

# Esquemas Simétricos

- Esquemas de confidencialidade e autenticidade
- Primitivas de cifra simétrica e modos de operação

# Sumário

- Hierarquia de mecanismos criptográficos
- Esquemas criptográficos
- Esquemas simétricos de cifra
- Primitivas de cifra e modos de operação
- Esquemas MAC

# Proteção criptográfica de dados

Situação	Ameaça	Ação
Dados em repouso: <ul style="list-style-type: none"><li>• No dispositivo do utilizador;</li><li>• Na rede interna;</li><li>• Na <i>cloud</i>.</li></ul>	Processo malicioso ou não autorizado pode ler ou modificar dados	Cifra e autenticação de ficheiros/discos e comunicações
Dados estão a ser transferidos entre computadores (ex: <i>browser</i> <-> servidor)	Um atacante com acesso à rede pode ler ou modificar os dados	

on the fly

# 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

# Introdução à criptografia computacional

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

	<b>Simétrico</b>	<b>Assimétrico</b>
<b>Confidencialidade</b>	Cifra simétrica	Cifra assimétrica
<b>Autenticidade</b>	MAC	Assinatura Digital

# Introdução à criptografia computacional

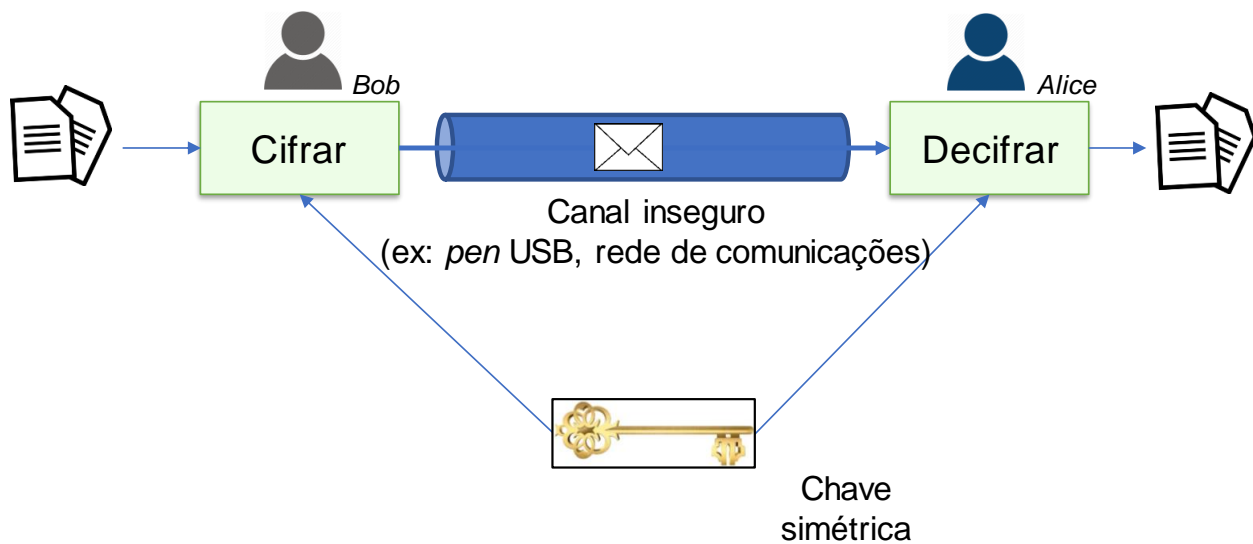
- Esquemas simétricos
  - Cifra e autenticidade
- Esquemas assimétricos
  - Cifra e assinatura digital

	<b>Simétrico</b>	<b>Assimétrico</b>
<b>Confidencialidade</b>	Cifra simétrica	Cifra assimétrica
<b>Autenticidade</b>	MAC	Assinatura Digital

## Características gerais da criptografia simétrica

- Processo de *proteção* e *desproteção* usando a mesma chave
- Chaves são normalmente usadas durante pouco tempo
- Chaves estabelecidas após um processo de negociação entre quem cifra e quem decifra

# Cifra simétrica





# Esquema de cifra simétrica

- Esquema de cifra simétrica – algoritmos (**G,E,D**)

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

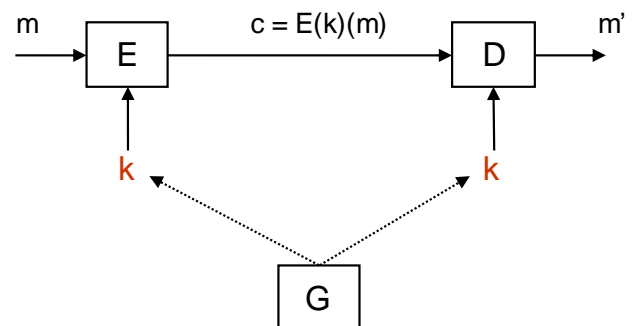
**G**:  $\rightarrow \text{Keys}$

- E** – função (probabilística) de cifra

**E**:  $\text{Keys} \rightarrow \{0,1\}^* \rightarrow \{0,1\}^*$

- D** – função (determinística) de decifra

**D**:  $\text{Keys} \rightarrow \{0,1\}^* \rightarrow \{0,1\}^*$



# Propriedades da cifra simétrica

- Propriedade da correção:
  - $\forall m \in \{0,1\}^*, \forall k \in \mathbf{Keys}: D(k)(E(k)(m)) = m$ 
    - **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

# Primitivas de cifra simétrica

- Para usar um esquema de cifra simétrica é preciso escolher uma primitiva de cifra;
  - AES, DES, Blowfish, ...
- Algumas primitivas estão especificadas em *standards* internacionais ou em publicações académicas
  - <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf> (AES)
  - <https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf> (DES, *deprecated*)
  - [https://www.schneier.com/academic/archives/1994/09/description\\_of\\_a\\_new.html](https://www.schneier.com/academic/archives/1994/09/description_of_a_new.html)
- Existem várias implementações disponíveis
- Boas práticas:
  - Usar algoritmos seguros públicos
  - Usar implementações de *confiança*

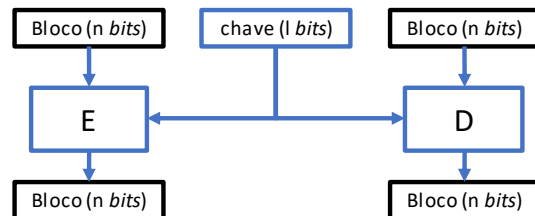
ver [letsencrypt.org](https://letsencrypt.org)  
[certbot.eff.org](https://certbot.eff.org)

# Primitivas de cifra simétrica

- Primitiva de cifra em bloco:

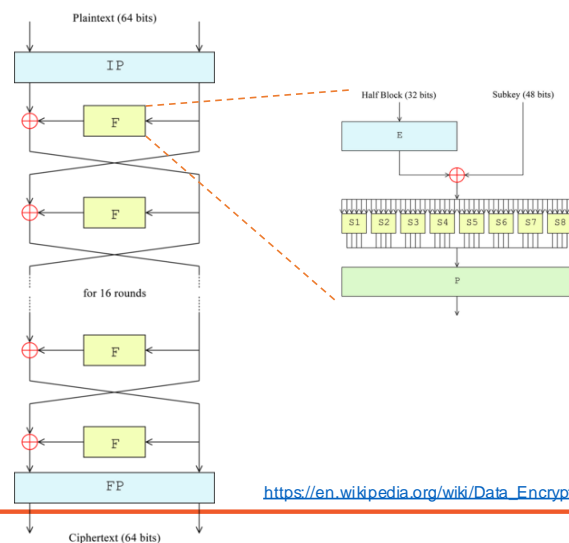
- Função  $E: \{0,1\}^l \rightarrow \{0,1\}^n \rightarrow \{0,1\}^n$   
tal que  $\forall k \in \{0,1\}^l$  a função  $E(k)$  é uma permutação
- Designa-se por  $D: \{0,1\}^l \rightarrow \{0,1\}^n \rightarrow \{0,1\}^n$  a função que verifica  
 $\forall k \in \{0,1\}^l$  e  $\forall m \in \{0,1\}^n: D(k)(E(k)(m)) = m$

- A dimensão do bloco é  $n$  (ex. 64 bit, 128 bit)
- A dimensão da chave é  $l$  (ex. 56 bit, 128 bit, 256 bit)



## Características gerais das primitivas simétricas

- A dimensão **n** do bloco deve ser suficientemente elevada para impossibilitar ataques baseados na estatística do texto em claro
- A dimensão da chave **l** deve ser suficientemente elevada para impossibilitar ataques de pesquisa exaustiva
- Elementos construtores
  - Substituições
  - Transposições

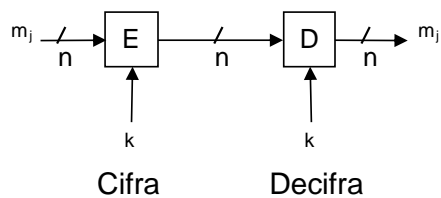


[https://en.wikipedia.org/wiki/Data\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Data_Encryption_Standard)

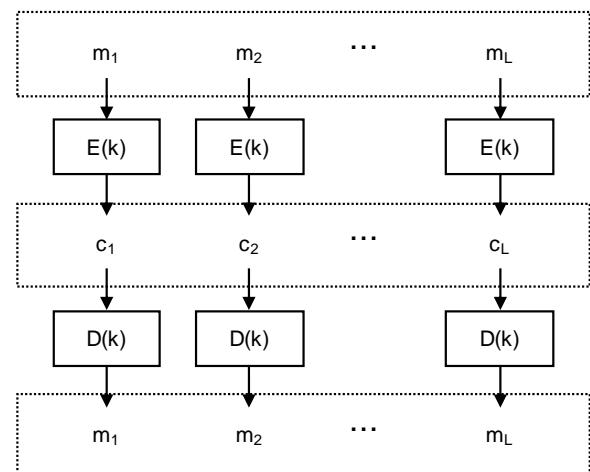
# Modos de operação

- Problema: Como efectuar a cifra de mensagens com dimensão superior à de um bloco?
- Considerações:
  - Padrões no texto em claro não deverão ser evidentes no texto cifrado
  - A eficiência do método usado não deverá ser muito inferior à eficiência da primitiva de cifra em bloco usada
  - A dimensão do texto cifrado deve ser aproximadamente igual à dimensão do texto em claro
  - Em algumas aplicações é importante que a decifra seja capaz de recuperar de erros, adições e remoções de *bits* ocorridos no texto cifrado
  - Acesso aleatório – capacidade de decifrar e alterar apenas parte do criptograma

# Modo Electronic-Codebook (ECB)



- A primitiva garante que os padrões do **bloco** em claro não passam para os **bloco** cifrado
- E se o bloco se repetir na mensagem?



## Modo *electronic-codebook* (ECB)

- Blocos de texto em claro iguais:
  - Blocos de texto em claro iguais, cifrados com a mesma chave, implicam blocos de texto cifrado iguais
- Interdependência na cifra:
  - A cifra é realizada de forma independente de bloco para bloco
- Propagação de erros: *evita*
  - A ocorrência de erros num bloco de texto cifrado afecta apenas a decifra desse bloco
- Acesso aleatório:
  - Permite acesso aleatório para decifra e “recifra” de múltiplos de blocos.



# Efeito dos modos de operação



Imagem original

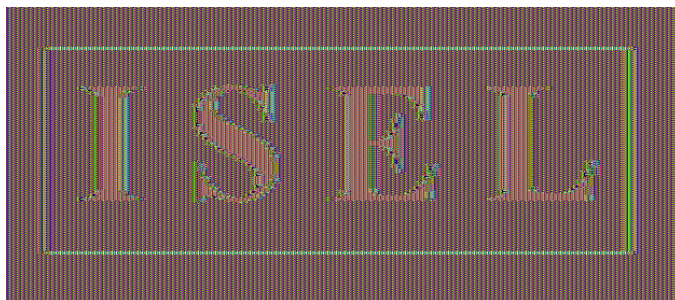


Imagem cifrada com **DES** em modo **ECB**

# Efeito dos modos de operação



Imagem original

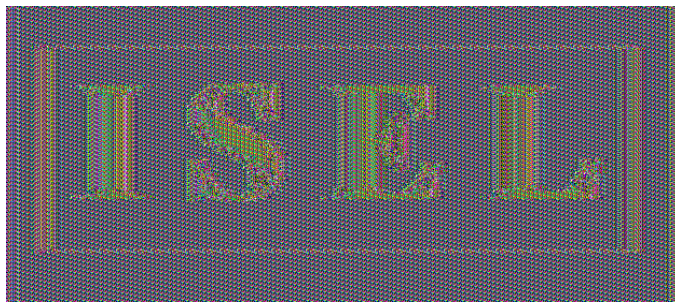


Imagem cifrada com **AES** em modo **ECB**

# Efeito dos modos de operação

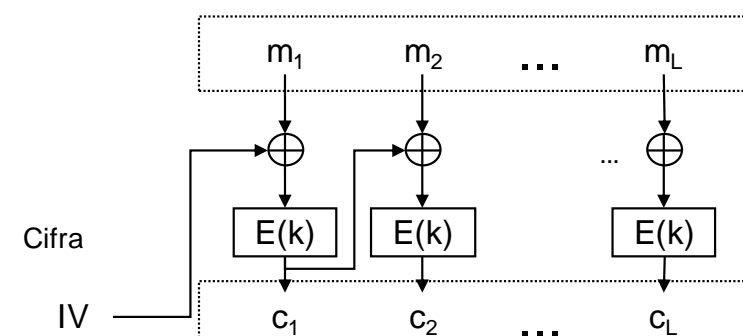


Imagem original



DES+**CBC** ou AES+**CBC**

# Modo *cipher block chaining* (CBC)



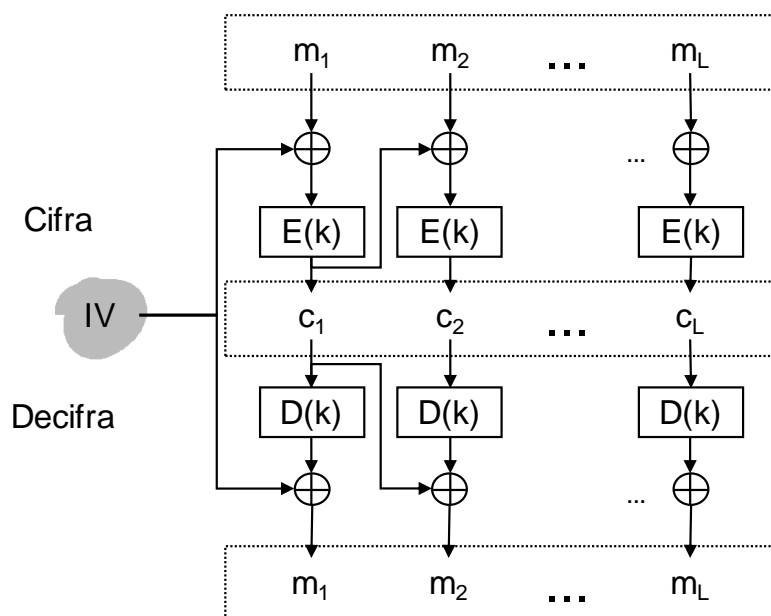
Decifra

?

# Modo *cipher block chaining* (CBC)

- Blocos de texto em claro iguais:
  - Sob a mesma chave e sob o mesmo vector de iniciação, duas mensagens iguais implicam criptogramas iguais
- Interdependência na cifra:
  - A cifra de um bloco de texto em claro afecta a cifra dos blocos seguintes
- Propagação e recuperação de erros:
  - A ocorrência de erros num bloco  $c_j$  de texto cifrado afecta a decifra do próprio bloco e a do bloco seguinte  $c_{j+1}$ . A decifra do bloco  $c_{j+1}$  terá erros nas mesmas posições que  $c_j$
- Observações:
  - A reordenação dos blocos de texto cifrado afecta a decifra
  - É relativamente fácil manipular um determinado bloco de texto em claro

# Modo cipher block chaining (CBC)



# Boas práticas sobre o IV

- Deve ser armazenado/transmitido em claro;
- Não se deve repetir (*uniqueness*);
- Não deve ser previsível;

# Padding

- Seja  $X$  o número de bytes a acrescentar para que a dimensão da mensagem seja múltipla da dimensão do bloco
- Ex: PKCS# 5 (CBC-PAD):
  - Acrescentar  $X$  bytes com o valor  $X$
  - Utilizações PKCS# 7, CMS, SSL
- A segurança do esquema depende da forma de *padding*?
- Ataque proposto por S. Vaudenay: *chosen ciphertext attack* utilizando o destinatário como *oráculo* que recebe criptogramas e retorna 1 ou 0 conforme o *padding* esteja correto ou não
  - <https://www.iacr.org/cryptodb/archive/2002/EUROCRYPT/2850/2850.pdf>



[www.openssl.org/docs](http://www.openssl.org/docs)

[www.schneier.com](http://www.schneier.com)

- Para fazer update obs CA em Ubuntu:  
update-ca-certificates

## Demonstração com OpenSSL

Cifra e MAC



openssl enc -aes-256-cbc -salt -in text.txt  
-out encrypted256cbc.bin -iter 10000

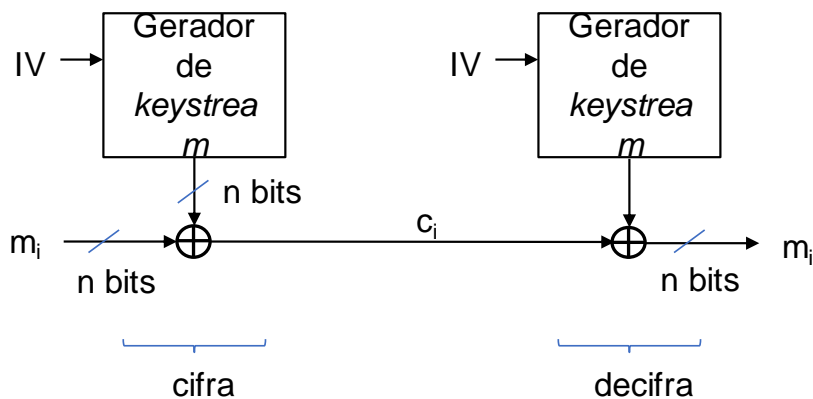
→ automatic salt

openssl enc -aes-256-cbc -d -in encrypted256cbc.bin  
-out decrypted-aes-256-cbc.txt -iter 10000

-d -decrypt

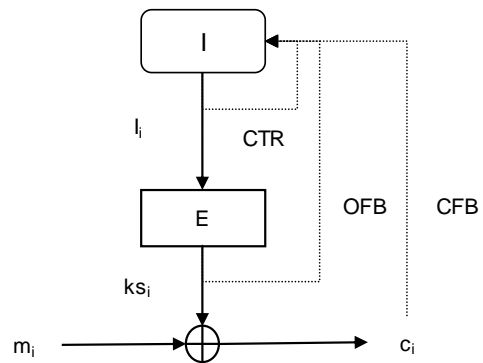
# Modos de operação em *stream*

- Modos como o CBC precisam de algoritmos diferentes para cifra e decifra;
- Modo de operação em *stream*:

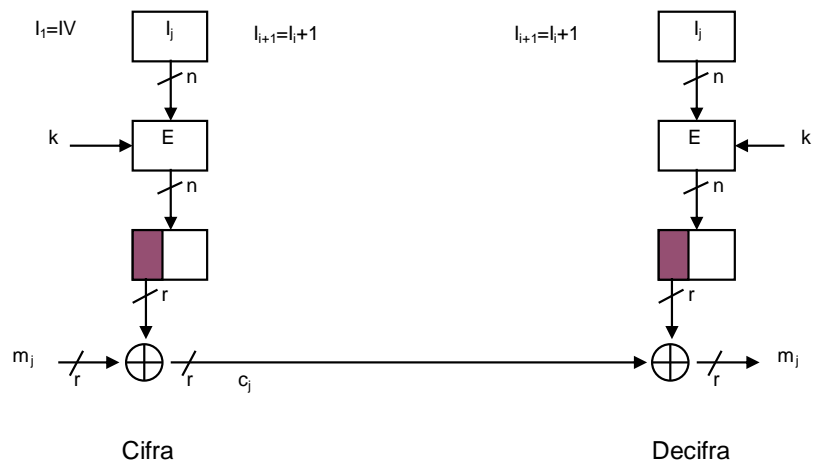


# Modos de operação em *stream*

- Modo *Stream*:
  - Estado  $I$
  - *Key stream*  $ks$
  - $ks_i = E(k)(I_i)$
  - $c_i = m_i \oplus ks_i$
  - *Cipher FeedBack* (CFB):
    - $I_i = c_{i-1}$
- *Output FeedBack* (OFB):
  - $I_i = ks_{i-1}$
- *Counter* (CTR):
  - $I_i = f(I_{i-1})$
- Problema:
  - se  $ks_i = ks_j$  então  $m_i \oplus m_j = c_i \oplus c_j$



# Modo *Counter* (CTR)



# Modo *Counter* (CTR)

- Blocos de texto em claro iguais:
  - Sob a mesma chave e sob o mesmo vector de iniciação, duas mensagens iguais implicam criptogramas iguais
- Propagação e recuperação de erros:
  - A ocorrência de erros num bloco de texto cifrado  $c_j$  afecta apenas a decifra desse bloco. O bloco  $m_j$  resultante da decifra do bloco  $c_j$  terá erros nas mesmas posições que  $c_j$
- Acesso aleatório:
  - Permite acesso aleatório para decifra e “recifra” de *bits*
- Observações:
  - É relativamente fácil manipular um determinado bloco de texto em claro

Ir ao Moodle e fazer um grupo de 2 ou 3 numa folha Excel

Buscar o Livro **Real-World Cryptography** - David Wong de 2021

## Introdução à criptografia computacional

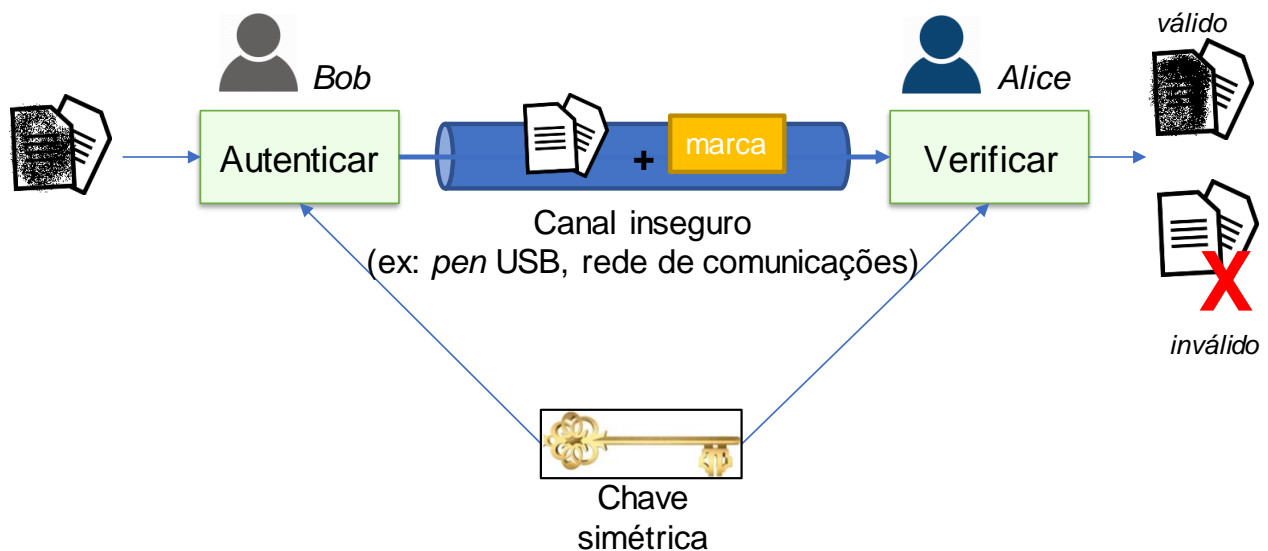
- Esquemas simétricos
  - Cifra e autenticidade
- Esquemas assimétricos
  - Cifra e assinatura digital

	Simétrico	Assimétrico
Confidencialidade	Cifra simétrica	Cifra assimétrica
Autenticidade	MAC	Assinatura Digital

e não repúdio  
Message Authentication Codes

- Garantir que a pessoa que enviou a mensagem tem o conhecimento da chave
- Prender uma marca ao conhecimento da chave que é secreta

## Autenticação de mensagens



# Esquema MAC

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

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

$G: \rightarrow \text{Keys}$

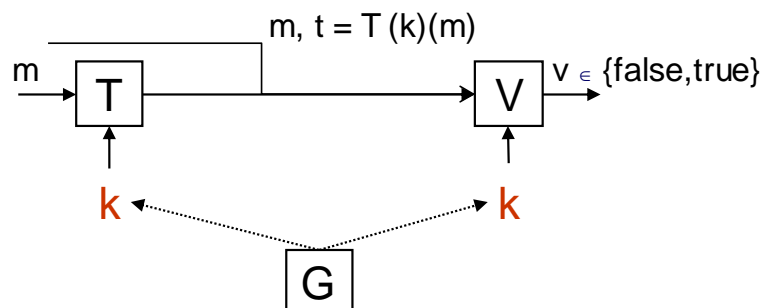
- $T$  – função (probabilística) de geração de marcas

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

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

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

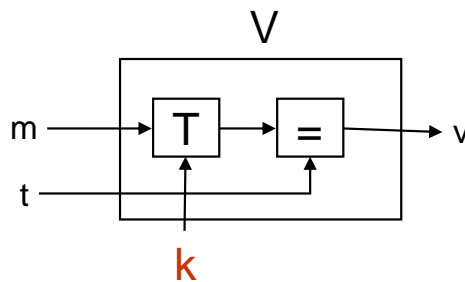
*(tags) e adicionados ao final da mensagem.*





## Esquema MAC (2) - forma para a verificação

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



volta-se a aplicar a função de  $T$  e verifica-se se o resultado é igual ao recebido.  
Por exemplo, o cálculo do checksum

- (1) Produzir uma mensagem de tal maneira que dê a mesma TAG
- (2) Criar mensagem e tag

## Propriedades do MAC

- Propriedade da correcção
  - $\forall m \in \{0,1\}^*, \forall k \in \text{Keys}: V(k)(T(k)(m), m) = \text{true}$
- Propriedades de segurança
  - Sem o conhecimento de  $k$ , é *computacionalmente infazível*
    - falsificação selectiva – dado  $m$ , encontrar  $t$  tal que  $V(k)(t, m) = \text{true}$  (1)
    - falsificação existencial – encontrar o par  $(m, t)$  tal que  $V(k)(t, m) = \text{true}$  (2)
- 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
  - Por exemplo: 160, 256, 512 *bits*
- Códigos detectores e correctores de erros não servem para esquemas de MAC
- Exemplos: HMAC-SHA1

ISEL - Não existe a propriedade de 'não-repúdio', não é possível saber se quem criou a tag foi o 'Bob' ou o 'Alice'.

- Funções de Hash, podem ser associadas à mensagem

- A TAG deve ter uma dimensão pequena e ter poucas colisões.

→ têm 2 chaves diferentes:  
1 p/ msg, outra para a TAG

## Cifra autenticada

- Para garantir confidencialidade e simultaneamente autenticidade, tem de se usar uma combinação dos esquemas Cifra e MAC
- Existem duas abordagens
  - *Encrypt-then-MAC* → *Permite não perder tempo a desencryptar mensagens inválidas, mas não garante nada sobre a msg original*
    - $E(k_1)(M) || T(k_2)(E(k_1)(M))$
    - A marca indica se houve alteração ou não do criptograma
  - *MAC-then-encrypt*
    - $E(k_1)(M || T(k_2)(M))$
    - A marca é gerada sobre a mensagem, e é posteriormente tudo cifrado
- Existem modos de operação cujo objectivo é produzirem uma *cifra autenticada*, combinando as operações num só algoritmo
  - *Galois Counter Mode (GCM)* → *está no trabalho*
  - Offset codebook mode (OCB)
  - Counter with CBC-MAC (CCM)



ISEL

Existe um 3º esquema, usado no SSH

Encrypt and MAC

GCM - modo de operação em fluxo, que adiciona integridade e calcula a TAG que é metida no fim

Baseado em Galois Fields (Math)

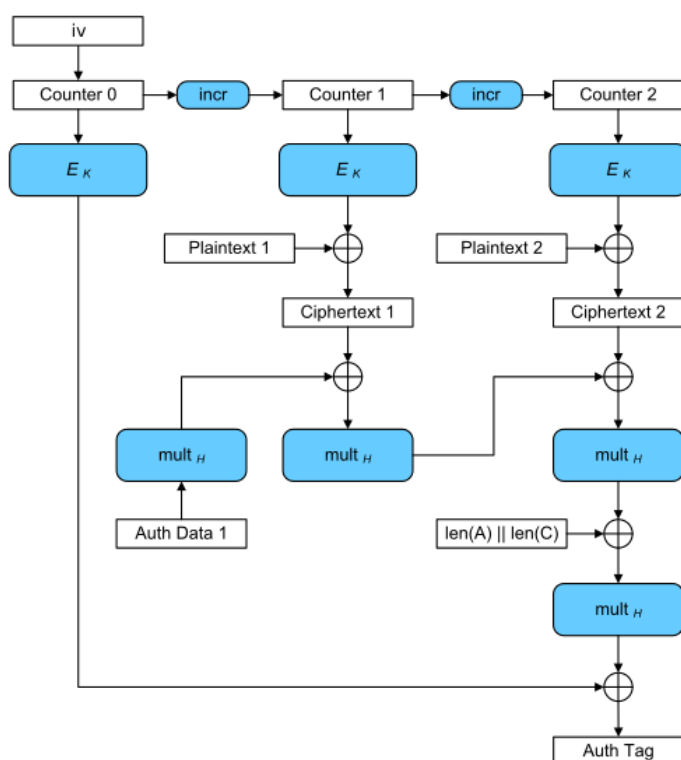
Está baseado em funções das GAs

Usa-se em WPA 3, HTTPS v1.3

AAD - Associated Authenticated Data - permite confirmar que as dados estão autenticados e íntegros

Standard do WIST diz que não há IV só counter, mas alguns criptólogos discordam

## Cifra autenticada - Exemplo



Galois Counter Mode

*Ver*

Adaptado de [https://en.wikipedia.org/wiki/Galois/Counter\\_Mode](https://en.wikipedia.org/wiki/Galois/Counter_Mode)

A primeira parte (bits menor peso) do IV são aleatórios, a segunda parte (bits menor peso) começa sempre a zero e vai aumentando.

- o bloco  $\emptyset$  não é usado para cifrar o texto, só o tag.