



Conhecimento é a nossa natureza

CCC306 - CIBERSEGURANÇA

Funções Hash

Prof. Leonardo Costella

Introdução

Uma função hash é um algoritmo que mapeia uma entrada de dados de tamanho arbitrário para uma saída de tamanho fixo, chamada de valor hash ou digest.

- As funções Hash são conhecidas por resumir o dado
 - Algoritmo de resumo de mensagem

```
Entrada: "senha123"
```

```
Hash (SHA-256): ef92b778ba5cae6417e9edb3... (64 caracteres hexadecimais)
```

Introdução

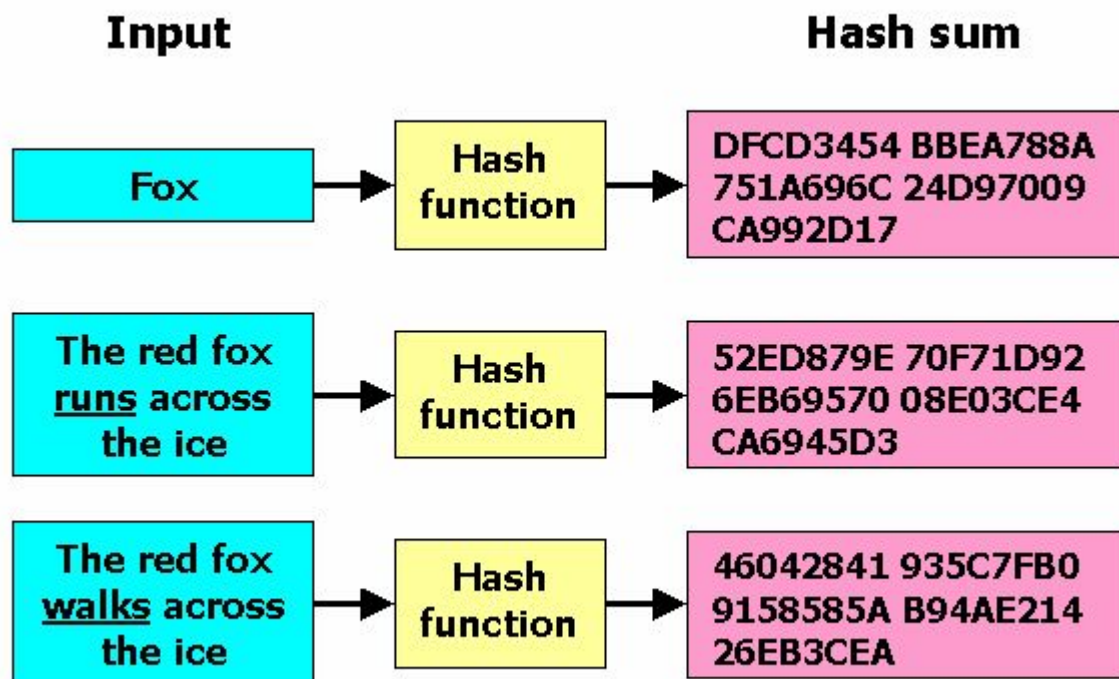
- **Confidencialidade:** disponibiliza as informações somente para usuários autorizados.
- **Integridade:** garante que as informações não tenham sido manipuladas.
- **Autenticação:** confirma a autenticidade das informações ou a identidade de um usuário.
- **Disponibilidade:** As informações precisam estar disponíveis para as pessoas, no momento certo em que precisarem, seguindo as regras de confidencialidade.
- **Não repúdio:** impede que um usuário negue compromissos ou ações anteriores.



Funções Hash x Criptografia



Funções Hash



Funções Hash - Características:

- **Saída de tamanho fixo:** independente do valor de entrada, as saídas possuem a mesma quantidade de letras e números.
- **Eficiência de operação:** a função não pode ser complexa ao ponto de comprometer a velocidade de processamento;
- **Determinística:** o valor de entrada (input) sempre possuirá equivalência ao valor da saída (output).



Funções Hash - Aplicações:

- Efetuar buscas de elementos em banco de dados e em estruturas de dados em memória - Construção de índices
- Fazer a verificação da integridade de arquivos baixados
 - Utilizam-se as funções de hash diretamente sobre o dado, salvando-se o resumo do dado que foi gerado. Depois que o dado é transmitido para o seu receptor, ele calcula o resumo do dado que foi recebido e, assim, adquire um novo resumo do dado. Se os resumos de dados forem iguais, então o dado é igual, é **íntegro**.
- Fazer o armazenamento e a transmissão de senhas de usuários
 - Armazenamento de apenas o resumo da senha no servidor: Quando o usuário dá entrada com a sua senha, uma função hash calcula o resumo da senha, e o servidor faz a comparação com o resumo que está armazenado. Se os resumos das senhas forem iguais, o usuário será autenticado

Funções Hash - Funcionamento

Existem muitos tipos de funções hash, e a complexidade do código do seu algoritmo vai depender de qual característica pretende-se garantir com a função:

- **Unidirecionalidade:** Essa característica significa que não é possível recuperar um dado original partindo do resumo do dado que foi gerado pela função.
 - A unidirecionalidade é uma das principais diferenças entre uma função *hash* e uma função de criptografia;
- **Resistência à colisão:** que acontece quando dados originais geram o mesmo resumo, ao ser aplicada a função. Quem elabora uma função hash tem como objetivo reduzir a colisão ao menor nível possível.
 - Resistir à colisão não significa evitar a sua existência ou fazer com que nenhuma colisão aconteça.

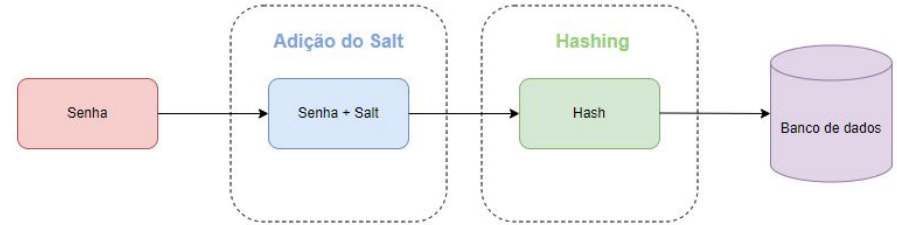
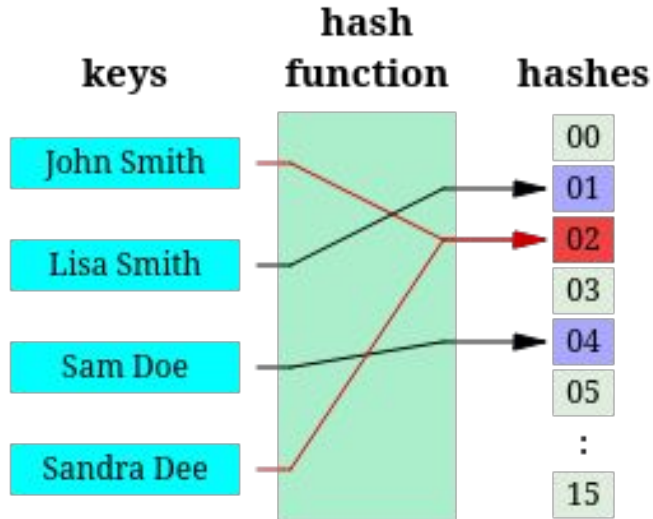



Funções Hash - Funcionamento

Existem muitos tipos de funções hash, e a complexidade do código do seu algoritmo vai depender de qual característica pretende-se garantir com a função:

- **Recorrência:** Sempre que uma função hash for aplicada sobre os mesmos dados originais serão gerados os mesmos resumos para os dados.
- **Grande quantidade de resumos possíveis:** Hoje em dia, as funções de hash são capazes de gerar resumos de até 512 bits, o que quer dizer que possuem 10^{48} possibilidades de resumo.
 - Além disso, é possível adicionar uma chave no momento da geração do resumo. Dessa forma, somente aquele que conhecer a chave será capaz de gerar o resumo de um determinado dado. Em caso de ataques na transmissão dos arquivos o atacante, sem conhecimento da chave, não tem condições de gerar um resumo que seja válido para a chave utilizada.

Funções Hash - Funcionamento



				
Password	p4s5w3rdz	p4s5w3rdz	p4s5w3rdz	p4s5w3rdz
Salt	-	-	et52ed	ye5sf8
Hash	f4c31aa	f4c31aa	1vn49sa	z32i6t0

Algoritmos Hash

- A transformação de um dado original em um hash envolve o cálculo do valor da função que vai colocar em prática essa transformação. O bloco de dados gerado pela função hash tem um comprimento fixo, enquanto os dados de entrada possuem tamanho variável. Qualquer que seja o comprimento dos dados de entrada, o hash de saída vai ser sempre um hash com o mesmo comprimento.
- A geração de números aleatórios faz com que as funções hash gerem identificadores com baixa probabilidade de colisões, mas não existe uma função hash que seja perfeita, que não apresente colisão nenhuma. Existem funções hash cuja taxa de colisão é bem pequena e levando em consideração que provavelmente vários dados vão dar origem ao mesmo hash, é preciso haver um método para trabalhar com as colisões encontradas.



Algoritmos Hash - Exemplos

- **MD5:**
 - Desenvolvido pela RSA Data Security em 1994 para substituição do **MD4**;
 - Foi muito popular e utilizado para a verificação da integridade de arquivos baixados da Internet e também para a verificação de login de usuários
 - Produz um valor hash de 128 bits: ***25d55ad283aa400af464c76d713c07ad***
 - Variação MD6: trabalha com hash de 224, 256, 384 e até 512 bits.
 - **Algumas vulnerabilidades foram encontradas:**
 - **Grande quantidade de colisões;**
 - **Falhas no algoritmo;**
- Recomenda-se a não utilização para aplicações que exigem maior segurança



Algoritmos Hash - Exemplos

- **SHA:** *Secure Hash Algorithm*, ou algoritmo de dispersão seguro) foi o primeiro algoritmo da família SHA, publicado pelo *National Institute of Standards and Technology (NIST)*, e é baseado no algoritmo MD5.
 - SHA-1: gera um *hash* de 160 bits (20 bytes) e corrige alguns problemas de segurança do SHA-0
 - SHA-2: trabalha com resumos de 224, 256, 384 e 512 bits.
 - SHA-3: criado em 2007 e liberado para domínio público em 2015
- **Utilização:**
 - TLS, SSL, PGP e IPsec;
 - Controle de revisão Git
 - Verificação de transações e cálculos de criptomoedas, como o Bitcoin.



Algoritmos Hash - Ataques comuns

Ataque do aniversário:

- O objetivo do ataque é encontrar entradas ou dados originais iguais, que irão gerar resumos iguais também.
- Podem ser elaboradas funções matemáticas para encontrar entradas iguais em um grupo, ou simplesmente escolher as entradas de maneira aleatória.
- Para evitar esse tipo de ataque, o comprimento da função hash empregado para os dados de saída deve ser suficientemente grande para que o ataque de aniversário se torne inviável matematicamente.
- **Para entender como é fácil encontrar dados originais iguais, que servirão para gerar um ataque de aniversário, basta imaginar que, em um grupo de mil pessoas, a probabilidade de duas ou mais pessoas terem nascido no mesmo dia é bem grande.**



Algoritmos Hash - Ataques comuns

Ataque de força bruta:

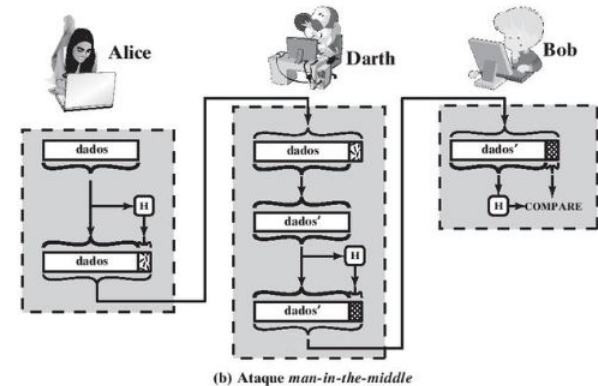
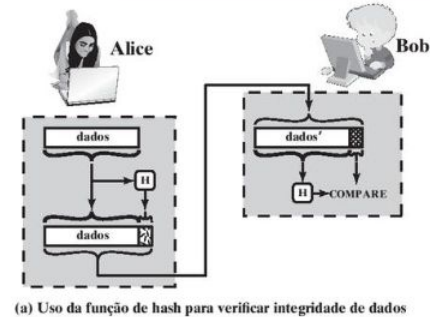
- Também pode ser conhecido como um ataque de busca de chave por exaustão
- Consiste na verificação sistemática em massa de todas as combinações possíveis de dados originais e resumos, até que a combinação correta seja encontrada.
 - O sucesso desse ataque envolve o conhecimento do tamanho correto dos dados originais e dos resumos gerados



Algoritmos Hash - Ataques comuns

Ataque *man in the middle*:

- Nesse tipo de ataque, os dados que são trocados entre o remetente e o destinatário sofrem interceptação de alguma maneira (são registrados e corrompidos ou alterados pelo indivíduo malicioso) sem que as vítimas percebam que estão sendo atacadas.



Algoritmos Hash - Ataques comuns

Ataque de colisão:

- O ataque de colisão é feito por meio da geração de uma colisão no resultado da aplicação de um algoritmo de hash. A colisão acontece quando dados originais diferentes entre si geram como resultado um hash igual.
- Utilizado pela Microsoft para provar a ineficácia do algoritmo MD5 com relação a certificados digitais. Nesse ataque, foram gerados dois certificados com um mesmo hash. Um dos certificados foi enviado para uma autoridade certificadora e, depois que ele foi assinado, essa assinatura foi copiada para o outro certificado, que se tornou assinado também, só que de maneira inválida.



Referências Bibliográficas

- BARRETO, Jeanine S.; ZANIN, Aline; MORAIS, Izabelly S.; et al. Fundamentos de segurança da informação. Porto Alegre: SAGAH, 2018. E-book. p.2. ISBN 9788595025875. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788595025875/>
- STALLINGS, William. Criptografia e segurança de redes: princípios e práticas. 6. ed. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br>.