

# Função Hash

- **Integridade ?**

- Um processo que permite verificar se uma mensagem (texto, código, imagem, etc) foi alterado, intencional ou acidentalmente, durante sua transmissão ou ao longo de sua existência.
- A técnica consiste em anexar a uma mensagem um resumo de tamanho relativamente pequeno, como 128 bits, através do qual pode-se verificar a integridade da mesma

# Função Hash

- Integridade ?

$$\text{Hash} = f \left( \begin{array}{c} \text{meditar} \\ \text{produz} \\ \text{sabedoria} \end{array} \right)$$

**Função resumo:** produz resultados diferentes, para documentos eletrônicos diferentes

# Função Hash

## Propriedades da Função Hash

- $H$  deve ser aplicada a qualquer tamanho de bloco
- $H$  deve produzir uma saída de tamanho fixo
- Fácil de computar  $y = H(x)$  em software ou hardware
- Inviável computar  $x = H^{-1}(y)$
- Dado  $x$ , é inviável obter  $y \neq x$  com  $H(y) = H(x)$
- É inviável obter-se  $(x,y)$  tal que  $H(x) = H(y)$

# Função Hash

- Função Hash simples

- $h_i = b_{i1} \oplus b_{i2} \oplus \dots \oplus b_{im}$

- Ex: Meditar Produz Sabedoria

- $h = 'M' \oplus 'e' \oplus 'd' \oplus \dots \oplus 'a' \Rightarrow 10101101$

# Secure Hash Algorithm (SHA)

- Documento é transformado em blocos de 512 bits
  - Inserido um **1** seguido de **0s**, tornado-o múltiplo de 512 menos 64
- Inserido tamanho original do documento
  - Acrescido um bloco de 64 bits que contém o seu tamanho original
- Inicializado buffer (160 bits) para resultados intermediários e final
  - **A** = 0x67452301 **B** = 0xefcdab89 **C** = 0x98badcfe **D** = 0x10325476 **E** = 0xc3d2e1f0
- Documento é processado em blocos de 512 bits:
  - São aplicadas 4 rodadas de 20 operações cada.
  - $t_{th} = f_t(b, c, d) + (a \lll 5) + e + W_t + K_t$
- O resultado é um resumo de **160** bits

# Secure Hash Algorithm (SHA)

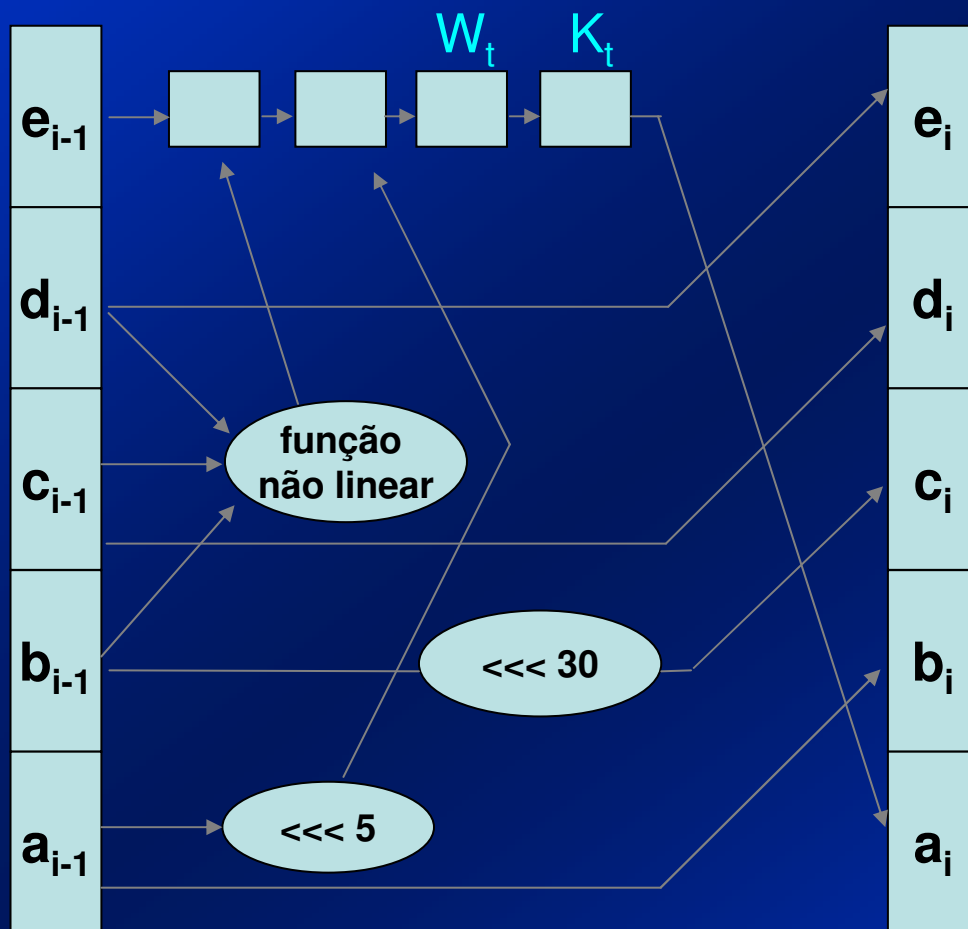
Função não Linear

- $f_t(X, Y, Z) = (X \wedge Y) \vee (\sim X \wedge Z)$ , for  $t = 0$  to  $19$ .
- $f_t(X, Y, Z) = (X \oplus Y \oplus Z)$ , for  $t = 20$  to  $39$ .
- $f_t(X, Y, Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$ , for  $t = 40$  to  $59$ .
- $f_t(X, Y, Z) = (X \oplus Y \oplus Z)$ , for  $t = 60$  to  $79$ .

Expansão

- $W_t = M_t$ , for  $t = 0$  to  $15$
- $W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \lll 1$ , for  $t = 16$  to  $79$

# Secure Hash Algorithm (SHA)



$K_t = 0x5a827999$ , for  $t = 0$  to  $19$

$K_t = 0x6ed9eba1$ , for  $t = 20$  to  $39$

$K_t = 0x8f1bbcdc$ , for  $t = 40$  to  $59$

$K_t = 0xca62c1d6$ , for  $t = 60$  to  $79$

# Benefício da Função Hash

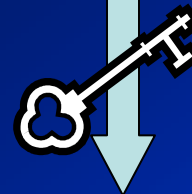
meditar  
produz  
sabedoria

Hash

integridade

010101010

- **Criptografia Assimétrica**
  - Autenticidade
  - Confidencialidade, sigilo



110100011  
101011010

autenticidade