Ибрагимов О.М., Құрақбаев Ж.С.

ҚОЛДАНБАЛЫ КРИПТОГРАФИЯ

Оқу құралы

ЭВЕРО
Алматы, 2022

ӘОЖ 616.832 (075.8)
КБЖ 57.14 я73
И 39

Пікір жазғандар:

Е.А. Нысанов – ф.-м.ғ.д., М.Әуезов атындағы Оңтүстік Қазақстан мемлекеттік университеті «Информатиканы оқытудың теориясы және әдістемесі» кафедрасының профессоры;

Б.Ш. Мырзахметова – т.ғ.к., Оңтүстік Қазақстан мемлекеттік педагогикалық институтының «Информатика» кафедрасының меңгерушісі;

Р.І. Ибрагимов – п.ғ.д., Қазақстан инженерлі-педагогикалық Халықтар достығы университетінің «Информатика және математика» кафедрасының профессоры.

Ибрагимов О.М.

И 39 Қолданбалы криптография: оқу құралы / Ибрагимов О.М., Құрақбаев Ж.С. – Алматы: Эверо, 2022. – 112 бет.

ISBN 978-601-327-153-2

Оқу құралында қолданбалы криптографияның жалпы мәселелері, яғни ақпаратты қорғау және оның мәселелері, криптографиялық шифрлардың тарихы, математикалық модельдері, шифрлеу алгоритмдері, криптографиялық қасиеттері мен шифрлеу стандарттары қаралған. Объектке бағытталған бағдарламалау ортасында криптожүйе құру технологиясы көрсетілген. Оқу құралында осы аталған мәселелер мысалдармен көрнекі түрде сипатталған. Сонымен қатар оқу құралында берілген жеке тапсырмалар мен тест сұрақтары, оның мазмұнын түсінуге көмектеседі.

Оқу құралы университеттің 6М060200 – «Информатика» мамандығы бойынша оқитын магистранттарға арналған.

ӘОЖ 616.832 (075.8)
КБЖ 57.14 я73

ISBN 978-601-327-153-2

Мазмұны

КІРІСПЕ

Ақпаратты қорғау қажеттілігі ежелгі цивилизацияларда, яғни ежелгі Римде, Араб, Үнді елдерінде, Мысырда туындаған. Олар ақпаратты қорғаудың төмендегі үш тәсілін қолданған: ақпаратты физикалық тәсілмен қорғау; стеганография; криптология. Бірінші тәсілде тек қана күшпен қорғалатын мүмкіндіктер, яғни ақпаратты тасушыны физикалық қорғау, құжатты арнайы курьермен тасымалдау т.б. Екінші тәсіл «стеганография» (steganographia, «құпия жазу») деп аталып, онда ақпаратты жасыратын әр түрлі құралдар, үй бұйымдары және заттардан пайдаланылған. Мысалы, химиялық жолмен алынатын «тамаша сиялар». Мұндай сиялармен жазылған ақпарат арнайы өңдеуден өткенде, мысалы қыздырылғанда, немесе химиялық қышқыл жағылғанда ғана көрінеді. Құпия ақпарат ішкі киімге, бет орамалға, галстукке, тағы басқа заттарға енгізілген. Ежелгі Грекияда ақпарат тақтайшалардың бетіне жазылған, содан кейін тақтайша балауызбен капталған. Сонымен жазу балауыздың астында қалып, көрінбейді. Қытайда ақпаратты жібек мата жолағына жазған. Ақпаратты жасыру үшін мәтіні бар жолақтан шарик жасалған және балауызбен капталған. Шабарман қауіп төнген жағдайда оны жұтып қоятын болған. Үшінші «криптология» (cryptology, «құпия ғылым») деп аталатын тәсілде, ақпарат мәтінінің әріптері басқа әріптермен немесе цифрлармен ауыстырылған, яғни шифрленген. Нәтижеде құпия мәтін мағынасыз, түсініксіз, әріптер тізбегіне айналады. Шифрленген мәтінді оқу үшін арнайы «кілт» қажет болады.

Қазіргі таңда, яғни ақпараттық технологиялардың қарқынды дамыған заманында, ақпаратты қорғаудың негізгі екі тәсілі қолданыста: стеганография және криптология. Цифрлық ерекше белгілерді енгізу негізінде құрылған стеганографиялық қорғану технологиясы аудио және видео деректерді заңсыз қолданудан қорғаудың және олардың көшірмелерін рұқсатсыз таратуды бақылаудың негізгі бағыты болып есептеледі. Авторлық құқық нысандарына көрінбейтін цифрлық ерекше белгі (digital watermark) енгізіледі. Аталған технология авторлық құқықтың заңдылығын немесе керісінше заңсыздығын дәлелдеуге мүмкіндік береді. Ал, криптология ақпаратты түрлендіру, шифрлеу және оны қорғаумен айналысады. Ол шартты түрде криптография және криптоталдау

деп екі бағытқа бөлінеді. Осы бағыттарға және терминдерге қысқаша тоқталайық.

Криптография (cryptography, «құпия жазу») ақпаратты заңсыз пайдаланудан қорғау мақсатымен оны шифрлеу және дешифрлеу әдістері туралы ғылым. Ол ақпаратты шифрлеудің математикалық әдістерін зерттеумен айналысады. Осы ғылыммен айналысатындарды криптограф деп атайды.

Криптоталдау шифрленген ақпаратты, оның кілтін білмей-ақ шифрден шығару мәселесімен, яғни қолданыстағы шифрлеу жүйесін бұзып ашумен айналысады. Сонымен, криптоталдау шифрленген ақпаратқа заңсыз қол жеткізуге бағытталған ғылым. Мұндай криптоаналитиктерді кейде кодтарды бұзушы (breaker), шабуылшы (attacker) және ұры (sneaker) деп те атайды. Сонымен, криптографтар ақпарратты жасыруға, ал криптоаналитиктер оны бұзып ашуға ұмтылады.

Криптографиялық жүйе (криптожүйе) – шифрлеу алгоритмі, сондай-ақ, алуан түрлі кілттердің, шифрлеу және дешифрлеу әдістерінің жиынтығы.

Криптографиялық алгоритм (*шифр, шифрлеу алгоритмі*) деп шифрлеу және кері шифрлеу үшін қолданылатын математикалық алгоритмдерді (функцияларды) атайды.

Жалпы, криптографияның даму тарихын үш кезеңге бөлуге болады. Бірінші кезең ақпаратты қолмен және қарапайым түрлендіру құрылғыларымен түрлендіруге байланысты. Бұл кезең ежелгі цивилизациялардан бастау алады. Шифрлеу әдістері біздің дәуірімізге дейінгі 4-ші мың жылдықта жазумен бір уақытта пайда болған. Құпия түрде ақпарат алмасу әдістері Мысыр, Шумер, Вавилон, Ассирияда және Қытай сияқты көптеген ертедегі қоғамдарда бір-бірінен тәуелсіз пайда болған.

Екінші кезең алдымен механикалық, сонан соң электромеханикалық және электрондық шифрлеу құрылғыларының пайда болуымен байланысты. Бұл кезеңді телеграфтық, одан кейін электрондық шифрлейтын құрылғылардың пайдалана басталуы деп есептеуге болады. Мұндай құрылғылар көптеп жасалды және олар практикада кең қолданыла бастады, құпияланған байланыс тораптары құрылды.

Үшінші кезең XX ғасырдың 70-ші жылдарында байланыс тораптарының, электрондық поштаның және ауқымды ақпараттық

жүйелердің дамуымен, ұлттық стандарт алгоритмдердің енгізілуімен байланысты. Осы кезде американдық математиктер Диффи және Хеллман абоненттерді алдын ала құпия кілттермен жабдықтауды талап етпейтін байланыс арнасын ұйымдастыруды ұсынған. Осының нәтижесінде криптографиялық жүйелер пайда бола бастады. Бұл кезең шифрленген байланыс арналарының автоматтандырылған жүйелерінің пайда болуымен сипатталады.

Ұсынылып отырған оқу құралында қолданбалы криптографияның жалпы мәселелері, яғни ақпаратты қорғау және оның мәселелері, криптографиялық шифрлардың тарихы, математикалық модельдері, шифрлеу алгоритмдері, криптографиялық қасиеттері мен шифрлеу стандарттары қаралған. Объектке бағытталған бағдарламалау ортасында криптожүйе құру технологиясы көрсетілген. Оқу құралында осы аталған мәселелер мысалдармен көрнекі түрде сипатталған. Сонымен қатар оқу құралында берілген жеке тапсырмалар мен тест сұрақтары, оның мазмұнын түсінуге көмектеседі.

1 АҚПАРАТТЫ ҚОРҒАУ ЖӘНЕ ОНЫҢ МӘСЕЛЕЛЕРІ

Ақпаратты қорғау. Қазіргі таңда Интернет арқылы ақпаратты тасымалдау, қаржылық операцияларды орындау, несие карточкаларын қолдану, электрондық дүкеннен тауар сатып алу, жабық ақпараттық ресурстарға кіру, т.б. іс-әрекеттер міндетті түрде ақпараттың қауіпсіздігін талап етеді. Сонымен қатар, кәсіпорындарда, банкте, жекеменшік мекемелерде дербес компьютер негізінде тиімді ақпараттық құрылымды құру, ақпаратты қорғау мәселесінің бірі болып есептеледі. Бұл мәселе ақпараттық деректерді заңды және физикалық түрде қорғау, компьютерлік жүйені ұйымдастыру, техникалық және бағдарламалық қорғау, жүйелік бағдарламаны арнайы бағдарламалар арқылы қорғау сияқты мәселелер жиынынан тұрады.

Жалпы, ақпаратты қорғау мәселесі, ақпараттық қауіпсіздікті камтамасыз етуге бағытталған іс-шаралар кешенінен тұрады. Ал, іс жүзінде ақпаратты қорғау деп деректерді сақтау, өңдеу және тасымалдау үшін қолданылатын ақпараттық жүйенің жұмыс үрдісін, тиімділігін түсінеді. Сонымен, ақпаратты қорғау дегеніміз – ақпаратты рұқсатсыз өзгертудің, түрлендірудің, ұрлаудың, бұрмалаудың, жоғалтудың, көшірмесін жасаудың алдын алу үшін жүргізілетін іс-шаралар кешені. Ақпараттық қауіпсіздікті камтамасыз ету үшін қойылған шектеулерді қанағаттандыруға арналған ұйымдастырушылық, бағдарламалық және техникалық әдістер мен құралдардың кешенінен тұрады.

Ақпаратты қорғаудың бірінші мәселесі – ақпарат бүтіндігін сақтау және соған байланысты басқару сұрақтарын қамтиды. Ақпарат бүтіндігін сақтау мәселесі ұйымдастыру және техникалық аспектілерден тұрады. Сонымен, ұйымдастыру аспектісі келесі ережелерден тұрады:

- ақпарат физикалық түрде қорғалуы тиіс;
- ақпарат сақталатын орын техникалық қауіпсіздік және өрт қауіпсіздігі ережелеріне сай болуы тиіс;
- ақпарат бөтен адамдар кіруге тиым салынған жерде сақталуы тиіс;
- магнитті жинақтауыштар техникалық ережеге сай сақталуы тиіс.

• өте қажетті ақпараттың көшірмесі бірнеше магниттік жинақтауыштарда сақталуы тиіс;

• ақпарат мазмұны мен мағынасы бойынша жіктеліп, бөлек атаулермен сақталуы тиіс;

Сол сияқты ақпарат бүтіндігін сақтау мәселесінің техникалық аспекті әр-түрлі шектеулер түрімен тікелей байланысты. Бұл аспект мәліметтер қорын басқару жүйесінің құрылымына сай келуі және тұтынушыға тиімді болуы қажет. Сонымен, техникалық аспект келесі ережелерден тұрады:

• бастапқы және соңғы ақпарат арасындағы байланысты сақтау үшін белгілі мәндерді жаңалауды шектеу;

• ақпарат атрибуттарының кейбір мәндерінің санын шектеу;

• кейбір диапазондар үшін айнымалылардың мәнін сақтауды шектеу.

Ақпараттық қауіпсіздік. Ақпараттық қауіпсіздік — мемлекеттік ақпараттық ресурстардың, сондай-ақ ақпарат саласында жеке адамның құқықтары мен қоғам мүдделері қорғалуының жай-күйі. Сонымен, ақпарат қауіпсіздігі - бұл компьютерлік жүйенің жұмыс барысында мәліметтерді жоғалтпауы, ақпарат шығынына, бөтен адамдардың ақпаратты ұрлауына, бұрмалауына, жалған ақпаратпен ауыстыруына жол бермеу, қойылатын қауіпсіздік талаптарына сәйкес тұтынушыға толығымен жеткізуді камтамасыз ету. Ақпараттық жүйелерде ақпарат шығыны, қандай да бір құпияны, мысалы, коммерциялық, әскери, медициналық, тұлғалық және т.с.с. ашу болып саналады.

Сонымен қатар, тек құпия ақпаратты ғана қорғау емес, құпия емес ақпаратты да қорғау қажет. Өйткені, құпия емес ақпаратты жоғалтудың салдары құпия ақпарат шығынына немесе ақпараттың бұрмалануына әкеліп соғуы мүмкін. Ақпараттық жүйелердің жұмыс жасау бағыты, ауқымы мен аймағына қарай ақпарат шығыны әр түрлі деңгейдегі келеңсіз нәтижелерге алып келеді, мысалы, өз ара сенімсіздіктен бастап өте үлкен қаржылық шығынға дейін.

Интернет пен бұқаралық ақпарат құралдарында дәл осы аталған мәселелерге көптеген мысалдар келтіріліп, алаңдаушылық білдірілуде. Әсіресе мұндай қылмыстар банк және бизнес құрылымдарына қатысты ақпараттық жүйелерінде көп кездеседі.

Интернеттегі аналитикалық сайттардағы мәлімет бойынша (http://www.vedomosti.ru/) компьютерлік қылмыстар нәтижесінде әлем бойынша банктердің көретін шығыны жыл сайын $6 миллиард долларды құрайды екен. Бұл жайында 2013 жылдың қазан айында өткен Securing Our eCity және Digital Crimes Consortium конференциясында американдық құзырлы мекемелердің өкілдері мәлімдеді. Ал, Group-IB эксперттерінің Вирустық зерттеу орталығы, ESET және LETA компанияларының аналитиктерімен бірге жүргізген зерттеулері нәтижелері бойынша компьютерлік қылмыстар шығыны $7 миллиард деп бағалануда.

Сонымен, қазіргі кезде ақпарат саласында жеке адамның құқықтары мен қоғам мүдделері, ақпараттық жүйелердің стратегиялық объекттерде қауіпсіздікті қамтамасыз етуіне және басқарумна тәуелді болып отыр. Мұндай обьекттерге телекоммуникация мен банк жүйелері, құпия ақпаратты сақтау, тасымалдау және өңдеуге арналған криптожүйелер жатады. Бұл жүйелердің сенімді және тиімді жұмыс жасауы үшін, олардың тұтастығы мен қауіпсіздігін қамтамасыз ету қажет.

Ақпаратты қорғау және оның мәселелері. Компьютерлік жүйелерде ақпаратты қорғаудың қауіпсіздік мәселелерін негізгі үш топқа бөлуге болады:

- ақпаратты ұрлау - тасымалданатын ақпарат бүтіндігі сақталады, дегенмен ақпарат иесінің авторлық құқықтары бұзылады;
- ақпаратты бұрмалау - тасымалданатын ақпарат өзгертіледі, модификацияланады немесе басқасымен ауыстырылады;
- авторлық ақпараттың ауыстырылуы - тасымалданатын ақпараттағы автордың реквизиттері жалған ақпаратпен алмастырылады.

Осы аталған мәселелердің нәтижесі келеңсіз жағдайларға алып келуі мүмкін. Мысалы, біреу сіздің атыңыздан хабар жіберуі (спуфинг), Web-сервердегі электрондық дүкен арқылы бұйыртпа жіберуі немесе несие карточкаларының реквизиттерін ұрлауы мүмкін.

Ақпаратты қорғаудың қауіпсіздік мәселесіне, «қауіпсіздік» терминінің төмендегі касиеттерін атауға болады:

• аутентификация - бұл компьютерлік жүйенің тұтынушыны танып алу үрдісі және тұтынушыға оның барлық құқықтары мен міндеттері жіберіледі;

• құпиялық - компьютерлік жүйеге рұқсат етілмеген кіруді болдырмау;

• тұтастық - ақпараттың мазмұны, бірыңғайлығы және бүтіндігін, ақпаратты бұзудан және заңсыз өзгертуден қорғау, әртүрлі жағдайларда сақтап қалу;

• сенімділік - әр түрлі құпиялық дәрежедегі ақпаратты тұтынушылар тобының қатынас құру құқығын бұзбай бір уақытта өңдеуін қамтамасыз ету;

• қол жетімділік (тиімділік) – ақпаратқа қол жеткізуге тиісті рұқсаты бар тұтынушыларды кажетті ақпаратпен кедергісіз қамтамасыз ету.

Ақпаратты қорғау әдістері. Компьютерлік жүйенің қауіпсіздігіне есептеуіш құралдары арқылы амалға асырылатын қауіптерді қарап шығайық:

1. Компьютерлік жүйенің жұмысына физикалық түрде араласу. Бұл қауіп түріне компьютерлік жүйелердің есептеуіш техникасын бұзу (ақпарат сақталатын дискерді немесе өзге де құрылғыларды ұрлау, жарамсыз ету, жұмысына кедергі жасау) және компьютерлік жүйелердің бағдарламалық өнімдеріне бөтен адамның рұқсатсыз кіруі жатады. Мұндай қауіпті қимылдарға қарсы қолданылатын тәсілдер ұйымдастырушылық (арнайы күзет қою, компьютерлік жүйелерге кіруді қадағалау, тұтынушыларға арнайы журнал арнау) түрінде ұйымдастырылады.

2. Компьютерлік жүйенің жұмысына техникалық түрде араласу. Бұл қауіп түріне компьютерлік жүйелердегі ақпараттың қауіпсіздігі мен тұтастығын техникалық құралдар көмегімен бұзу (ақпарат тасымалдаушы каналдарға электромагниттік әсер ету, есептеуіш құрылғыларынан электромагниттік сәулелену арқылы ақпарат қабылдау, сымсыз байланыс немесе Wi-Fi сигналдарын пайдалану) жатады. Мұндай қауіптерден сақтану үшін ұйымдастырушылық шараларынан басқа, техникалық (есептеуіш құрылғыларын сәулеленуден сақтау, ақпарат тасымалдаушы каналдарды бақылау) және бағдарламалық тәсілдер қолданылады.

3. Компьютерлік жүйенің жұмысына бағдарлама арқылы араласу. Бұл қауіп түріне бағдарламалық өнімдер көмегімен компьютерлік жүйелердің бағдарламалық компоненттерін бұзу, жою және кедергі жасау жатады. Мұндай өнімдерді - бұзушы бағдарламалық өнімдер деп атауға да болады. Оларға компьютерлік вирустар, троян аттары, хакерлік шабуылдар, зиянды плагиндер кіреді. Мұндай бағдарламаларға қарсы бағдарламалық және техникалық қорғау тәсілдері қолданылады.

Ақпаратты қорғау мәселесін әрі қарай жалғастырайық. Айта кетейік, ақпараттық жүйенің бір компьютерде жұмыс істеу үрдісінен бірнеше компьютерлер біріккен компьютерлік желіде жұмыс істеуге өту барысында ақпараттың қауіпсіздігі мәселесін қиындататын келесі себептер бар:

- компьютерлік желіде бірнеше тұтынушылардың жұмыс істеуі және олардың ауысып отыруы, тұтынушының атауы (логин) мен пароль арқылы ақпаратты бөтен тұтынушылардан қауіпсіздендіру тәсілі жеткіліксіз;
- компьютерлік желіде кездесетін көптеген бұзушы бағдарламалық өнімдердің ақпараттық жүйенің компоненттеріне кіріп кету мүмкіндігі;
- жұмыс барысында туындайтын техникалық және бағдарламалық ақаулардың орын алуы.

Ақпараттық жүйеде кез-келген басқа компьютермен қосымша байланысу немесе Интернет желісіне қосылу жаңа қауіпсіздік мәселесін туындатады, оған қоса компьютерлік вирустардың ақпараттық жүйеге кіру мүмкіндігін көбейтеді. Желідегі әрбір перифериялық құрылғы жоғарғы жиілікті электромагниттік сәулендірудің потенциалдық көзі болып табылады. Жүйеге электромагниттік сәулелендіруден басқа, контактілі емес коаксиалдық кабелді электромагнит әсер етеді. Кабельдік сымдарды қосудың бұл типтерін кабелдерьді физикалық жалғауда қолдану мүмкін. Егер ақпараттық жүйеге кіру үшін пароль белгілі болса, онда кейбір тұтынушылар компьютерлік желіге файл-сервер арқылы немесе алыстатылған жұмыс орындарының бірінен кіруі мүмкін. Сол себепті компьютерлік желіден алыста орналасқан ақпаратты сақтау құрылғыларынан да ақпарат жоғалуы мүмкіндігі

бар. Сонымен қатар, компьютерлік желіде ақпаратты қорғаудың бір тәсілі ретінде, арнайы «шуыл генераторын» да қолдануға болады.

Сонымен, компьютерлік желілерде ақпарат қауіпсіздігін қамтамасыз ететін құралдарды үш топқа бөлуге болады:

• *техникалық құралдар*, бұлар ақпарат сақтау орындарына физикалық түрде кіруге кедергі жасайды (кілттер, терезедегі темірторлар, сигнализация және т.б.). Техникалық құралдың құндылығы субъектілік факторларға тәуелсіздігімен және жоғары модификацияға беріктігімен анықталады. Оның кемшіліктері – материалдардың сапасына байланыстылығы, қымбат тұратындығы, артықша шығындар және т.б.

• *бағдарламалық құралдар*, оларға тұтынушыларды идентификациялауға арналған компоненттер, ақпаратты шифрлеу және дешифрлеу алгоритмдері, антивирустық бағдарламалар, уақытша файлдарды жою, жүйені қауіпсіздендіруге арналған тесттік бақылау және т.б. жатады. Бағдарламалық құралдың құндылығы ретінде бағдарламалардың эмбебаптығы, сенімділігі, криптоберіктігі, қарапайымдылығы, модификациялауға мүмкіншілігі және т.б. айтуға болады. Оның кемшіліктері – желінің физикалық мүмкіндіктерін шектеуі, файл-сервердің немесе алыстатылған жұмыс орындарының ресурстарын толық қолданбауы, кездейсоқ немесе келісілген өзгертулерге жоғары сезімталдығы және дербес компьютердің платформасына тәуелділігі.

• *ұжымдық құралдар*, сонымен қатар ұжымдық-техникалық және ұжымдық-құқықтық құралдар. Ұжымдық құралдардың құндылығы деп оларды құрудың қарапайымдылығын, әр түрлі мәселелерді шешуге құқықтық мүмкіндігін, желідегі өзгерістерге жоғары сезімталдығын, модификациялау мүмкіндігін айтуға болады. Кемшіліктері – субъектілік факторларға тәуелділігі және тұлғалардың өз ара әлсуметтік байланысы.

Ақпаратты қорғаудың техникалық құралдары. Техникалық нысандар мен ақпаратты қорғаудың техникалық құралдары физикалық тұйық ортаны құруға арналған. Бұл үшін төмендегі көрсетілген іс-шаралар амалға асырылады:

• ақпарат өңделетін және сақталатын бөлмелерге физикалық тоскауыл құралдарын орналастыру, кодтық құлыптар, сигнализациялар және терезелерге темірторлар орнату;

• ақпарат өңделетін және сақталатын бөлмелерден электромагниттік сәуленің таралуына жұқа металдармен немесе арнайы пластмассалармен кедергі жасау;

• құпия ақпаратты өңдеуге арналған есептеуіш техникасын электр энергиясын жеке бөлек алуын немесе бірыңғай электр жүйесінен арнайы желілік фильтрлер арқылы алуын қамтамасыз ету;

• ақпараттық арналарға қашықтықтан рұқсатсыз кіруге бөгет жасау үшін сұйық кристалданған мониторлар, арнайы лазерлік принтерлерді қолдануды қамтамасыз ету;

• автоматтық қорғаныс құралдарын есептеуіш техникасының сыртқы қабатына жапсыра қолдану және олардың ашылып-жабылуына бақылау орнату.

Ақпаратты қорғаудың бағдарламалық құралдары. Ақпаратты қорғаудың бағдарламалық қорғаныс құралдары мен тәсілдері дербес компьютерлер мен компьютер желілерінде кеңінен қолданылады. Олар төменде көрсетілген функционалдық іс-әрекеттерді орындайды:

• тұтынушылардың ақпараттық қорларға енуін реттеу, бақылау және шектеу;

• ағымдағы үрдістер мен тотенше оқиғаларды тіркеу және талдау;

• ақпаратты криптографиялық алгоритмдермен қорғау;

• тұтынушылар мен үрдістердің құпиялығы, тұтастығы, сенімділігі, қол жетімділігі, аутентификациясы және т.б.

Ақпаратты қорғаудың әкімшілік-ұйымдастыру құралдары. Әкімшілік-ұйымдастыру қорғаныс құралдарына ақпараттық жүйелердің функционалдық үрдістеріне кіруді регламенттеу, тұтынушылардың қызметтік міндеттерін сатылық регламенттеу және т.б. жатады. Олардың мақсаты ақпараттық қауіпсіздікті әкімшілік-ұйымдастыру ережелері бойынша заңды түрде қорғау болып табылады. Қазіргі таңда жиі тараған әкімшілік-ұйымдастыру қорғаныс құралдарына төмендегі тәсілдер кіреді:

• есептеу техникасы және көмекші ақпаратты өңдеу және сақтау құралдары орналасқан жерде кіріп шығуды бақылау - рұқсат беру тәсілін қолдану;

• арнайы куәліктерді дайындау және онымен өз қызметкерлерін толық қамтамасыз ету;

• құпия ақпаратты өңдеуге және сақтауға қатысы бар қызметкерлерді сынақтан өткізу;

• құпия ақпаратты өңдеуге, сақтауға немесе беруге тек қызмет бабымен арнайы рұқсаты бар қызметкерлерді ғана жіберу;

• ақпарат тасымалдаушы құралдарын (DVD-диск, флеш-диск) және тіркеу журналдарын сейфте немесе басқа адамдар кіре алмайтын, қорғалатын жерлерге сақтау;

• есептеу техникасы орнатылған және ақпаратты өңдеуге арналған бөлмеге тыңдағыш немесе электромагниттік сәулендіру құралдарын жасырын қоюға қарсы қорғанысты ұйымдастыру;

• құпия ақпарат туралы ресми құжаттардың қолдану аясы мен жойылу тәртібін ұйымдастыру;

• қызметкерлердің қызмет бабына байланысты есептеу техникасымен және ақпарат тасымалдаушы құралдарымен жұмыс істеуіне арналған ережелер жасау;

Брандмауэрлер. Proxy-серверлер. Ақпаратты қорғаудың арнайы бағдарламалық құралдарының санкциясыз кіру қауіпсіздігі орнатылған желілік амалдық жүйелерге қарағанда мүмкіндіктері көп. Шифрлеу алгоритмдерінен басқа ақпаратты қорғау бағдарламалары да көп қолданылады. Солардың ішінен төменде көрсетілген екі жүйе көп қолданыста. Олар ақпараттық ағынды шектеуге көмектеседі.

• Firewalls - *брандмауэрлер* Firewalls сөзі ағылшын тілінен аударғанда «отты дуал» деген мағынаны береді, яғни компьютер және Интернет желісі арасында арнайы аралық фильтр-бағдарламалар. Брандмауэр компьютердің қауіпсіздігін күшейтеді. Ол өзі аркылы өтетін желілік-транспорттық деңгейлер трафигін бақылап, фильтрлейді. Сонымен қатар, компьютерге аумақта желіден келіп түсетін ақпаратты қадағалайды, компьютердің қорғаныс шебін құрады. Бір сөзбен айтқанда, брандмауэрді компьютер мен Интернет арасындағы кеден бекеті деп түсінуге болады. Бұл бағдарлама желідегі санкциясыз кіруді азайтады, бірақ

оны түбімен жоя алмайды. Компьютерге локальдық немесе Интернет желісінен басқа біреу байланысуға сұраныс жасаса, брандмауэр оны бақылауға алады. Брандмауэр тұтынушыдан сұранысты блоктау немесе рұқсат беруді сұрайды. Егер тұтынушы рұқсат берсе, байланыс орнатылады, ал кері жағдайда сұраныс орындалмайды.

• Proxy-servers. Proxy сөзі ағылшын тілінен аударғанда «өкіл» деген мағынаны береді, яғни локальдық және Интернет желісі арасындағы барлық желілік-транспорттық деңгейлер трафигі толығымен шектеледі. Ақпарат ағыны маршрутталмайды, ал жергілікті желі Интернет желісімен арнайы делдал-сервер (Proxy-server) арқылы байланысады. Басында компьютер прокси-сервермен байланысады, содан соң қажетті ақпараттық ресерсқа сұраныс жасайды. Прокси-сервер өз кезегінде Интернет желісімен байланысады. Жалпы, прокси-серверлер төмендегі мақсаттар үшін қолданылады:

○ локальдық желі компьютерлерінен Интернет желісіне қосылуды қамтамасыз ету;

○ деректерді кэштеу; егер Интернеттегі кейбір ресурстарға жиі сұраныс жасалатын болса, осы ресурстардың нұсқасын прокси-серверде сақтап, трафикті азайту;

○ деректерді сығу; прокси-сервер Интернет желісінен алынған ақпаратты сығып, тұтынушыға ықшам түрде жеткізеді;

○ локальдық желіні сырткы байланыстан сақтау; локальдық желінің компьютерлері Интернет желісімен тек қана прокси-сервер арқылы байланыс жасайды, яғни олар Интернеттен көрінбейді;

○ локальдық желіден Интернет желісіне байланысты бақылау; Прокси-сервер локальдық желіден жасалған сұраныстарды фильтрлеп отырады, тұтынушыға кейбір сырткы ресурстармен байланысуға рұқсат бермейді, қажет емес деп саналатын ақпаратты өткізбейді.

2 КРИПТОГРАФИЯЛЫҚ ӘДІСТЕР ТАРИХЫ

Ежелгі симметриялық криптографиялық әдістер әдетте екіге бөлінеді, мәтіндегі әріпті басқа әріппен «ауыстыру» және әріптердің «орнын ауыстыру».

Юлий Цезарь шифры. Ауыстыру әдісіне мысал ретінде Гай Юлий Цезарь (Ежелгі Рим, б.э.д. I ғасыр) қолданған әдісті келтірейік. Цезарь (лат. Caesar) осы әдісті пайдаланғаны тарихтан белгілі. Сондықтан осы әдісті Цезарь шифры немесе Цезарь коды деп атайды.

Әдісті қазақ тілінің әліпбиі үшін түрлендіріп талдайық. Ол үшін әліпби әріптерін тізбектеп жазамыз. Ал қатардың астына әліпби әріптерін 3 әріпке оң жаққа қарай жылжытып жазамыз. Пайда болған қатарды Öåçàðü шифрының әліпбиі деп аламыз. Ал А, Ә, Б әріптерін Öåçàðü шифрының әліпбиінің соңына тіркеп қоямыз.

А, Ә, Б, В, Г, Ғ, Д, Е, Ё, Ж, З, И, Й, К, Қ, Л, М, Н, Ң, О, Ө, П, Р, С, Т, У, Ұ, Ү, Ф, Х, Һ, Ц, Ч, Ш, Щ, Ы, І, Ь, Ъ, Э, Ю, Я

В, Г, Ғ, Д, Е, Ё, Ж, З, И, Й, К, Қ, Л, М, Н, Ң, О, Ө, П, Р, С, Т, У, Ұ, Ү, Ф, Х, Һ, Ц, Ч, Ш, Щ, Ы, І, Ь, Ъ, Э, Ю, Я, А, Ә, Б

Құпия мәтінді шифрлеуде А әрпін В әрпімен ауыстырылады, ал Ә әрпі Г әрпімен ауыстырылады, әрі қарай осылай жалғаса береді. Егер осы әдіспен ОҚМУ сөзін шифрлесек, онда РНОФ шифрмәтін аламыз. Цезарь шифрында кілт ретінде әліпби әріптерін 3 әріпке жылжыту алынады. Мәтінді оқу үшін әріптерді сол жаққа қарай 3 әріпке жылжытып, мәтінді шифрден шығару керек.

Юлий Цезарь шифрының басқа варианттарын пайдалануға да болады. Мысалы, Юлий Цезарьдың мұрагері Цезарь Август шифрдегі кілт ретінде әліпби әріптерін тек қана 1 әріпке жылжытуды қолданған. Сол сияқты, кілт ретінде шифрдың төмендегі қатарындағы әріптердің кез келген орналасуынан пайдалануға болады. Онда шифр варианттарының жалпы саны 42! тең.

Қарапайым орын ауыстыру шифры. Шифрды талдау үшін мысал қарастырайық. Бүтін оң таңбалы сан таңдайық, мысалы 5. Онда екі қатары, бес бағаны бар қарапайым кесте аламыз. Кестенің бірінші қатарына 5-ке дейінгі сандарды өсу тәртібінде, ал екінші қатарға сандарды кез-келген тәртіпте жазамыз (кесте 1).

Кесте 1 – орын ауыстыру шифры

1	2	3	4	5
3	1	5	2	4

Құпия мәтін ретінде «ОҢТҮСТІК ҚАЗАҚСТАН МЕМЛЕКЕТТІК УНИВЕРСИТЕТІ» сөйлемін шифрлейік. Сөйлемді бос орынсыз бес-бестен бөліп жазамыз. Егер соңғы бөлікте әріптер 5-ке толмаса, онда оны кез-келген әріптермен толтыруға болады.

ОҢТҮС ТІККА ЗАҚСТ АНМЕМ ЛЕКЕТ ТІКУН ИВЕРС ИТЕТІ

Мәтінді шифрлеуде әр бір бөліктегі әріптерді кесте бойынша ауыстырамыз, яғни бірінші әріптің орнына үшінші әріпті, екінші әріптің орнына бірінші әріпті, үшінші әріптің орнына бесінші әріпті, т.с.с. жазамыз. Пайда болған мәтінді бос орынсыз, қосып жазамыз. Онда төмендегідей шифрмәтін аламыз:

ТОСҢУКТАІҚҚЗТАСМАМНЕКЛТЕЕКТНІУЕИСВРЕИІТТ

Келтірілген орын ауыстыру шифрында кілт ретінде таңдалған 5 саны және кестедегі сандардың орналасуы алынады. Шифрленген мәтінді оқу үшін осы амалдар кері қарай орындалады.

Сцитала шифры. Ең алғашқы орын ауыстыру шифрларын пайдаланған криптографиялық құрылғылардың бірі ретінде Сцитала құрылғысын атауға болады. Бұл құрылғы ежелгі Спарта елінде (б.з.д. V ғасыр) пайда болған.

Сцитала құрылғысымен шифрлеу келесі тәсілмен жүргізілген. Диаметрі алдынан белгілі цилиндрге пергаментті лента оралып, оған құпия мәтін жазылады. Сонан соң лента цилиндрден шешіліп, хабарды алушыға жіберіледі. Пергаментті лентада мәтін шашырай

орналасады. Мәтінді оқу үшін лентаны дәл сондай диаметрлі цилиндрге орау керек.

Мысалы: «ИНФОРМАТИКА» сөзін цилиндр осі бойынша көлденең жазсақ «ИРИ НМК ФАА ОТ» шифрмәтінін алуға болады (сурет 1). Шифрмәтінді қайта ашу үшін цилиндр диаметрін білу қажет, яғни кілт ретінде лента ұзындығы мен цилиндр диаметрі алынады.

Сцитала шифры бүгінгі таңда «маршрутты орын ауыстыру» деп аталатын шифрдың дербес бір жағдайы болып есептеледі. Бұл шифрдың мағынасы былай. Төртбұрышты n қатар және m бағаннан тұратын кестеге L ұзындықтағы мәтін жазылады ($L < n+m$). Бос қалған ұяшықтар кез келген әріптермен толтырылады. Ал егер $L > n+m$ болса, онда екінші кесте толтырылады. Шифрмәтін осы кестеден алдын-ала келісілген маршрут бойынша алынады. Шифрдің кілті ретінде n, m сандары және көрсетілген маршрут алынады.

Сурет 1 - Сцитала құрылғысы

Полибий шифры. Қарапайым ауыстыру шифрларының ең алғашқыларының бірі Полибий квадраты болып есептеледі. Грек тарихшысы Полибий (б.э.д. III ғасыр) шифрлеу мақсатында 5*5 өлшемді грек алфавиті әріптерімен толтырылған кестені ойлап шығарды (кесте 2).

λ	ε	ϑ	ω	ϒ
ρ	ς	δ	σ	ο
μ	η	β	ξ	τ
ϕ	π	θ	α	k
χ	ν	_	φ	I

Полибий квадраты әйтеуір орналасқан 24 әріптен және бос орыннан (пробел) тұрады. Шифрлеу кезінде осы квадраттан ашық мәтіннің кезекті әрпін тауып және шифрмәтінге осы әріптен төмен тұрған сол бағандағы әріпті жазып отырған. Егер ашық мәтіннің әрпі кестеде ең соңғы жолда тұрған болса, онда шифрмәтін үшін дәл сол бағандағы ең жоғарғы әріп алынған.

Полибий шифрын қазіргі латын әліпбиіне қолдансақ, онда келесі кестені аламыз (кесте 3). Кестеде I, J әріптері бір ұяшыққа жазылады. Құпия мәтінді шифрлеуде әріптің орнына Полибий квадратынан осы әріп тұрған ұяшықтың координатасын жазамыз.

Кесте 3 – Полибий квадраты (латын әліпбиімен)

	A	B	C	D	E
A	A	B	C	D	E
B	F	G	H	I,J	K
C	L	M	N	O	P
D	Q	R	S	T	U
E	V	W	X	Y	Z

Полибий шифрын қазақ әліпбиіне қолдансақ, онда келесі 6x7 өлшемді кестеден пайдаланамыз (кесте 4).

Кесте 4 – Полибий квадраты (казақ әліпбиімен)

	A	B	C	D	E	F	G
A	А	Ә	Б	В	Г	Ғ	Д
B	Е	Ё	Ж	З	И	Й	К
C	Қ	Л	М	Н	Ң	О	Ө
D	П	Р	С	Т	У	Ұ	Ү
E	Ф	Х	Һ	Ц	Ч	Ш	Щ
F	Ы	І	Ь	Ъ	Э	Ю	Я

Мысалы, құпия мәтін ретінде «УНИВЕРСИТЕТ» сөйлемін шифрлейік. Онда У әрпінің орнына DE, Н әрпінің орнына CD, И әрпінің орнына BE, т.с.с. жазамыз. Онда төмендегідей шифрмәтін аламыз:

DE CD BE AD BA DB DC BE DD BA DD

Ал шифрмәтінді шифрдан шығаруда қос әріптің орнына кестедегі сәйкес бір әріп жазылады. Айта кетейік, Полибий шифрында кілт жоқ, себебі әліпби әріптері ізбектеліп жазылады.

Полибий шифрының қиындатылған вариантында әріптер кестеге кез-келген тәртіпте жазылады. Мұндайда кейбір қиындықтар пайда болады. Тәртіпсіз жазылған әліпби әріптерін есте сақтау қиын, сондықтан әріптер жазылған кестені кілт ретінде сақтау керек. Онда құпиялыққа қауіп төнуі мүмкін. Сондықтан кілт ретінде құпия сөз - пароль ұсынылады. Жеңіл жатталатын пароль кестеге әріптерді қайталамай жазылады. Ал қалған ұяшықтарға әліпби әріптері, парольде кездесетін әріптерді қалдырып үздіксіз енгізіледі. Мысалы, кілт ретінде «ШЫМКЕНТ» сөзін алайық. Онда кесте төмендегі түрде толтырылады (кесте 5).

Кесте 5 – Полибий квадраты («ШЫМКЕНТ» кілт сөзімен)

	A	B	C	D	E	F	G
A	Ш	Ы	М	К	Е	Н	Т
B	А	Ә	Б	В	Г	Ғ	Д
C	Ё	Ж	З	И	Й	Қ	Л
D	Ң	О	Ө	П	Р	С	У
E	Ұ	Ү	Ф	Х	Һ	Ц	Ч
F	Щ	І	Ь	Ъ	Э	Ю	Я

Құпия мәтін ретінде «УНИВЕРСИТЕТ» сөйлемін шифрлейік. Онда төмендегідей шифрмәтін аламыз:

DG AF CD BD AE DE DF CD AT AE AT

Сонымен, мұндай жағдайда әріптер жазылған кестені өзімен бірге алып жүрмесе де болады. Оны кез келген уақытта парольмен қайта құруға болады.

Түрме шифры. Полибий шифрының бүгінгі таңда қолданылатын бір түрі – түрме шифры. Бұл шрифтте әріптердің орнына цифрлар қойылған кестеден пайдаланады. Түрме шифрын қазақ әліпбиіне қолдансақ, онда келесі 6х7 өлшемді кестені аламыз (кесте 6).

Кесте 6 – Түрме шифры (қазақ әліпбиімен)

	1	2	3	4	5	6	7
1	А	Ә	Б	В	Г	Ғ	Д
2	Е	Ё	Ж	З	И	Й	К
3	Қ	Л	М	Н	Ң	О	Ө
4	П	Р	С	Т	У	Ұ	Ү
5	Ф	Х	Һ	Ц	Ч	Ш	Щ
6	Ы	І	Ь	Ъ	Э	Ю	Я

Мысалы, құпия мәтін ретінде «БАХЫТ» сөзін шифрлейік. Онда Ь әрпінің орнына 13, А әрпінің орнына 11, Х әрпінің орнына 52, т.с.с. жазамыз. Онда төмендегідей шифрмәтін аламыз:

13 11 52 61 44

Жалпы алғанда, түрме шифры ақпаратты қайта кодтау (шифрде кілт жоқ, сондай-ақ құпия пароль қолданылмайды) болып табылады. Ақпарат *байланыс каналы* (түрме қабырғасы) арқылы әріпбы сигналмен (қабырғаны шерту) жіберіледі (Морзе әліпбиі сияқты)

Сиқырлы квадраттар. Орта ғасырларда Европада криптография қара магия, алхимия, оккультизм сияқты ағымдармен қатар дами бастады. Осы кезде сиқырлы квадраттардан пайдалану басталды. Алғашқы рет мұндай квадраттар Қытайда қолданылғаны туралы деректер бар.

Сиқырлы квадраттар деп 1-ден бастап саналатын сандармен толтырылған, әр бағанның, жолдың, диагональдың қосындысы бір санды беретін квадрат кестелерді атайды. Мысалы, 4х4 өлшемді квадратқа 1-ден 16-ға дейінгі сандар жазылады. Квадраттың сиқырлығы - әр бағанның, жолдың, диагональдың қосындысы 34-ке тең болып шығатыны.

Шифрленетін мәтін сандардың өсу тәртібіне сәйкес сиқырлы квадратқа жазады. Соңынан, осы квадраттың катары бойынша жазылып, кажетті шифрмәтін алынады. Мысалы: «ДЕКАН КЕЛМЕЙДІ» құпия сөзін шифрлеу үшін 4х4 өлшемді сиқырлы квадраттан пайдаланайық (кесте 7):

Кесте 7 – Сиқырлы квадрат (4х4)

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

1	К	Е	І
Н	Е	Й	Л
М	К	Е	Д
А	І	І	Д

Онда төмендегідей шифрмәтін аламыз:

1КЕІ НЕЙЛ МКЕД А11Д

Шифрмәтінді ашып оқу үшін мәтін сиқырлы квадратқа жазылып, ашық мәтін сандардың өсу тәртібінде оқылады. Айта кетейік, сиқырлы квадраттарда кілт болмайды. Бұл шифр - қарапайым орын ауыстыру шифрының бір түрі. Бірақта олардың көп қолданылатыны «сиқырлығында» деп есептеледі.

Тритемий шифры. Ақпаратты қорғау бойынша кең таралған әдістердің бірі Тритемий шифры. Шифр Тритемий кестесіне негізделген. Бұл кестенің қазақ әліпбиіндегі нұсқасын қарастырайық. Кестенің алғашқы қатарына әліпби әріптері толық жазылады. Әрі карай әліпби әріптері, екінші қатарға екінші әріптен бастап, үшінші қатарға үшінші әріптен бастап, т.с.с. жазылады. Қатарда бос қалған ұяшықтарға әліпби әріптері басынан бастап толтырылады (кесте 8).

А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	…
Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	…
Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	
В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	…
Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	…
Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	
Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	…
Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	…
Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	…
Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	…
З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	…
И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	…
Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	…
К	Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	…
Қ	Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	…
Л	М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	…
М	Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	…
Н	Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	…
Ң	О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	…
О	Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ң	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	…
Ө	П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	…
П	Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	…
Р	С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	…
С	Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	…
Т	У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	…
У	Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	…
Ұ	Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	…
Ү	Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	…
Ф	Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	…
Х	Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	…
Һ	Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	…
Ц	Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	…
Ч	Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	…
Ш	Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	К	…
Щ	Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	…
Ы	І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	…
І	Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	…
Ь	Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	…
Ъ	Э	Ю	Я	А	Ә	Б	В	Г	Г	Д	Е	Ё	Ж	З	И	Й	К	К	Л	М	Н	Ң	О	…
Э	Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	…
Ю	Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	…
Я	А	Ә	Б	В	Г	Ғ	Д	Е	Ё	Ж	З	И	Й	К	Қ	Л	М	Н	Ң	О	Ө	П	Р	…

Тритемий шифрында бірінші қатар мәтіннің әріптер қатары болып саналады. Құпия мәтін әріптері осы қатардан таңдалады. Содан соң шифрлеуде мәтіннің бірінші әрпі бірінші қатардан, екінші әріп екінші қатардан, үшінші әріп үшінші қатардан т.с.с. алынады. Мысалы: «АЛМАТЫ» құпия сөзін шифрлейік. Бірінші қатардан А әрпінің өзін аламыз. Бірінші қатардағы Л әрпі екінші қатарда астындағы М әрпімен алмастырылады. Бірінші қатардағы М әрпінің орнына үшінші қатарда астындағы Н әрпін жазамыз. Бірінші қатардағы А әрпінің орнына төртінші қатарда астындағы В әрпін аламыз. Осылай жалғастыра берсек, төмендегі шифрмәтін пайда болады:

АМНВХЯ

Тритемий кестесінде шифр-әліпби қатар сайын солға бір әріпке жылжып отырады. Сонымен қатар шифрде кілт қолданылмайды. Бұл жерде құпия ретінде шифрлеу әдісі саналады.

Тритемий шифрының басқа вариантарында шифр келесі түрде қиындатылады:

1. бастапқы шифр-әліпби әріптерінің орналасу тәртібі өзгертіледі;
2. әріптерді шифрлеуде қатарды таңдау тәртібі қиындатылады.

Айта кетейік, қарапайым ауыстыру шифры (Юлий Цезарь шифры) Тритемий шифрының дербес жағдайы болып келеді, онда әріптер тек қана бір қатарда шифрленеді.

Белазо шифры. Тритемий шифрының дамыған түрі ретінде 1553 жылы ойлап табылған итальяндық Жован Белазо шифрын атауға болады. Белазо шифрында жеңіл жатталатын сөз – «пароль» енгізіледі. Пароль қайта-қайта шифрленетін мәтін үстіне жазылады. Парольдың әріптері мәтіндегі сәйкес әріптің Тритемий кестесіндегі шифрленетін қатарын көрсетеді. Мысалы, «ЕРТЕҢ КЕШКЕ КЕЛЕМІН» құпия сөйлемін «АЛМАТЫ» паролімен шифрлейік. Құпия сөйлемнің үстіне парольді қайта-қайта жазамыз.

А Л М А Т Ы А Л М А Т Ы А Л М А Т
Е Р Т Е Ң К Е Ш К Е К Е Л Е М І Н

Құпия сөйлемнің бірінші әрпін А әрпі басталатын қатардан, яғни бірінші қатардан іздейміз. Онда Е әрпінің өзі жазылады. Екінші Р әрпін Л әрпінен басталатын қатардан іздейміз. Тритемий кестесі бойынша Р әрпіне сейкес Ь әрпі жазылған. Үшінші Т әрпіне М әрпінен басталған қатардан Ю әрпі сәйкес келеді. Осылай жалғастыра берсек, төмендегі шифрмәтінді аламыз:

ЕЬЮЕА АЕДХЕ ЬАЛРЧІЯ

Егер Белазо шифрында әліпби әріптерінің орналасу тәртібі өзгертілсе, онда шифрдың тұрақтылығы артады.

Белазо шифрына италяндық Дж. Кардананың мынадай идеясы бар. Ол пароль ретінде құпия сөйлемнің бірінші сөзін алуды ұсынған. Мысалы, «ЕРТЕҢ КЕШКЕ КЕЛЕМІН» құпия сөйлемін шифрлейік. Онда пароль ретінде «ЕРТЕҢ» сөзі алынады. Бірақта бұл идея осы күйінде қолданылмаған. Оған негізгі себеп, шифрмәтінді ашып оқу үшін де пароль қажет болады. Бұл идея қазіргі таңда шифромәтіндік жүйелер құруда қолданыс табуда.

Порта шифры. Бұл шифр биграммалық (екі әріпті) ауыстыру шифры, яғни ашық мәтіндегі әрбір қосақталған екі әріп арнайы ойлап табылған белгімен алмастырылады. Арнайы белгілер шифрмәтінде символдық-геометриялық түрінде берілген. Мағынасы бойынша Порта шифрлары қарапайым ауыстыру шифрларының бір түрі, мұнда тек қосақталған әріптер қолданылған.

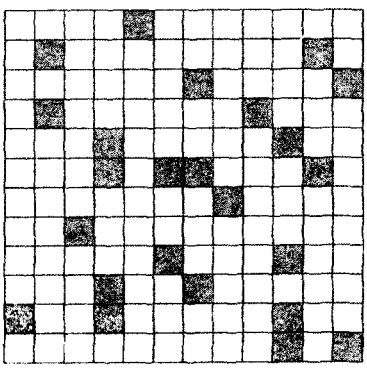
Сонымен қатар Порта шифрының Белазо шифрын дамытқан түрі де белгілі. Мұнда Тритемий кестесінің өзгертірілген варианты қолданылады. Осы кестенің қазақ әліпбиіндегі нұсқасын қарастырайық.

Кесте 9 – Тритемий кестесінің дамытылған варианты (қазақ әліпбиімен)

А	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Ә	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я
Б	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
В	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а
Г	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Ғ	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә
Д	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Е	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә	б
Ё	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	и	ң	о	о
Ж	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в
З	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
И	ұ	ү	ф	х	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г
Й	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
К	ү	ф	х	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ
Қ	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Л	ф	х	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д
М	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Н	х	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е
Ң	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
О	һ	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е	ё
Ө	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
П	ц	ч	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж
Р	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ц	о	ө
С	ч	щ	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж	з
Т	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
У	ш	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж	з	и
Ұ	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Ү	щ	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й
Ф	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Х	ы	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к
Һ	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Ц	і	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ
Ч	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Ш	ь	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л
Щ	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Ы	ъ	э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м
І	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н
Ъ	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Э	ю	я	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң
Ю	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о	ө
Я	я	а	ә	б	в	г	ғ	д	е	ё	ж	з	и	й	к	қ	л	м	н	ң	о

Кардано торлары. XVI ғасырда итальяндық Дж. Кардано криптографияда жаңа әдіс ұсынады, яғни мәтінді шифрлеуде кілт ретінде «Кардано торынан» пайдаланады. Тарихта «Кардано торы» деп аталатын шаблон былай әзірленеді: төрт бұрышты қатты материалға (картон, металл) тәртіпсіз квадрат ұяшықтар ойылады (кесте 10).

Кесте 10 – Кардано торы



Мәтінді шифрлеуде осы шаблон қағазға қойылып, ашық ұяшықтарға мәтін әріптері жазылады. Барлық ұяшықтар толғасын, шаблонды 90^0-қа бұрып, ашық ұяшықтарға қалған әріптер толтырылады. Осылай шаблон 4 рет бұрылады. Егер мәтін әріптері артып қалса, онда келесі қағазға осы әдіспен жалғасады.

Кардано торына қойылатын негізгі талап, бұрылғанда ұяшықтар бір-бірінің үстіне түспеуі қажет. Егер шаблонды алғасын қағазда бос орындар қалса, онда оларды кез келген әріптермен толтырады. Содан соң қағаздағы әріптер қатар бойынша жазылады. Осы қатар шифрмәтін болады.

Жалпы, Кардано әдісі орын ауыстыру шифры болып есептеледі. Кардано торларының саны шаблон өлшемі $n*n$ болғанда, 4^{n^2-1}-ге тең. Сонымен қатар бұл сан - Кардано шифрындағы кілттердің саны болып табылады.

Кардананың тағы бір идеясы, бір құпия мәтінді әр түрлі әдіспен кайта-кайта 3 рет шифрлеу болып табылады.

Виженер шифры. XVI ғасырда француз Блез де Виженер криптографиямен айналысады. Тарихта Виженер шифры деп аталатын шифрлеу әдісі Тритемий, Белазо әдістерінің жалғасы болып табылады. Виженер шифрында Тритемий кестесінен пайдаланады (кесте 8). Әліпбидің бір әрпі алдын-ала кілт ретінде алынады. Ашық мәтіннің бірінші әрпі осы әріп басталатын қатарда шифрленеді. Екінші әріп шифрленген бірінші әріптен басталатын қатарда шифрленеді. Әрі қарай осылай жалғаса береді. Сонымен ілгері Кардано енгізген идея амалға асырылады. Шифрдің кемшілігі ретінде оның әлсіз тұрақтылығын айтуға болады. Егер пайдаланылатын Тритемий кестесі белгілі болса, онда шифрден шығару үшін кілттік әріпті іздеп тапса, шифр ашылып етеді.

Виженер ұсынған екінші әдісте ұраннан пайдаланады. Жалпы алғанда, Виженер шифры Тритемий, Белазо және Портаның ашық мәтіндерді шифрлеу әдістерінің жалғасы болып есептеледі.

Жоғарыда классикалық криптографиялық әдістерге шолу жасадық. Олардың саны өте көп, бірақта негізгілерін атап өттік. Жалпы осы классикалық әдістер қазіргі таңда қолданылатын әдістердің бастауы болып есептеледі. Оларды тастап кетуге болмайды, себебі кейбіреулері әлі де колданыста.

3 СИММЕТРИЯЛЫҚ АЛГОРИТМДІ КРИПТОЖҮЙЕЛЕРДІҢ МАТЕМАТИКАЛЫҚ МОДЕЛЬДЕРІ

Шифрлердің математикалық модельдерін алғашқы болып К.Шеннон зерттеген. Ол «құпия жүйелерді» қарастырған болатын. Мұндай жүйелерде жіберілетін мәтін шифрленеді, бірақта шифрленген криптомәтіннің өзі құпия болмайды, яғни ашық таратылады. Ал мәтінді шифрлеу әдістері, сондай-ақ шифрден шығару әдістері құпия сақталады. Шифрленген криптомәтін сигналдар түрінде таратылады. Сонымен қатар қарсыластар осы сигналдарды ұстау және жазу үшін арнайы құралдармен жабдықталған деп пайымдалады. Таралатын сигналдар дискретті болып есептеледі, яғни криптомәтін дискретті символдар тізбегінен тұрады. Символдар қандай-да бір тілдің әріптері немесе сөздері, аудио немесе видеосигналдар, кванталған дыбыстың амплитудалық деңгейлері болуы мүмкін. «Құпия жүйелердің» ядросы болып нақты шифрлеу әдісі есептеледі. Жалпы, К.Шеннонның «құпия жүйелері» кәсіпқой таңдағы криптографиялық жүйелердің негізі болып келеді.

Шифрлердің математикалық модельдерін сипаттаудан алдын жиындар мен бейнелеу туралы түсініктерді қарастырайық.

Жиындар. Жиын және жиынның элементі ұғымдары негізгі ұғымдар болып есептеледі, яғни олар анықталмайды. Жиын элементтерден тұрады. Жиынды бас әріптермен X, Y, Z деп, ал элементтерді кіші әріптермен x, y, z деп белгілеу келісілген.

Жеткілікті дәрежеде үлкен әмбебап жиын бар деп есептейік және S деп белгілейік. Барлық қарастырып отырған жиындардың элементтері осы әмбебап жиында жатады деп пайымдайық.

Айталық, X жиын болсын. Егер кез келген $x \in S$ деп алсақ және бұл элементтің X жиынына тиісті немесе тиісті болмауы белгілі болса, онда X жиыны берілген дейміз. Егер x элементі - X жиынына тиісті болса, онда $x \in X$ деп, ал кері жағдайда $x \notin X$, яғни тиісті емес деп аламыз.

Анықтама 1. Егер X жиынының әр бір элементі Y жиынына да тиісті болса, онда X жиыны Y жиынының ішкі жиыны деп айтамыз және $X \subset Y$ деп белгілейміз.

Анықтама 2. Егер X және Y жиындарына бірдей элементтер тиісті болса, онда оларды тең деп атаймыз және $X=Y$ деп белгілейміз.

Бірінші анықтамадан $X\subset Y$ және $Y\subset X$ екені келіп шығады. Егер жиынның элементтері болмаса, онда оны бос жиын деп атаймыз және $\varnothing$ деп белгілейміз.

Анықтама 3. X және Y жиындарының бірігуін $X\cup Y$ деп белгілейміз және келесі түрде $X\cup Y=\{x\,|\,x\in X$ және $x\in Y\}$ анықтаймыз.

Анықтама 4. X және Y жиындарының қиылысуын $X\cap Y$ деп белгілейміз және келесі түрде $X\cap Y=\{x\,|\,x\in X$ немесе $x\in Y\}$ анықтаймыз.

Анықтама 5. X және Y жиындарының айырымын $X\setminus Y$ деп белгілейміз және келесі түрде $X\setminus Y=\{x\,|\,x\in X$ және $x\notin Y\}$ анықтаймыз.

Анықтама 6. X және Y жиындарының симметриялық айырымын $X\Delta Y$ деп белгілейміз және келесі түрде $X\Delta Y=(X\setminus Y)\cup(Y\setminus X)$ немесе $X\Delta Y=(X\cup Y)\setminus(X\cap Y)$ деп анықтаймыз.

Анықтама 7. X жиынының S жиынға қатысты толықтыруын $\overline{X}$ деп белгілейміз және келесі түрде $\overline{X}=S\setminus X$ анықтаймыз.

Бейнелеулер. Айталық, X және Y – жиындары, сонымен қатар $f-X$ жиынынан Y жиынына бейнелеу (функция), яғни әрбір $x\in X$ элементке $f(x)\in Y$ элементін сәйкес қоятын ереже берілген болсын. Егер X жиыннан Y жиынға бейнелеу берілсе, онда осы бейнелеуді $f:X\to Y$ деп белгілейміз.

Анықтама 8. $f_1:X_1\to Y_1$ және $f_2:X_2\to Y_2$ бейнелеулер *тең* деп есептеледі, егер $X_1=X_2$, $Y_1=Y_2$ және $f_1=f_2$ болса, яғни кез келген $x\in X_1(=X_2)$ үшін $f_1(x)=f_2(x)$ теңдігі орынды болса.

Анықтама 9. $f: X \to Y$, $B \subset Y$ берілген болсын. Онда f бейнелеудегі *B жиынының түпбейнесі деп келесі шарт бойынша анықталатын* $f^{-1}(B)(\subset X)$ жиынды айтады: $x \in f^{-1}(B) \Leftrightarrow f(x) \in B$.

Анықтама 10. $f: X \to Y$, $A \subset X$ берілген болсын. Онда f бейнелеудегі А жиынының бейнесі деп келесі шарт бойынша анықталатын $f(A)(\subset Y)$ жиынды айтады: $y \in f(A) \Leftrightarrow f^{-1}(\{y\}) \cap A \neq \varnothing$.

Теорема 1 (бейнелер мен түпбейнелердің қасиеттері) $f: X \to Y$; $A_1, A_2 \subset X$; $B_1, B_2 \subset Y$ болсын. Онда келесі өрнектер орынды:

- $f(A_1 \cup A_2) = f(A_1) \cup f(A_2)$;
- $f^{-1}(B_1 \cup B_2) = f^{-1}(B_1) \cup f^{-1}(B_2)$;
- $f(A_1 \cap A_2) \subset f(A_1) \cap f(A_2)$;
- $f^{-1}(B_1 \cap B_2) = f^{-1}(B_1) \cap f^{-1}(B_2)$.

Анықтама 11. $f: X \to Y$, $g: Y \to Z$ бейнелеулер берілген болсын. Онда f және g бейнелулердің композициясы (күрделі бейнелеу) деп келесі шарт бойынша анықталатын $g \circ f: X \to Z$ бейнелеуді айтады: $(g \circ f)(x) \overset{def}{=} g(f(x))$.

Теорема 2 (композициялардың ассоциативтігі). Егер $f: X \to Y$, $g: Y \to Z$, $h: Z \to W$ болса, онда $\forall x(\in X)$ үшін

$$\begin{array}{l} (h \circ (g \circ f))(x) = ((h \circ g) \circ f)(x) \\ (h \circ (g \circ f))(x) = h((g \circ f)(x)) = h(g(f(x))); \\ ((h \circ g) \circ f)(x) = (h \circ g)(f(x)) = h(g(f(x))), \end{array} \qquad (1)$$

яғни (1) формуланың сол жағы мен оң жағы тең.

Анықтама 12. Егер $\forall y(\in Y)$ үшін $f^{-1}(\{y\}) \neq \varnothing$ болса, онда $f: X \to Y$ бейнелеуі сюръективті деп аталады.

Анықтама 13. Егер $\forall x_1(\in X) \forall x_2(\in X)$ үшін $(\tilde{o}_1 \neq \tilde{o}_2) \Rightarrow f(x_1) \neq f(x_2)$ болса, онда $f: X \to Y$ бейнелеуі инъективті деп аталады.

Анықтама 14. Егер $f: X \to Y$ бейнелеуі сюръективті және инъективті болса, онда ол биективті деп аталады.

Теорема 3. (инъективті бейнелеулердің композициясы туралы). Егер $f: X \to Y$ және $g: Y \to Z$ инъективті бейнелеулер болса, онда $g \circ f: X \to Z$ инъективті бейнелеу болады.

Теорема 4. (сюръективті бейнелеулердің композициясы туралы). Егер $f: X \to Y$ және $g: Y \to Z$ сюръективті бейнелеулер болса, онда $g \circ f: X \to Z$ сюръективті бейнелеу болады.

Теорема 5. (биективті бейнелеулердің композициясы туралы). Егер $f: X \to Y$ және $g: Y \to Z$ биективті бейнелеулер болса, онда $g \circ f: X \to Z$ биективті бейнелеу болады.

Бұл теорема алдынғы теоремалардың салдары екендігі айқын. Әрі қарай, сюръективтікке келесі, жоғарыда енгізілген анықтамаға пара-пар анықтама береміз.

Анықтама 15. Егер $f(x) = y$ теңдеуі, мұндағы x – айнымалы, y – параметр, $y \in Y$ параметрдің кез келген мәнінде кемінде бір шешімге ие болса, онда $f: X \to Y$ бейнелеу сюръективті деп аталады.

Анықтама 16. X жиыны берілген болсын. X жиынында тепе-тең бейнелеу деп келесі шарт бойынша анықталатын $e_X: X \to X$ бейнелеуін айтады: $e_X(x) = x$, $\forall x \in X$.

Шифрдың математикалық моделі. Айталық, X, K, Y - ақырлы жиындары берілген болсын. Мұндағы, X - ашық мәтіндер жиыны, K - кілттер жиыны, Y - шифрленген мәтіндер жиыны (криптограмма). f функциясы (бейнелеу) X және K жиындарының тікелей көбейтіндісі ретінде берілсін, яғни

$$f: X \times K \to Y \ (f(x, \chi) = y, \quad x \in X, \quad \chi \in K, \quad y \in Y) \qquad (2)$$

f функциясы $f_\chi : X \to Y$, $\chi \in K$ бейнелеулерінің тобына сәйкес келеді, әрбір бейнелеу былайша беріледі: барлық $x \in X$ үшін $f_\chi(x) = f(x, \chi)$.

Сонымен, f_χ - $X \times \{\chi\}$ жиындағы f шектеу. Мұндағы $\{\chi\}$ -бір элементтен тұратын жиын. Осылардан

$$(f_\chi)_{\chi \in K},\ f_\chi : X \to Y \tag{3}$$

бейнелеулер тобы сәйкес

$$f : X \times K \to Y,\ f(x, \chi) = f_\chi(x) \tag{4}$$

бейнелеуін анықтайды.

Анықтама 17. Егер f функциясы сюръективті және кез келген $\chi \in K$ үшін f_χ функциясы инъективті болса, онда енгізілген X, K, Y жиындары $f : X \times K \to Y$ функциясымен

$$A = (X, K, Y, f) \tag{5}$$

шифрдың математикалық моделін құрайды.

Енгізілген (5) әрігі жалпы f функционалдық амалы бар үш негізді алгебраны анықтайды.

Әрі қарай нақты шифрлеу модельдерін қарастырайық.

Белгілеулер енгіземіз. I деп қандай да бір әліпбиді, ал I^* арқылы осы әліпбидегі барлық сөздер жиынын белгілейік, яғни $(i_1, i_2, \ldots, i_L)$, $i_j \in I$, $j \in \{1, \ldots, L\}$, $L \in \{1, 2, \ldots\}$.

Қарапайым ауыстыру шифры. Айталық, $X = M$ жиын, I^*-нің кандай да бір ішкі жиыны, ал $K = S(I)$ жиын, I әліпбидегі барлық ауыстырулар жиыны болсын. Әрбір $g \in K$ үшін, f_g-ді $f_g(i_1, i_2, \ldots, i_L) = g(i_1), g(i_2), \ldots, g(i_L)$ түрінде анықтаймыз. Қосымша $f(i_1, i_2, \ldots, i_L, g) = f_g(i_1, i_2, \ldots, i_L)$ және $Y = f(M) = \{f(i_1, i_2, \ldots, i_L, g)\}$:

$g \in S(I)$, $(i_1, i_2, \ldots, i_L) \in M$ енгіземіз. Онда, қарапайым ауыстыру шифрының моделі

$$A = (M, S(I), Y, f) \tag{5}$$

деп жазылады.

Орын алмастыру шифры. Айталық, X жиыны I әліпбидегі, ұзындығы T-ға еселік, ашық мәтіндердің жиыны болсын. Сонымен қатар, $K = S_T$ жиыны T дәрежелі симметриялық орын алмастырулар жиыны болсын. Онда $g \in S_T$ үшін f_g-ді $f_g(i_1, i_2, \ldots, i_L) = (i_{g(1)}, i_{g(2)}, \ldots, i_{g(T)})$ түрінде анықтаймыз, мұндағы $(i_1, i_2, \ldots, i_T) \in X$.

Әрі қарай, X жиынның қалған элементтерінен төмендегі ереже бойынша f_g-ді толық анықтаймыз: $x \in X$ мәтінді ұзындығы T кесінділерге бөлеміз. Әрбір кесінді g кілтімен жоғарыдағы шифрлеу ережесі бойынша шифрленеді. Шифрленген кесінділердегі әріптердің тізбегі x ашық мәтінін g кілтімен *шифрлеген* криптомәтінді құрайды.

Сонымен, орын алмастыру шифрының моделі

$$A = (X, S_T, Y, f) \tag{6}$$

деп жазылады. Мұндағы $f : X \times K \to Y$.

Ұзындығы T-ға еселік емес мәтінді шифрлеу үшін, мәтінге әріптер ұзындығы T-ға еселік болғанша қосылады.

Гаммалау шифры. Айталық, I әліпбиінің әріптері қандай-да бір ережемен тәртіптелген болсын. Әріптердің тәртіп номеріне әріптердің өзін сәйкес қоямыз, яғни $I = \{1, 2, \ldots, n\}$, $|I| = n$. Ал, X жиынын қандайда бір I^L, $K \subseteq I^L$ жиынының ішкі жиыны деп аламыз. Кілт $\gamma = \gamma_1, \gamma_1, \ldots, \gamma_1$ және $x = i_1, i_2, \ldots, i_L$ үшін $f_\gamma(i_1, i_2, \ldots, i_L) = y_1, y_2, \ldots, y_L$ қоямыз, мұндағы $\gamma \in K$, $x \in X$ және $y_j = i_j + \gamma_j \bmod(n)$, $j \in \{1, \ldots, L\}$.

Кейде гаммалау шифры деп $y_j = i_j - \gamma_j$ немесе $y_j = \gamma_j - i_j \bmod(n)$ шифрлеу әдістерін де айтады.

Ағындық шифры. Алдын ала I әліпби әріптерін шифрлеу үшін (I, Γ, Y, f) қосымша шифрын енгіземіз. Кілт $\gamma_1 \in \Gamma$ және ашық мәтін әріптері $i \in I$ үшін шифрленген мәтінді $f_{\gamma 1}(i) = y$ деп жазамыз. K арқылы ағындық шифр кілттерінің жиынын белгілейміз. Әрі қарай, l натурал саны үшін $\Phi : K \to \Gamma^L$ бейнелеуін енгіземіз. Фиксирленген $\chi \in K$ кілтіне $\Phi(\chi) = \gamma_1, \gamma_1, \ldots, \gamma_L$ қоямыз. Онда, ағындық шифрының моделі

$$A = (I^L, K, F, \Phi) \tag{7}$$

деп жазылады. Сонымен, ағындық шифры (7) қосымша шифр (I, Γ, Y, f) үшін $x = i_1, i_2, \ldots, i_L$ ашық мәтінін $\chi \in K$ кілтімен төмендегі ереже бойынша шифрлайды.

$$F_\chi(i_1, i_2, \ldots, i_L) = f_{\gamma 1}(i_1), f_{\gamma 2}(i_2), \ldots, f_{\gamma L}(i_L)$$

мұндағы, $f_\gamma(i) = f(i, \gamma)$.

Ағындық шифрларға мысал ретінде төмендегі ағындық ауыстыру және гаммалау шифрлерін айтуға болады. Ағындық ауыстыру шифры деп, тірек шифры $(X = I, K = \Gamma, Y = I, f)$ түрінде, ал I жиынындағы ауыстырулар $(i_\gamma)_{\gamma \in \Gamma}$ түрінде берілген ағындық шифрын айтамыз. Сол сияқты, гаммалау шифры деп, тірек шифры $I = A = \{0, 1, \ldots, n\}$, $f(i, \gamma) = i + \gamma \bmod |I|$ түрінде берілген ағындық шифрын айтамыз.

Бөлім соңында шифрлердің көбейтіндісі мен транзитивтігі ұғымдарын қарастырайық. Бұл екі анықтама шифрлердің параметрлері мен қасиеттерін сипаттауда қажет.

Анықтама 18. Берілген $A_1 = (X_1, K_1, Y_1, f_1)$ және $A_2 = (X_2, K_2, Y_2, f_2)$ шифрлерінің көбейтіндісі деп, $A = (X_1, K_1 \times K_2, Y_2, f)$ шифрын айтады, мұндағы $f(x, (\chi_1, \chi_2)) = f_2(f_1(x, \chi_1), \chi_2)$, $(\chi_1, \chi_2) \in K_1 \times K_2$.

Анықтама 19. Егер $f(x,\chi)=y$ өрнегіне $x\in X$ және $y\in Y$ үшін $\chi\in K$ табылса, онда $A=(X,K,Y,f)$ шифры транзитивті деп айтады.

Сонымен, бұл бөлімде симметриялық алгоритмді криптожүйелердің математикалық модельдерін сипаттадық.

Жоғарыдағы $f(x,\chi)=y$ өрнегі шифрлеу теңдеуі деп айтады, яғни x ашық мәтіні χ кілт сөзімен шифрленгенде y криптомәтіні шығады. Ал, $f_\chi^{-1}(y)=x$ немесе $f^{-1}(y,\chi)=x$ өрнегін шифрден шығару теңдеуі деп айтады, яғни $y=f(x,\chi)$ шифрленген криптомәтін χ кілт сөзімен шифрден шығарылғанда бастапқы x ашық мәтінін аламыз.

Жоғарыдағы (5) өрнегінде f_χ бейнелеуінің инъективтілігі криптомәтінді шифрден шығарудың бірмәнді екендігін анықтайды, яғни шифрленген криптомәтін мен кілт сөзден жалғыз ашық мәтін шифрден шығады. Ал f_χ бейнелеуінің сюръективтілігі қатаң талап етпесе де болатын шарт, тек қана математикалық сипаттауда нақтылық үшін қажет. Сондықтан кейбір шифрлерді талдауда, өрнектер қарапайым болуы үшін бұл талапты тастап кетеміз.

Жалпы, енгізілген (5) шифрдің математикалық моделі симметриялық алгоритмді криптожүйелердің шифрлеу және шифрден шығарудың тек қана функционалдық қасиеттерін сипаттайды. Бұл модельде ашық мәтін және криптомәтін тек қана X және Y жиындарының абстракт элементі ретінде анықталып, олардың әліпбиі, криптографиялық қасиеттері қаралмайды.

4 СИММЕТРИЯЛЫҚ ШИФРЛЕРДІҢ КРИПТОГРАФИЯЛЫҚ ҚАСИЕТТЕРІ

Алдыңғы бөлімде симметриялық алгоритмді криптожүйелердің математикалық модельдерін сипаттадық. Бөлімде қаралған шифрдің математикалық моделі (5) симметриялық криптожүйенің шифрлеу және шифрден шығару алгоритмдерінің тек қана функционалдық қасиеттерін сипаттайды. Ал оның криптографиялық қасиеттері қаралмайды. Сондықтан бұл бөлімде осы аталған қасиеттерді сипаттайық.

Әліпби. Криптографияда ашық мәтінді шифрлеу және шифрден шығаруда қандай да бір әліпбиден пайдаланады. Әліпби (alphabet) деп ашық және құпия мәтінде пайдаланылатын символдардың шектеулі жиынын айтады. Әліпбиді жалпы түрде былай жазамыз:

$$I = \{a_0, a_1, \ldots, a_{n-1}\}$$

Әліпбидегі әріптерді белгілі бір ереже бойынша түрлендіру арқылы жаңа әліпби құруға болады. Криптография әдістерінде әліпби әріптерін топтап біріктіру жиі кездеседі.

$I^2 = \{a_0a_0, a_0a_1, \ldots, a_{n-1}a_{n-1}\}$ - биграммалы әліпби.

$I^3 = \{a_0a_0a_0, a_0a_0a_1, \ldots, a_{n-1}a_{n-1}a_{n-1}\}$ - үшграммалы әліпби.

Жалпы жағдайда, m әріптері бойынша біріктірсек, онда m-граммалы I^m әліпбиін аламыз.

Мысалы: Қазақ әліпбиіндегі $n = 42$ әріптерді біріктіру арқылы төмендегі әліпбилерді құруға болады:

$I^2 = \{АА, \ldots, ЯЯ\}$ -биграммалы әліпби.

$I^3 = \{ААА, \ldots, ЯЯЯ\}$ -үшграммалы әліпби.

Криптографиялық түрлендіруді орындау кезінде әліпби әріптері бүтін сандарға 0, 1, 2, 3, ... ауыстыру да жиі кездеседі.

Мысалы:

I={АӘБВГҒДЕ ... ЮЯ}, I_{42}={0, 1, 2, ..., 41}; қазақ әліпбиі

I={ABCDEF ... YZ}, I_{26}={0, 1, ..., 25} ағылшын әліпбиі

Криптоберіктілік. Шифрдың негізгі қасиеттерінің бірі – шифрлеу әдісінің криптоберіктілігі болып есептеледі. Бұл қасиет криптоталдауда, яғни кілт белгісіз жағдайда құпия мәтінді шифрден шығару қиындығын анықтайды. Криптоберіктілікті көрсететін бірнеше көрсеткіштер кездеседі:

- барлық мүмкін кілттердің саны;
- шифрды ашу үшін қажетті уақыт мөлшері.

Классикалық криптография теориясының негізін қалаушы К.Шеннон криптоберіктіліктің екі түрін көрсетеді:

- теориялық;
- практикалық.

Теориялық криптоберіктілік құпия мәтінді ашуда кездесетін анықтықтардың жоқтығымен сипатталады. Қарсылас құпия мәтінге қол жеткізгенімен, оның жалғыз ашық мәні бар екенін анықтай алмайды. Көптеген практикалық шифрлерде мұндай белгісіздік, екіұштылық болмайды, яғни құпия мәтіннің жалғыз ашық мәтіні бар екендігі анық, тек қана құпия мәтінді дешифрлеу қажет. Егер дешифрлеу күрделі, көп уақытты талап ететін болса, онда шифрді практикалық криптоберікті деп айтады.

Кілт ұзындығы. Құпия кілт ұзындығы - шифрдың негізгі қасиеттерінің бірі, сенімділігін бағалау критерийлерінің ішінде ең маңыздысы.

Шифрды ашудың ең қарапайым әдісі, кілттің мүмкін ықтимал барлық варианттарын бірінен соң бірін таңдап алып, солардың әрқайсысымен криптомәтінді дешифрлеу және алынған нәтижені оқып көру, талдау. Бұл өте күрделі, сонымен қатар сенімді әдіс және оны барлық шифрлеу әдістеріне қолдануға болады. Сондықтан қолданылатын шифрды осы әдіспен дешифрлеуден қорғанудың бір тәсілі ретінде кілт ұзындығын ұзарту болып есептеледі. Сонда мүмкін ықтимал кілтті таңдау саны күрт өседі. Мұндайда

дешифрлеуде кілттердің барлығын тексеріп шығу үшін қажетті уақытты мүмкіндігі барынша шексіз ету керек.

Жылдамдық. Шифрдің негізгі қасиетінің бірі – оның жұмыс істеу жылдамдығы. Жоғарыда шифрдың сенімділігі үшін құпия кілттің ұзын болғаны жақсы деп айттық. Бірақта қолданылатын кілт ұзындығы артқан сайын, криптографиялық әдісті қолдану қиындығы да өседі. Сонымен қатар криптожүйеге қойылатын талаптар да жоғары болады.

Қарапайымдылық. Қазіргі таңда шифрлерге қойылатын талаптың бірі – шифрдің қарапайымдылық қасиеті. Шифрлеуде есептеу техникасын қолдану, оны техникалық және бағдарламалық түрде амалға асыру - шифрлеу әдісінің қарапайым, бағдарламалауға жеңіл болуын талап етеді. Мұның барлығын есептеу техникасының жылдамдығы, бағдарламашы біліктілігі сияқты факторлармен түсіндіруге болады.

Есептеу қателігі. Шифрлеуде есептеу техникасын қолдану барысында амалдардың орындалу жылдамдығы, бағдарламалау жүйесін таңдау сияқты мәселелер туындайды. Сонымен қатар, есептеу қателіктеріне де көңіл беру керек. Әсіресе сандар үстінде арифметикалық амалдар күрделі болса, дөңгелектеу қателігі барлық жұмысты жоққа шығаруы мүмкін, онда бағдарламалау үрдісін қайта қарап шығу керек болады. Орнне мұндайда қателіктер санын азайтуға әрекет жасалады.

Симметриялық шифрлеу әдістерін қолданудан алдын, олардың арасындағы ерекшеліктерді ажыратуға мүмкіндік беретін қасиеттерін анықтап алу қажет. Барлық қолданылып жүрген симметриялық әдістер бастапқы ашық мәтінді әркайсысы жеке шифрленетін бірнеше бөліктерге бөлуге байланысты:

1. Бастапқы ашық мәтіннің қандай өлшеммен бөлектеп (бит, байт немесе блок) шифрлеу қажеттігін анықтау керек. Мысалы, кейбір әдістерде байттармен, ал басқаларында блок деп аталатын биттер тізбегімен шифрлеу жүргізіледі. Шифрлеуде биттер тізбегін қолдану қасиеті блоктық шифрлеу деп аталады.

2. Шифрлеудің кейбір әдістері немесе функциялары ашық мәтіннің әліпби белгілеріне тәуелді болуы мүмкін. Мұндай қасиеті

бар шифрлеу әдістерінде қателер саны көбейеді. Егер шифрлеу кезінде ең болмаса бір бит қате шифрленсе немесе тасымалданса, онда криптомәтінді шифрден шығаруда қателер пайда болады. Шифрлеу әдісінің ашық мәтін әліпби белгілеріне тәуелділігі түйіншектілік қасиеті деп аталады.

3. Ашық мәтіннің жеке әліпби белгілерін шифрлеу олардың мәтіндегі орнына тәуелді болуы мүмкін. Мұндайда ашық мәтіннің кез келген бір белгісін қате шифрлеу, барлық мәтіннің келесі бөлімдерінің де қате шифрленуіне әкеледі. Шифр белгілерінің ашық мәтіндегі орнына тәуелсіздігі транзитивтілік қасиеті деп аталады.

Симметриялық шифрлеу әдістері шифрленетін бөліктерге байланысты екі түрлі болады: ағындық және блоктық.

Ағындық шифрлер. Ағындық шифрлер бастапқы ашық мәтіннің әр бір әліпби белгісін байт бойынша өңдейді, яғни ашық мәтіннің әрбір символы басқаларынан тәуелсіз түрде шифрленеді. Криптомәтіннің әрбір символы ашық мәтіннің сәйкес символын түрлендіру функциясының мәні болып келеді. Функция аргументі ретінде символ немесе байт, кейде мәтіннің бір бөлігі (бірнеше байт) алынуы мүмкін. Мұндай шифрлеу әдістерін байланыс арнасы арқылы ақпарат тасымалдаумен қатар жүргізуге болады. Ағындық шифрлеу әдісін амалда қолдану үшін ашық мәтіннің әрбір символын баскаларынан тәуелсіз түрде шифрлеуге арналған кілттік тізбектер генераторын құрастыру қажет.

Ағындық шифрлердің артықшылығы ретінде есептеу қателігінің аздығы, шифрдың қарапайымдылығы және шифрлеудің жоғары жылдамдығы сияқты қасиеттерін айтуға болады. Ал оның негізгі кемшілігі негізгі ақпаратты жіберуден алдын қабылдаушымен уақытты үйлестіру қажеттілігі. Ол үшін қабылдаушыға үйлестіру туралы дерек алдын-ала жіберіледі және мұндай дерек ақпаратты шифрден шығару басталғанға дейін жіберілуі тиіс. Ағындық шифрлер мемлекеттік байланыс, қауіпсіздік және әскери қызметтерінде, яғни ақпаратты цифрлік түрге және кері түрлендірілген дауыс сигналдарын шифрлеуге арналған жүйелерде кеңінен қолданылады.

Блоктық шифрлер. Мұндай шифрлеу әдісінде ашық мәтін бірнеше биттерден тұратын, ұзындығы бірдей блоктарға бөлініп

шифрленеді. Блоктар осындай ұзындықты криптомәтін блоктарына шифрлеуге арналған шифрлеу функциясымен өңделеді. Шифрлеу функциясы кілтке тәуелді болады.

Блоктық шифрлердің артықшылығы ретінде ашық мәтін блогын криптомәтін блоктарына шифрлеуге арналған функциясын айтуға болады. Осы функцияға немесе кілтке кішігірім өзгерістер енгізу, криптомәтінде үлкен және алдын ала болжаланбаған өзгерістер әкеледі. Бірақта блоктық шифрлердің бірнеше кемшіліктері бар.

Біріншіден, егер блоктардың барлығына бірдей кілт қолданылса, онда криптомәтінде «ұқсас» блоктар пайда болуы мүмкін. Бұл криптоаналитик үшін таптырмас «олжа» болып есептеледі, яғни дешифрлеуге «сөздікті криптоталдау» әдісін қолдануға болады. Ашық мәтін үлкен болған сайын, криптомәтінде «ұқсас» блоктар қайталана береді. Олар криптоаналитикке криптомәтінді ашудың және кілтке жасалған шабуылдың жемісті аяқталуына үміт береді.

Екіншіден, блок ішіндегі қателіктер саны жылдам көбейеді. Шифрмәтіннің бір блогында жіберілген қате, оның барлық блогына әсер етеді. Мұндайда криптомәтінді кері шифрлеу қате нәтижеге алып келуі мүмкін. Сондықтан қарапайым блоктық шифрлеу әдісі үшін ашық мәтіндерде шифрлеу үшін қолданылмайды. Оның орнына әрбір блок үшін әртүрлі кілт қолданылатын немесе блок ұзындығын қайта-қайта өзгертетін модификациялары қолданылады. Осындай блоктық шифрлардың криптобериктілігі де жоғары болады.

Ағындық және блоктық шифрлеудің әркайсысының қасиеттерін қолдана отырып, аралас шифрлеу жүйелерін құруға болады. Мұндай жүйелерде ағындық шифрлеу блоктардағы биттерді орын ауыстыру әдісімен үйлестіреді. Алдымен ашық мәтін ағындық шифрлеу әдісімен шифрленеді, содан соң алынған криптомәтін блоктарға бөлінеді де, әрбір блокта әртүрлі кілтке сәйкес қосымша орын ауыстырулар орындалады. Нәтижеде шифрлеу әдісі анағұрлым күрделі болып шығады және оның криптобериктілігі арта түседі.

Симметриялық шифрлеу әдістерін мынадай екі жағдайда қолдануға болады: ақпаратты тасымалдау кезінде қорғау және ақпаратты сақтау кезінде қорғау.

Ақпаратты тасымалдау кезінде қорғау. Бұл жағдайда ақпарат байланыс арнасы арқылы тасымалданады. Құпия мәтін ақпаратты жіберуші жақта шифрленеді, ал қабылдаушы жақта – шифрден шығарылады. Қарсыластар байланыс арнасынан ақпаратты ұстап алса да, кілт сөзі болмағандықтан, шифрден шығаруға мүмкіндіктері болмайды. Симметриялық шифрларда ақпарат бірдей кілттермен шифрленеді және шифрден шығарылады деп айтып өттік. Тек қана құпия кілтті ақпаратты қабылдаушы жаққа жеткізуде қиындықтар туындауы мүмкін. Сондықтан кілтті құпия түрде жеткізу үшін мынадай екі тәсіл қолданылады:

- кілттер физикалық түрде (электрондық кілттер, пластикалық карталар, жеке құпиясөздер түрінде, т. б.) алдын-ала жеткізіледі;
- кілттер шифрленген түрде байланыс арнасымен жіберіледі.

Кілттерді жеткізу мәселесінің күрделі болу себебінің бірі, шифрдың криптоберіктілігін жоғарылату үшін кілттерді жиірек ауыстырып отыру қажет. Ал кілттерді қабылдаушы жаққа физикалық түрде жеткізу, шығындардың өсуіне әкеледі және криптожүйені «баға-сапа» тұрғысынан қарағанда тиімсіз етеді.

Сондықтан іс жүзінде екі тәсіл қоса қолданылады, яғни қабылдаушы жаққа бастапқыда ұзақ уақыттық кілт физикалық түрде жеткізіледі, осы кілт көмегімен сеанстық деп аталатын кілттер қажетті кезде шифрленеді. Ақпаратты тасымалдау кезінде осы сеанстық кілттер қолданылып, құпия мәтін шифрленеді.

Ақпаратты сақтау кезінде қорғау. Әдетте ақпарат криптожүйені тұтынушы (жіберуші және қабылдаушы) компьютерінде немесе ақпарат сақталатын сыртқы құрылғыларда сақталады. Бірақта қазіргі ақпараттық қоғамда оларды сақтау да қауіпсіз емес. Қарсылас Интернет желісі арқылы немесе басқа жолмен заңсыз іс-әрекет жасауы мүмкін. Сондықтан архивте сақталынған ақпаратты қорғау да маңызды мәселенің бірі болып есептеледі.

Сақталатын ақпаратты тасымалданатын ақпаратты қорғаудан айырмашылығы, кілттерді таратудың қажеттігі болмайды. Ақпаратты шифрлеу мен шифрден шығаруды бір адам орындайды. Сондықтан шифрлеу әдісі мен кілт сөзді еркін таңдауға болады. Тек қана ескеретін жағдай, кілт сөз естен шықпауы қажет.

Сақталатын ақпаратты криптографиялық қорғау мәселесіне екі әр түрлі тұрғыдан қарау керек. Біріншісі, компьютердегі ақпараттың барлығын толық шифрлеу. Бұл мәселе криптожүйелерге жұмыс істеу жылдамдығы тұрғысынан ерекше талаптар қояды, яғни шифрлеу және шифрден шығару үрдісі компьютер тұтынушысы сезбейтіндей жылдамдықпен жүргізілуі тиіс. Екіншісі, өте бағалы ақпаратты ғана шифрлеу және ақпарат сақталатын сыртқы құрылғыларда сақтау. Бұл жағдайда ақпаратты қажеттілік туындаған уақытта ғана жаңалауға болады.

5 СИММЕТРИЯЛЫҚ ШИФРЛЕУ СТАНДАРТТАРЫ

Симметриялық криптожүйелер деп мәтінді шифрлеуде және шифрден шығаруда тек бір криптографиялық кілт қолданылатын жүйелерді айтады. Кілт алгоритмі мәтінді жіберуші мен тұтынушы екеуінде де бірдей болуы және құпия түрде сақталуы тиіс. Егер кілт болмаса, онда мәтін бұл тек қана белгілердің мағынасыз тізбегі болып қалады.

Симметриялық әдістерді жіктеу. Барлық белгілі симметриялық шифрлеу әдістерін мынадай топтарға бөлуге болады:

- ауыстыру;
- орын ауыстыру;
- гаммалау;
- аналитикалық түрлендіру.

Ауыстыру шифрында ашық мәтін әліпбиінің әрбір әрпі белгілі бір әріп, цифр немесе арнайы символмен ауыстырылады. Ауыстыру шифрлерінің қарапайым және күрделі түрлері бар. Қарапайым ауыстыру шифрлеріне Юлий Цезарь шифры, Полибий квадраты, Тритемий шифры, Белазо шифры, Кардано торлары т.б. жатады. Күрделі ауыстыру шифрлеріне Виженер шифры, Порта шифры, Гронсфельд шифры т.б. әдістері жатады.

Орын ауыстыру шифрында ашық мәтіннің әріптері қандай да бір әдіспен өзара орын ауыстырылады. Мысал ретінде қарапайым орын ауыстыру шифры, Сцитала шифры, сиқырлы квадраттар және т.б. келтіруге болады.

Гаммалау шифрында ашық мәтіннің символдары шифр гаммасы деп аталатын жалғанкездейсоқ (pseudo-random) сандар қатарымен қосылады. Шифр гаммасын (γ - грек әліпбиінің әрпі) құпия кілт негізінде компьютерде жалғанкездейсоқ сандардың генераторлары түрлендіріп береді. Бұл әдіс есептеу техникасымен амалға асырылады, сондықтан кең қолданылып жүр.

Аналитикалық түрлендіруде ашық мәтін қандайда бір аналитикалық ереже (формула) бойынша шифрленеді. Мысалы, векторды матрицаға көбейту ережесін қолдануға болады. Көбейтілетін матрица шифрлеу кілті болып есептеледі. Сондықтан оның көлемі мен мазмұны құпия сақталады.

Симметриялық шифрлеу стандарттары. Қазіргі таңда ақпараттық есептеу жүйелерінде қолданылатын барлық технологиялар стандарттар мен келісімдердің жиынымен реттеледі. Сол сияқты криптография саласы брйынша көптеген мемлекеттерде шифрлеудің ұлттық стандарттары енгізілген. АҚШ-та 2001 жылы AES симметриялық шифрлеу стандарты қабылданған. Бұл стандарт блок ұзындығы 128, 192 және 256 биттік Rijndael алгоритміне негізделген. AES алгоритмі одан алдын пайдаланылған DES алгоритмінің орнына қолданылады және тек қана Triple DES режимінде ғана пайдалану ұсынылады. Ал Ресейде кілт ұзындығы 256 биттік блокты шифрлеу алгоритмімен сипатталатын ГОСТ 28147-89 стандарты қолданыста.

Сонымен, қазіргі таңда көп таралған алгоритмдерге төмендегілер кіреді:

- AES (Advanced Encryption Standard)
- ГОСТ 28147-89
- DES (Data Encryption Standard)
- 3DES (Triple-DES)
- RC6
- Twofish
- IDEA (International Data Encryption Algorithm)
- SEED
- Camellia

Төменде осы алгоритмдерге қысқаша шолу жасайық.

Advanced Encryption Standard (AES) блокты шифрлеудің симметриялық алгоритмі. Кейде оны Rijndael деп те атайды. AES алгоритмін АҚШ-тың Стандарт және технологиялардың ұлттық институты NIST (National Institute of Standards and Technology) ұсынған. Оның блок ұзындығы 128 бит, ал кілт ұзындығы 128/192/256 бит. 2002 жылдан бастап АҚШ-та DES алгоритмінің орнына шифрлеу стандарты болып есептеледі. Қазіргі кезде бұл алгоритм жетерлі дәрежеде зерттелген және көп қолданылады.

ГОСТ 28147-89 XX ғасырдың 70-ші жылдарында Кеңес Одағының ұлттық қауіпсіздік комитетінің 8-ші Бас басқармасында құрылған. 1989 жылы Кеңес Одағының стандарты ретінде қабылданған. Алгоритм мемлекеттік құпияны құрайтын ақпараттарды шифрлеуге арналған. Ресей Федерациясының 1990

жылдан бастап симметриялық шифрлеу стандарды болып есептеледі. Сонымен қатар ТМД елдерінде де осы стандарт қолданылып жүр. Шифр алгоритмінің негізін Фейстель желісі құрайды. Алгоритмде блок ұзындығы 64 бит, кілт ұзындығы 256 бит және 32 түрлендіру циклы бар.

DES (Data Encryption Standard) симметриялық шифрлеу алгоритмін IBM корпорациясы жасап шығарған. Алгоритм қазіргі AES алгоритмінен алдын, 1977 жылдан бастап АҚШ-тың шифрлеу стандарты болып келген. Алгоритм негізін блок ұзындығы 64 бит, кілт ұзындығы 56 бит және 16 түрлендіру циклы бар Фейстель желісі құрайды. Алгоритмде сызықты (E, IP, IP-1 орын ауыстырулар) және сызықты емес (S-блоктар) түрлендірулер қатар қолданылады. DES алгоритмінің бірнеше жұмыс істеу режімі бар:

- ECB (Electronic Code Book) - электрондық кодты кітап, мұнда екі түрлі алгоритм қолданылады;
- CBC (Cipher Block Chaining) - тізбекті режім, деректер блогін шифрлеу алдынғы блокты шифрлеу нәтижелеріне тәуелді;
- OFB (Output Feedback) – шығу бойынша кері байланыс, жалғанкездейсоқ сандардың генераторы ретінде қолданылады;
- CFB (Cipher Feedback) – шифрлеуші бойынша кері байланыс, ақпараттың аутентификациялық кодтарын алу үшін қолданылады.

DES алгоритмі көптеген бағдарламалық өнімдердің құрамына енгізілген, жер шарында кеңінен таралған. Бұл шифр АҚШ ресми стандарты ретінде 2000 жылға дейін қызмет етті. Осы стандартты қолдана бастаған 1977 жылдан бері, есептеу техникасы қарқынды дамыды. Соңғы кездері бұл шифрды ашу мүмкіндігі пайда болды.

Triple DES (3DES) шифрлеу алгоритмін 1978 жылы Уитфилд Диффи, Мартин Хеллман және Уолт Тачман ұсынған. Алгоритм негізін DES алгоритмі құрайды. Ол DES алгоритмінің негізгі кемшілігі кілт ұзындығының аздығын (56 бит) түзету мақсатында құрылған. Егер кілт ұзындығы аз болса, онда әрқашада толық таңдау әдісімен кілтті бұзу мүмкіндігі болады. 3DES алгоритмінің жылдамдығы DES алгоритмінен 3 есе аз, бірақта оның криптотұрақтылығы өте жоғары, яғни оны бұзуға миллиард есе көп уақыт қажет. Сондықтан 3DES алгоритмі DES алгоритміне қарағанда көп қолданылады.

RC6 алгоритмін 1998 жылы Рональд Райвест, Мэтт Робшоу және Рэй Сидни (RSA Laboratories) ұсынған. Ол RC5 алгоритмінің жалғасы болып табылады. Оның блок ұзындығы 128 бит, ал кілт ұзындығы 128/192/256 бит. Сонымен қатар RC5 сияқты, блок және кілт ұзындығын үлкен диапазонда (0-ден 2040 битке дейін) өзгерте алады. Алгоритмнің негізгі артықшылығы оның жылдамдығы. Ол Pentium II, Pentium Pro, Pentium III, PowerPC микропроцессорлы жүйелерде Rijndael алгоритмінен де жылдам істейді. Бірақта Intel IA-64 архитектуралы жүйелерде кейбір қарапайым амалдарды орындауда жылдамдығын жоғалтады. Сондықтан да ол АҚШ-тың шифрлеу стандарты болып алынбаған.

Twofish симметриялық шифрлеу алгоритмін Брюс Шнайер басшылығындағы мамандар тобы ұсынған. Алгоритмде блок ұзындығы 128 бит, ал кілт ұзындығы 256 битке дейін. Алгоритм негізінен ілгері қолданылған Blowfish, SAFER және Square алгоритмдері құрайды. Алгоритмнің ерекшілігі ретінде алдын ала есептелген, кілтке байланысты қолданылатын S-box және шифрлеу кілттерін алудың қиын схемасын айтуға болады. Ал кемшілігіне алгоритм құрылымының күрделілігі, оны талдау қиындығы мен жай жылдамдығы жатады.

IDEA (International Data Encryption Algorithm, халықаралық деректерді шифрлеу алгоритмі) – алгоритмін Швейцарияның Ascom Tech фирмасы патенттеген. Алгоритмнің алғашқы нұсқасын 1990 жылы Сюэция Лай (Xuejia Lai) мен Джеймс Мэсси (James Massey) құрап болатын. Бастапқыда алгоритм DES алгоритмінің орнына ұсынылған, PES (Proposed Encryption Standard, ұсынылған шифрлеу стандарты) деп аталған. Содан соң Бихам мен Шамирдің дифференциалды криптоталдау бойынша жұмыстарынан кейін қайта өңделіп IPES (Improved Proposed Encryption Standard, жақсартылған ұсынылған шифрлеу стандарты) деп аталған. Қазіргі кезде ол IDEA алгоритмі деп аталады. Алгоритмде блок ұзындығы 64 бит, ал кілт ұзындығы 128 битке тең. Алгоритмнің ерекшілігі, әрбір блокта әр түрлі шифрлеу режимдері қолданылады. Әр бір блок 16 биттен тортке бөлініп, осы 16 битке алгебралық амалдар қолданылады. Мәтінді шифрлеуде және шифрдан шығаруда бір түрлі алгоритм қолданылады.

SEED блокты симметриялық шифрлеу алгоритмін Кореяның ақпараттық қауіпсіздік агенттігі KISA (Korean Information Security

Agency) 1998 жылы жасап шығарған. Алгоритм Фейстель жүйесі негізінде жұмыс істейді. Алгоритмде блок ұзындығы да, кілт ұзындығы да 128 бит. Алгоритм Оңтүстік Кореяның финанстық және банк ұйымдарында қолданыста. Бірақта ол көптеген браузерлерге енгізілмеген, сондықтан Оңтүстік Кореядан басқа жерлерде қолданылмайды.

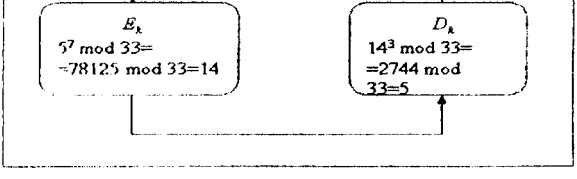
Camellia – блокты симметриялық шифрлеу алгоритмін 2000 жылы Nippon Telegraph and Telephone Corporation мен Mitsubishi Electric Corporation корпорациялары ұсынған. Camellia Жапонияның мемлекеттік және өндірістік мекемелерінде көп қолданылады. Ол E2 алгоритмінің дамытылған нұсқасы. Алгоритмде блок ұзындығы 128 бит, ал кілт ұзындығы да 128, 192, 256 бит. Алгоритм құрылымы Фейстель жүйесіне негізделген. Алгоритмде алдын ала және соңынан ақтау әдісі (whitening), ал цикл функциясында сызықты емес түрлендіру (S-блоктар), байттық орын ауыстыру және байттық XOR амалдары қолданылады. 2008 жылдан бастап Camellia алгоритмін Mozilla Firefox 3 браузері пайдалана бастады. Алгоритм патенттелген болсада, Жапониядан басқа мемлекеттерде де еркін лицензиямен қолданылып жүр.

6 RSA ШИФРЛАРЫ

Бірінші болып және ең көп тараған ашық кілтті криптографиялық жүйе 1978 жылы RSA деп аталатын жүйе ретінде ұсынылған. RSA жүйесінің атауы жүйені құрушылардың авторларының бас әріптерінен алынған, олар Р.Ривеста, А.Шамира, Л.Адлеман. Ол өте көп қиын бүтін сандарды қарапайым көбейткіштерге жіктеуге негізделген. Көбейту нәтижесінде шифрлеу ашық кілтті элементі қолданыла алады. Ағымдық қарапайым сандар шифрды қайта ашу үшін қолданылады, бірақ оны орнына қайта келтіру мәселесі криптоаналитиктердің шифрды бұзу жұмысына қарсы тұрып келеді. Осылайша біржақты құпия функцияны құруға болады. Мысал ретінде авторлардың шифрлеу принципін көрсетейік.

- Шифрлеу ашық кілті: n және e саны
- Шифрды ашу жабық кілті: p, q және d саны
- p ақпаратын шифрлеу алгоритмі: $E_k(\tilde{p}) = \tilde{p}^e (\mathrm{mod} n) = c$.
- c жабық ақпаратын қайта ашу алгоритмі: $D_k(c) = c^d (\mathrm{mod} n) = \tilde{p}$.

D_k қайта ашу алгоритмін табудың жалғыз әдісі, және е сандары белгілі болғанда, n санын қарапайым көбейткіштерге көбейтіп, p және q сандарының мәнін, сонымен бірге d санының мәнін табу керек. p және q сандарының разрядтығын дұрыс таңдау есептің n факторизациясын іс-жүзінде мүмкін емес етеді (RSA авторлары орнын 40-разрядтан төмен емес ондық сандарды қолдануды ұсынған). Келтірілген схема бойынша шифрлеу келесі суретте көрсетілген (сурет 1), мұндағы $e=7$, $d=3$ және $n=33$.



Сурет 1 - RSA схемасы бойынша шифрлеу

RSA авторлары өздерінің жүйелерінің жұмыс істеу принциптерін жазғанда төмендегі сөзді таңдап алды:

ITS ALL GREEC TO ME (Мен үшін мұның бәрі түсініксіз)

Бұл мәтінді үлкен санға айналдыру үшін сөздің арасындағы бос аралықты 0, А әрпін – 1, В әрпін – 2, С әрпін – 3, ... , Z әрпін – 26 деп кодтап алды. Әр символды көрсету үшін әрқайсысына 5 екілік разряд бөлді. Нәтижесінде келтірілген мәтінге келесі сан тең болды:

$$\tilde{p} = 09201900011212000718050511002015001305.$$

Шифрлеу үшін авторлар e=9007 және

n=1143816257578888676692357799761466420102182967212423625626518429357069352457338978305971235639587050589890751475992900268795435 41

сандарын таңдап алды. Шифрлағаннан соң

$$c = \tilde{p}^e \pmod{n} = 19993513146780510045231712274026064742320401705839146310370371740625971608948627504399209626725826750128935544613538237697480 26$$

санын алды. Кездейсоқ таңдап алынған n саны 64 және 65 дәрежелі p және q сандарының көбейтіндісі. Егер мәтін өте үлкен болған жағдайда онымен бір сан ретінде жұмыс жасау үшін оны блоктарға бөлу керек, әр блокты жеке сан ретінде қарастырып және шифрлеу барысында блоктар байланысын қолдану қажет.

RSA схемасы. RSA [Schn 96] шифрлеу үшін де және қол қою үшін де қолданылады.

P және q – үлкен қарапайым сандарын таңдап аламыз. $n = pq$ модуль болсын, келесі функция

$$\varphi(n) = (p-1)\cdot(q-1),$$

Эйлер функциясы.

Кез-келген $1 \le e < \varphi(n)$ аламыз, ол

$$\text{ҮОБ}(e, \varphi(n)) = 1.$$

Онда тек қана жалғыз $1 \le d < \varphi(n)$ болады, ол $ed(\mathrm{mod}\varphi(n)) = 1$-ге тең болады.

RSA шифрлеу жүйесі – ашық кілтті жүйе, мұндағы e ашық, ал d – құпия кілттер. Егер $0 \le x < n$ – ашық ақпарат болса, онда шифрланған ақпарат келесі түрде алынады:

$$C = x^e (\mathrm{mod}\ n).$$

Шифрды ашу мүмкіндігі келесі теорема бойынша анықталады:

Теорема. Егер p және q – үлкен қарапайым сандар болса және $ed(\mathrm{mod}\varphi(n)) = 1$, онда

$$\forall x,\ 0 \le x < n:\ (x^e)^d (\mathrm{mod}\ n) = x.$$

Дәлелдеу. $\text{ҮОБ}(x, n) = 1$ деп алайық. Онда

$$(x^e)^d = x^{e\,d} = x^{k\cdot\varphi(n)+1}.$$

Ферманың кіші Эйлер теоремасы бойынша

$$(x^e)^d (\mathrm{mod}n) = (x(x^{\varphi(n)} (\mathrm{mod}n)))(\mathrm{mod}n) = (x \cdot 1)(\mathrm{mod}n) = x.$$

Егер $\text{ҮОБ}(x,n) \ne 1$ болса, онда

$x \equiv 0(\mathrm{mod}n)$, немесе $\text{ҮОБ}(x,n) = p$, немесе $\text{ҮОБ}(x,n) = q$.

Егер $x \equiv 0(\mathrm{mod}\ n)$ болса, онда

$$x^{e\,d} = 0(\mathrm{mod}n), \quad \text{ҮОБ}(x,n) = p$$

деп аламыз.

Онда

$$x = x_1 p,$$

мұндағы $(x_1, n) = 1$.

$$x^{e\,d} = x^{k(p-1)(q-1)+1} = p x_1 p^{k(p-1)(q-1)} x_1^{\,k(p-1)(q-1)} \equiv y (\mathrm{mod} pq).$$

Егер $mp = y(\mathrm{mod}(pq))$ болса, онда $mp = pqk + y$, сәйкесінше, $y = py_1$. онда $m \equiv y_1 (\mathrm{mod} q$. Сәйкесінше,

$$x_1((px_1)^{k(p-1)})^{q-1} \equiv y_1 \ (\mathrm{mod}\ q).$$

Ферма теоремасы бойынша $z_1^{\,q-1} \equiv 1 \ (\mathrm{mod}\ q)$. Сондықтан

$$x = x_1 p (\mathrm{mod}\ (pq)) = y \ (\mathrm{mod}\ n) \equiv x^{e\,d} \ (\mathrm{mod}\ n).$$

Теорема дәлелденді.

Ашық және шифрланған мәтіндер тиімді анықталынады, егер жылдам дәрежеге жогарылату алгоритмі арқылы e және d белгілі болса. Егер d құпия кілтін белгілі e ашық кілті арқылы іздейтін болсақ, онда $\varphi(n)$–ді білу керек.

Теорема. $\varphi(n) = (p-1)(q-1)$ есептеу (полиномиалды күрделілігіне дәлме-дәл алгоритмге дейін) $n=pq$ факторизация санына тепе-тең.

Дәлелдеу. Айталық, n және $\varphi(n)$ белгілі болсын. Онда p және q жылдам табылады. Бұл келесі теңдеулерден байқалады:

$$\varphi(n) = (p-1)(q-1) = pq-p-q+1 = n-p-q+1.$$

Бұдан

$$p+q = n-\varphi(n)+1, \quad pq = n.$$

Виет теоремасына кері теорема бойынша, p және q квадраттық теңдеулердің түбірлері болып табылады:

$$x^2 - (n - \varphi(n) + 1)x + n = 0.$$

Түбірлерін есептеу – полиномиалды алгоритм болып табылады. Керісінше, егер p және q, $pq = n$ белгілі болса, онда $\varphi(n) = (p-1)(q-1)$ болады. Теорема дәлелденді.

RSA қол қою. M – қол қоятын ақпарат болсын. Қол қоя келесі алгоритм бойынша алынады:

$$C = M^d \pmod{n},$$

мұндағы (M,C) – қол қойылған ақпарат. Қойылған қол келесі түрде тексеріледі:

$$C^e \pmod{n} = M^{e\,d} \pmod{n} = M'.$$

Егер $M = M'$, болса, қойылған қол расталады.

SWIFT ақшаны халықаралық электронды аударымдар желісі, қаржы уақытта оның қызметін пайдаланатын банктік ұйымдардан тек осы криптографиялық жүйені қолдануды талап етіп отыр. Бұл криптографиялық жүйенің алгоритмі төмендегіше:

- Жіберуші екі өте үлкен санды таңдап алады, мысалы P және Q. Содан кейін екі көбейтіндісін есептейді $N = P * Q$ және $M = (P-1) * (Q-1)$;
- Одан соң кез-келген ойдан M-ге қатысты бүтін D санын таңдып алады да $D * E = 1 \bmod M$ шартын қанағаттандыратын E санын есептейді;
- Осы операциялардан кейін ол D мен N-ді ашық шифрлеу кілті ретінде жариялап E-ні жабық ретінде сақтап қалады;
- Егер S-ақпараты, оның ұзындығы өрнек мәні бойынша бүтін сандармен анықталатын және $(1,N)$ интервалында болатын ақпарат N модулі бойынша D дәрежесіне жоғарылатылып шифрланады да, қабылдап алушыға жіберіледі $S' = (S^D) \bmod N$;
- Ақпаратты қабылдап алушы оны N модулі бойынша E дәрежесіне көтеру арқылы шифрды қайта ашады. Мұндағы,

$$S = (S'^E) \bmod N = (S^{(D*E)}) \bmod N$$

Осылайша ашық кілт ретінде N және D сандар жұбы болады да, құпия кілт ретінде E саны болады. Бұл шифрлеу жүйесінің мағынасы Ферма теоремасын ескерсек анықтала түседі. Ол теоремада, қарапайым P саны және кез-келген P-дан кіші K бүтін саны $K^{(P-1)} = 1 \bmod P$ теңдігі орындалады. Бұл теорема қандайда бір санның қарапайым немесе құрама екендігін анықтауға мүмкіндік береді.

RSA әдісімен шифрлеу процесін төмендегіше келтіруге болады:

- Ашық типтегі кез-келген файл болсын, яғни құрамында кез-келген ақпарат болатын мәтіндік файлды таңдап аламыз. Бұл файл RSA әдісінің алгоритмі бойынша шифрланады. Нәтижесінде шифр мәтінді файл құрылады.
- Шифрланған мәтіндік файлды бағдарлама арқылы көрсетіп және бұл файлды RSA әдісінің шифрды қайта ашу алгоритмі бойынша ашамыз.

Шифрлеу коды:

```
AssignFile(F,'crypts\text1.txt');
Reset(F);
 AssignFile(G,'crypts\crypt1.txt');
  rewrite(G);
    While not EoF(F) do
      begin
        read(f,c);
          m := ord(c); d := 19; n := 259;  ot1 := 1;
           for i := 1 to d do
           begin
          ot2 := ((m*ot1) mod n);
         ot1 := ot2;
        end;
       c1 := chr(ot1);
     write(g,c1);
    end;
  CloseFile(F);
CloseFile(G);
```

Дешифрлеу коды:

```
AssignFile(F,'crypts\crypt1.txt');
 Reset(F);
  AssignFile(G,'crypts\uncrypt1.txt');
    rewrite(G);
      While not EoF(F) do
        begin
          read(f,c);
           m := ord(c);
          d := 91;  n := 259;   ot1 := 1;
         for i := 1 to d do
        begin
     ot2 := ((m*ot1) mod n);   ot1 := ot2;
    end;
 c1 :=  chr(ot1);
write(g,c1);
end;
 CloseFile(F);
   CloseFile(G);
```

7 DES ШИФРЛАРЫ

Американдық криптографтар стандартты DES (Data Encryption Algorithm Standard) криптографиялық ауыстыру негізіне сүйенетін алгоритмді ұсынды. Оның бір қадамын келесі схемамен көрсетуге болады. Берілгендердің кіріс блогы теңдей сол жақ L' және оң жақ R' болып екіге бөлінеді. Осыдан кейін шығыс массиві құрылады. Оның сол жақ L'' бөлігі оң жақ кіріс R' бөлігі секілді ұсынылып, ал оң жақ R'' бөлігі XOR операциясы арқылы L' мен R'-дің қомындысы ретінде түзіледі. Ары қарай шығыс массиві ауыстыру мен орын ауыстыру арқылы шифрланады. Барлық операциялардың ауыстырылғандығына және шифрды қайта ашу блок өлшеміне байланысты операция сандарына сызықты тәуелді болатынына көз жеткізуге болады. Жәнеде бірнеше осындай бөлшектеуден кейін шифрлеудің шығыс блогының әр биті ақпараттың әр битіне тәуелді болуы мүмкін.

DES шифрлеу жүйесін IBM фирмасы Lucifer деген атпен құрған және өзінің корректировкаларымен бірге 1976 жылы шифрлеу стандарты ретінде АҚШ Ұлттық Стандарттар Бюросына ұсынылған. Онда 56 биттен тұратын кілт қолданылған. Мұнда айта кететін мәселе DES стандартында арнайы типтегі орын ауыстырулар ғана қолданылғандықтан бұл стандарттың сыншылары, АҚШ Ұлттық Бюросы олардың теориясын өте жақсы білген және шифрды бұзу үшін алдын-ала белгілі болған әлсіз жерлерді пайдаланулары мүмкін дегенді алға тартқан. Бірақта бұл шифрлеу принципі ең кең көлемдегі апробациядан өтіп, оған көптеген публикациялар арналған. Сыншылардың бос сөзі тек 64 битті блок және 56 бит кілт ұзындығына байланысты қарап қалған ұлттық қауіпсіздік мәселелері жоқ деген болды. Бұл шифр криптоалгоритмнің ең үздік мысалы бола алады (11-кесте).

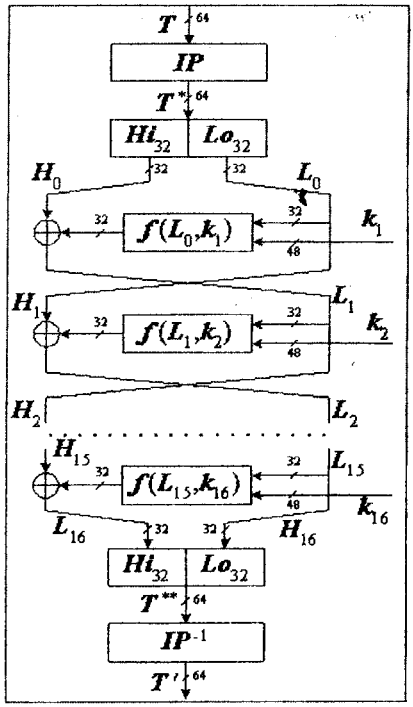
Кесте 11 - DES шифрлеу жүйесінің негізгі параметрлері

Аталуы/статусы	DES (Data Encryption Standard). 1977-2001 жылдардағы АҚШ федералды шифрлеу стандарты.
Уақыты/құрылу орны	1972-1975 жылдары IBM корпорациясының зерттеу лабораториясында құрылған. АҚШ федералды стандарты ретінде 1977 жылы қабылданған. 2001 жылдың желтоқсанында жаңа стандарттың енгізілуіне байланысты өз статусын жоғалтты.
Авторлары	Доктор У.Тачменнің басқаруындағы топ.
Архитектура	Классикалық теңестірілген жалпы түрдегі Файстель желісі, бастапқы және соңғы биттік орын ауыстыру.
Параметрлері	Блок өлшемі, бит 64 Кілт өлшемі, бит 56 Раунд саны 16 кілттік элемент өлшемі, бит 48 Кілттік элемент өлшемі 16 (раунд санына тең)
Патент	Патенттелмеген.
Ерекшеліктері	DES-те биттік орын ауыстыруларды кеңінен қолдану алгоритмді әмбебап процессорларда бағдарламалық жолмен іс-жүзіне асыруды қолайсыз етеді, ал мұндай іс-жүзіне асырулар тиімсіз болып есептеледі. DES-ті ресейлік шифрлеу стандартымен салыстырғанда, оның құрамында екі есе аз раунд саны бар, бірақ оны Intel x86 процессорларында тиімді түрде іс-жүзіне асыру ресейлік стандарттан жылдамдығы процессордың маркасына байланысты 3-5 есе аз, бұл айырмашылық ескі модельдерден жаңа модельдерге қарай ұлғая береді.

Алгоритмнің жалпы схемасы. Алгоритм теңестірілген жалпы түрдегі Файстель желісі, бастапқы және соңғы биттік орын ауыстыру ретінде болып келеді және соңғы орын ауыстыру бастапқысына байланысты болады. Блокты шифрлеуде берілгендерді айналдыру схемасы 2-суретте келтірілген, сәйкес алгоритмнің схемасы 3–суретте келтірілген.

64-биттік T берілгендер блогын шифрлеу (алгоритмнің кіріс параметрі, сурет 9, қадам 0) ондағы бастапқы биттердің орын ауыстыруынан басталады. (*IP*, қадам 1). Одан соң шифрланатын блок екі 32-биттік бөлікке бөлінеді (қадам 2), онымен 16 раундты айналдыру орындалады, ол Файстель желі архитектурасының принципіне сәйкес орындалады. $Hi_n(X)$ және $Lo_n(X)$ арқылы

функциялар берілген, олардың нәтижесіне өзінің аргументтеріне сәйкес n жоғары немесе кіші биттері тең болады.



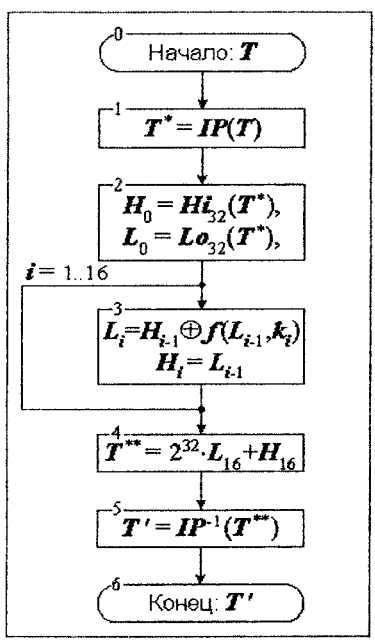
Сурет 2 - DES шифрлеу циклы - берілгендерді айналдыру схемасы

Әр раундта блоктың үлкен жартысы 2 модульмен бит бойынша қосылу ("⊕" операциясы) (f) шифрлеу функциясының есептеу нәтижесі, (L_{i-1}) және 48-биттік (k_i) кілттік элементті блоктың кіші жартысына тәуелді жолмен модифицияланады. Раундтардың арасында блоктың жоғары және кіші жартысы орын ауысады. Осылайша әр раундта блоктың соңғы кіші жартысынан басқасы жоғарыға айналады, ал шифрлеу функциясы арқылы модифицияланған жоғары блок – кіші блокқа айналады (қадам 3).

Соңғы раундта блок жартыларының мәндерінің ауысуынан басқа дәл соның өзі орындалады. Одан соң жарты блоктар толық бір блокқа бірігеді (қадам 4), онда соңғы айналымды биттік орын ауыстыру операциясы орындалады (IP^{-1}, қадам 5). Соңғы

операцияның нәтижесі T' шифрланған блоктың - шифрлеу циклының шығыс мәні болып табылады (қадам 6).

Он алты 48-биттік кілттік элементтер k_i, $1 \leqslant i \leqslant 16$, шифрлеу циклында колданылады да, алгоритмнің параметрлері болып табылады және төменде қарастырылған кілттік тізбек генерация процедурасы 56-биттік кілттен өңделеді. Берілгендер блогының шифрын ашу процедурасы шифрлеу процедурасы іспеттес, яғни ондағы кілттік элементтер бір ретпен орналасса, ал шифрлеу кезінде олар кері ретпен орналасады.



Сурет 3 - DES шифрлеу циклы - алгоритмнің схемасы

Ескеріңіз, егер шифрлеу кезінде кілттік элементтер «табиғи» ең ретпен орналасса

$$k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}, k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16},$$

онда шифрды ашу кезінде олар кері ретпен орындалуы керек:

$k_{16}, k_{15}, k_{14}, k_{13}, k_{12}, k_{11}, k_{10}, k_9, k_8, k_7, k_6, k_5, k_4, k_3, k_2, k_1$.

Бастапқы және соңғы биттік орын ауыстырулар. DES алгоритмінің бастапқы және соңғы биттік орын ауыстырулары төменде келтірілген кестеде берілген. Кестедегі әр ұяшық берілгендер блогындағы белгілі бір битке сәйкес болады, кестедегі кіші номермен көрсетілген сандар. Әр кесте екі вариантта көрсетілген, олар блоктардағы биттердің нөмірлерін қолданудың екі әдісіне сәйкес. Сол жақ бөлікте биттерді нөмірлеу варианты келтірілген, ол – үлкен биттердің бірлігінен басталып кішілеріне қарай оригинал стандарт бойынша қабылданған. Intel архитектурасының кең таралуын ескере отырып, кестенің оң жақ бөлігінде – кіші биттерден үлкенге қарай нөлден басталатын қабылданған нөмірлеу варианты келтірілген. Ұяшықтардағы орын ауыстыруларды анықтаған кезде ағымдағы блоктағы бит нөмірі көрсетіледі, ол орын ауыстыру кезінде сәйкес ұяшықтағы биттің орнына орналасады. Мысалы, кестеге сәйкес бастапқы орын ауыстырулар ең үлкен бит нәтижесінің орнына (сол жақ кестеде N1 және оң жақ кестеде N63) ағымдағы блоктағы N58 бит орналасады (оң жақ кестеде N6), N2(N62) биттің орнына N50(N14) бит, және т.с.с. (12-13-кестелер)

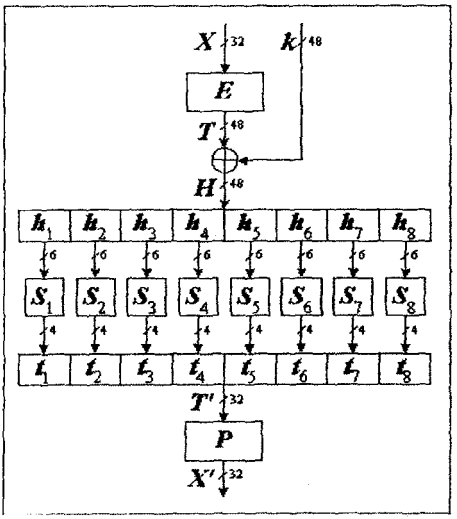
Кесте 12 - Бастапқы биттік орын ауыстыру (IP)

$_{1}58$	$_{2}50$	$_{3}42$	$_{4}34$	$_{5}26$	$_{6}18$	$_{7}10$	$_{8}2$	← үлкен байт→	$_{63}6$	$_{62}14$	$_{61}22$	$_{60}30$	$_{59}38$	$_{58}46$	$_{57}54$	$_{56}62$
$_{9}60$	$_{10}52$	$_{11}44$	$_{12}36$	$_{13}28$	$_{14}20$	$_{15}12$	$_{16}4$		$_{55}4$	$_{54}12$	$_{53}20$	$_{52}28$	$_{51}36$	$_{50}44$	$_{49}52$	$_{48}60$
$_{17}62$	$_{18}54$	$_{19}46$	$_{20}38$	$_{21}30$	$_{22}22$	$_{23}14$	$_{24}6$		$_{47}2$	$_{46}10$	$_{45}18$	$_{44}26$	$_{43}34$	$_{42}42$	$_{41}50$	$_{40}58$
$_{25}64$	$_{26}56$	$_{27}48$	$_{28}40$	$_{29}32$	$_{30}24$	$_{31}16$	$_{32}8$		$_{39}0$	$_{38}8$	$_{37}16$	$_{36}24$	$_{35}32$	$_{34}40$	$_{33}48$	$_{32}56$
$_{33}57$	$_{34}49$	$_{35}41$	$_{36}33$	$_{37}25$	$_{38}17$	$_{39}9$	$_{40}1$		$_{31}7$	$_{30}15$	$_{29}23$	$_{28}31$	$_{27}39$	$_{26}47$	$_{25}55$	$_{24}63$
$_{41}59$	$_{42}51$	$_{43}43$	$_{44}35$	$_{45}27$	$_{46}19$	$_{47}11$	$_{48}3$		$_{23}5$	$_{22}13$	$_{21}21$	$_{20}29$	$_{19}37$	$_{18}45$	$_{17}53$	$_{16}61$
$_{49}61$	$_{50}53$	$_{51}45$	$_{52}37$	$_{53}29$	$_{54}21$	$_{55}13$	$_{56}5$		$_{15}3$	$_{14}11$	$_{13}19$	$_{12}27$	$_{11}35$	$_{10}43$	$_{9}51$	$_{8}59$
$_{57}63$	$_{58}55$	$_{59}47$	$_{60}39$	$_{61}31$	$_{62}23$	$_{63}15$	$_{64}7$	←кіші байт→	$_{7}1$	$_{6}9$	$_{5}17$	$_{4}25$	$_{3}33$	$_{2}41$	$_{1}49$	$_{0}57$
DES оригиналында биттерді нөмірлеу									Intel-де биттерді нөмірлеу							

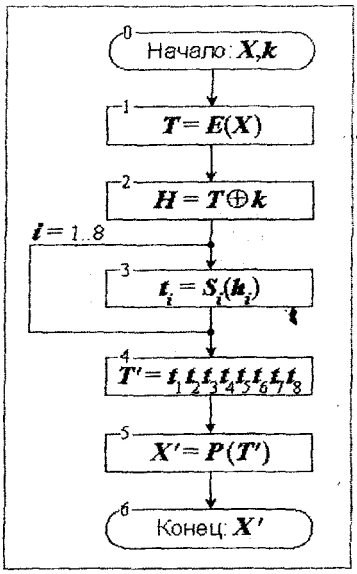
Кесте 13 - Соңғы биттік орын ауыстыру (IP^{-1})

40	8	48	16	56	24	64	32	← үлкен байт→	24	56	16	48	8	40	0	32
39	7	47	15	55	23	63	31		25	57	17	49	9	41	1	33
38	6	46	14	54	22	62	30		26	58	18	50	10	42	2	34
37	5	45	13	53	21	61	29		27	59	19	51	11	43	3	35
36	4	44	12	52	20	60	28		28	60	20	52	12	44	4	36
35	3	43	11	51	19	59	27		29	61	21	53	13	45	5	37
34	2	42	10	50	18	58	26		30	62	22	54	14	46	6	38
33	1	41	9	49	17	57	25	←кіші байт→	31	63	23	55	15	47	7	39
DES оригиналында биттерді нөмірлеу									Intel-де биттерді нөмірлеу								

Шифрлеу функциясы. DES алгоритмінде салыстырмалы түрде қарапайым шифрлеу функциясы қолданылады. Берілгендерді айналдыру схемасы төмендегі суретте көрсетілген (сурет 4), ал алгоритмнің схемасы 5-суретте көрсетілген.



Сурет 4 - Шифрлеу функциясы – берілгендерді айналдыру схемасы



Сурет 5 - Шифрлеу функциясы–алгоритмнің схемасы

Шифрлеу функциясына 32-биттік X блоктың шифрланатын жартысы, және k 48-биттік кілттік элемент түседі (сурет 4, қадам 0). Төменде көрсетілген кестеге сәйкес бірнеше екілік разрядты 32-биттік берілгендер блогы 48-битке дейін қайталану арқылы ұлғаяды (E, қадам 1). Одан кейін ұлғайтылған блок 2 модульдік бит бойынша кілттік элементпен қосылады ("⊕" операциясы, қадам 2). Алынған нәтижелік 48-биттік берілгендер блогының қосындысы сегіз 6-биттік берілгендер элементтеріне бөлінеді, олар сәйкесінше $h_1,h_2,\ldots,h_8$ болып белгіленген, мұнда h_1 құрамында 48-биттегі ең үлкен алты биттер бар, h_2 құрамында үлкендігі жағынан келесі алты бит орналасқан және т.с.с., ең соңында h_8 – блоктың ең кіші биттері орналасады.

Ары қарай h_i әр мәні жаңа t_i 4-биттік мәндеріне ауыстыру түйіні арқылы айналдырылады (S_i, қадам 3). Бұдан кейін алынған сегіз 4-биттік берілгендер элементтері қайтадан 32 T' биттік блогына сол ретпен біріктіріледі. (қадам 4). Ең соңында алынған 32-биттік блокта биттердің орын ауыстырулары орындалады (P, қадам 5), ол

төмендегі кестеде көрсетілген. Соңғы операция нәтижесі шифрлеу функциясының шығыс мәні болып табылады.

Кесте 14 - 32-биттік блоктың 48 битке дейін ұлғайтылуы (E)

$_{0}32$	$_{2}1$	$_{3}2$	$_{4}3$	$_{5}4$	$_{6}5$	$\leftarrow h_1 \rightarrow$	$_{47}0$	$_{46}31$	$_{45}30$	$_{44}29$	$_{43}28$	$_{42}27$
$_{4}$	$_{8}5$	$_{9}6$	$_{10}7$	$_{11}8$	$_{12}9$	$\leftarrow h_2 \rightarrow$	$_{41}28$	$_{40}27$	$_{39}26$	$_{38}25$	$_{37}24$	$_{36}23$
$_{08}$	$_{14}9$	$_{15}10$	$_{16}11$	$_{17}12$	$_{18}13$	$\leftarrow h_3 \rightarrow$	$_{35}24$	$_{34}23$	$_{33}22$	$_{32}21$	$_{31}20$	$_{30}19$
$_{12}$	$_{20}13$	$_{21}14$	$_{22}15$	$_{23}16$	$_{24}17$	$\leftarrow h_4 \rightarrow$	$_{29}20$	$_{28}19$	$_{27}18$	$_{26}17$	$_{25}16$	$_{24}15$
$_{16}$	$_{26}17$	$_{27}18$	$_{28}19$	$_{29}20$	$_{30}21$	$\leftarrow h_5 \rightarrow$	$_{23}16$	$_{22}15$	$_{21}14$	$_{20}13$	$_{19}12$	$_{18}11$
$_{20}$	$_{32}21$	$_{33}22$	$_{34}23$	$_{35}24$	$_{36}25$	$\leftarrow h_6 \rightarrow$	$_{17}12$	$_{16}11$	$_{15}10$	$_{14}9$	$_{13}8$	$_{12}7$
$_{24}$	$_{38}25$	$_{39}26$	$_{40}27$	$_{41}28$	$_{42}29$	$\leftarrow h_7 \rightarrow$	$_{11}8$	$_{10}7$	$_{9}6$	$_{8}5$	$_{7}4$	$_{6}3$
$_{28}$	$_{44}29$	$_{45}30$	$_{46}31$	$_{47}32$	$_{48}1$	$\leftarrow h_8 \rightarrow$	$_{5}4$	$_{4}3$	$_{3}2$	$_{2}1$	$_{1}0$	$_{0}31$
DES оригиналында биттерді нөмірлеу							Intel-де биттерді нөмірлеу					

Берілген схема тек бит жұптарын ғана қайталайды (32-1, 4-5, 8-9, ...,28-29) және келесі өрнекпен көрсетуге болады:

$$Lo_6(\boldsymbol{R}_{\leftarrow 4i+1}(X)),$$

Мұндағы $Lo_n(X)$ жоғарыда анықталған, ал $\boldsymbol{R}_{\leftarrow n}(X)$ өзінің X аргументінің мәнін беретін функцияны береді, ол n битке циклдық түрде солға қарай жылжып отырады.

Биттік топта ауыстырулар (S_i). 6-биттік блокты 4-биттік блокқа ауыстыру келесі ереже бойынша орындалады: ауыстырудың әр түрін 4×16 өлшемді кесте (матрица) бойынша беріледі, оның әрбір жасушасында 4-биттік элемент орналасады, әр қатардағы барлық элементтер әртүрлі. Ауыстырылатын 6-биттік мән екі бөлікке бөлінеді: блоктың үлкен және кіші биттерінен бүтін екі биттік блок құрылады, олар 0-ден 3-ке дейінгі мәндерді қабылдайды және қатар нөмірі ретінде қолданылуы мүмкін, орта төрт бит 0..15 аралығында бүтін сан ретінде интерпретацияланады және баған нөмірін көрсетеді. Осылайша табылған кесте ұяшықтарынан 4-биттік элемент таңдап алынады, ол ауыстыру операциясының нәтижесі болып табылады.

8 КРИПТОЖҮЙЕЛЕРГЕ ҚОЙЫЛАТЫН ТАЛАПТАР

Бұл бөлімде криптожүйені таңдау және криптографиялық алгоритмдерді іске асыру мәселеріне қысқаша тоқтайық. Криптографиялық жүйені таңдауға әсер етуші сипаттамаларға, ең алдымен, оның жұмыс істеу жылдамдығына қойылатын талап жатады. Егер криптожүйенің шифрлеу жылдамдығына жоғары талаптар қойылмаса, онда RSA криптографиялық жүйесін алуға болады. Блоктық шифрлеу криптожүйелерінің бағдарламалық түрі жылдамдығы төмен жүйелер қатарына жатады. Ал оның техникалық нұсқасы, мысалы, DES алгоритмі, кез келген жылдамдықпен жұмыс істей алады. Егер өте жоғары жылдамдық қажет болса, онда ағындық шифрлеу жүйесін қолданған жөн. Олардың бағдарламалық та, техникалық та түрлерінің жылдамдығы жоғары болып келеді. Егер қолданылатын байланыс арнасы қателерге душар арналар қатарына жататын болса, онда бұл қателерді криптожүйе одан әрі көбейтпеуі үшін ағындық шифрлеу жүйесін таңдап алған тиімді болады.

Әрі қарай, жоғарыда сипатталған алгоритмдер және стандарттар калай амалға асырылатынын карастырайық. Әрине, криптожүйелерді жобалау кезінде оларды тиімді және арзан бағдарламалар немесе құрылғылар түрінде құру мүмкіндігі көзделуі тиіс.

Техникалық шифрлеу құрылғыларына жоғары жылдамдық пен сенімділік тән. Олар электрондық тақшалар түрінде құрылады. Ақпаратты сақтау құрылғылары да техникалық түрде амалға асырылады. Оларға электрондық кілттер, пластикалық карталар және т.б. жатады. Бағдарламалық шифрлеу әмбебап есептеу техникасы көмегімен амалға асырылады. Олар техникалық шифрлеу құрылғыларымен салыстырғанда арзан және икемді.

Жалпы, криптографиялық алгоритмдерді қолданатын барлық бағдарламалар екіге бөлінеді: мамандандырылған кешендер және шифрлеу әдісі біреу болып келетін криптографиялық функциялар. Мұндай криптографиялық функциялар көптеген амалдық жүйелердің құрамына кіреді.

Әрі қарай симметриялық шифрлеу алгоритмдері мен криптожүйелерге қойылатын талаптарды қарастырайық.

Қолдану облысына қойылатын талаптар. Шифрлеу алгоритмдеріне қойылатын талаптардың бірі, олар әмбебап түрде құрылып, көптеген қосымшаларда қолданылуы тиіс:

- деректерді шифрлеу. Алгоритм деректер файлын және деректер ағымын шифрлеуде тиімді болуы тиіс;
- жалғанкездейсоқ сандарды генерациялау. Алгоритм нақты көлемді жалғанкездейсоқ биттерді құруда тиімді болуы тиіс;
- хэштеу. Алгоритм бір жаққа түрлендіретін хэш-функцияны құруда тиімді болуы тиіс.

Платформаларға қойылатын талаптар. Шифрлеу алгоритмдері барлық платформалы компьютерлерде қолданылуы тиіс. Орине, әр түрлі платформалы компьютерлерге қойылатын талаптар да әр түрлі болады, соның ішінде:

- арнайы техникалық құрылғылар. Алгоритм шифрлеу және дешифрлеуді орындауға арналған арнайы техникалық құрылғыларда тиімді болуы тиіс;
- үлкен процессорлар. Жылдам орындалатын қосымшалар арнайы техникалық құрылғыларда жасалғанымен, олар бағдарламалық түрінде де қолданылуы қажет. Алгоритм 32 және 64 разрядты процессорларда тиімді болуы тиіс;
- орташа өлшемді процессорлар. Алгоритм микроконтроллерде, чиптерде, т.б. орташа өлшемді процессорларларда қолданылуы тиіс;
- кіші процессорлар. Алгоритм пайдаланылатын жадысы өте кіші болса да смарт-карталарда қолданылуы тиіс.

Шифрлеу алгоритмдері жоғарыда көрсетілген талаптармен қатар, мүмкіндігі барынша төмендегі қосымша талаптарды да қанағаттандыруы тиіс:

- алгоритм бағдарлама кодын жазуда, мүмкін бағдарламалық қателерді болдырмау үшін қарапайым болуы тиіс,
- алгоритм құпия кілттердің жиынына ие болуы тиіс. Сонымен қатар кілт ұзындығы ретінде кез келген биттер тізбегін алу мүмкіндігі қарастылуы тиіс. Жеңіл кілттердің болмағаны дұрыс;
- алгоритм әртүрлі қауіпсіздік деңгейлеріне бағытталуы және осы деңгейлердің максимум және минимум талаптарын қанағаттандыруы тиіс;

• алгоритмдегі деректер үстіндегі амалдар байтқа немесе 32 разрядты машина сөзіне еселік болуы тиіс.

Криптожүйелерге қойылатын талаптар. Жоғарыда айтылғандарға сәйкес, ақпаратты криптографиялық түрлендіру жүйелеріне төмендегі талаптар тұжырымдалған:

• шифрленген ақпарат тек қана құпия кілт болғанда ғана ашылуы тиіс;

• ашық мәтін және оған сәйкес криптомәтін фрагментіні шифрлеу кілтін анықтауға арналған амалдар саны, жалпы мүмкін кілттер санынан аз болмауы тиіс;

• ақпаратты кездейсоқ кілттермен таңдап ашу әдісіндегі кілттер саны барлық мүмкін кілттер санынан сезілерлі дәрежеде төмен болуы тиіс (желілік есептеулер мүмкіндіктерін есепке алғанда);

• шифрлеу әдісін білу ақпаратты қорғау сенімділігіне әсер етпеуі тиіс;

• құпия кілттегі кішігірім өзгеріс шифрленген криптомәтінге күрделі өзгерістер енгізуі тиіс;

• шифрлеу алгоритмінің құрылымдық элементері тұрақты болуы тиіс;

• шифрлеу үрдісі барысында мәтінге енгізілетін қосымша биттер шифрленген криптомәтінде сенімді түрде толық білінбейтін болуы тиіс;

• шифрленген криптомәтін ұзындығы бастапқы ашық мәтін ұзындығымен бірдей болуы тиіс;

• шифрлеу әдісінде тізбектеліп пайдаланылатын кілттер өзара тәуелсіз болуы тиіс;

• құпия кілттер жиынындағы әрбір кілт ақпаратты сенімді түрлендіруі тиіс.

Ақпаратты криптографиялық түрлендіру үрдісі техникалық және бағдарламалық түрде орындалуы белгілі. Ақпаратты техникалық құрылғылармен түрлендіру қаржылық тұрғыдан қымбат болғанымен, оның өзіне тән қасиеттері бар, яғни жоғары өнімді, қарапайым, жоғары дәрежеде корғалған және т.б. Ал бағдарламалық түрлендіру практикалық тұрғыдан маңызды, қолдануға икемді.

9 СИММЕТРИЯЛЫҚ КРИПТОГРАФИЯ АЛГОРИТМДЕРІН БАҒДАРЛАМАЛАУ

Криптография әдістерін қолданудың қолмен шифрлеу және бағдарламалық шифрлеу тәсілдері арасында айырмашылықтар бар. Қол шифрлеріне негізінен классикалық әдістер жатады. Сонымен қатар, олармен шифрленетін ашық мәтіндердің ұзындығы қысқа болады. Осы себептен криптомәтіндерді кері шифрлеу нәтижелі орындалады. Ал бағдарламалық шифрлер есептеу жағынан өте күрделі және өте ұзын ашық мәтіндерді шифрлеуге арналған. Бағдарламалық шифрлеу тәсілдеріне қойылатын талаптар мен ұсыныстарды қарастырайық:

- деректерді мүмкіндігі барынша 16 немесе 32 биттен өңдеу қажет;
- блок ұзындығы 64 немесе 128 бит болғаны дұрыс;
- құпия кілт ұзындығы 256 битке дейін жылжымалы түрде болуы тиіс;
- мүмкіндігі барынша микропроцессорда жылдам орындалатын, қарапайым амалдарды қолдану қажет, мысалы, биттік амалдар, арифметикалық модуль, конъюнкция, дизъюнкция т.б.
- мүмкіндігі барынша жай орындалатын немесе кателік пайда болатын амалдарда қолданбаған жөн, мысалы, шартты өту, биттік орын ауыстыру, айнымалы ұзындықты биттік жылжыту т.б.
- алгоритмнің 8 биттік микропроцессорларда орындалатын, жедел жадығы минимум талап қоятын версиялары болуы тиіс;
- мүмкіндігі барынша кілттерді алдын-ала есептеп алу керек. Күрделі шифрлеу жүйелерінде кілттерді алдын-ала есептейтін блоктың модуль болғаны дұрыс;
- алгоритмдегі итерациялардың саны айнымалы болғаны дұрыс. Бағдарламада кілт ұзындығы қысқа болса, онда криптобериктілікті жоғарылату үшін, итерация санын көбейткен дұрыс. Сол сияқты, кілт ұзындығы жетерлі дәрежеде ұзын болса, онда сенімділікті бақылай отырып, итерация санын азайтуға болады (жылдамдықты асыру үшін);
- мүмкіндігі барынша жеңіл кілттерді бағдарламалық түрде алып тастау қажет. Егер ондай мүмкіндік болмаса, онда алдын-ала есептелген кілттердің ішінен визуалды түрде алып тастау қажет;
- кілттің бір жақты хэш-функциясының аргументі ретінде ішкі кілттерден пайдаланған дұрыс. Олар бағдарламада қауіпсіздік

қызметінің парольдері ретінде, сол сияқты ұзын кілттерден пайдалану мүмкіндігін береді.

Криптографиялық әдістерге бағдардама құрудан алдын, бағдарламалауда қолданылатын қарапайым амалдарды қарап шығайық.

Арифметикалық амалдар. Бүтін және нақты типті деректер үстінде төмендегі арифметикалық амалдар орындалады (15-кесте):

Кесте 15 – арифметикалық амалдар

-	унарлық минус
+	унарлық плюс
+	қосу
-	алу
/	бөлу
*	көбейту
div	бүтін бөлу
mod	арифметикалық модуль – бүтін бөлу амалының қалдығы

Логикалық амалдар. Логикалық амалдар тек қана логикалық *true* және *false* мәндері үстінде орындалады. Логикалық теріске шығару амалы бір аргументті, ал калған амалдар екі аргументті (16-17-кестелер).

Кесте 16 – логикалық амалдар

not	теріске шығару
and	конъюнкция (логикалық көбейту)
or	дизъюнкция (логикалық қосу)
xor	логикалық алу

Кесте 17 – логикалық амалдар нәтижесі

a	b	not a	not b	a and b	a or b
false	false	true	true	false	false
false	true	true	false	false	true
true	false	false	true	false	true
true	true	false	false	true	true

Мысалы: (x <= 1) and (x >= 7) амалдарын нәтижесін табайық.
- x = 0 болса, нәтиже *false;*
- x = 5 болса, нәтиже *true;*
- x = 9 болса, нәтиже *false;*

Салыстыру амалдары. Деректер үстінде төмендегі салыстыру амалдары орындалады (18-кесте):

Кесте 18 – салыстыру амалдары

>	үлкен
>=	үлкен немесе тең
<	кіші
<=	кіші немесе тең
=	тең
<>	тең емес

Салыстыру амалының нәтижесі *true* немесе *false* мәндеріне ие. Бұл амалдарда деректер кез келген типке, ал нәтиже тек қана логикалық типке тиісті.

Мысалы: *a = 5; b = 3; болсын, онда*
a >= b - true;
a = b - false болады.

Егер салыстырылатын деректер әр түрлі типке ие болса, онда қатаң түрде тең (тең емес) амалдарын қолданған жөн.

Биттік амалдар. Биттік амалдар төменгі деңгейлі программалауда қолданылады (19-20-кестелер).

Кесте 19 – биттік амалдар

not	инверсия – теріске шығару
and	конъюнкция – логикалық көбейту
or	дизъюнкция – логикалық қосу
xor	логикалық алу
shl	солға жылжыту
shr	оңға жылжыту

Мысалы: c = not a; - инверсия амалы;
c = a and b; - конъюнкция амалы;
c = a shl b; - a-ны солға b-разрядқа жылжыту;
c = a shr b; - a-ны оңға b-разрядқа жылжыту;

Кесте 20 – биттік амалдардың мәндері

a	b	not a	not b	a and b	a or b	a xor b
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

Әрі қарай, симметриялық алгоритмді шифрлеу әдістеріне бағдарлама құрайық.

Ауыстыру шифрлерін бағдарламалау. Ауыстыру шифрлары модульдік арифметикаға негізделген. Шифрлеу әдісі аналитикалық түрде былай жазылады:

$$E_k(M) = C \tag{8}$$

Сол сияқты, шифрден шығару аналитикалық түрде былай жазылады:

$$D_k(C) = M \tag{9}$$

Бұл жерде: E - Encrypt (шифрлеу), D - Decrypt (дешифрлеу), M - Message (ашық мәтін), C - Ciphertext (криптомәтін), k - кілт сөз.

Айталық, шифрлеу әдісі (Юлий Цезарь шифры), әліпби әріптерін k орынға жылжытады деп алайық, мұндағы k -шифрдің кілті. Шифрлеу әдісі әліпбидің i-ші әрпін $i+k$ әрпімен ауыстырады деп болжайық (n -әліпбидегі әріптер саны). Онда шифрлеу аналитикалық түрде былай жазылады:

$$E_k(i) = (i + k) \bmod n \tag{10}$$

Ал, шифрден шығару:

$$D_k(i) = (i + n - k) \bmod n \qquad (11)$$

Жоғарыда көрсетілген ауыстыру әдісі қарапайым және оны компьютерде бағдарламалық жолмен оңай көрсетуге болады. Бағдарламаны іс-жүзіне асыру үшін ASCII код символдары (American Standard Code for Information Interchange - ақпаратты алмасудың американдық стандарт коды) қолданылады. Сондықтан бұл әдістің 256 мүмкін болатын кілті бар.

Ауыстыру әдісі үшін шифрлеу үрдісін келесі түрде сипаттауға болады:

- ашық мәтін txt кеңеймесі бар мәтіндік файлда болсын. Файл ауыстыру әдісі алгоритмі бойынша шифрленеді. Нәтижеде шифрленген мәтінді файл құрылады.
- Шифрленген мәтіндік файлды ауыстыру әдісінің шифрден шығару алгоритмі бойынша ашылады. Нәтижеде ашық мәтінді файл құрылады.

Шифрлеу коды:

```
kilt := StrToInt (s);
  assignfile (f01, 'crypts\crypt1.txt ');
    {ашық мәтін файлы}
      reset (f01);
      assignfile (f02, 'crypts\uncrypt1.txt');
      {криптомәтін файлы}
    rewrite (f02);
  seek (f01, 0);
 seek (f02, 0);
 while not eof (f01) do
 begin
   read (f01, sym);
     crypt := chr ((ord (sym)+ kilt) mod 256);
    write (f02, crypt);
   end;
  closefile (f01);
closefile (f02);
```

Дешифрлеу коды:

```
kilt := StrToInt (Edit1.Text);
 assignfile (f01, 'crypts\crypt1.txt');
  {криптомәтін файлы}
    reset (f01);
     assignfile (f02, 'crypts\uncrypt1.txt');
       {ашық мәтін файлы}
        rewrite (f02);
        seek (f01, 0);
       seek (f02, 0);
      while not EoF (f01) do
     begin
   read (f01, sym);
  crypt := chr ((ord (sym) - kilt) mod 256);
write (f02, crypt);
end;
 closefile (f01);
   closefile (f02);
```

Орын ауыстыру шифрлерін бағдарламалау. Орын ауыстыру шифрлері дегенде ашық мәтіндегі әліпби әріптерінің қандайда бір ереже бойынша орнын ауыстыруды айтады. Мұндай шифрлеу әдісіне мысал ретінде Кардано торларын келіруге болады.

Шифрдің математикалық моделі келесі түрде жазылады, яғни

$$A = (X, S_T, Y, f),$$

мұндағы $f : X \times K \to Y$, X - ашық мәтіндер жиыны,

$K = S_T$ - кілттер жиыны,

Y - шифрленген мәтіндер жиыны (криптомәтін).

Орын ауыстыру әдісі үшін шифрлеу үрдісін келесі түрде сипаттауға болады:

• Берілген ашық мәтін txt кеңеймесі бар мәтіндік файлда сақталсын. Файл орын ауыстыру әдісі алгоритмі бойынша шифрленеді. Нәтижеде шифрленген мәтіндік файл құрылады.

• Шифрленген мәтіндік файлды орын ауыстыру әдісінің шифрден шығару алгоритмі бойынша дешифрленеді. Нәтижеде ашық мәтіндік файл құрылады.

Шифрлеу коды:

```
assignfile (f01, 'crypts\crypt1');
  {ашық мәтін файлы}
    reset (f01);
      assignfile (f02, ' crypts\uncrypt1.txt.txt');
       {криптомәтін файлы}
          rewrite (f02);
           kilt := (FileSize (f01) mod 32);
            if kilt <>0 then
            for i:=1 to 32- kilt do
           begin
         seek (f01, filesize (f01));
       sym := chr (32);
     write (f01, sym);
   end;
  closefile (f01);
   randomize;
    for i:=1 to 64 do
      str := str + chr (random (255));
        reset (f01);
           for i:=1 to 64 do
              write (f02, str [i]);
               while not EoF (f01) do
                 begin
                 katar := '';
                for i:=1 to 32 do
               begin
             if eof (f01) then exit;
           read (f01, sym);
        katar := katar + sym;
      end;
    for i:=1 to 63 do
   begin
  m := ord (str [i+1]) mod 32;
```

```
        n := ord (str [i]) mod 32;
        if n>m then
         begin
          bufer := m;
            m := n;
              n := bufer;
                end;
                 katar := copy (katar, m+1,32-m) + copy (katar, n+1,m-
n)
                  + copy (katar, 1, n);
                 end;
                for i:=1 to length (katar) do
               write (f02, katar [i]);
             end;
          closefile (f01);
       closefile (f02);
```

Дешифрлеу коды:

```
assignfile (f01, 'crypts\crypt1.txt.txt');
 {криптомәтін файлы}
   reset (f01);
    assignfile (f02,'crypts\uncrypt1.txt ');
      {ашық мәтін файлы}
       rewrite (f02);
         str := '';
          for i:=1 to 64 do
           begin
            read (f01, sym);
            str := str + sym;
           end;
          while not eof (f01) do
         begin
        katar := '';
       for i:=1 to 32 do
     begin
    if eof (f01) then exit;
  read (f01, sym);
```

```
  katar := katar + sym;
end;
for i:=length (str) downto 2 do
 begin
   n := ord (str [i-1]) mod 32;
     m := ord (str [i]) mod 32;
      if n>m then
        begin
          bufer:= m;
            m := n;
            n := bufer;
          end;
         katar := copy (katar, 32-n+1, n) + copy (katar, 32-m+1, m-n)
        + copy (katar, 1, 32-m);
      end;
    for i:= 1 to 32 do
  write (f02, katar[i]);
 end;
closefile (f01);
closefile (f02);
```

Вижинер шифрлеу алгоритмін бағдарламалау. Криптография әдістері ішінен ең танымалы ретінде Вижинер шифрлерін атауға болады.

Оның ерекшелігі ашық мәтіндегі әліпби әріптері сәйкес шифромәтіндегі әліпби әріптерімен ауыстырады. Вижинердің бір әліпбилік және көп әліпбилік деген екі түрі бар. Шифрлеу $n \times n$ өлшемді әліпбилік кесте бойынша жүзеге асырылады. Алдымен кілт таңдап алынады. Одан кейін шифрлеу әдісі келесі түрде орындалады. Ашық мәтіннің әр әрпінің астына тізбектелген түрде кілт әріптері жазылады, егер кілт ашық мәтін ұзындығынан қысқа болса, онда ол бірнеше рет қайталанып жазылады. Шифрмәтіннің әр әрпі ашық мәтіннің әріптерін анықтайтын кесте бағанының, кілт әріптерін анықтайтын қатар қиылысында орналасады. Мұндай амал ашық мәтін мен кілт символдарының ASCII кодтарын белгілі бір модуль бойынша қосу болып есептеледі.

Вижинер әдісі үшін шифрлеу үрдісін келесі түрде сипаттауға болады:

• Берілген ашық мәтін txt кеңеймесі бар мәтіндік файлда сақталсын. Файл Вижинер әдісі алгоритмі бойынша шифрленеді. Нәтижеде шифрленген мәтіндік файл құрылады.

• Шифрленген мәтіндік файлды Вижинер әдісінің шифрден шығару алгоритмі бойынша дешифрленеді. Нәтижеде ашық мәтіндік файл құрылады.

Шифрлеу коды:

```
kilt := length (Edit1.Text);
  katar := Edit1.Text;
    assignfile (f01,'crypts\crypt1.txt');
      {ашық мәтін файлы}
       reset (f01);
       assignfile (f02,'crypts\uncrypt1.txt');
       {криптомәтін файлы}
     rewrite (f02);
    while not eof (f01) do
  begin
  for i:=1 to kilt do
    begin
      if eof (f01) then exit;
       read (f01, sym);
        crypt := chr ((ord (sym) + ord (katar [i])) mod 256);
        write (f02, crypt);
       end;
     end;
   closefile (f01);
closefile (f02);
```

Дешифрлеу коды:

```
kilt := length (Edit1.Text);
  katar := Edit1.Text;
     if kilt <1 then exit;
       assignfile (f01, ' crypts\crypt1.txt ');
        {криптомәтін файлы}
         reset (f01);
         assignfile (f02, ' crypts\uncrypt1.txt ');
         {ашық мәтін файлы}
       rewrite (f02);
     while not eof (f01) do
    begin
 for i:=1 to kilt do
begin
if eof (f01) then exit;
 read (f01, sym);
   crypt := chr ((ord (sym) – ord (katar [i])) mod 256);
    write (f02, crypt);
     end;
     end;
   closefile (f01);
closefile (f02);
```

Сонымен, симметриялық криптожүйе үшін бірнеше алгоритмдердің бағдарламасы құрылды. Әрі карай оларды бір криптожүйеде жинақтау қажет. Жалпы, криптожүйеде криптографиялық түрлендіру әдістері жасырылмайды. Мұнда шифрлеу және дешифрлеу үшін қажетті кілт негізгі «құпия» болып есептеледі. Ақпаратты тек қана осы кілт көмегінде шифрлеуге және оқуға болады. Құпия кілтті ақпаратты жіберуші және қабылдаушы тараптар алдын-ала келісіп алулары қажет.

10 КРИПТОЖҮЙЕ ҚҰРУДА ҚОЛДАНЫЛАТЫН БАҒДАРЛАМАЛАУ ТЕХНОЛОГИЯСЫНЫҢ ҚАСИЕТТЕРІ

Соңғы кезде, яғни XXI ғасырдың басында, барлық бағдарламалау жүйелері - объектке бағытталған бағдарламалау технологиясын қолдануда. Бүгінгі таңда бұл бағдарламалау технологиясы бағдарламалау саласы дамуының жоғарғы эволюциялық сатысы болып саналады. Объектке бағытталған бағдарламалау технологиясы - басқа бағдарламалау технологияларының, мысалы, процедуралық, құрылымдық, модульдік, функционалдық т.б. жетістіктерін өз ішіне ала отырып, бағдарламалау үрдісінде абстракттық құрылымдардан пайдалана алады. Сонымен қатар, бағдарламалаудың «төменнен жоғарыға», «визуалдық», «рекурсивтік» т.б. әдістерін өз ішіне алады.

Сонымен, біздің алдымызда мынадай сұрақ туындайды: симметриялық алгоритмді криптожүйені құруда объектке бағытталған бағдарламалау технологиясының қандай қасиеттерін критерий ретінде алу керек?

Жалпы синтаксистік критерийлер. Объектке бағытталған бағдарламалау технологиясындағы бағдарламалау тілінің синтаксисі дегеніміз бағдарламаның жазылу ережесіне сәйкес символдар тізбегі болып есептеледі. Мысалы, біз қарастырып отырған Borland Delphi ортасында әліпби әріптерін k орынға жылжытуды алайық (Юлий Цезарь шифры), яғни

```
crypt := chr ((ord (sym)+ kilt) mod 256);
```

өрнегі дұрыс құрылым болып есептеледі. Ал

```
crypt := chr (ord (sym)+ kilt) mod 256);
```

дұрыс емес құрылым болып есептеледі. Бағдарламалау ортасы синтаксисінің негізгі қызметі бағдарламашы мен бағдарламалау ортасы компиляторы арасында ақпарат алмасуға арналған белгілеулер жүйесін қамтамасыз ету болып табылады. Жалпы, нақты синтаксистік құрылымды таңдау қандайда бір криптографиялық түрлендіру әдісіне немесе ақпараттың ашық

мәтініне байланысты емес. Мысалы, қандайда бір айнымалының типін көптеген тәсілдермен беруге болады. Сол сияқты, бағдарламаның синтаксистік элементтерін құру барысында ақпаратты бағдарламалау ортасы компиляторына жеткізуде криптографиялық түрлендіруге тікелей қатысы жоқ мәселелерді ескереді. Мұндай мәселелер өте көп, бірақ оларды негізгі төрт топқа біріктіруге болады.

- оқу жеңілдігі;
- жазу жеңілдігі;
- трансляциялау жеңілдігі;
- біркелкілік;

Оқу жеңілдігі. Бағдарламадағы криптографиялық түрлендіру алгоритмінің құрылымы мен онда берілген деректер типі бағдарлама кодын визуалды қарау барысында түсінікті болса, онда бағдарлама оқуға жеңіл деп саналады.

Мысалы: Юлий Цезарь шифры бағдарламалаудағы шифрден шығару формуласын $D_k(i) = (i + n - k) \bmod n$ жазайық, яғни

```
crypt := chr ((ord (sym) - kilt) mod 256);
```

Бір қарағанда, өрнек түсініксіз сияқты. Бұл өрнекті құрылымдық бағдарламалау технологиясында былай жазған дұрыс сияқты, яғни

$$d[i] = (i + n - k) \bmod 256;$$

Бірақта объектке бағытталған бағдарламалау технологиясы тұрғысынан жоғарғы өрнек тиімді. Өрнекті жазуда дайын синтаксистік құрылымдардан, нақты айтсақ, қатарлық функциялардан пайдаланған.

Сол сияқты,

```
kilt := StrToInt (s);
```

өрнегінің де құрылымдық бағдарламалау технологиясы тұрғысынан қарағанда мағынасы жоқ сияқты.

Жалпы, бағдарламаны оқу жеңілдігіне операторлардың табиғи форматтары, құрылымдық операторлар, қызметі сөздерді кеңінен пайдалану, бағдарламалар кодына түсініктемелер жазу мүмкіндіктері, идентификатор ұзындығының шектелмегендігі, мнемоникалық амал белгілері, бағдарламаны еркін форматта жазу стилі сияқты бағдарламалау тілінің ішкі қасиеттері әсер етеді.

Жазу жеңілдігі. Бағдарлама кодын жазуды жеңілдететін критерийлер көбінесе оны оқу жеңілдігіне қайшы болып келеді. Оған негізгі себеп, бағдарламашыға жазу жеңілдігі қысқа және бір текті синтаксистік құрылымдарды пайдалану арқылы берілсе, бағдарламаны оқу жеңілдігін қамтамасыз ету үшін, керісінше, түрліше құрылымдарды пайдаланған тиімді.

Жалпы, Borland Delphi ортасы бағдарлама жазу үшін өте қолайлы орта болғанымен, онда жазылған бағдарлама кодын оқу қиынға соғады. Бағдарламадағы визуалды компоненттердің қасиеттерін үнсіз қабылдау, арифметикалық амалдардағы деректер типтерін сипаттамауға мүмкіндік беретін айқын емес синтаксистік келісімдер бағдарламаның жазылуын қысқартады, бірақ оқылуын қиындатады. Бірақ, бұл екі мәселенің де дұрыс шешілуіне септігін тигізетін синтаксистік құрылымдар кездеседі. Оларға қарапайым құрылымдық операторлар, регуляр өрнектерден пайдалану, амалдардың мнемоникалық символдарын және ұзындығы шектелмеген идентификаторларды жазу жатады. Егер қандайда бір өрнекті бірнеше тәсілмен жазу мүмкін болса, мұндай синтаксистік ереже жеткілікті деп есептеледі. Мұндай жеткіліктілік критерийі бағдарлама кодын оқуды жеңілдетіп, компиляциялау барысында қателерді тексеруге мүмкіндік береді. Бірақта оның кемшілігі ретінде бағдарлама мәтінін ұзартып, жазылуын қиындатуын айтуға болады.

Верификация жеңілдігі. Бағдарлама кодын оқу және жазу жеңілдігі оның туралығына тікелей байланысты. Бұл критерийді бағдарлама верификациясы деп атайды. Қандай да бір оператордың жазылуын түсіну үлкен қиындық тудырмайды, бірақ бағдарламаның туралығын тексеру үрдісі өте күрделі болады. Сондықтан жазылған бағдарламаның жинақтылығын

математикалық тұрғыдан қатаң дәлелдейтін әдістер қажеттігі туындайды.

Трансляциялау жеңілдігі. Бағдарламаны оқу және жазу жеңілдігі критерийлеріне қосымша, оларға қарама-қайшы мағыналы бағдарламаны орындалатын кодқа трансляциялау жеңілдігі критерийі болып табылады. Жоғарыда сипатталған бағдарламаны оқу және жазу жеңілдігі бағдарламашы талаптарын қанағаттандырса, транляциялау жеңілдігі жазылған бағдарлама кодын өндейтін транслятордың талаптарын қанағаттандырады. Трансляциялауды жеңілдетудің ең негізгі құралы – бағдарлама құрылымы синтаксисінің бірізділігі болып есептеледі. Осы бірізділік арқасында бағдарламаның синтаксистік құрылымы бірнеше қарапайым ережелермен жазылуы мүмкін. Сондықтан, бағдарламада арнайы синтаксистік құрылымдарды қолдану, оны трансляциялау жеңілдігіне әсер стеді. Жалпы, криптографиялық түрлендіруде, шифрлеу алгоритмінің жылдамдық қасиеті бойынша, трансляциялау жеңілдігі алгоритмнің орындалу жылдамдығына әсер етпеуі тиіс.

Бірмәнділік. Кез келген бағдарламалау технологиясында негізгі шешімін таппаған мәселелердің бірі, құрама құрылымдарды пайдаланғанда, бірмәнділіктің жоғалуы. Жалпы, бағдарламалау тілінің синтаксисі, ондағы кез-келген құрылымның бірмәнділігін қамтамасыз етуі тиіс. Біркелкі емес синтаксистік құрылым, оны әр түрлі түсінуге алып келді. Бұл мәселе бағдарламаның қарапайым құрылымдарында емес, оларды өзара байланыстырғанда пайда болады.

Әрі қарай, объектке бағытталған бағдарламалау технологиясының негізгі ұғымы - объектті қарастырайық. Объект деп компоненттері әртүрлі типті деректер және осы деректерді пайдаланатын процедура мен функциялар болып келетін, абстракт құрылым. Объекттегі деректер - өрістер, ал процедуралар мен функциялар - әдістер деп аталады.

Объекттің жалпы форматы төмендегідей.

```
объект_аты = object
өрістер;
әдістер;
end;
```

Бағдарламада объектті сипаттауда өрістің аты мен типі көрсетіледі. Ал әдісті сипаттауда тек қана аты мен типі ғана көрсетіледі. Ал әдістің сипаттамасы мен оның іс-әрекеті процедура немесе функция түрінде объекттен бөлек жазылады. Объекттің орісіндегі деректер, осы объекттің іс-әрекеттерін орындайтын процедуралар мен функцияларында айқын емес түрде қолданылады. Әдісті пайдалануда, оның атының алдында объект аты көрсетіледі. Оларды бөлектеу үшін арасына нүкте белгісі қойылады.

Объектті сипаттауда төмендегі талаптар орындалуы тиім:

- объект бағдарламаның тек қана type бөлімінде сипатталады;
- локалдық объекттер процедура мен функцияларда сипатталмайды;
- объект өрістері әдістерден алдынғы қатарда жазылады;
- объект компоненті ретінде файл типі алынбайды;
- объектте сипатталған өрістегі деректерді әдістерде қайта сипаттамай пайдаланады.

Объектте бағытталған бағдарламалау технологиясы біркелкі абстракт объекттерді сыныпқа жинақтайды. Сыныптар визуалды кітапханаларда сақталады. Сыныптағы деректер өріс деп, ал процедуралар мен функциялар - әдіс деп аталады. Borland Delphi ортасында үйлесімділік келісімі бойынша, өріс аттары f (field) әрпінен басталуы тиіс. Сыныптар бағдарламаның type типтерді сипаттау болімінде сипатталады. Сол сияқты, обьекттер бағдарламаның var айнымалыларды сипаттау бөлімінде сипатталады. Объекттің төменде көрсетілген негізгі үш қасиеті бар:

- инкапсуляция
- мұрагерлік
- полиморфизм

Инкапсуляция қасиеті бойынша деректер, бір құрылымда процедура және функциялармен бірге сипатталады. Нәтижеде деректердің жаңа типі - объект анықталады. Мысалы, crypt объектін сипаттайық:

```
crypt = object
fname: string [10];
procedure show;
end;
```

Borland Delphi ортасында әдісті бағдарламаның қай жерінде шақыру қажеттігі және қандай мәндерді қабылдауы тиіс екендігі алдын ала көрсетіледі. Компиляция кезінде бағдарлама әдісті статикалық түрде шақырады. Ал процедураны бағдарламаның орындалу барысында шақыру қажет болса, онда виртуал әдістерден пайдаланады. Объектте әрбір виртуал әдіс үшін өз алдына виртуал конструктор қажет. Конструктор объекттің барлық әдістерінен алдын жазылады. Конструкторды сипаттау үшін арнайы constructor қызметші сөзі қолданылады.

Кейде конструкторда тек қана begin және end қызметші сөздері жазылады. Мұндайда конструктордың не қажеті бар деген сұрақ туындайды. Компилятор объекттегі конструкторды көріп, алдын ала виртуал әдіске дайындала бастайды, яғни жедел жадыдан объектке динамикалық түрде орын бөледі.

Мұрагерлік қасиеті, объекттер иерархиясын құру мүмкіндігін береді. Бұл объекттер бірлесе отырып, иерархиялық ағаш, яғни сынпты құрайды. Ағаштың жоғарғы сатысындағы объектті - аналық, ал төмендегілерін - мұрагер объекттер деп атайды. Мұрагер объекттер аналық объекттің деректері мен әдістерін үндеместік келісімі бойынша айқын емес түрде пайдалана алады.

Мұрагер объектті сипаттаудың жалпы форматы:

```
var
Мұрагер_Объект = Object (Аналық_Объект)
Өрістер;
Әдістер;
end;
```

Полиморфизм қасиеті, объекттегі әдістердің өз іс-әрекетін өзгерте алу қабілеті. Мұрагер объект аналық объекттің әдісін пайдалана отырып, әдіске жаңа деректер мен локал әдістер енгізе алады.

Объекттегі статикалық әдістер сияқты, виртуал әдістер де мұрагерлік қасиетінен пайдаланады. Мұрагер объекттерде әдістер қайтадан сипатталуы да мүмкін. Аналық объекттен мұрагерлік қасиетімен алынған виртуал әдістерде де, полиморфизм қасиетін пайдалана алады. Тек мұнда аналық объектте анықталған әдістің мағынын мұрагер объектте өзгертуге болмайды.

11 ОБЪЕКТКЕ БАҒЫТТАЛҒАН БАҒДАРЛАМАЛАУ ОРТАСЫНДА КРИПТОЖҮЙЕ ҚҰРУ ТЕХНОЛОГИЯСЫ

Қазіргі таңда есептеу техникасының жылдам дамуы және объектке багытталған программалау технологиясына болған сұраныс RAD-технологиясы (Rapid Application Development - қосымшаларды жылдам құру ортасы) деп аталатын программалау жүйелерінің пайда болуына алып келді. Жалпы, аталған программалау жүйесі қолданыстағы программалау компоненттерінің бір ортаға интегралданған жүйесі болып есептеледі. Осы объектке бағытталған программалау ортасы программа кодының редакторы, компилятор, программа жүктеушісі, визуалды компоненттер және динамикалық кітапханаларды бір интерфейспен біріктіреді.

Программалау ортасында жоба түсінігі енгізілген. Жоба түрлі оқиғалар жиынынан (мысалы, тышқан манипуляторын экранда қозғалту және шерту) және осы оқиғаларға жауап ретінде визуалды компоненттердің іс-әрекетінен тұрады. Программалау ортасы динамикалық кітапханалардан реалды уақыт режімінде пайдаланады. Сонымен қатар басқа да коптеген қосымша жұмыстарды операциялық жүйемен бірге автоматтық түрде орындайды. Ал тұтынушыға басқа жұмыстарға көңіл бөлмей, тек жоба үстінде жұмыс істеуге мүмкіндік береді. Тұтынушыға жоба құруда сұхбаттық терезелер мен оқиғаларды өңдеу функциялары арқылы алгоритм құрастыру жұмысын ғана қалдырады.

RAD-технологиясы негізінде жұмыс істейтін программалау жүйесіне Borland Delphi объектке бағытталған ортасы да жатады. Borland Delphi ортасын Borland International корпорациясы жасап шығарған болатын. Қазіргі таңда ол программалау жүйелерінің ішінде ең көп таралғандарының бірі болып есептеліп жүр. Ал Borland Delphi объектке бағытталған ортасының программалау тілі ретінде Object Pascal қолданылады. Бұл тіл әдеттегі Pascal тілінің версиясы.

Pascal программалау тілін швейцариялық ғалым Никлаус Вирт 1970 жылы ойлап тапқан болатын. Ал Pascal деп француз ғалымы Блез Паскальдің құрметіне атаған. Алғашқыда Pascal тілі колледж студенттеріне алгоритм негіздерін үйретуге арналған болатын. Дегенмен тіл алгоритм құруға қарапайым, қолдануға ыңғайлы,

талдауға түсінікті, үйренуге жеңіл болып шықты. Сондықтан тез арада стандарт программалау тілі ретінде бүкіл дүние жүзіне тарап кетті. Тілдің әр түрлі платформалы компьютерлерге көптеген стандарттары мен версиялары бар. Осы тілдің Ларри Теслер жасаған Object Pascal версиясы Borland Delphi ортасының программалау тілі болып қалыптасты.

Borland Delphi прогрaммалау ортасы алғашында 1995 жылы MicroSoft Windows операциялық жүйесі үшін жасап шығарылды. Келесі жылдары қайта өңделген Borland Delphi 2.0, 3.0, 4.0, 5.0, 6.0, 7.0 сияқты версиялары жарық көрді. Мысалы, Borland International корпорациясы Borland Delphi 5.0 версиясын - 1999 жылы, Borland Delphi 6.0 версиясын 2001 жылы, Borland Delphi 7.0 версиясын - 2002 жылы жасап шығарды.

Borland Delphi програмалау ортасының алғашқы версиялары 32 разрядты MicroSoft Windows операциялық жүйесі негізінде жұмыс істеуге арналған болатын. Дегенмен, оның соңғы версиялары, атап айтқанда Borland Delphi 6.0 версиясынан бастап, Unix операциялық жүйесі негізінде де жұмыс істейді. Оларды Kylix программалау ортасы деп атайды. Соңғы кезде Borland Delphi 2005, 2006, 2008, 2010 версиялары да жасап шығарылды.

Соңғы жылдары Borland Delphi программалау ортасын құрумен CodeGear компаниясы айналысады. Бұл компания 1999 жылдан бастап Embarcadero корпорациясы құрамына енген. Сондықтан 2010 жылдан бастап Borland Delphi программалау ортасы RAD Studio пакеті құрамында шығып отыр. Embarcadero корпорациясы Borland Delphi программалау ортасына көптеген жаңалықтар енгізіп жатыр. Соның бірі, программалау тілі ретінде Object Pascal тілімен қатар, C#, Java және PHP тілдерін де қолданылады. Ресейде Borland Delphi программалау ортасын қолдаушылардың ресми Web-сайты бар.

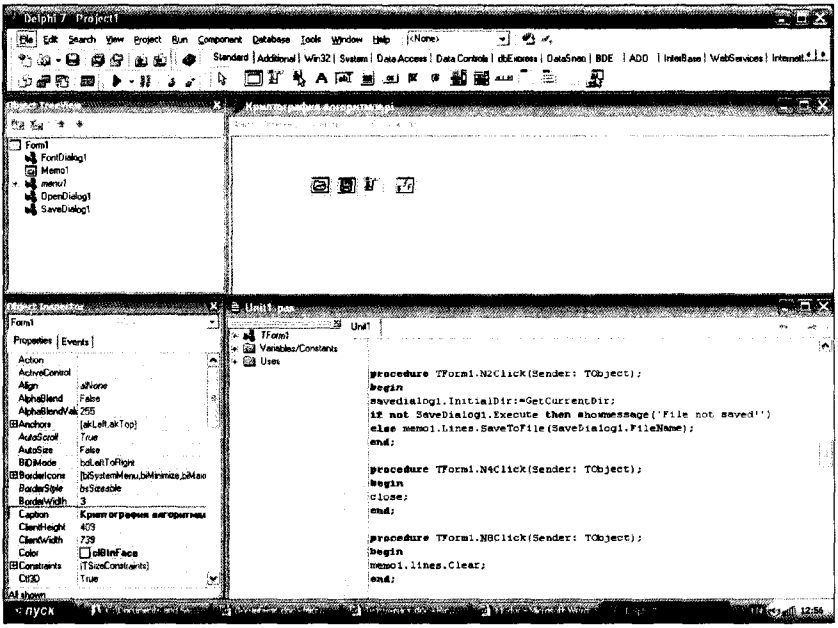
Әрі қарай, осы айтылғандарды пайдалана отырып, Borland Delphi ортасында криптожүйе құру технологиясын қарастырайық. MS Windows амалдық жүйесінде Borland Delphi ортасын іске қосу үшін Пуск (Іске қосу) – Все программы (Барлық бағдарламалар) - Borland Delphi 7 → Delphi 7 нұсқаулар тізбегін орындаймыз. Монитор экранында Borland Delphi ортасының негізгі терезесі көрінеді. Delphi ортасы графикалық интерфейсті, көптерезелік болып есептеледі. Сондықтан тұтынушы баптауына байланысты,

оның жүктелгеннен кейін көрінісі әр түрлі болуы мүмкін. Интерфейстің құрамына негізгі 4 терезе кіреді (сурет 6):

- Неғізгі терезе (Project 1);
- Объект бақылаушысы (Object Inspector);
- Қалып құрастырушысы терезесі (Form1);
- Бағдарлама кодының терезесі (Unit1.pas).

Borland Delphi ортасында құрылатын бағдарламаны жоба (project) деп атайды. Жобаны құруда пайдаланатын іс-әрекеттерге қысқаша тоқталайық.

Негізгі терезе жобаны құрудағы тұтынушының іс-әрекеттерін басқарады. Терезе Delphi ортасы іске қосылып тұрған кезде, міндетті түрде монитор экранының жоғарғы қатарына орналасады. Терезеде Delphi ортасының тақырыбы, негізгі мәзір жүйесі, пиктограммалық нұсқау батырмалары мен компоненттер палитрасы орналасады (сурет 6).

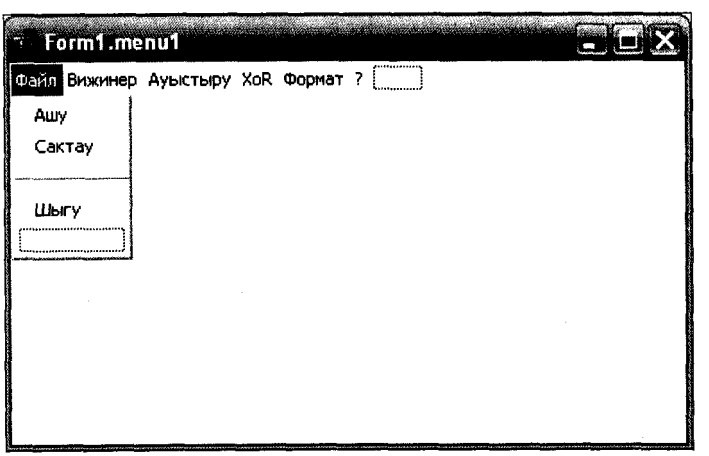


Сурет 6 - Borland Delphi ортасының негізгі терезесі

Қалып құрастырушысы немесе қалып терезесі, құрылатын Windows-қосымшаның терезесі болып саналады. Бір жобада бірнеше қалып терезесін байланыстырып пайдалануға болады. Бастапқыда қалып терезесінде тек қана Windows-қосымшаның тақырыбы мен интерфейстік басқару батырмалары, яғни үлкейту, кішірейту, жабу батырмалары ғана болады. Терезенің жұмыс аймағы жұмысқа ыңғайлы болу үшін, координаталық торлы нүктелермен реттелген болады.

Жобаны құруды осы қалып терезесін қажетті визуалды компоненттермен толтырудан бастаймыз. Бұл ерекшелік визуалды бағдарламалау технологиясының негізі болып табылады. Визуалды компоненттер негізгі терезедегі компоненттер палитрасында орналасқан. Delphi ортасының құрамына 200-дей визуалды компоненттер кіреді. Олар бірнеше стандарт топтарға бөлінеді. Сонымен қатар жаңа компоненттерді құру немесе енгізу мүмкіндігі қаралған.

Жобаның мәзірін құру үшін Standard тобына жататын MainMenu компонентін қалып терезесіне тышқан манипуляторы арқылы орналастырамыз. Бұл компонент TMainMenu сыныбына тиісті. Терезеде пайда болған MainMenu компонентін тышқан манипуляторы арқылы бір рет шертсек Form1.menu1 терезесі пайда болады (сурет 7).

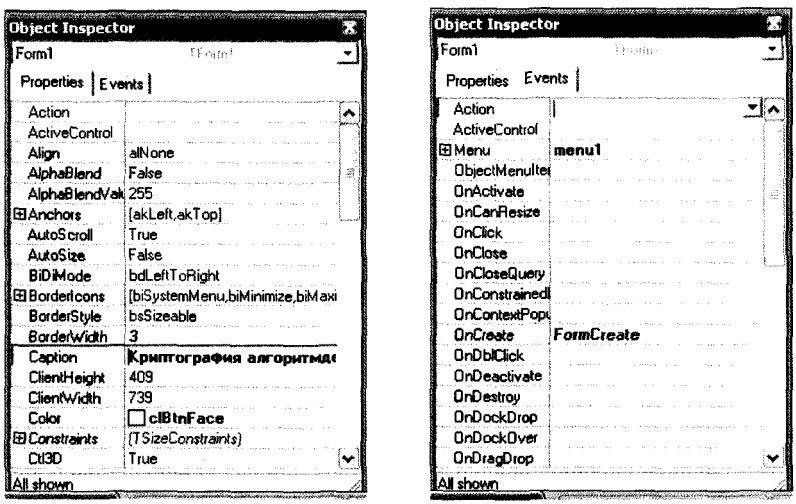


Сурет 7 - Form1.menu1 терезесі

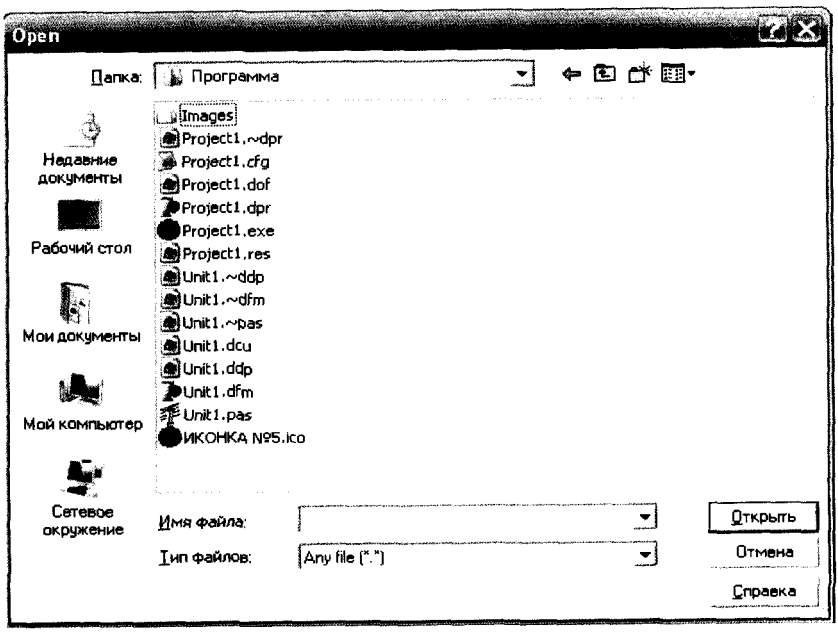
Терезедегі қажетті өрістерді Объект бақылаушысы (Object Inspector) көмегінде толтырамыз. Объект бақылаушысынан компоненттің қасиеттерін орнатуда пайдаланады. Оның екі жапсырмасы бар: Properties және Events жапсырмалары (сурет 8).

Properties жапсырмасында екі баған бар. Сол жағындағы бағанда компонент қасиеттерінің тізімі орналасқан. Оң жақтағы баған арқылы сол қасиеттердің мәнін өзгертуге болады. Events жапсырмасының сол жағындағы бағанда компонент оқиғасын өңдеуші қасиеттер тізімі орналасқан. Оң жағында оқиғаны өңдеуші қасиеттердің сәйкес мәндері жазылады. Осы жерге өрістердің мәнін жазамыз, мысалы, Properties жапсырмасының caption қасиетіне оның мәнін «Криптография алгоритмдері» деп жазамыз. Сол сияқты Events жапсырмасына оны өңдейтін оқиғаның мәнін жазамыз, мысалы, OnClick оқиғасының мәні N121Click. Бұл оқиға жоба мәзірінің «шифрлеу/дешифрлеу» өрісін басқан кезде, сәйкес процедураны шақырады.

Жобада шифрленетін ашық мәтін орналасқан файлды шақыру үшін Dialogs тобына жататын OpenDialogs1 компонентін қалып терезесіне орналастырамыз. Бұл компонент TOpenDialogs сыныбына тиісті. Терезеде пайда болған OpenDialogs1 компонентін тышқан манипуляторы арқылы бір рет шертсек «Ореп» терезесі пайда болады (сурет 9).



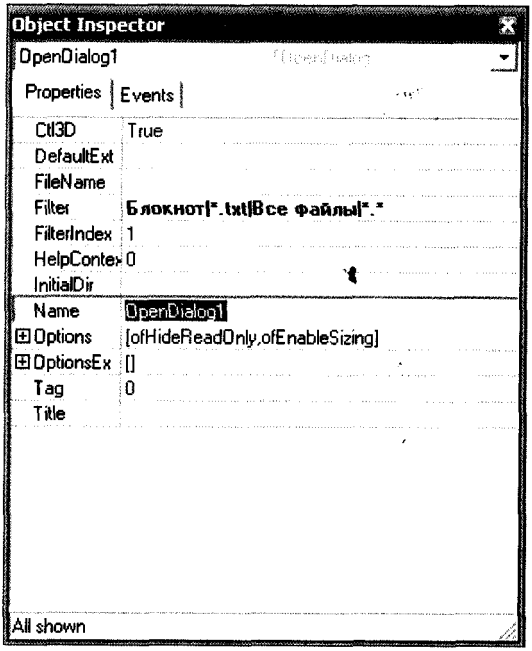
Сурет 8 - Object Inspector терезесі



Сурет 9 - TOpenDialogs сыныбының «Open» терезесі

Терезедегі қажетті өрістерді Объект бақылаушысы көмегінде толтырамыз. Осы жерге өрістердің мәнін жазамыз, мысалы, Properties жапсырмасының Filter қасиетіне оның мәнін «Блокнот|*.txt|Все файлы|*.*» жазамыз (сурет 10).

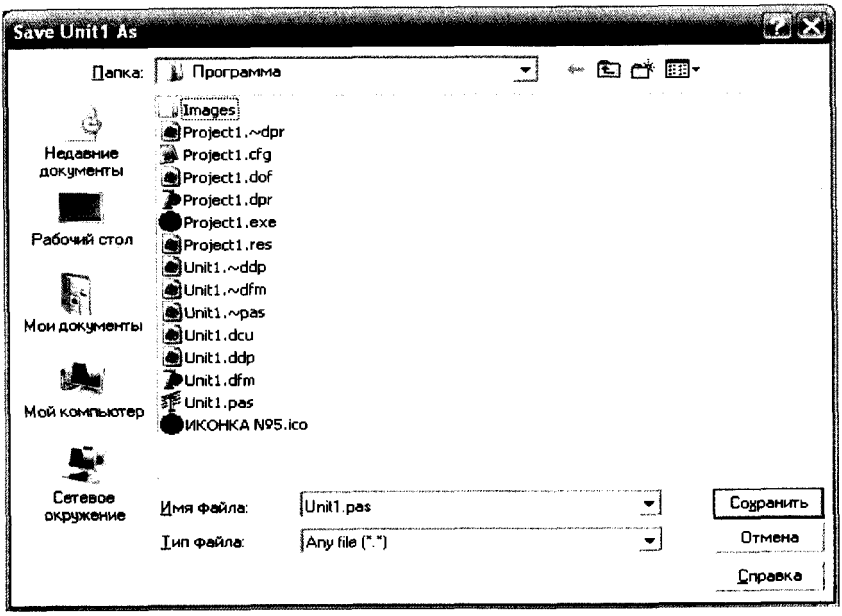
Жобада шифрленген криптомәтінді файлға сақтау үшін Dialogs тобына жататын SaveDialogs1 компонентін қалып терезесіне жоғарыда көрстілген сияқты орналастырамыз. Бұл компонент TSaveDialogs сыныбына тиісті.



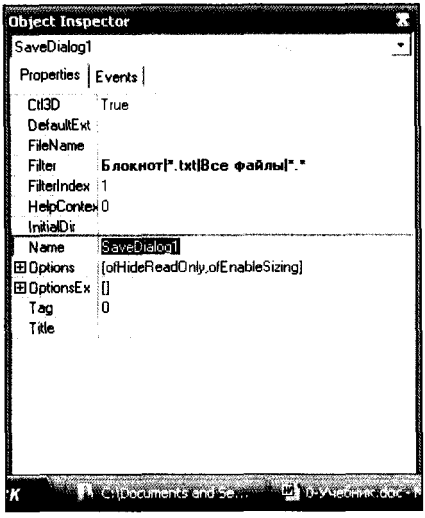
Сурет 10 - OpenDialogs1 компонентінің Object Inspector терезесі

Терезеде пайда болған SaveDialogs1 компонентін тышқан арқылы бір рет шертсек «Save» терезесі пайда болады (сурет 11).

Терезедегі қажетті өрістерді Объект бақылаушысы көмегінде толтырамыз. Осы жерге өрістердің мәнін жазамыз, мысалы, Properties жапсырмасының Filter қасиетіне оның мәнін «Блокнот|*.txt|Все файлы|*.*» жазамыз (сурет 12).

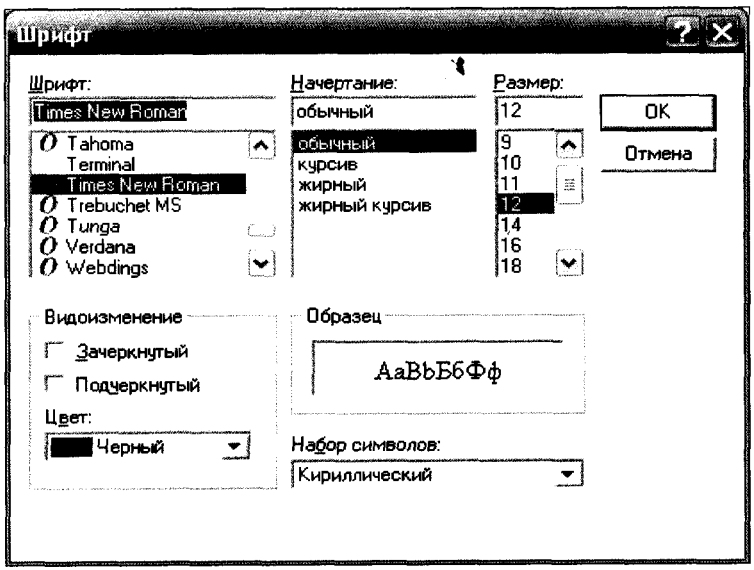


Сурет 11 - TSaveDialogs сыныбының «Save» терезесі



Сурет 12 - SaveDialogs1 компонентінің Object Inspector терезесі

Жобада мәтіннің шрифттерін (әліпбиін) өзгерту үшін Dialogs тобына жататын FontDialogs1 компонентін тышқан манипуляторы арқылы қалып терезесіне орналастырамыз. Бұл компонент TFontDialogs сыныбына тиісті. Терезеде пайда болған FontDialogs1 компонентін тышқан арқылы бір рет шертсек «Font» терезесі пайда болады (сурет 13).



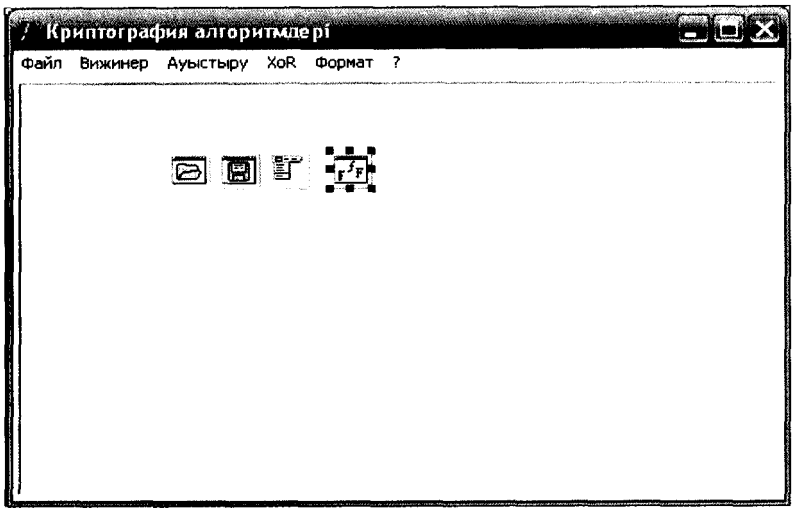
Сурет 13 - TFontDialogs сыныбының «Font» терезесі

Жобаның қалып терезесіне MainMenu, OpenDialogs1, SaveDialogs1 және FontDialogs1 компоненттерін орналастырғасын, негізгі қалыпқа «Криптография алгоритмдері» деп атау береміз және қалып өлшемдерін анықтаймыз (сурет 14). Атауды және қалып өлшемдерін беруде қажетті қасиеттерді Объект бақылаушысы көмегінде жоғарыда көрсетілген сияқты толтырамыз.

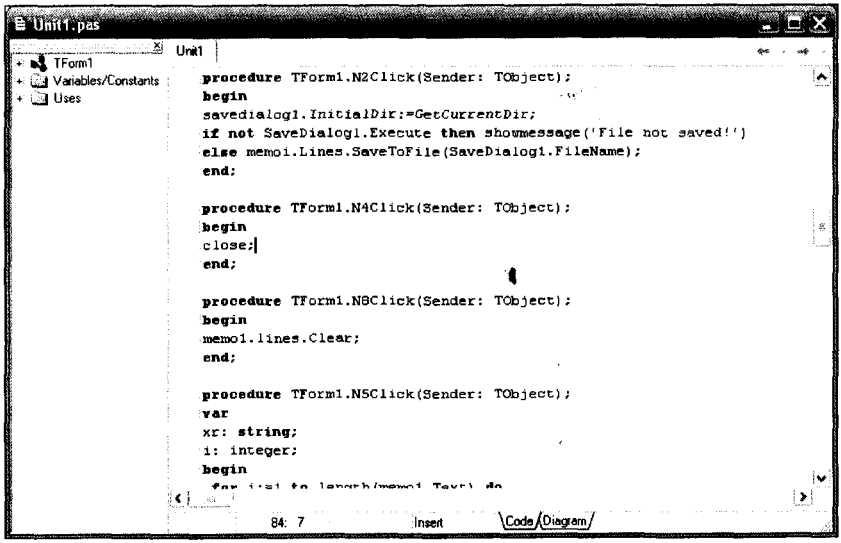
Жобаның бағдарламасы Бағдарлама кодының терезесінде (Unit1.pas) орналасады (сурет 15).

Терезеде енгізілген компоненттердің бағдарлама мәтіні пайда болады. Мысалы, жоғарыда көрсетілген FontDialogs1 компонентінің коды төмендегіше жазылған:

```
procedure TForm1.N10Click (Sender: TObject);
begin
fontdialog1.Font.Color:=memo1.Font.Color; {шрифт түсі}
fontdialog1.Font.Style:=memo1.Font.Style; {шрифт стилі}
fontdialog1.Font.size:=memo1.Font.Size; {шрифт өлшемі}
fontdialog1.Font.Charset:=memo1.Font.Charset; {шрифт символы}
fontdialog1.Font.Name:=memo1.Font.Name; {шрифт атауы}
if fontdialog1.Execute then
begin
memo1.Font.Color:=fontdialog1.Font.Color; {шрифт түсі}
memo1.Font.Style:=fontdialog1.Font.Style; {шрифт стилі}
memo1.Font.Size:=fontdialog1.Font.size; {шрифт өлшемі}
memo1.Font.Charset:=fontdialog1.Font.Charset; {шрифт символы}
memo1.Font.Name:=fontdialog1.Font.Name; {шрифт атауы}
end
end;
```



Сурет 14 – «Криптография алгоритмдері» қалып терезесі



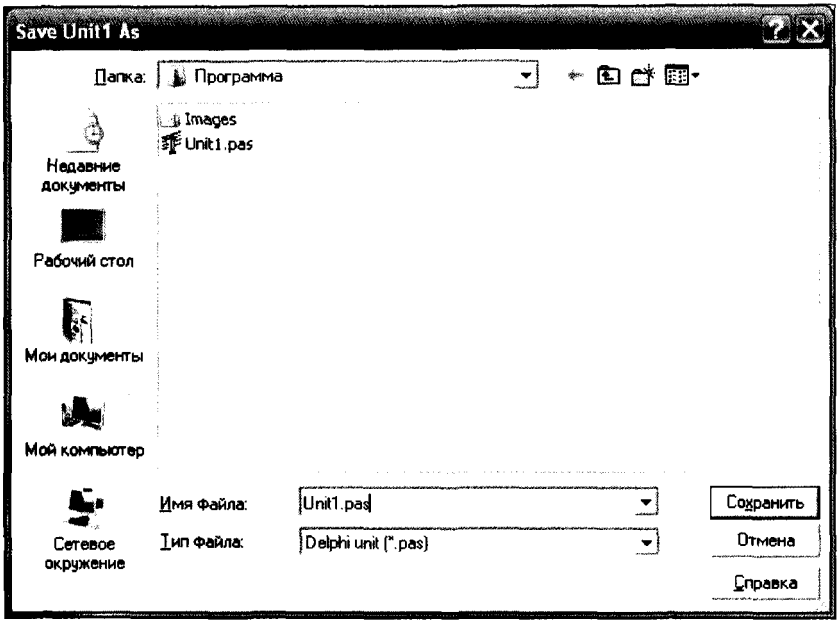
Сурет 15 – Бағдарлама коды терезесі (Unit1.pas)

Жалпы, қалыпта орнатылған визуалды компоненттің сипаттамалары dfm кеңеймесі бар файлда да сақталған. Егер қалып немесе визуалды компоненттердің қасиеттерін, өлшемін өзгертсек, онда осы өзгерістер аталған файлда автоматты түрде сақталады.

Бағдарламаға жоғарыда берілген криптографиялық алгоритмдерді сәйкес түрде процедураларға енгіземіз. Сонымен, жобаны құрып болғасын, оны сақтап қою қажеттігі туындайды. Жобаны сақтаудың бірнеше тәсілі бар. Әдетте жобаға арналған жеке буманы Windows жүйесінде құрады. Мысалы, жоба сақталатын «Криптография» бумасын жұмыс столында алдын ала құрамыз.

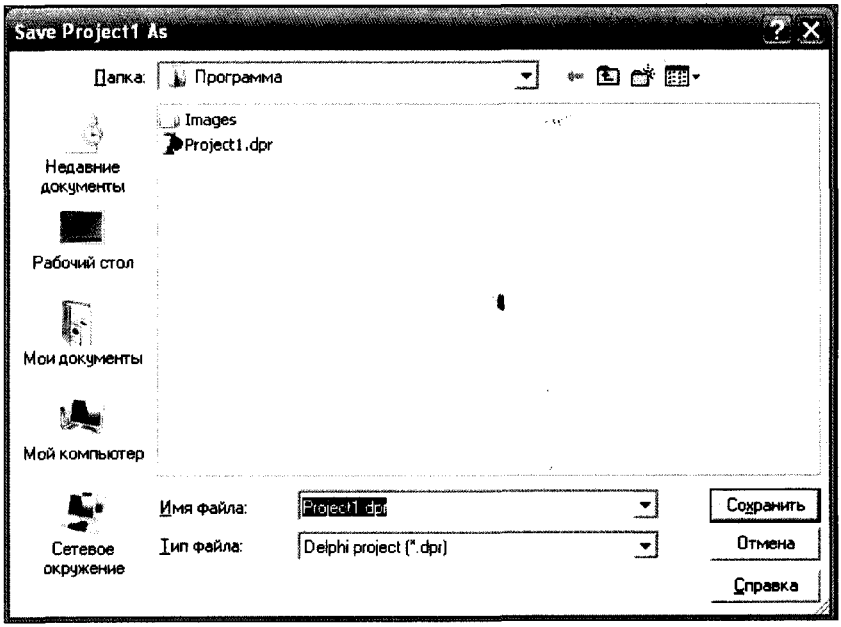
Әрі қарай бағдарламаны (модульді) сақтау үшін Delphi ортасында File – Save As нұсқауын береміз. Монитор экранына Save Unit1 As терезесі шығады. Бұл терезе әдетте жоба модулін алғаш рет сақтау үшін қажет. Терезенің файл атауы енгізілетін өрісінде автоматты түрде Unit1.pas аты көрініп тұрады. Осы атауды өшіріп, файлдың жаңа атын енгізіп (өзгертпесе де болады), «Сохранить» пернесін басу керек (сурет 16).

Жоба файлын сақтау үшін File – Save Project As нұсқауын береміз. Монитор экранына Save Project1 As терезесі шығады. Жобаны сақтау жоғарыдағы модульді сақтау сияқты орындалады (сурет 17). Модуль файлына автоматты түрде pas кеңеймесі, ал жоба файлына dpr файл кеңеймесі беріледі.



Сурет 16 – Save Unit As терезесі

Жалпы, жұмыс барысында File - Save All нұсқауын пайдалану тиімді. Бұл нұсқау жоба құрамындағы барлық файлдардың қажетті бумаға толық сақталуын қамтамасыз етеді.



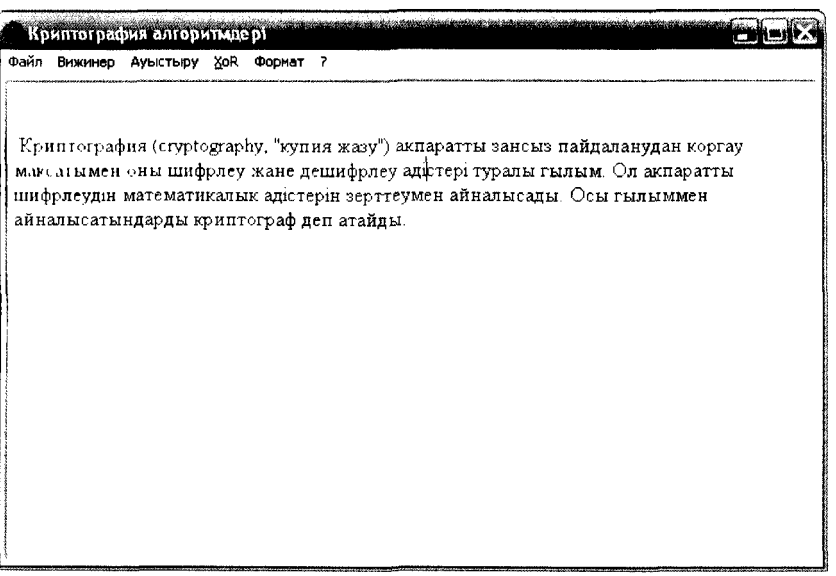
Сурет 17 – Save Project1 As сұхбаттық терезесі

Әрі қарай жобаны компиляциялау үшін Run нұсқауын орындаймыз. Егер бағдарламада синтаксистік қателер кездеспесе, монитор экранында жоба терезесі пайда болады (сурет 18).

1-модільдің бағдарлама коды:

```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
Dialogs, StdCtrls, Menus;
type
TForm1 = class(TForm)
Memo1: TMemo;
menu1: TMainMenu;
File1: TMenuItem;
Edit1: TMenuItem;
```

N1: TMenuItem;
N2: TMenuItem;
N3: TMenuItem;
N4: TMenuItem;
N5: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;
OpenDialog1: TOpenDialog;
SaveDialog1: TSaveDialog;
FontDialog1: TFontDialog;
N6: TMenuItem;
N9: TMenuItem;
N10: TMenuItem;
XoR1: TMenuItem;



Сурет 18 – Жоба терезесі

```
N11: TMenuItem;
N21: TMenuItem;
N12: TMenuItem;
N121: TMenuItem;
n13: TMenuItem;
N122: TMenuItem;
N1241: TMenuItem;
N14: TMenuItem;
procedure N1Click(Sender: TObject);
procedure N2Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure N8Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure N9Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure XoR1Click(Sender: TObject);
procedure N121Click(Sender: TObject);
procedure N122Click(Sender: TObject);
procedure N12451Click(Sender: TObject);
procedure N21641Click(Sender: TObject);
procedure N1241Click(Sender: TObject);
procedure n13Click(Sender: TObject);
procedure N14Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;
xr_num: string;
implementation
{$R *.dfm}
procedure TForm1.N1Click(Sender: TObject);
begin
openDialog1.InitialDir:=GetCurrentDir;
if not openDialog1.Execute then showmessage('File not selected!')
```

```
else memo1.Lines.LoadFromFile(openDialog1.FileName);
end;
procedure TForm1.N2Click(Sender: TObject);
begin
savedialog1.InitialDir:=GetCurrentDir;
if not SaveDialog1.Execute then showmessage('File not saved!')
else memo1.Lines.SaveToFile(SaveDialog1.FileName);
end;
procedure TForm1.N4Click(Sender: TObject);
begin
close;
end;
procedure TForm1.N8Click(Sender: TObject);
begin
memo1.lines.Clear;
end;
procedure TForm1.N5Click(Sender: TObject);
var
xr: string;
i: integer;
begin
for i:=1 to length(memo1.Text) do
begin
xr:=xr+chr(ord(memo1.Text[i]) xor strtoint(xr_num));
end;
memo1.Text:=xr;
end;
procedure TForm1.N9Click(Sender: TObject);
begin
if n9.Checked then
begin
n9.Checked:=false;
memo1.scrollbars:=ssboth;
end
else begin
n9.Checked:=true;
memo1.ScrollBars:=ssVertical;
end;
```

```
end;
procedure TForm1.N10Click(Sender: TObject);
begin
fontdialog1.Font.Color:=memo1.Font.Color;
fontdialog1.Font.Style:=memo1.Font.Style;
fontdialog1.Font.size:=memo1.Font.Size;
fontdialog1.Font.Charset:=memo1.Font.Charset;
fontdialog1.Font.Name:=memo1.Font.Name;
if fontdialog1.Execute then begin
memo1.Font.Color:=fontdialog1.Font.Color;
memo1.Font.Style:=fontdialog1.Font.Style;
memo1.Font.Size:=fontdialog1.Font.size;
memo1.Font.Charset:=fontdialog1.Font.Charset;
memo1.Font.Name:=fontdialog1.Font.Name;
end
end;
procedure TForm1.FormCreate(Sender: TObject);
var
Str:String;
i:Integer;
begin
xr_num:='2';
if (ParamCount > 0) then
begin
Str:=ParamStr(1);
for i:=2 to ParamCount do Str:=Str+' '+ParamStr(i);
end;
if str<>'' then memo1.Lines.LoadFromFile(str);
end;
procedure TForm1.XoR1Click(Sender: TObject);
begin
if not InputQuery('X0R', 'xor санын енгіз:',xr_num) then exit;
end;
procedure TForm1.N121Click(Sender: TObject);
var
xr: string;
i: integer;
begin
```

```
xr_num:='29';
for i:=1 to length(memo1.Text) do
begin
xr:=xr+chr(ord(memo1.Text[i]) xor strtoint(xr_num));
end;
memo1.Text:=xr;
end;
procedure TForm1.N122Click(Sender: TObject);
var
xr: string;
i: integer;
begin
xr_num:='49';
for i:=1 to length(memo1.Text) do
begin
xr:=xr+chr(ord(memo1.Text[i]) xor strtoint(xr_num));
end;
memo1.Text:=xr;
end;
procedure TForm1.N12451Click(Sender: TObject);
var
xr: string;
i: integer;
begin
xr_num:='69';
for i:=1 to length(memo1.Text) do
begin
xr:=xr+chr(ord(memo1.Text[i]) xor strtoint(xr_num));
end;
memo1.Text:=xr;
end;
procedure TForm1.N21641Click(Sender: TObject);
begin
memo1.lines.Clear;
end;
procedure TForm1.N1241Click(Sender: TObject);
begin
memo1.lines.Clear;
```

```
end;
procedure TForm1.n13Click(Sender: TObject);
begin
memo1.lines.Clear;
end;
procedure TForm1.N14Click(Sender: TObject);
begin
ShowMessage('М.Ауезов атындағы ОКМУ');
end;
end.
```

Сонымен, Borland Delphi объектке бағыттылған бағдарламалау ортасында криптожүйе құру технологиясын қарастырдық. Криптожүйенің интерфейсі Windows-қосымшаларға қойылатын талаптарға сәйкес. Криптожүйенің интерфейсі қарапайым, платформасы Windows, жедел жадыда алатын көлемі 414 Kb.

Жалпы, құрылған криптожүйе осындай симметриялық криптожүйелерге қойылатын талаптарға жауап береді. Жобадағы криптографиялық алгоритмдердің орындалу жылдамдығы мен криптобериктілігі алгоритмдердің приоритетіне байланысты. Қолдану облысы, ақпаратты сактау кезінде қорғауға арналған, бірақта ақпаратты тасымалдау кезінде қорғауда да пайдалануға болады.

ЖЕКЕ ЖҰМЫСҚА ТАПСЫРМАЛАР

Тапсырма №1. Ақпаратты қорғау және оның мәселелері

Төменде берілген тақырыптарға сипаттама беріңіз. Тапсырманы орындауда график, кесте, диаграмма т.с.с. пайдалануға болады. Тапсырма реферат түрінде қабылданады (вариант бойынша).

1. Ақпаратты қорғау және оның мәселелері
2. Ақпарат қауіпсіздігі
3. Ақпаратты қорғау әдістері
4. Криптография
5. Компьютерлік вирустар
6. Антивирустер
7. Архиваторлар
8. Техникалық құралдар
9. Бағдарламалық құралдар
10. Ұжымдық құралдар

Тапсырма №2. Брандмауэрлер опциялары

Төменде берілген тақырыптарға сипаттама беріп, талдап, мысалдармен түсіндіріңіз. Тапсырманы орындауда график, кесте, диаграмма т.с.с. пайдалануға болады. Тапсырма презентация түрінде қабылданады (вариант бойынша).

1. Firewalls сервері (брандмауэр) туралы жалпы мәліметтер
2. Proxy-servers туралы жалпы мәліметтер
3. Брандмауэрдің желілік компоненттері
4. Windows брандмауэрін қосу және өшіру
5. Windows брандмауэрінің қауіпсіздік журналын басқару
6. Windows брандмауэрінің файлдарға тікелей байланыс опциясы
7. Windows брандмауэрінің порттарын басқару
8. Windows брандмауэрінің үндеместен жұмыс істеу параметрлері
9. Windows брандмауэрінің саясатын басқару
10. UpnP технологиясы

Тапсырма №3. Криптографиялық шифрлеу әдістері

Берілген мәтінді (магистранттың аты-жөні, адресі) төменде көрсетілген криптографиялық шифрлеу әдісімен шифрланыз (вариант бойынша). Кілт ретінде магистранттың аты алынады:

1. Юлий Цезарь шифры
2. Қарапайым орын ауыстыру шифры
3. Сцитала шифры
4. Полибий квадраты
5. Түрме шифры
6. Сиқырлы квадраттар
7. Тритемий шифры
8. Белазо шифры
9. Порта шифры
10. Кардано торлары

Тапсырма №4. Криптографиялық шифрлеу әдістерін бағдарламалау

Turbo Pascal ортасында көрсетілген криптографиялық шифрлеу әдісіне бағдарлама құрыңыз (3-тапсырмаға қараңыз). Криптомәтін ретінде магистранттың аты-жөні, адресі алынады.

Тапсырма №5. Криптожүйе құру

Borland Delphi ортасында симметриялық алгоритмді криптожүйе құрыңыз. Криптожүйеде бірнеше криптографиялық әдістерден пайдаланыңыз (мысалы, орын алмастырулар шифрлары, Вижинер шифрлау жүйесі, RSA шифрлау жүйесі, DES шифрлау жүйесі, XoR шифрлау жүйесі).

ТЕСТ СҰРАҚТАРЫ

1. Ашық мәтіннің бөгде адамдарға мағынасын түсінбеу үшін жасалынатын өзгерту үрдісі қалай аталады?

a) шифрлеу
b) криптография
c) дешифрлеу
d) кері шифрлау
e) криптология

2. Шифрмәтінді ашық текстке өзгерту үрдісі?

a) реттеу
b) решифрлеу
c) шифрлеу
d) цифрлеу
e) дешифрлеу

3. Шығысында C болу үшін E шифр функциясының кірісіне P берілсін. Шифрдың математикалық үлгісі қалай жазылады?

a) E(P)=C
b) E(P)=P
c) P(P)=C
d) E(P)= E(P)
e) C(P)=E

4. *Ешқандай да кілтсіз криптомәтінді ашудың мүмкіндіктерін* кандай ғылым зерттейді?

a) криптоберіктілік
b) криптология
c) криптоталдау
d) криптофункция
e) криптография

5. ASCII стандартты кодына кіретін символдар қай алфавиттің құрамына кіреді?

a) бинарлық алфавиттің
b) алфавит z33-тің
c) сегіздік және он алтылық алфавиттің

d) алфавит z256-ның
e) алфавит z42-нің

6. Ашықмәтіннің әр әрпінің шифрмәтіннің сол символына алмастырылуы алмастырудың қай түріне жатады?
a) омофонды алмастыру
b) бір алфавитті алмастыру
c) блоктап алмастыру
d) көп алфавитті алмастыру
e) полиалфавитті алмастыру

7. Симметриялық криптожүйеде шифрлеу үшін қанша кілт қолданады?
a) 0
b) 1
c) 2
d) 3
e) кілтсіз шифрланады

8. Шифрдің кілтсіз дешифрлеу әдісіне шыдамдылығын анықтайтын қасиеті?
a) электронды жазба
b) криптоберіктілік
c) криптожүйе
d) жылдамдық
e) есептеу қателігі

9. Ашық мәтінді кілт көмегінде шифрлеу үшін кейбір кездейсоқ тізбектерден пайдалану әдісі?
a) алмастыру
b) блоктық шифрлау
c) гаммалау
d) ауыстыру
e) көп алфавитті қойылым

10. АҚШ және Ресей цифрлеу стандарттары шифрлеу әдістерінің қай класына негізделген?
a) блоктық шифрлар

b) алмастыру әдісі
c) көп алфавиттік қойылым
d) гаммалау
e) орын ауысу әдісі

11. Криптографияның қай бөлімі мәтінді қабылдаған өзге пайдаланушы арқылы ақпараттың авторлығын және түп-нұсқасын тексеруге мүмкіндік береді?
a) ашық кілтті криптожүйе
b) электронды жазба жүйесі
c) кілттерді басқару
d) симметриялық криптожүйе
e) жабық кілтті криптожүйе

12. Криптографиялық алгоритмдер - ....
a) ақпараттарды өңдеуге арналған құралдар
b) шифрлеуге қажетті математикалық амалдар
c) ақпаратқа физикалық тосқауылдар қою әдістері
d) шифрлеу әдістері немесе алгоритімдік шифрлеу
e) дешифрлеу әдістері

13. Құпия жазу қай ұғымға тән ?
a) түрлендіру
b) криптография
c) кодтау
d) криптология
e) криптоанализ

14. Эллипстік қисықтар - ...
a) математикалық құрылымдар
b) өңдеуді қамтамассыз ететін құрылымдар
c) пайдаланушыларға жадыны сақтайтын құрылымдар
d) физикалық құрылымдар
e) бағдарламалық құрылымдар

15. Моно алфавиттік орын басудың жалпы формуласы?
a) yi=k1*xi+k2(mod N)
b) kj=(j mod r)

c) p:ta^p(t)
d) p:=ta^p1(p2t)
e) SYM(m!)

16. K=(K0,K1,...,KN) тізбегінде қайталынатын шексіз тізбекті қай кілтке жатқызамыз?
a) колданушы кілт
b) ашық кілт
c) жабық кілт
d) бөлінбейтін кілт
e) блокты кілт

17. Кілттің бірі ашық болса екіншісі қандай болуы тиіс?
a) курделі
b) шифрлік
c) дешифрлік
d) жабық
e) гаммалық

18. Ашық кілтпен шифрлеу алгоритімдерінің кең қолданыс тапқан жүйесін көрсет?
a) RSA криптожүйесі
b) SYM криптожүйесі
c) RAC криптожүйесі
d) моно криптожүйесі
e) Виженер криптожүйесі

19. Рон Ривест, Ади Шамир, Леонард Эйдельман қай жүйенің қалаушылары ?
a) MONO
b) RSA
c) SYM
d) RAC
e) DOC

20. Орын алмастыру әдісін қолданып "бағдарлама" сөзін 6-3-1-4-5-2 кілті аркылы шифрленген түрін көрсет
a) ргбдаа_мла_а

b) ргбдаа_мла_т
c) ргбдаа_кма_п
d) ргбдаа_мка_а
e) ргбдаа_мба_б

21. Орын ауыстыру әдісінде "компьютер" ашық мәтін к1=3-1-2, к2=4-3-2-1 кілттерімен шифрлегенда қандай шифрмәтін шығады?
a) ою_кьрпе_мт_
b) _пе_мт_ою_ркь
c) ркь_омп_юте
d) ьрк__ою_мт_пе
e) юте_ркь_омп_юте

22. Орын алмастыру әдісінде "ақпарат" ашық мәтінін к1=1-2, к2=2-4-1-3 кілттерімен шифрлегенде қандай шифрмәтін шығады?
a) а_пт_кра
b) арпт_ака
c) ак_арат
d) каа_ ар_пт
e) ак_атар

23. Орын алмастыру әдісінде "студент" ашық мәтіннің қандай кілтпен шифрлағанда "_тнедуст" шифрмәтінде шығады?
a) 8-7-6-5-4-3-2-4
b) 5-3-1-7-8-6-4-2
c) 5-3-7-6-4-2-1-8
d) 1-5-3-1-7-6-4-2
e) 4-2-8-7-1-6-5-3

24. Ашық мәтіннің келесі символы жазылатын блок, келесі формула арқылы анықталады:
a) k=(Ri-1)n+Sj
b) Yi=k1Xi+k2(mod n)
c) Kj=(j mod r)
d) J:ta^p(t)
e) p:=ta^p1(p2t)

25. Морзе алфавиті, алмастыру және орын басу әдістері қай кодтауға жатады?

a) шифрлік
b) символдық
c) ақпараттық
d) мәтіндік
e) логикалық

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР

1. Бабаш А.В., Шанкин Г.П. Криптография. –М.: СОЛОН-Р, 2002. -512 с.

2. Шнайер Б. Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си. 2-е изд. –М.: Триумф, 2002.

3. Щербаков Л.Ю., Домашев А.В. Прикладная криптография. Использование и синтез криптографических интерфейсов. –М.: «Русская Редакция», 2003. -416. ил.

4. Рябко Б.Я., Фионов А.Н. Криптографические методы защиты информации: Учебное пособие для вузов. –М.: Горячая линия - Телеком, 2005. -229 с. ил.

5. Баричев С., Гончаров В.В., Серов Р.Е, Основы современной криптографии: Учебный курс. - М.: Горячая линия-Телеком, 2002. - 175 с.

6. Введение в криптографию. Под.ред. Ященко В.В., -М., МЦНМО, 2000.

7. Домарев В.В. Безопасность информационных технологий. - М.: 2002.

8. Баричев С. Основы современной криптографий. –М.: 2002.

9. Ибрагимов О.М., Оразов И. Криптография әдістері. // Элек. учебник. Свидетельство о гос. регистрации объекта в Комитете по правам интеллект. собст. МЮ РК. №678 от 29.04.2011.

10. Сухарев М. Delphi. Полное руководство. Спб.: Наука и техника, 2010. -1040 с.: ил.

11. Культин Н.Б. Delphi в задачах и примерах. –Спб.: БХВ-Перербург, 2005. -288 с.: ил.