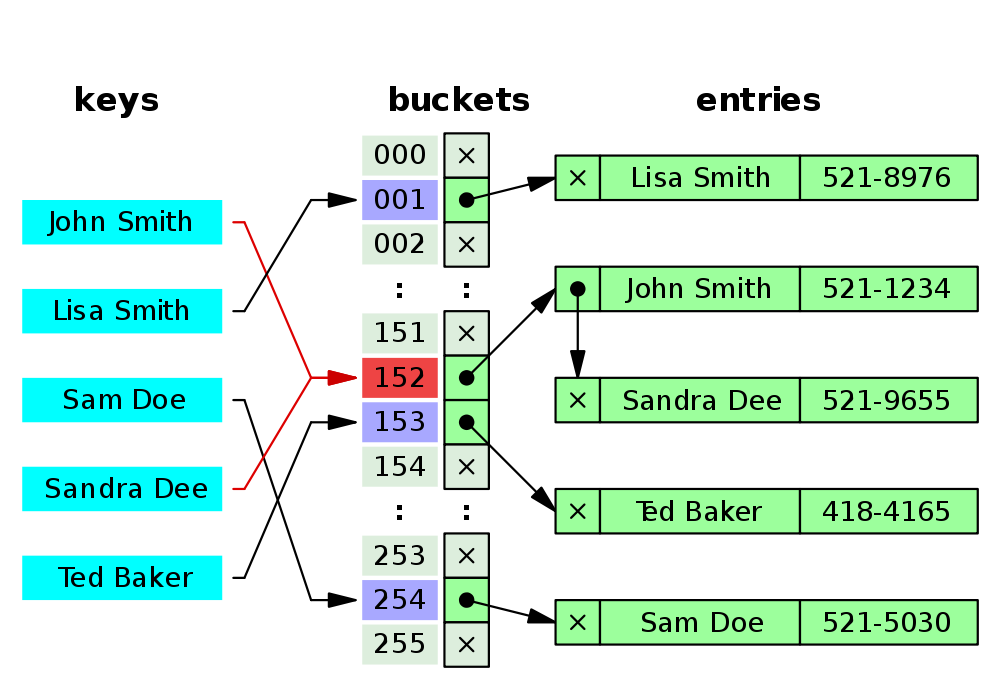
**Aplicações**

Vamos abordar três aplicações: busca de elementos em bases de dados, verificação de integridade de dados grandes e o armazenamento de senhas com segurança. A busca de elementos baseado em resumos é usada tanto em bancos de dados quanto em estruturas de dados em memória. O funcionamento se baseia na construção de índices, que funcionam de forma semelhante ao índice de livros.

Quando você busca algo em um livro, antes olha no índice para ver em que página está a informação que você precisa. Após, dirige-se diretamente para a página desejada. Nos índices baseados em Hashs, a busca pela página da informação é feita pelo resumo e não pelo dado. Em geral, calcula-se o resumo do dado e, com o resumo, sabe-se em que página o dado está. Não precisa-se varrer todo o índice para descobrir a página em que o dado se encontra.



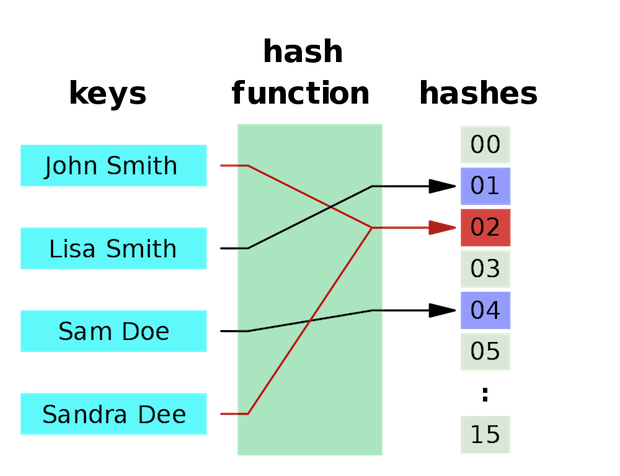
Exemplo de aplicação de busca por Hash em uma tabela de nomes e telefones (Foto: Reprodução/Wikipedia.org )

Na verificação de integridade, aplica-se a função Hash diretamente sobre o dado e salva-se o resumo gerado. Após o dado ser transmitido para o receptor, este calcula o resumo sobre o dado recebido e obtém um novo resumo. Se os resumos forem iguais, assume-se que o dado é igual. Se forem diferentes, recomenda-se um novo download do arquivo. Já para o armazenamento da senha de forma segura, usa-se um procedimento semelhante. No servidor, apenas o resumo da senha do usuário é armazenado. Quando o usuário insere a senha, calcula-se a função Hash da senha e o servidor compara com o resumo armazenado. Se os resumos forem iguais, o usuário é autenticado.

**Propriedades**

Existem diversos tipos de função resumo e a sua complexidade depende, basicamente, das propriedades que se deseja garantir. A propriedade básica de todas as funções resumo é ser unidirecional. No contexto da teoria matemática, significa dizer que a função não tem inversa.

Na prática, ser unidirecional representa que não é possível recuperar o dado original a partir do resumo gerado. Isso ocorre devido ao fato de diversos dados serem mapeados no mesmo resumo. Um exemplo de função resumo é a função resto da divisão. Digamos que utilizamos a função resto pela divisão por 10. Assim, todos os números da sequencia 1, 11, 21, 31, 41, … terão resumo igual a 1. Nesse caso, dado que se tem o resumo 1, não se sabe qual número gerou esse resumo.



Função Hash para mapear nomes em números inteiros que possui colisão no valor 02 (Foto: Reprodução / Wikipedia)

Essa repetição de resumos leva à segunda propriedade de todas as funções Hash: **colisão**. Quando dois dados originais geram o mesmo resumo, tem-se uma colisão. O objetivo principal dos projetistas de função Hash é reduzir ao máximo a probabilidade de ocorrência das colisões. A ferramenta mais usada para reduzir essa probabilidade é o ajuste da distribuição dos resumos. Quanto mais **uniforme** e **dispersa** é a função resumo, menor é a sua probabilidade de colisão.

saiba mais

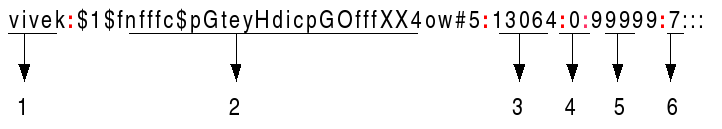
* Código de César com chave de 3 unidades (Foto: Reprodução / Wikipedia) (Foto: Código de César com chave de 3 unidades (Foto: Reprodução / Wikipedia))  
  [**O que é criptografia?**](http://www.techtudo.com.br/artigos/noticia/2012/06/o-que-e-criptografia.html)
* **Topologia da Internet em 2007 (Reprodução / Caida) (Foto: Topologia da Internet em 2007 (Reprodução / Caida))**[**O que é IP?**](http://www.techtudo.com.br/artigos/noticia/2012/05/o-que-e-ip.html)

Dessa forma, busca-se criar funções Hash bastante uniformes. Isso significa buscar que todos os possíveis resultados da função Hash sejam obtidos o mesmo número de vezes se gerados a partir de dados distintos.

Um exemplo de função bastante uniforme é a função resto do exemplo anterior. Cada um dos valores possíveis de resultado da função é gerado pela mesma quantidade de números. No entanto, a função resto não possui boa dispersão dos resumos gerados, ou seja, os resumos de números próximos são próximos. Como os dados a serem resumidos tendem a ter alguma correlação, funções com baixa dispersão aumentam a probabilidade de colisões na aplicação.

No entanto, para serem usadas na prática, as funções resumo precisam seguir uma importante propriedade. Os resultados das funções precisam ser recorrentes, ou seja, sempre que um mesmo dado for avaliado, deve retornar o mesmo resumo. Apenas funções que respeitam essa propriedade podem ser utilizadas para buscas, verificação de integridade e proteção de senhas armazenadas. Isso ocorre, pois, em uma busca por Hash, o resumo do dado salvo deve ser o mesmo resumo do dado buscado. Essa é a única forma de recuperar o dado armazenado.

Obviamente, pode-se aplicar elementos aleatórios ao Hash, mas eles devem ser incluídos no resumo, e a função de verificação deve utilizar esse mesmo elementos aleatório. Um exemplo de uso de elemento aleatório ocorre com o armazenamento das senhas de usuários Linux, as quais são adicionados uma semente aleatório chamada Salt.



Exemplo de linha do arquivo de senhas do Linux. Inclui usuário, hash da senha e alguns outros atributos do usuário (Foto: Reprodução/Cyberciti.biz)

Para garantir a segurança das funções Hash, essas propriedades são importantes, não são suficientes. As funções Hash seguras que conhecemos, como o MD5 e a família SHA (SHA-1, SHA-256 e SHA-512), por exemplo, precisam ter alta dispersão e uniformidade dos resultados. No entanto, também precisam ter uma grande quantidade de resumos possíveis, pois uma forma de ataque que se pode ter é a busca por um outro dado que gere o mesmo resumo.

Esse ataque pode ser usado, por exemplo, para descobrir a senha de um usuário num sistema que tenha tido sua base de dados de resumos de senha comprometidos. Por isso, as funções Hash seguras utilizam resumos não tão pequenos como o resumo da função mostrada na figura acima. Atualmente, as funções Hash seguras usadas geram resultados de 160 a 512 bits, o que representa um mínimo de 1048 possibilidades de resumo.

Para aumentar a segurança das funções de resumo, pode-se adicionar uma chave para a geração do resumo. Assim, apenas quem tiver a chave é capaz de gerar o resumo do dado. Esse método é utilizado na verificação de arquivos transmitidos. Como o resumo do arquivo também é transmitido, um atacante na rede poderia interceptar o dado e o resumo, trocar o dado e inserir o novo resumo no dado. Tal prática faria com que o receptor achasse erradamente que a mensagem não foi alterada. Se ambos o receptor e o emissor combinarem uma senha para a geração do resumo, o atacante não consegue gerar um resumo válido.