Tópicos em Redes de Computadores

Aula de Hoje: Parte 1
Hash (Resumo Digital)

Prof. Elias P. Duarte Jr.
Universidade Federal do Paraná (UFPR)
Departamento de Informática
www.inf.ufpr.br/elias/topredes

Sumário

- Vamos estudar os hashes – resumo digital

O que é um *hash*?

- Uma função que faz o mapeamento de uma entrada A em uma saída B
- Mas não faz o mapeamento da saída B na entrada A
- Em outras palavras: é impossível descobrir a entrada, dada a saída
- *It's a one-way function*
- Também chamada de *message digest* ou, em português: resumo digital

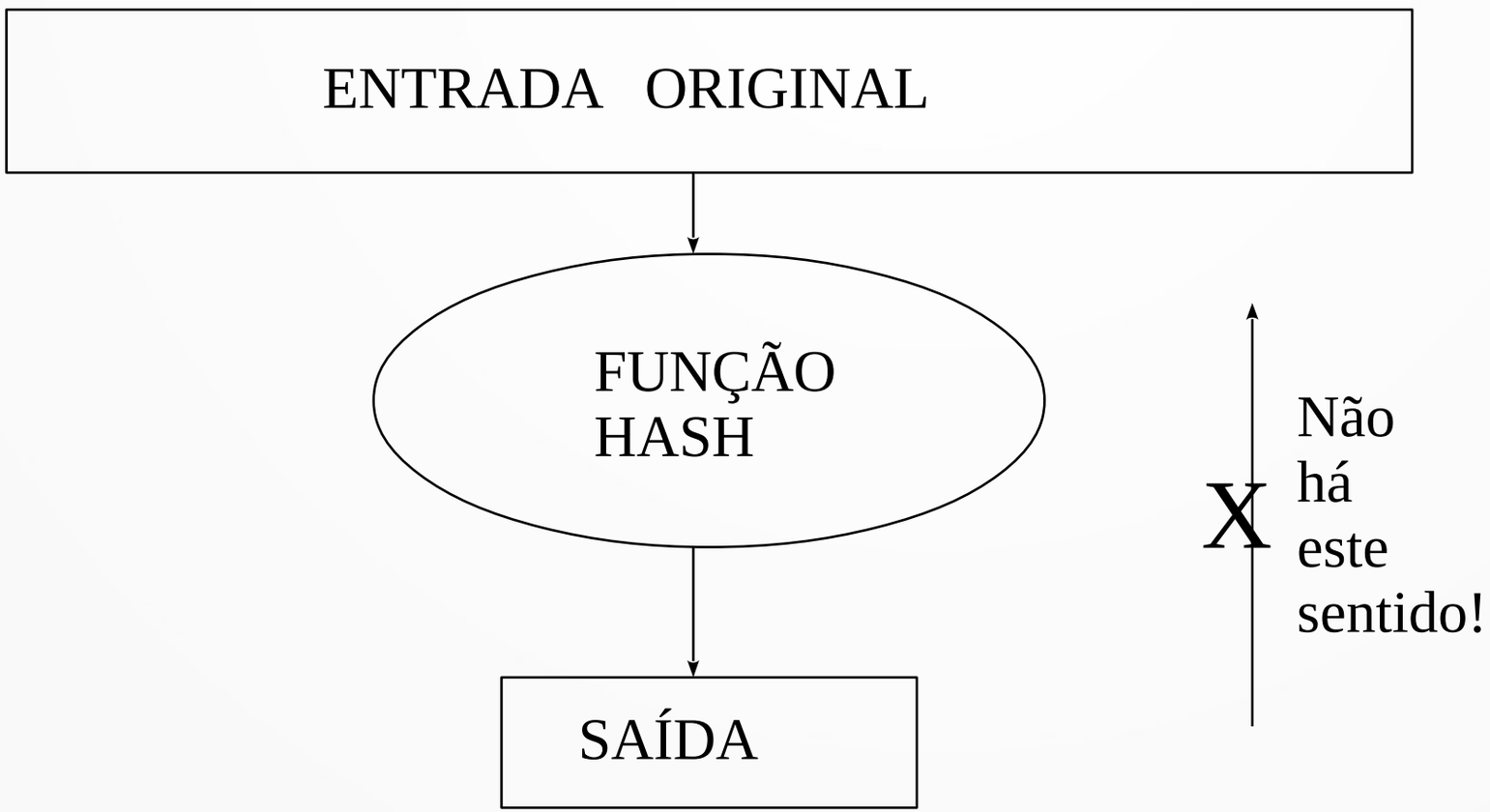
Exemplo de Aplicação

- Armazenamento de *passwords* - quando o usuário digita o sistema simplesmente calcula o hash do string
- Uma observação: não há volta da função, há perda de informação no processo
- Assim é impossível recuperar a senha pela informação armazenada

Hash: Entradas e Saídas

- As entradas comuns são
  - arquivos
  - strings
- As saídas são, em geral, números entre
  - 128
  - 160
  - ou 256 bits

Diagrama de uma função hash



Características Básicas

- A entrada tem tamanho arbitrário em número de bits, a saída tem número fixo de bits
- Deve ser computacionalmente difícil encontrar uma entrada que produza a saída
- Da mesma forma, deve ser difícil encontrar duas entradas que produzam o mesmo hash

O Hash Parece Aleatório

- Não deve ser possível prever como será um bit da saída, dada a entrada
- Cada saída deve ter, idealmente, metade dos bits setados em 1, a outra metade em 0
- Dadas 1000 saídas, um bit específico deve ser 1 em aproximadamente metade delas
- Mesmo que duas entradas sejam muito parecidas, as saídas devem ser totalmente diferentes

Propriedades do Hash

- Cada bit da saída é influenciado por cada bit da entrada
- Se um bit da entrada é trocado, cada um dos bits da saída tem 50% de chance de ser trocado
- Dada uma entrada e seu correspondente resultado de hash (mensagem não criptografada, hash usado para integridade) deve ser difícil obter um segundo arquivo com o mesmo resultado

Hashes Importantes

- MD2: *Message Digest* #2; desenvolvido por Ronald Rivest - produz hashes de 128 bits, é a mais segura das funções de Rivest, mas a mais lenta de calcular...
- MD4: *Message Digest* #4; também desenvolvida por Ron Rivest - e também produz hashes de 128 bits, mas é rápida, e mostrou-se insegura... (foi publicada forma de encontrar duas entradas com mesmo hash)

Os Mais Usados Hoje

- MD5: *Message Digest* #5; mesmo autor das anteriores; resultado também é de 128 bits, a mais usada hoje, é uma melhorias de MD4, é o mais usado de todos – foi quebrado também...
- SHA: *Secure Hash Algorithm*; produzida pelo NSA; resultado de 160 bits, versões 1 e 2

Usos de Hash

- Não são usados para criptografia pura
- E sim para:
  - criação de assinaturas digitais
  - autenticação da origem de mensagens
- É uma excelente ferramenta para verificar se houve uma pequena mudança num arquivo

MD5 em Funcionamento

- Por exemplo, com a entrada:
  - MD5(A chave está escondida no tapete)
- A saída seria
  - 05f8cfc03f4e58cbee731aa4a14b3f03
- Se fizermos uma pequena alteração:
  - MD5(A chave está escondida no tapete!)
- A saída muda completamente!
  - d6dee11aae89661a45eb9d21e30d34cb

MD5: Continuando

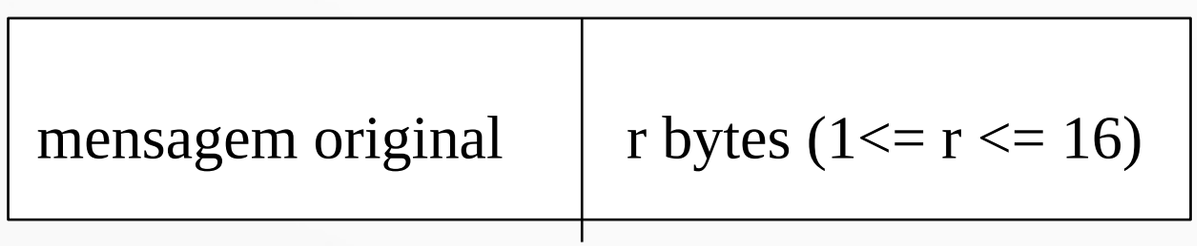
- Mas, lembre-se, para a mesma entrada
- MD5(A chave está escondida no tapete)
- é sempre gerada a mesma saída
- 05f8cfc03f4e58cbee731aa4a14b3f03
- Mudando a entrada
- MD5(Hoje o dia está ensolarado)
- Muda a saída completamente
- 050f3905211cddf36107ffc361c23e30d

Como calcula um hash?

- Vamos ver como um hash MD2 é calculado
- O processo pode ser visto da seguinte forma:
- A entrada é vista uma sequência de bytes individuais (8 bits)
- É expandida (padding) p/ múltiplo de 16 bytes
- Além disso: um checksum de 16 bytes é anexado
- Pedaços de 16 bytes são processados um por vez calculando um valor intermediário

Expandindo a Entrada

- Para que o número de bytes da entrada se transforme em múltiplo de 16, devem ser acrescentados de 1 a 16 bytes
- Se a mensagem já for um múltiplo de 16, 16 bytes são acrescentados
- Inclui informação sobre o # de bytes inserido

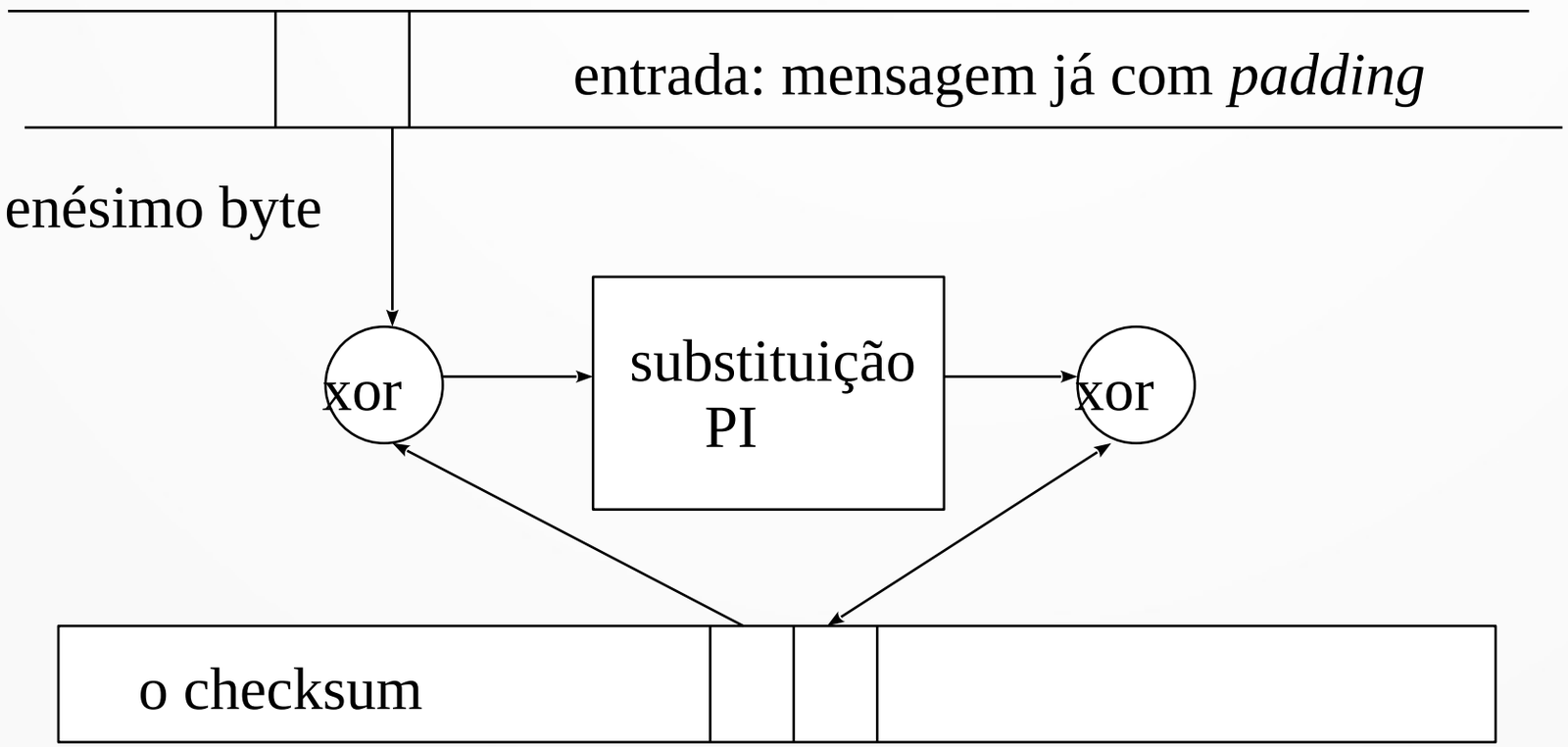


O Cálculo do Checksum do MD2

- Um checksum de 16 bytes (0..15) é calculado
- É inicializado em 0 (zero)
- Em cada passo seguinte, um byte da mensagem atualiza um byte do checksum
- Circular: quando o byte 15 do checksum é atualizado, o algoritmo volta ao byte 0
- Assim, cada byte é atualizado várias vezes

O Checksum do MD2

O cálculo do checksum do MD2 é feito como mostrado a seguir:



Cálculo do Checksum do MD2

- A substituição constitui de uma tabela mapeando os 256 possíveis valores de um byte
  - para valores bem distintos
- Exemplos: 0 -> 41, 1 -> 46, 2-> 67,..., 253 -> 17, 254 -> 131, 255 ->20
- Os autores dizem que usaram os dígitos do número PI para calcular a substituição

O Passo Final do MD2

- É similar ao cálculo do checksum
- Recebe como entrada a mensagem, com padding e checksum e vai calculando um valor intermediário de 48 bytes
- Estes 48 bytes passam por 18 (dezoito!) passos intermediários
- No final: os 16 primeiros bytes são o hash da mensagem

Os Hashes MD5 e SHA

- Os algoritmos são do mesmo estilo
- Vários passos sobre os bytes da entrada
- Fazendo substituições e transposições
- Aplicando funções específicas
- MD5: já foi quebrado, inseguro para aplicações sensíveis
- SHA: *Secure Hash Algorithm* – o mais recomendado!

A Popularidade dos Hashes

- Um algoritmo que calcula hash é muito mais rápido que um algoritmo de criptografia de chave secreta
- Assim - quando a tarefa permitir - deve ser usado hash no lugar de criptografia (ex.: verificação de integridade)
- Além disso não há patentes controlando os hashes, não há restrições à importação nos EUA e outros países

Hash e Criptografia de Chave Pública

- Os algoritmos de criptografia com chave pública são bastante pesados
- Ao invés de assinar todo um documento/mensagem, é assinado apenas seu hash!

Ataque ao Hash

- Dado um hash e sabendo o algoritmo usado para gerá-lo: achar uma [segunda?] mensagem que produz o mesmo hash
- Mesmo problema de sempre: a força-bruta é sempre possível...
- ... mas deve ser computacionalmente difícil

Outro Ataque ao Hash

- Em outro ataque ao hash, dada uma mensagem e seu hash, o objetivo é alterar a mensagem
- Esta [carta/mensagem] é para [informar/lamentar que o ouro [não chegará/será entregue] [no dia 7/ nesta semana]
- A ideia é ir tentando substituir palavras

O Paradoxo do Aniversário

- Se houver 23 pessoas em uma sala, a probabilidade de duas delas fazerem aniversário no mesmo dia é de 50%!
- N pessoas, N*(N-1)/2 pares de pessoas (23X22)/2 = 253
- Cada par tem a probabilidade de 1/365 de ser igual

Generalizando

- Considere N entradas e k saídas
- Existem N(N-1)/2 pares na entrada
- Se o número de pares > k a chance de um par “empatar” é boa
- Um empate tem grande probabilidade quando

  N > raiz de k

O Paradoxo do Aniversário em Hashes

- Se o hash for de 64 bits então há uma probabilidade de que tentando 232 mensagens seja possível achar uma que substitua a original

Conclusão

- Vimos os hashes: o 3º tipo de criptografia
- Extremamente importantes: assinatura digital
- Algoritmos mais recomendado hoje: SHA

Obrigado!

Lembrando: a página da disciplina é:
https://www.inf.ufpr.br/elias/topredes