FUNÇÃO HASH CRIPTOGRAFADA (MD5, E A FAMÍLIA SHA)

Muriel Mazzetto

Função Hash Criptográfica

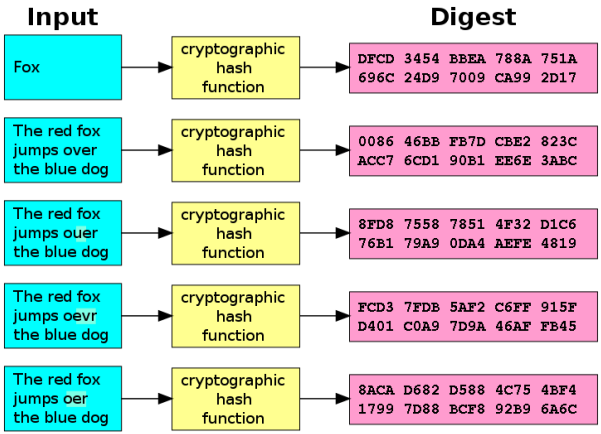
Uma **função de dispersão criptográfica** ou **função hash criptográfica** é uma função hash considerada praticamente impossível de inverter, isto é, de recriar o valor de entrada utilizando somente o valor de dispersão.

Uma função de dispersão criptográfica deve possuir quatro propriedades principais:

- deve ser fácil computar o valor de dispersão para qualquer mensagem
- deve ser difícil gerar uma mensagem a partir de seu resumo
- deve ser difícil modificar a mensagem sem modificar o seu resumo
- deve ser difícil encontrar duas mensagens diferentes com o mesmo resumo.

Função Hash Criptográfica

- Possuem várias aplicações em segurança da informação, principalmente em **assinatura digital**, **código de autenticação de mensagem** (MACs), e outras formas de autenticação.
- Elas também podem ser utilizadas como funções hash, para indexar dados em tabelas hash, para impressão digital, para detectar dados duplicados ou identificar arquivos únicos, e como **checksum** para detectar corrupção de dados acidental.
- No contexto da segurança da informação, valores de dispersão criptográficos são às vezes conhecidos como **impressão digital**.

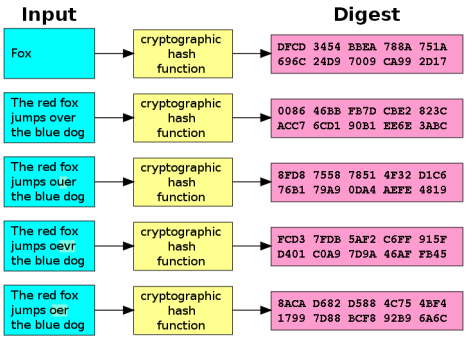


Propriedades

- Resistência à pré-imagem: Dado um valor hash h deve ser difícil encontrar qualquer mensagem m tal que h = hash(m). Este conceito está relacionado ao da **função de mão única** (ou função unidirecional). Funções que não possuem essa propriedade estão vulneráveis a ataques de pré-imagem.

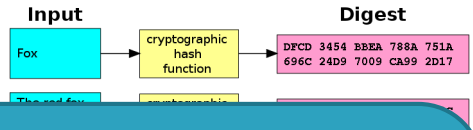
- Resistência à segunda pré-imagem: Dada uma entrada m1 deve ser difícil encontrar outra entrada m2 tal que hash(m1) = hash(m2). Funções que não possuem essa propriedade estão vulneráveis a ataques de segunda pré-imagem. **Duas mensagens com mesmo hash**.

- Resistência à colisão: Deve ser difícil encontrar duas mensagens diferentes m1 e m2 tal que hash(m1) = hash(m2). Tal par é chamado de colisão hash criptográfica. Essa propriedade também é conhecida como forte resistência à colisão. Ela requer um valor hash com pelo menos o dobro do comprimento necessário para resistência à pré-imagem; caso contrário, colisões podem ser encontradas através de um ataque do aniversário.

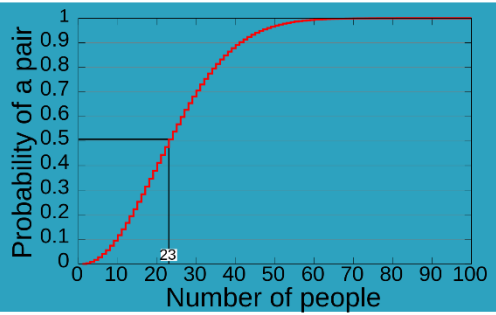


Propriedades

- Resistência à pré-imagem: Dado um valor hash h deve ser difícil encontrar qualquer mensagem m tal que h = hash(m). Este conceito está relacionado ao da **função de mão única** (ou função unidirecional). Funções que não possuem essa propriedade estão vulneráveis a ataques de pré-imagem.



Ataque baseado no paradoxo do aniversário e problema da casa dos pombos.
Propriedade de probabilidade para se encontrar 2 seres distintos para o mesmo aniversário/casa.



menos o dobro do [illegible]ssário para resistência à pré-imagem; caso contrário, colisões podem ser encontradas através de um ataque do aniversário.

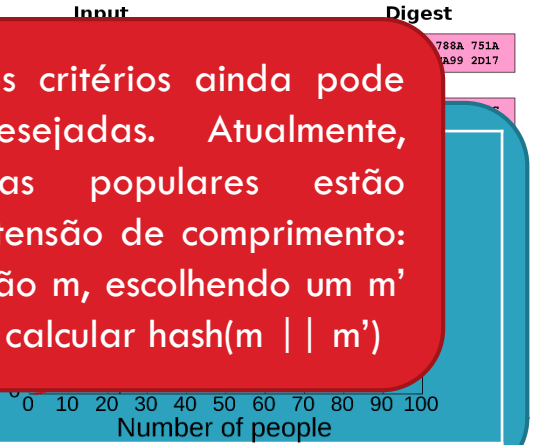
Propriedades

- Resistência à pré-imagem: Dado um valor hash h deve ser difícil encontrar qualquer mensagem m tal que h = hash(m). Este conceito es
(ou função un
propriedade

Uma função com todos esses critérios ainda pode possuir propriedades indesejadas. Atualmente, funções hash criptográficas populares estão vulneráveis a ataques de extensão de comprimento: dado hash(m) e len(m) mas não m, escolhendo um m' adequado um atacante pode calcular hash(m || m')

Ataque
aniversá

Propried
se encon
mesmo aniversário/casa.



menos o dobro do [illegible]ssário para resistência à pré-imagem; caso contrário, colisões podem ser encontradas através de um ataque do aniversário.

Aplicações

- Verificação da integridade de arquivos ou mensagens: informar códigos de hash para comparar se não houve alteração no arquivo.
- Verificação de senha: armazenar o hash de senhas. Ainda não muito adequado. (Salted Hash).
- Prova-de-trabalho: ao ser solicitado para realizar um trabalho, primeiro pede que o solicitante resolva um hash que consuma tempo de execução razoável, para minimizar problemas com DoS.
- Geradores pseudo-aleatórios e derivação de chave: a partir de uma chave segura, gerar chaves com o hash.

Algoritmos Hash Criptografados

Mesmo que uma função hash nunca tenha sido quebrada, um ataque bem sucedido quanto uma variante mais fraca, pode enfraquecer a confiança de profissionais da área e levar a seu abandono.

Em Agosto de 2004, foram descobertas fraquezas em um número de funções hash bastante populares na época, incluindo SHA-0, RIPEMD, e MD5. Nem SHA-0 nem RIPEMD costumam ser utilizadas desde que foram substituídas por suas versões mais fortes.

Os mais famosos são: MD5, SHA-1, SHA-2 e o mais novo SHA-3.

MD5

O MD5 (Message-Digest algorithm 5) é uma função de dispersão criptográfica de 128 bits unidirecional desenvolvido pela RSA Data Security, muito utilizado por softwares com protocolo ponto-a-ponto na verificação de integridade de arquivos e logins.

Foi desenvolvido em 1991 para suceder ao MD4 que tinha alguns problemas de segurança. Por ser um algoritmo unidirecional, uma hash md5 não pode ser transformada novamente no texto que lhe deu origem.

Em 2008, Ronald Rivest e outros, publicaram uma nova versão do algoritmo o MD6 com hash de tamanhos 224, 256, 384 ou 512 bits. O algoritmo MD6 iria participar do concurso para ser o novo algoritmo SHA-3, porém logo depois removeu-o do concurso por considerá-lo muito lento, anunciando que os computadores de hoje são muito lentos para usar o MD6.

Formalização e pseudo código fornecido pela RFC 132.

MD5 - Vulnerabilidade

Como o MD5 faz apenas uma passagem sobre os dados, se dois prefixos com o mesmo hash forem construídos, um sufixo comum pode ser adicionado a ambos para tornar uma colisão mais provável. Deste modo é possível que duas strings diferentes produzam o mesmo hash. O que não garante que a partir de uma senha codificada em hash específica consiga-se a senha original, mas permite uma possibilidade de descobrir algumas senhas a partir da comparação de um conjunto grande de hash de senhas através do método de comparação de dicionários.

Salting: Para aumentar a segurança em alguns sistemas, usa-se a tática de adicionar um texto fixo no texto original a ser codificado. Deste modo se o sal for "wiki" e a senha for "1234", a pseudo-senha poderá ser "wiki1234" e assim mesmo que alguém tenha o MD5 de 1234 por ser uma senha comum ele não terá de wiki1234. Porém caso o "sal" seja simples como no exemplo e houver o MD5 de "wiki1234" é possível descobrir o sal e deste modo descodificar as senhas mais comuns. Por este motivo geralmente o "sal" é algo complexo.

SHA

Consistem na família de algoritmos de hash criptografados denominados de Secure Hash Algorithm, publicados pela National Institute of Standards and Technology (NIST).

Fazem parte da família SHA:

- SHA-0;
- SHA-1;
- SHA-2;
- SHA-3.

SHA-0

Baseado em métodos similares ao MD5.

Em 12 de Agosto de 2004, uma colisão para o SHA-0 completo foi anunciada por Joux, Carribault, Lemuet e Jalby. Isso foi feito utilizando-se uma generalização do ataque de Chabaud e Joux. Encontrar uma colisão tinha complexidade de 251 e levava aproximadamente 80.000 horas de CPU em um supercomputador com 256 processadores Itanium 2 (equivalente a 13 dias de uso completo do computador).

Em luz a esses resultados para SHA-0, alguns especialistas sugeriram planos para o uso de SHA-1 em novos criptossistemas.

SHA-1

Projetado pela Agencia Nacional de Segurança (NSA) dos Estados Unidos é outro padrão publicado pela NIST.

Produz um valor de dispersão de 160 bits (20 bytes) conhecido como resumo da mensagem. Um valor de dispersão SHA-1 é normalmente tratado como um número hexadecimal de 40 dígitos.

Publicada em 1995, SHA-1 é muito similar à SHA-0, mas altera a especificação de dispersão do SHA original para corrigir as fraquezas alegadas. No entanto, a NSA não forneceu nenhuma explicação mais aprofundada nem identificou exatamente qual era a falha que foi corrigida.

SHA-1

SHA-1 faz parte de várias aplicações e protocolos de segurança amplamente utilizados, incluindo TLS e SSL. O hashing de SHA-1 também é utilizado em sistemas de controle de revisão distribuídos como Git, para identificar revisões, assim como detectar corrupção ou adulteração de dados.

Criptógrafos produziram pares de colisão para SHA-0 e encontraram algoritmos que devem produzir colisões em SHA-1 em muito menos avaliações que as esperadas 2^{80}.

Em 2015, um paper foi publicado sobre um ataque eficiente de colisão para o SHA-1, com descrição no site The Shappening.

Em novembro de 2013, a Google anunciou sua política de rebaixamento sobre o SHA-1, de acordo com a qual o Chrome irá deixar de aceitar certificados SHA-1 em SSL de forma gradual em 2017. Mozilla também pretende parar de aceitar certificados baseados em SSL até 2017. Ao resultado obtido em 2015, pede-se a aceleração do processo de extinção.

SHA-2

SHA-2 foi publicada em 2001 pelo NIST como um padrão federal dos Estados Unidos (FIPS). A família SHA-2 de algoritmos está patenteada sob uma licença livre de royalties.

Apesar de ataques bem sucedidos sobre o SHA-2 nunca terem sido relatados, ele é algoritmicamente similar ao SHA-1.

A família SHA-2 é composta por seis funções hash com resumos (valores de hash) que são de 224, 256, 384 ou 512 bits: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256.

SHA-2

A função hash SHA-2 é implementada em algumas aplicações de segurança e protocolos amplamente usados, incluindo TLS e SSL.

SHA-256 é usado como parte do processo de autenticação de pacotes de software Debian GNU/Linux e no padrão DKIM de assinatura de mensagens.

SHA-512 é parte de um sistema para autenticar vídeos de arquivos do Tribunal Penal Internacional para o genocídio de Ruanda.

SHA-256 e SHA-512 foram propostos para utilização no DNSSEC.

Várias criptomoedas como Bitcoin usam SHA-256 para verificar transações e calculam a prova-de-trabalho.

SHA-2

SHA-256 e SHA-512 são funções hash inovadoras computadas com palavras de 32 bits e 64 bits, respectivamente. Eles usam quantidades de deslocamento e constantes aditivas diferentes, mas as suas estruturas são praticamente idênticas, diferindo apenas no número de rodadas.

SHA-224 e SHA-384 são simplesmente versões truncadas das duas primeiras, calculadas com valores iniciais diferentes.

SHA-512/224 e SHA-512/256 também são versões truncadas de SHA-512, mas os valores iniciais são gerados usando o método descrito no FIPS PUB 180-4.

SHA-3

Em 2 de Novembro de 2007 o NIST anunciou uma competição pública para projetar um novo algoritmo de hash que substituísse os algoritmos SHA-1 e SHA-2, em 2012. O novo algoritmo é denominado SHA-3. Esta competição é uma resposta aos avanços em criptoanálise do algoritmo SHA-1.

- Para a primeira Rodada o NIST recebeu, em outubro de 2008, 64 propostas de funções de hash e após, em dezembro de 2008, só 51 foram aceitas.
- Na segunda rodada, em julho de 2009, somente 14 propostas de funções de hash ficaram na competição: BLAKE, Blue Midnight Wish, Cbe-Hash, ECHO, Fugue, Grostl, Hamsi, JH, Keccak, Luffa, Shabal, Shavite-3, SIM e Skein.
- Em dezembro de 2010 foram publicados pela NIST as 5 funções finalistas para a terceira e última rodada da competição, que são: BLAKE, Grøstl, JH, Keccak e Skein.
- Em 2 de outubro de 2012, o algoritmo **Keccak**, foi declarado vencedor da competição.

SHA-3

O SHA 3 foi liberado como padrão em 5 de agosto de 2015, com patente de domínio público.

Diferente de todos os outros algoritmos, baseados no MD4, sempre exibindo características similares, como MD5, SHA-0, SHA-1 e SHA-2, o SHA-3 é um formato que não deriva do SHA-2.

SHA-3 possui variantes, assim como o SHA-2. Duas delas, SHAKE128 e SAHKE256, oferecem a função de saída de tamanho variável.

SHA-3

O SHA 3 foi liberado como padrão em 5 de agosto de 2015, com patente de domínio público.

Instance	Definition
SHA3-224(*M*)	Keccak[448](*M*‖01, 224)
SHA3-256(*M*)	Keccak[512](*M*‖01, 256)
SHA3-384(*M*)	Keccak[768](*M*‖01, 384)
SHA3-512(*M*)	Keccak[1024](*M*‖01, 512)
SHAKE128(*M*, *d*)	Keccak[256](*M*‖1111, *d*)
SHAKE256(*M*, *d*)	Keccak[512](*M*‖1111, *d*)

Diferente de eados no MD4, sempre exibi mo MD5, SHA-0, SHA-1 e S que não deriva do SHA-2.

SHA-3 possu -2. Duas delas, SHAKE128 e SAHKE256, oferecem a função de saída de tamanho variável.

Comparação

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Rounds
MD5 (as reference)		128	128 (4 × 32)	512	Unlimited[3]	64
SHA-0		160	160 (5 × 32)	512	$2^{64} - 1$	80
SHA-1		160	160 (5 × 32)	512	$2^{64} - 1$	80
SHA-2	*SHA-224* *SHA-256*	224 256	256 (8 × 32)	512	$2^{64} - 1$	64
	SHA-384 *SHA-512* *SHA-512/224* *SHA-512/256*	384 512 224 256	512 (8 × 64)	1024	$2^{128} - 1$	80
SHA-3	*SHA3-224* *SHA3-256* *SHA3-384* *SHA3-512*	224 256 384 512	1600 (5 × 5 × 64)	1152 1088 832 576	Unlimited[5]	24[6]
	SHAKE128 *SHAKE256*	*d* (arbitrary) *d* (arbitrary)		1344 1088		

Comparação

Algorithm and variant		Operations	Security (bits)	Example performance[2] (MiB/s)
MD5 (as reference)		And, Xor, Rot, Add (mod 2^{32}), Or	<64 (collisions found)	335
SHA-0		And, Xor, Rot, Add (mod 2^{32}), Or	<80 (collisions found)	-
SHA-1			<80 (theoretical attack[4])	192
SHA-2	*SHA-224* *SHA-256*	And, Xor, Rot, Add (mod 2^{32}), Or, Shr	112 128	139
	SHA-384 *SHA-512* *SHA-512/224* *SHA-512/256*	And, Xor, Rot, Add (mod 2^{64}), Or, Shr	192 256 112 128	154
SHA-3	*SHA3-224* *SHA3-256* *SHA3-384* *SHA3-512*	And, Xor, Rot, Not	112 128 192 256	-
	SHAKE128 *SHAKE256*		min(*d*/2, 128) min(*d*/2, 256)	-

Vulnerabilidade

- Para o MD5 e SHA-1, além da vulnerabilidade por colisão, outro mecanismo já causava problemas para sistemas de senhas.
- Banco de dados com vários Hashs simples calculados pelo padrão, e armazenados para buscas. Tendo o hash é possível encontrar seu significado nesses sites.
- Foi o início do uso de “Salted Hash”.

Referências

- https://pt.wikipedia.org/wiki/Fun%C3%A7%C3%A3o_hash_criptogr%C3%A1fica
- https://pt.wikipedia.org/wiki/Paradoxo_do_anivers%C3%A1rio
- https://pt.wikipedia.org/wiki/MD5
- https://tools.ietf.org/html/rfc1321
- https://en.wikipedia.org/wiki/Secure_Hash_Algorithm
- https://pt.wikipedia.org/wiki/SHA-1

Referências

- https://it.slashdot.org/story/15/10/09/1425207/first-successful-collision-attack-on-the-sha-1-hashing-algorithm
- https://sites.google.com/site/itstheshappening/
- https://pt.wikipedia.org/wiki/SHA-2
- http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf
- https://pt.wikipedia.org/wiki/SHA-3
- https://en.wikipedia.org/wiki/SHA-3
- http://www.asecuritysite.com/encryption/hashing