1.Hashes

Hashes são usados para verificar e garantir a integridade dos dados. Eles também são usados para verificar a autenticação. O hash é baseado em uma função matemática unilateral que é relativamente fácil de calcular, mas significativamente mais difícil de reverter.

Com funções hash, é computacionalmente inviável que dois conjuntos diferentes de dados apresentem a mesma saída hash. Além disso, o valor do hash muda toda vez que o é mudado ou alterado. Por causa disso, os valores de hash criptográficos são freqüentemente chamados de "impressões digitais"

Eles podem ser usados para detectar arquivos de dados duplicados, alterações de versão de arquivos e aplicativos semelhantes. Esses valores são usados para proteger contra uma alteração acidental ou intencional dos dados ou corrupção acidental dos dados.

1.1 MD5

Desenvolvido por Ron Rivest e usado em uma variedade de aplicações de internet, MD5 é uma função unidirecional que produz uma mensagem hash de 128 bits. MD5 é considerado um algoritmo legado e deve ser evitado e usado apenas quando não houver alternativas melhores disponíveis. Recomenda- se que SHA-2 ou SHA-3 sejam usados em vez disso.

1.2 SHA-1

Desenvolvido pela U.S. National Security Agency (NSA) em 1995. É muito semelhante às funções hash MD5. Existem várias versões. O SHA-1 cria uma mensagem de 160 bits e é um pouco mais lento que o MD5. O SHA-1 possui falhas e é um algoritmo antigo.

1.3 SHA-2

Desenvolvido pela NSA. Inclui SHA-224 (224 bits), SHA-256 (256 bits), SHA-384 (384 bits) e SHA-512 (512 bits). Se você estiver usando SHA-2, então os algoritmos SHA-256, SHA-384 e SHA-512 devem ser usados sempre que possível.

1.4 SHA-3

SHA-3 é o algoritmo de hashing mais recente e foi apresentado pelo Instituto Nacional de Padrões e Tecnologia (NIST) como uma alternativa e substituição eventual para a família SHA-2 de algoritmos de hash. SHA-3 inclui SHA3-224 (224 bits), SHA3-256 (256 bits), SHA3-384 (384 bits) e SHA3-512 (512 bits). A família SHA-3 são algoritmos de última geração e devem ser usados sempre que possível.

2.Colisões em funções hash

Colisões em funções hash ocorrem quando dois conjuntos de dados distintos produzem o mesmo valor de hash. Em outras palavras, é quando duas entradas diferentes são mapeadas para o mesmo resultado de hash. Essa situação é indesejável, pois uma das principais propriedades de uma função hash é a unicidade, ou seja, cada conjunto de dados de entrada deve gerar um valor de hash único. As colisões podem ser classificadas em duas categorias: colisões acidentais e colisões intencionais.

2.1 Colisões acidentais

São ocorrências não planejadas e imprevisíveis, resultado de características das funções hash e do espaço limitado de valores de hash. Embora seja extremamente improvável que uma função hash perfeitamente distribuída e de alta qualidade apresente colisões acidentais, é possível que, em casos reais, elas ocorram devido à natureza probabilística das funções hash.

2.2 Colisões intencionais

São uma preocupação maior, especialmente em contextos de segurança. São o resultado de ataques deliberados, nos quais um adversário procura encontrar duas entradas diferentes que produzem o mesmo valor de hash. O objetivo de um ataque de colisão intencional pode ser comprometer a integridade dos dados ou até mesmo encontrar uma fraqueza na função hash.

As colisões são indesejáveis porque podem levar a problemas de segurança e confiabilidade. Em aplicações como criptografia, assinatura digital e autenticação, as colisões podem ser exploradas para falsificar informações ou comprometer a integridade dos dados.

Portanto, é fundamental utilizar funções hash que apresentem uma resistência adequada a colisões, dificultando a ocorrência tanto de colisões acidentais quanto de ataques de colisão intencionais.

Para garantir a segurança e a confiabilidade das funções hash, os algoritmos utilizados devem ser cuidadosamente projetados e analisados para minimizar as chances de colisões e resistir a tentativas de encontrar colisões intencionais.

3.Assinatura digital

As assinaturas digitais são uma técnica matemática usada para fornecer autenticidade, integridade e não repúdio. As assinaturas digitais têm propriedades específicas que permitem autenticação de entidade e integridade de dados. Além disso, as assinaturas digitais fornecem não repúdio da transação. Em outras palavras, a assinatura digital serve como prova legal de que o intercâmbio de dados ocorreu. As assinaturas digitais são comumente usadas em a**ssinaturas de códigos e certificados digitais**.

Assinar digitalmente o código fornece várias garantias sobre o código como garantir que o código é autêntico e é realmente originado pela editora; O código não foi modificado desde que saiu do editor do software e a editora publicou inegavelmente o código. Isso fornece não repúdio do ato de publicação.

3.1 Algoritmos DSS (*Digital Signature Standard*)

3.1.1 DSA

DSA é o padrão original para gerar pares de chaves públicas e privadas e para gerar e verificar assinaturas digitais.

3.1.2 RSA

RSA é um algoritmo assimétrico que é comumente usado para gerar e verificar assinaturas digitais

3.1.3 ECDSA

O ECDSA é uma variante mais recente do DSA e fornece autenticação de assinatura digital e não repúdio com os benefícios adicionais da eficiência computacional, tamanhos de assinatura pequenos e largura de banda mínima.