Autenticação de Mensagens

Redes e Segurança Informática

2º semestre > 1º ano

Mário Pinto | mjp@ua.pt

Cofinanciado por:
NORTE2020 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

Index

Autenticidade

- A ocultação do conteúdo de uma mensagem é importante
- Também é importante garantir a **autenticidade** da mensagem
  - Não foi adulterada
  - O emissor é válido
- É possível utilizar três aspetos a este nível:
  - Códigos e funções de Hash para autenticação de mensagens
  - Encriptação com chave pública
  - Assinatura digital com chaves públicas

Autenticação de Mensagens

- Encriptação simples permite ocultar o conteúdo
  - Evita os ataques passivos
- E os ataques ativos?
  - Obtendo uma **chave** e **algoritmo** é possível:
    - **Adulterar mensagens** fidedignas
    - Enviar **mensagens falsas**
- Soluções?
  - Autenticação de mensagens

Autenticação de Mensagens

- Uma **mensagem** (ficheiro, documento, etc) é **Autêntica** se:
  - É **genuína** (não foi adulterada)
  - É enviada pelo **verdadeiro emissor**
- **Autenticação de Mensagens**
  - Procedimento que **permite às partes** em comunicação **verificar que as mensagens** recebidas são **autênticas**
    - Conteúdo não foi adulterado
    - Fonte é autêntica
  - Pode ter encriptação ou não

Autenticação de Mensagens

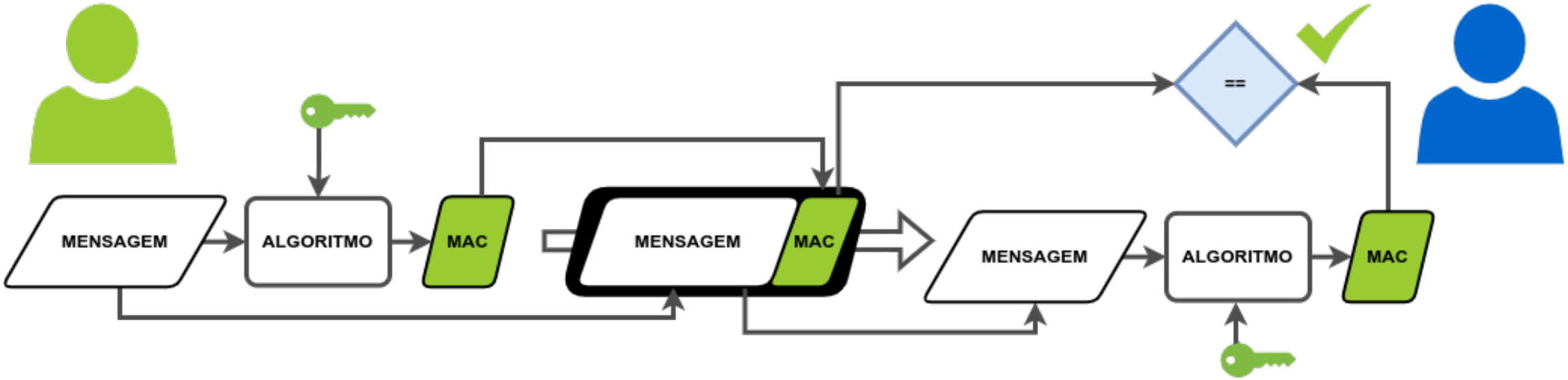
- A **Autenticação de Mensagens** pode ocorrer **sem encriptação**
  - Não garante a confidencialidade
- Pode ser preferível
  - A mesma mensagem é enviada para vários destinatários (ex. serviços de rede)
  - Não existe capacidade de encriptação (ex. grandes volumes de informação)
  - Comunicações no mesmo equipamento (ex. dados de processamento)

Message Authentication Code

- MAC | Message Authentication Code
  - Utiliza uma **chave secreta** do conhecimento de **ambas as partes**
  - **Gera** um pequeno **bloco de dados**
    - Designado de MAC – Message Authentication Code
  - **[Mensagem + MAC]** são enviados
  - Destinatário obtém **[Mensagem + MAC]**
  - **Gera o MAC** da mensagem **e compara com o MAC recebido**

Message Authentication Code

- Garante que a mensagem não foi adulterada
  - Se for alterada a mensagem não se pode alterar o MAC pois não se sabe a chave
- O emissor é válido
  - Como mais ninguém sabe a chave teve que ser ele a gerar o MAC válido



Message Authentication Code

- Para gerar o MAC podem ser utilizados diferentes algoritmos
  - Pode ser utilizado o algoritmo DES sobre a mensagem
  - Utilizam-se depois 16 bits ou 32 bits finais da mensagem encriptada como MAC
  - Semelhante à encriptação mas sem ser necessário desencriptar o MAC
  - Não necessita de ser reversível pois é utilizado apenas para efeitos de comparação
  - Menos vulnerável a ataques do que na encriptação

Funções de Hash

- Funções de Hash Seguras (One-Way)
  - Produzem uma **"impressão digital" de uma mensagem** (ficheiro, bloco, etc)
  - Uma função para ser utilizada em **Autenticação de Mensagens** deve:
    - Poder ser aplicada a **blocos de qualquer tamanho**
    - Produzir **hash de comprimento fixo**
    - **Computacionalmente fácil** de gerar hash
    - **Resistente a pré-imagem -** encontrar o objeto que origina uma hash
    - **Resistente a pré-imagem alternativa -** encontrar objeto diferente que origine a mesma hash
    - **Resistente a colisões -** encontrar pares de objetos com a mesma hash

Funções de Hash | Segurança

O ataque a funções de Hash pode ser feito por:

- Criptoanálise
  - Tenta perceber fraquezas na lógica do algoritmo
- Força Bruta
  - Depende apenas do comprimento da hash gerada
  - Para uma hash de **n** bits o nível de esforço é proporcional ao comprimento:
    - **Resistente a pré-imagem:** 2^n
    - **Resistente a pré-imagem alternativa:** 2^n
    - **Resistente a colisões:** $2^{n/2}$

Funções de Hash Simples

Os princípios de funcionamento são semelhantes entre todas as funções de hash

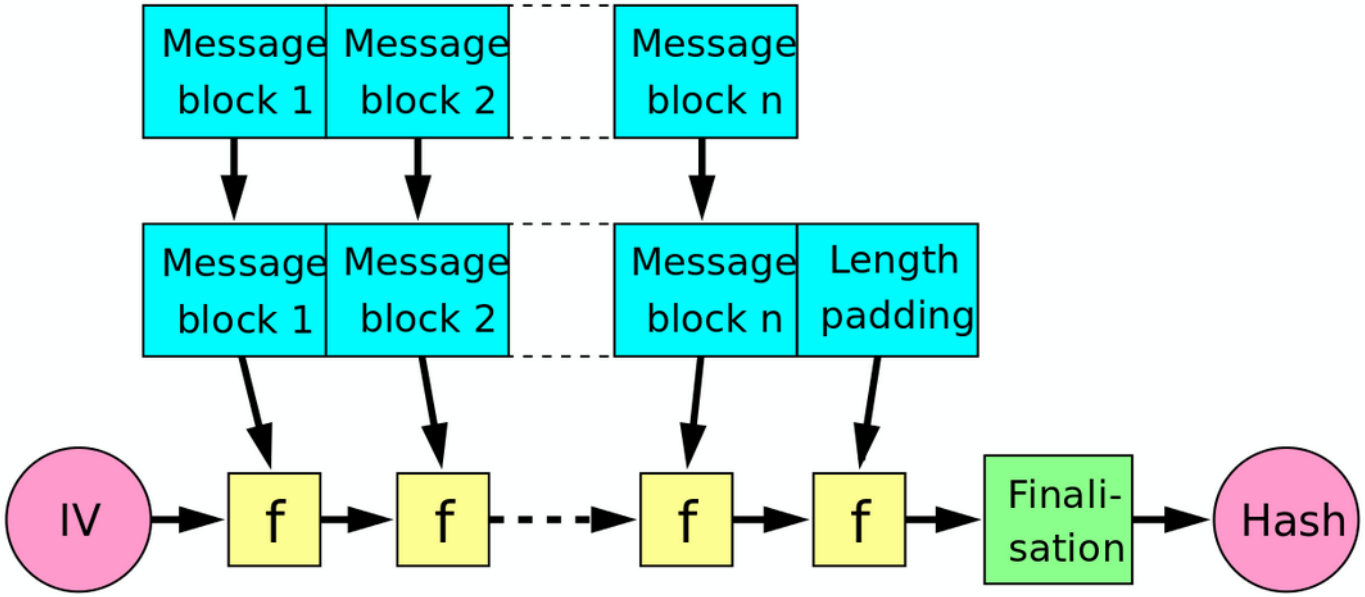
- Input é processado por blocos de **n** bits
- Hash produzida tem **n** bits de comprimento

	Bit 1	Bit 2	...	Bit n
Bloco 1	b_{11}	b_{21}	...	b_{n1}
Bloco 2	b_{12}	b_{21}	...	b_{n2}
...	...	...	...	...
Bloco m	b_{1m}	b_{2m}	...	b_{nm}
Hash	C_1	C_2	...	C_n

Tabela para explicação da aplicação de uma função de hash simples: XOR entre todos os bits de cada posição dos blocos.
Será um bit de paridade para cada posição.

Construção Merkle–Damgård

- Método para construção de funções de hash
  - One-way
  - Resistentes a Colisões
- Descrito por Ralph Merkle e Ivan Damgård
  - Define vetor de inicialização
  - Cria blocos de comprimento fixo
  - Aplica a função f (IV e bloco)
  - O último bloco é preenchido



Esquema: https://en.wikipedia.org/wiki/Merkle%E2%80%93Damg%C3%A5rd_construction#/media/File:Merkle-Damgard_hash_big.svg

Message Digest

MD2 | Message Digest version 2

The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input.

- Desenhada por Ronald Rivest em 1989 e publicada na RFC1319 (1992)
- Utiliza o método de Merkle–Damgård
- Gera hash de 128 bits (16 bytes)
  - Atualmente tem pouca resistência a Colisões
  - Permite obter pré-imagem e pré-imagem alternativa
  - Pode revelar as chaves (secretas) utilizadas
  - Não deve ser utilizado em assinaturas digitais

RFC1319 | https://www.rfc-editor.org/rfc/rfc1319

Message Digest

MD5 | Message Digest version 5

- Função de hash muito conhecida e utilizada
- Desenhada por Ronald Rivest em 1991 e publicada na RFC1321 (1992)
- Gera hash de 128 bits (16 bytes)
  - Atualmente muito insegura
  - Colisões são muito rapidamente calculadas
  - Ainda é utilizada...

RFC 1321 | https://www.rfc-editor.org/rfc/rfc1321

Secure Hash Standard

SHA | Secure Hash Algorithm

*This standard specifies a Secure Hash Algorithm (SHA) which can be used to **generate a condensed representation of a message** called a **message digest**.*

- Desenvolvida pelo NIST e publicada no FIPS 180 (1993)
- Para utilização com o **Digital Signature Algorithm** (DSA)
- Utilizada pelo emissor e recetor para criar e verificar uma assinatura digital

FIPS 180 | https://csrc.nist.gov/publications/detail/fips/180/archive/1993-05-11

Secure Hash Standard

- SHA-0
  - Desenvolvida pelo NIST e publicada no FIPS 180 (1993)
  - SHA-0 conhecida por Secure Hash Algorithm
- SHA-1
  - Versão atualizada e publicada no FIPS 180-1 (1995)
  - Gera hash com 160 bits
- SHA-2
  - Versão publicada no FIPS 180-2 (2002) e atualizada FIPS 180-3 (2008), FIPS 180-4 (2015)
  - **SHA-256**, **SHA-384** e **SHA-512** respetivamente com hashes de 256, 384 e 512 bits
- SHA-3
  - Lançado concurso em 2007
  - Ainda não substituiu a SHA-2

FIPS 180 | https://csrc.nist.gov/publications/detail/fips/180/archive/1993-05-11
FIPS 180-1 | https://csrc.nist.gov/publications/detail/fips/180/1/archive/1995-04-17
FIPS 180-2 | https://csrc.nist.gov/publications/detail/fips/180/2/archive/2002-08-01
FIPS 180-3 | https://csrc.nist.gov/publications/detail/fips/180/3/archive/2008-10-31
FIPS 180-4 | https://csrc.nist.gov/publications/detail/fips/180/4/final

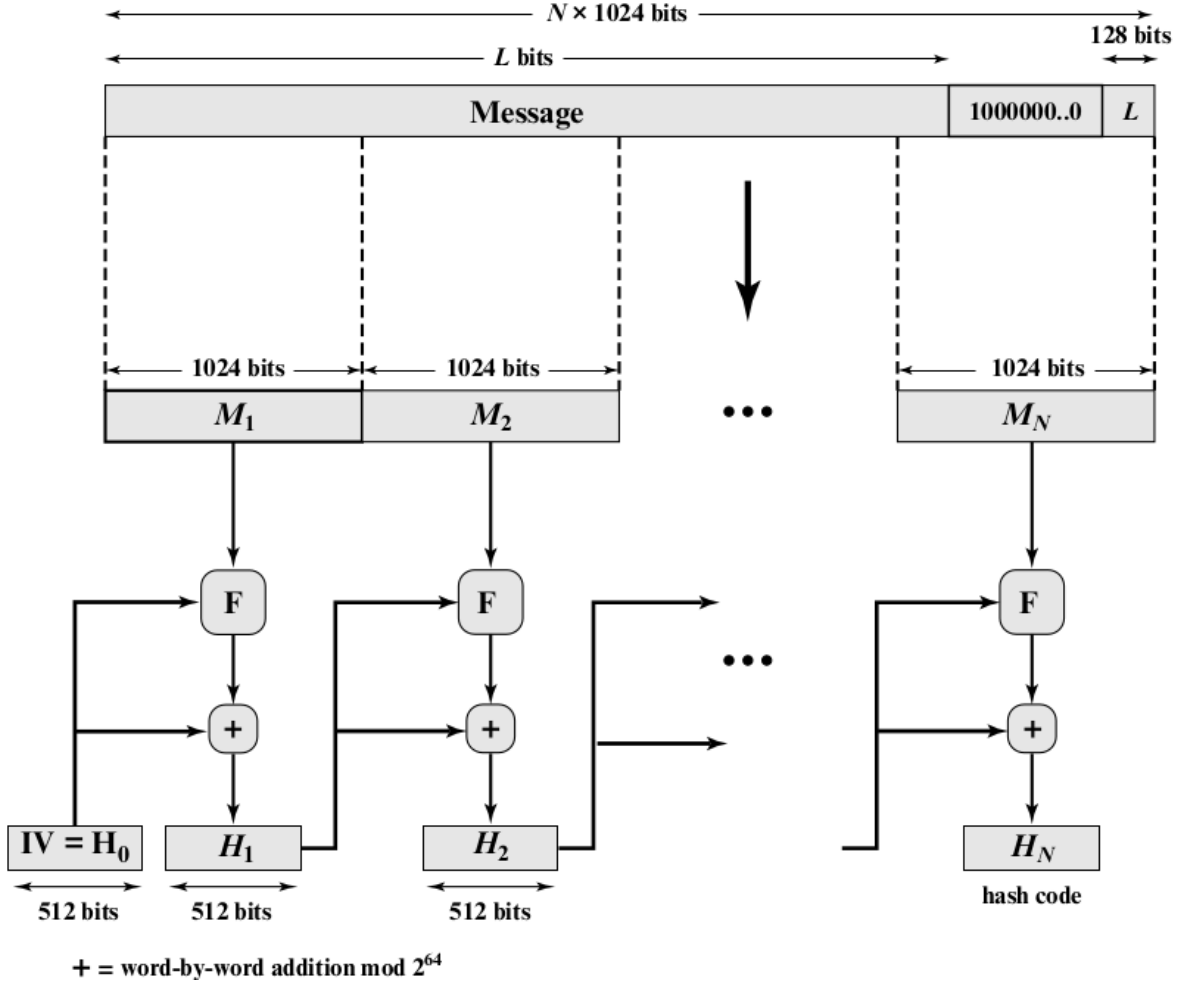
Secure Hash Standard

SHA512 | Visão Geral do Algoritmo:

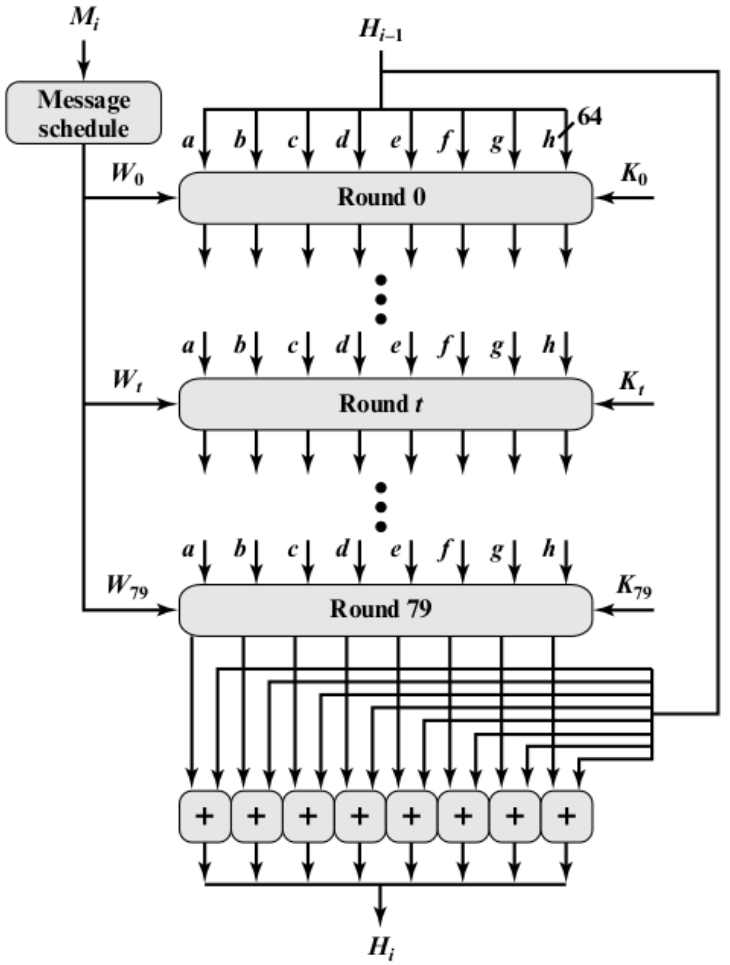
- Preenchimento com bits
  - Até ficar com último bloco com 896 bits
- Acrescentados 128 bits no final da mensagem
  - com o comprimento da mensagem inicial
  - Blocos de 1024 bits
- Inicializado o buffer de hash de 512 bits
  - 8 registos de 64 bits
- Processamento dos blocos 1024 bits
  - 80 etapas
- Resultado
  - Hash com 512 bits relativa ao último buffer

Secure Hash Standard

SHA512 | Esquema Geral do Algoritmo



SHA512 | Processamento de um Bloco 1024 bits



William Stallings: Network Security Essentials Applications and Standards, 6th Edition

Encriptação e Funções de Hash

- Houve interesse em utilizar funções de Hash Seguras em Autenticação de Mensagens
  - As funções de Hash Seguras são mais rápidas do que os algoritmos de encriptação
  - Existem mais implementações de funções de Hash Seguras
  - Assim, é mais vantajoso desenvolver soluções com funções de Hash Seguras
- SHA-1 não previa a utilização de chaves
  - Desenvolvidas adaptações para permitir a utilização de chave secreta
  - Com chave secreta pode ser utilizada para Autenticação de Mensagens

HMAC

HMAC: Keyed-Hashing for Message Authentication

*A mechanism for **message authentication using cryptographic hash functions**. HMAC can be used with any iterative cryptographic hash function, e.g., MD5, SHA-1, in combination with a secret shared key.*

- Definido em 1997 pela RFC2104
- Destinado a ser utilizado em comunicações seguras
  - Protocolos IP
  - Transport Layer Secutiry (TLS)
  - Secure Eletronic Transaction (SET)

RFC2104 | https://www.rfc-editor.org/rfc/rfc2104

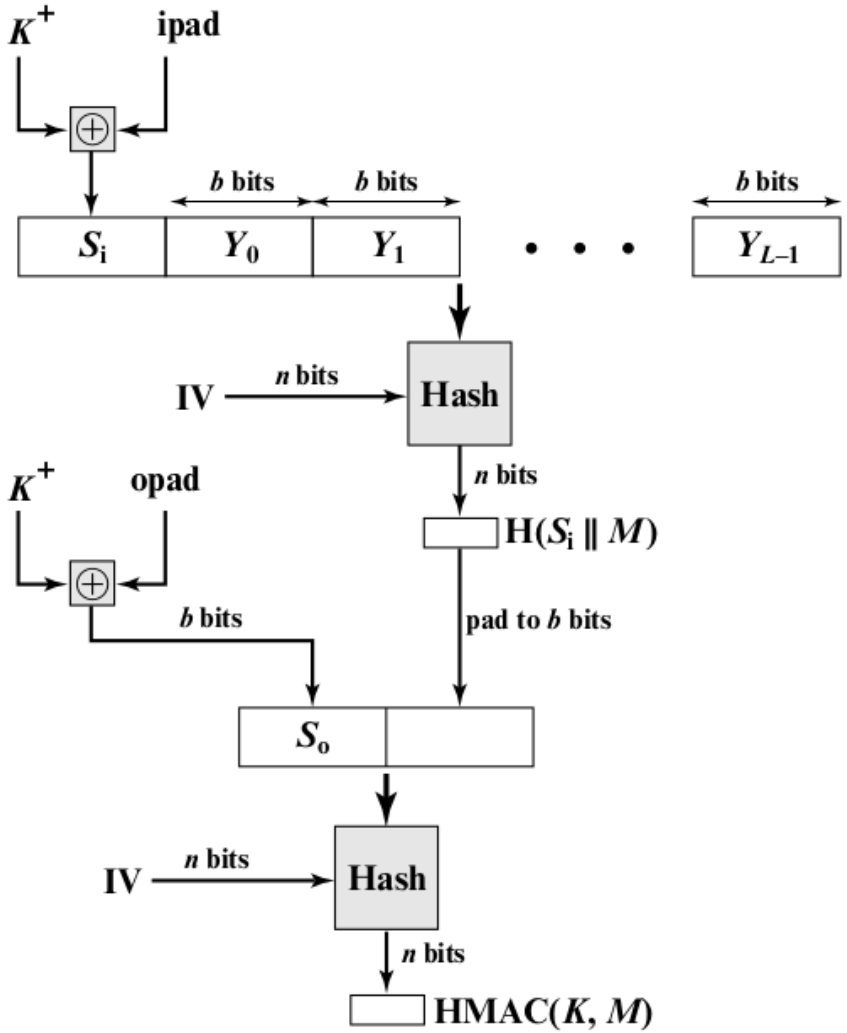
HMAC

- Especificação
  - Utilização, sem modificação, de funções de hash disponíveis (em particular as gratuitas e facilmente acessíveis)
  - Permitir a substituição da função de hash utilizada (para melhorar a segurança ou a velocidade, conforme a necessidade)
  - Preservar o desempenho da função de hash e não provocar a sua degradação
  - Utilizar e gerir chaves de forma simples
  - Permitir uma análise criptográfica da força do mecanismo a partir da função de hash

RFC2104 | https://www.rfc-editor.org/rfc/rfc2104

HMAC | Estrutura

- b – tamanho do bloco em bits
- K^+ – K com zeros à esquerda (até ter b bits)
- ipad – 00110110 repetido b/8 vezes
- S_i – bloco obtido a partir de XOR entre ipad e K^+
- Y_i – bloco i da Mensagem M (que tem L blocos)
- Acrescentar blocos de M a S_i (menos o último)
- Aplicar função de hash H ao resultado anterior [S_i+M]
- opad – 01011100 repetido b/8 vezes
- S_0 – bloco obtido a partir de XOR entre opad e K^+
- Acrescentar a S0 o resultado da hash
- Aplicar função de hash H ao resultado anterior



William Stallings: Network Security Essentials Applications and Standards, 6th Edition

CMAC

CMAC | Cipher-Based Message Authentication Code

*This Recommendation specifies a message authentication code (MAC) algorithm based on a **symmetric key block cipher**. This block cipher-based MAC algorithm, called CMAC, may be used to provide assurance of the authenticity and, hence, the integrity of binary data.*

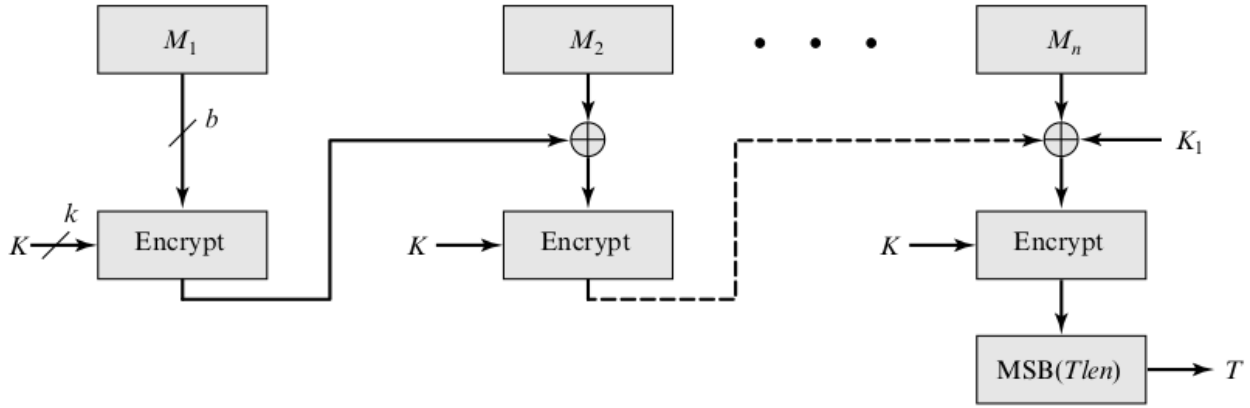
- Definido pelo SP 800-38B
- Utiliza AES ou 3DES
- Permite obter o CMAC ou ***tag*** de uma mensagem

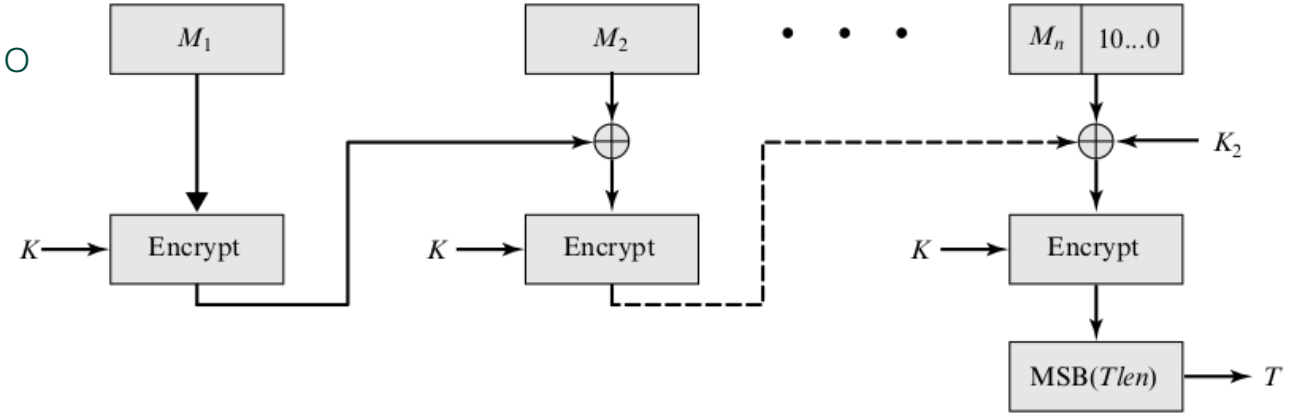
SP800-38B | https://csrc.nist.gov/publications/detail/sp/800-38b/final

CMAC

CMAC | Cipher-Based Message Authentication Code

- M tem n blocos de comprimento b bits
- K chave com k bits e K_1 sub-chave com n bits obtida aplicando a encriptação a um bloco apenas com 0, rodando o resultado para esquerda um bit e aplicando XOR a uma constante
- T é a *tag* ou o CMAC pretendido
- Quanto M não é múltiplo do comprimento do bloco é preenchido com 100...
- Funcionamento igual ao caso anterior mas utilizando outra sub-chave K_2 obtida a partir de K_1 utilizando o mesmo mecanismo





William Stallings: Network Security Essentials Applications and Standards, 6th Edition

AEAD

AEAD | Authenticated Encryption with Associated Data

- Conhecida como **Encriptação Autenticada**
- Combina **Encriptação** com **Autenticação**
- Para o seu funcionamento é necessário:
  - Mensagem encriptada
  - Nonce (ou IV)
  - *Tag* MAC

CCM

CCM | Counter with Cipher Block Chaining-Message Authentication Code

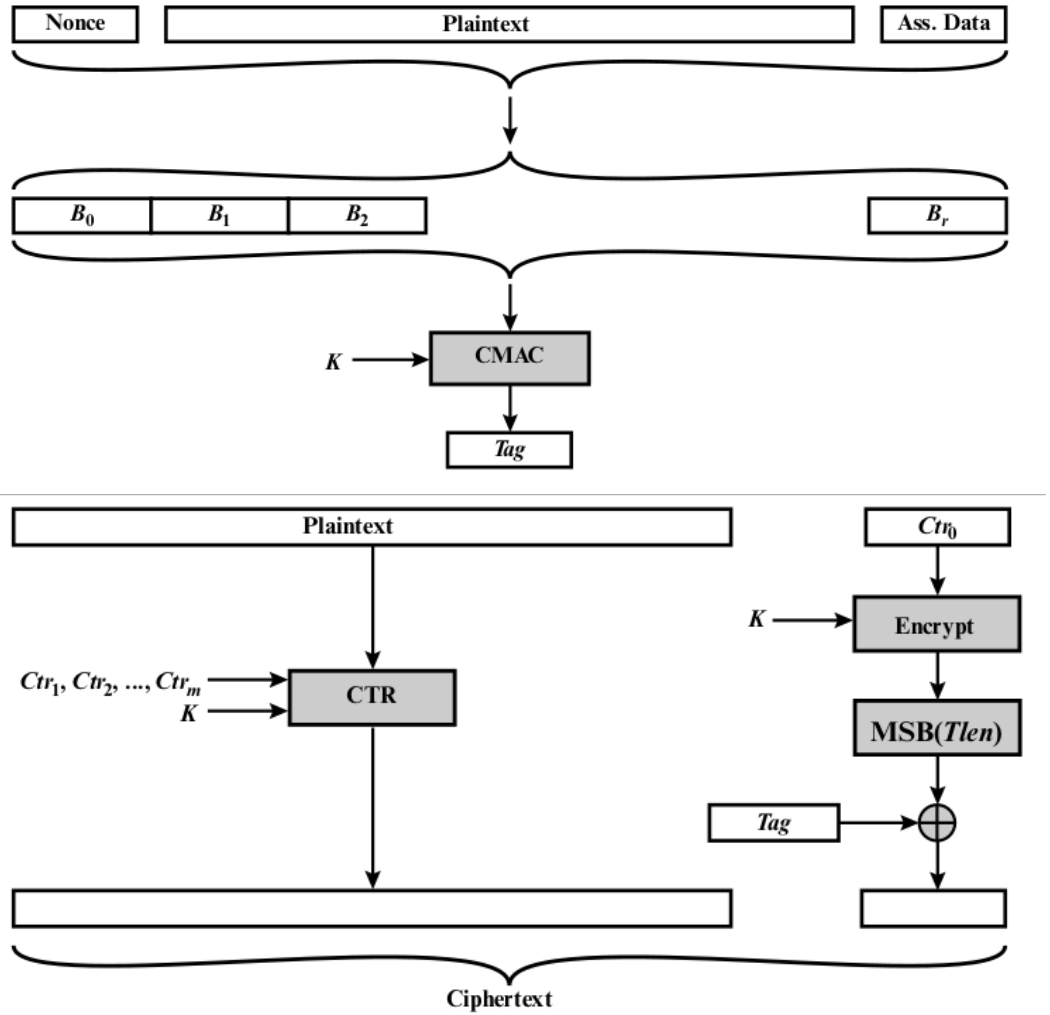
*This Recommendation defines a mode of operation, called Counter with Cipher Block Chaining-Message Authentication Code (CCM), for a **symmetric key block cipher algorithm**. CCM may be used to provide assurance of the **confidentiality and the authenticity** of computer data by combining the techniques of the Counter (CTR) mode and the Cipher Block Chaining-Message Authentication Code (CBC-MAC) algorithm.*

- Definido pelo SP 800-38C
- Conhecida como **Encriptação Autenticada**
- Autentica as mensagens com CMAC
- Encripta o conteúdo com AES-CTR
- Utiliza uma única chave para autenticar e encriptar

SP800-38C | https://csrc.nist.gov/publications/detail/sp/800-38c/final

CCM

- Autenticação
  - Utilizando Nonce e outros dados combina a mensagem inicial em blocos B_i
  - Utiliza o algoritmo CMAC e a chave K para obter a tag
- Encriptação
  - A sequência de contadores C_{tri} são gerados de independente ao Nonce
  - Utiliza o AES-CTR e a chave K para encriptar a mensagem original
  - A partir do C_{tr0} utiliza o AES-CTR para encriptar a *tag* do CMAC
  - A Mensagem final é a concatenação destes dois elementos



William Stallings: Network Security Essentials Applications and Standards, 6th Edition

GCM

GCM | Galois Counter Mode

*This Recommendation specifies the **Galois/Counter Mode (GCM)**, an algorithm for authenticated encryption with associated data, and its specialization, GMAC, for generating a message authentication code (MAC) on data that is not encrypted. GCM and GMAC are modes of operation for an underlying approved symmetric key block cipher.*

- Definido pelo SP 800-38D
- Utiliza Nonce de 96 bits (12 bytes)
- Origina MAC *tag* com 128 bits (16 bytes)

SP800-38D | https://csrc.nist.gov/publications/detail/sp/800-38d/final