Дәріс 9. Хэш функция

Жоспар:

1. Хэш функция анықтамасы, функциясы
2. MD5 (message Digest Algorithm 5) алгоритмі
3. Хэш–функцияны есептеуге арналған MD5 алгоритмі
4. *SHA–1* алгоритмі

h хэш функциясы – ерікті ұзындықтағы бит жолдарында анықталған және бекітілген ұзындықтағы бит жолдарын қайтаратын функция. Басқаша айтқанда («қысады»), хэш функциясы ерікті ұзындықтағы хабарламаны белгіленген ұзындықтағы хабарламаға түрлендіреді. Хэш функцияларының әртүрлі қосымшалары бар. Мысалы, әртүрлі ұзындықтағы хабарламалардың үлкен тізімінен хабарлама іздегенде, хабарламалардың өзін емес, олардың ұзындығы бірдей хэш мәндерін салыстыру ыңғайлы. Сонымен қатар, хэш функциясының нәтижелерін бақылау сомасы ретінде пайдалануға болады.

Криптографиялық хэш функциялары стандартты хэш–функциялардан ерекшеленеді, өйткені олар бір жақты функциялар болуы керек. Сонымен қатар, криптографиялық функциялар M хабарламасын $H = h(M)$ мәнінен қайта құруды мүмкін етпеу үшін қажет.Жоғарыда айтылғандардан басқа, криптографиялық хэш функциялары келесі талаптарды қанағаттандыруы керек:

1) кез келген M_1 хабарламасы үшін $h(M_1) = h(M_2)$ болатындай M_2, $M_1 \neq M_2$, хабарламасын табу мүмкін емес;

2) M_1 және M_2, $M_1 \neq M_2$, мәндерінің жұбын табу мүмкін емес, осылайша $h(M_1) = h(M_2)$;

3) аргумент мәндерін (хабарламаларды) кездейсоқ таңдаған кезде хэш функциясының мәндері біркелкі таратылуы керек;

4) егер M_1 және M_2 аргументтерінің мәндері бір–бірінен аз айырмашылығы болса, олардың хэш мәндері жақын болмауы керек.

Криптографияда хэш функциялары келесі есептерді шешу үшін қолданылады:

1) беру және сақтау кезінде деректер тұтастығын бақылау жүйелерін құру;

2) аутентификация процедураларында;

3) электрондық цифрлық қолтаңбаны қалыптастыру кезінде.

Хэш–функцияның дизайн принципі оның мәндері қарлы әсерге ие болуы керек – аргументтегі кішкене өзгеріс хэш функциясының мәніне үлкен әсер етеді. Әдетте, хэш–функциялар бір сатылы қысу функцияларына негізделген

$$y = f(u, v)$$

екі айнымалы. Мұндағы u, v – L ұзындығының екілік векторлары, ал y – ұзындығы хэш функциясы мәндерінің бекітілген ұзындығымен анықталатын

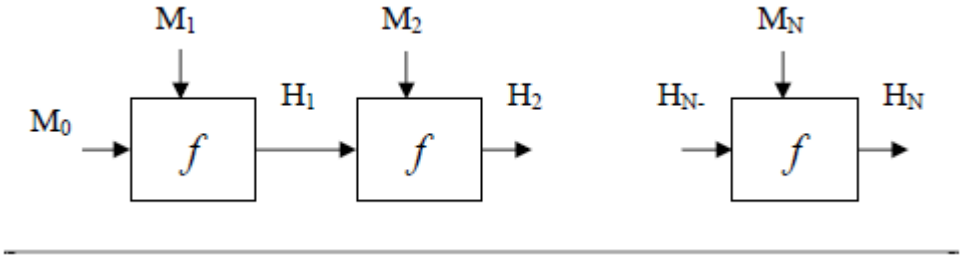
екілік векторы. Егер M хабары берілсе, онда $H = h(M)$ есептеу үшін M алдымен N блокқа бөлінеді

$$M_1, M_2, \dots, M_N$$

ұзындығы L. Егер M ұзындығы L еселігі болмаса, онда соңғы блок белгілі бір ережеге сәйкес L ұзындығына толықтырылады. Содан кейін алынған блоктарға келесі есептеулер тізбегі қолданылады:

$$H_0 = M_0,$$
$$H_i = f(M_i, H_i - 1), i = 1, \dots, N,$$
$$h(M) = H_N,$$

мұндағы H_0 – хэш–функцияның бастапқы мәні, M_0 әдіс тұрақтысы. Хэш–функцияны есептеудің жалпы сұлбасы 5.1 – суретте көрсетілген.



5.1 – сурет. Хэш функциясын есептеудің жалпы схемасы

Хэш функциясының мәндері қайталанбауы үшін олардың ұзындығы кемінде 128 бит болуы керек. Дегенмен, 160 биттік ұзындыққа артықшылық беріледі.

MD5 алгоритмі

MD5 (message Digest Algorithm 5) алгоритмі 1991 жылы Рональд Ривест жасаған криптографиялық хэш функциясы. Ол еркін ұзындықтағы кірістерді қабылдайды және 128 биттік (16 биттік) хэш мәнін есептейді. Алгоритм кіріске ерікті ұзындықтағы хабарламаны қабылдайды және шығысқа 128 биттік мән береді. Кіріс дәйекті түрде 512 биттік блоктарға бөлінеді.

MD5 алгоритмі мыналарды пайдаланады:

Мәндер жиымы T [i],i = 1,2,…,64,

T[1] = D76AA478	T[17] = F61E2562	T[33]= FFFA3942	T[49] = F4292244
T[2] = E8C7B756	T[18] = C040B340	T[34] = 8771F681	T[50] = 432AFF97
T[3] – 242070DB	T[19] = 265E5A51	T[35] = 699D6122	T[51] = AB9423A7
T[4] = C1BDCEEE	T[20] = E9B6C7A	T[36] = FDE5380C	T[52] = FC93A039
T[5] = F57COFAF	T[21] = D62F105D	T[37] = A4BEEA44	T[53] = 655B59C3
T[6] = 4787C62A	T[22] = 02441453	T[38] = 4BDECFA9	T[54] = 8FOCCC92
T[7] = A8304613	T[23] = D8A1E681	T[39] = F6BB4B60	T[55] = FFEFF47D

T[8] = FD469501	T[24] = E7D3FBC8	T[40] = BEBFBC70	T[56] = 85845DD1
T[9] = 698098D8	T[25] = 21E1CDE6	T[41] = 289B7EC6	T[57] = 6FA87E4F
T[10] = 8B44F7AF	T[26] = C33707D6	T[42] = EAA127FA	T[58] = FE2CE6EO
T[11] = FFFF5BB1	T[27] = F4D50D87	T[43] = D4EF3085	T[59] = A3014314
T[12] = 895CD7BE	T[28] = 455A14ED	T[44] = 04881D05	T[60] = 4E0811A1
T[13] = 6B901122	T[29] = A9E3E905	T[45] = D9D4D039	T[61] = F7537E82
T[14] = FD987193	T[3O] = FCEFA3F8	T[46] = E6DB99E5	T[62] = BD3AF235
T[15] = A679438E	T[31] = 676F02D9	T[47] = 1FA27CF8	T[63] = 2AD7D2BB
T[1б] = 49B40821	T[32] = 8D2A4C8A	T[48] = C4AC5665	T[64] = EBE6D391

$T\ [i]$ элементі санның бүтін бөлігіне тең

$$2^{32} \times abs(sin(i)),$$

мұндағы i радиандағы бұрыш мәніне сәйкес келеді.

Төрт айнымалы *A, B, C, D* әрқайсысы 32 бит. Бұл айнымалылар тізбекті айнымалылар деп аталады және олар келесі мәндермен инициализацияланады:

A = 0x6745012301,
B = 0xEFCDAB89,
C = 0x98BADCFE,
D = 0x10325476.

Бұл деректер сөздің ең аз маңызды байты кіші мекенжаймен орналастырылған кезде байттардың тікелей жазуы ретіндегі пішімде сақталады. Осылайша, инициализация мәнінің 32–биттік жолдары келесідей болады

A: 01 23 45 67,
B: 89 AB CD EF,
C: FE DC BA 98,
D: 76 54 32 10.

Төрт функция:

1. $F(b,c,d) = (b \wedge c) \vee (b \wedge d)$;
2. $G(b,c,d) = (b \wedge d) \vee (c \wedge d)$;
3. $H(b,c,d) = b \oplus c \oplus d;$
4. $I(b,c,d) = c \oplus (b \vee d)$.

Әрбір функция үш 32 биттік b, c және d айнымалыларын бір 32 биттік мәнге түрлендіреді.

Хэш–функцияны есептеуге арналған MD5 алгоритмі

Хэш функциясын есептеуге арналған MD5 алгоритмі-еркін ұзындықтағы кірістерді 128 биттік хэш мәніне түрлендіретін криптографиялық хэш функциясы. MD5 алгоритмінің негізгі жұмыс қадамдары хабарламаны қажетті ұзындыққа қосуды, оны 512 биттік блоктарға бөлуді, төрт 32 биттік регистрлерді инициализациялауды, сызықтық емес функциялар мен тұрақтыларды қолдана отырып 64 түрлендіру раундын орындауды, содан кейін нәтижелерді қорытынды хэшке біріктіруді қамтиды. Хэш функциясын есептеу үшін MD5 алгоритмінің келесі негізгі қадамдары

орындалады:

1–қадам. Биттер бастапқы хабарламаға қосылады. Хабарлама толтырылған, сонда оның биттегі жалпы ұзындығы 448 модуль 512 мәнімен салыстырылады. Хабардың ұзындығы L болсын, онда L үшін формула қанағаттандырылуы керек $L \equiv 448\, mod\, 512$.

Бұл толтырылған хабарламаның биттегі ұзындығы болуы керек дегенді білдіреді 64 бит 512 санының ең жақын еселігінен аз. Толтыру операциясы хабарлама қажетті ұзындықта болса да әрқашан орындалады. Егер хабарлама ұзындығы 448 бит болса, онда хабарламаға тағы 512 бит қосылады, нәтижесінде кеңейтілген хабарламаның ұзындығы 960 бит болады. Толықтауыш құрылымы келесідей: толықтырылған мәннің бірінші биті 1-ге, ал қалғандары 0-ге тең.

2–қадам. Бастапқы хабарламаның ұзындығы 1–қадамдағы хабарламаға қосылады. Бастапқы хабарламаның 64 биттік ұзындығы 1-қадамның нәтижесіне қосылады, яғни. хабарлама ұзындығы 1-қадамға дейін. Ең аз маңызды ұзындық байт бірінші жазылады. Егер бастапқы хабарламаның ұзындығы 264 асса, онда ұзындық мәнінің тек төменгі 64 биттері пайдаланылады. Осылайша, хабарлама ұзындығының сәйкес өрісі 264 бастапқы хабарлама модулінің ұзындығын қамтиды.

3–қадам. 2-қадамның нәтижесі 512 бит блоктарға бөлінеді

$$M_1, M_2\ \dots, M_K\ ,$$

мұндағы K мұндай блоктардың саны (L= K⊕ 512). Өз кезегінде әрбір блок $M_i, i\ =\ 1, 2, \dots\ ,\ K,$ әрқайсысы 32 биттен тұратын 16 ішкі блокқа бөлінеді.

$$X[0], X[1], \dots, X[k], k = 0, 1, \dots, 15.$$

4–қадам. Хэш функциясын есептеу кезінде 128 биттің аралық мәндерін жазу үшін әрқайсысы 32 биттен тұратын төрт a, b, c және d айнымалыларынан тұратын H буфері пайдаланылады. Бұл айнымалылар тиісінше бұрын анықталған A, B, C және D тұрақтылары арқылы инициализацияланады.

5–қадам. $M_i, i\ =\ 1, 2, \dots\ ,\ K$ блоктарын H_i профиліне 512 битке түрлендіру

Ұзындығы 128 бит. Еске салайық, Мі блогы әрқайсысы 32 биттен тұратын 16 қосалқы блокқа бөлінген $X[0], X[1], \dots, X[k], k = 0, 1, \dots, 15.$

Содан кейін әрбір j қадамында келесі операциялар қолданылады:

бірінші раунд:

$a=b+ ((a+F(b,c,d)+X[k]+T[j]) <<< s_j, j=1, \dots, 16;$

екінші тур:

$a=b+ ((a+G(b,c,d)+X[k]+T[j]) <<< s_j, j=17, \dots, 32;$

үшінші раунд

$a=b+ ((a+H(b,c,d)+X[k]+T[j]) <<< s_j, j=33, \dots, 48;$

төртінші айналым:

$a=b+ ((a+I(b,c,d)+X[k]+T[j]) <<< s_j, j=49, \dots, 64.$

Әр айналымның амалдарын сәйкесінше былай белгілейік

$F(abcd\ \ k\ \ j\ \ s)$,
$G(abcd\ \ k\ \ j\ \ s)$,
$H(abcd\ \ k\ \ j\ \ s)$,
$I(abcd\ k\ j\ s)$.

Содан кейін 16 қадамнан тұратын төрт айналым келесідей болады
1–раунд:

F ($abcd$ 0 1 7)
F ($dabc$ 1 2 12)
F ($cdab$ 2 3 17)
F ($bcda$ 3 4 22)
F ($abcd$ 4 5 7)
F ($dabc$ 5 6 12)
F ($cdab$ 6 7 17)
F ($bcda$ 7 8 22)
F ($abcd$ 8 9 7)
F ($dabc$ 9 10 12)
F ($cdab$ 10 11 17)
F ($bcda$ 11 12 22)
F ($abcd$ 12 13 7)
F ($dabc$ 13 14 12)
F ($cdab$ 14 15 17)

2–раунд:

G ($dabc$ 6 18 9)
G ($cdab$ 11 19 14)
G ($bcda$ 0 20 20)
G ($abcd$ 5 21 5)
G ($dabc$ 10 22 9)
G ($cdab$ 15 23 14)
G ($bcda$ 4 24 20)
G ($abcd$ 9 25 5)
G ($dabc$ 14 26 9)
G ($cdab$ 3 27 14)
G ($bcda$ 8 28 20)
G ($abcd$ 13 29 5)
G ($dabc$ 2 30 9)
G ($cdab$ 7 31 14)
G ($bcda$ 12 32 20)

3–раунд:

H ($abcd$ 5 33 4)

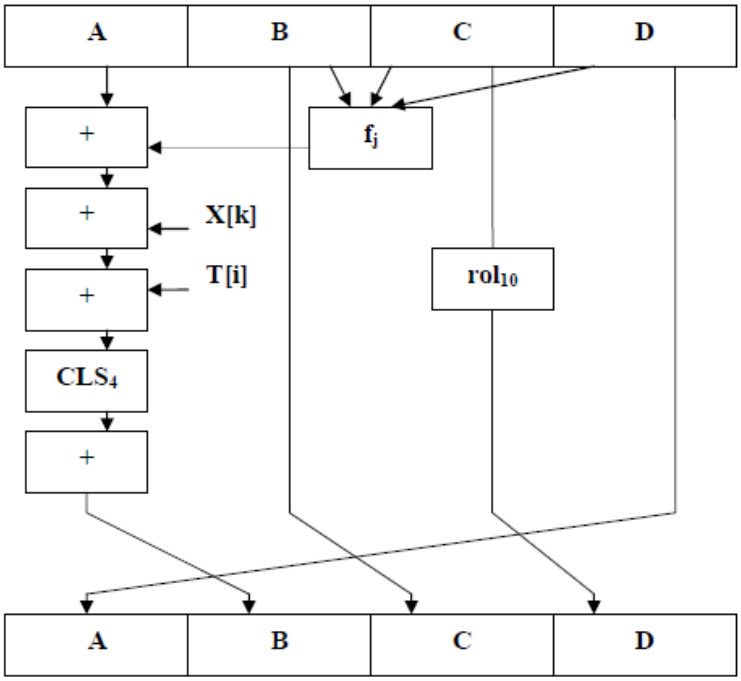
H (*dabc* 8 34 11)
H (*cdab* 11 35 16)
H (*bcda* 14 36 23)
H (*abcd* 1 37 4)
H (*dabc* 4 38 11)
H (*cdab* 7 39 16)
H (*bcda* 10 40 23)
H (*abcd* 13 41 4)
H (*dabc* 0 42 11)
H (*cdab* 3 43 16)
H (*bcda* 6 44 23)

4 раунд:

I (*dabc* 7 50 10)
I (*cdab* 14 51 15)
I (*bcda* 5 52 21)
I (*abcd* 12 53 6)
I (*dabc* 3 54 10)
(*cdab* 10 55 15)
I (*bcda* 1 56 21)
I (*abcd* 8 57 6)
I (*dabc* 15 58 10)
I (*cdab* 6 59 15)
I (*bcda* 13 60 21)
I (*abcd*
I (*dabc* 11 62 10)
I (*cdab* 2 63 15)
I (*bcda* 9 64 21)

Әрбір айналымда кіріс ағымдағы 512 биттік M_i блогы және 128 биттік буфер мәні *abcd* болып табылады. Алгоритмнің әрбір қадамынан кейін a, b, c, d *a, b, c, d* айнымалыларынан тұратын буфердің мазмұны жаңартылады. Әрбір айналымда бұрын анықталған T массивінің төрттен бір бөлігі де қолданылады.

$(i + 1)$ қадамы аяқталғанда, төртінші айналымның шығыс мәні бірінші айналымның кіріс мәніне қосылады (біздің H_i белгілеуімізде), нәтижесінде $(i + 1)$ қадам (біздің белгілеуімізде) шығады. Төменде 5.2 – суретте бір айналымның диаграммасын көрсетеді.



5.2 – сурет. MD5 қарапайым операциясы (раундтың бір қадамы)

6-қадам. Барлық M_i, i = 1,2, … K, 512 бит блоктарын өңдегеннен кейін өңдеудің шығысы 128–биттік хабарлама профилі болып табылады.

SHA–1 алгоритмі

SHA–1 алгоритмі еркін ұзындықтағы хабарламаны түрлендіреді 160 бит мәні. Хабарлама 512 биттік блоктарға бөлінген

$$M_1, M_2 \ \ldots, M_K,$$

мұндағы K мұндай блоктардың саны ($L = K \times 512$). Алгоритм $M_1, M_2 \ \ldots, M_K$ блоктарын дәйекті түрде түрлендіреді және әрқайсысы 20 қадамнан тұратын төрт айналымнан тұрады. Алгоритмнің ағымдағы қадамын $t, 0 \leq t \leq 79$ деп белгілейік. Содан кейін қадамдарды айналым бойынша бөлу келесідей:

1) бірінші раунд үшін $0 \leq t \leq 19$;

2) екінші айналым үшін $20 \leq t \leq 39$;

3) үшінші раунд үшін $40 \leq t \leq 59$;

4) төртінші айналым үшін $60 \leq t \leq 79$.

Өз кезегінде әрбір блок M_i, i = 1, 2, …,K, 16–ға бөлінеді 32 биттік қосалқы блоктар $X[0], X[1], \ldots, X[k], k = 0, 1, \ldots, 15$.

Әрі қарай, 16 блоктан $X[0], \ldots, X[k], k = 0,1, \ldots, 15,$, 80 блок W_t, $0 \leq t \leq 79$, 32 бит келесі (5.1), (5.2) формулалар бойынша құрастырылады.

$$W_t = M_t, \qquad (5.1)$$

$0 \leq t \leq 15$ мәндері үшін;

$$W_t = (W_t\text{-}3 \leq W_t\text{-}8 \leq W_t\text{-}14 \leq W_t\text{-}16) <<< 1, \quad (5.2)$$

$16 \leq t \leq 79$ мәндері үшін. Мұндағы «<<< 1» түрінің өрнегі сөздің 1 битке циклдік ығысуын білдіреді.

SHA–1 алгоритмі мыналарды пайдаланады:

Төрт тұрақты:

K_t = 0x5A827999,

бірінші раунд үшін ($0 \leq t \leq 19$);

K_t = 0x6ED9EBA1,

екінші раунд үшін ($20 \leq t \leq 39$);

K_t = 0x8F1BBCDC,

үшінші раунд үшін ($40 \leq t \leq 59$);

K_t = 0xCA62C1D6

төртінші раунд үшін ($60 \leq t \leq 79$).

Бес айнымалы A, B, C, D және E, әрқайсысы 32 бит өлшемі. Бұл айнымалылар тізбекті айнымалылар деп аталады және олар келесі мәндермен инициализацияланады:

A = 0x67452301,
B = 0xEFCDAB89,
C = 0x98BADCFE,
D =0x10325476,
E = 0xC3D2E1F0.

Алғашқы төрт мән *MD5*-те қолданылатын мәндерге сәйкес келеді.

Алайда, *MD5*-тен айырмашылығы, *SHA*–1 бұл мәндер кері байт реті форматында сақталады. Бұл сөздің ең маңызды байты төменгі мекенжай позициясында сақталатынын білдіреді.

Төрт функция:

$$f_t(b,c,d) = (b\text{^}c) \vee (b\text{^}d)$$

1–раунд үшін ($0 \leq t \leq 19$);

$$f_t(b,c,d) = b \oplus c \oplus d,$$

2–раунд үшін ($20 \leq t \leq 39$);

$$f_t(b,c,d) = (b\text{^}c) \vee (b\text{^}d)) \vee (c\text{^}d),$$

3–раунд үшін ($40 \leq t \leq 59$);

$$f_t(b,c,d) = b \oplus c \oplus d,$$

4–раунд үшін $(60 \leq t \oplus 69)$.

Әрбір функция үш 32 биттік b, c және d айнымалысын біреуіне түрлендіреді
32 биттік мән.

Қолданылған әдебиеттер тізімі

1. Рогауэй П., Шримптон Т. Криптографические хэш-функции: Теория и практика. / Рогауэй П. – М.: «Кудиц-образ», 2022. – 368 с.

2. Криптографические методы: Защита информаций, часть 1,2. Математические аспекты, под редакцией В.М. Фомичева учебник Д.А. Мельников

3. М.Я. Кельберт, Ю.М. Сухов «Теория информации и кодирования» Вероятность и статистика в примерах и задачах»