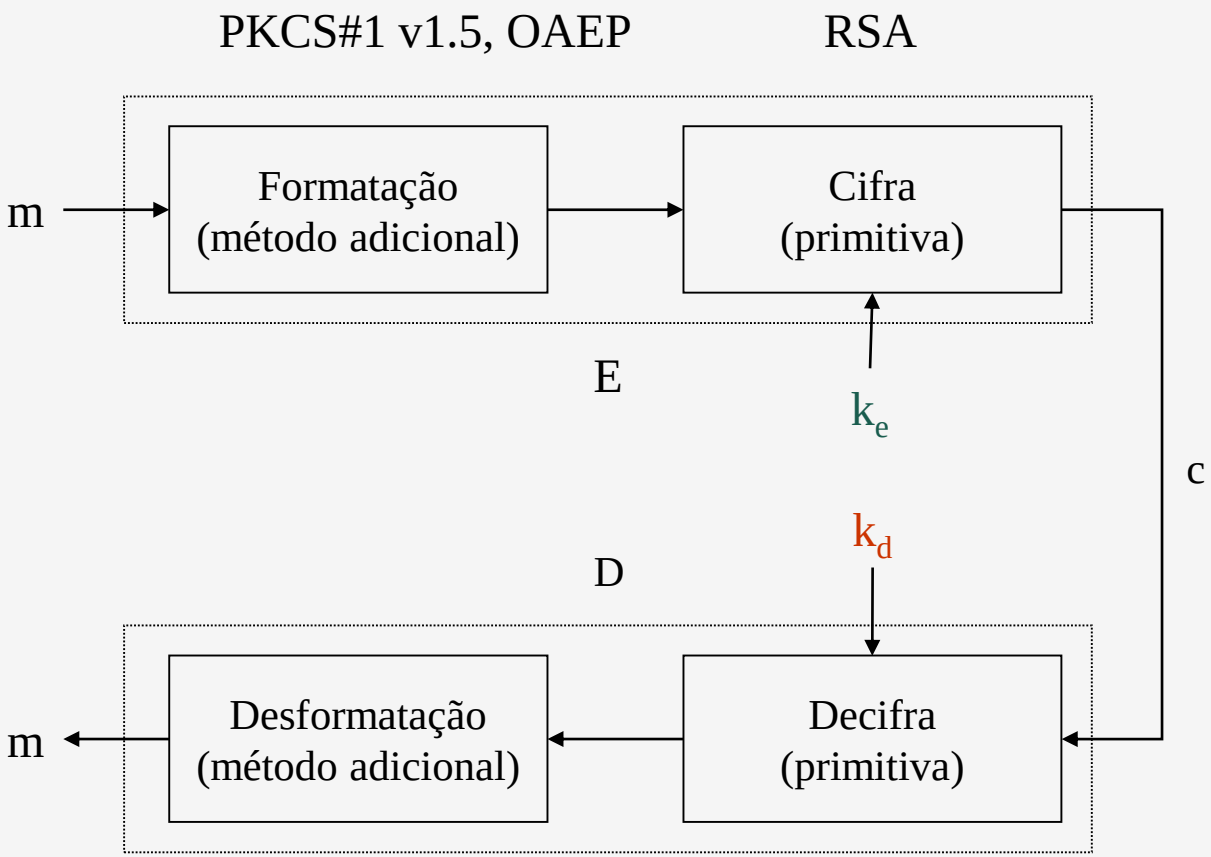


Funções de *hash* com chave

- 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: arquitectura interna



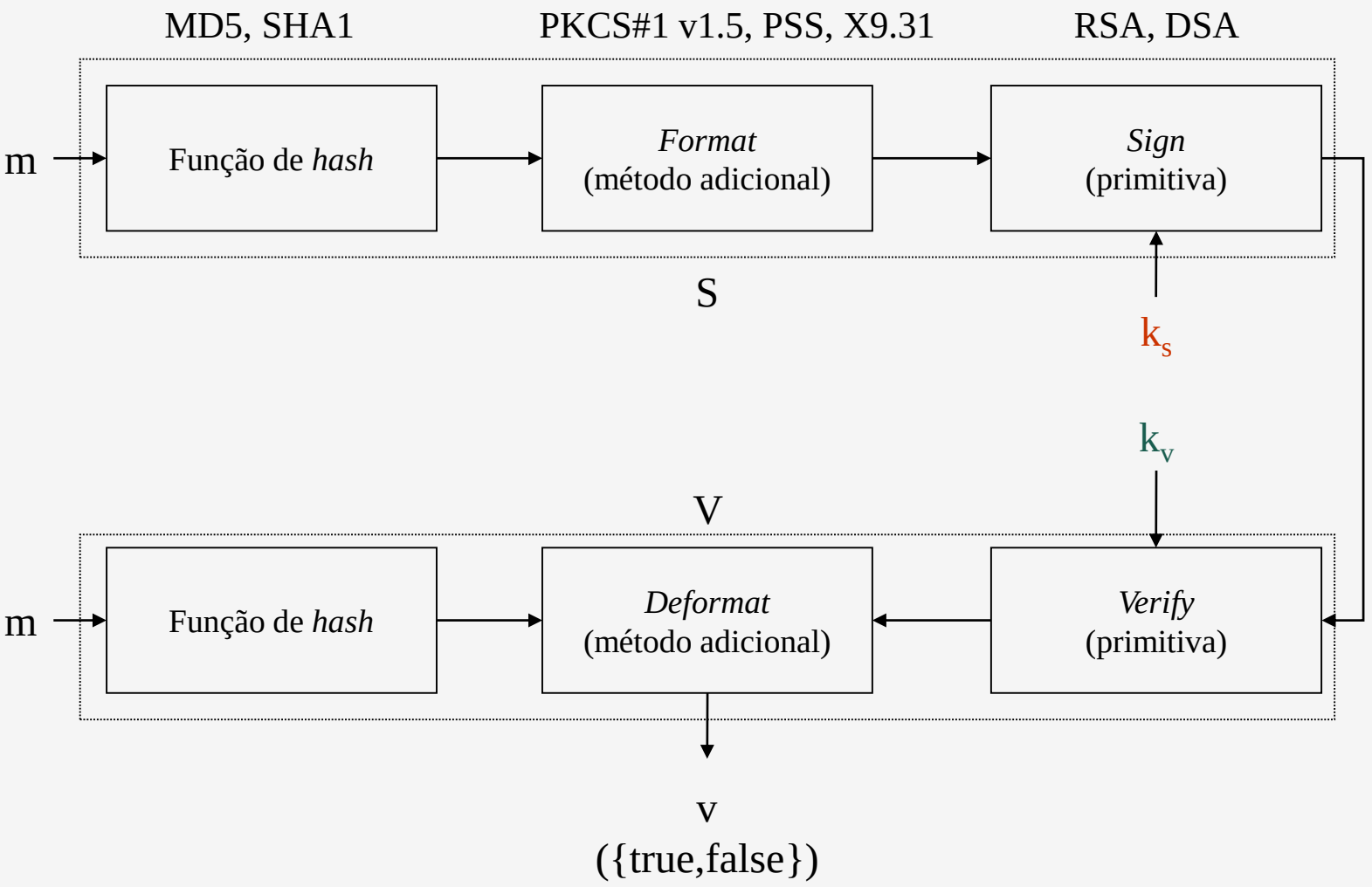
Cifra assimétrica

- 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: arquitectura 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