

Criptografia Aplicada

Autenticação de mensagens

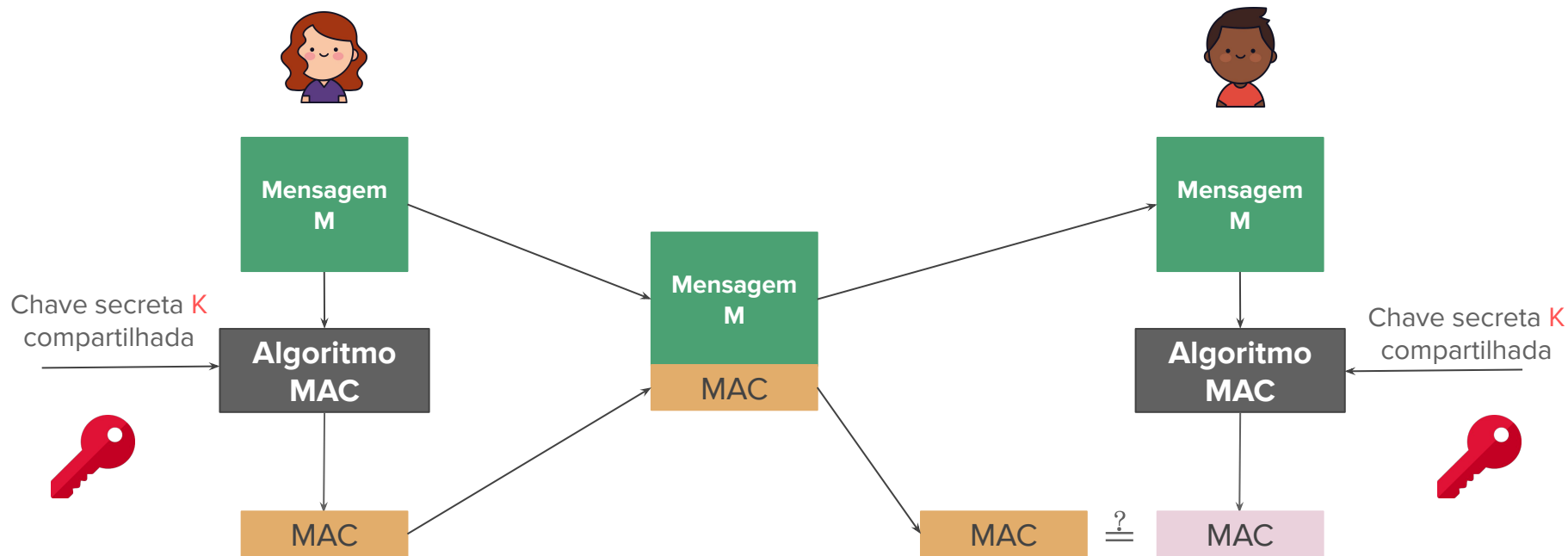
Sumário

- Definições Básicas
- Requisitos e segurança
- HMAC
- CMAC
- MAC na prática

Recapitulando..

- Cifragem de mensagens provê *confidencialidade*
 - protegendo a mensagem de um atacante passivo
- Funções de hash provêm *integridade* de dados
 - auxiliando na identificação de modificações em uma mensagem
- E a *autenticidade* dos dados?
 - garantia de que o dado é genuíno, provêm da fonte alegada e não foi alterado sem permissão
- Note que a função de hash vista na aula passada só garante a não-alteração dos dados!

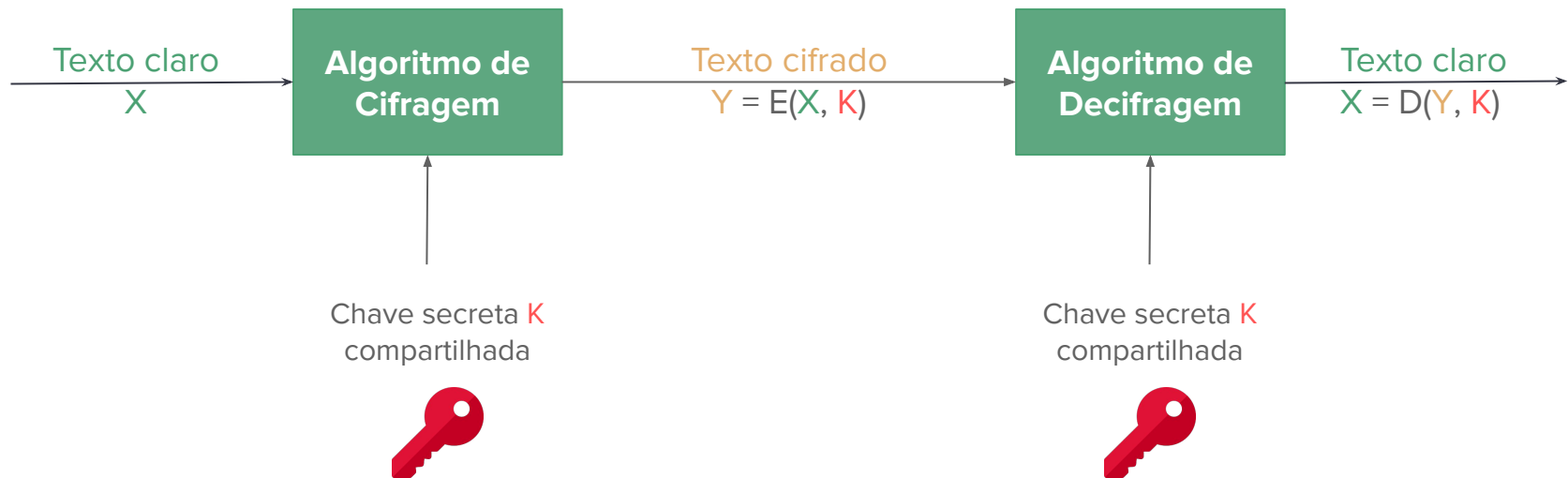
Message Authentication Code



Message Authentication Code (MAC)

- Um MAC utiliza uma chave **K** para gerar um pequeno bloco de tamanho fixo chamado **checksum, tag** ou **MAC**
 - que é anexado à mensagem e enviado ao destinatário
- Assume-se que ambas as partes comunicantes compartilham uma chave secreta **K**
 - e apenas os dois conhecem essa chave
- Se o MAC recebido é igual ao MAC calculado pelo destinatário, ele sabe que:
 - a mensagem não foi modificada (integridade)
 - a mensagem veio da fonte esperada (autenticidade)
- Se um atacante não conhece **K**, ele não pode modificar **M** e **MAC** e passar despercebido

MAC vs Criptografia Simétrica



MAC vs Criptografia Simétrica

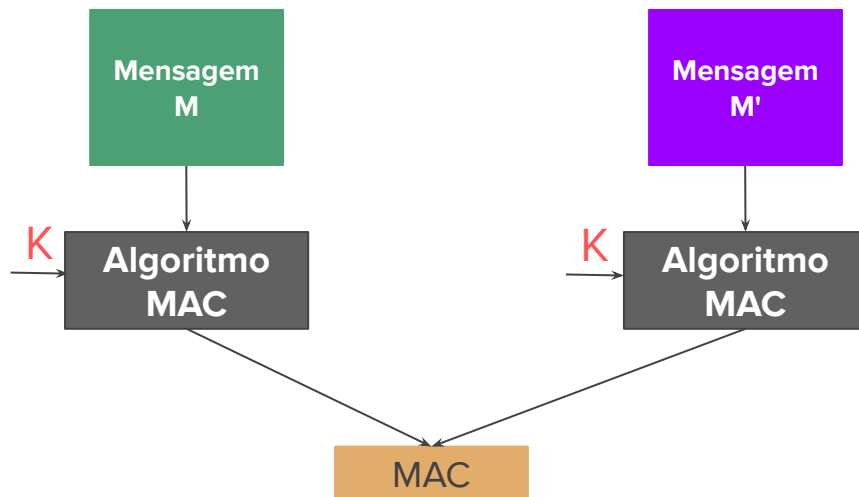
- Criptografia simétrica garante um certo nível de integridade e autenticidade
 - apenas quem possui a chave secreta pode cifrar e decifrar
 - se a mensagem for alterada durante a transmissão, a decifragem resultará em dados sem sentido
- Entretanto, MACs são mais interessantes quando:
 - a confidencialidade não é um requisito importante.
 - por exemplo, verificação de autenticidade de programas de computador
 - existe uma alta demanda e necessidade de eficiência
 - não é possível decifrar todas as mensagens recebidas, a autenticação é feita de maneira seletiva
 - a mensagem é enviada para vários destinatários (broadcast)
 - mensagens são enviadas em claro para todos, mas apenas um destinatário possui a chave secreta e verifica a integridade/autenticidade
 - se ocorrer um problema, os outros destinatários são notificados por um alarme geral

Sumário

- Definições Básicas
- **Requisitos e segurança**
- HMAC
- CMAC
- MAC na prática

Requisitos e segurança

- O **MAC** é uma função de muitos-para-um, ou seja, para uma chave **K**, muitas mensagens diferentes produzem o mesmo **MAC**
 - questões de colisão também são observadas aqui



Requisitos e segurança

- Assumimos que um atacante conheça a mensagem **M** e o **MAC** produzido, mas não conheça **K**
- O algoritmo MAC deve satisfazer os seguintes requisitos:
 - o atacante não deve ser capaz de criar uma outra mensagem **M'** que tenha o mesmo **MAC** da mensagem conhecida **M**
 - o algoritmo deve ser uniformemente distribuído, de maneira que um ataque de força bruta seja muito difícil
 - deve ser muito difícil para um atacante modificar **M** de maneira a produzir uma mensagem modificada **M'** que tenha o mesmo **MAC** da primeira.
- Medimos a segurança de um algoritmo MAC comparando o esforço para quebrá-lo com o esforço requerido para um ataque de força bruta
 - um algoritmo ideal requer um esforço igual ou superior

Algoritmos MAC

- Existem diversas maneiras de implementar um algoritmo de MAC
- Vamos explorar duas possibilidades:
 - MACs baseados em funções de hash (HMAC)
 - MACs baseados em cifras simétricas (CMAC)

Sumário

- Definições Básicas
- Requisitos e segurança
- **HMAC**
- CMAC
- MAC na prática

Keyed-Hash Message Authentication Code (HMAC)

- Funções de hash não possuem chaves e portanto não podem ser utilizadas diretamente na construção de MACs
- O HMAC surgiu da proposta de incorporação de uma chave secreta em uma função de hash
- Os principais objetivos do HMAC são:
 - utilizar funções de hash disponíveis sem modificá-las
 - permitir fácil substituição da função de hash, caso uma mais rápida ou mais segura seja criada
 - preservar a rápida performance das funções de hash
 - usar e manipular chaves de maneira simplificada
 - análise criptográfica bem compreendida, baseada na função de hash
- Padronização do HMAC: [FIPS PUB 198-1](#)
- Descrição e detalhes: [RFC 2104](#)

HMAC

- $\text{HMAC}(K, M) = H[(K \oplus \text{opad}) \parallel H[(K \oplus \text{ipad}) \parallel M]]$

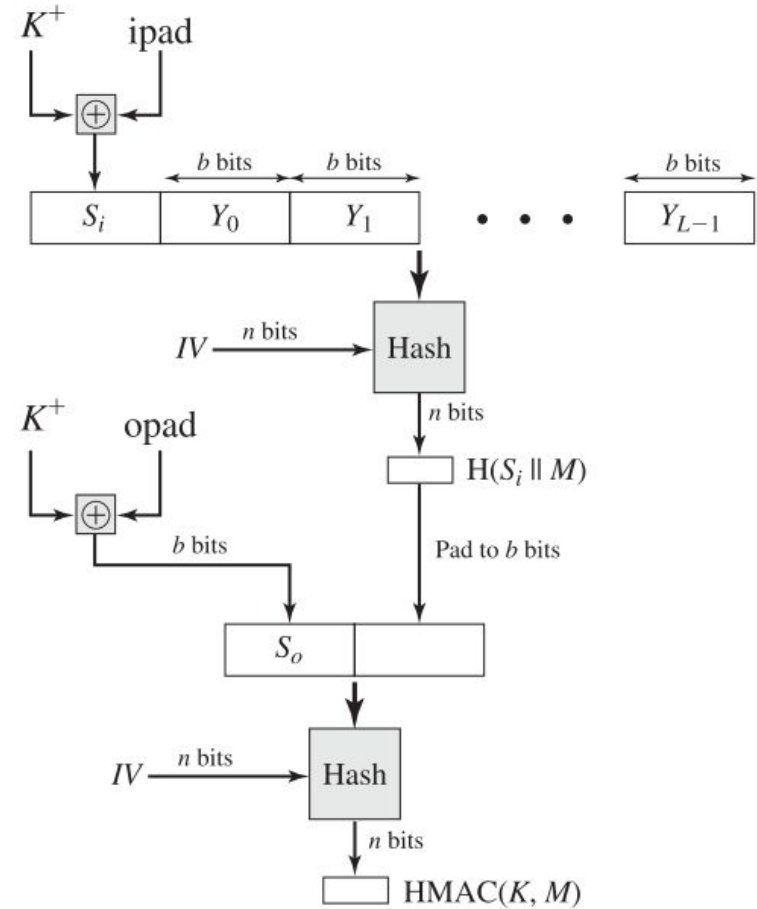


Figure 12.5 HMAC Structure

Fonte: W. Stallings. Cryptography and network security. Cap 12.5

HMAC

- $\text{HMAC}(K, M) = H[(K \oplus \text{opad}) \parallel H[(K \oplus \text{ipad}) \parallel M]]$
- Mensagem M é dividida em L blocos de b bits cada

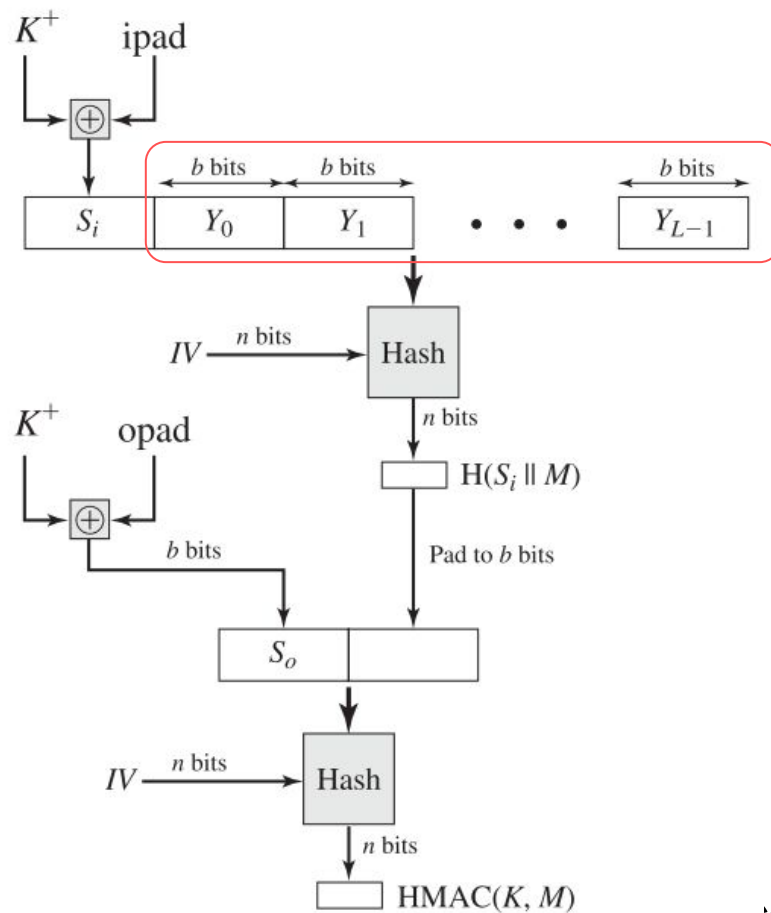


Figure 12.5 HMAC Structure

Fonte: W. Stallings. Cryptography and network security. Cap 12.5

HMAC

- $\text{HMAC}(K, M) = H[(K^+ \oplus \text{opad}) \parallel H[(K^+ \oplus \text{ipad}) \parallel M]]$
- Mensagem M é dividida em L blocos de b bits cada
- K^+ é a chave K concatenada com 0s à esquerda para completar b bits

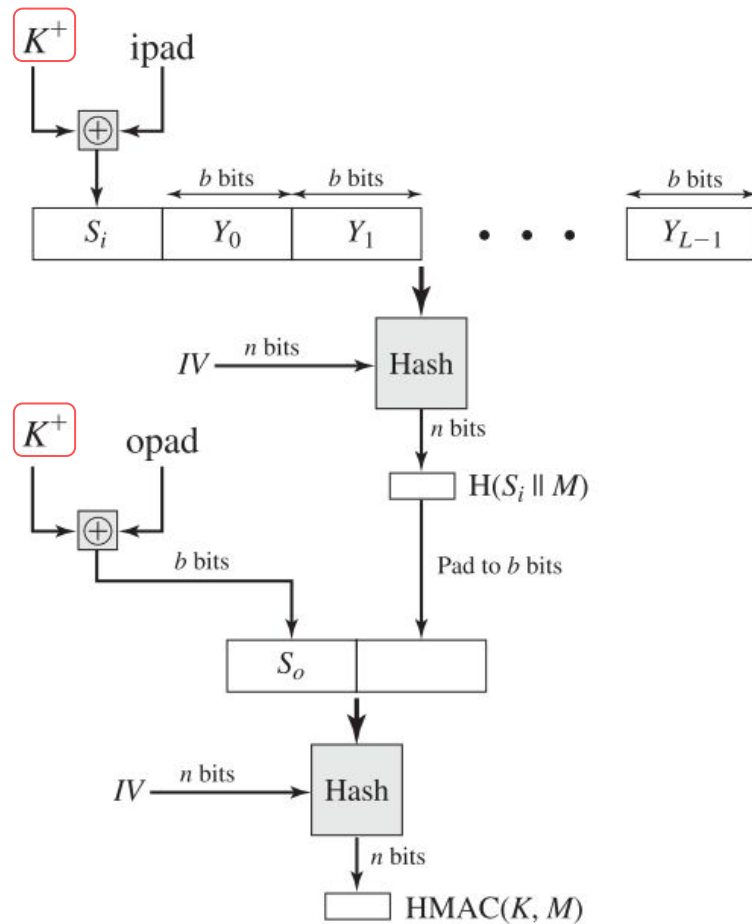


Figure 12.5 HMAC Structure

Fonte: W. Stallings. Cryptography and network security. Cap 12.5

HMAC

- $\text{HMAC}(K, M) = H[(K^+ \oplus \text{opad}) \parallel H[(K^+ \oplus \text{ipad}) \parallel M]]$
- Mensagem M é dividida em L blocos de b bits cada
- K^+ é a chave K concatenada com 0s à esquerda para completar b bits
- ipad: 00110110 = 0x36 (repetido $b/8$ vezes)
- opad: 01011100 = 0x5C (repetido $b/8$ vezes)

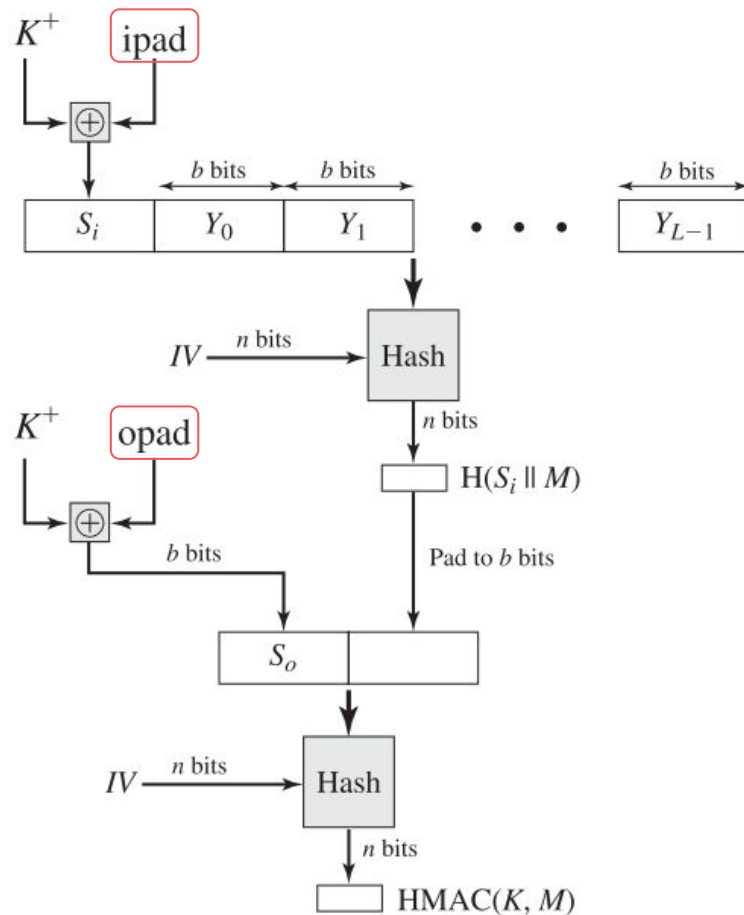


Figure 12.5 HMAC Structure

Fonte: W. Stallings. Cryptography and network security. Cap 12.5

HMAC

- $\text{HMAC}(K, M) = H[(K^+ \oplus \text{opad}) \parallel H[(K^+ \oplus \text{ipad}) \parallel M]]$
- Mensagem M é dividida em L blocos de b bits cada
- K^+ é a chave K concatenada com 0s à esquerda para completar b bits
- ipad: 00110110 = 0x36 (repetido $b/8$ vezes)
- opad: 01011100 = 0x5C (repetido $b/8$ vezes)
- opad e ipad fazem um flip dos bits de K

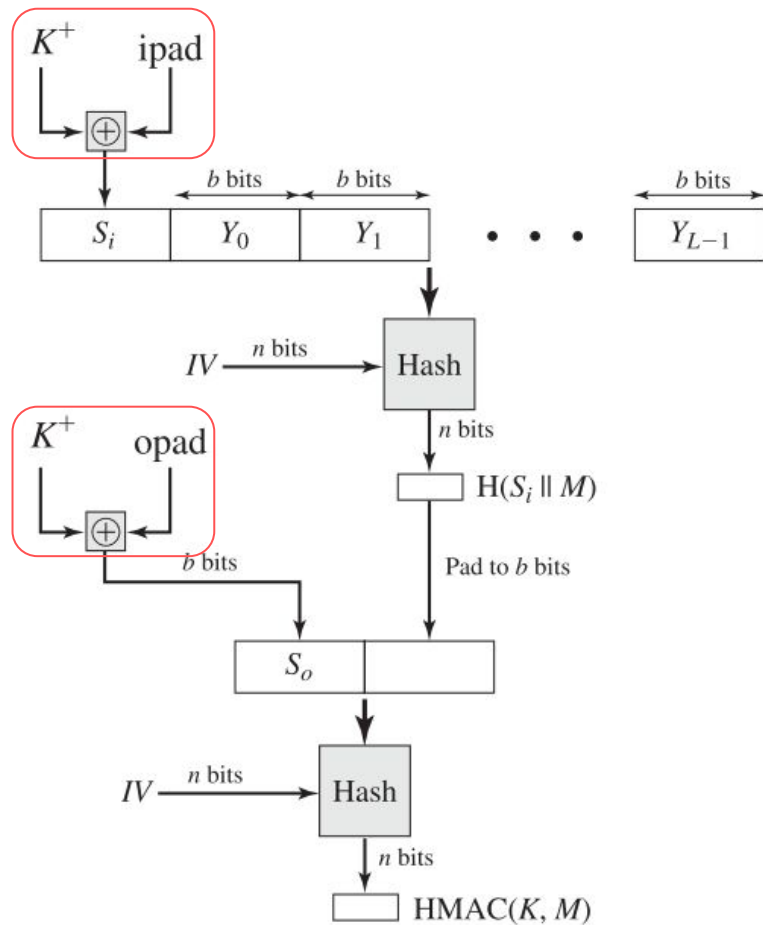


Figure 12.5 HMAC Structure

Fonte: W. Stallings. Cryptography and network security. Cap 12.5

HMAC

- $\text{HMAC}(K, M) = H[(K^+ \oplus \text{opad}) \parallel H[(K^+ \oplus \text{ipad}) \parallel M]]$
- Mensagem M é dividida em L blocos de b bits cada
- K^+ é a chave K concatenada com 0s à esquerda para completar b bits
- ipad: 00110110 = 0x36 (repetido $b/8$ vezes)
- opad: 01011100 = 0x5C (repetido $b/8$ vezes)
- opad e ipad fazem um flip dos bits de K
- podemos pré-computar o cálculo de S_i e S_o

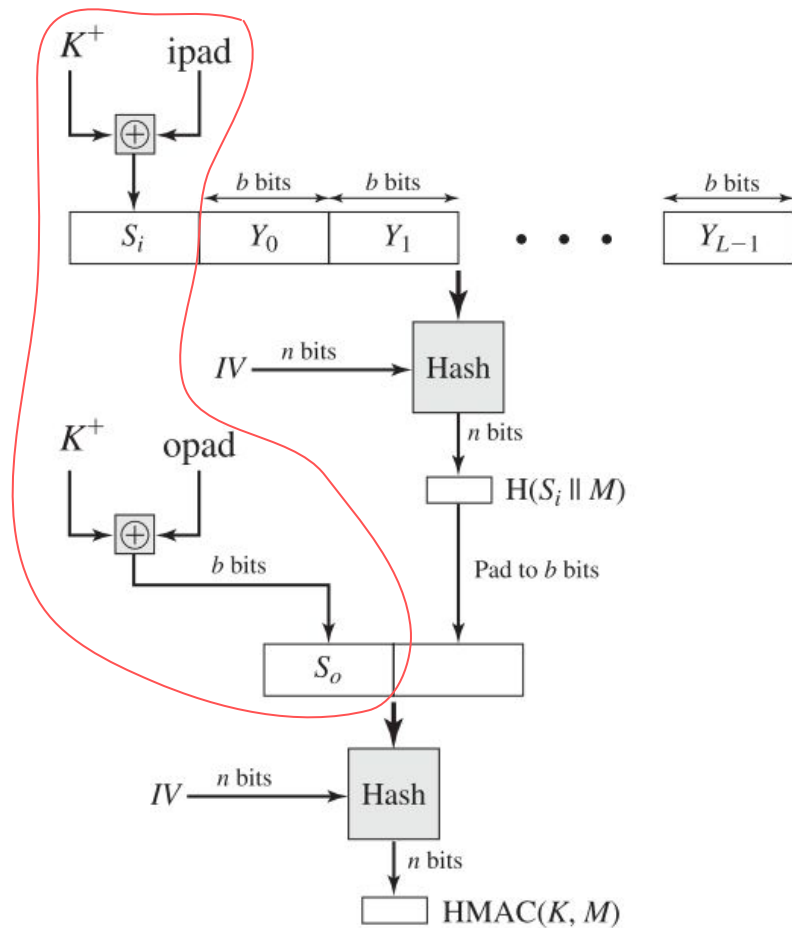


Figure 12.5 HMAC Structure

Fonte: W. Stallings. Cryptography and network security. Cap 12.5

HMAC - considerações finais

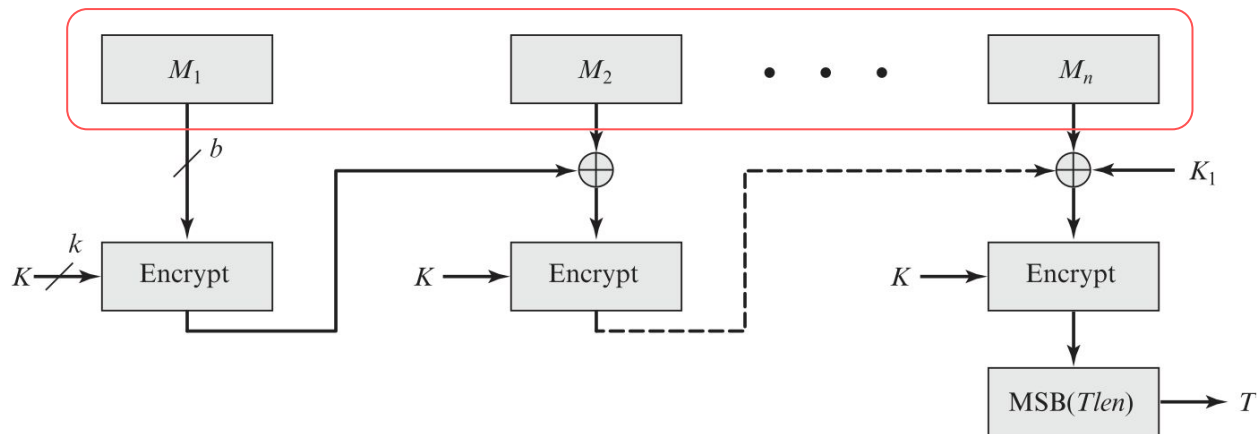
- A segurança de qualquer MAC baseado em funções de hash depende da segurança da função de hash utilizada;
- A probabilidade de sucesso de um ataque no HMAC é equivalente à probabilidade de se encontrar colisões na função de hash utilizada;

Sumário

- Definições Básicas
- Requisitos e segurança
- HMAC
- **CMAC**
- MAC na prática

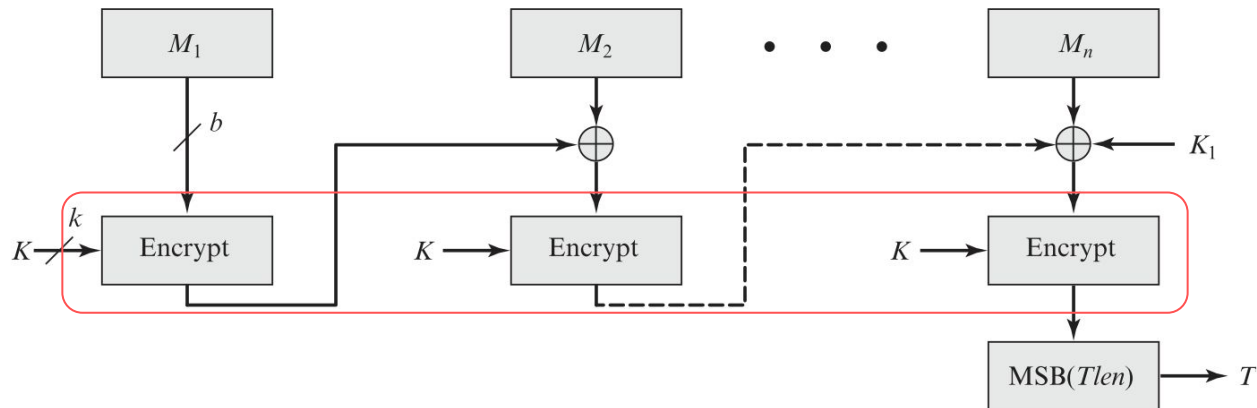
Cipher-Based Message Authentication Code (CMAC)

- Mensagem é dividida em n blocos de tamanho b



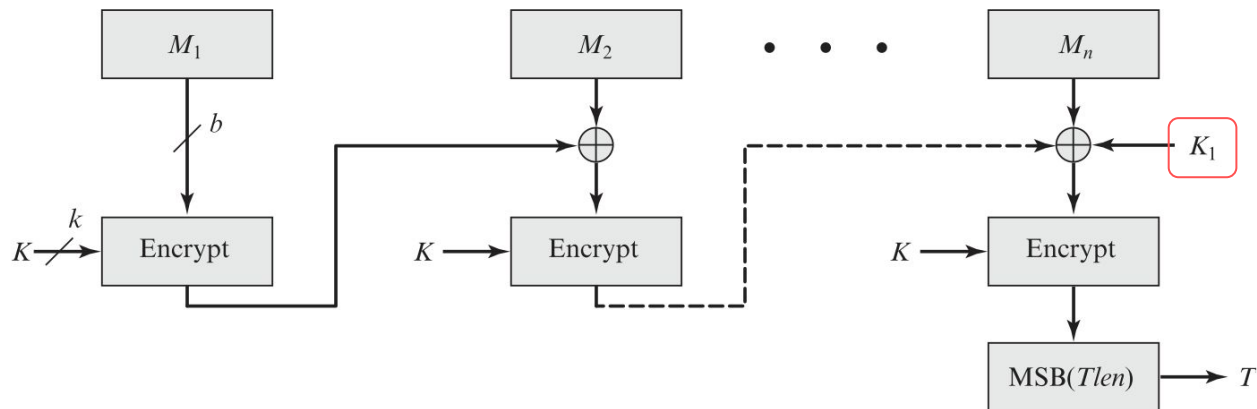
Cipher-Based Message Authentication Code (CMAC)

- Mensagem é dividida em n blocos de tamanho b
- Cada bloco é cifrado utilizando um algoritmo simétrico e uma chave K
 - b depende do algoritmo escolhido



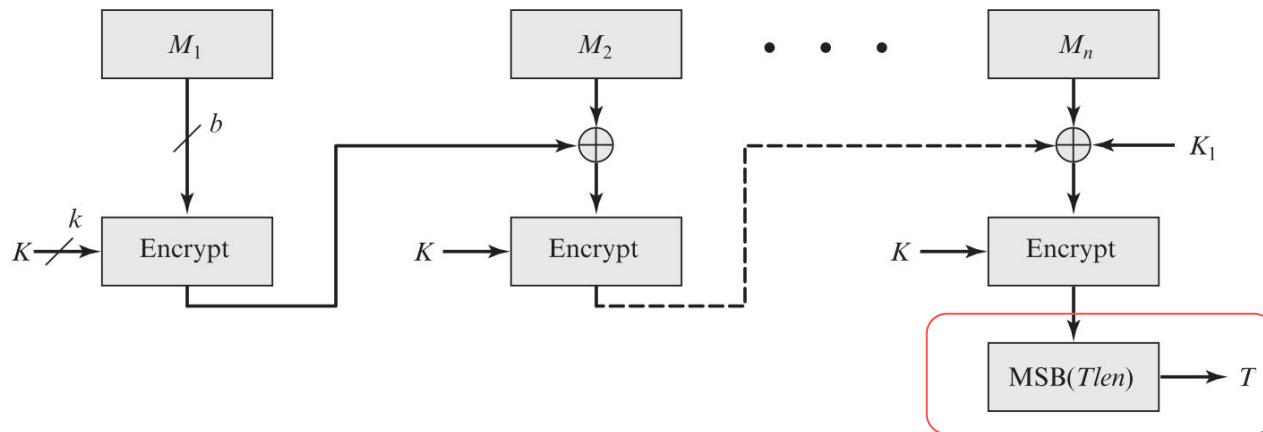
Cipher-Based Message Authentication Code (CMAC)

- Mensagem é dividida em n blocos de tamanho b
- Cada bloco é cifrado utilizando um algoritmo simétrico e uma chave K
 - b depende do algoritmo escolhido
- Uma sub-chave K_1 é utilizada no último bloco



Cipher-Based Message Authentication Code (CMAC)

- Mensagem é dividida em n blocos de tamanho b
- Cada bloco é cifrado utilizando um algoritmo simétrico e uma chave K
 - b depende do algoritmo escolhido
- Uma sub-chave K_1 é utilizada no último bloco
- A saída consiste nos bits mais significativos (mais à esquerda)



Considerações finais

- CMAC é especificado na [NIST SP 800-38B](#)
- A cifra simétrica indicada é AES ($b = 126$)
 - o 3DES ($b = 64$) não é mais permitido desde 2023
- Funções de hash geralmente são mais eficientes em software do que cifras simétricas
- Aplicações com hardware dedicado para cifras podem oferecer um desempenho superior ao CMAC
- HMAC oferece maior flexibilidade e implementação mais simples, já que pode ser utilizado com diversas funções de hash que são amplamente disponíveis nas bibliotecas

Sumário

- Definições Básicas
- Requisitos e segurança
- HMAC
- CMAC
- **MAC na prática**

Aplicações

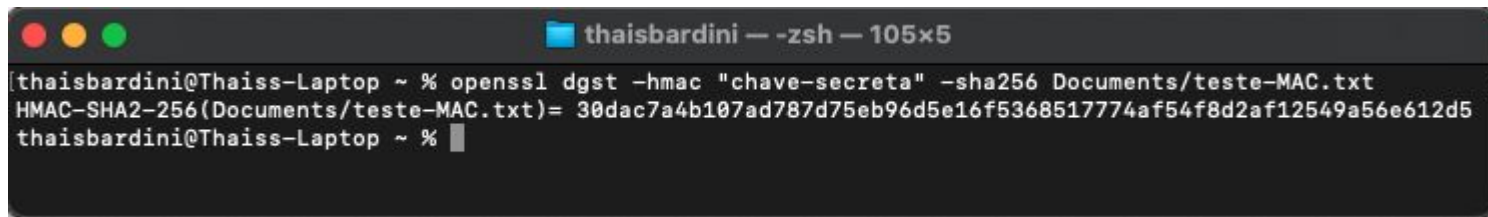
- No protocolo TLS (versão 1.2), utilizado para garantir a integridade e autenticidade das mensagens trocadas;
- No JSON Web Token (JWT) para prover segurança nos tokens;
- No Amazon Web Services (AWS), para garantir integridade e autenticidade de mensagens trocadas entre os diferentes componentes e serviços e autenticar requisições de usuários;
- entre outras.

Atividade: calculando MACs

- Vamos praticar utilizando o openssl:

`openssl dgst -hmac "chave" -sha256 nome-arquivo`

- Calcule o HMAC do arquivo teste-MAC.txt usando a senha "chave-secreta" e compare o valor obtido:



```
thaisbardini — -zsh — 105x5  
[thaisbardini@Thaiss-Laptop ~ % openssl dgst -hmac "chave-secreta" -sha256 Documents/teste-MAC.txt  
HMAC-SHA2-256(Documents/teste-MAC.txt)= 30dac7a4b107ad787d75eb96d5e16f5368517774af54f8d2af12549a56e612d5  
thaisbardini@Thaiss-Laptop ~ % ]
```

- Modifique o arquivo e gere novamente o HMAC
 - observe o efeito avalanche ao se modificar o valor de entrada
- Modifique a senha e observe a modificação no valor de saída
- Modifique a função de hash e observe a modificação no tamanho da saída

Resumo

- Message Authentication Codes (MACs) são utilizados na garantia de integridade e autenticidade de dados;
- Produzem uma saída de tamanho fixo conhecida por MAC, *tag* ou *checksum*;
- Podem ser baseadas em funções de hash ou em cifras simétricas;
- A segurança dos MACs depende da segurança da função de hash ou cifra utilizada;

Referências

- W. Stallings. *Cryptography and network security*. 7a edição.
 - Capítulos 12.1, 12.2, 12.3, 12.4, 12.5, 12.5
- D. Stinson e M. Paterson. *Cryptography: Theory and Practice*. 4a edição.
 - Capítulos 5.5 e 5.6
- [RFC 2104](#)
- [FIPS PUB 198-1](#)
- [NIST SP 800-38B](#)