Contramedidas

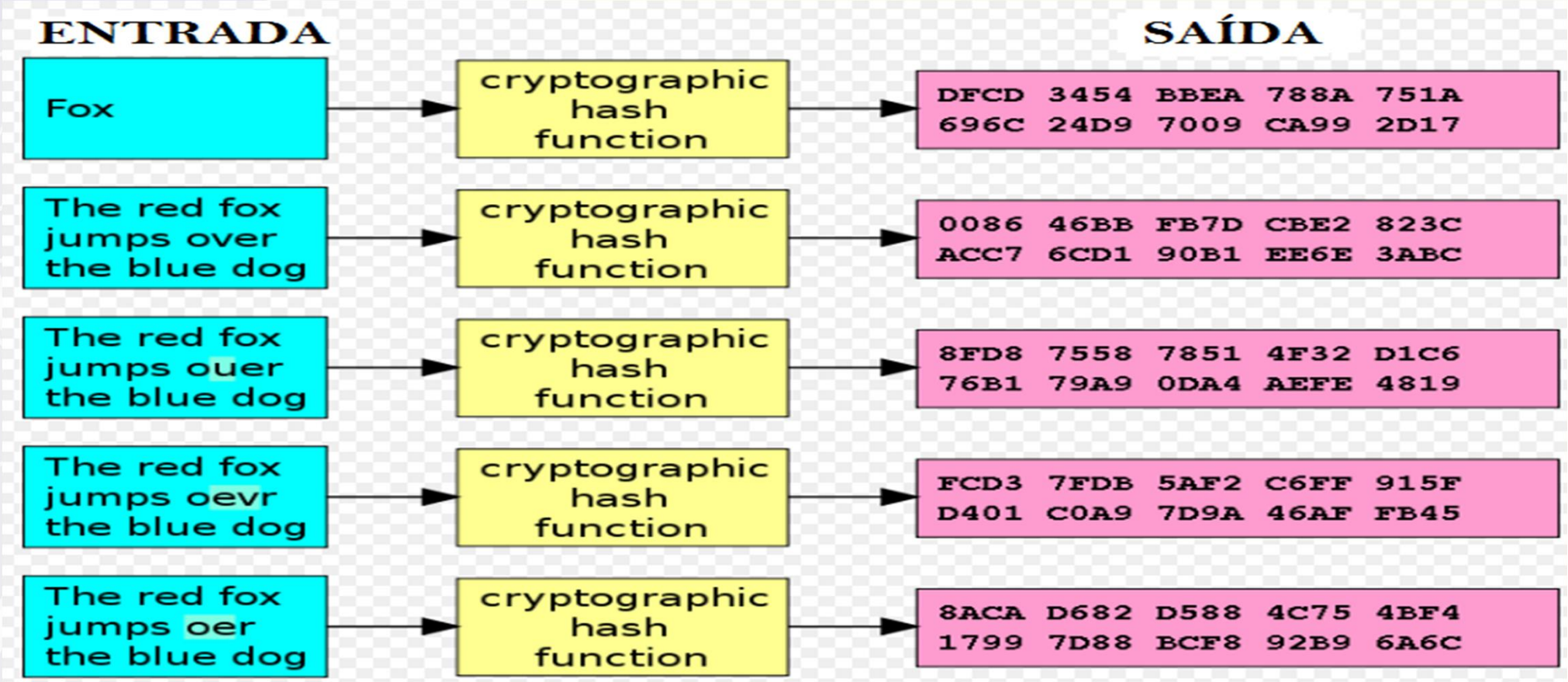
Segurança Lógica - ALGORITMOS HASH

Uma **algoritmo de hash** é uma função matemática que mapeia dados de entrada de tamanho arbitrário para uma saída de tamanho fixo, geralmente representada por uma sequência de caracteres.

Ela é frequentemente usada para garantir a **integridade** dos dados, permitindo que os destinatários verifiquem s e os dados foram alterados ou corrompidos durante a transmissão.

Contramedidas

Segurança Lógica - ALGORITMOS HASH



Segurança Lógica - ALGORITMOS HASH

PRINCIPAIS

ALGORITMOS

DE HASH

Contramedidas

Segurança Lógica - ALGORITMOS HASH

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.

Contramedidas

Segurança Lógica - ALGORITMOS HASH

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.

Segurança Lógica - ALGORITMOS HASH

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.

Segurança Lógica - ALGORITMOS HASH

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.

Contramedidas

Segurança Lógica - ALGORITMOS HASH

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.

Contramedidas

Segurança Lógica - ALGORITMOS HASH

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.

Contramedidas

Segurança Lógica - ALGORITMOS HASH

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*	224	256 (8 × 32)	512	$2^{64} - 1$	64
	SHA-256	256				
	SHA-384	384	512 (8 × 64)	1024	$2^{128} - 1$	80
	SHA-512	512				
	SHA-512/224	224				
	SHA-512/256	256				
SHA-3	*SHA3-224*	224	1600 (5 × 5 × 64)	1152	Unlimited[5]	24[6]
	SHA3-256	256		1088		
	SHA3-384	384		832		
	SHA3-512	512		576		
	SHAKE128	*d* (arbitrary)		1344		
	SHAKE256	*d* (arbitrary)		1088		

Contramedidas

Segurança Lógica - ALGORITMOS HASH

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)	-

Contramedidas

Segurança Lógica - ALGORITMOS HASH

Algoritmo de hash	Tamanho hash	Kbytes/s
Abreast Bavies-Meyer (c/IDEA)	128	22
Davies-Meyer (c/DES)	64	9
GOST-Hash	256	11
NAVAL (3 passos)	Variável	168
NAVAL (4 passos)	Variável	118
NAVAL (5 passos)	Variável	95
MD4 – Message Digest 4	128	236
MD5 – Message Digest 5	128	174
N-HASH (12 rounds)	128	29
N-HASH (15 rounds)	128	24
RIPE-MD	128	182
RIPE-MD-160	160	--
SHA – Secure Hash Algorithm	160	75
SNEFRU (4 passos)	128	48
SNEFRU (8 passos)	128	23

Fonte: GUELFI (2002)

Contramedidas

Segurança Lógica - ALGORITMOS HASH

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 determinados sites.
- Foi o início do uso de “Salted Hash”.

Segurança Lógica

GERANDO HASHES EM PYTHON:

```
import hashlib

frase = input("Digite uma frase: ")

hash_md5 = hashlib.md5(str(frase).encode('utf-8'))
hash_256 = hashlib.sha256(str(frase).encode('utf-8'))
hash_512 = hashlib.sha512(str(frase).encode('utf-8'))

print('Hash MD5      : ', hash_md5.hexdigest(), "Tamanho: ", len(hash_md5.hexdigest()))
print('Hash SHA256: ', hash_256.hexdigest(), "Tamanho: ", len(hash_256.hexdigest()))
print('Hash SHA512: ', hash_512.hexdigest(), "Tamanho: ", len(hash_512.hexdigest()))
```

Contramedidas

Segurança Lógica - ALGORITMOS HASH

Roteiro de Prática: HASH

https://drive.google.com/file/d/1x3mZitNGjk3_GyQpVQlZUDAtB5xSHL3z/view?usp=sharing