METODE CRIPTOGRAFICE DE PROTECȚIE A INFORMAȚIEI

Tema 5: Funcții hash criptografice

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA

La 31 ianuarie 1992, NIST (National Institute for Standards and Technology, USA) a publicat în Registrul Federal standardul SHS (Secure Hash Standard), care a fost revăzut de mai multe ori, ultima versiune fiind cea din 2015 [10]. Acest standard conține descrierea algoritmului SHA-1 (Secure Hash Algorithm) și a familiei de algoritmi SHA-2. Aceste funcții hash sunt utilizate în cadrul schemelor de semnătură digitală DSA (Digital Signature Algorithm) propuse în cadrul standardului DSS (Digital Signature Standard) [11]. Algoritmii SHA reprezintă variante cu securitate sporită ale lui MD4.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

Algoritmul SHA-1

Principalele diferențe între SHA-1 și MD4 sunt următoarele:

1. Valoarea hash generată de SHA-1 este de 160 biți, iar variabila de legătură constă din 5 cuvinte pe 32 biți.
2. Funcția de compresie conține 4 runde (ca și în MD5) în raport cu cele 3 ale lui MD4. Funcțiile logice f, g, h ale lui MD4 sunt folosite după cum urmează: f - în prima rundă; g - în runda trei; h - în rundele doi și patru. Fiecare rundă conține 20 de pași în loc de 16. Amplificarea numărului de pași per rundă permite ca fiecare cuvânt al variabilei de legătură să fie supus transformării de 4 ori per rundă.
3. Fiecare bloc format din 16 cuvinte ale mesajului este extins la un bloc de 80 cuvinte, printr-un procedeu în baza căruia cuvântul cu numărul 17,...,80 este obținut, folosind operația XOR (sau exclusiv) a 4 cuvinte de pe pozițiile precedente ale blocului extins. Cele 80 de cuvinte formează intrarea pentru cei 80 de pași, câte un cuvânt la un pas.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

4. Modificările efectuate la nivel de operații: rotația ciclică utilizată este una constantă pe 5 biți; variabila de legătură constă din 5 cuvinte; cuvintele blocului de mesaj extins sunt accesate secvențial; cuvântul C este reînnoit în baza lui B, care este rotit ciclic la stânga cu 30 de biți.
5. SHA-1 folosește 4 constante aditive nenule, pe când MD4 – 3 constante, dintre care doar două sunt nenule.
6. Ordonarea octeților folosită pentru conversia dintre șirurile de octeți și cuvintele pe 32 biți, este big endian, pe când în MD4 este little endian.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

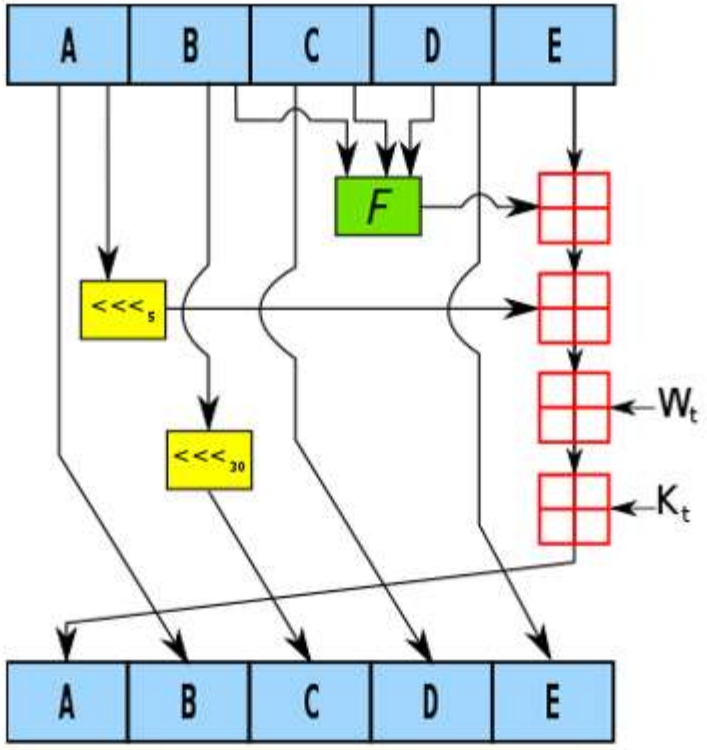

Figura 1.3.8. O iterație (rundă) în cadrul funcției de compresie SHA-1

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

Algoritmul SHA-1

Date de intrare:

x - *șirul de biți de lungime arbitrară* $l \geq 0$, *pentru care se calculează valoarea hash*

Date de ieșire:

$h(x)$ - valoarea hash pe 160 biți a lui x

Pașii algoritmului:

1. *Definirea constantelor*

Se definesc 5 cuvinte pe 32 biți, $h_i, i=\overline{1,5}$, în reprezentare hexazecimală, care definesc variabila de legătură inițială IV:

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

$$h_1 = 0x67452301,\ h_2 = 0xefcdab89,\ h_3 = 0x98badcfe,\ h_4 = 0x10325476,\ h_5 = 0xc3d2e1f0\ .$$

Se definesc constantele aditive pe 32 biți (în reprezentare hexazecimală), aceleași pentru fiecare rundă:

$$y_1 = 0x5a827999,\ y_2 = 0x6ed9eba1,\ y_3 = 0x8f1bbcdc,\ y_4 = 0xca62c1d6\ .$$

Nu este necesară specificarea ordinii de accesare a cuvintelor blocului de mesaj și nici specificarea numărului de biți în cadrul operației de rotație ciclică la stânga.

2. *Preprocesare*

Ca și în MD-4 se aplică procedura de completare și cea de MD-fortificare. Șirul de intrare constă din $16m$ cuvinte pe 32 biți: $x_0 x_1 \ldots x_{16m-1}$. Se inițializează variabilele de legătură:

$$H_1 := h_1, H_2 := h_2, H_3 := h_3, H_4 := h_4, H_5 := h_5\ .$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

3. Procesarea

Pentru fiecare $i=\overline{0,m-1}$ se copiază blocul i de 16 cuvinte pe 32 biți în locația temporară $X_j := x_{16i+j}$, $j=\overline{0,15}$, după care aceasta se procesează în cadrul a patru runde a câte 20 pași înainte de a reînnoi variabilele de legătură:

Se extinde blocul de 16 cuvinte într-un bloc de 80 cuvinte:

Pentru $j=\overline{16,79}$ execută

$$X_j := rol\left(X_{j-3} \oplus X_{j-8} \oplus X_{j-14} \oplus X_{j-16}, 1\right).$$

Se inițializează variabila de legătură: $A := H_1, B := H_2, C := H_3, D := H_4, E := H_5$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

Runda 1. (Adunările se efectuează modulo 2^{32})

Pentru $j = \overline{0,19}$ execută

$t := rol(A,5) + f(B,C,D) + E + X_j + y_1;\ E := D;\ D := C;\ C := rol(B,30);\ B := A,\ A := t$.

Runda 2.

Pentru $j = \overline{20,39}$ execută

$t := rol(A,5) + h(B,C,D) + E + X_j + y_2;\ E := D;\ D := C;\ C := rol(B,30);\ B := A,\ A := t$.

Runda 3.

Pentru $j = \overline{40,59}$ execută

$t := rol(A,5) + g(B,C,D) + E + X_j + y_3;\ E := D;\ D := C;\ C := rol(B,30);\ B := A,\ A := t$.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

Runda 4.

Pentru $j = \overline{60,79}$ execută

$t := rol(A,5) + h(B,C,D) + E + X_j + y_4;\ E := D;\ D := C;\ C := rol(B,30);\ B := A,\ A := t$.

Se reînnoiesc variabilele de legătură:

$$H_1 := H_1 + A, H_2 := H_2 + B, H_3 := H_3 + C, H_4 := H_4 + D, H_5 := H_5 + E .$$

4. Completarea

Valoarea hash $h(x)$ a mesajului inițial x este $H_1 \| H_2 \| H_3 \| H_4 \| H_5$.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

În baza notațiilor următoare:

$$f(i,u,v,w) := f(u,v,w),\ i = \overline{0,19},$$
$$f(i,u,v,w) := h(u,v,w),\ i = \overline{20,39},$$
$$f(i,u,v,w) := g(u,v,w),\ i = \overline{40,59},$$
$$f(i,u,v,w) := h(u,v,w),\ i = \overline{60,79},$$

$$X_{ij},\ i = \overline{0,m-1},\ j = \overline{0,79},$$

$$y_i = 0x5a827999,\ j = \overline{0,19}, \qquad y_i = 0x6ed9eba1,\ j = \overline{20,39},$$
$$y_i = 0x8f1bbcdc,\ j = \overline{40,59}, \qquad y_i = 0xca62c1d6,\ j = \overline{60,79}$$

atunci etapa de procesare a algoritmului SHA-1 se poate scrie sub forma:

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

$A := H_1, B := H_2, C := H_3, D := H_4, E := H_5$

Pentru $i = \overline{0, m-1}$ execută

{

Pentru $j = \overline{0, 79}$ execută

{

dacă $j > 15$ atunci

{

$X_{ij} := X_{i,j-3} \oplus X_{i,j-8} \oplus X_{i,j-14} \oplus X_{i,j-16}$

}

$t := rol(A,5) + f(j,B,C,D) + E + X_{i,j} + y_j;\ E := D;\ D := C;\ C := rol(B,30);\ B := A;\ A := t$

}

$H_1 := H_1 + A, H_2 := H_2 + B, H_3 := H_3 + C, H_4 := H_4 + D, H_5 := H_5 + E$

}

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. ALGORITMUL SHA-1

Tabel cu valori hash (în reprezentare hexazecimală) pentru testarea algoritmului:

"The quick brown fox jumps over the lazy dog"	2fd4e1c67a2d28fced849ee1bb76e7391b93eb12
„The quick brown fox jumps over the lazy cog"	de9f2c7fd25e1b3afad3e85a0bd17d9b100db4b3
„abc"	a9993e364706816aba3e25717850c26c9cd0d89d
„ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	761c457bf73b14d27e9e9265c46f4b4dda11f940
„" (mesajul vid)	da39a3ee5e6b4b0d3255bfef95601890afd80709

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Descrierea pentru algoritmii din familia SHA-2 ce urmează este bazată pe standardul FIPS PUB 180-4 din august 2015 [10]. Standardul dă o specificare pentru următorii algoritmi de calcul al valorilor hash: SHA-1, SHA-224, SHA- 256, SHA – 384, SHA – 512, SHA – 512/224 și SHA – 512/256. Toți algoritmii descriși sunt algoritmi iterativi și reprezintă funcții hash unidirecționale pentru care este computațional dificil

1) de găsit un mesaj ce reprezintă preimaginea unei valori hash date;

2) de găsit două mesaje diferite care să producă aceeași valoare hash.

Orice modificare a mesajului va conduce cu o probabilitate foarte înaltă la generarea unei alte valori hash (astfel se verifică integritatea mesajului). Această proprietate este utilă la generarea și verificarea semnăturilor digitale, a codurilor de autentificare a mesajelor, precum și la generarea numerelor aleatoare.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Fiecare algoritm poate fi descris în două etape: preprocesare și calculul valorii hash. Preprocesarea implică completarea mesajului (procedura de padding), divizarea mesajului completat pe blocuri de m biți și setarea valorilor de inițializare. Valoarea hash este generată iterativ.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Algoritmii diferă prin dimensiunile blocurilor și a cuvintelor utilizate pe parcursul algoritmului, dar și prin dimensiunea valorilor hash (a se vedea Tabelul 1.3.1).

Algoritmul	Lungimea mesajului (în biți)	Lungimea blocului (în biți)	Lungimea cuvântului (în biți)	Lungimea valorii hash (în biți)
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

Tabelul 1.3.1. Parametrii ce caracterizează algoritmii SHA

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Cuvântul este un șir pe w biți care poate fi reprezentat ca și un șir de numere în hexazecimal. Pentru a converti un cuvânt în hexazecimal, fiecare șir de 4 biți este convertit în echivalentul său în hexazecimal. În SHA exprimarea cuvintelor pe 32 sau 64 de biți folosește convenția big-endian, astfel în cadrul fiecărui cuvânt cel mai semnificativ bit este amplasat la stânga șirului.

Un număr întreg se poate reprezenta ca un cuvânt sau o pereche de cuvinte. Un întreg din intervalul $\left[0, 2^{32}-1\right]$ poate fi reprezentat ca un cuvânt pe 32 biți. Cei mai puțin semnificativi 4 biți ai numărului întreg sunt reprezentați de către numărul hexazecimal cel mai din dreapta din reprezentarea în hexazecimal a cuvântului.

Numărul întreg din intervalul $\left[0, 2^{64}-1\right]$ poate fi reprezentat ca un cuvânt pe 64 biți.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Numărul întreg din intervalul $\left[0, 2^{64}-1\right]$ poate fi reprezentat ca un cuvânt pe 64 biți.

Dacă Z este un număr întreg, $0 \le Z < 2^{64}$, atunci $Z = 2^{32}X + Y$, unde $0 \le X < 2^{32}$ și $0 \le Y < 2^{32}$. Deoarece X și Y pot fi reprezentate ca și cuvinte pe 32 biți, x și y respectiv, întregul Z poate reprezentat ca și o pereche de cuvinte (x, y). Această proprietate este utilizată în algoritmii SHA-1, SHA-224, SHA-256.

Dacă Z este un număr întreg, $0 \le Z < 2^{128}$, atunci $Z = 2^{64}X + Y$, unde $0 \le X < 2^{64}$ și $0 \le Y < 2^{64}$. Deoarece X și Y pot fi reprezentate ca și cuvinte pe 64 biți, x și y respectiv, întregul Z poate reprezentat ca și o pereche de cuvinte (x, y). Această proprietate este utilizată în algoritmii SHA-384, SHA-512, SHA-512/224, SHA-512/256.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Blocul de mesaj de m biți depinde de algoritmul utilizat:

a) pentru SHA-224 și SHA-256 fiecare bloc de mesaj are 512 biți, care formează un șir de 16 cuvinte pe 32 biți.

b) pentru SHA-384, SHA-512, SHA-512/224 și SHA-512/256 fiecare bloc de mesaj are 1024 biți, care formează un șir de 16 cuvinte pe 64 biți.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Vom aplica următoarele operații asupra cuvintelor:

1. Operații logice pe biți: and, or, $\oplus$, not.
2. Adunarea modulo 2^w ($w=32$ sau $w=64$) a cuvintelor x și y, $x+y$, este definită astfel: Cuvintele x și y reprezintă numerele întregi X și Y, unde $0 \le X < 2^w, 0 \le Y < 2^w$. Se determină $Z = \mathrm{mod}\left(X+Y, 2^w\right)$. Atunci avem $0 \le Z < 2^w$. Se convertește numărul întreg Z în cuvântul z și se definește astfel $z = x + y$.
3. Operația de deplasare la dreapta $x >> n$ ($0 \le n < w$).
4. Operația de rotație circulară la dreapta $ror(x,n)$, unde x este un cuvânt pe w biți, iar n este un întreg ce satisface $0 \le n < w$, este definită astfel: $ror(x,n) = (x >> n) or (x << w-n)$.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

1.3.2.2.2.1 Funcțiile și constantele utilizate

Vom defini funcțiile neliniare și constantele utilizate de către fiecare algoritm din familia SHA2.

1.3.2.2.2.1.1 Funcțiile și constantele utilizate în SHA-224 și SHA-256

SHA-224 și SHA-256 folosesc câte 6 funcții logice, iar fiecare funcție operează pe cuvinte pe 32 biți, rezultând la fel cuvinte pe 32 biți. Dacă u, v, w sunt cuvinte pe 32 biți, atunci funcțiile neliniare din cadrul SHA-224 și SHA-256 se definesc astfel:

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

$$f(u,v,w) = (u \; and \; v) \; or \; (not(u) \; and \; w) \quad \text{(multiplex);}$$

$$g(u,v,w) = (u \; and \; v) \; or \; (u \; and \; w) \; or \; (v \; and \; w) \quad \text{(majority);}$$

$$\sum_0^{\{256\}}(u) = ror(u,2) \oplus ror(u,13) \oplus ror(u,22);$$

$$\sum_1^{\{256\}}(u) = ror(u,6) \oplus ror(u,11) \oplus ror(u,25);$$

$$\sigma_0^{\{256\}}(u) = ror(u,7) \oplus ror(u,18) \oplus (u >> 3);$$

$$\sigma_1^{\{256\}}(u) = ror(u,17) \oplus ror(u,19) \oplus (u >> 10).$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

SHA-224 și SHA-256 folosesc același șir de 64 cuvinte constante pe 32 biți: $K_0^{\{256\}}, K_1^{\{256\}}, \ldots, K_{63}^{\{256\}}$. Aceste cuvinte reprezintă primii 32 biți ai părții fracționare pentru rădăcinile cubice ale primelor 64 de numere prime. În format hexazecimal cuvintele constante menționate sunt următoarele (de la stânga la dreapta pe linii):

428a2f98	71374491	b5c0fbcf	e9b5dba5	3956c25b	59f111f1	923f82a4	ab1c5ed5
d807aa98	12835b01	243185be	550c7dc3	72be5d74	80deb1fe	9bdc06a7	c19bf174
e49b69c1	efbe4786	0fc19dc6	240ca1cc	2de92c6f	4a7484aa	5cb0a9dc	76f988da
983e5152	a831c66d	b00327c8	bf597fc7	c6e00bf3	d5a79147	06ca6351	14292967
27b70a85	2e1b2138	4d2c6dfc	53380d13	650a7354	766a0abb	81c2c92e	92722c85
a2bfe8a1	a81a664b	c24b8b70	c76c51a3	d192e819	d6990624	f40e3585	106aa070
19a4c116	1e376c08	2748774c	34b0bcb5	391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
748f82ee	78a5636f	84c87814	8cc70208	90befffa	a4506ceb	bef9a3f7	c67178f2

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

1.3.2.2.2.1.2 Funcțiile și constantele utilizate în SHA-384, SHA-512, SHA-512/224 și SHA-512/256

Algoritmii SHA-384, SHA-512, SHA-512/224 și SHA-512/256 folosesc câte 6 funcții logice, iar fiecare funcție operează pe cuvinte pe 64 biți, rezultând la fel cuvinte pe 64 biți. Dacă u, v, w sunt cuvinte pe 64 biți, atunci funcțiile neliniare din cadrul SHA-384, SHA-512, SHA-512/224 și SHA-512/256 se definesc astfel:

$$f(u,v,w) = (u \; and \; v) \, or \, (not(u) \; and \; w) \quad \text{(multiplex)};$$

$$g(u,v,w) = (u \; and \; v) \, or \, (u \; and \; w) \, or \, (v \; and \; w) \quad \text{(majority)};$$

$$\sum_0^{\{512\}}(u) = ror(u,28) \oplus ror(u,34) \oplus ror(u,39);$$

$$\sum_1^{\{512\}}(u) = ror(u,14) \oplus ror(u,18) \oplus ror(u,41);$$

$$\sigma_0^{\{512\}}(u) = ror(u,1) \oplus ror(u,8) \oplus (u >> 7);$$

$$\sigma_1^{\{512\}}(u) = ror(u,19) \oplus ror(u,61) \oplus (u >> 6).$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

SHA-384, SHA-512, SHA-512/224 și SHA-512/256 folosesc același șir de 80 cuvinte constante pe 64 biți: $K_0^{\{512\}}, K_1^{\{512\}}, \ldots, K_{79}^{\{512\}}$. Aceste cuvinte reprezintă primii 64 biți ai părții fracționare pentru rădăcinile cubice ale primelor 80 de numere prime. În format hexazecimal cuvintele constante menționate sunt următoarele (de la stânga la dreapta pe linii):

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

428a2f98d728ae22	7137449123ef65cd	b5c0fbcfec4d3b2f	e9b5dba58189dbbc
3956c25bf348b538	59f111f1b605d019	923f82a4af194f9b	ab1c5ed5da6d8118
d807aa98a3030242	12835b0145706fbe	243185be4ee4b28c	550c7dc3d5ffb4e2
72be5d74f27b896f	80deb1fe3b1696b1	9bdc06a725c71235	c19bf174cf692694
e49b69c19ef14ad2	efbe4786384f25e3	0fc19dc68b8cd5b5	240ca1cc77ac9c65
2de92c6f592b0275	4a7484aa6ea6e483	5cb0a9dcbd41fbd4	76f988da831153b5
983e5152ee66dfab	a831c66d2db43210	b00327c898fb213f	bf597fc7beef0ee4
c6e00bf33da88fc2	d5a79147930aa725	06ca6351e003826f	142929670a0e6e70
27b70a8546d22ffc	2e1b21385c26c926	4d2c6dfc5ac42aed	53380d139d95b3df
650a73548baf63de	766a0abb3c77b2a8	81c2c92e47edaee6	92722c851482353b
a2bfe8a14cf10364	a81a664bbc423001	c24b8b70d0f89791	c76c51a30654be30
d192e819d6ef5218	d69906245565a910	f40e35855771202a	106aa07032bbd1b8

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

19a4c116b8d2d0c8	1e376c085141ab53	2748774cdf8eeb99	34b0bcb5e19b48a8
391c0cb3c5c95a63	4ed8aa4ae3418acb	5b9cca4f7763e373	682e6ff3d6b2b8a3
748f82ee5defb2fc	78a5636f43172f60	84c87814a1f0ab72	8cc702081a6439ec
90befffa23631e28	a4506cebde82bde9	bef9a3f7b2c67915	c67178f2e372532b
ca273eceea26619c	d186b8c721c0c207	eada7dd6cde0eb1e	f57d4f7fee6ed178
06f067aa72176fba	0a637dc5a2c898a6	113f9804bef90dae	1b710b35131c471b
28db77f523047d84	32caab7b40c72493	3c9ebe0a15c9bebc	431d67c49c100d4c
4cc5d4becb3e42b6	597f299cfc657e2a	5fcb6fab3ad6faec	6c44198c4a475817

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

1.3.2.2.2.2 Preprocesarea

Preprocesarea include trei etape: procedura de completare (padding) a mesajului, divizarea mesajului pe blocuri și setarea valorii hash inițiale.

1.3.2.2.2.2.1 Completarea mesajului

Obiectivul procedurii de padding este de a asigura că mesajul completat are lungime un multiplu al lui 512 sau 1024 biți, în dependență de algoritmul aplicat.

Fie mesajul M de lungime l biți.

În cazul algoritmilor SHA-224 și SHA-256 (SHA-384, SHA-512, SHA-512/224 și SHA-512/256) vom proceda astfel. Se anexează bitul „1" la sfârșitul mesajului, după care urmează k biți de „0", unde k este cel mai mic număr întreg nenegativ pentru care se satisface condiția

$$\mathrm{mod}(l+1+k,512)=\mathrm{mod}(448,512).$$

$$(\mathrm{mod}(l+1+k,1024)=\mathrm{mod}(896,1024)).$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

În continuare, este anexat un bloc pe 64 biți (pe 128 biți) care este reprezentarea binară a numărului l.

Exemplul 1.3.1. Mesajul „abc” (fiecare caracter este reprezentat pe 8 biți) are lungimea $8\times3=24$. Mesajul este completat cu un bit de 1, urmat de 448-(24+1)=423 biți de zero (896-(24+1)=871 biți de zero), iar apoi urmează reprezentarea binară pe 64 biți (pe 128 biți) a lungimii mesajului $l=24$. Astfel, se obține mesajul completat pe 512 biți (pe 1024 biți):

$$\underbrace{01100001}_{\text{"}a\text{"}}\ \underbrace{01100010}_{\text{"}b\text{"}}\ \underbrace{01100011}_{\text{"}c\text{"}}\ 1\ \overbrace{00\ldots00}^{423}\ \overbrace{00\ldots0\underbrace{11000}_{l=24}}^{64}$$

$$(\ \underbrace{01100001}_{\text{"}a\text{"}}\ \underbrace{01100010}_{\text{"}b\text{"}}\ \underbrace{01100011}_{\text{"}c\text{"}}\ 1\ \overbrace{00\ldots00}^{871}\ \overbrace{00\ldots0\underbrace{11000}_{l=24}}^{128}\ )$$

Lungimea mesajului completat trebuie să fie acum un multiplu al lui 512 biți (al lui 1024 biți). □

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

1.3.2.2.2.2.2 Divizarea mesajului pe blocuri

În cazul algoritmilor SHA-224, SHA-256 (SHA-384, SHA-512, SHA-512/224, SHA-512/256) mesajul completat este divizat în N blocuri pe 512 biți (pe 1024 biți) $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$. Deoarece 512 biți (1024 biți) ai blocului de intrare pot fi reprezentați ca și 16 cuvinte pe 32 biți (pe 64 biți), primii 32 biți (64 biți) ai blocului i sunt notați prin $M_0^{(i)}$, următorii 32 biți (64 biți) – prin $M_1^{(i)}$, ș.a.m.d., ultimii 32 biți (64 biți) – prin $M_{15}^{(i)}$.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

1.3.2.2.2.2.3 Setarea valorii hash inițiale

La inițializarea calculului iterativ al valorii hash se determină o valoare hash inițială $H^{(0)}$. Lungimea și numărul de cuvinte ale lui $H^{(0)}$ depinde de lungimea valorii hash finale.

Valoarea hash inițială pentru SHA-224

Valoarea hash inițială $H^{(0)}$ constă din următoarele 8 cuvinte pe 32 biți, scrise în hexazecimal:

$$H_0^{(0)} = c1059ed8,\ H_1^{(0)} = 367cd507,\ H_2^{(0)} = 3070dd17,\ H_3^{(0)} = f70e5939,$$

$$H_4^{(0)} = ffc00b31,\ H_5^{(0)} = 68581511,\ H_6^{(0)} = 64f98fa7,\ H_7^{(0)} = befa4fa4.$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Valoarea hash inițială pentru SHA-256

Valoarea hash inițială $H^{(0)}$ constă din următoarele 8 cuvinte pe 32 biți, scrise în hexazecimal:

$$H_0^{(0)} = 6a09e667,\ H_1^{(0)} = bb67ae85,\ H_2^{(0)} = 3c6ef372,\ H_3^{(0)} = a54ff53a,$$

$$H_4^{(0)} = 510e527f,\ H_5^{(0)} = 9b05688c,\ H_6^{(0)} = 1f83d9ab,\ H_7^{(0)} = 5be0cd19.$$

Aceste cuvinte sunt obținute prin preluarea primilor 32 biți din părțile fracționare ale rădăcinilor pătrate ale primilor 8 numere prime.

Valoarea hash inițială pentru SHA-384

Valoarea hash inițială $H^{(0)}$ constă din următoarele 8 cuvinte pe 64 biți, scrise în hexazecimal:

$$H_0^{(0)} = cbbb9d5dc1059ed8,\ H_1^{(0)} = 629a292a367cd507,\ H_2^{(0)} = 9159015a3070dd17,$$

$$H_3^{(0)} = 152fecd8f70e5939,\ H_4^{(0)} = 67332667ffc00b31,\ H_5^{(0)} = 8eb44a8768581511,$$

$$H_6^{(0)} = db0c2e0d64f98fa7,\ H_7^{(0)} = 47b5481dbefa4fa4.$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Valoarea hash inițială pentru SHA-512

Valoarea hash inițială $H^{(0)}$ constă din următoarele 8 cuvinte pe 64 biți, scrise în hexazecimal:

$$H_0^{(0)} = 6a09e667f3bcc908,\ H_1^{(0)} = bb67ae8584caa73b,\ H_2^{(0)} = 3c6ef372fe94f82b,$$

$$H_3^{(0)} = a54ff53a5f1d36f1,\ H_4^{(0)} = 510e527fade682d1,\ H_5^{(0)} = 9b05688c2b3e6c1f,$$

$$H_6^{(0)} = 1f83d9abfb41bd6b,\ H_7^{(0)} = 5be0cd19137e2179.$$

Aceste cuvinte sunt obținute prin preluarea primilor 64 biți din părțile fracționare ale rădăcinilor pătrate ale primelor 8 elemente din șirul numerelor prime.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Valoarea hash inițială pentru SHA-512/224

Valoarea hash inițială $H^{(0)}$ constă din următoarele 8 cuvinte pe 64 biți, scrise în hexazecimal:

$$H_0^{(0)} = 8C3D37C819544DA2,\ H_1^{(0)} = 73E1996689DCD4D6,\ H_2^{(0)} = 1DFAB7AE32FF9C82,$$

$$H_3^{(0)} = 679DD514582F9FCF,\ H_4^{(0)} = 0F6D2B697BD44DA8,\ H_5^{(0)} = 77E36F7304C48942,$$

$$H_6^{(0)} = 3F9D85A86A1D36C8,\ H_7^{(0)} = 1112E6AD91D692A1.$$

Valoarea hash inițială pentru SHA-512/256

Valoarea hash inițială $H^{(0)}$ constă din următoarele 8 cuvinte pe 64 biți, scrise în hexazecimal:

$$H_0^{(0)} = 22312194FC2BF72C,\ H_1^{(0)} = 9F555FA3C84C64C2,\ H_2^{(0)} = 2393B86B6F53B151,$$

$$H_3^{(0)} = 963877195940EABD,\ H_4^{(0)} = 96283EE2A88EFFE3,\ H_5^{(0)} = BE5E1E2553863992,$$

$$H_6^{(0)} = 2B0199FC2C85B8AA,\ H_7^{(0)} = 0EB72DDC81C52CA2.$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

1.3.2.2.2.3 Descrierea algoritmilor din familia SHA-2

Mai întâi vom da o descriere a algoritmului SHA-256, deoarece specificarea lui SHA-224 este aproape identică cu cea pentru SHA-256, cu excepția că valorile hash inițiale sunt distincte, iar valoarea hash finală este trunchiată la 224 biți pentru SHA-224. Același lucru este valabil și pentru SHA-512, deoarece în SHA-384 valoarea hash finală este trunchiată la 384 biți, în SHA-512/224 – la 224 biți, iar în SHA-512/256 – la 256 biți.

1.3.2.2.2.3.1 Descrierea algoritmului SHA-256

SHA-256 poate fi utilizat pentru a calcula valoarea hash a mesajului M ce are lungimea l biți, unde $0 \le l < 2^{64}$. Algoritmul folosește următoarele:

1) un bloc de mesaj format din 64 cuvinte pe 32 biți fiecare;
2) 8 variabile de lucru pe 32 biți fiecare;
3) o valoare hash din 8 cuvinte pe 32 biți fiecare.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Rezultatul final întors de SHA-256 este o valoare hash pe 256 biți.

Vom folosi următoarele notații:

$W_0, W_1, \ldots, W_{63}$ - cuvintele blocului de mesaj;

A, B, C, D, E, F, G, J - cele 8 variabile de lucru;

$H_0^{(i)}, H_1^{(i)}, \ldots, H_7^{(i)}$ - cuvintele valorii hash;

T_1, T_2 - cuvinte temporare.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Algoritmul SHA-256

Date de intrare:

M - *șirul de biți de lungime arbitrară* $l \geq 0$, *pentru care se calculează valoarea hash*

Date de ieșire:

$h(M)$ - Valoarea hash pe 256 biți a lui M

Pașii algoritmului:

1. *Definirea constantelor și a funcțiilor*

   Constantele și funcțiile utilizate în SHA-256 sunt precizate la secțiunea 1.3.2.2.2.1.1.

2. *Preprocesarea*

   Se inițializează valoarea hash $H^{(0)}$, iar mesajul este completat și divizat pe blocuri conform procedurii expuse la secțiunea 1.3.2.2.2.2.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

3. Procesarea

Operația de adunare + este realizată modulo 2^{32}. Pe rând este procesat fiecare bloc de mesaj $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$ în modul următor:

pentru $i = \overline{1, N}$ execută

{

Se determină blocul de mesaj $\{W_t\}$:

$$W_t := \begin{cases} M_t^{(i)}, & 0 \le t \le 15 \\ \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16}, & 16 \le t \le 63 \end{cases}$$

Se inițializează cele 8 variabile de lucru A, B, C, D, E, F, G, J cu valoarea hash de indice $i-1$:

$$A := H_0^{(i-1)},\ B := H_1^{(i-1)},\ C := H_2^{(i-1)},\ D := H_3^{(i-1)},\ E := H_4^{(i-1)},\ F := H_5^{(i-1)},\ G := H_6^{(i-1)},\ J := H_7^{(i-1)}.$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

pentru $t = \overline{0,63}$ execută

{

$T_1 := J + \sum_1^{\{256\}}(E) + f(E,F,G) + K_t^{\{256\}} + W_t$, $T_2 := \sum_0^{\{256\}}(A) + g(A,B,C)$

$J := G,\ G := F,\ F := E,\ E := D + T_1,\ D := C,\ C := B,\ B := A,\ A := T_1 + T_2$

}

Se calculează valoarea hash intermediară $H^{(i)}$:

$H_0^{(i)} := A + H_0^{(i-1)},\ H_1^{(i)} := B + H_1^{(i-1)},\ H_2^{(i)} := C + H_2^{(i-1)},\ H_3^{(i)} := D + H_3^{(i-1)},\ H_4^{(i)} := E + H_4^{(i-1)},$

$H_5^{(i)} := F + H_5^{(i-1)},\ H_6^{(i)} := G + H_6^{(i-1)},\ H_7^{(i)} := J + H_7^{(i-1)}$

}

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

4. Completarea

Valoarea hash finală $h(M)$ pe 256 biți este concatenarea tuturor $H_i^{(N)}, i = \overline{0,7}$:

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)} .$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Tabel cu valori hash (în hexazecimal) pentru testare:

"The quick brown fox jumps over the lazy dog"	d7a8fbb307d7809469ca9abcb0082e4f8d5651e46d3cdb762d02d0bf37c9e592
„The quick brown fox jumps over the lazy cog"	e4c4d8f3bf76b692de791a173e05321150f7a345b46484fe427f6acc7ecc81be
„abc"	ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad
„ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	db4bfcbd4da0cd85a60c3c37d3fbd8805c77f15fc6b1fdfe614ee0a7c8fdb4c0

Detalii cu rezultatele de la fiecare etapă de realizare a algoritmului pot fi găsite la adresa:

https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/example-values

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

1.3.2.2.2.3.2 Descrierea algoritmului SHA-224

SHA-224 poate fi utilizat pentru a calcula valoarea hash a mesajului M ce are lungimea l biți, unde $0 \le l < 2^{64}$. Algoritmul funcționează în exact aceeași manieră ca și SHA-256 cu următoarele două excepții:

1. Valoarea hash inițială $H^{(0)}$ va fi inițializată după cum a fost specificat la secțiunea 1.3.2.2.2.2.3.
2. Valoarea hash finală pe 224 biți este obținută prin trunchierea valorii hash finale, mai exact, prin preluarea doar a celor mai din stânga 224 biți ai acesteia:

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)}.$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Tabel cu valori hash (în hexazecimal) pentru testare:

"The quick brown fox jumps over the lazy dog"	730e109bd7a8a32b1cb9d9a09aa2325d2430587ddbc0c38bad911525
„The quick brown fox jumps over the lazy cog"	fee755f44a55f20fb3362cdc3c493615b3cb574ed95ce610ee5b1e9b
„abc"	23097d223405d8228642a477bda255b32aadbce4bda0b3f7e36c9da7
„ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	bff72b4fcb7d75e5632900ac5f90d219e05e97a7bde72e740db393d9

Detalii cu rezultatele de la fiecare etapă de realizare a algoritmului pot fi găsite la adresa:

https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/example-values

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

1.3.2.2.2.3.3 Descrierea algoritmului SHA-512

SHA-512 poate fi utilizat pentru a calcula valoarea hash $h(M)$ a mesajului M ce are lungimea l biți, unde $0 \le l < 2^{128}$. Algoritmul folosește următoarele:

1) un bloc de mesaj format din 80 cuvinte pe 64 biți fiecare;
2) 8 variabile de lucru pe 64 biți fiecare;
3) o valoare hash din 8 cuvinte pe 64 biți fiecare.

Rezultatul final întors de SHA-512 este o valoare hash pe 512 biți.

Vom folosi următoarele notații:

$W_0, W_1, ..., W_{79}$ - cuvintele blocului de mesaj;

A, B, C, D, E, F, G, J - cele 8 variabile de lucru;

$H_0^{(i)}, H_1^{(i)}, ..., H_7^{(i)}$ - cuvintele valorii hash;

T_1, T_2 - cuvinte temporare.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Algoritmul SHA-512

Date de intrare:

M - *șirul de biți de lungime arbitrară* $l \geq 0$, *pentru care se calculează valoarea hash*

Date de ieșire:

$h(M)$ - Valoarea hash pe 512 biți a lui M

Pașii algoritmului:

1. *Definirea constantelor și a funcțiilor*

   Constantele și funcțiile utilizate în SHA-512 sunt precizate la secțiunea 1.3.2.2.2.1.2.

2. *Preprocesarea*

   Se inițializează valoarea hash $H^{(0)}$, iar mesajul este completat și divizat pe blocuri conform procedurii expuse la secțiunea 1.3.2.2.2.2.

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

3. *Procesarea*

Operația de adunare + este realizată modulo 2^{64}. Pe rând este procesat fiecare bloc de mesaj $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$ în modul următor:

pentru $i = \overline{1, N}$ execută

{

Se determină blocul de mesaj $\{W_t\}$:

$$W_t := \begin{cases} M_t^{(i)}, & 0 \le t \le 15 \\ \sigma_1^{\{512\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{512\}}(W_{t-15}) + W_{t-16}, & 16 \le t \le 79 \end{cases}.$$

Se inițializează cele 8 variabile de lucru A, B, C, D, E, F, G, J cu valoarea hash de indice $i-1$:

$$A := H_0^{(i-1)},\ B := H_1^{(i-1)},\ C := H_2^{(i-1)},\ D := H_3^{(i-1)},\ E := H_4^{(i-1)},\ F := H_5^{(i-1)},\ G := H_6^{(i-1)},\ J := H_7^{(i-1)}.$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

pentru $t = \overline{0,79}$ execută

{

$T_1 := J + \sum_1^{\{512\}}(E) + f(E,F,G) + K_t^{\{512\}} + W_t$, $T_2 := \sum_0^{\{512\}}(A) + g(A,B,C)$

$J := G,\ G := F,\ F := E,\ E := D + T_1,\ D := C,\ C := B,\ B := A,\ A := T_1 + T_2$

}

Se calculează valoarea hash intermediară $H^{(i)}$:

$H_0^{(i)} := A + H_0^{(i-1)},\ H_1^{(i)} := B + H_1^{(i-1)},\ H_2^{(i)} := C + H_2^{(i-1)},\ H_3^{(i)} := D + H_3^{(i-1)},\ H_4^{(i)} := E + H_4^{(i-1)},$

$H_5^{(i)} := F + H_5^{(i-1)},\ H_6^{(i)} := G + H_6^{(i-1)},\ H_7^{(i)} := J + H_7^{(i-1)}$

}

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

4. Completarea

Valoarea hash finală $h(M)$ pe 512 biți este concatenarea tuturor $H_i^{(N)}, i = \overline{0,7}$:

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)} .$$

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Tabel cu valori hash (în hexazecimal) pentru testare:

"The quick brown fox jumps over the lazy dog"	07e547d9586f6a73f73fbac0435ed76951218fb7d0c8d788a309d785436bbb642e93a252a954f23912547d1e8a3b5ed6e1bfd7097821233fa0538f3db854fee6
„The quick brown fox jumps over the lazy cog"	3eeee1d0e11733ef152a6c29503b3ae20c4f1f3cda4cb26f1bc1a41f91c7fe4ab3bd86494049e201c4bd5155f31ecb7a3c8606843c4cc8dfcab7da11c8ae5045
„abc"	ddaf35a193617abacc417349ae20413112e6fa4e89a97ea20a9eeee64b55d39a2192992a274fc1a836ba3c23a3feebbd454d4423643ce80e2a9ac94fa54ca49f
„ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	1e07be23c26a86ea37ea810c8ec7809352515a970e9253c26f536cfc7a9996c45c8370583e0a78fa4a90041d71a4ceab7423f19c71b9d5a3e01249f0bebd5894

Detalii cu rezultatele de la fiecare etapă de realizare a algoritmului pot fi găsite la adresa:

https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/example-values

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

1.3.2.2.2.3.4 Descrierea algoritmilor SHA-384, SHA-512/224 și SHA-512/256

SHA-384 (SHA-512/224 sau SHA-512/256) poate fi utilizat pentru a calcula valoarea hash a mesajului M ce are lungimea l biți, unde $0 \le l < 2^{128}$. Algoritmul funcționează în exact aceeași manieră ca și SHA-512 cu următoarele două excepții:

1. Valoarea hash inițială $H^{(0)}$ va fi inițializată după cum a fost specificat la secțiunea 1.3.2.2.2.2.3.
2. Valoarea hash finală pe 384 biți (pe 224 biți sau pe 256 biți) este obținută prin trunchierea valorii hash finale, mai exact, prin preluarea doar a celor mai din stânga 384 biți (224 biți sau 256 biți) ai acesteia:

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)}$$

($H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| tr_{32}\left(H_3^{(N)}\right)$ sau $H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)}$, unde $tr_{32}\left(H_3^{(N)}\right)$ - cei mai din stânga 32 biți ai lui $H_3^{(N)}$).

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Tabel cu valori hash (în hexazecimal) pentru testarea lui SHA-384:

"The quick brown fox jumps over the lazy dog"	ca737f1014a48f4c0b6dd43cb177b0afd9e5169367544c494011e3317dbf9a509cb1e5dc1e85a941bbee3d7f2afbc9b1
„The quick brown fox jumps over the lazy cog"	098cea620b0978caa5f0befba6ddcf22764bea977e1c70b3483edfdf1de25f4b40d6cea3cadf00f809d422feb1f0161b
„abc"	cb00753f45a35e8bb5a03d699ac65007272c32ab0eded1631a8b605a43ff5bed8086072ba1e7cc2358baeca134c825a7
„ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	1761336e3f7cbfe51deb137f026f89e01a448e3b1fafa64039c1464ee8732f11a5341a6f41e0c202294736ed64db1a84

FUNCȚII HASH FĂRĂ CHEIE CUSTOMIZATE. FAMILIA DE ALGORITMI SHA-2

Tabel cu valori hash (în hexazecimal) pentru testarea lui SHA-512/224:

"The quick brown fox jumps over the lazy dog"	944cd2847fb54558d4775db0485a50003111c8e5daa63fe722c6aa37
„The quick brown fox jumps over the lazy cog"	2b9d6565a7e40f780ba8ab7c8dcf41e3ed3b77997f4c55aa987eede5
„abc"	4634270f707b6a54daae7530460842e20e37ed265ceee9a43e8924aa
„ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	a8b4b9174b99ffc67d6f49be9981587b96441051e16e6dd036b140d3

Tabel cu valori hash (în hexazecimal) pentru testarea lui SHA-512/256:

"The quick brown fox jumps over the lazy dog"	dd9d67b371519c339ed8dbd25af90e976a1eeefd4ad3d889005e532fc5bef04d
„The quick brown fox jumps over the lazy cog"	cc8d255a7f2f38fd50388fd1f65ea7910835c5c1e73da46fba01ea50d5dd76fb
„abc"	53048e2681941ef99b2e29b76b4c7dabe4c2d0c634fc6d46e0e2f13107e7af23
„ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	cdf1cc0effe26ecc0c13758f7b4a48e000615df241284185c39eb05d355bb9c8

BIBLIOGRAFIE RECOMANDATĂ

1. A. Menezes, P. van Oorschot and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.
2. B. Forouzan, Criptography&Network Security, McGraw-Hill Science/Engineering/Math, 2007.
3. B. Preneel, Analysis and Design of Cryptographic Hash Functions, 2003.
4. National Institute of Standards and Technology (NIST), "FIPS Publication 180-4: Secure Hash Standard (SHS)," August 2015.
5. R. Rivest, "The MD4 message digest algorithm," Advances in Cryptology, Proc. Crypto'90, 1991.
6. R. Rivest, "The MD5 message-digest algorithm," April 1992.
7. R. Rivest, "The MD6 hash function. A proposal to NIST for SHA-3," MIT, Cambridge, 2008.