Лекция 14 ШИФРЛАУ, БӨГЕУІЛГЕ ТҰРАҚТЫ КОДТАУ ЖӘНЕ АҚПАРАТТЫ СЫҒУ

Көзден тұтынушыға ақпаратты беру барысында ақпаратқа әртүрлі жағымсыз факторлар әсер етеді. Криптографиялық әдістер ақпаратты бұзу әсерлердің бір түрінен ғана қорғайды - ақпаратты әдейі бұзудан немесе бұрмалаудан. Бірақ, тәжірибеде ақпаратты берген кезде байланыс желіде кездейсоқ бөгеуілдер болу мүмкін, аппаратураның қателіктері мен істен шығуы, деректер тасушының қирауы және т.б. Ақпаратты беру проблемаларды шешу үшін нақты байланыс жүйелерде түрлі әдістер мен тәсілдердің комплекстік пайдалануы қажет. Бұл бөлімде бөгеуілге тұрақты кодтарды пайдалану және деректерді сығу алгоритмдер қарастырылған.

14.1 Ақпаратты беру проблемалар және оларды комплекстік шешу

Көзден тұтынушыға ақпаратты беру барысында ақпаратқа әртүрлі жағымсыз факторлар әсер етеді. Криптографиялық әдістер ақпаратты бұзу әсерлердің бір түрінен ғана қорғайды - ақпаратты әдейі бұзудан немесе бұрмалаудан. Бірақ, тәжірибеде ақпаратты берген кезде байланыс желіде кездейсоқ бөгеуілдер болу мүмкін, аппаратураның қателіктері мен істен шығуы, деректер тасушының қирауы және т.б. Сонымен, нақты байланыс жүйелерде кездейсоқ әсерлеуден ақпаратты қорғау проблемасы бар.

Бұрың шифрланатын және берілетін хабарлардың негізгі типі мәтіндік хабар болатын, XXI ғасырда криптографиялық қорғау цифрлық видео- және сөз хабарларды, жер карталарды беруде, видеоконференцияны ұйымдастыруда қолданылады. Дәл сондықтан соңғы кезде орасан зор ақпараттық массивтерді шифрлау проблемасы туады. Интерактивті жүйелер үшін (телеконференция сияқты, аудио- немесе видеобайланыс) осындай шифрлау нақты уақыт тәртібінде жүзеге асырылып пайдаланушыға байқалмайтын болу керек.

Көрсетілген проблемаларды ақпарат теориясының жетістіктерін комплекстік түрде пайдаланып шешуге болады.

Ақпарат теориясында ақпарат түрлендірудің үш түрін ажыратады: криптографиялық шифрлау, бөгеуілге тұрақты кодтау және сығу (немесе компрессия). Бұрыңғы кейбір ғылыми жұмыстарда түрлендірудің үш түрін кодтау деп аталатын: криптографиялық кодтау, бөгеуілге тұрақты кодтау және тиімді кодтау (деректерді сығу).

Олардың ортақтығы – ақпарат, мағынасын емес, көрсету нысаның қандай болса да жолмен өзгертеді. Түрлі кодтаулардың айырмашылығы өткізілетін түрлендіру мақсатына тәуелді.

Мысалы, криптографиялық түрлендірудің мақсаты рұқсатсыз қатынаудан қорғау, аутентификациялау және әдейі өзгертуден қорғау. Бөгеуілге тұрақты кодтау ақпаратты беру және сақтау кезінде кездейсоқ бөгеуілден қорғау мақсатымен орындалады. Тиімді кодтау берілетің және сақталынатын деректердің көлемін минимизациялау мақсатымен жасалынады.

Тәжірибеде ақпарат түрлендірудің осы үш түрі әдетте бірлесіп пайдаланылады. Мысалы, кейбір программалық пакеттер шифрлаудың алдында өңделетін деректерді архивтейді. Басқа жағынан, ақпаратты берудің нақты жүйелерінің құрамында (жергілікті және ғаламдық желілер, CD немесе DVD-дискілер) ақпаратты қорғау жүйелер, бақылау және кездейсоқ қателіктерді түзету құралдар әрқашан бар болады.

Сонымен, криптографиялық шифрлау, бөгеуілге тұрақты кодтау және сығу бір бірін толықтырады, және олардың комплекстік пайдалануы берілетін ақпаратты сенімді қорғау үшін байланыс арналарды тиімді пайдалануға көмектеседі. Ақпаратты қорғау

жүйелерде пайдаланатын бөгеуілге тұрақты және тиімді кодтау теориясының негізгі қағидаларын қарап шығайық.

14.2 Бөгеуілге тұрақты кодтау

Айтылғандай, ақпаратты криптографиялық түрлендіру сұрақтары хабарларды бөгеуілге тұрақты кодтау сұрақтарымен тығыз байланысты. Ол бір (теориялық) жағынан криптографиялық шифрлауда және бөгеуілге тұрақты кодтауда бірдей ақпарат теориясының заңдарының пайдаланатына себепті. Басқа (тәжірибелік) жағынан ақпаратты жинау, сақтау және беру процестері бөгеуіл әсер ететін жағдайда өтеді, олар сақталынатын және өңделетін деректерді бұрмалау мүмкін. Сондықтан, осындай қателіктерді табатын және түзететін әдістерді әзірлеу мен пайдалану өте актуальды. Математикалық көзқарасынан бұл есеп **бөгеуілге тұрақты кодтарды** синтездеуге жатады.

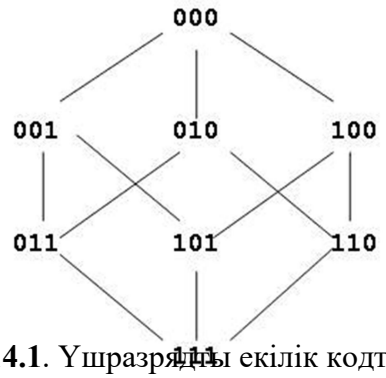
Бөгеуілге тұрақты кодтауды және хабарларды сығу сұрақтарды талқылауда код ұғымы енгізіледі. Жалпы **код** дегеніміз – бұл белгілердің жинағы және ақпаратты белгілер жиынтығы ретінде көрсете алатын ережелер жүйесі. Мүмкін болатын белгілердің кез келген қатары **кодты сөз** деп аталады. Мысалы, екілік 1100 санды екілік 4-разрядты кодты сөз деп санауға болады.

Бөгеуілге тұрақты кодтаудың ортақ идеясы - барлық мүмкін болатын кодты сөздерден рұқсат етіледі бәріне емес, тек олардың кейбіреуіне. Мысалы, жұптылық бойынша бақылауы бар кодта тек бірліктер жұп саны бар сөздерге рұқсат етіледі. Қателік мүмкін болатын сөзді мүмкін болмайтын сөзге айналдырады және сондықтан табылады.

Бөгеуілге тұрақты кодтар *блокты* және *үйіртқіліге* бөлінеді. Блокты кодтар ақпаратты тұрақты ұзындығы бар фрагментке бөледі және олардың әрбіреуін жеке өңдейді, ал үйірткілі блоктар деректермен үздіксіз ағын сияқты жұмыс істейді.

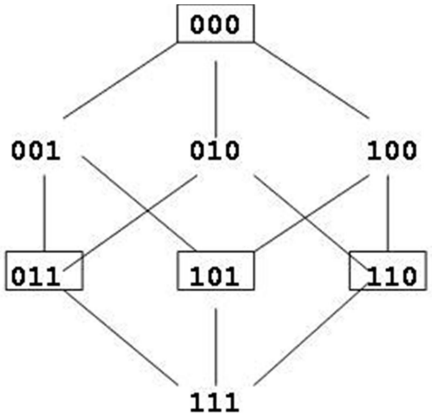
Блокты кодтар минимал кодтық ара қашықтықпен сипатталады. Екі кодты сөздер арасындағы **Хэмминг бойынша қашықтық** деп разрядтар санын атайды, оларда сөздер әртүрлі. Мұнда **минимал кодтық ара қашықтық** ретінде кез келген қос түрлі кодты сөздер үшін ең кіші Хэмминг бойынша қашықтықты таңдайды.

Мысалы, біз тек үшразрядты екілік сөздерді пайдаланайық. Барлығы осындай кодты сөздер сегіз болу мүмкін. Кодты сөздердің бір бірінен айырмашылығы тек бір бірлікке тең болса оларды **көршілес** деп атайды. Мысалы, кодты сөздер 101 мен 111 – көршілес, өйткені олар орта разрядпен ғана ерекшеленеді, ал 101 мен 110 сөздер - көршілес емес, себебі оларда ерекшеленеді соңғы екі разряды. Барлық үшразрядты екілік комбинацияларды бейнелеп көрсетейік және көршілес кодты сөздерді сызықтармен қосайық. Онда біз 14.1 суреттегідей схеманы аламыз. Бөгеуілге тұрақты емес әдеттегі кодтын минимал кодтық ара қашықтығы бірге тең.



Сурет 14.1. Үшразрядты екілік кодты сөздер

Барлық үшразрядты екілік сөздерді хабарды беру үшін пайдалану кезінде, олардың бәрі мүмкін болатын деп саналады. Жұптылық шарты бойынша бақылауды қолданайық. Онда тек рамкамен бөлінген бірліктердің жұп саны бар сөздер рұқсат етілетін болады (14.2 сур. қара).



Сурет 14.2. Жұптылық бойынша бақылауда мүмкін болатын үшразрядты кодты сөздер

Жұптылық бойынша бақылауы бар кодтың мүмкін болатын сөздерінің минимал ара қашықтығы екіге тең (14.2 суреттен көрінеді, ешқандай рамкадағы екі кодты сөздер сызықтармен қосылмаған, яғни көршілес емес). Дәл осы себептен кодты сөздегі жеке қателік бұл сөзді жарамайтынға айналдырады.

Бөгеуілге тұрақтылыққа жету үшін кәдімгі кодпен салыстырғанда сөз ұзындығын үлкейту қажет. Берілген мысалда тек екі разряды ғана ақпараттық болып табылады. Олар төрт әртүрлі сөзді құрайды. Үшінші разряд бақылаушы болады және тек мүкін сөздердің ара қашықтығын үлкейту үшін қызмет етеді. Ақпаратты беруге бақылаушы разряд қатыспайды, өйткені ол ақпаратқа сызықты тәуелді болады. Жұптылық бойынша бақылауы бар коды деректер блоктарда жеке қателіктерді табуға мүмкіндік береді. Бірақ ол екі ретті қателіктерді табалмайды, себебі екі ретті қателік кодты сөзді мүмкін болатын сөздер аралығынан өткізіп және оны басқа мүмкін болатынға айналдырады.

Сонымен, кодтың қателіктерді тауып түзету қабілеті болу үшін, оның артықшылықсыздығынан бас тарту қажет. Ол үшін екілік символдардың мүмкін комбинациялар жиының екі ішкі жиынға бөледі: мүмкін және мүмкін емес кодты сөздер. Бөлуді мүмкін сөздер арасындағы минимал кодты қашықтықты үлкейту оймен орындайды. Бұл жағдайда кез келген бір ретті қателік мүмкін кодты сөзді мүмкін емеске айналдырады, сол себептен оны табуға болады.

Әрине, қосымша бақылау разрядтың енгізілуі кодталған ақпаратты сақтауға немесе беруге арналған шығындарды көбейтеді. Ал пайдалы ақпараттың нақты көлемі өзгеріссіз қалады. Бұл жағдайда бөгеуілге тұрақты кодтың **артықтығы** туралы айтуға болады. Оны формальді түрде бақылау разряд санының кодты сөздің жалпы разрядтар санына қатынасы деп анықтауға болады.

Бақылау разрядтар ақпаратты бермейді, сондықтан олар пайдасыз. Бақылау разрядтың салыстырмалы саның **бөгеуілге тұрақты кодтың артықтығы** Q деп атайды

$$Q = \frac{k}{n} 100\%,$$

мұндағы n – блоктағы екілік разрядтың жалпы саны; k – бақылау разрядтар саны.

Мысалы, қарастырылған жұптылық бойынша бақылауы бар үшразрядты кодтың артықтығы тең:

$$Q = \frac{k}{n} 100\% = \frac{1}{3} 100\% \approx 33\%$$

Артықтық кодтың өте маңызды сипаттамасы, артықтықтың тым өсуі жағымсыз. Ақпарат теориясының маңызды есебі – минимал артықтығы бар кодты синтездеу, олар қойылған табу және түзету қабілетін қамтамасыз етеді.

Суреттейтің мысал ретінде бір ретті қателікті табуға және түзетуге мүмкіндік беретің қарапайым кодтардын біреуін қарастырайық – Хэмминг кодын. Ұзындығы n кодты сөзде ақпараттық k және бақылау m разряды бар. Бұрмаланған i-ші разрядтын түзетуі – бұл қабылданған кодты сөзді, i-ші разрядында бірлігі бар 0…010…0 түрлі вектормен қосу (модулі 2 бойынша).

n-разрядты кодты сөз үшін бір ретті қателікке сай келетін осындай n векторлар, және қателіксіз сөзді қабылдау жағдайға сәйкес келетін бір нөлдік векторы болады. Сонымен, m бақылау разряды n+1 қателік векторды құрастыруға қамтамасыз ету керек, яғни $n \leq 2^m$-1 теңсіздік орындалу керек. Нәтижесінде Хэмминг коды деп аталатын кодты (2^m-1, 2^m-1-m) аламыз.

m=3 сәйкес ең қарапайым жағдай (7,4)-кодты құрастырады, оны былай синтездеуге болады. Әрбір қателік векторға реттік нөмір – синдром сәйкестендіреміз (кесте 14.1). Осы кезде нөлдік қателік векторға нөлдік синдром сай болады.

Кесте 14.1. Қателік векторы және оған сәйкес синдромдар

a_6	a_5	a_4	a_3	a_2	a_1	a_0	s_2	s_1	s_0
1	0	0	0	0	0	0	0	1	1
0	1	0	0	0	0	0	1	0	1
0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	1	1	1
0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0

s_i функцияны модулі 2 бойынша үйірткі ретінде қарастыра отырып, аламыз:

$$s_0 = a_0 \oplus a_3 \oplus a_5 \oplus a_6$$
$$s_1 = a_1 \oplus a_3 \oplus a_4 \oplus a_6$$
$$s_2 = a_2 \oplus a_3 \oplus a_4 \oplus a_5$$

s_i функциялар оларды құрастыратын разрядтын біреуінде қателік болғанда бірге айналады, және нөлге – қателік болмағанда. Осы талапты қамтамасыз етуге болады, егер разрядтардын бір бөлігі арнайы түрде құрастырылатын болса. a_0, a_1, a_2 разрядтарды басқа разрядтардың модулі 2 бойынша үйірткісі ретінде қарастыруға болады, олар келесі теңдеуге қатысады:

$$a_0 = a_3 \oplus a_5 \oplus a_6$$
$$a_1 = a_3 \oplus a_4 \oplus a_6$$
$$a_2 = a_3 \oplus a_4 \oplus a_5$$

Табылған тәуелсіздіктер берілген ақпараттық сөздер бойынша кодты сөздерді генерациялаға, және де қабылданған кодты сөздер үшін синдромды есептеуге мүмкіндік береді.

Бастапқы ақпараттық сөз 1101 тең болсын, яғни a_6=1, a_5=1, a_4=1, a_3=1. Бөгеуілге тұрақты Хэмминг (7,4)-кодының мүмкін кодты сөзіне оны түрлендіру үшін бақылау разрядтарды есептейік:

$$a_1 = 1 \quad 0 \quad 1 = 1,$$

$$\oplus \quad \oplus$$
$$a_1 = 1 \oplus 0 \oplus 1 = 0,$$
$$a_2 = 1 \oplus 0 \oplus 1 = 0.$$

Сонымен, бақылау разрядтарды еске алғанда кодты сөз 1101001 тең болады.

Егер беру немесе сақтау процесінде сөз бұрмаланбаған болып қалса, онда оның синдромы $s_0 \ldots s_2$ сәйкес тең болады:

$$s_0 = 1 \oplus 1 \oplus 1 \oplus 1 = 0,$$
$$s_1 = 0 \oplus 1 \oplus 0 \oplus 1 = 0,$$
$$s_2 = 0 \oplus 1 \oplus 0 \oplus 1 = 0.$$

Тек нөлден ғана тұратын синдром қателік жоқ дегенді көрсетеді және нөлдік қателік векторға сәйкес.

Беру немесе сақтау барысында сыртқы факторлар әсерінен кодты сөздің жеке разряды бұрмаланған болсын. Мысалы, 1101001 сөздің орнына 1001001 сөз қабылданды. Бұл жағдайда синдром нөлге тең болмайды: $s_0 \ldots s_2$ сәйкес тең болады:

$$s_0 = 1 \oplus 1 \oplus 0 \oplus 1 = 1,$$
$$s_1 = 0 \oplus 1 \oplus 0 \oplus 1 = 0,$$
$$s_2 = 0 \oplus 1 \oplus 0 \oplus 0 = 1.$$

Синдром 101 қателік векторға 0100000 сай болады, бұл кезде бірлік қателік болған разрядты көрсетеді. Оны түзету үшін қабылданған бұзылған кодты сөзді қателік векторы мен модулі 2 бойынша қосу жеткілікті.

Хэмминг (7,4)-кодының артықтығын есептейік:

$$Q = \frac{k}{n} \cdot 100\% = \frac{7-4}{7} \cdot 100\% \approx 43\%$$

Бұл өте үлкен мән. Тәжірибеде одан күрделі кодтар жиі қолданылады, олар бөгеуілге тұрақты жақсығырақ сипаттамаларын қамсыздандырады.

14.3 Деректерді сығу принциптері

Жоғарыда айтылғандай, шифрлауға деректерді дайындаудың маңызды есебінің бірі олардың артықтығын азайту және қолданатын тілдің статистикалық заңдылықтарын тегістеу. Артықтықты жарым-жартылай жоюына деректерді сығу арқылы жетеді.

Ақпаратты сығу бұл бастапқы хабарды бір кодты жүйеден басқаға түрлендіру процесі, нәтижесінде хабар мөлшері азаяды. Ақпаратты сығу үшін арналған алгоритмдарды екі үлкен топқа бөлуге болады: *шығынсыз сығу* (қайтымды сығу) және *шығыны бар сығу* (қайтымсыз сығу).

Қайтымды сығу декодтаудан кейін деректерді абсолют дәл қалпына келтіруді айтады және ол кез келген ақпаратты сығу үшін қолданылу мүмкін. Ол әрқашан ақпараттық құрылымын жоғалтпай ақпараттың шығу ағын көлемінің азаюына әкеледі. Онан әрі, шығу ағыннан, қайталанатын немесе декомпресстау алгоритмы көмегімен, кіру ағынды алуға болады, ал қалпына келтіру процесі *декомпрессия* немесе *түйіншекті шешу* деп аталады. Шығынсыз сығу мәтін, орындалу файлдар, сапалы дыбыс және графика үшін қолданылады.

Қайтымсыз сығуда әдетте сығу дәрежесі салыстырмалы жоғары, бірақ декодталған деректердің бастапқыдан кейбір ауытқулар болу мүмкін. Тәжірибеде декомпрессиядан кейін бастапқы ақпаратты дәл қалпына келтіруді талап етпейтін бірнеше міндеттер бар: дыбыс, фото- немесе видео бейнелер. Мысалы, мультимедиалық ақпарат форматта JPEG пен MPEG қайтымсыз сығу қолданылады. Қайтымсыз сығу криптографиялық шифрлаумен бірге әдетте пайдаланбайды, себебі криптожүйеге қойылатын негізгі талап дешифрланған деректердің бастапқыға толық ұқсастығы.

Деректерді қайтымсыз сығудың кейбір ең кең таралған тәсілдерін қарап шығайық.

Ең танымал қарапайым жол және ақпаратты қайтымсыз сығу алгоритмы – бұл

тізбектер сериясын кодтау (Run Length Encoding – RLE). Осы жолдың мағынасы – қайталанатың байт тізбектерін немесе серияларын бір кодтайтын байт-толтырушыға және олардың қайталау санын есептеуішке ауыстыру. Осындай барлық ұқсас әдістердің проблемасы - нәтижелі байттар ағынында кодталған серияны басқа кодталмаған байттар тізбектерінен айыру. Проблеманы кодталған тізбектердің басына белгілер қойып шешуге болады. Осындай белгілер ретінде кодталған серияның бірінші байттағы биттердің сипатты мәні және кодталған серияның бірінші байт мәні болу мүмкін. RLE әдістің кемшілігі төмен сығу дәрежесі немесе аз сериялар саны бар файлды кодтау бағасы.

Ақпаратты бір қалыпты кодтауда хабарға, оның пайда болу ықтималдығына қарамастан, бірдей бит саны бөлінеді. Сонымен қатар, егер жиі кездесетін хабарларды қысқа кодты сөздермен кодтаса, ал сирек кездесетіндерді - ұзынырақ сөздермен кодтаса, онда берілетің хабарлардың жалпы ұзындығы азаяды деп болжау жасауға болады. Мұнда айнымалы ұзындығы бар кодты сөзден тұратын кодты пайдалану қажеттілігіне байланысты проблемалар туады. Осындай кодты құруға бірнеше жолдар бар.

Тәжірибеде кең қолданылатылардың біреуі *сөздік әдістер*, олардың негізгі өкілдері Зива мен Лемпел алгоритмдары. Оның негізгі идеясы - кіру ағынның фрагменттері («сөйлемдер») мәтінде бұрын болған орынға көрсететін сілтегішпен ауыстырылады. Әдебиетте осындай алгоритмдар LZ сығу алгоритмы деп белгіленеді.

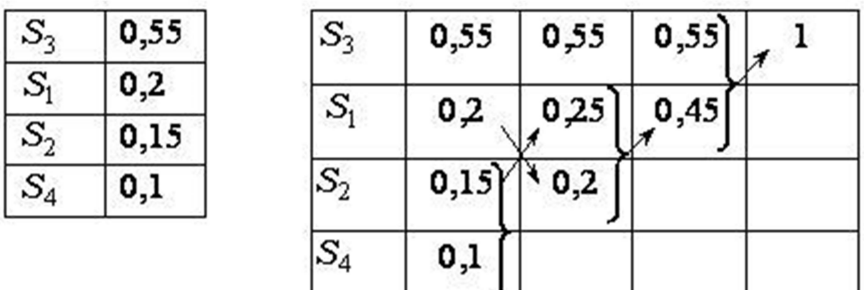
Сол сияқты әдіс мәтін құрылымына тез лайықтанады және қысқа функционалды сөздерді кодтай алады, себебі олар мәтінде жиі кездеседі. Жаңа сөздер мен сөйлемдер бұрын кездескен сөздердің бөліктерінен де құрастырылу мүмкін. Сығылған мәтіннің декодтауы тікелей жүргізіледі, - сілтегішті сөздіктен алынған дайын сөйлеммен ауыстырады. Тәжірибеде LZ-әдісі жақсы сығуды жеткізе алады, оның маңызды қасиеті декодердің өте жылдам жұмысы.

Ақпаратты сығудың тағы бір жолы кодер мен декодеры қарапайым аппараттық жүзеге асыруы бар Хаффман коды болып табылады. Алгоритмның идеясы мұндай: хабарға символдардың кіру ықтималдығын біле отырып, айнымалы ұзындығы бар бүтін бит санынан тұратын кодты құру процедурасын бейнелеуге болады. Үлкен ықтималдығы бар символдарға қысқа код беріледі, ал сирек кездесетін символдарға – ұзынырақ. Осы себептен кодты сөздің орта ұзындығы қысқарады және сығудың тиімділігі өседі. Хаффман кодында бірегей префиксы (кодты сөздің басы) бар, сондықтан оларды айнымалы ұзындығына қарамастан бір мәнді декодтауға болады.

Хаффман классикалық кодын синтездеу процедурасын жүргізу үшін хабар көзінің статистикалық сипаттамасы туралы априорды ақпарат болу қажет. Басқаша айтқанда, хабарды құрайтын қандай да болса символдардың пайда болу ықтималдығы құрастырушыға белгілі болу керек. Хаффман кодының синтезін қарапайым мысал көмегімен қарастырайық.

Ақпарат көзі төрт әртүрлі символды $S_1 \ldots S_4$ генерациялайтын болсын, олардың шығу ықтималдығы $p(S_1)=0{,}2$, $p(S_2)=0{,}15$, $p(S_3)=0{,}55$, $p(S_4)=0{,}1$. Символдарды шығу ықтималдығы азаю бойынша сұрыптайық және кесте түрінде көрсетейік (сурет 14.3, а).

Кодты синтездеу процедурасы үш негізгі кезеңнен тұрады. Бірінші кезеңде кесте жолдары бүктеледі: ең кіші ықтималдығы бар символдарға сәйкес екі жол жиынтық ықтималдығы бар бір жолға ауыстырылады, осыдан кейін кесте қайта реттеледі. Үйірткі, кестеде жиынтық ықтималдығы бірге тең, бір жол ғана қалғанша жүргізіледі (сурет 14.3, б).

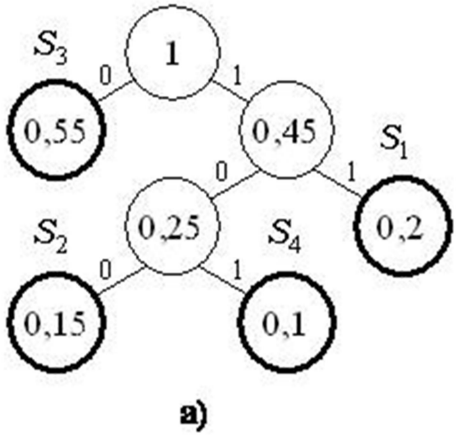


Сурет 14.3. Хаффман кодтаудың бірінші кезеңі

Екінші кезеңде бүктеулі кесте бойынша кодты ағашты құру жүзеге асырылады (сур. 14.4, а). Ағаш кестенің соңғы бағаннан бастап салынады.

Ағаш түбірін соңғы бағанда орналасқан бірлік құрайды. Қарастырылған мысалда бұл бірлік ағаштың екі торап түрінде көрсетілген 0,55 және 0,45 ықтималдықтан жасалынады. Олардың біріншісі S_3 символға сәйкес, және осы тораптың онан әрі бұтақтануы болмайды.

Екінші 0,45 ықтималдықпен белгіленген торап, үшінші деңгейдің ықтималдықтары 0,25 және 0,2 екі торабымен қосылады. 0,2 ықтималдық S_1 символға сәйкес, ал 0,25 ықтималдығы S_2 символдың пайда болуының ықтималдығынан 0,15 және S_4 символдың пайда болуының ықтималдығынан 0,1 құрылады.



S_1	11
S_2	100
S_3	0
S_4	101

б)

Сурет 14.4. Хаффман кодтаудың екінші кезеңі

Кодты ағаштың жеке тораптарын қосатын қабырғалар 0 және 1 цифрымен нөмірленеді (мысалы, сол жақ қабырғалар – 0, ал оң жақтылары - 1).

Үшінші, соңғы кезеңде, кесте құрылады, онда көз символдары және оларға сәйкес Хаффман кодының кодты сөздері салғастырылады. Осы кодты сөздер қабырғаларды белгілейтін цифрларды оқудан пайда болады, қабырғалар ағаш түбірінен сәйкес символға жолды құрайды. Қарастырылған мысал үшін Хаффман коды оң жақ кестеде көрсетілген түрге келеді (сур. 14.4, б).

Бірақ классикалық Хаффман алгоритмның бір маңызды кемшіліг бар. Сығылған хабар мазмұнын қалпына келтіру үшін кодтаушы пайдаланған жиілік кестесін декодер білу керек. Демек, сығылған хабардың ұзындығы жиілік кестенін ұзындығына өсу керек. Ол деректер алдында жіберілу қажет, сондықтан хабарды сығуға жұмсалған күштер босқа кету мүмкін.

Статикалық Хаффман кодтаудың басқа варианты - кіру ағынды қарап шығу және жиналған статистика негізінде кодтауды құрастыру. Мұнда файл бойынша екі өту қажет болады – біреуі статистикалық ақпаратты қарау және жинау үшін, екіншісі – кодтау үшін. Статикалық Хаффман кодтауда кіру символдарға (түрлі ұзындығы бар бит тізбектері) айнымалы ұзындығы бар бит тізбектері сәйкестікке қойылады (олардың коды). Әрбір символдың код ұзындығы оның жиілігінің теріс таңбалы екілік логарифмына пропорционал алынады. Ал барлық кездескен түрлі символдардың ортақ жиынтығы ағын алфавиті болып табылады.

Басқа да әдіс бар – адаптивті немесе динамикалық Хаффман кодтауы. Оның жалпы принципі – кіру ағынның өзгерісіне байланысты кодтау схеманы өзгерту. Осындай жолда бір өтімді алгоритм алынады және пайдаланған кодтау туралы ақпаратты айқын түрде сақтауды қажет етпейді. Адаптивті кодтау статикалыққа қарағанда үлкен сығу дәрежесін береді, өйткені кіру ағын жиілігінің өзгерістері толығырақ еске алынады.

Хаффман әдістері жеткілікті жоғары жылдамдық және сығудың орташа жақсы сапасын береді. Бірақ Хаффман кодтауында минимал артықтық болады, егер әрбір символ екі биттен {0, 1} тұратын жеке тізбекпен кодталатын болса. Осы әдістің негізгі кемшілігі - сығу дәрежесінің символдар ықтималдықтарының 2-нің кейбір теріс дәрежесіне жақындығына тәуелділігі, өйткені әрбір символ бүтін бит санымен кодталады.

Әбден басқа шешімді арифметикалық кодтау ұсынады. Бұл әдіс кіру ағынды қалқыма нүктесі бар бір санға түрлендіруге негізделген. Арифметикалық кодтау кіру алфавиттің символдарын шығынсыз буып-түюге мүмкіндік береді, егер осы символдардың жиілік үлестіруі белгілі болса.

Арифметикалық кодтау әдісі арқылы сығуда қажетті символдар тізбегі кейбір [0, 1) аралығындағы екілік бөлшек ретінде қарастырылады. Сығу нәтижесі осы бөлшек жазудың екілік цифрлар тізбегімен ұсынылады. Әдіс идеясы келесі: бастапқы мәтін осы бөлшектің жазуы болып қарастырылады, мұнда әрбір кіру символ шығу ықтималдығына пропорционал салмағы бар «цифр» болып табылады. Осымен ағында символ шығуының минимал және максимал ықтималдығына сәйкес интервал түсіндіріледі.

Қарастырылған әдістер деректердің қайтымды сығуын қамтамасыз етеді. Тәжірибеде олардың программалық пен аппараттық іске асыруы қолданылады. Сығу коэффициенті сығылатын ақпарат типіне байланысты 20-40% болу мүмкін.

Сонымен, криптографиялық шифрлау, бөгеуілге тұрақты кодтау және сығу бір бірін аздап толықтырады, олардың комплекстік қолдануы берілетін ақпаратты сенімді қорғау үшін байланыс арналарды тиімді пайдалануына көмектеседі.

Негізгі терминдер

Артықтық - қәдімгі бөгеуілге тұрақты емес кодпен салыстырғанда, кодты сөздің ұзындығы қаншаға үлкейгенің көрсететін бөгеуілге тұрақты кодтың сипаттамасы. Көп бөгеуілге тұрақты кодтар үшін артықтықты бақылау разрядтар санның кодты сөздің жалпы разряд санына қатынасы ретінде анықтауға болады.

Код – белгілер жиынтығы және ақпаратты осы белгілер жинағы түрінде ұсынатын ережелер жүйесі.

Кодты сөз – пайдаланатын ережелер жүйесіне сәйкес мүмкін болатын белгілердің кез келген қатары.

Минимал кодтық ара қашықтық – кодты құрайтын кез келген қос түрлі кодты сөздер үшін ең кіші Хэмминг бойынша қашықтық.

Бөгеуілге тұрақты код – хабарларды сақтау мен берудегі қателіктерді табуға және түзетуге мүмкіндік беретін код.

Хэмминг бойынша қашықтық – кодты сөздердің разрядтар саны, оларда сөздер әртүрлі.

Ақпаратты сығу – бастапқы хабарды бір кодты жүйеден басқаға түрлендіру процесі, нәтижесінде хабар мөлшері азаяды.

Көршілес кодты сөздер – тек бір разряд мәнімен ерекшеленетің кодты сөздер.

Сұрақтар

1. Ақпаратты комплексті қорғау үшін ақпараттың қандай түрлендіру түрі пайдаланады?
2. Хабарларды бөгеуілге тұрақты кодтаудың негізгі принциптері қандай?
3. Хэмминг кодымен хабарларды кодтаған кезде қателік синдромы қалай пайдаланады?
4. Хабарлардың сығуын қамтамасыз ететін код мысалдарын келтіріңіз.
5. Хаффман әдісімен хабарларды кодтаған кезде хабарлардың сығуы нелікпен қамтамасыз етіледі?
6. Хаффман кодты сөзі қалай құрастырылады?
7. Шығыны бар сығу алгоритмды қандай деректер типі үшін пайдалану орынды? Бұл немен байланысты?