ҚАЗ УТУ

А.Ш. Тұрым
Б.М. Мұстафина
А.Ә. Шайқұлова

АҚПАРАТ ЖАБУ НЕГІЗДЕРІ

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И.СӘТБАЕВ атындағы ҚАЗАҚ ҰЛТТЫҚ
ТЕХНИКАЛЫҚ УНИВЕРСИТЕТІ

А.Ш. Тұрым, Б.М. Мұстафина, А.Ә. Шайқұлова

АҚПАРАТ ЖАБУ НЕГІЗДЕРІ

Университеттің Ғылыми-әдістемелік кеңесі
оқу құралы ретінде ұсынған

Алматы 2011

А37
ЖОК 004(075-8)
ББК 32.81я 73

Тұрым А.Ш., Мұстафина Б.М., Шайқұлова А.Ә. Ақпарат жабу негіздері. Оқу құралы. - Алматы: ҚазҰТУ, 2011, 164 б.
Сурет – 40. Кесте – 23. Әдебиеттер тізімі – 39 атау.

ISBN 978-601-228-275-7

Оқу құралы ақпарат қорғаудың тиімді бағыттарының бірі болып саналатын криптографияға (криптографиялық қорғанышқа) арналған. Оқу құралы 3 тараудан тұрады.

Оқу құралында криптология тарихы және бүгінгісі, криптология ғылымының қалыптасуы, криптографиялық жүйелер мен шифрлау әдістерінің жіктелігі қарастырылған. Дәстүрлі және қазіргі заманғы симметриялық, асимметриялық криптожүйелер және қауіпсіз хэш-функциялар, цифрлық қолтаңбалар мен цифрлық сертификаттардың түрлері және пайдалану мысалдары келтірілген. Сондай-ақ, стеганографияның түрлеріне, негізгі атқаратын қызметтері мен қолданылу мақсаттарына түсінік берілген.

Оқу құралы «050704 – Есептеу техникасы және бағдарламалық қамтамасыз ету», «051002 – Ақпараттық қауіпсіздік жүйелері» мамандықтарында оқитын жоғары оқу орындары студенттеріне арналған.

ББК 32.81я 73

Пікір жазғандар: *М.А. Ахметова* – техн. ғыл. канд., доцент;
З. Жунусов – техн. ғыл. канд., доцент.

Қазақстан Республикасы Білім және ғылым министрлігінің 2011 жылғы жоспарына сай басылды

ISBN 978-601-228-275-7

АЛҒЫ СӨЗ

Құпиясыз мемлекет болуы мүмкін емес. Құпиялар – ғылымның, техниканың және саясаттың негізін құрайды. Бірнеше ғасыр бұрын ойлап табылған жазудың жалпы қол жеткізерлік қасиеті бар. Хабар алушыға байланысты бұл қасиетті – пайдалы немесе зиянды деп қарауға болады. Жазумен қатар, құпия хат (грек тілінде криптография) дамиды. Құпия – хат-хабардың мағынасын адамдардан жасыруға және оған тек белгілі бір ғана тұлғалардың қолын жеткізе алырықтай етіп істеуге арналған. Сондықтан, ақпарат арналмаған адамдардан хабарды жасыру тәсілдері туралы ғылым, яғни криптография пайда болған [16, 19].

Құжаттар әзірлеу, өңдеу, тасымалдау және сақтау кезінде ақпараттық технологияларды қол данудың кеңеюі – белгілі бір жағдайларда олардың мазмұнының жасырындылығын сақтауды, дұрыстығын, тұтастығын және шүбәсіздігін қамтамасыз етуді талап етеді. ЭЕМ-ның кең таралуымен байланысты, компьютерлердегі жасырын деректерді рұқсатсыз қол жеткізуден қорғау мәселелеріне көбірек назар аударыла басталды. Ақпаратты қорғау – ақпараттың сыртқа кетуінің, оны ұрлаудың, жоғалтудың, рұқсатсыз жоюдың, өзгертудің, маңызына тимей түрлендірудің, рұқсатсыз көшірмесін алудың, бұғаттаудың алдын алу үшін жүргізілетін ұйымдастырушылық және техникалық шаралар кешені.

Құпияларды қорғаудың негізгі екі тәсілі бар. Біріншіден, құпияның болу фактісін жасыруға тырысуға болады: құпия жоқ – оны білмекші болатындар да жоқ. Осы мақсатты жүзеге асыруға арналған негізгі тәсіл – ***стеганография***, яғни құпия ақпаратты жалпы қол жеткізерлік хабар ретінде тасымалдау. Бұрынғы кезде бұл мақсат үшін белгілі бір әріптері немесе сөздері осы құпия жолдаудың мәтінін құрайтын хаттар, кітаптар немесе газеттік мақалалар қолданылған. Қазір, жаппай ақпараттандыру кезінде, ең көп таралған стеганографикалық материал – бастапқы ақпараттың онша мәні жоқ бөлігінің орнына кез келген (керек жағдайда, құпия) хабарды

орналастыруға болатын форматы бар графикалық, дыбыстық, т.б. мультимедиялық деректер.

Цифрлық ерекше белгілерді ендіру негізінде құрылған стеганографиялық қорғаныш технологиясы – аудиовизуальдық деректерді рұқсатсыз қолданудан қорғау және олардың көшірмелерін рұқсатсыз таратуды бақылаудың болашағы зор бағыты болып келеді. Аудиовизуальдық деректерге (авторлық құқық объектілеріне) көрінбейтін тамға («ерекше белгі») ендіріледі. Аталған технология, авторлық құқықтың заңдылығын немесе заңсыздығын дәлелдеуге мүмкіндік береді. Цифрлық ерекше белгілер (digital watermark) – мультимедиялық деректер ішіне жасырылған авторлық құқықтың тамғасы.

Екінші тәсіл – тура қарама-қарсы қағидатқа негізделген: маңызды құпия ақпарат тасымалдап немесе сақтап жатқанымызды ешкімнен жасырмаймыз, бірақ ақпарат оның нағыз мәнін тек кімге осы хабар арналған болса ғана, түсіне алатындай түрде тасымалданады немесе сақталады. Мұны тек ***криптография*** ғана жүзеге асыра алады. Криптография (криптографиялық қорғаныш) ақпарат қорғаудың тиімді бағыттарының бірі. Ол мемлекеттік және коммерциялық құрылымдарда кеңінен пайдаланылады. Құпия деректер белгілі бір алгоритм арқасында шифрланады және оларды дұрыс оқу үшін, келген хабарды алдымен кері шифрлау керек болады. Ақпаратқа қол жеткізуді реттеу – хабарды шифрлау және кері шифрлауға қажетті кілттік ақпаратты қолдану арқылы қамтамасыз етіледі. Мұндай кілттік ақпаратқа тек құпия деректерге қатынас құруға рұқсат етілгендер ғана ие бола алады.

Ақпараттық технологиялардың дамуымен, әсіресе Интернет жүйесінің кең таралуына байланысты, жасырын ақпаратты қорғау – осы заманның аса өзекті мәселелерінің бірі. Қазір шифрлардың түрі өте көп, ал қазіргі заманғы есептеу құралдары криптографияда бұрын қолданылмай келген алгоритмдерді пайдалануға мүмкіндік береді.

Компьютерлік технологиялардың кеңінен таралуы, криптография іс-әрекеттерінің аймағы өзгеруіне септігін тигізді. Яғни, әр түрлі мәселелермен толықты: электрондық цифрлық

қолтаңбалар жүйелерін әзірлеу, қашықтағы пайдаланушыларды идентификациялау, жалған хабар алуға мүмкіндік бермеу әдістерінің пайда болуы, электрондық төлем жүйелерін қорғау құралдарын жасау, т.б.

Бұл оқу құралы оқырмандарды шифрлар жайындағы ғылыммен және оның қысқаша тарихымен таныстырады. Бағалы кеңестер мен ескертулер жасаған оқырмандар мен мамандарға алғысымызды айтамыз.

Авторлар

1. НЕГІЗГІ ТҮСІНІКТЕР МЕН АНЫҚТАМАЛАР

Ақпарат қорғау әдістерінің даму үрдісі соңғы жылдарда криптография және стеганография әдістерінің дамуына көбірек назар аударылатынын көрсетеді. Internet жүйесінің тез дамуы, сондай-ақ Internet жүйесіндегі авторлық құқықты қорғау, электрондық поштаны және сауданы ұйымдастыру, хакерлердің заңға қайшы іс-әрекеттері, т. б. осы сияқты шешілмеген қайшы проблемалар криптография және стеганография әдістерінің даму үрдісіне өз ықпалын тигізіп отыр.

Компьютерлік стеганография және криптография әдістерін кешенді түрде қолдану, деректер қорғаудың болашағы зор бағыты деуге болады. Бұл жол (бағыт) ақпарат қорғаудың белгілі әдістерінің әлсіз жақтарын жоюға және ақпараттық қауіпсіздікті қамтамасыз етудің тиімдірек әдістерін әзірлеуге мүмкіндік береді [1, 5, 16, 19, 32].

Стеганография (steganography) – байланыс бар екендігінің өзін жасыратын байланыс ұйымдастыру әдісі. Стеганография әдістері, ендірілген (қоса салынған) құпия жолдаулар бар екендігіне күдік тудырмайтындай етіп оларды зиянсыз хабарлар құрамына ендіруге мүмкіндік береді.

Криптография (cryptographic) – ақпаратты рұқсатсыз пайдаланушылардан қорғау мақсатында оны түрлендіру әдістері жайындағы ғылым. Ол ақпаратты оқу (бұрынғы қалпына келтіру), тек оның кілтін білген кезде ғана мүмкін болатындай етіп түрлендіреді.

1.1. Криптография және криптоанализ

Дипломатиялық, әскери және өнеркәсіптік құпиялар, әдетте шифрланған түрде жіберіледі немесе сақталады. Криптографиялық түрлендіру арқылы құпия хабарларды тасымалдаудың классикалық сұлбасы мынадай. Жіберуші жақта хабар белгілі бір кілт арқасында шифрланады, одан кейін осылайша даярланған шифрқұжат, қабылдаушыға ашық байланыс арнасымен тасымалданады, ал кілт болса (құпиялыққа

кепілдік беретін) жабық арнамен жіберіледі. Қабылдаушы жақ өзіне белгілі кілт арқылы шифрқұжатты кері шифрлайды. Сөйтіп, келген хабарды бастапқы қалпына келтіреді. Құпиялау мақсатына байланысты бұл сұлба біршама өзгертілуі мүмкін.

Криптографиялық түрлендірулер ақпарат қорғау кезінде екі мақсатты көздейді. Біріншіден, олар кілті жоқтардың ақпаратқа қол жеткізе алмауын қамтамасыз етеді. Екіншіден, ақпараттың рұқсатсыз бұрмалануын тиісті сенімділікпен іздеп-табуға мүмкіндік береді.

Классикалық криптография – тиімді криптографиялық алгоритм қолданылғанда және кілттің құпиялылығы мен тұтастылығы сақталған кезде ғана қорғанышқа кепілдік береді.

Криптология – ақпаратты түрлендіру арқылы, оны қорғаумен шұғылданады. Криптология ғылымы – ақпаратты шифрлау және кері шифрлау, сондай-ақ шифрларды әзірлеу, шифрды ашу мәселелерімен де шұғылданады. Криптология - kryptos (*құпия*) және logos (*ғылым, ой*) деген грек сөздерінен шыққан. Оны шартты түрде криптография және криптоанализ деп екі бағытқа бөлуге болады. Бұл екі бағыттың мақсаттары қарама-қарсы.

Криптография (cryptographic) – ақпаратты түрлендірудің математикалық әдістерін іздеу және зерттеумен, яғни деректерді шифрлау және кері шифрлаумен шұғылданады. Сонымен қатар, криптография – ақпаратты бұрмалаудың алдын алу немесе оның пайда болу себебін растау үшін де қолданылады. Өзге адамдардан ақпараттың құпиясын сақтап қалу, криптографияның негізгі мақсаты болып есептеледі. Ақпаратпен рұқсатсыз таныспақшы болған адамдарды *қаскөйлер (қаскүнемдер)* деп атайды.

Криптоанализ (криптоталдау) – көбінесе шифрқұжатты оның кілтін білмей-ақ, қалайша кері шифрлау керек мәселесімен және кей кезде, қолданылып жүрген шифрлау жүйесін бұзып-ашумен айналысады. Сонымен, криптоанализ – шифрланған хабардың бастапқы ашық мәтініне қол жеткізуге бағытталған. Сәтті жүргізілген криптоаналитикалық зерттеулер негізінде хабардың бастапқы ашық мәтінімен қатар, оның кілтін де ашуға

болады. Криптоаналитик шифрланған хабарды немесе кілтті (немесе екеуінде) оқуға мүмкіндік беретін криптожүйенің осал жерлерін іздеумен шұғылданады. Сонымен, *криптоаналитик* деп – шифрды ашу мүмкіндігін зерттейтін адамды атайды. Шетелде олар өздерін кодтарды бұзып-ашушылар (breaker), шабуылшылар (attacker) және ұрылар (sneaker) деп те атайды. Сонымен, криптографтар құпиялықты қамсыздандыруға, ал криптоаналитиктер – оны бұзып-ашуға ұмтылады.

Криптографиялық жүйе (криптожүйе) – шифрлау алгоритмі, сондай-ақ алуан түрлі кілттердің, ашық және шифрланған мәтіндердің жиынтығы.

Криптографиялық алгоритм (***шифр*** немесе ***шифрлау алгоритмі***) деп – шифрлау және кері шифрлау үшін қолданылатын математикалық функцияны атайды. Дәлірек айтсақ, мұндай функция екеу: біреуі шифрлау үшін, ал екіншісі – кері шифрлау үшін қолданылады [2].

Кілт (key) – ақпаратты шифрлау және кері шифрлау, сондай-ақ оған қол қоюға арналған цифрлық код. Кілттің ортақ, жекеменшік және құпия деп аталатын түрлері болады.

Шифрлау – E функциясы да, кері шифрлау - D функциясы, осы (K) кілтке тәуелді болады: $E_K(P) = C$, $D_K(C) = P$.

Сонда мына тепе-теңдік әділ болады: $D_K(E_K(P)) = P$.

Бұл жерде, P (plaintext) – ашық мәтін, ал C (ciplertext) – E функциясы мен K кілті арқылы шифрланған мәтін (шифрмәтін).

Шифр (cipher) – қаскөйде мәлімет (құпия ашу кілті) болмаған жағдайда, ашық ақпаратты бастапқы қалпына келтіре алмайтындай етіп түрлендіру үшін қолданылатын шартты белгілер тізбегі. “Шифр” терминінің түбірі араб сөзінен шыққан. XV ғасырдың басында жарық көрген *“*Шауба Әл-Аша*”* деп аталатын араб энциклопедиясында шифрлар жайында арнаулы бөлім болған.

Криптоберіктілік – шифрдың негізгі сипаттамасы болып саналады. Ол кілті белгісіз жағдайда, шифрдың кері шифрлауға (яғни криптоанализге) беріктігін анықтайды. Әдетте бұл сипаттама шифрды ашуға керек уақыт мөлшерін анықтайды.

Классикалық криптография теориясының негізін қалаушы Клод Шеннон, криптоберіктіліктің екі түрін ажыратады: *теориялық* және *практикалық* [39].

Шифрланған хабарларда кездейсоқтықтарды зерттеу, олардың криптоберіктілігін анықтау үшін өте маңызды. Криптограммада статистикалық заңдылықтар мен корреляциялардың болуы, криптоанализді жеңілдететіні белгілі. Криптоберіктілікті жоғарылату үшін криптограмманың үлестірім заңын мүмкіндігінше бірқалыптыққа жақындату керек.

Шифрлау (ciphering, encryption) – белгілі бір адамнан басқалар оқи алмайтындай етіліп ақпаратты математикалық, алгоритмдік (криптографиялық) түрлендіру әдісі. Қабылдаушы жақ бұл ақпаратты дұрыс оқу үшін оны кері шифрлауы (decryption) керек. Шифрлау – бөлшекті (әрбір кезекті бөлшек тәуелсіз шифрланады) және ағынды (әрбір таңба бір-бірінен тәуелсіз шифрланады) түрде жүргізілуі мүмкін.

Криптографиялық техникаға шифрлау және кері шифрлау алгоритмдерінен басқа, ***құпия кілттер*** де жатады. Кілттерді қол жетпестей, ал олардың оқылғанын белгілі ету үшін әр түрлі айлалар қолданылады. Кілттерді криптографиялық қойын дәптерлерде (блокноттарда) сақтайды. Көбінесе қойын дәптерлерге кілттердің өздерін емес, тек олардың шифрмәтінін жазады, ал кілтті шифрлаушы адам оны жадында сақтайды. Құпия кілттермен айырбас жасау, бірқатар жағдайда проблема болып табылады. Сондықтан, соңғы жылдары ашық кілтті шифрлау жүйелерді қолдану бағытында қарқынды зерттеулер жүргізілуде. Шифрлау үшін мұндай жүйелерде ашық кілт, ал кері шифрлау үшін – құпия кілт болады. Криптографияны екі жағдайда қолдануға болады: деректер тасымалдау кезінде оларды қорғау және деректерді сақтау кезінде оларды қорғау [19].

Деректер тасымалдау кезінде оларды қорғау. Бұл кезде құпия ақпарат, жіберуші жақта шифрланады, ал қабылдаушы жақта – кері шифрланады. Қаскөйлер байланыс арнасында оны жолай ұстап алса да, кілттік ақпараты (кілттері) болмағандықтан, шифрмәтінді олардың кері шифрлауға мүмкіншіліктері болмайды.

Кілттік ақпаратты абоненттерге құпия түрде жеткізу керек және оны мына екі тәсілмен істеуге болады:

- кілттер физикалық түрде (электрондық кілттер, пластикалық кәртішкелер, әкімші жекеше хабарлайтын құпия сөздер түрінде, т. б.) апарылады;
- кілттер шифрланған түрде байланыс арнасымен жіберіледі.

Кілттерді тарату проблемасының күрделі болу себебі: шифрлаудың беріктілігін жоғарылату үшін оларды мүмкіндігінше жиірек ауыстыру қажет. Ал ол, кілттерді физикалық түрде жеткізуге кететін шығындардың өсуіне әкеледі және жүйенің бәрін «баға/сапа» тұғырынан қарағанда тиімсіз етеді. Сондықтан тәжірибе жүзінде әдетте, қисындастырылған үлгілер қолданылады: абоненттерге ұзақ уақыттық кілттер физикалық түрде жеткізіледі, олардың көмегімен сеанстық деп аталатын кілттер шифрланып тасымалданады, осыдан кейін ғана оларды қолдана отырып, құпия ақпарат шифрланады.

Деректерді сақтау кезінде оларды қорғау. Архивте сақталынған ақпаратты қорғау да маңызды мәселенің бірі. Бұл – қаскөйлердің компьютерлерге немесе ақпарат сақталатын сыртқы құрылғыларға рұқсатсыз қатынас құру қауіп-қатерімен байланысты. Сақталатын ақпаратты шифрлау кезінде кілттерді таратудың қажеттігі болмайды: шифрлауды да, кері шифрлауды да бір адам жүзеге асырады (тасымалданатын деректерді қорғаудан айырмашылығы). Сол себептен, сақталатын ақпаратты криптографиялық қорғау үшін баяу асимметриялық алгоритмдер қолданылмайды.

Сақталатын деректерді криптографиялық қорғау мәселесін әр түрлі екі тұрғыдан қарау керек: компьютердегі ақпараттың барлығын толық жабу және қатты дискілердегі немесе сыртқы сақтауыштардағы тек қана өте бағалы ақпаратты ішінара шифрлау. Бірінші мәселе – криптожүйелерге олардың жұмыс істеу жылдамдығы тұрғысынан қойылатын ерекше талаптармен өзгешеленеді: шифрлау және кері шифрлау (компьютер пайдаланушысы сезбейтіндей) тез жүргізілуі керек.

Криптоанализ элементтері

Қол шифрлары арқылы жабылатын хабарлар онша ұзын болмайды. Сондықтан, оларды кері шифрлауды адамдар анағұрлым нәтижелі орындай алады. *Машиналық шифрлар* – есептеу жағынан өте күрделі және өте ұзын хабарларды криптографикалық жабуға арналған. Мұндай хабарды қолмен кері шифрлауға тырысудың қажеттігі жоқ. Бұл жағдайда бас рөлді криптоаналитиктер атқарады. Әрқашан шифрдың түрі мен хабардың тілі белгілі деп саналады: оларды анықтау шифрқұжаттың әліпбиі мен статистикалық қасиеттері негізінде жүргізіледі. Сондықтан, тек кілт қана белгісіз деп есептеледі, тек оны бұзу-ашу керек болады.

Шифрлау жүйелерін криптоанализ жасау кезінде оларды бұзу-ашу мүмкіндігі, көбіне бұзушының біліктілігіне тәуелді болады. Криптоаналитик криптографияға қатысы бар математикалық пәндердің әдістерін жақсы меңгеруге тиісті. Сонымен қатар, шифрды бұзып-ашуға қол жеткізетін әдісті табуға бейім сезім болуы керек [16].

Шифрды бұзып-ашу қиындығы, тек оның құрылмасына тәуелді. Сондықтан, кез келген шифрды бұзып-ашуға жарамды криптоанализдың жалпы қағидаттары өте аз. Шифрды бұзып-ашуға неғұрлым көп уақыт қажет болған сайын, оны берік деп санауға себеп көп. Бірақ, шифрдың беріктілігі міндетті түрде оның қауіпсіз шифр екендігін білдірмейді. Бұл шифрды бұзып-ашу әдісі әзірше табылмағанын немесе табылған әдістің жария етілмегенін білдіруі де мүмкін. Тасымалданатын ақпарат беріктілігі жайындағы ұғымды алғашқы рет А.Россиньол (Франция) былайша тұжырымдаған: “Әскери шифрдың беріктілігі бұйрықты орындауға қажетті мезгіл ішінде құпиялықты қамтамасыз етуі керек. Дипломатиялық шифрдың беріктілігі, құпиялықты бірнеше он жылдар бойы қамтамасыз етуі керек”.

Қазіргі заманғы криптологияда шифрдың беріктілігі, тек қолданылатын кілттің құпиялығымен ғана анықталады деп қабылданған.

Криптографиялық алгоритмдерге қойылатын талаптардың негізгілері – сенімділік, бұзу-ашу әрекеттеріне тұрақтылық. Бұзып-ашылмайтын шифрлар жоқ. Шифркұжаттарды бұзу-ашу шараларының құны, хабар ішіндегі ақпараттан әдейі қымбат істелінеді немесе бұзу-ашу уақыты өте ұзартылады. Шифрдың пайдаланылу (өмір) уақытын 25 жылдан артық алу орынсыз. Мәселен, Британияда үкіметтің өте құпия шешімдері осы мезгіл өткесін ғана тарихшылар үшін жария етіледі. Қазақстан Республикасының заңнамасына сәйкес мемлекеттік құпия болып саналатын мәліметтердің құпиялылық мерзімі 30 жылдан аспауы керек.

Шифрды бұзу-ашудың ең қарапайым әдісі – кілттердің барлық вариантарын бірінен соң бірін таңдап алып, солардың әрқайсысымен криптограмманы кері шифрлау және алынған нәтижелерге талдау жасау [19]. Бұл ең баяу, сонымен қатар ең сенімді жол және оны дәстүрлі шифрлау алгоритмдердің бәріне қолдануға болады. Шифрды осы әдістің көмегімен бұзу-ашудан қорғаудың бір тәсілі: мүмкін болатын кілттердің саны мен ұзындығын арттыру. Кілттерді жаппай тексеріп шығу әдісінен басқа, криптографиялық алгоритмдерді ашудың біраз аналитикалық келістері де бар. Олар криптографиялық алгоритмдердің осал жерлерін қолдануға бейімделген.

Криптографиялық алгоритмдерге жасалынатын шабуылдардан қорғану үшін, құпия кілттің ұзын болғаны жақсы дегенбіз. Бірақ, неғұрлым қолданылатын кілт ұзын, әрі криптографиялық алгоритм неғұрлым күрделі болса, есептеу қорларына қойылатын талаптар да солғұрлым жоғары болады. Осыдан, шифрларға қойылатын екінші талап, яғни жұмыс істеу жылдамдығы шығады.

1.2. Криптология тарихы мен бүгінгісі

Криптографияның даму тарихында үш негізгі кезеңді атап кетуге болады [1, 16]. Ісі тек қолдық шифрлармен болған және ертеде басталып, XX ғасырдың 30-шы жылдарының соңында ғана аяқталған *бастапқы кезең*.

Екінші кезең – алдымен механикалық, сонан соң электромеханикалық және электрондық шифрлау құрылғыларын жасау, оларды практикада кең енгізу және құпияланған байланыс желілерін құрумен байланысты. Осы кезеңді, ұзын бір реттік кілтті қолданатын телеграфтық шифрлайтын машиналардың пайдаланыла бастауы деп есептеуге болады. XX ғасырдың 70-жылдарының ортасында (байланыс желілерінің, электрондық поштаның және ауқымды ақпараттық жүйелердің дамуына байланысты) құпия кілттерді тарату және авторлықты растау, ең негізгі мәселелер болып есептеледі.

1976 жыл, криптология дамуының *үшінші кезеңінің* басы болып есептеледі. Осы кезде американдық математиктер Диффи мен Хеллман абоненттерді алдын ала құпия кілттермен жабдықтауды талап етпейтін (ашық кілтті шифрлау деп аталатын) құпияланған байланыс ұйымдастырудың мүлде жаңа түрін ұсынған. Осының нәтижесінде, қырқыншы жылдары К. Шеннон қалыптастырған келісім негізінде құрылған криптографиялық жүйелер пайда бола бастады. К. Шеннон, шифрды оны ашу – қазіргі заманғы ЭЕМ мүмкіншіліктерінен басым түсетін есептеу көлемдерінің орындалуын талап ететін, математикалық есепті шешуге парапар болатындай етіп жасауды ұсынды. Криптография дамуының жаңа кезеңі, шифрланған байланыстың толық автоматтандырылған жүйелерінің пайда болуымен сипатталады.

Шифрлардың пайда болуы

Шифрлау жүйелері біздің дәуірімізге дейінгі IV-ші мыңжылдықта жазумен бір уақытта пайда болған. Құпия түрде хат алысып тұру әдістері Мысыр, Шумер және Қытай сияқты көптеген ертедегі қоғамдарда бір-бірінен тәуелсіз ойлап табылған. Кодтар Вавилонда және Ассирияда өте белгілі болған, ал ежелгі египеттіктер (мысырлықтар) ең болмағанда 3 шифрлау жүйесін қолданған.

Ежелгі үнділердің қолжазбаларында мәтін түрлендірудің 64 тәсілі баяндалған. Қолжазбадағы таңбалар ретсіз, белгілі бір ережелерге сай жазылған. Осындай тәсілдердің көбін криптографиялық деп қарауға болады. Қолданылатын шифр жүйелері жайындағы ең шынайы мәліметтер, ежелгі грек мемлекеттерінің пайда болу дәуіріне жатады. Бұл дәуірде криптографияны қолданушылар қатарында әкімшілік және діни билік құрылымдары болған.

Араб мемлекеттерінің өркендеу дәуірінде (біздің дәуірдің VIII ғ.) криптографияда жаңа даму қарқын алды. “Шифр” және “цифр” – араб текті сөздер. 855 жылы шыққан “Адамның ертедегі жазу жұмбағын ашудағы үлкен ынтасы жайындағы кітапта” шифр жүйелерінің сипаттамалары келтірілген. 1412 жылы Шехаб Әл-Кашканди құрастырған 14 томдық энциклопедияда криптография туралы бөлім бар. Онда авторға белгілі барлық шифрлау тәсілдері берілген. Сонымен қатар, ашық және шифрланған мәтіннің жиілік сипаттамалары негізінде шифр жүйесіне криптоанализ жасау туралы айтылған. Энциклопедияның осы бөлімінде Құран мәтінін талдау негізінде араб әріптерінің қайталану жиілігі бойынша орналасқан тізбесі келтірілген.

Біздің дәуірімізге дейінгі II-ші мыңжылдықта ертедегі семиттік әліпбиде барлығы 30-ға жақын таңба болды. Мәтіндер қарапайым ауыстыру әдісімен шифрланатын болған. Бірінші әріптің орнына – әліпбидің соңғы әрпі жазылған, екінші әріптің орнына – соңғының алдындағы, т.б. Бұл ертедегі шифрлау әдісі – *атбаш* деп аталған.

Келесі шифр ақпарат жіберуші мен қабалдаушыға мәлім белгілі бір ереже бойынша хабар әріптерінің орнын ауыстырумен байланысты. Біздің дәуірге дейін (б. д. д.) V–VI ғасырда (грек мемлекеттерінің бірі) Спартада дамыған криптография болған. Дәл осы уақытта шифрлауға арналған арнаулы таяқ (“сцитала”) пайда болды. Ол орын ауыстыру шифрында қолданылған. Сыртына таспа (папирус жапырағының тілімі) оратылатын таяқтың атына сәйкес бұл шифр *сцитала* деп аталған. Шифрлау алгоритмі мынадай: таяққа таспаны орайды

да, оралған таспаның үстінен таяқтың бойымен ашық мәтінді жазады. Оралған таспада шифрмәтін жазылып шығады, ол ыңғайлы және жылдам. Шифрдың кілті – таяқтың жуандығы және әліпби.

Ежелгі грек ғалымы Аристотель (б. д. д. 384–322 жж.) криптографияда сцитала шифрын ашу тәсілінің авторы ретінде белгілі: таяқтың дәл диаметрін білмей-ақ, Аристотель конус тәрізді таяққа шифрланған таспаны орап, мәтін дұрыс оқылып басталғанша, таспаны таяқ бойында әрлі-берлі жылжытқан.

Ежелгі грек тарихшысы Полибийдің шифрында (б. д. д. шамамен 200–120 жж.) немесе полибий квадратында, мынадай шифрлау алгоритмі болған. Өлшемі 5х5 ұялы квадрат грек әліпбиінің символдарымен кездейсоқ түрде толтырылған. Шифрдың кілті – квадрат толтыру тәртібі.

Гай Юлий Цезарьдың (б. д. д. 102 немесе 100–44 жж.) шифры қарапайым ауыстыру шифрының бір түрі болып келеді және ол мына алгоритмге сәйкес құрылған: бірінші әріптің орнына – төртінші әріпті оқу керек. Шифрдың кілті – ығыстыру аралығы және әліпбидің өзі. 26 әріптен тұратын әліпбиде Цезарь шифрын қолданғанда VENI VIDI VICI (келдім, көрдім, жеңдім) ашық мәтінінен **YHQL YLGL YLFL** шифрмәтіні алынады.

Шифрлауға арналған құралдар ертедегі заманда да болған. Мәселен, б. д. д. V ғасырда Спарта мемлекетінде құпия әскери байланыс жүйесі болған. Бірінші криптографиялық құрылғы (сцитала) көмегімен олар қарапайым ауыстыру әдісін қолдана отырып, хабарларды шифрлаған. Біздің д. д. IV ғасырда римдіктер шифрлау процедурасын оңайлату үшін шифрлауыш дискілерді қолдана бастаған.

Криптология ғылымының қалыптасуы

Орта ғасырдың соңына қарай, криптография қайтадан қолданыла бастайды. Сол кездің қол шифрларында кестелер жиі қолданылды. Олардың көмегімен хабардағы әріптердің орнын ауыстырудың қарапайым, шифрлауыш процедуралары жүзеге асырылды. Кілт ретінде кесте өлшемі, орын ауыстыруды көрсететін сөйлем немесе кестелердің арнайы ерекшелігі

қолданылды. ***Кілтсіз жалғыздалған (жай) орын ауыстыру*** – ең қарапайым шифрлау әдістерінің бірі. Шифрды күрделендіруі үшін кестенің бірінші қатарына кілттік сөз қосылып, кілт әріптерінің реттік нөмірлеріне сәйкес бағандардың орнын ауыстырады. Бұл шифрлау әдісі ***кілт бойынша жалғыздалған (жай) орын ауыстыру*** деп аталады. Қосымша жасыру үшін шифрланған хабарды қайтадан шифрлауға болады. Бұл тәсіл - ***екі рет орын ауыстыру*** деп аталады. Бірінші кестеде бағандардың, ал екінші кестеде – қатарлардың орындары ауыстырылады.

Орта ғасыр ғалымдары қатарлар мен бағандар (және әрбір диагональ) бойынша саналған сандардың сомасы бір мәнге тең болып келген квадраттардың сиқыршылық күші бар деп есептеген. Олар осындай ***сиқырлы квадраттарды*** деректерді шифрлау үшін пайдаланған. Бір қарағанда, сиқырлы квадраттар саны өте аз сияқты. Бірақ, олардың саны квадрат өлшемінің артуымен өте жылдам өседі. Мәселен, 3х3 өлшемді кестеде бір сиқырлы квадрат бар, 4х4 өлшемді кестеде – 880, 5х5 өлшемді кестеде – 25000.

Орта ғасырларда сауданың кеңінен дамуы ерекше шифрларды талап етті. Кілттік сөз негізінде құрылған мұндай қарапайым шифрлар – ***цифрларды әріптерге ауыстыру*** деп аталады. Пайдаланушылыр алдын ала әріптері цифрларға сәйкес келетін ортақ кілттік сөзді қолдануға келіскен.

Гронсфельд шифры (1734 жылы бельгиялық Хосе де Бронкхор, граф де Гронсфельд жасаған) Цезарь шифрының өзгертілген бір түрі болып келеді. Бұл алгоритмде ығыстыру аралығы тұрақты сан арқылы емес, кілт (гамма) арқылы беріледі. Шифрмәтін құру үшін ашық мәтін әрпінің орнына, әліпбидің кілт цифрына жылжытылған әрпі таңдап алынады. Бұл шифрлардан басқа көбінесе *қарапайым ауыстыру шифры* қолданылған. Мұнда хабардың әрбір әрпі шифрдың оған сәйкес әрпімен ауыстырылады.

Күрделі ауыстыру шифрлары – көпәліпбилік деп аталады. Себебі, негізгі хабардың әрбір символын шифрлау үшін өзінің қарапайым ауыстыру шифры қолданылады. ***Көп әліпбилік***

ауыстыру шифрын итальян ғалымы Леон Батист Альберти ұсынған. Оның 1466 жылы жазылған "Шифр туралы трактат" кітабы криптология саласындағы (араб қолжазбаларын есептемегенде) әлемдегі бірінші ғылыми еңбек болып саналады. Бұл кітапта әр түрлі шифрлау тәсілдері қарастырылған. Ол шифрдан басқа, оны жүзеге асыруға арналған айналатын доңғалақтардан тұратын құрылғыны (шифрлайтын дискіні) толық сипаттап берген. Кейінірек Альберти қайта шифрлау кодын ойлап тапты.

Вижинер шифры 400 жыл бойы кері шифрланбайтын шифр деп саналған, сондықтан әскери шифр ретінде кеңінен қолданылған. Шифрдың бұл түрі осы күнге дейін жеткен.

1518 жылы Германияда баспадан криптография жайында бірінші кітап шықты. Иоганнес Трисемус өзінің "Полиграфия" атты кітабында бірталай шифрлар жайында мәлімет келтірген. Олардың біреуінде ол көпәліпбилік ауыстыру идеясын одан әрі дамытады. Сонымен қатар, ол осы трактатта бірінші болып кездейсоқ ретте әліпбимен толтырылған шифрлауыш кестелерді қолдануды жүйелі түрде сипаттаған. Шифрлау бір-бір әріп бойынша жүргізілетіндіктен, мұндай кестелік шифрлар – *монограммалы шифрлар* деп аталады. Трисемус бірінші болып бір мезгілде екі-екі әріптен шифрлауға болатынын байқаған. Мұндай шифрлар – *биграммалы* деп аталады. Ең белгілі биграммалы шифрға мысал ретінде Плейфер (Playfair) шифрын келтіруге болады. Бұл шифрды Ұлыбритания бірінші дүниежүзілік соғыста қолданған.

Сол жылдары итальян математигі, әрі философы Джераломо Кардано криптография жайында бірнеше кітап жазып, ***трафареттер әдісін*** сипаттап берген.

XV–XVIII ғасырларда математикада криптографияда шифрларды талдау және кері шифрлау үшін қолданылатын аппарат негіздері жасалған. Сонымен, XVIII ғасырдың басында криптография дербес ғылым түрінде қалыптасты.

1894 жылы ағылшын Чарльз Уитстон ***"қос квадрат"*** деп аталатын биграммалармен шифрлаудың жаңа әдісін тапты. Оның ***Полибий әдісінен*** айырмашылығы – "қос квадратта" бір

уақытта көлденең орналасқан екі кесте қолданылады, ал шифрлау – Плейфер шифрындағы сияқты биграммалар арқылы жүргізіледі.

XX ғасырдың 40-жылдарында қолданбалы математиканың дамуындағы болған сапалы серпіліс қана, криптографияны ғылым ретінде қарауға мүмкіндік берді. Криптографияның осы кезеңінің тарихы математик Элвуд Шеннон атымен байланысты. Ол математикалық әдістермен шифрлаудың сенімділігін зерттеген. Осы зерттеулердің нәтижесі: символдардың кездейсоқ тізбегі ешқандай мәнді алып жүрмейді, ал ақпараттанудың криптологиямен байланысы – шифрқұжаттың, кілттің және хабардың табылған статистикалық қасиеттерін хабарды кері шифрлау (яғни хабардың нақты мазмұнын табу) үшін қолдануға мүмкіндік береді.

Шифрлау және кері шифрлау үрдісін автоматтандыру. 1790 жылы Томас Джефферсон (АҚШ-тың болашақ үшінші президенті) *цифрлық шифрлауыш доңғалақ* ойлап тапқан. Мұндай машиналардың жұмыс істеу қағидаты арифмометрге өте ұқсас. Этьен Базери 1891 жылы *Базери цилиндрі* деп аталған құралды ұсынды. Ол құрсауына кездейсоқ түрде әліпби қондырылған 20 дискіден тұрған. Шифрлар алдында дискілер кілтке сәйкес анықталатын тәртіппен, ортақ белағашқа орналастырылған. Практикада қолдануға жарайтын *бірінші криптографиялық машинаны* Жильбер Вернам тек 1917 жылы ұсынған [16].

Дүниежүзілік соғыстар аралығында барлық алдыңғы қатарлы елдерде электромеханикалық шифрлауыштар пайда болған. Олар коммутациялық (немесе роторлар) және күпшекті дискілер негізінде жасалған. Шифрлауыштың бірінші түрінің мысалы ретінде *"Энигма"* шифромашинасын, ал екіншісінің мысалы ретінде американдық *М-209* шифромашинасын кетіруге болады [19].

XX ғасырдағы *компьютерлер* шифрларға және оларды кері шифрлауға деген көзқарасты толық өзгертуге мәжбүр етті. Өз құпияларын қорғауды бұрын армандамағандар зор

мүмкіншіліктерге ие болды. Ал қаскөйлердің қарамағында бөтен құпияларға енуге арналған құралдар пайда болды.

Цифрлық қолтаңба және сертификаттар

Кейбір шифржүйелерде хабардың цифрлық қолтаңбасы құпия кілттің көмегімен алынады. Ең оңайы – хабарды осы кілтпен шифрлап тастау, бірақ хабардың ұзындығы өте үлкен болуы мүмкін. Сондықтан, үнемдеу үшін тек хабардың бақылау қосындысы ғана шифрланады. Мұндай қосынды – ***имитоендірме*** деп аталады. Құжаттың иесін анықтау мәселесін, электрондық цифрлық қолтаңбаны қолдану арқылы шешуге болады [1, 3, 22, 34, 37].

Объектіні *идентификациялау* (identification) – қорғау жүйесі функцияларының бірі. Егер объект желіге кірмекші болса, онда бұл функция бірінші кезекте орындалады. Егер идентификациялау операциясы сәтті аяқталса, онда объект заңды деп есептеледі.

Келесі қадам – объектіні *аутентификациялау* (объектінің түпнұсқалығын тексеру, authentication). Бұл процедура объект өзін атағанда, ол шын мәнінде сол объект екендігін анықтайды. Электрондық құжаттарды аутентификациялаудың мақсаты, әр түрлі бұзушылық іс-әрекеттерден оларды қорғау. Мұндай іс-әрекеттерге мыналар жатады: жолай ұстау – желіге қосылған бұзушы құжаттарды (файлдарды) жолдан ұстап алып, оларды өзгертеді; мүләйімсу – С абоненті В абонентіне А абоненті атынан құжат жібереді; опасыздық (сатып кетушілік) – А абоненті В абонентіне хабарды жібере отырып, жібермедім деп мәлімдеме жасайды; ауыстыру – В абоненті құжатты өзгертеді немесе жаңа құжат жасайды және оны А абонентінен алдым деп мәлімдейді; қайталау – С абоненті, А абонентінің В абонентіне бұрын жіберген құжатын қайтадан жібереді.

Объект ұқсастырылып, оның шынайылығы дәлелденген соң, оның іс-әрекетінің аймағы мен жүйенің ол қатынас құра алатын ресурстары анықталады. Бұл процедураны – өкілеттік беру (*авторландыру*) деп атайды.

Есептеу желілерінде шынайылықты растау процедурасы байланыс сеансының басында, абоненттерді байластыру үрдісі кезінде орындалады. Бұл процедураның мақсаты – бүкіл ақпарат толығымен тиісті жеріне жетеді деген сенімділікті қамтамасыз ету үшін қажет.

Байласу орнатылғаннан кейін, деректер алмасу кезінде ақпарат қорғау талаптарының орындалуын қамтамасыз ету керек;

а) хабарды алушы деректер көзінің шынайы екеніне сенімді болуы керек;

ә) хабарды алушы тасымалданатын деректердің шынайы екеніне сенімді болуы керек;

б) жіберуші деректердің алушыға дейін жеткеніне сенімді болуы керек;

в) хабарды жіберуші алушыға жеткен деректердің шынайы екеніне сенімді болуы керек.

(а) және (ә) талаптарын орындауға арналған қорғау құралдарына цифрлық қолтаңба жатады. (б) және (в) талаптарын орындау үшін жіберуші ”қолына тапсырылды” деген хабарландыру алуы керек.

Егер осы аталған төрт талап іске асырылса, онда деректерді байланыс арнасы бойынша тасымалдаған кезде, қорғаныш функциялары толығымен қамтамасыз етілді деп есептеуге болады.

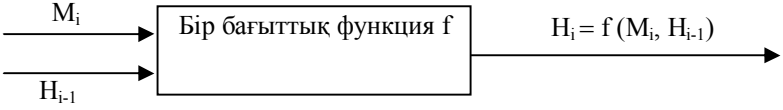
Хэш-функция

Хэш-функция (hash function) – әр түрлі ұзындығы бар M хабарды (қатарды) тұрақты ұзындықты бүтін санға сәйкестендіретін функция. Бұл функция хабарды қысу функциясы деп те аталады. h(·) хэш-функциясы аргумент есебінде кез келген ұзындықты M хабарын қабылдап алады да, тұрақты ұзындықты h(M)=H хэш-мәнін (қысылған хабарды) қайтарады. Әдетте, хэштелген ақпарат – кез келген ұзындықты негізгі хабардың қысылған екілік көрсетімі. Қысылған хабардың

ұзындығы, әдетте бастапқы хабардың ұзындығынан әлдеқайда кем болады [3, 22, 34, 37].

Екі әр түрлі хабарды бір хэш-функция көмегімен түрлендірген кезде бірдей қысылған хабар алынбайды. Қысылған хабарды қолданудың мәні: хабарға енгізілген кез келген өзгерту хэш-функция арқылы өңдеу барысында, оның қысылған хабарының өзгеруіне әкеледі. Егер хэш-функцияның қысқаша хабар бойынша бастапқы ақпаратты есептеуді күрделендіретін қасиеті болса, онда мұндай хэш-функцияны – *бір бағыттық хэш-функция* деп атайды.

Хэш-функциялардың көбісі бір бағыттық f(·) функциясы негізінде құрылады.



1.1-сурет. Бір бағыттық хэш-функция

Бұл функция берілген n ұзындықты екі кіріс мәндерін n ұзындықты бір шығыс мәніне түрлендіреді. Кіріс мәндері – бастапқы мәтіннің M_i блогы және мәтіннің алдыңғы блогының H_{i-1} хэш мәні (1.1-сурет). Мәтіннің ең соңғы блогын енгізгенде есептелген хэш-мәні, барлық M хабарының хэш-мәні болып табылады. Бір бағыттық хэш-функция үнемі тұрақты n ұзындығы бар H_i хабар қалыптастырып тұрады.

Хэш-функцияның қарапайым түрі ретінде хабардың бақылау қосындысын атауға болады. Бұл функцияны қолданған кезде хабарға әдейі қате (өзгеріс) енгізіп, бірақ бақылау қосындысының бастапқы мәнін сақтап қалуға болады. Сондықтан, мұндай хэш-функция криптографияда қолдануға жарамсыз. Криптографиялық хэш-функцияларға мынадай талаптар қойылады:

– кез келген M хабар (қатар) үшін h(M) тез есептелу керек;

– h(·) белгілі болған уақытта M табу қиын болуы (іс жүзінде тіпті мүмкін болмауы) керек;

– белгілі M хабар үшін h(M′)=h(M) болатын басқа бір M′≠M хабарды табу қиын болуы керек;

– h(M′)=h(M) болатын әр түрлі M және M′ хабарлар жұбын табу қиын болуы керек.

Мәселен, бірінші талап әруақытта орындалуы керек, ал электрондық қолтаңба үшін үшінші талаптың орындалуы өте маңызды.

Хэш-функциялардың мысалдары ретінде Message Digest алгоритмдер тобына кіретін MD2, MD4 және MD5 хэш-функцияларын, SHA-1, SHA-256, SHA-384, SHA-512, RIPEMD-160, ГОСТ Р34.11-94, т.б. атап кетуге болады.

Цифрлық қолтаңба

Цифрлық қолтаңба – криптографиялық әдістер негізінде авторлықты және шынайылықты сенімді түрде анықтауға мүмкіндік беретін құралдар [1, 2, 23, 34, 37]. Немесе электрондық цифрлық қолтаңба (ЭЦҚ) – электрондық цифрлық қолтаңба құралдарымен жасалған және электрондық құжаттың дұрыстығын, оның тиесілілігін және мазмұнының өзгермейтіндігін растайтын электрондық цифрлық нышандар терімі [22].

Цифрлық қолтаңба куәландырылатын құжат мәтініне, тек куәландырушыға белгілі құпия кілт пен жалпы қол жеткізерлік құпия емес кілтке тәуелді болады. Тек қана құпия кілт құжаттың және цифрлық қолтаңбаның жалған істелмегеніне кепілдік береді. Цифрлық қолтаңба жүйесінің әрбір пайдаланушысы өзінің құпия кілтін, құпияда сақтауды қамсыздандыруға тиісті.

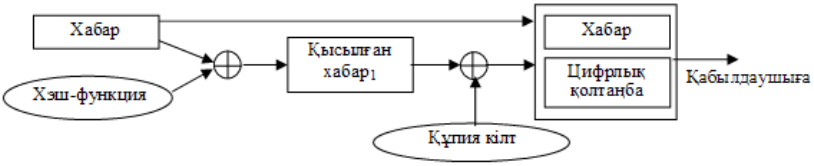
Цифрлық қолтаңбаны жасаған кезде құжат, құпия кілт, жалпы қол жеткізерлік кілт пен цифрлық қолтаңба арасындағы күрделі математикалық тәуелділік пайдаланылады. Электрондық қолтаңбаны жалған жасатпау мүмкіншілігі, оған қажетті математикалық есептеулер көлемінің үлкендігінен туады. ЭЦҚ жүйесінде пайдаланушының құпия кілтін білмей, оның цифрлық қолтаңбасын қолдан (жалған) жасауға мүмкіндік жоқ екендігін атап кету керек.

Электрондық қолтаңба – хабардың бақылау қосындысы мен пайдаланушының жабық кілтін қосу нәтижесінде алынған мән. Ашық кілтті криптожүйе ұзын хабарларды шифрлауға бейімделмегендіктен, хабардың бәрі емес, тек хэш-функциялар арқылы алынған оның қысылған нұсқасы ғана қолтаңбаланады [1, 20, 34, 37].

Әрбір қолтаңбада мынадай ақпарат болады: қол қойылған күні; осы қолтаңба кілтінің жарайтын мерзімінің аяқталу уақыты; файлға қол қойған тұлға туралы мәлімет (аты-жөні, қызметі, жұмыс орнының қысқаша аты); қол қоюшының ұқсастырғышы (ашық кілттің аты); цифрлық қолтаңбаның өзі.

Цифрлық қолтаңба жүйесінің құрамында екі процедура бар: қол қою және қолтаңбаны тексеру процедурасы. Қол қою процедурасында хабар жіберушінің құпия кілті, ал қолтаңбаны тексеру процедурасында –жіберушінің ашық кілті пайдаланылады.

Цифрлық қолтаңбаны қалыптастыру кезінде хабар жіберуші ең алдымен қол қойылатын M мәтіннің h(M) хэш-функциясын есептеп шығарады (1.2-сурет). Есептеп табылған h(M) хэш-функциясы бүкіл M мәтінін сипаттай алатын бір қысқа ғана ақпараттардың m блогынан тұрады. Осы хэш-функция көмегімен қысылған хабар қалыптастырылады. Жіберушінің құпия кілтін қолдана отырып, қысылған хабардан цифрлық қолтаңба алынады. Содан соң жіберілетін хабар цифрлық қолтаңбамен бірге қабылдаушыға жіберіледі.

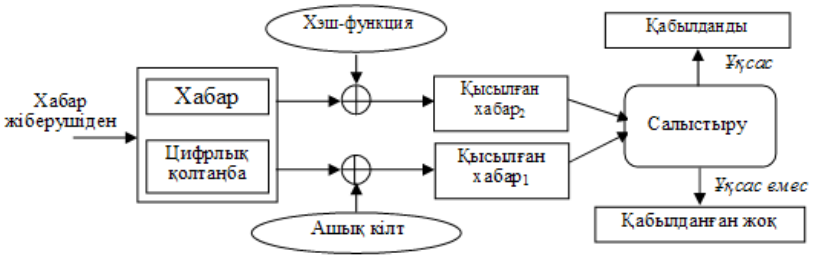


1.2-сурет. Цифрлық қолтаңба қалыптастыру

Цифрлық қолтаңбаны тексеру үшін хабар алушы M мәтінді қабылдаған кезде қайтадан m=h(M) хэш-функциясын есептейді. Содан соң хабар жіберушінің ашық кілті көмегімен хэш-

функцияның есептеп табылған m мәнінің алынған қолтаңбаға сәйкестігі тексеріледі (1.3-сурет).

Сонымен, қабылдаушы жақта электрондық қолтаңбаның түпнұсқалығына көз жеткізу үшін бірнеше операция орындалады. Біріншіден, хабар жіберуші пайдаланған хэш-функция сияқты, функцияның көмегімен қысылған хабар генерацияланады. Одан кейін, қабылданған хабардағы цифрлық қолтаңба мен жіберушінің ашық кілті арқылы, тағы бір қысылған хабар алынады. Оның мақсаты – цифрлық қолтаңба, тиісті жабық кілттің көмегімен жасалған-жасалмағанын тексеру.



1.3-сурет. Цифрлық қолтаңбаны тексеру сұлбасы

Тексерудің келесі кезеңінде қысылған екі хабар салыстырылады. Егер қысылған хабарлар бір-біріне сәйкес келсе, онда цифрлық қолтаңба шынайы деп есептеледі. Бұл – цифрлық қолтаңба жіберушінің жабық кілтімен жасалынғанын растайды. Сонымен қатар, қысылған хабарлардың ұқсастығы, хабардың тасымалдану барысында бұрмаланбағанын да көрсетеді. Сөйтіп, цифрлық қолтаңба деректердің тұтастығын растап, оларға енгізілген кез келген өзгертуді айқындауға мүмкіндік береді.

Цифрлық қолтаңба мысалдары: RSA негізіндегі, Фиат-Шамир, Эль-Гамаль, DSA (DSS), Шнорр, Нибер-Руппел, т.б. Көптеген елдерде электрондық (цифрлық) қолтаңбаның стандарттары қабылданған. Мәселен, ГОСТ Р34.10-94, ГОСТ Р34.10-2001 (Ресей) және FIPS 186 (АҚШ).

Куәландырушы орталық – электрондық цифрлық қолтаңба ашық кілтінің оның жабық кілтіне сәйкестігін куәландыратын, сондай-ақ тіркеу куәлігінің дұрыстығын растайтын заңды тұлға. Куәландырушы орталықтың функциялары [22]:

– электрондық цифрлық қолтаңбаның жабық кілттерін рұқсатсыз қол жеткізуден қорғау үшін шаралар қолдана отырып, электрондық құжат айналымы жүйесіне қатысушылардың өтініші бойынша, электрондық цифрлық қолтаңбаның кілттерін жасайды;

– тіркеу куәліктерін береді, тіркейді, кері қайтарып алады, сақтайды, белгіленген тәртіппен берілген тіркеу куәліктерінің тіркелімін жүргізеді;

– қолданылып жүрген және кері қайтарып алынған тіркеу куәліктерін есепке алуды жүзеге асырады;

– куәландырушы орталық тіркелген электрондық цифрлық қолтаңба, ашық кілтінің тиесілілігін және жарамдылығын растайды, т.б.

Куәландырушы орталық тіркеу куәлігін береді. Ол – электрондық цифрлық қолтаңбаның белгіленген талаптарға сәйкестігін растайтын қағаздағы немесе электрондық құжат. Тіркеу куәлігінде мынадай мәліметтер болады:

- тіркеу куәлігінің нөмірі мен оның қолданылу мерзімі;
- электрондық цифрлық қолтаңбаның иесін бірдейлендіруге мүмкіндік беретін деректер;
- электрондық цифрлық қолтаңбаның ашық кілті;
- электрондық цифрлық қолтаңбаның тиісті жабық кілтін жасау үшін пайдаланылатын электрондық цифрлық қолтаңба құралдары туралы деректер;
- электрондық цифрлық қолтаңбаны қолдану салалары мен оны қолдануды шектеу туралы ақпарат;
- тиісті куәландырушы орталықтың реквизиттері.

Сертификаттар және сертификаттау орталықтары

Цифрлық қолтаңбаны пайдалану, ашық кілттерді сертификаттау мәселесін шешуге мүмкіндік берді.

Асимметриялық криптожүйелердің ашық кілті жайындағы ақпаратты барлық пайдаланушыларға жеткізудің бір амалы – осындай деректерді тексерілген делдалдар (келістірушілер) арқылы тарату. Бұл үшін *сертификаттар беретін қызмет* (Certification Authority – CA) құрылған. Сертификаттау орталығы – өзіңіз толық сенетін үшінші жақ, арнаулы делдал (TTP – trusted third party). Сертификаттау орталығы – абоненттерді тіркеу, ашық кілттердің сертификаттарын жасау, дайындалған сертификаттарды сақтау, қолданыстағы сертификаттар анықтамасын өзекті күйінде ұстап тұру және (жарамсыз) шақырылып алынған сертификаттардың тізімін жариялап тұруға арналған. Көп абоненттері бар желілер үшін бірнеше сертификаттау орталығы құрылады. Мұндай орталықтар бұтақ тәрізді құрылымға біріктіріледі де, оның түбірінде бас сертификаттау орталығы орналасады.

Цифрлық сертификат (ресми куәлік, құжат) деп – ортақ кілттің түпнұсқалығын растау үшін қолданылатын цифрлық қолтаңбасы бар хабарды айтады. Мұндай сертификат – орталықтың цифрлық қолтаңбасымен куәландырылған және құрамына ашық кілт пен қосымша атрибуттардың тізімі кіретін деректер жиыны болып табылады. CA мекемесінде өзінің кілтін тіркеген және аутентификациялаудан өткен кез келген пайдаланушыға сертификат беріледі. Сертификатты пайдалану тек жеке тұлғаларға ғана емес, кез келген қолданбаға немесе басқа объектіге де берілуі мүмкін. Тіркеуден өткен пайдаланушы, оның абоненті болып саналады. CA сертификатты өзінің жабық жеке кілтімен қолтаңбалайды және өзінде сақталатын абоненттер кілттерінің түпнұсқалығына жауап береді. Абонент керек болған жағдайда CA-ға сұрау жіберіп, одан өзіне қажетті абоненттің кілтін ала алады. Сонымен, пайдаланушы (абонент) тек CA-ның ашық кілтін білсе жеткілікті, оған қалған барлық абоненттердің ашық кілтін білудің қажеттігі жоқ.

X.509 сертификатының құрамына әдетте мынадай мәліметтер кіреді [37]: осы сертификат сейкес келетін X.509 стандарт нұсқасының нөмірі, сертификаттың сериялық нөмірі,

сертификат орталығы (CA) алгоритмінің идентификаторы, сертификат жасаушының аты (яғни, CA аты), сертификаттың қолданылып басталу және аяқталу күндері, сертификат субъектісінің (яғни, ашық кілті қолтаңбаланатын тұлғаның) аты-жөні (email мекен-жайы), субъектінің ашық кілті, CA қолтаңбасы.

X.509 стандартының кейбір кемшіліктерін жою үшін **SPKI** (Simple Public Key Infrastructure – ашық кілттердің қарапайым инфрақұрылымы) деп аталатын сертификат әзірленген.

Сертификациялау қызметі жалпы түрде былайша ұйымдастырылған. Екі абонент (мысалы, Алиса мен Боб) белгілі бір сертификат беруші орталықта өздерінің ашық кілттерін тіркейді. Алиса Бобқа (өзінің меншікті жабық кілті көмегімен алынған) цифрлық қолтаңбасы бар хабар жібереді. Боб осы хабарды алысымен, CA-дан Алисаның ашық кілті сақталатын сертификатты талап етеді. Боб CA-дан қол қойылған сертификаты бар жауап алғаннан кейін, ең алдымен CA-ның ашық кілтін қолдана отырып, CA қолтаңбасының түпнұсқалығын тексереді. Содан соң Боб, өзінің сертификатындағы Алисаның ашық кілтінің көмегімен, хабар жіберушінің (яғни, Алисаның) қолтаңбасының түпнұсқалығын анықтайды.

CA-да куәландыру құжаттарынан басқа (жарамсыз), шақырылып алынған сертификаттардың тізімі де (*CRL* – Certificate Revocation List) болады. Біріншіден, әрбір сертификаттың жарамды уақыты шектелген. Екіншіден, егер белгілі бір ашық кілтке сәйкес келетін құпия кілт көпшілікке белгілі болып қалса, онда бұл сертификат мерзімі бітпей-ақ, жарамсыз деп жариялануы мүмкін.

Криптографиялық жүйелер мен шифрлау әдістерінің жіктелуі

Бүгінгі қалыптасқан терминологияға сәйкес криптожүйелер бірнеше түрге бөлінеді [1, 18, 19, 34, 37].

1)♦ Бүгінгі күнге дейін криптографияның ***"әскери"*** (немесе "құпия") және ***"ашық"*** деп бөлінуі сақталған. Бұл оқу

құралында тек ашық криптографияда әзірленген жүйелерге шолу жүргізілген.

2)♦ Беріктілігі неменеге негізделгеніне байланысты криптожүйелер – *қолданылуы шектеулі* және *көпшілік қолданатын* деп екіге бөлінеді.

Қолданылуы шектеулі криптожүйе деп – беріктілігі шифрлау және кері шифрлау алгоритмдерінің сипаттамасын (яғни, алгоритмнің мәнін) құпияда сақтауға негізделген криптографиялық жүйені айтады. Осындай жүйенің тарихи мысалы ретінде, қарапайым ауыстыру шифры саналатын Цезарь шифрын келтіруге болады. Бұл шифрда ашық мәтіннің әрбір символы, әліпбидегі одан кейінгі үшінші тұрған символмен ауыстырылады. Шектеулі алгоритмдер қазіргі заманда шифрлауға қойылатын талаптарға сай келмейтіндіктен, қазіргі уақытта олар қолданылмайды.

Криптографиялық жүйе – егер оның беріктілігі кілттің құпиялығын сақтауға негізделген болса, онда ол ***көпшілік қолданатын криптожүйе*** деп аталады.

3)♦ Кілттерді тарату қағидаты бойынша криптожүйелер – *симметриялық* (немесе құпия кілтті) және *асимметриялық* (немесе ашық кілтті) болып бөлінеді.

Құпия кілтті (симметриялық) криптожүйеде хабарды шифрлау және кері шифрлау, бір-ақ құпия кілтпен жүзеге асырылады. Хабар жіберуші мен қабылдаушы шифрлау мен кері шифрлау үшін қолданатын және тек оларға ғана белгілі ақпарат туралы алдын ала келіседі. Бұл ақпарат (құпия кілт) қасқүнемдерден (қаскөйлерден) құпияда сақталуы тиісті.

Ашық кілтті (асимметриялық) криптожүйеде екі кілт қолданылады: ашық және құпия кілттер (шифрлау кілтін білгендік шифрды ашу кілтін анықтауға жеткіліксіз). Хабар жіберуші ақпаратты ашық (білем деген көпшілікке белгілі) кілтпен шифрлайды. Қабылдаушы өз кезегінде, шифрланған хабарды бастапқы қалпына келтіру үшін (тек хабар алушыға ғана мәлім) құпия кілтті пайдаланады. Ашық кілтті жүйелердің негізгі айырмашылығы – шифрлау кілтінің белгілі болғаны, хабарды кері шифрлауға жеткіліксіз және керісінше.

Асимметриялық шифрлау үлгісі мынадай: *Алиса* деген абонент кілттер жұбын генерациялайды, шифрлау кілтін ашық (керек деушілердің бәріне жария) етеді, ал кері шифрлау кілтін құпия түрде қалдырады. *Боб* деген абонент, *Алисаға* хабар жіберу үшін оны *Алисаның* ашық кілтімен шифрлайды да, байланыс арнасына жібереді. *Алиса* хабарды қабылдап алған соң, оны өзінің құпия кілтімен кері шифрлайды.

Ашық кілтті алгоритмдерді қолдану кезіндегі негізгі қауіп-қатер қаскөйдің ресми ашық кілтті өзінікімен ауыстыру мүмкіндігімен байланысты: бұдан кейін ол жолай ұстап алынған шифрланған хабарларды оқи алады. Осыған ұқсас шабуылдан қорғану үшін, ашық кілттің шындығын растау (сертификаттау) әдістері әзірленген.

Асимметриялық шифрлардың басқа бір кемшілігі – олардың төмен өнімділігі және есептеу қорларына қоятын жоғары талаптары. Бұл мәселенің шешімі – дәстүрлі симметриялық және жаңа асимметриялық криптожүйелерді бірге қолдану: деректер симметриялық алгоритммен шифрланады. Ал шифрлауға арналған кілттер асимметриялық алгоритммен жабылып, сеанс басталар алдында байланыс арнасымен серіктеске жіберіледі.

Ашық кілтті шифрлау алгоритмдері жоғары жылдамдықты арналарда қолданылуы мүмкін емес. Сондықтан, мұндай жүйелердің блоктық шифрлауда қолданылуы тек кілттерді тарату, аутентификациялау және цифрлық қолтаңба қалыптастырумен шектеледі.

XX ғасырдың 80-жылдарының басында *аралас жүйелер* деп аталатын жүйе пайда болды. Мұндай жүйелерде ашық кілтті шифрлау процедуралары тек кілттер мен цифрлық қолтаңбаны тасымалдау үшін қолданылады. Ал тасымалданатын ақпарат DES тәріздес классикалық алгоритммен қорғалады да, оның кілті ашық кілтті шифрлау әдісі арқылы жіберіледі.

Кез келген криптографиялық жүйе – шифрлау алгоритмінен, сондай-ақ түрлі кілттердің, ашық және шифрланған мәтіндердің жиынтығынан тұрады. Криптография - қолданылатын барлық технологиялардың стандарттар мен келісімдердің жиынымен реттеледі.

- Криптожүйенің қандай әдіснаманы қолдауына байланысты шифрлау алгоритмдері екі сыныпқа бөлінеді: симметриялық және асимметриялық [1, 19, 34, 37].

Симметриялық алгоритмдер. Мұнда шифрлау және кері шифрлау үшін бір-ақ құпия кілт қолданылады. Симметриялық алгоритмдердің мысалдары: DES, 3-DES, ГОСТ 28147-89, IDEA, Skipjack, RC4, RC5, RC6, CAST, Blowfish, RIJNDAEL, Mercy, Raiden, RTEA, Camellia-II, т. б.

Мәселен, *DES (Data Encryption Standard)* – симметриялық блоктық шифрлау алгоритмі. Оны IBM фирмасы әзірлеген. 1977 жылы АҚШ ресми стандарт ретінде бекіткен. Алгоритм 64 биттен тұратын блоктарды пайдаланады. 56 биттік кілт қолданылады. DES алгоритмі көптеген бағдарламалық өнімдердің құрамына енгізіліп, кеңінен таралған бірталай блоктық шифрларды әзірлеу кезінде қолданылған. Бұл шифр АҚШ-тың ресми стандарты ретінде 2000-шы жылға дейін қызмет етті.

Triple DES, 3DES. DES алгоритмінің беріктілігін арттыру үшін алгоритмнің жаңа нұсқасы – "үш еселі DES" әзірленген. Бұл криптожүйеде хабар әр түрлі кілттермен 3 рет шифрланады. Осының нәтижесінде кілттің ұзындығы 168 битке дейін ұзарып, шифрлау сенімділігі елеулі жоғарылайды. Бірақ бұл әдіс дағдылы DES әдісінен үш есе баяу істейді. Үш еселі DES шифрлаудың түрлері: DES-EEE3, DES-EDE3, DES-EEE2 және DES-EDE2.

IDEA алгоритмі – халықаралық шифрлау алгоритмі. Кілт ұзындығы 128 бит, ал блок ұзындығы 64 бит. IDEA алгоритмін Джеймс Мэсси (1990 ж.) әзірлеген. Оның дәстүрлі DES алгоритміне қарағанда беріктілігі жоғары және ол PGP шифрлау бағдарламасының негізін құрайды. PGP құрамына IDEA қатар, RSA алгоритмі де кіреді.

Асимметриялық алгоритмдер – деректерді шифрлауға арналған. Бұл алгоритмдер симметриялық сеанстық кілттерді шифрлау үшін, асимметриялық криптожүйелерде қолданылады. Әр түрлі екі кілт қолданылады – біреуі бәріне белгілі, ал екіншісі құпияда сақталады. Шифрлау және кері шифрлау үшін, әдетте осы кілттердің екеуді де қолданылады. Бірақ, кілттердің

біреуімен шифрланған деректерді тек екінші кілттің көмегімен ғана кері шифрлауға болады. Асимметриялық алгоритмдердің мысалдары: Шамир шифры, RSA, ECC, Эль-Гамаль алгоритмі, Рабин криптожүйесі, Мак-Элис шифржүйесі, т. б.

RSA алгоритмін 1978 жылы Р. Райвист (Ronald Rivest), А. Шамир (Adi Shamir) және Л. Адлеман (Leonard Adleman) ұсынған. Шифр беріктілігі кілт ұзындығына тәуелді және егер біз жылдамдыққа мән бермей, жеткілікті мөлшерлі кілттік тізбекті қолданатын болсақ, онда көзделген сенімділікке қол жеткізуге болады. RSA – өте баяу істейтін алгоритм. RSA алгоритмі негізінде ISO/IEC/DIS 9594-8 және X.509 стандарттары жасалған.

Эль Гамаль жүйесі – ашық шифрлау жүйелерінің мысалы. Ол 1985 жылы әзірленген. RSA және Эль Гамаль стандарттарының арасында айтарлықтай айырмашылық жоқ. Олардың тек криптоберіктілік жағынан елеулі өзгешіліктері бар. Эль Гамаль алгоритмі негізінде MD 20899 стандарты жасалған.

ECC (эллипстік қисықтар негізінде жасалған криптожүйе) – асимметриялық шифрлау алгоритмін жүзеге асыруға арналған эллипстік қисық нүктелерінің терминдерінде сипатталған алгебралық жүйені қолданады. Басқа ашық кілтті шифрлау алгоритмдеріне қарағанда бұл криптожүйе, бірдей (барабар) беріктілік кезінде қысқа кілттерді қолданады және оның өнімділігі жоғары.

Шифрлау әдістерін жіктеу үшін шифрлар арасындағы ерекшелікті ажыратуға мүмкіндік беретін бірнеше белгілерді анықтап алу қажет [1, 2, 10, 16, 19].

1. Бастапқы мәтіннің қандай объектілерімен (бит, байт немесе блок) криптографиялық түрлендірулер жүргізілетінін анықтау керек. Мысалы, кейбір шифрларда операция байттармен, ал DES алгоритмінде *блок* деп аталатын биттер жиынымен жүргізіледі. Жүйенің биттер блогын қолдану қасиеті - *блоктылық* деп аталады.

2. Шифрлау функцияларының хабар белгілерінен тәуелділігі – *түйіншектілік* деп аталады. Осындай қасиеті бар жүйелерде қателердің көбеюі байқалады: егер тасымалдау

кезінде ең болмаса бір бит бұрмаланса, онда кері шифрлаудан кейін, мәтінде біраз қателер пайда болады.

3. Хабардың жеке таңбаларын шифрлау, олардың мәтіндегі орнына тәуелді болуы мүмкін. Сондықтан, тасымалдау кезінде шифрмәтіннің кез келген бөлігінің жоғалуы, хабардың барлық келесі бөлімдерінің дұрыс кері шифрланбауына әкеледі. Шифр таңбаларының, олардың бастапқы мәтіндегі орнына тәуелсіздігі – *транзитивтілік* деп аталады.

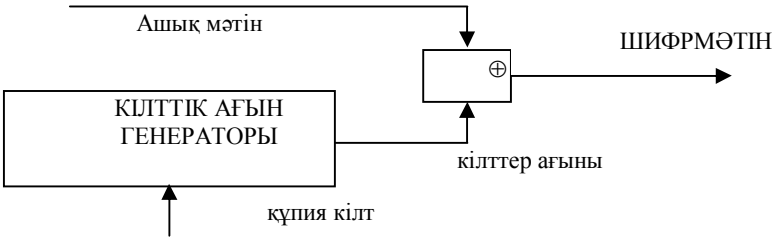
• Ақпарат жабу құралдарын қолдану тәсіліне байланысты шифрлаудың симметриялық алгоритмдері екі түрлі болады: ағындық және блоктық алгоритмдер (немесе шифрлар).

Шифрларды ағындық және блоктық деп бөлу, қолданылып жүрген элементтік базалардың (процессорлардың разрядтылығы, шағынсұлбалардың шапшаңдылығы, компьютер жадысының көлемі сияқты) мүмкіншіліктерін пайдаланатын шифрлайтын түрлендірулердің алгоритмдік және техникалық ерекшеліктерімен байланысты.

Ағындық алгоритмдер (шифрлар) – ашық мәтінді бит-бит бойынша өңдейді. Бастапқы мәтіннің әрбір символы, басқаларынан тәуелсіз түрлендіріледі. Шифрмәтіннің әрбір таңбасы ашық мәтіннің сәйкес таңбасының мәні мен орнының функциясы болып келеді. Таңба ретінде биттер, байттар және сирек, мәтіннің үлкенірек бірлігі алынуы мүмкін. Мұндай шифрлауды, байланыс арнасы арқылы деректер тасымалдаумен қатар жүргізуге болады. Ағындық шифрлауды жүзеге асыру үшін ашық мәтінді таңба-таңба бойынша шифрлауға арналған кілттік тізбектер генераторы қажет (1.4-сурет). Биттердің кездейсоқ ағыны құпия кілт бойынша – *кілттік ағын генераторы* деп аталатын ашық алгоритмнің көмегімен генерацияланады. Шифрмәтіннің биттері мына ереже бойынша қалыптастырылады: $C_i = m_i \oplus k_i$, бұл жерде m_0, m_1, … – М ашық мәтіннің биттері, ал k_0, k_1, … – кілттер ағынының биттері. Кері шифрлау формуласы: $m_i = C_i \oplus k_i$.

Ағындық шифрлардың артықшылықтарына – қателер көбеюінің жоқтығы, жүзеге асырылуының қарапайымдылығы мен шифрлаудың жоғары жылдамдығы жатады. Негізгі

кемшілігі: хабар бастамасының алдында уақытқа үйлесімді ақпарат жіберу қажеттігі (бұл ақпарат кез келген хабарды кері шифрлап бастағанға дейін қабылдануға тиісті).



1.4-сурет. Ағындық шифрлар

Вернам шифры, B152, RC4, SEAL, WAKE, PIKE, GOAL, ORYX, ISAAC, Chameleon, STEN (Ресей), РСЛОС (сызықтық кері байланысы бар жылжыту регистрі) сияқты көптеген ағындық симметриялық алгоритмдер бар.

Мәселен, *RC4 алгоритмі* – айнымалы ұзындықты кілті бар симметриялық ағындық шифр. Оны Рональд Райвист (Ronald Rivest) 1987 жылы әзірлеген. Көптеген ірі компаниялар бұл шифрды өздерінің бағдарламалық өнімдерінің құрамына енгізді. IEEE 802.11 сымсыз желілерінде тасымалданатын ақпаратты шифрлау үшін WEP және WPA стандарттарында қолданыс тапқан.

Блоктық алгоритмдер (шифрлар) кезінде ашық мәтін бірнеше биттен тұратын ұзындығы бірдей блоктарға бөлінеді. Блоктар оларды сондай ұзындықты шифрмәтін блоктарына түрлендіруге арналған (кілтке тәуелді) шифрлау функциясымен өңделеді. Блоктық шифрлаудың негізгі екі түрі бар: орын ауыстыру шифрлары және ауыстыру шифрлары.

Блоктық шифрлар, ашық мәтін блоктарына белгілі бір базалық түрлендірулерді бірнеше қайтара қолдану арқылы жүзеге асырылады. Әдетте, екі түрлі базалық түрлендірулер қолданылады: шифрланатын блоктардың жеке бөлшектерімен жүргізілетін күрделі жергілікті түрлендірулер және шифрланатын блоктардың бөлшектерін өзара орын

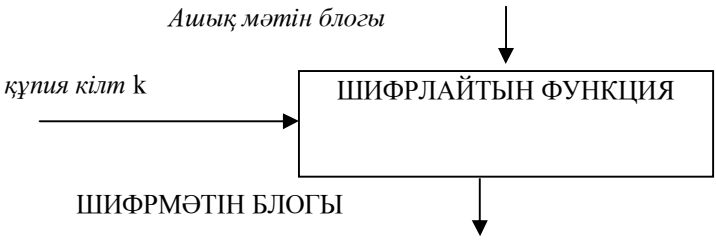
ауыстыратын қарапайым түрлендірулер. Біріншісі – *араластырушы* түрлендірулер, ал екіншісі – *таратып орналастырушы* түрлендірулер деп аталады. Түрлендірулер ашық және шифрланған мәтіндердің статистикалық және аналитикалық қасиеттерінің өзара байланысын, бұрынғы қалпына келтіруді қиындатады. Шифрлау алгоритмі бірнеше циклды (итерация) орындайды. Әрбір циклде түрлендірудің белгілі бір түрі қолданылады.

Базалық түрлендіргіш ретінде жылжыту регистрін қолданған ыңғайлы. Фейстель желісі деп аталатын құрылма да жиі қолданылады.

1.5-суретте блоктық шифрлау алгоритмінің сұлбасы келтірілген. Шифрлау және кері шифрлау мына ережеге сәйкес жүргізіледі:

$$C = E_k(M) \quad \text{және} \quad M = D_k(C) = E_k^{-1}(C);$$

мұндағы M – ашық мәтін блогы, k – құпия кілт, E – шифрлайтын функция, D – кері шифрлайтын функция, C – шифрланған мәтін (шифрмәтін) блогы, E^{-1} – кері шифрлайтын түрлендіру.



1.5-сурет. Блоктық шифрлар

Блоктық шифрлаудың негізгі қасиеті: шифрмәтін блогының әрбір биті, ашық мәтіннің сәйкес блогының барлық биттерінің функциясы болып келеді.

Қарапайым блоктық шифрлаудың негізгі артықшылығы: жақсы құрылмаланған жүйеде ашық мәтіннің немесе кілттің кішігірім өзгерістері шифр мәтінінде үлкен және алдын ала болжанбайтын өзгертулерге әкеледі. Бірақ, қарапайым блоктық

шифрлардың біраз кемшіліктері де бар. Біріншісі – егер барлық блоктарға бәріне бірдей кілт қолданса, онда тіпті блоктың ұзындығы үлкен болса да оған "сөздікті" криптоанализ қолданылуы мүмкін. Блоктық шифрлардың келесі кемшілігі – блок ішіндегі қателердің көбеюіне байланысты. Шифркұжаттың қабылданған блогындағы бір биттің өзгеруі, барлық блоктың дұрыс кері шифрланбауына әкеледі.

Бұл шифрлар олардың жоғары криптоберіктілігінің арқасында іс жүзінде жиі пайдаланылады. Блоктық шифрлаудың қазіргі компьютерлік алгоритмдерінде блок ұзындығы әдетте 64, 128 битке тең. Блоктық симметриялық алгоритмдердің мысалы ретінде DES, ГОСТ 28147-89, IDEA, RIJNDAEL (AES), RC2, RC5, RC6, SkipJack, Blowfish, MISTY1, Raiden, RTEA, Camellia-II, т.б. алгоритмдерді атап кетуге болады.

Мәселен, *RC6 алгоритмі* – 1998 жылы Рональд Райвист (Ronald Rivest) ұсынған блоктық шифр. Шифрдың нақты нұсқасы RC6-w/r/l деп белгіленеді. Бұл жерде: w – сөз мөлшері (16, 32 немесе 64 бит), r – раундтар саны, l – кілт ұзындығы (0-256 байт аралығында). Мысалы, RC6-32/20/16. Блок әр уақытта 4 сөзден тұрады.

ГОСТ 28147-89 – 1989 жылы Кеңес Одағының стандарты ретінде қабылданған, ал 90-жылдардың басынан бастап, Ресей стандарты болып саналады. Бұл алгоритм DES алгоритмі сияқты – классикалық блоктық шифр. Бірақ біраз айырмашылықтары да бар. Мәселен, кілттің ұзындығы 256 бит, блоктың мөлшері 64 бит, алгоритмді аппараттық та, бағдарламалық түрде де жүзеге асыру көзделген. Сондай-ақ, жұмыс істеу жылдамдығы да американдық стандарттан жоғарырақ.

RIJNDAEL алгоритмі. Rijndael алгоритмі АҚШ-та 2000 жылы ең жақсы криптоалгоритмге жарияланған конкурстың жеңімпазы болып шықты. 2001 жылы Rijndael алгоритмі АҚШ-тың жаңа шифрлау стандарты (*AES*) деп жарияланды. Авторлары – бельгиялық Йоан Дамен (Joan Daemen) мен Винсент Раймен (Vincent Rijmen) деген екі ғалым. Бұл стандарт ескірген DES стандартының орнын баспақшы. Rijndael алгоритмі Square блоктық шифрының бір варианты болып

табылады. Блоктың және кілттің ұзындығы 128, 192, 256 бит. AES ретінде блоктың ұзындығы 128 битке тең варианты ғана қабылданды.

Ағындық және блоктық шифрлардың ерекшеліктері [37]:

– блоктық шифр жалпылау болып табылады (ол ағындық шифрға жеңіл түрлендіріледі).

– ағындық шифрдың құрылымы біраз математикаландырылған. Бұл бір жағынан, шифрды бұзып-ашу мүмкіндігін ұлғайтады, ал екінші жағынан, шифрдың беріктілігін дәл бағалауды жеңілдетеді.

– ағындық шифрлар – бағдарламалық қамтамасыздандыру жағынан қарағанда онша ыңғайлы емес. Бірақ, оларды аппараттық түрде жүзеге асыру өте тиімді болып келеді.

– блоктық шифрлар – бағдарламалық құралдар үшін де, аппараттық құралдар үшін де ыңғайлы, бірақ ағындық шифрлармен салыстырғандағы өңдеу жылдамдығы төмен.

Ағындық және блоктық шифрлаудың әрқайсысының жақсы қасиеттерін қолдана отырып, олардың араласқан жүйелерін құруға болады.

- Барлық белгілі шифрлау тәсілдерін мынадай топтарға бөлуге болады: ауыстыру, орын ауыстыру, аналитикалық түрлендіру, гаммалау, т.б. Бұл тәсілдердің әрқайсысы бірнеше түрлі бола алады [1, 10, 37].

Ауыстыру шифрында – әліпбидің әрбір әрпіне белгілі бір әріп, цифр, символ немесе олардың қисындасуы сәйкес келеді. Сөйтіп, шифрқұжатта әріптердің орны өзгеріссіз қалады (ашық мәтіндегі сияқты), бірақ символдары ауыстырылады. Олардың қарапайым, әрі күрделі түрлері бар. Қарапайым ауыстыру шифрларына мысал: Полибий квадраты, Цезарь шифры, Трисемус шифры, Кардано торы, Плейфер шифры, т.б. Күрделі ауыстыру шифрларының мысалы ретінде Вижинер квадратын, бір реттік шифрлауыш жүйесін, Гронсфельд шифрын, Уитстонның “қос квадрат” шифрын, Вернам әдісін, т.б. келтіруге болады.

Орын ауыстыру шифрында – хабардың әріптері (қандай да болмасын бір тәсілді қолдана отырып) өзара орын

ауыстырылады. Бұл кезде ашық мәтіннің барлық әріптері өзгеріссіз қалады, тек олардың ашық мәтіндегі орындары ғана ауыстырылады. Мысалы: шифрлайтын кестелер, сиқырлы квадраттар, т.б. Анаграмма (дыбыстардың орнын ауыстыру арқылы сөздердің мағынасын өзгерту) – орын ауыстыру шифрына жатады.

Гаммалау арқылы шифрлау – шифрланатын мәтіннің символдары, шифр гаммасы деп аталатын кейбір кездейсоқ тізбек символдарымен қосылады. ЭВМ көмегімен шексіз шифр гаммасын жасауға болатындықтан, шифрлаудың бұл түрі автоматтандырылған жүйелерде ақпаратты шифрлайтын негізгі әдістердің бірі болып табылады.

Аналитикалық түрлендіру бойынша шифрлау – шифрланатын мәтін, қайсыбір аналитикалық ереже (формула) бойынша түрлендіріледі. Мысалы, векторды матрицаға көбейту ережесін қолдануға болады. Көбейтілетін матрица – шифрлау кілті болып табылады (сондықтан оның көлемі мен мазмұны құпия күйінде сақталуы керек), ал көбейтілетін вектордың символдары – шифрланатын мәтіннің символдары бір ізділікпен атқарылады. Мысалдың басқа түріне, ашық кілтті криптожүйелерді құратын бір бағыттық функциялар кіреді.

- **Кілттердің ашық таратылуы.** Құпия байланыс ұйымдастыру мәселелерінің біреуі – абоненттер арасында кілттерді тасымалдау. 1976 жылы Уитфилд Диффи (Whitfield Diffie) мен Мартин Хеллман (Martin Hellman) кілттерді ашық тарату хаттамасын ұсынған (***Диффи-Хеллман алгоритмі)***. Бұл хаттамада байланыс орнатпақшы пайдаланушылар жұбының әрбір абоненті бір-бірінен тәуелсіз, өзіндік кездейсоқ санды генерациялайды. Оны белгілі бір процедура арқылы түрлендіреді, байланыс ашық арна бойымен түрлендірілген сандарды өзара айырбастайды және байланыс барысында қарсы жақтан алынған ақпарат негізінде ортақ құпия кілт есептеледі. Әрбір осындай кілт, тек бір сеанс бойы немесе тіпті оның бөлігіне ғана жарамды.

Сонымен, кілттердің ашық таратылуы – пайдаланушылардың әрбір жұбына ортақ құпия кілтті өздері

қалыптастырып алуға мүмкіндік береді. Сөйтіп, құпия кілттерді тарату процедурасын оңайлатады.

- Криптографиялық жүйелер мен шифрлау әдістерінің жіктелуін қарастырып шыққаннан кейін, енді криптожүйені таңдау және криптографиялық алгоритмдер мен стандарттарды іске асыру мәселеріне қысқаша тоқталайық.

Криптографиялық жүйені таңдауға әсер етуші сипаттамаларға, ең алдымен шифрлау жылдамдығына қойылатын талап жатады. Мәселен, шифрлау жылдамдығына жоғары талаптар қойылмаған жағдайда, RSA криптографиялық жүйесін қолдануға болады. Блоктық шифрлау жүйелері бағдарламалық түрде жүзеге асырылған кезде төмен жылдамдықтылар қатарына жатады. Бірақ, аппараттық нұсқасы (мысалы, DES алгоритмімен) кез келген жылдамдықтармен жұмыс істей алады. Егер өте жоғары жылдамдықтар қажет болса, онда бағдарламалық та, аппараттық та түрде шапшаңдылығы жоғары болып келетін ағындық шифрлау жүйесін қолданған жөн [16, 37].

Аппараттық шифрлауыштарға, жоғары жылдамдық пен сенімділік тән. Олар электрондық тақшалар түрінде орындалады. Шифрлауыштардан басқа, жеке құрылғылар ретінде жиі кілттік ақпарат қоймалары да орындалады, оларға – электрондық кілттер, пластикалық электрондық карталар, т.б. жатады.

Бағдарламалық түрде жүзеге асырылу, аппараттықпен салыстырғанда икемді және үнемділеу. Криптографиялық алгоритмдерді қолданатын барлық бағдарламалар екіге бөлінеді: мамандандырылған дестелер (негізгі мақсаты – құпия ақпараттың жасырындылығын қамтамасыз ету) және шифрлау функциясы, көрсетілетін қызметтер жиынының тек біреуі болып келетін бағдарламалар. Криптографиялық функциялар көптеген операциялық жүйелердің құрамына кіреді.

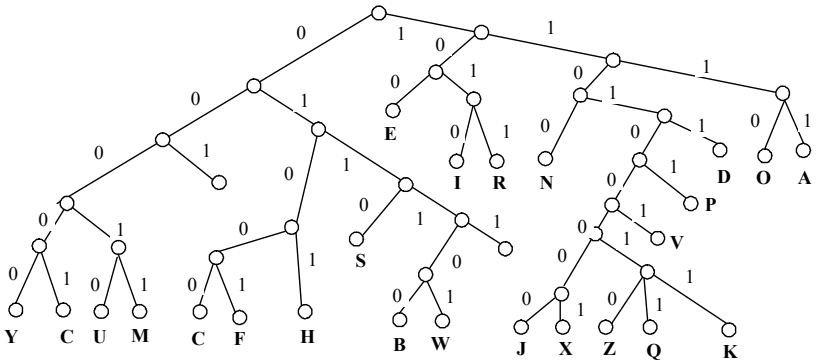
4)♦ **Кодалар** – дипломатияда, әскери салада, т.б. жерлерде ақпарат тасымалдау кезінде қолданылатын шартты белгілердің немесе аттардың жүйесі [16]. Тасымалдау сапасын жоғарылату үшін кодтау жиі пайдаланылады. Байланыс арналарымен хабар тасымалдау немесе ЭЕМ жадында деректер сақтау кезінде қателерді түзетуші кодтар қолданылады. Кодтардың басқа бір

түрі деректер және дискілерді қысу құралдарында (мәселен, ARJ, ICE, ZIP сияқты бағдарламаларда) кеңінен қолданылады. Мұндай кодтарды қолдану, құпиялықпен байланысты емес.

Ақпаратты криптографиялық жабу құралдарының бірі – кодтау болып саналады. Кодтау деп – жабылатын деректердің элементтерін цифрлық, әріптік немесе олардың қисындасқан тіркестерімен ауыстыруды айтады. Ақпаратты кодалау және оның ауыстыру әдістерімен шифрлау арасында біраз ұқсастық бар. Сондай-ақ, бұл әдістердің арасындағы ерекшеліктерді де табуға болады.

Бұл әдісте жиі кездесетін символдарды жазу үшін қысқа екілік кодтар қолданылады, ал сирек кездесетіндерді жазу үшін ұзын кодтар пайдаланылады.

Мәтіндік файлдарда басқаларға қарағанда ***Хоффман кодтауы*** жиірек қолданылады (1.6-сурет). Бұл кезде мәтін символдары әр түрлі ұзындықты биттер тізбегімен ауыстырылады. Әліпби әріптеріне арналған екілік код графтың түбірінен бастап, осы белгілі бір әріпке сәйкес келетін тармақтың соңына дейінгі бағдарғы бойына 0 мен 1 тізбегін жазу ұйымдастырылады. Сонда Е әрпі 001 кодымен, D - 11011, В - 001110 т.с.с кодталады.



1.6-сурет. Хоффман кодасы

Егер кодтау графы құпияда сақталса, онда мұндай кодалаудың криптоберіктілігі, қарапайым ауыстыру арқылы шифрлау деңгейінде болады.

Кодалар жиі шифрларға ұқсағанмен, криптологтар оларды шифрлар және кодалар деп ажыратады (себебі іс жүзінде бұлар - әр түрлі жүйелер). Кодтар, *сөздерді қарапайым ауыстыру* шифры ретінде қаралады. Әдетте кодтық кестелер, әрбір сөзіне кодалық эквивалент берілген сөздіктен тұрады. Құпиялыққа жету үшін кодаларды шифрлау керек.

Мәселен, "Көл едеуір лайланыпты" деген хабар ерекше кода болып табылады. Егер сөздердің тек бірінші әріптерін ғана оқысақ, онда "КЕЛ" жасырын хабары шығады. Осындай кодалау – ***акрокода*** деп аталады (грек тілінде, Akro – шеткі, яғни сөздердің немесе қатарлардың бірінші әріптері).

Кодалар хабардың мазмұнын жасыру үшін де қолданылуы мүмкін. Мәселен, 1936 жылы таратылған "барлық Испания төбесінде ашық аспан" деген хабар азамат соғысы басталуының кодалық сигналы болған екен [16].

5)♦ **Кванттық криптография** – оптикалық арналармен құпия ақпарат тасымалдау тәсілі. Бұл тәсілде деректер жеке фотондар арқылы кодаланады. Кванттық криптографиялық жүйелердің құпиялығы, кванттық физика заңдарын классикалық криптография процедураларымен қатар қолдануға негізделген [7].

Кванттық үдерістерді зерттеудегі айтулы жетістіктердің бірі – кванттық криптография болып табылады. Ч. Беннет (IBM фирмасы) пен Ж. Брэссард (Монреаль университеті) өз зерттеулері барысында мынадай шешімге келген: фотонды (жарық квантын) тек ақпарат сақтау үшін ғана емес, оны тасымалдау үшін де пайдалануға және оның негізінде, құпия кілттерді ашықтан-ашық таратуға арналған кванттық арна әзірлеуге болады. Криптографияда байланыс желісі әрқашан қаскөйдің бақылауында болады. Бірақ, ақпарат фотонның бейортогональды күй-жайларымен кодаланса, онда қаскөй осы үдерістің тұтастығын бұзбай, ондағы мәліметті ала алмайды (ал тұтастық бұзылса, ол лезде белгілі болып қалады).

Осындай технология, дәстүрлі шифрлау алгоритмдеріндегі құпия кілттерді тасымалдау үшін өте ыңғайлы. Таратушы жақ кілтті генерациялап, оны кванттық арна арқылы қабылдаушы жаққа жібереді. Егер тасымалдау барысында үшінші жақтың қол сұғу фактісі анықталса, онда жіберілген кілт жойылып, жаңа кілт генерацияланады да, қайтадан арна арқылы қабылдаушы жаққа жіберіледі. Осы әрекеттер, тасымалдаудың жасырындылығына екі жақтың да көзі жеткенше қайталанады.

Әлемнің көптеген елдерінің зертханаларында кванттық арналар жасалып, сынақтан өткізіліп жатыр, ал АҚШ-та кванттық криптография қағидаты бойынша жұмыс істейтін бірінші оптикалық байланыс арнасы пайдалануға берілген. Осы технологиямен байланысты осы уақытқа дейін шешілмеген бірнеше мәселелер бар. Олардың бірі – деректер тасымалдау жылдамдығының төмендігі.

Кванттық криптографиялық жүйелерді қолдану, қазіргі қолданыстағы криптографиялық жүйелермен салыстырғанда, қауіпсіздікті елеулі жоғарылатуға мүмкіндік береді.

Жалғанкездейсоқ сандардың генераторлары

Криптографиялық түрлендірулердің негізін, құпия кілттер құрайды. Шифрлайтын жүйенің беріктілігі тек кілттің құпиялығымен анықталады. Классикалық криптографияның негізгі мәселесі – ұзақ уақыт бойы қысқа кездейсоқ кілттің көмегімен алдын ала болжаланбайтын, үлкен ұзындықты екілік тізбектерді генерациялау қиындығында. Бұл мәселені шешу үшін екілік жалғанкездейсоқтық (pseudo-random) тізбектердің генераторлары кең қолданылады [6, 14, 16, 27, 34].

Кілттен бағдарлама көмегімен алынған сандардың кездейсоқ немесе жалғанкездейсоқ қатарлары – *гамма* (γ – грек әліпбиінің әрпі) деп аталады. Нағыз кездейсоқ қатарларын алу және көбейту өте күрделі, әрі қымбат. Кездейсоқтықты физикалық құбылыстар (радиоактивтік сәулелену немесе электрондық шамшықтардағы бытыралық шуыл) арқылы физикалық үлгілеу, нағыз кездейсоқ үдерістер бермейді.

Сондықтан, гамманы генерациялау үшін физикалық үдерістердің орнына, ЭЕМ-ға арналған бағдарламалар қолданылады. Кездейсоқ сандардың генераторлары деп аталса да, бірақ шын мәнінде олар тек өз қасиеттері бойынша ғана кездейсоқ болып көрінетін детерминалдық сандар қатарларын береді.

Жалғанкездейсоқ тізбектің немесе гамманың криптографиялық берік генераторына мынадай негізгі талаптар қойылады:

– гамманың периоды әр түрлі ұзындығы бар хабарды шифрлау үшін жеткілікті үлкен болуға тиісті;

– гамма алдын ала қиын болжамдалатындай болуы тиісті. Ол мынаны білдіреді: егер криптоаналитикке гамманың қайсыбір бөлігі белгілі болса да, ол оның алдындағы немесе одан кейінгі биттерді анықтай алмайды;

– гамманы генерациялау – үлкен техникалық және ұйымдастырушылық қиындықтар тудырмауы тиіс;

– қалыптастырылатын тізбектің периоды үлкен болуы керек.

Жалғанкездейсоқтық сандар генераторының ең маңызды сипаттамасы – *периодтың ақпараттық ұзындығы*. Бұдан кейін сандар қайталана бастайды немесе оларды алдын ала болжауға болады. Бұл ұзындық жүйе кілттерінің мүмкін болатын санын нақты анықтайды да, жалғанкездейсоқтық сандар алу алгоритмінен тәуелді болады. Периодтың қажетті ұзындығы, деректердің құпиялық дәрежесімен анықталады. Неғұрлым кілт ұзын болса, солғұрлым оны таңдап, іріктеп алу қиын.

Гамманы кездейсоқ деп есептеу үшін, ең кемінде оның периоды өте үлкен болу керек. Ал белгілі бір ұзындықты биттердің әр түрлі қисындасулары, оның бойында біркелкі орналасқан болуға тиісті. Сонымен, қатарға қойылатын екінші талап – оның қасиеттерінің нағыз кездейсоқ іріктемеге ұқсауының статистикалық түрде расталғандығы.

Гамма тізбектерінің криптографиялық беріктілігін тексеру үшін криптоанализдің әр түрлі әдістері қолданылады. Осы жағдайда, гамманы ашу – криптоанализ әдісімен белгілі ашық

мәтін бар кезде шифрды ашумен парапар болып саналады. Осындай әдістердің бірі – Зигентхальтер ұсынған, *гамманың корреляциялық қасиеттерін талдау әдісі*. Оның талдауы гаммадағы жеке тізбектерді (барлығын толық қарастырып шығу арқылы) анықтайды.

Зенг, Янг және Рао ұсынған *алгебралық әдіс,* гамма генераторларының жасырын сызықтығын пайдаланады. Ол, кездейсоқ коэффициенттері бар сызықты алгебралық теңдеулер жүйесінің қайшылықсыздығын дәл бағалау негізінде құрылған. Бұл әдіс гамма кесіндісі бойынша ортақ кілттен ішкілт (subkey) бөліп шығаруға және матрица коэффициенттері тек ішкілттен ғана тәуелді болатын, сызықты теңдеулер жүйесін құруға тырысады. Егер ішкілт дұрыс бөлінген болса, онда осы теңдеулер жүйесі үлкен ықтималдықпен, қайшылықсыздық талабын қанағаттандырады. Бұдан кейін тізбектің қалған бөліктері бойынша барлық кілтті табуға болады.

Көптеген гамма генераторлары бейсызықты логикалық функцияларды қолданумен, екі немесе одан көп генераторлардың қисындасуы негізінде жасалған. Сызықтық кері байланысы бар екі ығыстыру регистрінің қисындасу тәсіл-дерінің ең бір қарапайымы – ауыстырып қосылатын разрядтарының қатынасы 2:1 болатын ауыстырып-қосқышты қолдану. Бұл тәсіл – *Джеффи генераторы* деп аталып жүр. Генератордың осал жері – мұндай жүйе сызықтық синдромды қолданатын криптоанализ әдісімен жеңіл ашылуы мүмкін. Сызықтық кері байланысы бар ығыстыру, регистрлер техникасының қазіргі күй-жағдайында шықпалық тізбектерді (гамманың жолай ұстап алынған ұзындығы мәтіннің бір қатарына тең сегменті бойынша) ашуға болады. Осыған ұқсас генератордың тағы біреуі (*Дженнинг генераторы*) сызықтық кері байланысы бар екі регистрді біріктіруге арналған ауыстырып-қосқышты пайдаланады.

Жалғанкездейсоқ сандарды ЭЕМ-де генерациялаудың [6, 14, 27, 34] ең алғашқы тәсілін 1946 жылы Джон фон Нейман ұсынған. Бұл тәсіл бойынша: әрбір келесі кездейсоқ сан, алдыңғы санды екілік дәрежесіне шығарып (квадраттап), кіші

және үлкен разрядтағы цифрларды алып тастау арқылы пайда болады. Алайда бұл тәсіл сенімсіздеу болды да, кейін одан бас тартты.

Жалғанкездейсоқ бүтін сандар тізбегін генерациялайтын белгілі процедуралар ішінде ең жиі қолданылатыны - *сызықты конгруэнтті генератор*. Бұл генератор Y_1, Y_2, …, Y_{i-1}, Y_i, … жалғанкездейсоқ сандар тізбегін келесі қатынасты пайдалана отырып құрастырады: $Y_i = (a*Y_{i-1} + b) \bmod m$, мұнда Y_i – тізбектің і-ші (ағымдағы) саны; Y_{i-1} – тізбектің алдыңғы саны; **a, b** және **m** – тұрақтылар; **m** – модуль; **a** – коэффицент, **b** – өсімше; Y_1 – тудырушы сан (алғашқы мәні). **m** модулінің мәні 2^n тең деп немесе қарапайым санға (мысалы $m=2^{31}-1$) тең деп алынады. **b** өсімі **m** санымен өзара қарапайым, ал **a** коэффициенті – тақ сан болуы керек. Ағымдағы Y_i жалғанкездейсоқ санын табу үшін, алдыңғы Y_{i-1} санын **a** коэффициентіне көбейтіп, оған b өсімшесін қосады, одан кейін қосындыны m модуліне бөліп, нәтиже ретінде қалдық алынады. Бұл теңдеу таңдап алынған a, b, m параметрлеріне байланысты және m мәніне жете алатын қайталау периодымен жалғанкездейсоқ сандарды генерациялайды.

Жалғанкездейсоқ сандар тізбектерін генерациялайтын, сызықты рекурентті қатынастарға негізделген тәсіл бар. Рекурентті қатынастар мен олардың айырым тендеуін қарастыралық,

$$\sum_{j=0}^{k} h_j a_{i+j} = 0, \qquad a_{i+k} = -\sum_{\mathbf{j=0}}^{\mathbf{k-1}} h_j a_{i+j},$$

мұндағы $h_0 \neq 0$, $h_k = 1$, әрбір h_i GF(q) өрісіне жатады.

Бұл теңдеулер шешімі GF(q) өрісіне жататын a_0, a_1, a_2, … элементтер тізбегі болып табылады. a_0, a_1, a_2, … a_{k-1} мәндері бойынша a_k мәні анықталады. a_0, a_1, a_2, … a_k - мәндері бойынша a_{k+1} мәні анықталады, т.с.с. Белгілі болған a_0, a_1, a_2, … a_{k-1} мәндері бойынша шексіз тізбектер құруға болады, яғни тізбектің әр келесі мүшесі, алдыңғы k-мүшелер көмегімен анықталады.

Тізбектердің мұндай түрі компьютерде оңай іске асырылады. Сонымен қатар, егер барлық h_i және a_i мәндері GF(2) өрісінде тек 0 және 1-ге тең болса, онда оны іске асыру тіпті оңай.

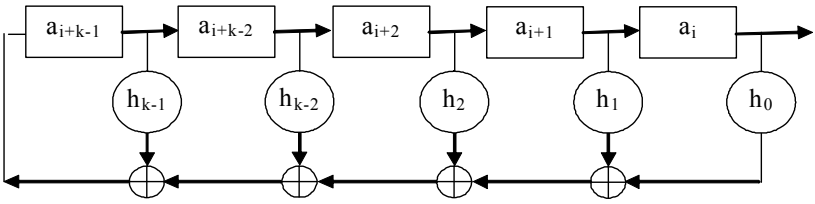
Сызықтық кері байланысы бар ығысу регистрлері негізінде жасалған генераторлардың (LFSR – Linear Feedback Shift Register) криптобериктілігі жоғары болады. Екілік LFSR құру үшін мынадай көпмүше қолданылады:

$$\mathbf{h(X)=\sum_{j=0}^{k} h_j X^j};$$

мұндағы X – айнымалы, h_j – X^j коэффициенті (мәні 0 немесе 1) және n – (X^n-1)-деген көпмүше, h(X) мәніне бөліне алатын, ең аз мөлшерлі бүтін оң сан.

Сонымен қатар, $h(X)=\sum h_j X^j$ көпмүшесінің түрі ығысу регистрлері негізінде жасалған генераторда кері байланыстың пішінүйлесімін h_j анықтайды. Басқаша айтқанда, егер h(X) көпмүшесінің коэффициенті $h_j=1$ болса, онда генератор сұлбасында h_j кері байланысы бар, ал егер h(X) көпмүшесінің коэффициенті $h_j=0$ болса, онда генераторда h_j кері байланысы жоқ.

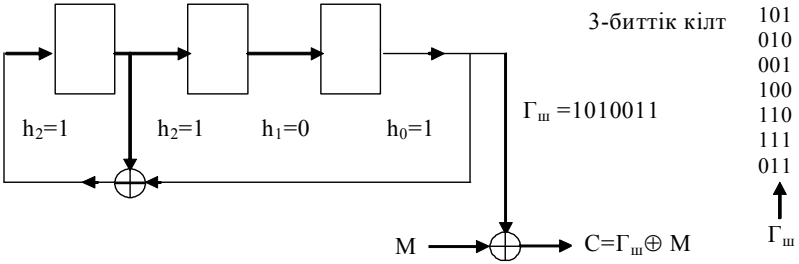
1.7-суретте ауыстырып-қосу сұлбасы көрсетілген [27]. Бұл суретте мынадай шартты белгілер пайдаланылған: бірінші қатардағы төртбұрыштар – a элементтерін сақтайтын ұяшықтар (ұяшықтың шықпасында a=0 немесе a=1); екінші қатардағы дөңгелектер – h беріліс коэффициенті бар кері байланыстар (h=0 немесе h=1); үшінші қатарда – екі модулі бойынша қосындылауыштар.



1.7-сурет. Ығысу регистрлері негізінде жасалған генератор

Егер құрастырушы көпмүше қарапайым болса, онда 2^m-1 және m сандары өзара қарапайым болған кезде ғана m разрядты генератор ең үлкен 2^m-1 периодқа ие болады.

Мысал ретінде $h(X)=x^3 + x^2 +1$ қарапайым көпмүшеге сәйкес құрылған сызықтық кері байланысы бар үш разрядты ығысу регистрін қарастыралық (1.8-сурет). Мұндағы коэффициенттер мәні: $h_3=1$, $h_1=0$, $h_0=1$.



1.8-сурет. Кері байланысы бар үш разрядты ығысу регистрі ($Г_ш$ шифр гаммасының генераторы)

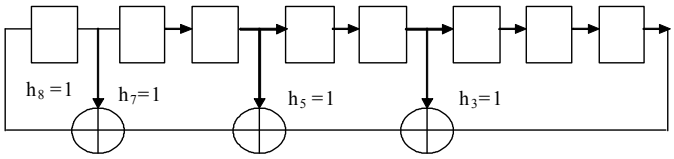
Кілттің мәні – 101 болсын. Регистр осы күйден жұмыс істей бастайды. Регистрдің күй-жайы суретте көрсетілген. Регистр өзінің барлық 7 нөл емес күйі арқылы өтіп, қайтадан өзінің 101 қалпына келеді. Бұл – сызықтық кері байланысы бар регистрдің ең ұзын периоды. Мұндай тізбек, ығысу регистрінің ең үлкен тізбегі (MLSRS – Maximal Lenght Shift Register Sequence) деп аталады.

Питерсон мен Уэлдон [27] зерттеуі бойынша кез келген m бүтін саны үшін периоды (2^m-1)-ге тең m биттік MLSRS тізбегі болады. Мысалы, егер m=100 болса, онда тізбектің периоды 2^{100}-1 болады да, оны 1Мбит/с жылдамдықты байланыс жолдарымен жібергенде ол 10^{16} жыл өткенше қайталанбайды.

Мысалдағы кері байланысы бар ығысу регистрінің шықпа тізбегі $Г_ш$ (шифр гаммасы) 1010011 тізбегі болып табылады. Ол циклды түрде қайталанып отырады. Бұл тізбекте 4 бірлік пен 3 нөл бар. Егер тізбектелген биттердің жұптарын талдап қарайтын болсақ, онда 10 және 01 жұптары 2 реттен, ал 00 және 11

жұптары – 1 реттен кездеседі екен. Ең көп ұзындығы бар тізбектің біркелкі таралу-орналасуға жақын болуы, оларды жалғанкездейсоқ тізбектер ретінде криптографиялық жүйелерде пайдалануға болатынын көрсетеді.

Екілік LFSR құру үшін қажетті бастапқы ақпарат – құрастырушы көпмүше. Бұл көпмүшенің дәрежесі ығысу регистрінің разрядтылығын, ал нөл емес коэффициенттері – кері байланыстың сипаттамасын анықтайды [6, 14]. 1.9-суретте $\Phi(x) = x^8 + x^7 + x^5 + x^3 + 1$ көпмүшесі негізінде құрылған Галуа генераторы және оның күй-жайының диаграммасы келтірілген.



1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1
1 1 0 1 0 1 0 0
0 1 1 0 1 0 1 0
0 0 1 1 0 1 0 1
. . .
0 0 1 0 1 0 0 1

1.9-сурет. Галуа генераторы және оның күйлерінің диаграммасы

1.3. Сандар теориясы элементтерінен қысқаша мәлімет

Модулярлық арифметикадағы кәдімгі $a \equiv b \pmod n$ жазба былайша оқылады: “a саны b санымен n модулі бойынша салыстырымды”. Бұл қатынас a, b және $n \neq 0$ сандарының бүтін мәндері үшін мына жағдайда ғана дұрыс болады [31]: $a = b + k*n$, мұндағы k – кез келген бүтін сан.

Егер $a \equiv b \pmod n$ болса, онда b санын, a санының n модулі бойынша ***қалындысы*** деп атайды.

a санының n модулі бойынша қалдығын табу $a \pmod n$ операциясын, a санын n модулі бойынша *келтіру* немесе *модуль бойынша келтіру* деп атайды.

Мәселен, $17 \bmod 12 = 5$ немесе $17 \equiv 5 \pmod{12}$. Бұл жерде, 5 саны 17 санының 12 модулі бойынша қалдығы болып табылады.

0-ден (n-1)-ге дейінгі бүтін сандар жиыны n модулі бойынша, *қалдықтардың толық жиыны* деп аталады.

Мысалы n=12 үшін қалдықтардың толық жиыны мынадай болады:

$$\{0, 1, 2, \ldots, 11\}.$$

Әдетте, $r \in \{0, 1,2, \ldots, n-1\}$ қалындыларды пайдалану ыңғайлы деп саналады.

Модулярлық арифметика әдеттегі арифметикаға ұқсас келеді: коммутативтік, ассоциативтік, дистрибутивтік заңдары сақталады. Сандарды ең алдымен n модулі бойынша келтіріп, содан соң операцияларды орындауға немесе алдымен операцияларды орындап, содан соң n модулі бойынша келтіруді орындауға болады. Мәселен:

$$(a + b) \bmod n = [a(\bmod n) + b(\bmod n)] \bmod n,$$
$$(a – b) \bmod n = [a(\bmod n) – b(\bmod n)] \bmod n,$$
$$(a * b) \bmod n = [a(\bmod n) * b(\bmod n)] \bmod n,$$
$$[a * (b + c)] \bmod n = \{[a* b(\bmod n)] + [a * c(\bmod n)]\} \bmod n.$$

Криптография n модулі бойынша есептелетін көптеген есептеулерді қолданады. Өйткені, дискреттік логарифмдер мен квадрат түбірлерін есептеулер өте күрделі болып келеді. Модулі бойынша есептеулерді қолданудың тағы бір артықшылығы – олар барлық аралық есептелетін шамалар мен нәтижелер диапазонын шектеуге мүмкіндік береді.

Ұзындығы k бит болатын n модульдері үшін қосудың, азайтудың немесе көбейтудің аралық нәтижелерінің ұзындығы 2k биттен артық болмайды. Сондықтан, модулярлық арифметикада дәрежелеу амалын қиындықсыз орындауға болады.

n модулі бойынша a санының дәрежесін, яғни $a^x \ mod \ n$, есептеу үшін бірнеше көбейту және бөлу амалдарын орындауға болады. Оларды тез орындаудың бірнеше тәсілдері бар.

Мәселен, a^8 mod n есептеуді орындау керек болса, онда ол үшін жеті рет көбейту нәтижесінде алынған үлкен санды n модулі бойынша бір рет келтіру

$(a * a * a* a* a* a* a* a) \bmod n$

жасаудың қажеті жоқ. Оның орнына, үш көбейту мен үш рет модуль бойынша келтіру жасаған ыңғайлы:

$((a^2 \bmod n)^2 \bmod n)^2 \bmod n$.

Осы тәсілді қолдана отырып, мына есептеуді де орындауға болады:

$a^{16} \bmod n = (((a^2 \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n$.

x мәні 2-нің дәрежесіне тең болмаған жағдайда $a^x \bmod n$ есептеу, тек біраз қиындық тудырады. Дегенмен, одан шығу жолдары да бар. Екілік санау жүйесінде x санын 2 дәрежелерінің қосындысы ретінде көрсетуге болады.

Мәселен, $x = 25_{(10)} \rightarrow 11001_{(2)}$, сондықтан $25 = 2^4 + 2^3 + 2^0$. Онда

$a^{25} \bmod n = (a * a^{24}) \bmod n = (a * a^8 * a^{16}) \bmod n =$

$a*((a^2)^2)^2 *(((a^2)^2)^2)^2 \bmod n = ((((a^2 *a)^2)^2)^2 * a) \bmod n$.

Аралық нәтижелерді тиімді қолдану кезінде тек алты көбейту амалын орындау қажет болады:

$(((((((a^2 \bmod n) *a) \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n) * a) \bmod n$.

Бұл әдіс есептеу көлемін орташа есеппен 1,5xk операцияға дейін кемітеді (мұндағы k – санның битпен өлшенген ұзындығы).

Ең үлкен ортақ бөлгішті табуға арналған Евклид алгоритмі

Егер белгілі бір k бүтін сан үшін $b = k * a$ теңдігі орындалатын болса ғана, бүтін a саны, бүтін b санын қалдықсыз бөледі деп саналады. Бұл жағдайда a санын b санының *бөлгіші* немесе b санын көбейткіштерге жіктегендегі *көбейткіші* деп атайды.

a саны 1-ден үлкен ($a>1$) бүтін сан болсын делік. Егер осы санның бөлгіштері тек 1 және осы санның өзі болса, онда a саны *қарапайым сан* болады, ал керісінші жағдайда a саны – *құрама сан* деп аталады.

a және b сандарының ***Ең үлкен ортақ бөлгіші (ЕҮОБ)*** деп – a және b сандарын бір уақытта бөлетін ең үлкен бүтін санды айтады. Оны НОД(a,b) немесе (a,b) деп белгілейді. НОД(a,b) – a және b сандарын бөлетін және өзі де a және b сандарын бөлетін

санға бөлінетін натурал сан. Егер НОД(a,b)=1 болса, онда бүтін a және b сандары – *өзара қарапайым* сандар.

Ең үлкен ортақ бөлгішті ***Евклид алгоритмі*** бойынша есептеуге болады. Енді мынадай шартты белгілер енгізелік: q_i – бөлінді; r_i – қалдық. Онда алгоритмді келесі көрсетілген теңдіктер тізбегі ретінде көрсетуге болады:

$$\begin{aligned} a &= b * q_1 + r_1, \quad 0< r_1<b, \\ b &= r_1 * q_2 + r_2, \quad 0< r_2< r_1, \\ r_1 &= r_2 * q_3 + r_3, \quad 0< r_3< r_2, \\ &\vdots \\ r_{k-2} &= r_{k-1} * q_k + r_k, \quad 0< r_k< r_{k-1}, \\ r_{k-1} &= r_k * q_{k+1}. \end{aligned}$$

Бөлінгеннен қалған r_i қалдықтары натурал сандардың қатаң түрдегі кему тізбегін құрайтындықтан, бұл алгоритмнің санап бітуі кепілді болып саналады. Аталған тізбектен r_k ортақ бөлгіш екені көрінеді. Сонымен, r_k = НОД(a,b) немесе r_k = (a,b).

Кері шамаларды есептеу

Нақты сандар арифметикасында (нөлге тең емес) a саны үшін кері a^{-1} шамасын есептеу қиын емес: $a^{-1} = 1/a$ немесе $a * a^{-1} = 1$. Мәселен, 4 санының кері шамасы 1/4 болады, өйткені $4 * (1/4) = 1$.

Модулярлық арифметикада кері шаманы есептеу, күрделілеу болып келеді. Мұндай есептің жалпы қойылымы: $a * x \pmod n = 1$ орындалатындай бүтін x санын табу. Бұл өрнекті былайша да жазуға болады:

$a^{-1} \equiv x \pmod n$.

Бұл есептің шешімі бірде бар, бірде жоқ. Мысалы, 5 санының 14 модулі бойынша кері шамасы 3-ке тең, өйткені $5 * 3 = 15 \equiv 1 \pmod{14}$. Екінші жағынан, 2 санының 14 модулі бойынша кері шамасы болмайды.

Жалпы жағдайда, егер a және n өзара қарапайым сандар болса, онда

$a^{-1} \equiv x \pmod n$

салыстыруының бір ғана шешімі болады. Егер a және n өзара қарапайым емес сандар болса, онда салыстырудың шешімі болмайды.

Кері шамаларды табудың негізгі тәсілдерін қарастыралық.

Егер НОД(a,n) = 1 болса, онда $a * a^{-} \equiv 1 (\mathrm{mod}\ n)$ орындалатын және $0 < a^{-1} < n$ аралығында жататын кері a^{-1} саны болады.

0-ден (n-1)-ге дейінгі бүтін сандар жиынын n модулі бойынша – қалдықтардың *толық жиыны* деп атайды. Қалдықтардың толық жиынынан n санымен өзара қарапайым болатын қалдықтар жиынын бөліп алайық. Бұл жиынды – *қалдықтардың келтірілген жиыны* деп атайды. Жалпы, қарапайым n санының қалдықтарының келтірілген жиыны n-1 элементтен тұрады.

1-мысал. Модуль n=11 – қарапайым сан болсын. Онда 11 модулі бойынша қалындылардың толық жиыны: {0, 1, 2, …, 10}. Қалдықтардың келтірілген жиынын қалыптастыру үшін одан тек бір ғана элемент (0) алынып тасталады. Сонда 11 модулі бойынша қалдықтардың келтірілген жиыны 11–1=10 элементтен тұрады.

Қарапайым сандардың көбейтіндісі p*q=n үшін қалдықтарының келтірілген жиыны мына (p-1)(q-1) элементтерден тұрады. Егер n=p*q=2*5=10 болғанда, келтірілген жиындағы элементтер саны (p-1)(q-1)=(2-1)(5-1)=4 болады.

2-мысал. Модуль n=10 болсын. Онда 10 модулі бойынша қалдықтардың толық жиыны {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} болады. Оның ішінде тек 1, 3, 7, 9 сандары 10 санымен ортақ көбейткіші жоқ. Сондықтан, 10 модулі бойынша қалдықтардың келтірілген жиыны мынандай: {1, 3, 7, 9}. Бұл келтірілген жиынды қалыптастырғанда мына элементтер алынып тасталды: 0 (1 элемент), 2 еселік (4 элемент) және 5 еселік (1 элемент). Яғни, барлығы алты элемент. Оларды 10-нан алып тастасақ: 10–1–4–1=4, яғни келтірілген жиында төрт элемент қалады.

Қарапайым дәрежелі n^r модуль үшін қалдықтардың келтірілген жиыны $n^{r-1}(n-1)$ элементтен тұрады. Егер n=3, r=3 болса, онда $3^{3-1}(3-1) = 3^2*2 = 18$.

3-мысал. $27=3^3$ модулі бойынша қалдықтардың келтірілген жиыны 18 элементтен тұрады: {1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26}. Қалыңдықтардың толық жиынынан 3-ке еселенген (барлығы тоғыз) элементтер алынып тасталды.

Эйлер функциясы деп аталатын φ(n) функциясы, қалыңдықтардың келтірілген жиынындағы элементтер санының сипаттайды (1.1-кесте). Басқаша айтқанда, φ(n) функциясы – n санымен өзара қарапайым (n санынан кем) болатын оң бүтін сандар саны.

1.1-кесте

n модулі	φ(n) функциясы
n – қарапайым n^2 ⋮ n^r	n-1 n (n-1) ⋮ $n^{r-1}(n-1)$
p*q (p,q – қарапайым) ⋮ $\prod_{i=1}^{t} p_i^{e_{i-1}}$ (p_i – қарапайым)	(p-1)(q-1) ⋮ $\prod_{i=1}^{t} p_i^{e_{i-1}}$ (p_i –1)

Кері шамаларды $a^{-1} \equiv 1(\text{mod } n)$ табудың негізгі тәсілдері:

1. $a * a^{-1}(\text{mod } n) \equiv 1$ орындалатындай $a^{-1} \equiv 1 \text{ (mod n)}$ табылғанша 1, 2, …, n-1 мәндерін бірінен соң бірін тексеріп шығу.

2. Егер φ(n) Эйлер функциясы белгілі болса, онда тез дәрежелеу алгоритмін қолдана отырып $a^{-1}(\text{mod } n) \equiv a^{\varphi(n)-1}(\text{mod n})$ есептеп шығу.

3. Егер φ(n) Эйлер функциясы белгісіз болса, онда кеңейтілген Евклид алгоритмін пайдалануға болады.

Аталған тәсілдерді мысалдар арқылы көрсетеміз.

1) $a*x \equiv 1(\text{mod } n)$ орындалатындай $x \equiv a^{-1} \text{ (mod n)}$ табылғанша 1, 2, …, n-1 мәндерін, бірінен соң бірін тексеріп шығу.

Мәселен, n=7, a=5. Табу керек: $x \equiv a^{-1}(\text{mod } n)$.

$a * x \equiv 1(\text{mod } n)$,

$5 * x \equiv 1(\text{mod } 7)$, $n-1=7-1=6$.

Тексеру нәтижелері 1.2-кестеде келтірілген.

1.2- кесте

x	5*x	5*x (mod 7)
1	5	5
2	10	3
3	**15**	**1**
4	20	6
5	25	4
6	30	2

Сонымен, $x \equiv 5^{-1}(\text{mod } 7) = 3$.

2) φ(n) Эйлер функциясы белгілі болған кезде $a^{-1}(\text{mod } n)$ анықтау.

Мәселен, n=7, a=5. Табу керек: $x \equiv a^{-1}(\text{mod } n) = 5^{-1}(\text{mod } 7)$. n=7 модулі – қарапайым сан. Сондықтан, Эйлер функциясы $\varphi(n)=\varphi(7)=n-1 = 6$ тең болады. mod 7 бойынша 5 санының кері шамасы былайша есептеледі: $a^{-1}(\text{mod } n) \equiv a^{\varphi(n)-1}(\text{mod } n) =$

$=5^{6-1} \text{ mod } 7 = 5^5 \text{ mod } 7 = 5^2 \text{ mod } 7)(\ 5^3 \text{ mod } 7) \text{ mod } 7 =$

$= (25 \text{ mod } 7)(\ 125 \text{ mod } 7) \text{ mod } 7 = (4*6) \text{ mod } 7 = 24 \text{ mod } 7 = 3.$

Сонымен, $x = 5^{-1}(\text{mod } 7) = 3$.

3) Кеңейтілген Евклид алгоритмі арқылы $a^{-1}(\text{mod } n)$ кері шамасын анықтау.

Евклид алгоритмінің кеңейтілген тәсілінің практикалық маңызы зор. Бұл алгоритм берілген теріс емес бүтін a және b сандары үшін $a* u_1 + b* u_2 = u_3 = НОД(a,b)$ орындалатындай (u_1, u_2, u_3) векторын анықтайды.

Есептеу барысында қосымша (v_1,v_2,v_3), (t_1,t_2,t_3) векторлары пайдаланылады. Есептеу барысында векторлармен әрекеттер жасау үшін мына қатынастар орындалуы қажет:

$a*t_1 + b*t_2 = t_3$, $a*u_1 + b*u_2 = u_3$, $a*v_1 + b*v_2 = v_3$.

$a^{-1}(\text{mod } n)$ кері шамасын есептеу үшін кеңейтілген Евклид алгоритмінің жекеше жұмыс тәртібі қабылданған: b=n және НОД(a,n)=1. Бұл алгоритм мына векторды анықтайды:

$$(u_1, u_2, u_3),$$

мұнда: $u_3=1$, $a*u_1 + n*u_2 = \text{НОД}(a,n) = 1$,

$(a*u_1 + n*u_2) \text{ mod } n \equiv a*u_1 (\text{mod } n) = 1$, $a^{-1} (\text{mod } n) \equiv u_1 (\text{mod } n)$

Алгоритм қадамдары:

1. Бастапқы тағайындаулар: $(u_1, u_2, u_3) := (0, 1, n)$, $(v_1, v_2, v_3) := (1, 0, a)$.

2. $u_3=1$ екендігін тексеру. Егер $u_3=1$ болса, онда алгоритм аяқталады.

3. Бөлу, алу.

Мынадай тағайындаулар орындалады: алдымен $q:=[u_3/v_3]$, одан кейін

$$(t_1, t_2, t_3) := (u_1, u_2, u_3) - (v_1, v_2, v_3)*q,$$
$$(u_1, u_2, u_3) := (v_1, v_2, v_3),$$
$$(v_1, v_2, v_3) := (t_1, t_2, t_3).$$

Екінші қадамға қайтадан өтеміз.

4. Алгоритмнің соңы.

Мәселен, n=23, a=5. Осы модуль бойынша $a^{-1} (\text{mod } 23)$ кері шаманы, яғни $x=5^{-1} (\text{mod } 23)$ табу керек.

Кеңейтілген Евклид алгоритмін қолданып, есептеулер жасаймыз. Жеке қадамдардың нәтижелері 1.3-кестеде келтірілген.

1.3-кесте

q	u_1	u_2	u_3	v_1	v_2	v_3
-	0	1	n=23	1	0	a=5
4	1	0	5	-4	1	3
1	-4	1	3	5	-1	2
1	5	-1	2	-9	2	1
-	-9	2	1			

$u_3=1$, $u_1=-9$, $u_2=2$ болған кезде:

$(a*u_1+ n*u_2) \bmod n = (5*(-9) + 23*2) \bmod 23 = 5*(-9) \bmod 23 \equiv 1$,

$a^{-1}(\bmod n) = 5^{-1}(\bmod 23) = (-9) \bmod 23 = (-9 + 23) \bmod 23 = 14$.

Сонымен, $x = 5^{-1}(\bmod 23) = 14 (\bmod 23) = 14$.

Бұдан да күрделі

$$a*x \equiv b (\bmod n), \text{ яғни } b \neq 1, x = ?$$

салыстыруларды шешу үшін келесі тәсіл қолданылады.

Алдымен $a * y \equiv 1(\bmod n)$ салыстыру шешіледі, яғни $y = a^{-1} (\bmod n)$ анықталады. Содан кейін $x = a^{-1} b (\bmod n) = y * b(\bmod n)$ табылады.

Мысал. $5*x \equiv 9 (\bmod 23)$ салыстыруы үшін x-ті табу керек. Ол үшін алдымен $5*y \equiv 1(\bmod 23)$ салыстыру шешіледі. Яғни, $y=5^{-1}(\bmod 23)=14$ болады. Содан соң $x = 5^{-1}*9 (\bmod 23) = 14*9 (\bmod 23) = 126 (\bmod 23) \equiv 11 (\bmod 23)$ табылады. Сонымен, $x = 11$ болады.

2. КРИПТОГРАФИЯЛЫҚ ЖҮЙЕЛЕР

Қазіргі замандағы криптография 4 ірі бөлімнен тұрады: *симметриялық криптожүйелер, ашық кілтті криптожүйелер, электрондық қолтаңба жүйелері және кілттерді басқару*.

Криптографиялық әдістерді қолданудың негізгі бағыттары мыналар: жасырын ақпаратты байланыс арналары (мысалы, электрондық пошта) арқылы тасымалдау, жіберілген хабарлардың түпнұсқалығын анықтау, ақпаратты (құжаттарды, дерекқорларды) шифрланған түрде тасуыштарда сақтау.

Ақпаратты кодалау үшін пайдаланылатын таңбалардың шектеулі жиынтығы – ***әліпби*** (алфавит, alphabet) деп аталады. Жалпы түрде кез келген әліпбиді былай көрсетуге болады: $\sum=\{a_0, a_1, a_2, ..., a_{m-1}\}$.

Белгілі бір ереже бойынша (әліпбидегі әріптерді біріктіру арқылы) жаңа әліпби құруға болады. Жалпы жағдайда, n әріптері бойынша біріктірсек, онда m^n n-граммалары бар $\sum^n$ әліпбиі шығады.

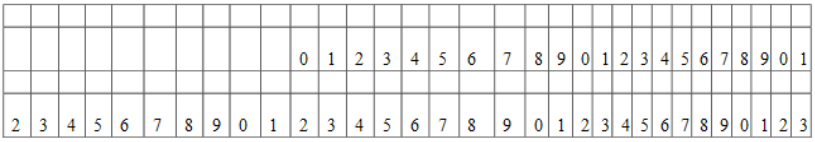
Мәселен: $\sum=\{ABCDEFGH ... WXYZ\}$ ағылшын әліпбиіндегі m=26 әріптерді біріктіру арқылы

- $26^2=676$ (AA, AB, ..., XZ, ZZ) биграммалары бар әліпби;
- $26^3=17576$ (AAA, AAB, ..., ZZX, ZZZ) үшграммалары бар әліпби.

Криптографиялық түрлендіруді орындау кезінде әліпби әріптерін бүтін сандарға 0, 1, 2, 3, ... ауыстырған пайдалы. Мысалы:

- қазақ әліпбиі $\sum_{қаз}=\{АӘБВГҒДЕ ... ЮЯ\}$, $\overline{Z}_{42}=\{0, 1, 2, …, 41\}$;
- орыс әліпбиі $\sum_{орыс}=\{АБВГДЕ ... ЮЯ\}$, $\overline{Z}_{31} = \{0, 1, 2, …, 30\}$;
- ағылшын әліпбиі $\sum_{ағыл.}=\{ABCDEF ... YZ\}$, $\overline{Z}_{26} = \{0, 1, …, 25\}$.

2.1-суретте қазақ тілінің метаәліпбиі (цифрлар және тыныс белгілері ескерілген әліпбиі) келтірілген.



2.1-сурет. Қазақ тілінің метаәліпбиі

2.1. Симметриялық криптожүйелер

Симметриялық криптожүйеде шифрлау және кері шифрлау үшін бір кілт пайдаланылады (шифрлау кілтін білу, шифрды ашу кілтін білуге мүмкіндік береді) [1, 3, 24,31, 32, 34, 37].

2.1.1. Ауыстыру шифрлары

Шифрлау кезінде ашық мәтіннің символдары (ауыстырудың алдын ала қойылған ережесіне сәйкес), сол немесе басқа әліпбидің символдарымен ауыстырылады. Мұндай шифрлардың қарапайым және күрделі түрлері болады.

1) Қарапайым ауыстыру шифрлары

Қарапайым ауыстыру шифрында бастапқы мәтіннің әрбір символы мәтіннің соңына дейін, сол әліпбидің бірдей символдарымен ауыстырылады (бір әліпбилік ауыстыру шифрлары деп те аталады). Қарапайым ауыстыру шифрларының мысалдары: Полибий квадраты, Цезарь шифры, Трисемус шифры, Кардано торы, Плейфер шифры, т.б.

Полибий квадраты

Полибий квадраты – қарапайым ауыстырудың алғашқы шифрларының бірі болып есептеледі. 5х5 квадраттың әрбір орнына кездейсоқ түрде алынған бір әріп жазылады (2.2-сурет). Грек әліпбиінің әріптерімен толтырғанда бір орын бос қалған, ал латын әліпбиінің әріптерімен толтырғанда – бір орынға екі әріп (i және j) жазылған.

λ	ε	υ	ω	γ
ρ	ζ	δ	σ	ο
μ	η	β	ξ	τ
ψ	π	θ	α	X
χ	ν		φ	ι

а) грек алфавиті

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

ә) латын алфавиті

2.2-сурет. Полибий квадраты

Полибий квадраты көмегімен шифрлау кезінде ашық мәтіннің кезекті әрпінің орнына, сол бағанда одан төмен орналасқан әріп – шифрмәтінге жазылады. Егер ашық мәтіннің әрпі кестенің төменгі қатарында болса, онда шифрмәтін үшін осы бағанның ең жоғарғы әрпі алынады. Мысалы, ψ α ν ρ ο σ сөзі үшін χ φ δ μ τ ξ шифрмәтіні шығады. Егер әрбір әріпті екі цифрмен (қатардың және бағанның нөмірі арқылы) белгілесек, онда бастапқы мәтін мынадай 41 44 13 21 25 24 сандар тізбегімен шифрланады.

Цезарь шифрлау жүйесі

Цезарь шифрын (бір әліпбилік ауыстыру) қолданған кезде бастапқы мәтіннің әрбір әрпі, сол әліпбидің одан К=3 әріпке тең ығысу аралығында орналасқан әріппен ауыстырылады. Ю.Цезарь шифрында (2.1-кесте) хабардағы латын әліпбиінің бірінші әрпі (A) төртінші (D), екіншісі (B) – бесінші (E), осылайша, ең соңғысы (Z) – үшінші (C) әріппен ауыстырылған.

2.1-кесте

Бір әліпбилік ауыстырулар (K=3, m=26)

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

Мысалы [24], Фарнак патшаны жеңгеннен кейін өзінің досы Аминтийге Цезарьдың жіберген VENI VIDI VICI (келдім,

көрдім, жеңдім) деген жолдауы шифрланғаннан кейін, мынадай түрде жазылады: YHQL YLGL YLFL.

Осыған ұқсас екінші бір шифр: Рим императоры Август хабар жібергенде бірінші әріпті (A) екіншісімен (B), екіншісін (B) - үшіншісімен (C), осылайша, ең соңғысын (Z) - бірінші (A) әріппен ауыстырған. Яғни K=1, сонда:

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

Бір әліпбилік ауыстыру жүйесіне қарсы криптоаналитикалық шабуыл, символдардың мәтінде кездесу жиілігін есептеуден басталады. Содан шифрмәтіндегі әріптердің қайталану жиіліктері, әліпбидегі әріптердің қайталану жиілігімен салыстырылады. Шифрмәтінде жоғары жиілікті әріп, әліпбидегі қайталану жиілігі жоғары әріппен ауыстырылады. Мәселен, ағылшын мәтінінде ең жиі кездесетін әріп E (қайталану жиілігі 0,13), одан кейін T (0,105), A (0,081), т.с.с., орыс мәтінінде О (0,090), Е (0,072), И (0,062), А (0,062), т.б. [24].

Кілттік сөзі бар Цезарь жүйесі

Бұл жүйенің ерекшелігі – ауыстыру әліпбиіндегі символдардың ығысқан және өзгертілген реті үшін кілттік сөздің қолданылуы. Кілттік сөз ретінде K санын, $0 \le K < 25$ және сөз немесе қысқа сөз тіркестері таңдап алынады. Кілттік сөздің әріптері әр түрлі болғаны жақсы.

Мәселен, кілт ретінде DAMELI сөзі және K=5 таңдалсын. Кілттік сөз, әліпби әріптерінің астына таңдалған K санына сәйкес келетін әріптен басталып жазылады:

0 1 2 3 4 5 10 15 20 25
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
D A M E L I

Ауыстыру әліпбиінің қалған әріптері әліпбилік ретпен, кілттік сөзден кейін (қалғандары алдыңғы жағынан) жазылады:

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

Сөйтіп, хабардың әрбір әрпі үшін ауыстыру әрпі анықталды. Енді KAZAKHSTAN мәтіні былайша шифрланады: IVUVIMNOVF.

Кілттік сөздің барлық әріптері әр түрлі болуы тиісті деген талаптың міндетті емес екенін ескеру керек. Кілттік сөзді (немесе сөздер тіркесін), жай бірдей әріптерді қайталамай ғана жазу керек. Мысалы, кілттік сөйлем АУЫЛДЫҢ ЖАНЫ ТЕРЕҢ САЙ және К=3 болғанда, ауыстырудың келесі кестесі алынады (2.2-кесте):

2.2-кесте

а	ә	б	в	г	Ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	h	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я
э	ю	я	*а*	*у*	*ы*	*л*	*д*	*ң*	*ж*	*н*	*т*	*е*	*р*	*с*	*й*	ә	б	в	г	ғ	з	и	к	қ	м	о	ө	п	ұ	ү	ф	х	h	ц	ч	ш	щ	ъ	і	ь

Трисемустың шифрлайтын кестесі

Осындай ауыстыру шифрын алу үшін, әдетте әліпби әріптері мен кілттік сөз (немесе сөздер тіркестігі) жазбасына арналған кесте қолданылған. Кестеге алдымен кілттік сөз жазылып, қайталанатын әріптері алынып тасталады. Содан кейін бұл кесте әліпбидің кілтке кірмей қалған әріптерімен реттелген түрде толықтырылады.

Қазақ әліпбиі үшін шифрлайтын кестенің өлшемі 6х7 болады. Кілт ретінде БҮРКІТ сөзін алайық. Осындай кілтпен шифрлайтын кесте 2.3-суретте көрсетілген. Шифрлау кезінде Полибий квадратындағы сияқты осы кестеден ашық мәтіннің кезекті әрпін тауып, одан төменгі бағанда орналасқан әріпті шифрмәтінге жазады. Егер бастапқы мәтіннің әрпі кестенің төменгі қатарында болса, онда шифрмәтін үшін сол бағандағы ең жоғарғы әріп алынады.

Бастапқы мәтін АҚПАРАТТЫ ҚОРҒАУ
Шифрмәтін ЖПЦЖГЖЕЕҮАПХГҚЖШ

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

2.3-сурет. БҮРКІТ кілттік сөзімен шифрлайтын кесте

Мұндай кестелік шифрларда шифрлау бір әріп бойынша орындалатындықтан олар – *монограммалы шифрлар* деп аталады. Трисемус шифрлайтын кестелердің екі әріптері бойынша шифрлауға болатынын байқаған. Мұндай шифрлар – *биграммалы* деп аталады.

Плейфердің биграммалы шифры

Плейфер жүйесінің шифрлау және шифрды ашу процедураларында Трисемустың шифрлайтын кестесі қолданылады.

Шифрлау процедурасы келесі қадамдардан тұрады:

1) Бастапқы хабардың ашық мәтіні әріптер жұбына (биграммаларға) бөлінеді. Мәтінде әріптердің саны жұп болу керек және құрамында екі бірдей әріп болмауы керек. Егер бұл талаптар орындалмаса, онда мәтін – мәні жоқ орфографиялық кестелердің көмегімен түрлендіріледі.

2) Ашық мәтіннің биграммалар тізбегі (шифрлайтын кестенің көмегімен) келесі ережелер бойынша түрлендіріледі:

- Егер ашық мәтін биграммасының екі әрпі де бір қатарға не бағанға (2.3-суреттің кестесіндегі Ғ және Ы әріптері сияқты) түспесе, онда берілген әріптердің жұбымен анықталатын тік бұрыш бұрышындағы әріптер ізделінеді. Біздің мысалда бұл – ҒЫВЭ әріптері. ҒЫ әріптер жұбы ВЭ жұбына бейнеленеді. Шифрмәтіндегі биграммаларда әріптердің тізбегі ашық мәтіннің биграммасындағы әріптер тізбегінің қатынасы бойынша айнадай орналасу керек.
- Егер ашық мәтін биграммасының екі әріптері де кестенің бір бағанында орналасса, онда шифрмәтіннің әріптері оның астында жатқан әріптер болып есептеледі. Мәселен, ШТ биграммасы шифрмәтіннің ЯЕ биграммасымен ауыстырылады. Егер ашық мәтіннің әрпі төменгі қатарда орналасса, онда шифрмәтін үшін осы бағанның жоғарғы қатарындағы сәйкес келетін әріп алынады.
- Егер ашық мәтіннің биграммасының екі әрпі де, кестенің бір қатарында орналасса, онда шифрмәтіннің әріптері олардың оң жағында жатқан әріптер болып есептеледі. Мысалы,

ҒД биграммасы шифрмәтіннің ДЕ биграммасымен ауыстырылады. Егер ашық мәтіннің әрпі соңғы оң жақ бағанда орналасса, онда шифр үшін осы қатардағы сол жақ бағандағы сәйкес келетін әріп алынады.

Мысал ретінде БАҒДАРҒЫЛАУЫШТАР мәтінін шифрлайық. Бұл мәтіннің биграммаларға бөлуі мынаны береді: БА ҒД АР ҒЫ ЛА УЫ ШТ АР. Осы биграммалар тізбегі шифрлайтын кестенің (2.3-сурет) көмегімен мынадай тізбекке түрлендіріледі: ҮБ ДЕ БК ВЭ ІН ОЯ ЯЕ БК.

Шифрды ашу кезінде аталған әрекеттер керісінше орындалады.

Стандартты әліпбиді қолданған кезде жай ауыстыру шифрларының олқылықтары бар: әліпби әріптерінің қайталану жиіліктерінің кестесі (жадуалы) бір немесе бірнеше символды анықтауға мүмкіндік береді. Ал бұл – кейде хабарды толығымен кері шифрлауға жеткілікті болады. Сондықтан, кері шифрлауды қиындату үшін әр түрлі тәсілдер қолданылады. Мәселен:

- шифрлаудың көпәріптік жүйесі – бір символға екі және одан да көп символдардың бір немесе бірнеше қисындасуы;
- бірнеше әліпбиді пайдалану – әрбір символдың орнына, оның өзімен немесе жіберіліп жатқан хабардағы оның орнымен қандай да болмасын бір тәсілмен байланысқан кілтке тәуелді, басқа бір әліпби қолданылады.

2) Күрделі ауыстыру шифрлары

Күрделі ауыстыру шифрларын – көпәліпбилік деп атайды. r-әліпбиін ауыстыру кезінде негізгі хабардың x_0 символы B_0 әліпбиіндегі y_0 символымен, x_1 символы B_1 әліпбиіндегі y_1 символымен ауыстырылады. Ал x_{r-1} символы B_{r-1} әліпбиіндегі y_{r-1} символымен және x_r символы B_0 әліпбиіндегі y_r символымен ауыстырылады. Күрделі ауыстыру шифрларының мысалы ретінде Вижинер квадратын, бір реттік шифрлауыш жүйесін, Гронсфельд шифрын, Уитстонның “қос квадрат” шифрын, Вернам әдісін, т.б. келтіруге болады.

r=4 болған жағдайда көпәліпбилік ауыстырудың жалпы сұлбасы 2.4-суретте келтірілгендей болады.

Енгізу символы	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
Ауыстыру алфавиті	B_0	B_1	B_2	B_3	B_0	B_1	B_2	B_3	B_0	B_1

2.4-сурет. Көпалфавитті ауыстырудың жалпы сұлбасы (r=4)

Гронсфельд шифры

Гронсфельд шифры деп аталатын бұл күрделі ауыстыру шифры, Цезарь шифрының өзгертілген бір түрі болып табылады. Ол үшін негізгі хабар әріптерінің астына, сандар түріндегі кілттің цифрлары жазылады. Егер кілт хабардан қысқа (аз) болса, онда кілттің цифрлары қайталана береді. Мысалы, кілт ретінде 2718 санын қолдана отырып, ҰЛЫ ЖІБЕК ЖОЛЫ хабары үшін келесі шифрмәтін (2.3-кесте) алынады:

2.3-кесте

Гронсфельд шифрымен шифрлау

Хабар		Ұ	Л	Ы		Ж	І	Б	Е	К		Ж	О	Л	Ы
Кілт		*2*	*7*	*1*		*8*	*2*	*7*	*1*	*8*		*2*	*7*	*1*	*8*
Шифрмәтін		Ф	Р	І		Н	Э	З	Ж	П		И	Ұ	М	Б

Хабардың бірінші Ұ әрпін шифрлау үшін кілттің бірінші цифры 2 екенін ескере отырып, әліпбидегі Ұ әрпінен бастап екінші әріпті (Y, Ф), яғни Ф әрпін таңдап алу керек. Әрі қарай шифрлау осылайша жалғаса береді.

Вижинер шифрлау жүйесі

Вижинер жүйесі, Цезарь шифрлау жүйесіне ұқсайды. Вижинер кестесі n^2 элементтен тұратын квадраттық матрица болып есептеледі. Бұл жерде n – қолданылатын әліпби символдарының саны. Бірінші қатарда әліпбидің барлық әріптері жазылады. Әрбір келесі қатарда бір әріпке ығыстырылады.

Осындай әрекетті қайталаудың нәтижесінде тең квадрат кесте құрылады (2.5-сурет).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
а	ә	б	в	г	Ғ	д	е	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	ъ	ы	і	ь	э	ю	
ә	б	в	г	ғ	д	е	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	ъ	ы	і	ь	э	ю	я	а
б	в	г	ғ	д	е	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	ъ	ы	і	ь	э	ю	я	а	ә
в	г	ғ	д	е	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	ъ	ы	і	ь	э	ю	я	а	ә	б
г	ғ	д	е	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	ъ	ы	і	ь	э	ю	я	а	ә	б	в
ғ	д	е	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	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г
д	е	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	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ
е	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	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д
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	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е
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	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж
и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з
й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и
к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й
қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к
л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ
м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л
н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м
ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н
о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң
ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о
п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө
р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п
с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р
т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с
у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т
ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у
ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ
ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү
х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф
һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х
ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ
ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц
ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	қ	л	м	н	ң	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч
щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	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
ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	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	э	ю	я	а	ә	б	в	г	ғ	д	е	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	ъ
і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	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	ъ	ы
ь	э	ю	я	а	ә	б	в	г	ғ	д	е	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	ъ	ы	і
э	ю	я	а	ә	б	в	г	ғ	д	е	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	ъ	ы	і	ь
ю	я	а	ә	б	в	г	ғ	д	е	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	ъ	ы	і	ь	э
я	а	ә	б	в	г	ғ	д	е	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	ъ	ы	і	ь	э	ю

2.5-сурет. Қазақ әліпбиіне арналған Вижинер кестесі

Кестенің екі кірісі бар: негізгі ашық мәтіннің әрпін анықтайтын жоғарғы қатардың символдары мен кілттің сол жақтағы шеткі бағаны.

Шифрлауды орындау үшін, әріптерден тұратын кілт таңдап алынады. Шифрлау былайша жүзеге асырылады: шифрланатын мәтіннің әрбір әрпінің астына кілттің әріптері жазылады; кестенің жоғарғы қатарынан ашық мәтіннің кезекті әрпімен, оның астында орналасқан кілттің әрпін байластыратын

сызықтардың қиылысқан жерінде тұрған кестенің әрпі табылады; шифрланатын мәтіннің әрпі кестенің осы әрпімен ауыстырылады. Осылайша шифрмәтіннің келесі әрпі табылады. Мысал ретінде “Жеруйық” кілтіне арналған жұмыс матрицасы 2.6-суретте келтірілген.

Ашық мәтін - **“ЕТЕМЕН ҚАЗАҚСТАН”**, кілт - **“Жеруйық”**																																						
а	ә	б	в	г	Ғ	д	е	ж	з	и	й	к	к	л	м	н	н	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я
Ж	з	и	й	к	Қ	л	м	н	н	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е
Е	ж	з	и	*й*	К	к	л	м	н	н	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д
Р	с	т	у	ұ	Ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	к	л	м	н	н	о	ө	п
Ұ	ү	ф	х	һ	Ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к	к	л	м	н	н	о	ө	п	р	с	т	у
И	к	к	л	м	Н	н	*о*	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и
Ы	і	ь	э	ю	Я	а	ә	б	в	г	ғ	д	е	ж	з	*и*	й	к	к	л	м	н	н	о	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ
Қ	л	м	н	н	О	ө	п	р	с	т	у	ұ	ү	ф	х	һ	ц	ч	ш	щ	ъ	ы	і	ь	э	ю	я	а	ә	б	в	г	ғ	д	е	ж	з	и	й	к

Хабар	Е	г	е	м	е	н	Қ	а	з	а	к	с	т	а	н
Кілт	*ж*	*ө*	*р*	*ұ*	*й*	*ы*	*қ*	*ж*	*ө*	*р*	*ұ*	*й*	*ы*	*қ*	*ж*
Шифрмәтін	м	й	х	я	о	и	ү	ж	н	р	э	щ	н	к	у

2.6-сурет. Вижинер кестесінің көмегімен шифрлау

Уитстонның “қос квадрат” шифры

1894 жылы Чарльз Уитстон “қос квадрат” деп аталатын биграммалармен шифрланатын әдіс тапты. “Қос квадрат” шифры, Плейфер шифрындағы сияқты биграммалармен шифрлау жүргізілетін екі кестені бірден қолданады.

Қазақ әліпбиінің кездейсоқ орналасқан символдары бар екі кесте берілсін (2.7-сурет). Шифрлар алдында бастапқы хабарды биграммаларға бөледі. Әрбір биграмма бөлек шифрланады. Биграмманың бірінші әрпі сол жақтағы кестеден, ал екінші әрпі – оң жақ кестеден алынады.

Мәселен, БҮ негізгі мәтіннің биграммасы делік. Б әрпі сол жақтағы кестенің 5-бағаны мен 7-қатарында, ап Ү әрпі оң жақтағы кестенің 3-бағаны мен 5-қатарында орналасқан. Тік төртбұрыштың төбелері 5 және 7-қатарларда, сондай-ақ сол жақтағы кестенің 5 және оң жақтағы кестенің 3-бағандарында орналасқан. Сондықтан, шифрмәтіннің биграммасына оң жақтағы кестедегі 3-баған мен 7-қатарда орналасқан І әрпі және сол жақтағы кестедегі 5-баған мен 5-қатарда орналасқан Ч әрпі кіреді.

Бастапқы хабар	БУ	ГІ	Н_	ЖА	НБ	ЫР	ЛЫ	Қ_	ҮН
Шифрмәтін	ІЧ	ЕФ	СЕ	УЭ	ИУ	**ШЕ**	ИА	РЙ	УР

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

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

2.7-сурет. “Қос квадрат” шифрына арналған қазақ әліпбиінің кездейсоқ орналасқан символдары бар екі кесте

Егер бастапқы хабар биграммасының екі әрпі де бір бағанда жатса, онда шифрмәтін әріптері де осы бағаннан (қатардан) алынады.

Бір реттік шифрлау жүйесі, Вернам шифры

Бір реттік шифрлау жүйесінің ерекшелігі, кілттік тізбектің бір рет қолданылуы болып табылады. Бұл шифрлау жүйесі $X=(X_0, x_1, \ldots, x_{n-1})$ бастапқы ашық мәтінді мынадай $Y_i = (X_i + K_i) \bmod m$, $0 \le i \le n-1$ Цезарь ауыстыруын қолдана отырып $Y=(Y_0, y_1, \ldots, y_{n-1})$, шифрмәтініне түрлендіреді. Бұл жерде K_i - кездейсоқ кілттік тізбектің і-элементі.

Шифрды ашу процедурасы мына теңдікке сәйкес орындалады.

$X_i = (Y_i - K_i) \bmod m$.

Бір реттік шифрлау жүйесін 1917 жылы Дж.Моборн мен Г.Вернам ойлап тапқан. Вижинер шифрлау жүйесінің m=2 болған кездегі түрі – Вернам *шифрлау жүйесі* деп аталған. Ол кезде, $K=(k_0, k_1, k_2 \ldots, k_{n-1})$ екілік кілттердің кездейсоқ тізбегі алдын ала қағаз таспаға жазылатын болған. Алты қосымша символдармен кеңейтілген ағылшын {A, B , C, D, …, Z} әліпбиіндегі бастапқы мәтіннің әрбір әрпі алдымен, Бодо телеграф

кодасын қолдана отырып, 5-биттік символға (b_0, b_1 ,..., b_4) аударылған. Одан кейін осы Бодо мәтініне екі модулі бойынша кілт қосылады, яғни шифрмәтін символдары: $y = x \oplus k$. Кері шифрлау, шифрмәтіннің сол **k** кілттер тізбегі мен **y** символдары екі модулі бойынша қосылуынан анықталады: $y \oplus k = x \oplus k \oplus k = x$.

Бұл ауыстыру жүйесін іске асыру үшін, бір реттік блокнот пайдаланылады. Екінші дүниежүзілік соғыс кезінде шифрлау үшін қағаздан жасалған блокнот кеңінен қолданылған және Вернам шифрын ***"бір реттік блокнот"*** деп атау қалыптасып кеткен. Блокнот парақтарының әрқайсында K_i кездейсоқ сандар (кілттер) басылған. Блокноттың бірдей екі данасы жасалынады. Хабар жіберуші өз мәтінін блокноттың бірінші парағы арқылы шифрлайды. Хабарды шифрлап біткен соң, ол пайдаланылған парақты жояды да, шифрланған ақпаратты екінші абонентке жібереді. Қабылдаушы жақ келген хабарды кері шифрлап біткеннен кейін, блокноттың пайдаланылған парағын жояды. Сөйтіп, хабардың әрбір символы үшін K_i кілті тек бір рет қолданылады. Жаңа хабарды шифрлау үшін жаңа парақтан бастайды.

Бір реттік блокноттың кейбір вариантарында, мысалы хабарды жіберуші мен алушыға белгілі кілт-кітаптағы алдын ала келісілген парақтармен анықталады. Кілттік тізбек сол кітаптың көрсетілген жерінен басталып, Вижинер жүйесіндегі сияқты қолданылады. Кейде мұндай шифрды – ***шексіз кілті бар шифр*** деп атайды. Егер кілт белгісіз болса, онда шифрланғаннан кейін, бастапқы мәтінді бұрынғы қалпына келтіру (яғни кілтсіз кері шифрлауға) мүмкін емес.

Мәселен, М. Әуезовтің "Абай жолы" романының, I томының 103 бетінде: «*Ұлжан ол хабарға сасқан жоқ. Айғыз екеуі екі күндей қам істеді. Үлкен-үлкен теңдерді шешіп, қымбат кілем, әсем тұскиіз, алаша, көрпелерді алып, Зере отырған үлкен үйді де, қонақ үйді де, Айғыз үйін де жақсы жасап қойысты. Астау-астау бауырсақ пісіріп, қой үйтіп, құрт ездіріп, астарын да ықшамдады. ...»* Осы үзіндіден басталатын парақтарды шексіз кілт ретінде қолдана отырып, 2.1-суретте келтірілген қазақ тілінің метаәліпбиін (m=64)

пайдаланып, **“Туған жерім менің Қазақстаным ...”** деген сөздерден басталатын мәтінді шифрлайық (2.8-сурет). Сонда алынған шифрмәтін мынадай символдар жиынынан басталады:

8ЮҚАЩЮФРЗБМ”МХРІДИҚСТАЦИШОЦҮГ …

Т	у	ғ	а	Н		ж	е	р	і	м		м	е	н	і	н		К	а	з	а	қ	с	т	а	н	ы	м			
24	25	5	0	17	52	9	7	22	37	16	52	16	7	17	37	18	52	14	0	10	0	14	23	24	0	17	36	16	52		
Ұ	*л*	*ж*	*а*	*н*		*о*	*л*		*х*	*а*	*б*	*а*	*р*	*ғ*	*а*			*с*	*а*	*с*	*қ*	*а*	*н*		*ж*	*о*	*қ*	.		*А*	*й*	*ғ*
26	15	9	0	17	52	19	15	52	29	0	2	0	22	5	0	52	23	0	23	14	0	17	52	9	19	14	55	52	0	12	5
50	40	14	0	34	40	28	22	10	2	16	54	16	29	22	37	6	11	14	23	24	0	31	11	33	19	31	27	4	52		
8	Ю	Қ	А	Щ	Ю	Ф	Р	З	Б	М	"	М	Х	Р	І	Д	И	Қ	С	Т	А	Ц	И	Ш	О	Ц	Ү	Г			

2.8-сурет. Шексіз кілт көмегімен шифрлау

Жоғарыда аталған Вижинер шифрының негізгі кемшілігі – кілттің ұзындығы қысқа болған жағдайда, бастапқы мәтіннің статистикалық қасиеттері көріне бастайды. Бұл кемшілікті жою үшін кілтті ұзарту керек. Осындай шифр – ***Вернам шифры***. Бұл шифрда шифрлау және кері шифрлау операциялары үшін Вижинер шифрының алгоритмі қолданылады. Бірақ кілттің ұзындығы бастапқы мәтіннің ұзындығына тең немесе одан артық болады: $Y_i = (X_i + K_i) \bmod m$, $i = 1, 2, \ldots, M$, бұл жерде M - бастапқы (ашық) мәтіннің ұзындығы.

Вернам шифрының шүбәсіз құпиялылықты қамтамасыз ететінін бірінші болып К.Шеннон [39] дәлелдеген. Шүбәсіз құпиялылықты қамтамасыз ететін криптожүйе – *Вернамның m модулі бойынша шифрлау жүйесі* деп аталады.

2.1.2. Орын ауыстыру шифрлары

Бұл әдіс кезінде шифрланатын мәтіннің символдары белгілі бір ереже бойынша орын ауыстырылады. Орын ауыстыру шифрлары ең қарапайым, сондай-ақ ең ежелгі шифр болып табылады [1, 3, 31].

Шифрлайтын кестелер

Жай орын ауыстыру әдісі – кестелік шифрлардың ең қарапайым түрі болып табылады. Бұл әдісте кестенің өлшемі, кілт міндетін атқарады. Мысалы, **КОМПЬЮТЕРЛІК ЖҮЙЕЛЕРДІ ҚОРҒАУ КЕРЕК** хабарын шифрлау үшін, ол

кестеге баған-баған бойынша жазылады. 4 қатардан және 7 бағаннан тұратын кестені толтыру нәтижесі 2.9-суретте көрсетілген.

К	Ь	Р	Ж	Л	І	Ғ	Е
О	Ю	Л	Ұ	Е	Қ	А	Р
М	Т	І	Й	Р	О	У	Е
П	Е	К	Е	Д	Р	К	К

2.9-сурет. 4 қатар және 8 бағаннан тұратын кестенің толтырылуы

Шифрмәтінді қалыптастыру үшін кестедегі жазылған мәлімет қатар-қатар бойынша оқылады. Егер шифрмәтінді 7 әріп бойынша топтайтын болсақ, онда мынадай шифрланған хабар алынады:

КЬРЖЛІҒЕ ОЮЛҮЕҚАР МТІЙРОУЕ ПЕКЕДРКК

Шифрды ашу кезінде іс-әрекеттер кері ретпен орындалады.

Кілт бойынша орын ауыстыру әдісі. Бұл әдісте кестенің бағандары кілттік сөз, сөздер тіркесі немесе сандар жиыны бойынша орын ауыстырылады. Егер кілттік сөзде бірдей әріптер кездессе, онда нөмірлеу солдан оңға қарай жүргізіледі. Мысалы, кілт ретінде БҰРҚАСЫН сөзін қолданайық, ал хабардың мәтінін алдыңғы мысалдан аламыз (2.10-сурет). Сол жақтағы (а) кестенің жоғарғы қатарында кілт, ал екінші қатарында кілт әріптерінің әліпбидегі реттік нөмірі көрсетілген. Оң жақтағы (б) кестенің бағандары кілт әріптерінің реттелген нөміріне сәйкес орын ауыстырылған. Сонда оң жақтағы кестеде мынадай шифрланған хабар алынған: **ЛКЖЕРІЬҒ ЕОҮРЛҚЮА РМЙЕІОТУ ДПЕККРЕК**

Кілт

Б	*Ұ*	*Р*	*Қ*	*А*	*С*	*Ы*	*Н*
2	7	5	3	1	6	8	4
К	Ь	Р	Ж	Л	І	Ғ	Е
О	Ю	Л	Ұ	Е	Қ	А	Р
М	Т	І	Й	Р	О	У	Е
П	Е	К	Е	Д	Р	К	К

а) орын ауыстыруға дейін

А	Б	Қ	Н	Р	С	Ұ	Ы
1	2	3	4	5	6	7	8
Л	К	Ж	Е	Р	І	Ь	Ғ
Е	О	Ұ	Р	Л	Қ	Ю	А
Р	М	Й	Е	І	О	Т	У
Д	П	Е	К	К	Р	Е	К

ә) орын ауыстырудан кейін

2.10-сурет. 4 қатар және 8 бағаннан тұратын кестенің толтырылуы

Екі рет орын ауыстыру. Қосымша жасырындылықты қамтамасыз ету үшін шифрланудан өткен хабарламаны қайта шифрлауға болады. Шифрлаудың мұндай әдісі – екі рет орын ауыстыру деп аталады. Бұл әдісте орын ауыстыру кестелері жеке баған және жеке қатар үшін анықталады. Кестеге алдымен хабардың мәтіні жазылады, содан кейін кезек кезекпен бағандар, сосын қатарлар ауыстырылады. Шифрды ашу кезінде ауыстырулар кері ретте жүргізіледі. 2.11-суретте екі рет орын ауыстыру әдісін жүзе асыру мысалы көрсетілген. Бастапқы кесте бағандарының нөмірлері мен қатарлары нөмірлерінің тізбегі (231 және 31524) кілт міндетін атқарады. Шифрлау нәтижесінде оң жақтағы (б) кестеде мынадай шифрмәтін алынады: **ААРРҚ ОПАҚУ ҒАЫТТ**.

	2	3	1
3	А	Қ	П
1	А	Р	А
5	Т	Т	Ы
2	Қ	О	Р
4	Ғ	А	У

а) бастапқы кесте

	1	2	3
3	П	А	Қ
1	А	А	Р
5	Ы	Т	Т
2	Р	Қ	О
4	У	Ғ	А

ә) бағандардың орнын ауыстыру

	1	2	3
1	А	А	Р
2	Р	Қ	О
3	П	А	Қ
4	У	Ғ	А
5	Ы	Т	Т

б) қатарлардың орнын ауыстыру

2.11-сурет. Екі рет орын ауыстыру әдісінің мысалы

Сиқырлы квадраттар

Сиқырлы квадрат деп – оның клеткаларына әрбір бағанының, әрбір қатарының және әрбір диагоналының қосындысы бірдей сан беретін 1-ден басталатын натурал сандардың тізбегі жазылған квадраттық кестені атайды.

Шифрланатын мәтін сиқырлы квадратқа, оның клеткаларының нөмірленуіне сәйкес жазылады. Егер содан кейін қатар-қатар бойынша осындай кестенің құрамын жазып алса, онда бастапқы хабарлама әріптерінің орнын ауыстыру арқылы жинақталған шифрмәтін алынады.

2.12-суретте АҚПАРАТТЫ ҚОРҒАУ мәтінін сиқырлы квадраттың көмегімен шифрлау мысалы көрсетілген. Оң жақтағы кестеде мынадай шифрмәтін алынған: .ПҚҒ РҚОТ ЫАТР АУАА

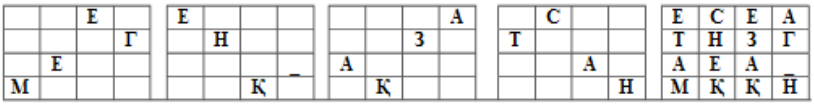
16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

.	П	Қ	Ғ
Р	Қ	О	Т
Ы	А	Т	Р
А	У	А	А

2.12-сурет. 4х4 сиқырлы квадратын пайдалану

Кардано торы

Джераломо Кардано сиқырлы квадраттар теориясымен әуестену нәтижесінде, орын ауыстыру шифрларының *торлар* немесе *трафареттер* деп аталатын жаңа түрін ашқан. Мұндай квадрат кестеде ұяшықтардың ширегі төрт рет бұру кезінде, квадраттың барлығын жабатындай етіліп кесілген. Мәтіннің кесілген ұяшықтарына жазу және торды бұру, барлық квадрат толтырылғанға дейін созылады. Мысалы, 2.13-суретте 4х4 торымен шифрлау үрдісі көрсетілген. ЕГЕМЕН_ҚАЗАҚТАН мәтінін шифрлаған соң **ЕСЕА ТНЗГ АЕА_ МҚҚН** шифрмәтіні алынған.



2.13-сурет. 4х4 торымен шифрлау мысалы

Мұндай торлардың саны олардың өлшеміне байланысты тез өседі: 2х2 торы біреу, 4х4 торы 256, ал 6х6 өлшемді торлардың саны жүз мыңнан асады.

2.1.3. Гаммалау әдісі бойынша шифрлау

Шифрдың гаммасы – ашық деректерді шифрлау және шифрланған деректерді кері шифрлау үшін, берілген алгоритм бойынша алынған жалғанкездейсоқ (pseudo-random) екілік сандар тізбегі. ***Гаммалау*** – белгілі бір заңға сәйкес шифрдың гаммасын ашық деректердің үстіне салу (беттестіру) үрдісі.

Бұл әдісте шифрланатын мәтіннің символдары – *гамма* деп аталатын арнаулы тізбектің символдарымен қосылады. Кейде

белгілі бір заң бойынша ашық деректер үстіне шифрдың гаммасы беттестіріледі. Сондықтан бұл әдіс – *гаммалау* деп аталады, ал шифрдың гаммасы – белгілі бір алгоритм бойынша ашық деректерді шифрлауға және шифрланған деректерді ашуға арналып жасалған жалғанкездейсоқ тізбек.

Гаммалау арқылы шифрлаудың мәні мынада: жалғанкездейсоқ сандар генераторының көмегімен шифрдың гаммасын генерациялау және алынған гамманы бастапқы мәтінге қайтадан кері аударуға болатындай етіп (мысалы, екі модулі бойынша қосу операциясын пайдалану арқылы) беттестіру.

Шифрлар алдында ашық деректерді, ұзындығы бірдей, әдетте 64 биттен, $T_0^{(i)}$ блоктарына бөледі. Шифрдың гаммасы осыған ұқсас, ұзындығы $Г_ш^{(i)}$ блоктарынан тұратын тізбектер түрінде құрылады. Шифрлау теңдеуі мына түрде болады: $T_ш^{(i)} = Г_ш^{(i)} \oplus T_0^{(i)}$, i=1÷M. Бұл жерде $T_ш^{(i)}$ - мәтін-шифрдың і-блогы; $Г_ш^{(i)}$ – шифр-гамманың і-блогы; $T_0^{(i)}$ – ашық мәтіннің і-блогы; M – ашық мәтін блоктарының саны.

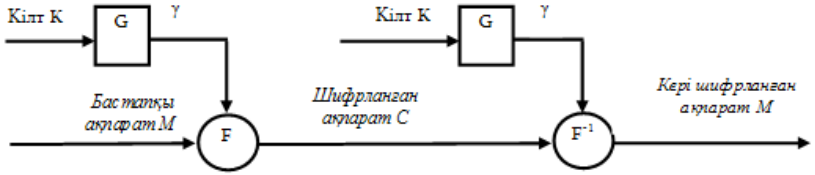
Кері шифрлау үрдісі шифр гаммасын қайтадан генерациялау және осы гамманы шифрланған деректермен беттестіруден тұрады. Кері шифрлау теңдеуінің түрі мынадай: $T_0^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)}$.

Осындай әдіспен алынған шифрмәтін, ашуға қиындық тудырады, өйткені оның кілті – айнымалы шама. Шифр гаммасы әр шифрланған блок үшін кездейсоқ түрде өзгеріп тұруы қажет. Егер гамма периоды барлық шифрланған мәтін ұзындығынан көп болса және шифрды бұзушыға бастапқы мәтіннің ешқандай бөлігі белгілі болмаса, онда мұндай шифрды тек кілттің барлық варианттарын түгел қарастыру арқылы ғана ашуға болады. Бұл жағдайда шифрдың криптограммалық беріктілігі, кілт ұзындығымен анықталады.

Жалғанкездейсоқ сандар тізбегін генерациялау әдістері. Гаммалау әдісімен шифрлағанда кілт есебінде биттердің кездейсоқ қатары пайдаланылады. Бұл қатар екілік түрде берілген (мысалы A=00000, B=00001, C=00002, т.б.) ашық мәтінмен қосылады. Бұл қосылу, екі модулі бойынша биттерді өзара қосу арқылы жүзеге асырылады. Нәтижесінде шифрланған

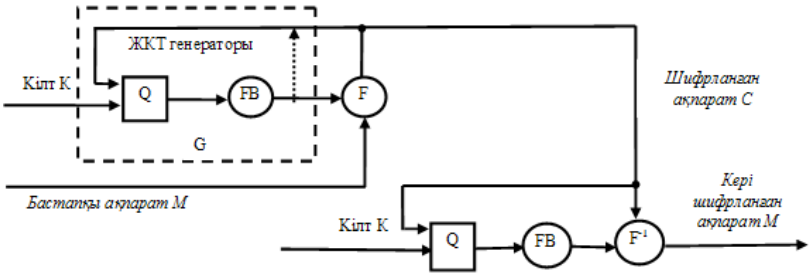
мәтін пайда болады. Күні бұрын болжауға болмайтын ұзындығы үлкен екілік тізбектерді генерациялау, классикалық криптографиядағы маңызды проблемалар қатарына жатады. Бұл проблеманы шешу үшін екілік жалғанкездейсоқ тізбектер генераторлары пайдаланылады [6, 14, 27].

Практика жүзінде көбінесе 2.14-суретте көрсетілген гаммалау сұлбасы пайдаланылады [6]. Мынадай шартты белгілер қабылданған: G – жалғанкездейсоқ тізбектер (ЖКТ) генераторы; F – сызықтық (мәселен, XOR немесе mod p) немесе бейсызықтық функция; FB – ЖКТ генераторының кері байланыс функциясы; Q – ЖКТ генераторының жад әлементі. Бұл сұлбаның сенімділігі, қолданылатын ЖКТ генераторының сапасымен анықталады. Бұл криптографиялық алгоритм – *OFB (Output FeedBack) режімінде шифрлау* деп аталады. Бастапқы m тізбегінің әрбір m_i элементі γ тізбегінің тиісті γ_i элементін қолдана отырып, басқаларынан тәуелсіз шифрланады.



2.14-сурет. Гаммалау сұлбасы (синхрондық ағындық шифрлау)

Кері байланысы бар гаммалау сұлбасын (2.15-сурет) пайдаланған кезде бастапқы m тізбегінің әрбір элементін шифрлаудың нәтижесі, оның алдындағы барлық элементтерге тәуелді болады. Мұндай криптоалгоритмді *CFB (Ciphertext FeedBack) режімінде шифрлау* деп атайды.



2.15-сурет. Кері байланысы бар гаммалау сұлбасы
(өзі уақытүйілесімделген ағындық шифрлау)

2.1.4. Блоктық шифрлардың жұмыс істеу режімдері

Шифрланатын деректердің көлемі, блоктың ұзындығынан артық болған жағдайларда да, блоктық шифрлау алгоритмін қолдануға болады. Бірақ, шифрлау үшін арнайы шифрлау режімдері қолданылуы керек. Шифрлау режімдерінің көптеген варианттары бар. Шифрлау режімдері барлық блоктық шифрлау алгоритмдері үшін ортақ.

Блоктық алгоритмінде бес шифрлау режімі көзделген [10, 18, 24, 34]: электрондық кодалық кітап (ECB); шифрмәтін блоктарын іліністіру (CBC); шифрмәтін бойынша кері байланыс (CFB); шықпа бойынша кері байланыс режімі (OFB); санауыш режімі (CTR).

Электрондық кодалық кітап (ECB - Electronic Code Book)

Бастапқы M мәтін, ұзындығы шифрлау алгоритмнің блок ұзындығына тең n блоктарға бөлінеді: $M=(M_1, M_2,...,M_n)$. Егер блоктық шифрдың K кілтке тәуелді шифрлауды E_K және кері шифрлауды D_K деп белгілесек, онда электрондық кодалық кітап режіміндегі шифрлауды мына түрде көрсетуге болады: $C_i=E_K(M_i)$, $i=1,...,n$. Яғни, бастапқы мәтіннің блоктары жеке

шифрланады, ал кері шифрлау – шифрлауға қарама-қарсы болады: $M_i=D_K(C_i)$, $i=1,...,n$.

ECB әдісінің бірнеше кемшіліктері бар. Бірінші кемшілік – шифрланатын деректердің ұзындығы, блок ұзындығына еселі болмау керектігінен туады. Осының салдарынан, шифрланатын деректердің соңғы блогының ұзындығы кем болып шығады. Блоктық шифрларда криптографиялық түрлендіру тек қана толық блок үшін орындалатындықтан, мұндай жағдайда соңғы блокты толықтыруға тура келеді. Бұл кемшілік – кейбір жүйелерде техникалық қиыншылықтар тудырады. Екінші кемшілік тікелей криптографиялық тұрақтылыққа қатысты. Себебі, ашық мәтіннің бірдей блоктарын шифрлау нәтижесінде бірдей шифрмәтін блоктары қалыптастырылады. Осы кемшіліктерге орай бұл әдіс, тек ұзындығы қысқа деректер үшін қолданылады.

Шифрмәтін блоктарын іліністіру
(CBC – Cipher Block Chaining)

Бұл режімде блогты шифрлау үрдісі, алдындағы деректер блогын шифрлау нәтижелеріне тәуелді. CBC режімі ақпараттың үлкен көлемдерін шифрлау үшін жиі қолданылатын режім болып саналады. ECB режіміндегі сияқты, бастапқы M мәтін ұзындығы шифрлау алгоритмнің блок ұзындығына тең n блоктарға бөлінеді. Одан кейін деректер былайша өңделеді:

а) бірінші блок екі модулі бойынша C_0 бастапқы векторымен қосылады;

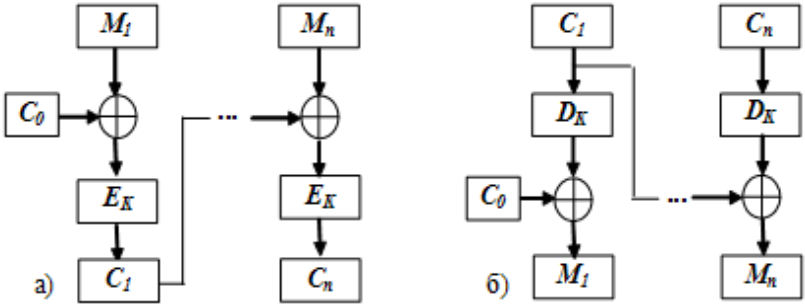
ә) алынған мән, блоктық шифрлау алгоритмнің көмегімен шифрланады;

б) нәтижесінде шифрмәтіннің алғашқы блогы алынады. Бұл блок келесі ашық мәтіннің блогын шифрлау үшін бастапқы вектор ретінде қолданылады.

Шифрмәтін блоктарын іліністіру режіміндегі түрлендіруді мына формуламен сипаттауға болады: $C_i=E_K(M_i \oplus C_{i-1})$, i=1,...,n. Кері шифрлау шифрлауға қарама-қарсы жүргізіледі:

$M_i=D_K(C_i)\oplus C_{i-1}$, i=1,...,n. Бұл үдерістердің сұлбасы 2.16-суретте келтірілген.

Бұл режімді қолданғанда алынатын шифрмәтін электрондық кодалық кітап режімі көмегімен алынған мәтінмен салыстырғанда, криптографиялық тұрақтылығы жоғары болады. Мұның себебі – шифрмәтіннің әр блогы одан бұрын шифрланған блоктардың барлығына тәуелді.



2.16-сурет. Шифрмәтін блоктарын тізбектеу режіміндегі
(а) шифрлау және (ә) кері шифрлау алгоритмнің сұлбасы

Бұл режімді қолданғанда алынатын шифрмәтін, электрондық кодалық кітап режімі көмегімен алынған мәтінмен салыстырғанда криптографиялық тұрақтылығы жоғары болады. Мұның себебі – шифрмәтіннің әр блогы одан бұрын шифрланған блоктардың барлығына тәуелді.

Шифрмәтін блоктарын іліністіру режімінің бұл қасиеті қосымша мүмкіндік береді – соңғы блок, бастапқы мәтін мен шифрмәтіннің барлық блоктарына тәуелді болғандықтан, оны хабардың тұтастығын бақылау үшін қолдануға болады. Бірақ, бұл режімде соңғы блоктың толық болмау мәселесі шешілмеген. Бұл мәселені шешу үшін блоктық шифрлау алгоритмді ағындық шифрлау жүйесіне түрлендіретін шифрмәтін бойынша кері байланыс және шықпа бойынша да, кері байланыс режімдері қолданылады.

Шифрмәтін бойынша кері байланыс
(CFB - Cipher Feed Back)

Блоктық шифрдан ағындық шифр алу үшін қолданылатын режім (33-сурет). Блоктың мөлшері шифр блогының өлшеміне тең немесе кіші болады.

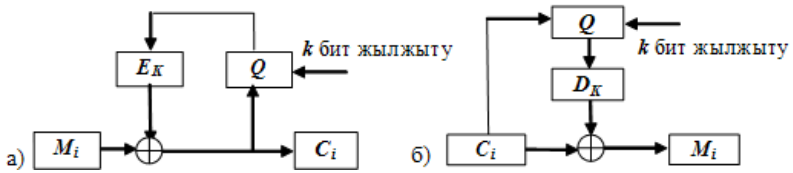
Бұл режімді қолданғанда деректер былай өңделеді:

а) Q жылжыту регистрі болып табылады және алғашқыда ол C_0 бастапқы мәнімен толтырылады. Одан кейін Q мәні шифрланады;

ә) шифрланған блоктың алғашқы k биті ашық деректер блогымен екі модулі бойынша қосылады;

б) Q солға қарай k позицияға жылжытылады, бұл биттердің орны шифрмәтін блогымен толтырылады да, қайта шифрланады, т.с.с.

Шифрмәтін бойынша кері байланыс режіміндегі шифрлау сұлбасы 2.17а-суретте, ал кері шифрлау сұлбасы 2.17ә-суретте көрсетілген. Бұл режімнің негізгі артықшылығы – деректер блогының ұзындығы кез келген болуы мүмкін.



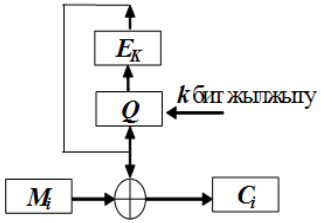
2.17-сурет. Шифрмәтін бойынша кері байланыс режіміндегі шифрлау және кері шифрлау алгоритмнің сұлбасы

Шықпа бойынша кері байланыс (OFB - Output Feed Back)

Шықпа бойынша кері байланыс режімі, шифрмәтін бойынша кері байланыс режіміне ұқсас. Олардың айырмашылығы – жылжытылған Q биттерінің орны шифрмәтін блогымен емес, бастапқы мәтінмен қосылатын гаммамен

толтырылады. Нәтижесінде, бұл режімде блоктық шифр классикалық ағындық шифр сияқты жұмыс істейді.

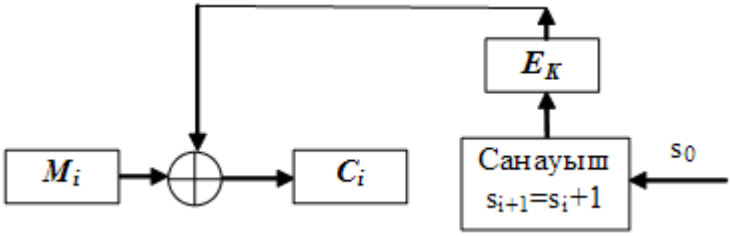
Шифрмәтін бойынша кері байланыс режімімен салыстырғандағы мұның ең үлкен айырмашылығы – бастапқы мәтін блоктары бір-бірінен тәуелсіз шифрланады. 2.18-суретте шықпа бойынша кері байланыс режіміндегі шифрлау алгоритмнің сұлбасы келтірілген. Кері шифрлау, осыған ұқсас жүргізіледі (C_i мен M_i блоктарының орындары ауысады).



2.18-сурет. Шықпа бойынша кері байланыс режімінде шифрлау

Санауыш режімі (CTR - CounTeR)

Санауыш (counter) режімі OFB режіміне өте ұқсайды, тек бұл режімде шифрдың алдыңғы шықпасы шифрланбайды. Оның орнына әрбір қадам сайын, мәні 1-ге өсіп тұратын санауыштың мәні шифрланады (2.19-сурет). CTR режімі OFB режіміне қарағанда сапалы деп саналады.

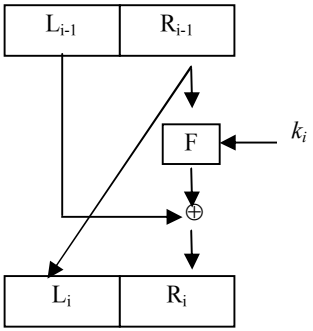


2.19-сурет. Санауыш режіміндегі шифрлау алгоритмі деп саналады

2.1.5. Фейстель желісі

Хорст Фейстель әзірлеген көп рет қайталанатын құрылым - Фейстель желісі немесе Фейстель құрылмасы (Feistel cipher, Feistel network) деген атқа ие болды. Осы құрылма көмегімен шифрлау бағдарламалақ деңгейде де, аппараттық деңгейде де жеңіл жүзеге асырылады. Блоктық шифрлардың көбісі Фейстель желісін пайдаланады 1, 34, 37].

Ашық мәтін белгілі бір ұзындығы бар блоктарға бөлінеді. Блок ұзындығы жұп болу керек (2.20-сурет). Шифрлау кезінде ашық мәтін блогы тең екі бөлікке бөлінеді – оң жақ (R) және сол жақ (L). Әрбір циклда оң бөлік R раундтық k_i кілтінің көмегімен f(R,k) функциясы бойынша түрлендіріледі (k_i кілттері бастапқы құпия кілттен алынады). Операция нәтижесі, сол бөлікпен L екі модулі бойынша қосылады. Содан кейін оң және сол жақтар бір-бірімен орын ауыстырады. Әр циклдағы түрлендірулер ұқсас болып келеді, соңғы циклды түрлендіру орындалмайды.



2.20-сурет. Фейстель желісінің жалпы құрылымы

Фейстель желісінің негізгі бір ерекшелігі – раунд функциясының f функциясы қасиеттеріне тәуелсіз қайтымды болғаны. Мұндай құрылым бір алгоритмді шифрлауға да, кері шифрлауға да қолдануға мүмкіндік береді.

Шифрлау үрдісінің і-ші циклындағы Фейстель желісінің (құрылмасының) жүзеге асыратын түрлендіруінің түрі мынадай:

$L_i = R_{i-1}$,

$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$.

Бұл жерде: L_i, R_i – блокты k_i кілтін қолдана отырып, f функциясының көмегімен шифрлау нәтижесі.

Кері шифрлау үрдісі мына формула түрінде жазылады:

$L_{i-1} = R_i \oplus f(L_i, k_i)$,

$R_{i-1} = L_i$.

IDEA, Blowfish, CAST-128, XTEA-3, RC5, RC6, RTEA сияқты, т. б. алгоритмдердің пайда болу кезінде Фейстель желісі кең таралған. DES және ГОСТ 28147-89 блоктық шифрларының негізінде де Фейстель желісі деп аталатын құрылым жатыр. Ал блоктық Rijndael алгоритмінде SP-желісі (ауыстыру-орыналмастыру желісі) пайдаланылады.

2.1.6. DES стандарты

DES (Data Encryption Standart) стандартын 1977 жылы АҚШ-тың Ұлттық стандарттар бюросы жариялаған. Бұл алгоритмді АҚШ стандарттар мен технологиялардың Ұлттық институты 1980 жылы маңызды, бірақ құпия емес ақпаратты рұқсатсыз қатынас құрудан қорғау үшін деректерді шифрлау стандарты ретінде қабылдаған [1, 31, 37].

Бұл стандарттың артықшылығы – кілттік жүйенің қарапайымдылығы, аппараттық және бағдарламалық жүзеге асыру жылдамдығы мен криптографиялық беріктілігінің жоғарылығы.

DES алгоритмі – бірқатар орын ауыстыру мен ауыстырулардың қисындасуын қолдана отырып, 64-биттік деректер блогын 56-биттік кілттің көмегімен шифрлауды жүзеге асырады. Кілттегі жұптылықты бақылауға арналған 8 тексеру биттерін қоса есептесе, оның ұзындығы 64 бит болады.

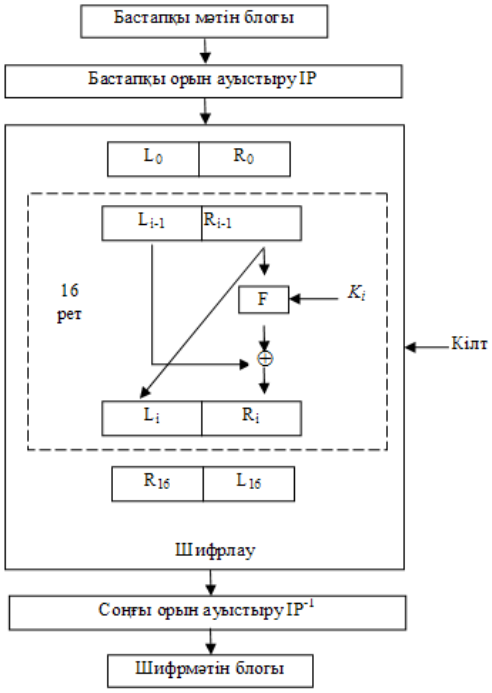
DES алгоритміндегі шифрлау үрдісінің жалпылама сұлбасы 2.21-суретте келтірілген. Шифрлау үрдісі кезінде ең алдымен 64 биттік блоктың биттерінің орны ауыстырылады. Одан кейін ол 56 биттік кілтпен шифрланып (16 раундтың әрқайсысында бастапқы 56 биттік кілттен әр түрлі 48 биттік кілт алынады), сосын биттердің соңғы орын ауыстырылады.

DES алгоритмінде мынадай шартты белгілер пайдаланылады:

- L және R - сол (left) және оң (right) биттердің тізбектері;
- LR - L және R тізбектерінің конкатенциясы. LR биттер тізбегінің ұзындығы L және R ұзындықтарының қосындысына

тең. LR биттер тізбегінде L тізбегінің биттерінен соң R биттері жазылады;

- ⊕ - екі модулі бойынша битті битке қосу операциясы.



2.21-сурет. DES алгоритміндегі шифрлаудың жалпылама сұлбасы

Бастапқы мәтін сақталатын файлдан кезекті 64 биттік (8 байттық) T блогы оқылады. Бұл T блогы IP бастапқы *орын ауыстыру матрицасы* көмегімен түрлендіріледі (2.4-кесте). Бұл кестедегі (және осыған ұқсас ауыстыру кестелеріндегі), мысалы бірінші қатардың бірінші бағанында орналасқан 58 саны. IP матрицасы кірме деректерінің (яғни, T блогының) 58-битін 1-орынға, 50-битін 2-орынға, т.с.с. ауыстырады дегенді білдіреді. Осылайша ауыстырылғаннан кейін IP(T) блогы екі бөлінеді: L_0 – 32 үлкен биттерден, R_0 – 32 кіші биттерден тұратын бөлігі.

Содан соң 16 қадамнан тұратын шифрлаудың итеративтік үрдісі орындалады. $T_i=L_iR_i$, мұндағы T_i – і-ші итерацияның нәтижесі; $L_i=t_1t_2\ldots t_{32}$ (бастапқы 32 бит); $R_i(t_{33}t_{34}\ldots t_{64}$ (соңғы 32 бит).

Сонда і-ші итерациясының нәтижесі мына формула арқылы жазылады:

$L_i=R_{i-1}$, i=1, 2, …, 16;

$R_i=L_{i-1}\oplus f(R_{i-1}, K_i)$, i=1, 2, …, 16.

f-функциясы – *шифрлау функциясы* деп аталады. Оның аргументтері, итерацияның алдыңғы қадамында алынған R_{i-1}-тізбегі мен 64 биттік К-шифрын түрлендіруден пайда болған 48 биттік K_i-кілті болып табылады (f-шифрлау функциясы мен K_i-кілт алгоритмі төменде келтірілген).

Итерацияның соңғы қадамында R_{16} және L_{16} тізбегі алынады (орын ауыстырусыз). Олар 64 биттік $R_{16}L_{16}$ тізбегіне конкатенцияланады.

Шифрлау аяқталған соң, биттер кері ауыстыру IP^{-1} матрицасы (2.5-кесте) көмегімен алғашқы орнына келтіріледі. IP^{-1} матрицасы мен IP матрицасының бірінші қатарындағы элементтердің өзара қатынасы 2.6-кестеде келтірілген.

Кері шифрлау былайша жүргізіледі: кері шифрланатын деректер алдымен IP^{-1} матрицасы бойынша орын ауыстырылады, содан соң $R_{16}L_{16}$ биттер тізбегіне шифрлау үрдісінде болатын амалдардың кері түрі қолданылады.

Кері шифрлау үрдісі мына формула түрінде жазылады:

$R_{i-1}=L_i$, i=1, 2, …, 16;

$L_{i-1}=R_i\oplus f(L_i, K_i)$, i=1, 2, …, 16.

Сонымен, орын ауыстырылған $R_{16}L_{16}$ кірме блогы бар кері шифрлау процесі үшін 1-итерацияда - K_{16} кілті, 2-де - K_{15} кілті т.с.с. пайдаланылады. 16-итерацияда K_1-кілті пайдаланылады. Итерацияның ең соңғы қадамында L_0 және R_0 тізбектері алынады. Олар 64 биттік L_0R_0 - тізбегіне конкатенцияланады. Содан соң осы тізбектегі 64 бит IP матрицасына сәйкес орын ауыстырылады. Бұл түрлендіру нәтижесінде бастапқы биттер тізбегі алынады (кері шифрланған 64 биттік мән).

$f(R_{i-1},K_i)$-функциясының мәндерін есептеу үшін мыналар пайдаланылады:

- E функциясы (32 биттен 48 кеңеюі);
- S_1, S_2, …, S_8 функциясы (6 биттік санды 4 биттік санға алмастыру);
- P функциясы (32 биттік тізбектегі биттердің орын ауыстыруы).

2.4-кесте

IP бастапқы орын ауыстыру матрицасы

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

2.5-кесте

IP^{-1} кері ауыстыру матрицасы

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

Осы функциялардың анықтамасын келтірелік. f - шифрлау функциясының аргументтері R_{i-1}(32 бит) және K_i (48 бит) болып табылады. E (R_{i-1}) функциясының нәтижесі - 48 биттік сан. 32 биттен 48 битке дейін кеңейтуді орындайтын (32 биттік блокты қабылдап, 48 биттік блокты тудырады) E кеңейту функциясы 2.7-кесте бойынша анықталады.

2.7-кестеге сәйкес $E(R_{i-1})$: алғашқы үш биті – 32, 1 және 2 биттер, ал соңғы - 31, 32, 1. Алынатын нәтиже (оны $E(R_{i-1})$ деп белгілейік) екі модулі бойынша K_i кілтінің ағымдағы мәнімен қосылады (XOR операциясы), содан соң 6 биттік B_1, B_2, ..., B_8 блоктарына бөлінеді: $E(R_{i-1}) \oplus K_i = B_1 B_2 \ldots B_8$.

2.6-кесте

Матрица элементтерінің байланысы

IP^{-1} матрица элементі	IP матрица элементі
40	01
8	02
48	03
16	04
56	05
…	…

2.7-кесте

Е кеңейту функциясы

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Әрі қарай бұл блоктардың әрқайсысы, 4 биттік мәні бар S_1, S_2, …, S_8 - функция-матрицалар элементтерінің нөмірі ретінде пайдаланылады (2.8-кесте).

S-блоктар DES шифрының криптоберіктілігіне негізгі үлес қосатын сыңары болып саналады. Сегіз S-блоктың әрқайсысы 4 қатардан және 8 бағаннан тұрады. S-блокқа келіп түсетін 6 бит ауыстыру үрдісін іске асыру үшін, матрицаның қай қатарын және қайсы бағанын пайдалану керектігін анықтайды. Бірінші және алтыншы биттер қатардың нөмірін, ал қалған биттер – бағанның нөмірін көрсетеді.

S_j матрицасында элементті таңдау ерекше түрде өтеді. S_j-матрицасының кірісіне 6 биттік $B_j=b_1\ b_2\ b_3\ b_4\ b_5\ b_6$ блок кірсін делік. Мысалы, егер S_1 матрицасының кірісіне $B_1=100110$ алты биттік блок түссе, онда $b_1b_6=10_{(2)}=2_{(10)}$ екі биттік сан

S_1 матрицасының нөмірі екінші қатарын, ал $b_2b_3b_4b_5=0011_{(2)}=3_{(10)}$ төрт биттік сан – нөмірі 3 бағанын көрсетеді. Бұл S_1 матрицасындағы B_1=100110 блогы нөмірі 2 жол мен нөмірі 3 баған қиылысқан жеріндегі элементі, яғни $8_{(10)}=1000_{(2)}$ таңдап алады. Сәйтіп, алты биттік B_1, B_2, …, B_8 блоктар жиынтығы S_1 ,S_2, …, S_8 әр матрицасындағы 4 биттік элементті таңдап алуды қамтамасыз етеді. Сонда $S_1(B_1)S_2(B_2)…S_8(B_8)$ блоктардан 32 биттік блок алынады.

2.8-кесте

S_1 ,S_2, ..., S_8 түрлендіру функциялары

		Баған нөмірі																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Жол нөмірі	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S_1
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
	2	4	1	4	8	13	6	2	11	15	12	9	7	3	10	5	0	
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S_2
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	S_3
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	S_4
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	S_5
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	S_6
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6	
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	S_7
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	S_8
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Бұл 32 биттік блоктар, биттердің орын алмастыру P-функциясы бойынша түрлендіріледі (2.9-кесте). Бұл P-функция осы кестеге сәйкес орын ауыстыра отырып, S-блоктар шықпасындағы төрт биттік элементтерден тұратын 8 топты 32 биттік жолға (қатарға) түрлендіреді. Сонымен, шифрлау функциясы мынадай: $f(R_{i-1}, K_i) = P(S_1(B_1), …, S_8(B_8))$.

2.9-кесте

Биттерді орын ауыстыру Р-функциясы

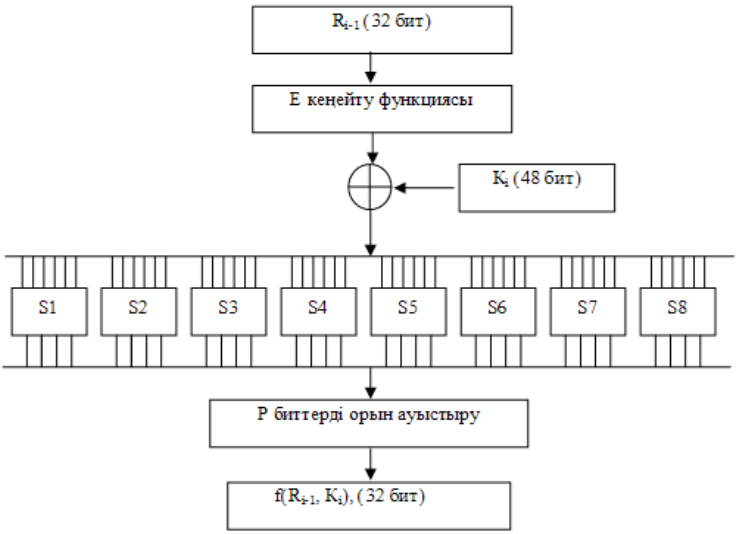
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

2.10-кесте

Кілтті бастапқы дайындау G функциясы (1-орын ауыстырып іріктеме)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Шифрлау функциясының мәнін есептеу үшін Е кеңейту функциясы, сегіз $(S_1, S_2, ..., S_8)$ S-блоктарды түрлендіруден құрастырылған S түрлендіру және Р орын ауыстыру қолданылады. f-шифрлау функциясының алгоритмі 2.22-суретте көрсетілген. f-шифрлау функциясының аргументтері – вектор R_{i-1} (32 бит) және вектор K_i (48 бит) [1].



2.22-сурет. F функциясының мәнін есептеу сұлбасы

Е функциясы R_{i-1} 32 биттік вектордың оның кейбір биттерін қайталау арқылы 48 биттік $E(R_{i-1})$ векторын алуға (кеңейтуге) мүмкіндік береді. Бұл кездегі $E(R_{i-1})$ векторындағы биттердің орналасуы 2.7-кестеде келтірілген.

Раундтық (итерациялық) кілттерді қалыптастыру

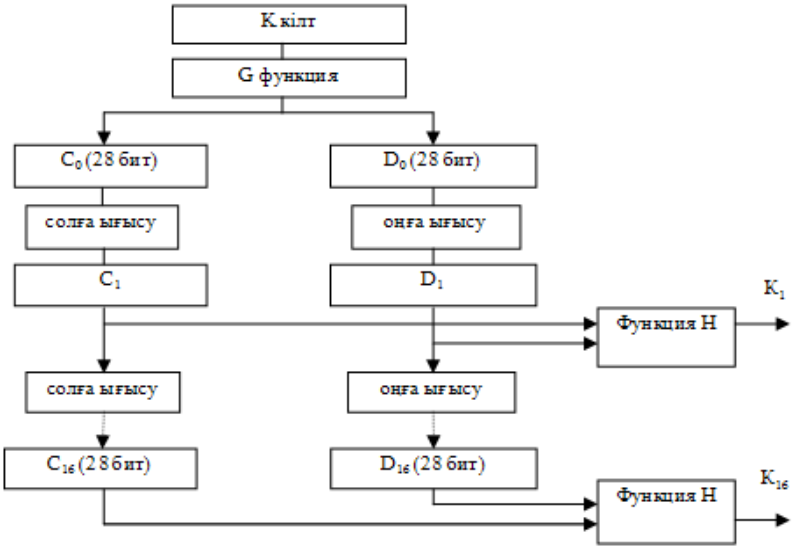
Алдымен пайдаланушы кездейсоқ 56 биттен тұратын бастапқы К кілтін таңдап алады. Әрбір итерация сайын K_i кілтінің (ұзындығы 48 бит) жаңа мәні пайдаланылады. K_i кілтінің жаңа мәні бастапқы К кілтінен есептеледі (2.23-сурет). Бастапқы К кілті есебінде 64 биттік блокты алуға болады. Оның 8 биті (8, 16, 24, 32, 40, 48, 56, 64 орындарында орналасқан) жұптық бақылау биттері деп аталады. Бақылау биттерінің мәні (0 немесе 1) бастапқы кілттің әрбір байтында бірлердің саны тақ болатындай етіп алынады. Бұл – кілттерді алмастыру және сақтау кезінде қателерді анықтау үшін қолданылады. Ұзындығы 48 биттік K_i кілттерін құрастыру үшін бастапқы К кілтінен бақылау биттерін алып тастап, қалған 56 бит 2.10-кестеге сәйкес түрлендіріледі (кілтті алдын ала дайындайтын G функциясы пайдаланылады).

2.10-кесте екі бөлікке бөлінген. G(K) түрлендіру нәтижесі әрқайсысы 28 биттік екі C_0 және D_0 бөлігіне бөлінген. G-матрицасының бірінші 4 жолы C_0 тізбегінің биттері қалай таңдалатынын анықтайды (C_0 тізбегінің бірінші биті шифр кілтінің 57-биті, содан кейінгісі 49- биті, т.с.с, ал соңғы биттері - кілттің 44- және 36-биттері болады). G матрицасының келесі 4 жолы D_0 тізбегінің биттері қалай таңдалатынын анықтайды (яғни D_0 тізбегі шифр кілтінің 63, 55, 47, …, 12, 4 биттерінен тұрады). Кестеде келтірілгендей, C_0 және D_0 тізбегін генерациялау үшін шифр кілтінің 8, 16, 24, 32, 40, 48, 56, 64 биттері пайдаланылмайды.

C_0 және D_0 анықталған соң, рекурсивті түрде C_i және D_i анықталады (i=1, 2, …, 16). Егер C_{i-1} және D_{i-1} анықталған болса, онда C_i және D_i кілттері итерация нөміріне байланысты бір

немесе екі битке циклдық түрде солға ығыстыру операциясы негізінде анықталады (2.11-кесте).

Итерацияның әр қадамында анықталатын K_i кілті C_iD_i блогының биттерінен (2.12-кестеге сәйкес) іріктеп алынған 48 биттен тұрады. Басқаша айтқанда, K_i кілті мынаған тең $K_i=H(C_iD_i)$, мұндағы H-функциясы кілтті өңдеуді аяқтайтын матрица арқылы анықталады. K_i кілтінің 1- биті C_iD_i тізбегінің 14- биті, екінші – 17-биті, 47 биті болып C_iD_i-дің 29 биті, ал 48-шісі – 32 биті. 2.12-кестеден көрініп тұрғандай, K_i кілтінің құрамына C_iD_i-дің 56 битінің 8 биті (9, 18, 22, 25, 35, 38, 43, 54) кірмеген.



2.23-сурет. K_i кілттерін есептеу алгоритмінің сұлбасы

2.11-кесте

Кілтті есептеуге арналған S_i ығысу кестесі

Итерация немірі, i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Ығысусаны (бит)	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

2.12-кесте

Кілтті өңдеуді аяқтайтын H-функциясы
(2-орын ауыстырып іріктеу)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

DES алгоритмінің бірнеше жұмыс істеу режімі бар:

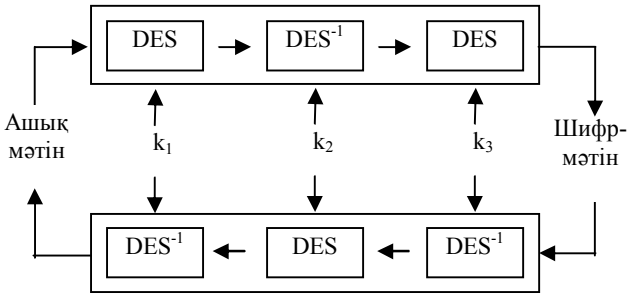
- электрондық кодтық кітап (ECB, Electronic Code Book;
- блоктар ілінісу (CBC, Cipher Block Chaining);
- шықпа бойынша кері байланыс (OFB, Output Feed Back);
- шифрмәтін бойынша кері байланыс (CFB, Cipher Feed Back).

DES алгоритмінде аталған режімдерден басқа, m-биттік кері байланыс режімі бар. Бұл режім m-биттік ($1 \le m \le 64$) блоктармен жұмыс істейді.

Қазіргі заманғы көптеген алгоритмдер үшін кілттің ұзындығы 56 бит болғаны жеткіліксіз болып жүр. Сондықтан DES алгоритмін қолдана отырып, екі және үш қайтара шифрлау нұсқалары әзірленген [1, 37].

Кейбір жағдайда M ашық мәтін блогы k_1 және k_2 екі кілттің көмегімен 3 рет шифрланады. Шифрлау теңдеуі мынадай: $C=E_{k1}(D_{k2}(E_{k1}(M)))$. Бұл сұлбаға кері шифрлау D_{k2} операциясының енгізілуі, осы сұлбаның DES алгоритмінің стандарттық сұлбасымен үйлесімдігін қамтамасыз етеді ($k_1=k_2$ кезде).

Үш еселі шифрлау кезінде әр түрлі үш кілт (168 бит) пайдаланылады және стандарттық үдеріс 3 рет қайталанады. Шифрлау теңдеуі мынадай: $C=E_{k3}(E_{k2}(E_{k1}(M)))$. Осылайша DES беріктілігін арттыру үшін әзірленген алгоритмнің жаңа нұсқасы "үш еселі DES" (***Triple DES, 3DES)*** деп аталады (2.24-сурет).



2.24-сурет. “Үш еселі DES” (3DES) сұлбасы

Үш еселі DES шифрлаудың түрлері:

- DES-EEE3: үш рет үш әр түрлі кілттермен шифрланады;
- DES-EDE3: үш әр түрлі кілттермен “шифрлау-кері шифрлау-шифрлау” үш DES-операциясы орындалады;
- DES-EEE2 және DES-EDE2: алдындағыларға ұқсайды, бірақ бірінші және үшінші операцияларда бірдей кілт қолданылады.

2.1.7. RIJNDAEL шифрлау алгоритмі

Rijndael – бұл «Квадрат» сәулеті бар блоктық шифр. Блоктардың ұзындығы айнымалы, кілттердің ұзындығы әр түрлі. Блок пен кілттің ұзындығы бір біріне тәуелсіз 128, 192 немесе 256 бит болуы мүмкін. AES стандарты ретінде блоктың ұзындығы 128 битке тең Rijndael варианты қабылданған [18, 31, 34, 37].

1) Деректер блогының пішімі және раундтық түрлендіру

Криптографиялық түрлендірулер барысында бастапқы және шифрланған деректердің блоктары, сондай-ақ шифрлау үрдісінің барлық аралық нәтижелері байттардың тік бұрышты массивтері түрінде сақталады. Криптоалгоритм кезінде орындалатын түрлендірулердің аралық нәтижелері *қалып-күй*

(State) деп аталады. Қалып-күйді, байттардың тік бұрышты массиві түрінде көрсетуге болады (2.25а-сурет). Мұндағы a_i – деректер блогының байты, ал әрбір баған – бір 32-разрядтық сөз.

Блоктың ұзындығы 128-битке тең болған жағдайда, бұл 16-байттық массив (2.25б-сурет) 4 қатардан және 4 бағаннан тұрады (әрбір қатар, әрбір баған 32-разрядтық сөз ретінде қарастырылады). Кіріс деректері s_{00}, s_{10}, s_{20}, s_{30}, s_{01}, s_{11}, s_{21}, s_{31}, … қалып-күй байттары ретінде көрсетіледі. Шифрлаудың аяқталуынан кейін, шығыс деректері қалып-күй байттарынан осы тәртіп бойынша құрылады. Жалпы жағдайда N_b бағандар саны, блок ұзындығын 32-ге бөлгендегі санға (бөліндінің мәніне) тең. Мәселен, блоктың ұзындығы 128-битке тең болғанда: $N_b=128/32=4$.

Шифрлау кілті де төрт қатарлы тік бұрышты массив түрінде ($N_k=4$) көрсетілген (2.25в-сурет). N_k бағандар саны кілт ұзындығын 32-ге бөлгендегі бөліндінің мәніне (санға) тең. Мұндағы s_{ij} және k_{ij} – і-ші қатар мен j-ші бағанның қиылысуындағы *State* массивінің және кілттің байттары.

a_0	a_4	a_8	a_{12}
a_1	a_5	a_9	a_{13}
a_2	a_6	a_{10}	a_{14}
a_3	a_7	a_{11}	a_{15}

а (State)

s_{00}	s_{01}	s_{02}	s_{03}
s_{10}	s_{11}	s_{12}	s_{13}
s_{20}	s_{21}	s_{22}	s_{23}
s_{30}	s_{31}	s_{32}	s_{33}

ә

k_{00}	k_{01}	k_{02}	k_{03}
k_{10}	k_{11}	k_{12}	k_{13}
k_{20}	k_{21}	k_{22}	k_{23}
k_{30}	k_{31}	k_{32}	k_{33}

б

2.25-сурет. *State* массиві және деректер мен кілт блоктарын көрсетімдеу ($N_b=4$, $N_k=4$)

Стандартта кілттердің барлық үш ұзындығы анықталған – 128, 192 және 256 бит, яғни 4, 6 және 8 32-разрядтық сөздер (немесе бағандар). RIJNDAEL алгоритмінде N_r раундтар саны N_b және N_k мәндеріне байланысты (2.13-кесте).

2.13-кесте

N_r раундтар санының N_k кілт және N_b блок ұзындығына тәуелділігі

	$N_b = 4$	$N_b = 6$	$N_b = 8$
$N_k = 4$	10	12	14
$N_k = 6$	12	12	14
$N_k = 8$	14	14	14

Раунд төрт түрлі түрлендіруден тұрады:

1) SubBytes() – байттарды ауыстыру – S-блоктарда кестедегі байттарды ауыстыру;

2) ShiftRows() – қатарларды жылжыту – *State* массивінің қатарларын әр түрлі байттар санына жылжыту;

3) MixColumns() – бағандарды араластыру – қалып-күйдің бағандарын үшінші дәрежелі g(x) көпмүшесіне x^4+1 модулі бойынша көбейту;

4) AddRoundKey() – раундтық кілтпен қосу – қалып-күйге раундтық кілттің ағымдағы мәнін XOR арқылы қосу.

SubBytes() процедурасы S-блоктарының көмегімен сызықсыз түрлендіруді жүзеге асырады. Мұндай түрлендіру дифференциалдық, сызықтық және криптоанализдің қазіргі заманғы басқа да әдістерін пайдалануға мүмкіндік бермейді. *ShiftRows() және MixColumns()* процедураларын қолдану статистикалық байланысты қалқалау (жасыру) мақсатымен жүргізілетін блок символдарының өзара араласу дәрежесінің жоғары болуына кепілдік береді. Деректерді шифрлау *AddRoundKey()* процедурасы кезінде XOR (яғни $\oplus$) арқылы іске асырылады.

Байттарды ауыстыру (SubBytes). SubBytes() түрлендіруі қалып-күйдің әрбір байтымен тәуелсіз түрде орындалатын, байттардың сызықсыз орын ауыстырылуы болып табылады. Түрлендіру келесі теңдеулерге сәйкес жүргізіледі:

$b_i' = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$,

мұнда, $c_0 = c_1 = c_5 = c_6 = 1$, $c_2 = c_3 = c_4 = c_7 = 0$, b_i және b'_i – і-ші биттің бастапқы және түрлендірілген мәні, $i = \overline{0,7}$.

Бұл түрлендіруді былайша орындау болады:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

0 мен 255 аралығындағы әрбір байт үшін осы түрлендіру алдын ала саналып, S кестесіне ендіріліп қойылған (2.14-кесте). Қалып-күйдің барлық байттарына көрсетілген S-блоктың қолданылуы *SubBytes(State)* деп белгіленеді. {xy} байтын түрлендіру кезіндегі S-блоктың жұмыс істеу логикасы 2.14-кестеде көрсетілген. Мәселен, {53} байтты түрлендіру {ed} нәтижесі 5-қатар мен 3-бағанның қиылысында орналасқан.

2.14-кесте

S-блок ауыстыру кестесі

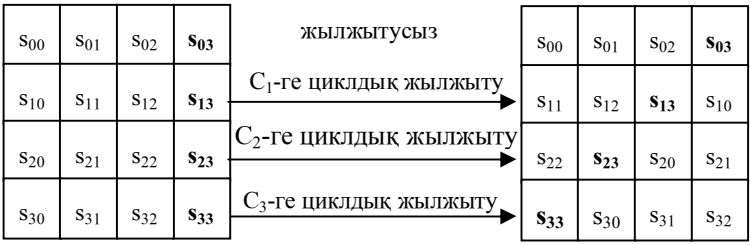
x	y															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Қатарларды жылжыту (ShiftRows). Қалып-күйдің соңғы 3 қатары циклдық түрде бірнеше байтқа солға жылжытылады: 0-қатар жылжытылмайды, 1-қатар C_1 байтқа, 2-қатар – C_2 байтқа, 3-қатар – C_3 байтқа жылжытылады. C_1, C_2, C_3 жылжыту мәндері RIJNDAEL-де N_b блоктың ұзындығына байланысты (2.15-кесте). Қалып-күйдің соңғы үш қатарын жылжыту операциясын *ShiftRows(State)* деп атайды. Мәселен, ұзындығы 128-битке тең (яғни, $\mathbf{N_b}$=4) блок үшін: C_1=1, C_2=2 және C_3=3. Түрлендірудің қалып-күйге әсері 2.26-суретте көрсетілген.

2.15-кесте

Блок ұзындығына байланысты C_1, C_2, C_3 мәндері

N_b	C_1	C_2	C_3
4	1	2	3
6	1	2	3
8	1	3	4



2.26-сурет. ShiftRows-нің қалып-күй қатарларына әсері

Бағандарды араластыру (MixColumns). Бұл түрлендіруде қалып-күйдің бағандары $GF(2^8)$ көпмүшелері ретінде қарастырылып, $x^4 + 1$ модулі бойынша g(x) көпмүшесіне көбейтіледі. g(x) көпмүшесінің түрі:

$g(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$.

Сөйтіп, *MixColumns* түрлендіруі әрбір s_c бағанына мына төмендегі ереже бойынша әсер етеді

$s_c(x) = g(x) \cdot s_c(x) \bmod (x^4 + 1), \qquad c = 0, 1, 2, 3.$

Бұл операцияны матрица түрінде былайша жазуға болады:

$$\begin{bmatrix} s'_{0c} \\ s'_{1c} \\ s'_{2c} \\ s'_{3c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0c} \\ s_{1c} \\ s_{2s} \\ s_{3c} \end{bmatrix}, \qquad 0 \le c \le 3,$$

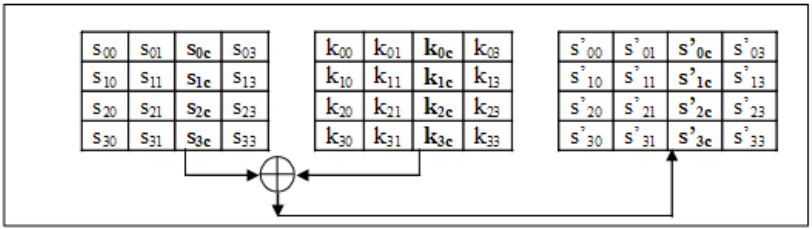
мұндағы c – *State* массивінің баған нөмірі.

Осылай көбейту нәтижесінде баған байттары s_{0c}, s_{1c}, s_{2c}, s_{3c} мынадай байттарға ауыстырылады:

$$s'_{0c} = (\{02\} \bullet s_{0c}) \oplus (\{03\} \bullet s_{1c}) \oplus s_{2c} \oplus s_{3c},$$
$$s'_{1c} = s_{0c} \oplus (\{02\} \bullet s_{1c}) \oplus (\{03\} \bullet s_{2c}) \oplus s_{3c},$$
$$s'_{2c} = s_{0c} \oplus s_{1c} \oplus (\{02\} \bullet s_{2c}) \oplus (\{03\} \bullet s_{3c}),$$
$$s'_{3c} = (\{03\} \bullet s_{0c}) \oplus s_{1c} \oplus s_{2c} \oplus (\{02\} \bullet s_{3c}).$$

Бұл операцияның қалып-күйдің барлық төрт бағанына қолданылуы *MixColumns(State)* деп көрсетіледі.

Раундтық кілттің қосылуы (AddRoundKey). Бұл операцияда раундтық кілт қалып-күйге XOR (екі модулі бойынша қосу операциясы) арқылы қосылады. Раундтық кілтті қалып-күйге XOR-мен қосу арқылы түрлендіру *AddRoundKey (State, RoundKey)* деп белгіленеді (2.27-сурет).



2.27-сурет. Қалып-күй мен раундтық кілті XOR операциясымен қосу

2) Кілттерді қалыптастыру алгоритмі (Key Schedule)

Раундтық кілттер, шифрлау кілтінен кілттерді қалыптастыру алгоритмі *(Key Schedule)* көмегімен алынады. Оның екі сыңары бар: *кілтті кеңейту (Key Expansion)* және *раундтық кілтті таңдау (Round Key Selection)*. Алгоритмнің негізгі қағидаттары мынадай:

- раундтық кілттер биттерінің жалпы саны раундтар санын блоктың ұзындығына 1 қосып, көбейткендегі мәнге тең.

Мәселен, блок ұзындығы 128 бит болса, онда 10 раунд үшін раундтық кілттер биттерінің саны 1408-ге тең болады;

- шифрлау кілті *кеңейтілген кілтке* (Expanded Key) кеңейтіледі;

- раундтық кілттер, кеңейтілген кілттен былайша алынады: бірінші раундтық кілт алғашқы N_b сөздерден құралады, екінші – келесі N_b сөздерден, т.с.с.

Кілттің кеңейтілуі (Key Expansion). Rijndael шифрында кілттің ұзындығы 128, 192 немесе 256 бит болады. Раунд ішінде қолданылатын кілт элементінің ұзындығы, блок өлшемімен бірдей. Шифрлау раундтардың саны 10-14 аралығында жатады. Кілт элементтерінің массиві шифрлау кілтінен төменде сипатталған кілтті кеңейту процедурасы көмегімен алынады.

Кілтті кеңейту алгоритмі 32 биттік w[i] кілт сөздерімен жұмыс істейді. Олар төрт элементтен тұратын байт массивтері деп қарастырылады. Rijndael алгоритмінде кеңейтілген кілт $N_b(N_r+1)$ 4 байттық сөздерден тұратын w[i] массив болып табылады, i=0, 1, …, $N_b(N_r+1)$. AES алгоритмінде w[i] массиві $4(N_r+1)$ 4 байттық сөздерден тұрады, i=0, 1, …, $4(N_r+1)$.

Кілттерді қалыптастыру алгоритмі N_k мәніне тәуелді болады. Алғашқы N_k сөздер шифрлау кілтімен толтырылады (2.28-сурет). Әрбір w[i] келесі сөзі w[i-1] алдыңғы сөзімен және N_k орынға ілгері орналасқан w[i-N_k] сөзімен XOR операциясын орындау арқылы есептеледі:

$w[i] = w[i-1] \oplus w[i-N_k]$.

Орналасқан орындары N_k мәніне еселі сөздер үшін XOR операциясы алдында w[i-1]-ге түрлендіру қолданылады, содан соң тағы *Rcon* раундтық тұрақты қосылады. Түрлендіру екі қосымша функция көмегімен іске асырылады: 32-разрядтық сөзді $\{a_0a_1a_2a_3\} \rightarrow \{a_1a_2a_3a_0\}$ формуласы бойынша байт-байттық жылжыту орындайтын *RotWord()* функциясы және SubBytes() функциясының S-блогын қолдана отырып, байт-байттық ауыстыру орындайтын *SubWord()* функциясы. Rcon[j] мәні 2^{j-1} тең. Бұл жағдайда w[i] мәні:

$w[i] = \text{SubWord}(\text{RotWord}(w[i-1])) \oplus \text{Rcon}[i/N_k] \oplus w[i-N_k]$.

W_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	...
0 раундтық кілт				1 раундтық кілт				2 раундтық кілт					

2.28-сурет. Кілттің кеңейтілуі және раундтық кілтті таңдау

Раундық кілтті таңдау (Round Key Selection). Әрбір і раундтық кілт раундтық кілт массивінің W[N_bі]-дан W[N_b (і+1)]-ге дейінгі сөздерінен қалыптасады. 2.28-суретте кілттің кеңейтілуі және N_k=4 болғандағы раундтық кілтті таңдау келтірілген.

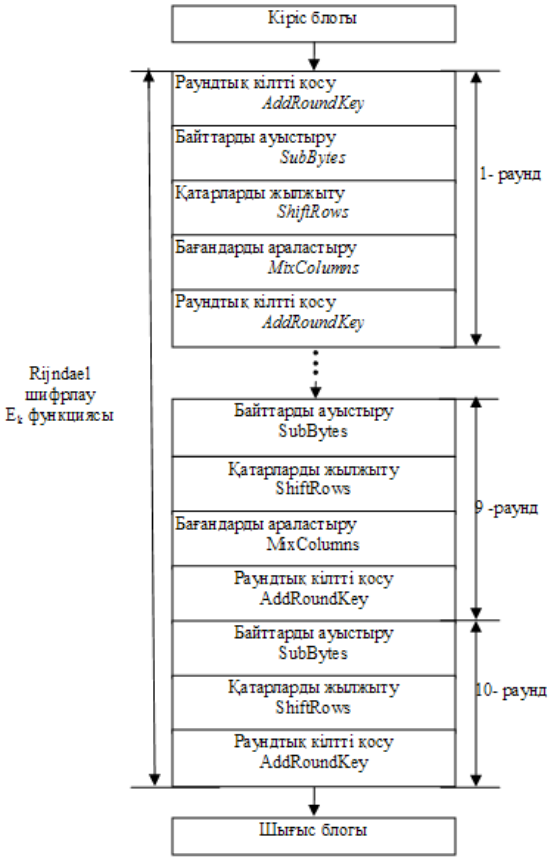
Кілттерді өндіру алгоритмін w[i] массивін қолданбай-ақ жасауға болады. Кеңейтілген кілт әрқашан шифрлау кілтінен жасалуы керек. Шифрлау кілтін таңдау кезінде оған ешқандай шектеулер қойылмайды.

3) Шифрлау функциясы

Rijndael шифры раундық кілтті бастапқы қосудан, N_r-1 раундтар мен (құрамына MixColumns() операциясы кірмейтін) қорытынды раундтан тұрады (2.29-сурет).

Алгоритмнің кірісіне State деректер блоктары енгізіледі. Түрлендірулер барысында блок құрамы өзгереді де, шығысында *State* блоктары түрінде ұйымдастырылған шифрмәтін пайда болады.

Бірінші раунд басталар алдында 2 модулі бойынша бастапқы шифрлау кілтімен қосындылау жүргізіледі. Одан кейін, кілттің ұзындығына байланысты 10, 12 немесе 14 раундтар бойы *State* байттар массивінің түрленуі жүзеге асырылады. Соңғы раундтың бастапқылардан ерекшелігі: бағандардағы байттарды араластыру *MixColumns()* функциясы бұл раундта қолданылмайды.



2.29-сурет. E_k шифрлау функциясының сұлбасы ($N_k=N_b=4$)

4) Кері шифрлау

Rijndael шифрлау және кері шифрлау процедуралары арасында біраз айырмашылықтар бар. Кілт элементтерінің кері шифрлау кезінде қолданылу реті, шифрлауда қолданылу ретіне қарама-қарсы болады.

Теріс кері шифрлау функциясы

Егер *SubBytes(), ShiftRows(), MixColumns()* және *AddRoundKey()* функцияларының орнына олардың инверстелген түрлендірулерін теріс (теріс тәртіпте) орындаса, онда теріс кері шифрлау функциясын құруға болады. Бұл кезде раундтық кілттерді пайдалану тәртібі шифрлау кезіндегіге қарама-қарсы болады.

InvSubBytes түрлендіруі. {xy} байтын түрлендіру кезіндегі инверстік S-блоктың жұмыс істеу логикасы 2.16-кестеде көрсетілген.

2.16-кесте

Инверстік S-блоктың ауыстыру кестесі (S^{-1})

	Y															
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

InvShiftRows түрлендіруі. Қалып-күйдің соңғы 3 қатары циклдық түрде бірнеше байтқа оңға жылжытылады: 0-қатар жылжытылмайды, 1-қатар C_1 байтқа, 2-қатар – C_2 байтқа және 3-қатар – C_3 байтқа жылжытылады. C_1, C_2, C_3 жылжыту мәндері RIJNDAEL-де N_b блоктың ұзындығына байланысты (2.15-кесте).

InvMixColumns түрлендіруі. Бұл түрлендіруде қалыпкүйдің бағандары $GF(2^8)$ көпмүшелері ретінде қарастырылып, $x^4 + 1$ модулі бойынша $g^{-1}(x)$ көпмүшесіне көбейтіледі. $g^{-1}(x)$ көпмүшесінің түрі:

$g^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$.

Бұл операцияны матрица түрінде былайша жазуға болады:

$$\begin{bmatrix} s'_{0c} \\ s'_{1c} \\ s'_{2c} \\ s'_{3c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0c} \\ s_{1c} \\ s_{2c} \\ s_{3c} \end{bmatrix}, \qquad 0 \le c \le 3$$

Осылай көбейту нәтижесінде мынадай байттарға ауыстырылады:

$$s'_{0c} = (\{0e\} \bullet s_{0c}) \oplus (\{0b\} \bullet s_{1c}) \oplus (\{0d\} \bullet s_{2c}) \oplus (\{09\} \bullet s_{3c})$$
$$s'_{1c} = (\{09\} \bullet s_{0c}) \oplus (\{0e\} \bullet s_{1c}) \oplus (\{0b\} \bullet s_{2c}) \oplus (\{0d\} \bullet s_{3c})$$
$$s'_{2c} = (\{0d\} \bullet s_{0c}) \oplus (\{09\} \bullet s_{1c}) \oplus (\{0e\} \bullet s_{2c}) \oplus (\{0b\} \bullet s_{3c})$$
$$s'_{3c} = (\{0b\} \bullet s_{0c}) \oplus (\{0d\} \bullet s_{1c}) \oplus (\{09\} \bullet s_{2c}) \oplus (\{0e\} \bullet s_{3c})$$

AddRoundKey() функциясы, онда қолданылатын XOR операциясының қасиеттері салдарынан, өзіне-өзі теріс болып саналады.

Сонымен, теріс кері шифрлау алгоритмінде операция-функциялардың қолданылуы, шифрлау алгоритміндегі операциялар тәртібіне қарама-қарсы болады. Бірақ, екеуінде де бірдей параметрлер (кеңейтілген кілт) пайдаланылады.

Тура кері шифрлау функциясы

Rijndael шифрлау алгоритмінің кейбір қасиеттері кері шифрлау кезінде (кейбір параметрлерді, нақтылап айтса, кеңейтілген кілтті өзгерту арқылы) функциялардың қолданылу тәртібін сақтап қалуға мүмкіндік береді. Тура кері шифрлау

алгоритмін қолдану, мына екі қасиеттің арқасында мүмкін болып тұр:

– *SubBytes()* және *ShiftRows()* функцияларының, сондай-ақ *InvSubBytes()* және *InvShiftRows()* функцияларының қолданылу тәртібі маңызды емес;

– *MixColumns()* операциясы кіріс деректеріне қатысты сызықтық операция болып табылады. Сонымен, функциялардың қолданылу тәртібі шифрлау алгоритміндегі сияқты, болатын кері шифрлау тәсілін жүзеге асыруға болады.

2.17-кестеде шифрлау процедурасы мен кері шифрлау процедурасының екі нұсқасы келтірілген. Түрлендіру тәртібі екі раундтық Rijndael үшін көрсетілгенмен, кестедегі мәлімет кез келген раунд санына жарамды.

2.17-кесте

Rijndael-дың екіраундтық нұсқасындағы түрлендірулердің тәртібі

Функция шифрования	**Теріс кері шифрлау функциясы**	**Тура кері шифрлаудың функциясы**
AddRoundKey	AddRoundKey	AddRoundKey
SubBytes	InvShiftRows	InvSubBytes
ShiftRows	InvSubBytes	InvShiftRows
MixColumns	AddRoundKey	InvMixColumns
AddRoundKey	InvMixColumns	AddRoundKey
SubBytes	InvShiftRows	InvSubBytes
ShiftRows	InvSubBytes	InvShiftRows
AddRoundKey	AddRoundKey	AddRoundKey

Rijndael алгоритмінде төрт шифрлау режімі көзделген: электрондық кодалық кітап; шифрмәтін блоктарын іліністіру; шифрмәтін бойынша кері байланыс; шықпа бойынша кері байланыс режімі.

2.1.8. ГОСТ 28147–89 стандарты

ГОСТ 28147-89 – классикалық блоктық құпия кілттік шифр. Бұл алгоритм 64 биттік блоктарды 256 биттік кілттің көмегімен шифрлауды жүзеге асырады [1, 3, 34].

Мұнда мынадай шартты белгілер пайдаланылады:

- L және R- биттер тізбектері;
- LR - L және R тізбектерінің конкатенциясы, мұнда R тізбегінің биттері L тізбегінің биттерінен кейін орналасады;
- $\oplus$ - биттерді өзара екі модулі бойынша қосу операциясы;
- [+] - екі 32 разрядтық екілік сандарды 2^{32} модулі бойынша қосу операциясы;
- [+]' - екі 32 разрядтық екілік сандарды $(2^{32}-1)$ модулі бойынша қосу операциясы.

Екі бүтін $\boldsymbol{a}$ және $\boldsymbol{b}$ сандары $0 \le \boldsymbol{a}, \boldsymbol{b} \le 2^{32}-1$, яғни $a=(a_{32}a_{31} \ldots a_2a_1)$ және $b=(b_{32}b_{31} \ldots b_2b_1)$ екілік түрде берілсін делік:

$a=a_{32}2^{31}+a_{31}2^{30}+ \ldots +a_2 2^1+a_1,$

$b=b_{32}2^{31}+b_{31}2^{30}+ \ldots +b_2 2^1+b_1.$

[+] және [+]' операциялары келесі ережеге сәйкес орындалады:

$a[+]b=a+b$, егер $a+b<2^{32}$,

$a[+]b=a+b-2^{32}$, егер $a+b(2^{32}$;

$a[+]'b=a+b$, егер $a+b<2^{32}-1$,

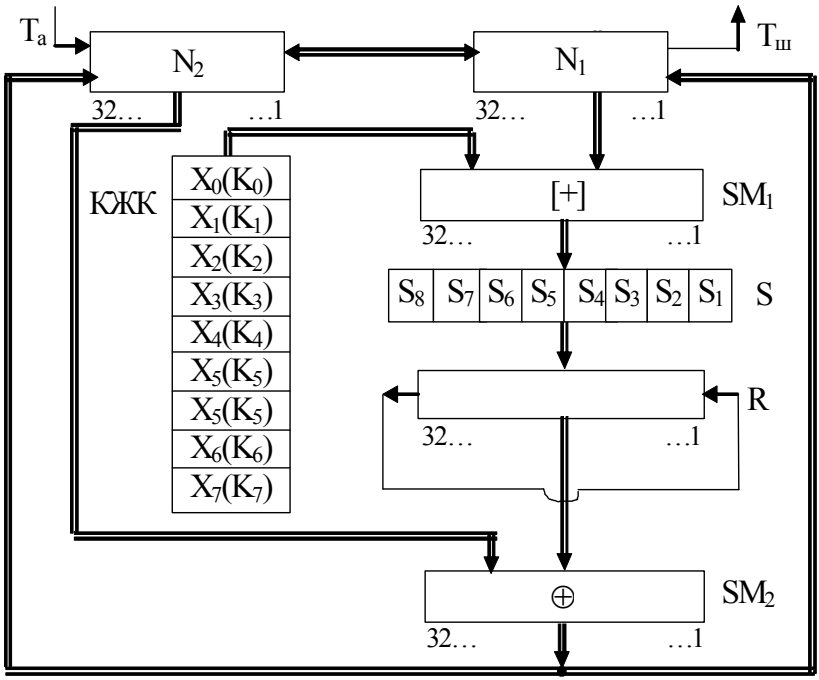
$a[+]'b=a+b-(2^{32}-1)$, егер $a+b \ge 2^{32}-1$.

Алгоритмнің төрт жұмыс режімі бар: қарапайым ауыстыру режімінде деректерді шифрлау, гаммалау режімінде деректерді шифрлау, кері байланысы бар гаммалау режімінде деректерді шифрлау және имитоендірмені жасау. Бірінші режім – тек кілттік ақпаратты шифрлауға арналған. Басқа деректерді шифрлау үшін, қалған екі гаммалау режімдері қолданылады. Имитоендірме – өңдіру режімі шифрланатын деректерді имитоқорғау (яғни, оларды кездейсоқ немесе рұқсат етілмеген, қасақана өзгертулерден қорғау) үшін керек.

Төменде тек қарапайым ауыстыру режімі ғана толығырақ қарастырылған.

1) Қарапайым ауыстыру режімі

Қарапайым ауыстыру режімінде деректерді шифрлау алгоритмін іске асыру үшін, жалпы криптожүйе блоктарының бір бөлігі ғана пайдаланылады (2.30-сурет).



2.30-сурет. Қарапайым ауыстыру режімін іске асыру сұлбасы

Шартты белгілер:

- N_1, N_2 – 32 разрядтық жинақтағыштар;
- $SM_1 – 2^{32}$ модулі бойынша 32 разрядтық қосындылауыш ([+]);
- SM_2 – екі модулі бойынша 32 разрядтық қосындылауыш (⊕);
- R – циклдік ығыстырудың 32 разрядтық регистрі;
- КЖҚ – кілттік жаттайтын құрылғы (256 биттік). Ол 32 разрядтық сегіз $X_0, X_1, \ldots, X_7$ жинақтағыштан тұрады;
- K_i – ішкілттер (subkey), раундық кілттер, i=0÷7;

- S – 8 ауыстыру (S_1, S_2, S_3, …, S_7, S_8) блогынан тұратын ауыстыру блогы;
- T_a – ашық деректер блогы (64 разряд);
- $T_ш$ – шифрланған деректер блогы (64 разряд).

♦ ***Ашық деректерді қарапайым ауыстыру режімінде шифрлау.*** Шифрлауға жататын ашық деректерді T_a деген 64 разрядтық блоктарға бөледі. T_a блоктарды шифрлау процедурасы 32 циклдан тұрады (i=1÷32). КЖҚ-ға К кілтінің 256 битін сегіз 32 разрядтық $К_i$ ішкілттері (раундық кілттер) түрінде енгізеді: $К=К_7К_6К_5К_4К_3К_2К_1К_0$.

$T_a=(a_1(0), a_2(0), …, a_{31}(0), a_{32}(0), b_1(0), b_2(0), …, b_{31}(0), b_{32}(0))$ биттер тізбегін 32-биттік екі тізбекке (сөзге) бөледі: b(0) және a(0). Бұл жерде b(0) сол немесе үлкен биттер, ал a(0) оң немесе кіші биттер.

Бұл тізбектерді N_1 және N_2 жинақтағыштарына шифрлаудың бірінші циклының басында енгізеді. N_1 жинақтағышының бастапқы толтырылымы мынадай: a(0) = ($a_{32}(0), a_{31}(0), …, a_2(0), a_1(0)$), ал N_2 жинақтағышының: b(0) = ($b_{32}(0), b_{31}(0), …, b_2(0), b_1(0)$).

64 разрядты блогы бар ашық деректерді шифрлау процедурасының бірінші циклын (j=1) келесі теңдеу арқылы жазуға болады.

$$\begin{cases} a(1) = f(a(0)\ [+]\ K_0) \oplus b(0), \\ b(1) = a(0). \end{cases}$$

Мұндағы a(1) – шифрлаудың бірінші циклынан кейінгі N_1 толтырылымы; b(1) – шифрлаудың бірінші циклынан кейінгі N_2 толтырылымы; f – шифрлау функциясы. f-функциясының аргументі – 2^{32} модулі бойынша алынған a(0) санымен $К_0$ санының қосындысы; a(0) – N_1 жинақтағышының бастапқы толтырылымы; $К_0$ – КЖҚ-ның $Х_0$ жинақтағышынан оқылған ішкілт. Мұндағы әрбір санның ұзындығы 32 битке тең. f-функциясында алынатын 32 разрядтық қосындыға екі операция жасалынады ($a(0)[+]К_0$).

Бірінші операция – *ауыстыру* деп аталады да, S ауыстыру блогы арқылы орындалады. S-блогы 0÷15 арлығындағы сандармен кездейсоқ түрде толтырылады. Бұл кестенің ішіндегі сандар шифрдың қосымша құпия параметрі болып табылады (мәселен, DES алгоритмінде осындай S-блоктар тұрақты және құпия емес).

S-блогы сегіз ауыстыру блогынан (S_1, S_2, …, S_8) тұрады. Оның әрбіреуінің жады көлемі 64 бит. SM_1-дан S ауыстыру блогына түсетін 32 разрядтық вектор 8 төрт разрядтық векторларға бөлінеді. Олардың әрқайсысы 4 разрядтық векторға түрленеді. Әр ауыстыру блогын диапазоны 0000…1111 болып келетін 16 төрт разрядтық екілік сандардың орын ауыстыру кестесі түрінде көрсетуге болады. Кіру векторы кестедегі қатар (жол) мекен-жайын көрсетеді. Ал бұл жолдағы сан, шығу векторы болып табылады. Содан соң 4 разрядтық шығу векторлары бірінен соң бірі 32 разрядтық векторға бірігеді. Бұл ауыстыру блоктары өте құпия сақталады.

Екінші операция кезінде S ауыстыру блогының шықпасынан алынған 32 разрядтық вектор солға циклдік (11 разрядқа) ығыстырылады. Циклдік ығыстыру R ығыстыру регистрі арқылы орындалады. Одан соң f шифрлау функциясының нәтижесімен N_2 жинақтағышының 32-разрядтық бастапқы b(0) толтырылымы SM_2 қосындылауышта екі модулі бойынша қосылады.

Қосындыда N_2 жинақтағышының 32 разрядтық бастапқы b(0) толтырылымы болады. Содан соң SM_2 шықпасынан алынған нәтиже (a(1) мәні) N_1 жинақтағышына жазылады, ал N_1-нің ескі мәні (a(0) мәні) N_2 жинақтағышына көшіріліп жазылады (яғни b(1)=a(0)). Бірінші цикл осымен аяқталады.

Келесі циклдар осыған ұқсас жүргізіледі. Яғни, екінші циклда КЖҚ-дан X_1 толтырылымы, яғни $К_1$ оқылады, үшінші циклда - $К_2$, ал сегізінші циклда $К_7$ оқылады. 9 циклдан 24-ке дейінгі циклдарда КЖҚ-дан ішкілттер, мына тәртіпте оқылады: $К_0$, $К_1$, …, $К_6$, $К_7$. Ал 25-тен 32-ге дейін КЖҚ ішкілттері кері

тәртіпте: K_7, K_6, …, K_1, K_0 оқылады. Сонымен, КЖҚ-дағы ішкілттерді таңдау мына тәртіпте жүргізіледі:
$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, \ K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7$, …,
$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, \ K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0$

32 циклда SM_2 қосындылауыштағы нәтиже N_2 жинақтағышына енгізіледі, ал N_1 жинақтағышында бұрынғы толтырылым сақталады. Шифрлаудың 32 циклының нәтижесінде алынған N_1 мен N_2 жинақтағыштардың толтырылымдары (T_a ашық деректер блогына сәйкес келетін) $T_ш$ шифрланған деректер блогы болып табылады.

Қарапайым ауыстыру режіміндегі шифрлау теңдеулерінің түрі мынадай:

j=1, …, 24 болғанда: $\begin{cases} a(j) = f\,(a(j-1)[+]\,K_{j-1(mod8)}) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases}$

j=25, …, 31 болғанда: $\begin{cases} a(j) = f\,(a(j-1)[+]\,K_{32-j}) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases}$

j=32 болғанда: $\begin{cases} a(32) = a(31) \\ b(32) = f\,(a(31)[+]\,K_0) \oplus b(31). \end{cases}$

Бұл жерде:

- $a(j)=a_{32}(j)$, $a_{31}(j)$, .., $a_1(j)$ - j-ші циклдан кейінгі N_1 жинақтағышының толтырылымы;
- $b(j)=b_{32}(j)$, $b_{31}(j)$, …, $b_{31}(j)$ - j-ші циклдан кейінгі N_2 жинақтағышының толтырылымы, $j=1\div 32$;
- $T_ш$ - шифрланған деректер блогы (64 разряд). N_1 және N_2 жинақтағыштарынан келесі тәртіппен шығарылады: алдымен N_1 жинақтағышының 1, …, 32 разрядтары, сосын N_2 жинақтағышының 1, …, 32 разрядтары, яғни кіші разрядтардан бастап $T_ш$ ($(a_1(32), a_2(32), …, a_{32}(32), b_1(32), b_2(32), …, b_{32}(32))$.

Ашық деректердің басқа блоктары, қарапайым ауыстыру режімінде осыған ұқсас шифрланады.

♦Қарапайым ауыстыру режімінде кері шифрлау. Қарапайым ауыстыру режімінде шифрды ашудың алгоритмін жүзеге асыратын криптографиялық сұлбаның түрі шифрлау кезіндегі сұлба сияқты болады. КЖҚ-ға шифрлау кезінде пайдаланған кілттің 256 биті енгізіледі. Кері шифрлауға дайындалған деректер әрбіреуінде 64 биттер бар $T_ш$ деген

блоктарға бөлінеді. Кез келген $T_ш$ блогын $T_ш = (a_1(32), a_2(32), \ldots, a_{32}(32), b_1(32), b_2(32), \ldots, b_{32}(32))$ N_1 жинақтағышының бастапқы мәні $a_{32}(32), a_{31}(32), \ldots, a_2(32), a_1(32)$, ал N_2 жинақтағышының бастапқы мәні $b_{32}(32), b_{31}(32), \ldots, b_1(32)$ болатындай етіп енгізеді.

Шифрды ашу, шифрлау алгоритміндей жүргізіледі. Тек оның айырмашылығы - $X_0, X_1, \ldots, X_7$ жинақтағыштарын толтыруда. КЖҚ-дан ішкілттер мына тәртіппен оқылады:

$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, \ldots,$
$K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$

Кері шифрлау тендеуінің түрі мынадай:

j=1, …, 8 болғанда: $\begin{cases} a(32\text{-}j) = f\,(a(32\text{-}j+1)[+]\,K_{j-1}) \oplus b(32\text{-}j+1) \\ b(32\text{-}j) = a(32\text{-}j+1) \end{cases}$

j=9, …, 31 болғанда: $\begin{cases} a(32\text{-}j) = f\,(a(32\text{-}j+1)[+]\,K_{32\text{-}j(\mathrm{mod}8)}) \oplus b(32\text{-}j+1) \\ b(32\text{-}j) = a(32\text{-}j+1) \end{cases}$

j=32 болғанда: $\begin{cases} a(0) = a(1) \\ b(0) = f(a(1)[+]\,K_0) \oplus b(1). \end{cases}$

32 циклдан кейін N_1 және N_2 жинақтағыштарында T_a ашық деректер блогы пайда болады: $T_a = (a_1(0), a_2(0), \ldots, a_{32}(0), b_1(0), b_2(0), \ldots, b_{32}(0))$.

Ол шифрланған $T_ш$ деректер блогына сәйкес келеді. Бұл жағдайда N_1 жинақтағышының күйі: $(a_{32}(0), (a_{31}(0), \ldots, a_2(0), a_1(0))$, ал N_2 жинақтағышының күйі мынадай болады: $(b_{32}(0), b_{31}(0), \ldots, b_2(0), b_1(0))$.

Шифрланған деректердің қалған блоктары да, осыған ұқсас түрде кері шифрланады.

2) Гаммалау режімі

Синхрожіберілім – криптографиялық түрлендіру алгоритмінің ашық параметрлерінің бастапқы мәндері.

♦ ***Ашық деректерді гаммалау режімінде шифрлау.*** Шифрлау алгоритмін іске асыратын криптографиялық сұлбада ашық деректер 64-разрядтық блоктарға бөлінеді: $T_o^{(1)}$, $T_o^{(2)}$, …, $T_o^{(i)}$, …, $T_o^{(m)}$, мұнда $T_o^{(i)}$ – і-ші ашық деректердің 64 разрядтық блогы, i=1÷m, m – шифрланатын деректердің көлемімен анықталады.

Бұл блоктар кезек-кезегімен гаммалау режімінде шифрланады. Гаммалау режіміндегі деректерді шифрлау теңдеуінің түрі мынадай: $T_ш^{(i)} = T_o^{(i)} \oplus Г_ш^{(i)}$, мұнда: $Г_ш^{(i)} = A(Y_{i-1}[+]C_2, Z_{i-1}[+]' C_1)$, i=1, …, m; $T_ш^{(i)}$ – шифрлаған мәтіннің 64 разрядтық і-ші блогы; $A(\overline{S})$ – қарапайым ауыстыру режіміндегі шифрлау функциясы; $\overline{S}$ – синхрожіберілім (64 разрядтық екілік тізбек); C_1, C_2 – 32 разрядтық екілік тұрақтылар; Y_i, Z_i – 32 разрядтық екілік тізбектер. Y_i, Z_i шамалары $Г_ш$ гаммасы қалыптасуы кезінде анықталады: $(Y_0, Z_0) = A(\overline{S})$, $(Y_i, Z_i) = Y_{i-1}$

♦***Гаммалау режімінде кері шифрлау.*** Кері шифрлау теңдеуі:

$$T_o^{(i)} = T_ш^{(i)} \oplus Г_ш^{(i)} = T_ш^{(i)} \oplus A(Y_{i-1}[+]C_2, Z_{i-1}[+]' C_1),\ i=1\div m.$$

Деректерді кері шифрлау – тек синхрожіберілім болған жағдайда ғана мүмкін болатынын айта кетуіміз керек. Синхрожіберілім – шифрдың құпия элементі емес, сондықтан оны ЭЕМ жадысында сақтауға да, байланыс арналары бойынша шифрланған деректермен бірге жіберуге де болады.

3) Кері байланысы бар гаммалау режімі

♦***Кері байланысы бар гаммалау режімінде ашық деректерді шифрлау .***

Кері байланысы бар гаммалау режіміндегі деректерді шифрлау теңдеуінің түрі төмендегідей:

$$T_ш^{(1)} = A(\overline{S}) \oplus T_o^{(1)} = Г_ш^{(1)} \oplus T_o^{(1)},$$
$$T_ш^{(i)} = A(T_ш^{(i-1)}) \oplus T_o^{(i)} = Г_ш^{(i)} \oplus T_o^{(i)},\ i=2\div m.$$

Итерациялық алгоритмнің бірінші қадамында $A(\overline{S})$ функцияның аргументі ретінде 64 разрядтық $(\overline{S})$

синхрожіберілім, ал келесі қадамдарында – шифрланған деректердің $T_ш^{(i-1)}$ блогы алынады.

♦ ***Кері байланысы бар гаммалау режімінде кері шифрлау***. Кері шифрлау теңдеуі:

$$T_o^{(1)} = A(\overline{S}) \oplus T_ш^{(1)} = Г_ш^{(1)} \oplus T_ш^{(1)},$$

$$T_o^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)} = A(T_ш^{(i-1)}) \oplus T_i^{(i)}, \quad i=2\div m.$$

Кері шифрлаудың жүзеге асырылуы шифрлау алгоритміне ұқсайды.

4) Имитоендірме құрастыру режімі

Имитоқорғау – шифрланған байланыс жүйесін ол арқылы жалған деректерді зорлап жіберуден қорғау.

Имитоендірме (алдамшы қоспа) – кілтті пайдаланып, ашық деректерден белгілі ереже бойынша (Р биттен тұратын) алынған және имитоқорғауды қамтамасыз ету үшін шифрланған деректерге қосылатын, бекітілген (тағайындалған) ұзындығы бар ақпарат бөлшегі (блок).

Имитоендірмені жасау үшін ашық деректерді 64 битті блоктарға бөледі. $T_o^{(i)}$ ашық деректердің $T_o^{(1)}$ бірінші блогы N_1 және N_2 жинақтағыштары жазылады. Содан кейін N_1 және N_2 жинақтағыштарының толтырылымдары қарапайым ауыстыру режімінде шифрлау алгоритмнің бастапқы 16 циклы бойынша түрлендіріледі. Имитоендірме жасау кезінде КЖҚ-да деректерді шифрлауға арналған 256 биттік кілт пайдаланылады.

Алушы жақ шифрмәтінді кері шифрлап, өзі имитоендірмені жасайды. Алынған және өзі есептеп шығарған имитоендірмелерді бір-бірімен салыстырып, алынған деректердің шынайылығын біледі.

2.1.9. Диффи-Хеллман алгоритмі

1976 жылы ұсынылған Диффи-Хеллман (Diffie-Hellman) алгоритмі, ашық кілттері бар алғашқы алгоритмдердің бірі болып саналады. Бұл алгоритм екі пайдаланушыға – олардың арасындағы байланыс қорғалмаған қатынас арнасы арқылы

ұйымдастырылғанына қарамастан, тек екеуіне ғана белгілі, ортақ құпия кілт құрастыруға мүмкіндік береді. Одан кейін бұл құпия кілт симметриялық криптожүйеде деректерді шифрлау үшін пайдаланылады.

Құпия кілттер құрастыру үшін Диффи-Хеллман алгоритмін қолдану тәртібін қарастыралық. Екі пайдаланушы (Алиса және Боб) қорғалмаған қатынас арнасын ұйымдастырады деп есептелік.

1-қадам. Екі пайдаланушы алдын ала N модулі (N жай сан болуы керек) және $g \in Z_N$ ($1 \le g < N-1$) қарапайым элементі туралы келіседі. g элементтері Z_N жиынының барлық нөл емес элементтері болып табылады, яғни $\{g, g^2, ..., g^{n-1} = 1\} = Z_N - \{0\}$. Бұл N және g бүтін сандары құпия емес, мәндері жүйенің барлық пайдаланушыларына ортақ сандар болып табылады.

Алиса мен Боб бір-бірінен тәуелсіз, k_A және k_B кездейсоқ үлкен бүтін сандарды таңдайды. Бұл сандар - *құпия кілттер* болып саналады.

2-қадам. Әрі қарай, Диффи-Хеллман алгоритмі қолданылады. Алиса $Y_A = g^{k_A} \pmod N$ ашық кілтін, ал Боб - $Y_B = g^{k_B} \pmod N$ ашық кілтін есептейді. Содан соң Алиса мен Боб есептелген Y_A және Y_B *ашық кілттер* мәндерімен, қорғалмаған арна бойынша алмасады.

3-қадам. Алиса мен Боб келесі салыстыруларды пайдаланып, ортақ құпия кілтті есептейді:

Алиса: $K = (Y_B)^{k_A} = (g^{k_B})^{k_A} \pmod N$

Боб: $K^* = (Y_A)^{k_B} = (g^{k_A})^{k_B} \pmod N$.

Мұнда $K=K^*$, өйткені $(g^{k_B})^{k_A} = (g^{k_A})^{k_B} \pmod N$. Деректерді шифрлау және кері шифрлау кезінде бұл K кілті, мәселен DES кілті ретінде пайдаланылады.

Екі пайдаланушы $C=E_K(M)=M^K \pmod N$ шифрлау түрлендіруін қолданып, хабарларды шифрлауы мүмкін. Хабар алушы кері шифрлауды орындау үшін $K * K^* \equiv 1 \pmod{N-1}$ салыс-

тыру көмегімен кері шифрлау кілтін табады, ал содан соң $M=D_K(C)=C^{K^*}(\text{mod } N)$ хабарды қалпына келтіреді.

Мысал. Модуль N=47, ал қарапайым элемент g=23 делік. А және В пайдаланушылары өз құпия кілттерін $k_A=12$ және $k_B=33$ деп таңдаған болсын.

Олар К ортақ құпия кілтті алу үшін алдын ала дербес ашық кілттердің мәндерін есептейді:

$Y_A=g^{k_A}=23^{12}(\text{mod } 47)=27$, $Y_B=g^{k_B}=23^{33}(\text{mod } 47)=33$.

Енді А және В пайдаланушылары Y_A және Y_B өз мәндерімен алмасып, ортақ құпия кілтін есептейді:

$K=(Y_B)^{k_A}=(Y_A)^{k_B}=33^{12}(\text{mod } 47)=27^{33}(\text{mod } 47)=25$.

Одан кейін, олар $K*K^*\equiv 1(\text{mod } N-1)$ салыстыруын қолданып, кері шифрлау құпия кілтін $K^*=35$ (mod 47) табады.

M=16 болғанда криптограмма $C=M^K=16^{25}(\text{mod } 47)=21$ болады, хабар алушы хабарды қалпына келтіреді: $M=C^{K^*}=21^{35}(\text{mod } 47)=16$.

2.2. Асимметриялық криптожүйелер

Деректерді криптографиялық қорғау жүйелерінің ішінде асимметриялық криптожүйелер – тиімді жүйелер қатарына жатады. Оларды ***ашық кілтті криптожүйелер*** деп те атайды. Мұндай жүйелерде деректерді шифрлау үшін бір кілт, ал оларды кері шифрлауға – басқа кілт қолданылады (асимметриялық деп аталу себебі де осыдан). Бірінші кілт ашық болады және ол деректерін шифрлаймын деген барлық пайдаланушылар, қолдану үшін жария етілуі мүмкін. Деректерді кері шифрлауға ашық кілт *(public key)* жарамайды [1, 3, 24, 31, 34, 37].

Шифрланып келген деректерді кері шифрлау үшін қабылдаушы жақ екінші кілтті пайдаланады. Ол – *құпия кілт (private key)* деп аталады. Сөйтіп, бұл криптожүйеде екі түрлі кілт қолданылады: K_B – жіберушінің ашық кілті, k_B – қабылдаушының құпия кілті. Құпия кілтті қорғалмаған арна арқылы жібермеу үшін кілттер генераторын хабар алушы

жағында орналастырған тиімді. k_B құпия кілтін белгілі K_B ашық кілт бойынша ашу, шешілмейтін мәселе болуы керек.

Асимметриялық криптожүйелерге тән ерекшеліктер:

– K_B ашық кілті мен С криптограммасы қорғалмаған арна бойынша жіберіледі, яғни қарсы жаққа K_B және С белгілі;

– шифрлау және кері шифрлау алгоритмдері E_B:M→C, D_B:C→M ашық болады.

Асимметриялық криптожүйелердегі ақпаратты қорғау k_B кілтінің құпиялығына тікелей байланысты. Асимметриялық криптожүйелерінің қауіпсіздігін қамтамасыз ететін талаптар:

1. Қабылдаушы үшін бастапқы жағдай негізінде (K_B, k_B) кілттер жұбын есептеп шығару қарапайым болу керек.

2. А жіберуші K_B ашық кілтін және М хабарын біліп, криптограмманы өте оңай есептеп шығара алады: $C = E_{Kв}(M) = E_B(M)$.

3. В қабылдаушы k_B құпия кілтін және С криптограммасын пайдаланып, бастапқы хабарды оңай қалпына келтіре алады.

$M=D_{Kв}(C)=D_B(C)=D_B[E_B(M)]$.

4. Қарсы жақ K_B ашық кілтін біліп, құпия k_B кілтін есептеп табу кезінде шешуге болмайтын есептеу проблемасына кез болады.

5. Қарсы жақ (K_B, С) жұбын біліп, бастапқы М хабарын есептеп табуды ешқандай жолмен шеше алмайды.

Ашық кілтті асимметриялық криптожүйелер тұжырымдамасында бір бағыттық функцияларды *(one-way function)* қолдану көзделген. X және Y – берілген кез келген жиын делік. Егер барлық $x \in X$ үшін y=f(x) оңай есептеп табуға болатын болса (мұнда $y \in Y$), онда f: X→Y функциясы бір бағытты деп саналады.

2.2.1. RSA криптожүйесі

RSA алгоритмін 1978 жылы Ronald Rivest, Adi Shamir және Leonard Adleman әзірлеген. RSA алгоритмі ашық кілтті алгоритмдердің алғашқысы болып саналады. Ол – деректерді

шифрлау режімінде де, электрондық (цифрлық) қолтаңба режімінде де жұмыс істей алады. Алгоритм сенімділігі үлкен сандарды жіктеу қиындығы мен дискретті логарифмдерді есептеу қиындығына негізделген [1, 3, 24, 34, 37].

RSA криптожүйесінде K_B ашық кілті, k_B құпия кілті, M хабары және C криптограммасы Z_N ({0, 1, 2, ..., N-1} бүтін сандар жиынына жатады. Мұнда N-модуль: N=P∗Q, ал P және Q – кездейсоқ, үлкен қарапайым сандар. Максимальды қауіпсіздікті қамтамасыз ету үшін P және Q сандарының ұзындығын бірдей қылып таңдап алып, оны құпияда ұстайды.

K_B ашық кілтін келесі шарттар орындалатындай етіп, кездейсоқ түрде таңдап алады:

$$1< K_B \le \varphi(N), \quad НОД(K_B, \varphi(N)=1, \quad \varphi(N)=(P-1)(Q-1),$$

мұндағы φ(N) – Эйлер функциясы. Эйлер функциясы 1-ден N-ге дейінгі аралықтағы N санымен өзара қарапайым, оң бүтін сандардың санын көрсетеді. Жоғарыда көрсетілген екінші шарт, K_B ашық кілті мен φ(N) Эйлер функциясы өзара қарапайым болу керек екенін көрсетеді. Әрі қарай, кеңейтілген Евклид алгоритмін пайдаланып, k_B құпия кілтін есептейді:

$$k_B * K_B \equiv 1 \pmod{\varphi(N)} \text{ немесе } k_B = K_B^{-1} \pmod{(P-1)(Q-1)} \quad (2.1)$$

Мұны іске асыруға болады, өйткені хабар алушы (P, Q) қарапайым сандар жұбын біледі және φ(N) функциясын оңай таба алады. k_B және N өзара қарапайым болуы керек. K_B ашық кілті деректерді шифрлауға, ал k_B құпия кілті, кері шифрлау үшін пайдаланылады.

C криптограммасын (K_B ашық кілт және M хабар жұбы арқылы) келесі формулаға сәйкес табуға болады:

$$C = E_{K_B}(M) = E_B(M) = M^{K_B} \pmod{N}. \quad (2.2)$$

Кері есепті, яғни C криптограммасын кері шифрлау есебін (k_B құпия кілт және C криптограмма жұбын пайдалана отырып) келесі формуламен шығаруға болады:

$$M = D_{k_B}(C) = D_B(C) = C^{k_B} \pmod{N}. \quad (2.3)$$

Кері шифрлау үрдісін былайша жазуға болады:

$$D_B(E_B(M)=(M \qquad (2.4)$$

(2.4)-ке (2.2) және (2.3) мәндерін қойсақ:

$$(M^{K_B})^{k_B} = M(\text{mod } N) \text{ немесе } M^{K_B k_B} = M(\text{mod } N) \qquad (2.5)$$

Сөйтіп, хабар алушы – екі параметрді қорғайды: k_B құпия кілтті және көбейтіндісі N модулін беретін (P, Q) сандар жұбын. Екінші жағынан, хабар алушы N модуль мәні мен k_B ашық кілтті ашады.

Қарсы жаққа K_B және N мәндері белгілі. Егер ол N санын P және Q көбейткіштеріне жіктей алса, онда "құпия жолды", яғни $\{P, Q, K_B\}$ үш санын білер еді. Әрі қарай $\varphi(N)=(P-1)(Q-1)$ Эйлер функциясының мәнін есептеп тауып, ол арқылы k_B құпия кілтінің мәнін анықтар еді. Бірақ, мәні өте үлкен N санын көбейткіштерге жіктеу мүмкін емес (егер P және Q сандарының ұзындығы 100 ондық таңбадан кем болмаса).

RSA криптожүйесінде шифрлау және кері шифрлау процедуралары

Бір пайдаланушы (*Алиса*) RSA криптожүйесін қолдана отырып, шифрланған түрде екінші пайдаланушыға (*Бобқа*) хабар жібергісі келеді делік. Жоғарыда көрсетілгендей, RSA криптожүйесін хабарды алушы құрастыруы керек, яғни Боб. Алиса мен Бобтың іс-әрекеттері былайша ұйымдастырылады.

1. Боб кез келген екі қарапайым үлкен мәнді P және Q сандарын таңдап алады.

2. Боб $N=P\cdot Q$ модулінің мәнін есептейді.

3. Боб $\varphi(N)=(P-1)\cdot(Q-1)$ Эйлер функциясын есептеп $1<K_B\le\varphi(N)$, $НОД(K_B,\varphi(N))=1$ шарты орындалатындай жағдайда, кездейсоқ түрде ашық кілт K_B мәнін таңдап алады.

4. Боб $k_B=K_B^{-1} \pmod{\varphi(N)}$ салыстыруды шешкенде кеңейтілген Евклид алгоритмін пайдалана отырып, k_B құпия кілттің мәнін есептеп шығарады.

5. Боб Алисаға (N, K_B) сандар жұбын қорғалмаған арна арқылы жібереді.

Егер Алиса Бобқа M хабарын жібергісі келсе, онда ол келесі қадамдарды орындайды.

6. Алиса бастапқы M ашық мәтінді блоктарға бөледі. Әрбір блок M_i = 0, 1, ..., N-1 сандарымен көрсетімделуі мүмкін.

7. Алиса M_i сандар тізбегі ретінде берілген мәтінді $C_i=M_i^{K_B} \pmod N$ формуласы бойынша шифрлап, C_1, C_2, C_3, ..., C_i, ... криптограммасын Бобқа жібереді.

8. Боб қабылданған C_1, C_2, C_3, ..., C_i, ... криптограммасын k_B құпия кілтін қолдана отырып, $M_i(C_i^{k_B} \pmod N$ формуласы бойынша кері шифрлайды.

Осының нәтижесінде бастапқы M мәтінін құрайтын M_i сандарының тізбегі алынады. RSA алгоритмін іс жүзінде қолдану үшін көп шығынсыз үлкен мәнді қарапайым сандарды генерациялау және K_B мен k_B кілттерінің мәндерін тез есептеу мүмкіндігі болуы керек.

Мәселен: ОТАН хабарын шифрлау (есептеуге оңай болу үшін мәні аса үлкен емес сандар пайдаланылды, бірақ іс жүзінде мәні аса үлкен сандар қолданылады).

Бобтың іс-әрекеттері.

1. P=5 және Q=11 сандарын таңдайды.

2. $N=P\cdot Q=5*11=55$ есептейді.

3. N=55 үшін Эйлер функциясын есептейді:

$\varphi(N)=\varphi(55)=(P-1)(Q-1)=4*10=40$

K_B ашық кілт ретінде $1\le K_B\le 40$, НОД(K_B,40)=1 шартты орындалатындай, кез келген санды таңдайды. Мейлі K_B =3 болсын.

4. $k_B=3^{-1}(\text{mod}40)$ салыстыруды шешкенде кеңейтілген Евклид алгоритмін пайдаланып, k_B құпия кілтінің мәнін есептейді. Шешімі – k_B=27.

5. Алисаға (N=55, K_B=3) сандар жұбын жібереді.

Алисаның іс-әрекеттері.

6. Шифрланған хабарды 0, ..., 41 бүтін сандар тізбегі ретінде көрсетімдейді. О әрпі 19 саны, Т – 24 саны, А – 1 саны,

Н – 17 саны болсын (2.1-сурет). Сонда ОТАН деген хабар 19, 24, 1, 17 сандар тізбегі болады, яғни M_1=19, M_2=24, M_3=1, M_4=17.

7. M_1, M_2, M_3, M_4 сандар тізбегі түрінде берілген мәтінді K_B=3, N=55 пайдалана отырып, $C_i = M_i^{K_B} \pmod N = M_i^3 \pmod{55}$ формуласы бойынша шифрлайды. Мынадай нәтиже алады:

$C_1=19^3 \pmod{55} = 6859 \pmod{55} =39$,

$C_2=24^3 \pmod{55} = 13824 \pmod{55} =19$,

$C_3=1^3 \pmod{55} = 1 \pmod{55} =1$,

$C_4=17^3 \pmod{55} = 4913 \pmod{55} =18$.

Бобқа мына криптограмманы жібереді C_1, C_2, C_3 , C_4 = 39, 19, 1, 18

Бобтың іс-әрекеттері.

8. k_B=27 құпия кілтін қолдана отырып, қабылдап алынған C_1, C_2, C_3, C_4 криптограммасын $M_i = C_i^{k_B} \pmod N = C_i^{27} \pmod{55}$ формуласы бойынша кері шифрлайды.

27 санын мына түрде көрсетуге болады: 27=16+8+2+1, яғни

27=2∗2∗2∗2+2∗2∗2+2+1.

Сонымен мынадай нәтиже алынады:

$M_1 = 39^{27} \pmod{55} = (((39)^2)^2)^2)^2 * ((39^2)^2)^2 * 39^2 * 39 \pmod{55} =$

$=(((39)^2)^2)^2)^2 * \mathrm{mod}55) * ((39^2)^2)^2 \pmod{55} * 39^2 \pmod{55} * 39 \pmod{55} =19$

$M_2 = 19^{27} \pmod{55} = (((19)^2)^2)^2)^2 * ((19^2)^2)^2 * 19^2 * 19 \pmod{55} =24$

$M_3 = 1^{27} \pmod{55} =1$

$M_4 = 18^{27} \pmod{55} = (((18)^2)^2)^2)^2 * ((18^2)^2)^2 * 18^2 * 18 \pmod{55} =17$

Сөйтіп, хабар алғашқы қалпына (ОТАН - 19,24,1,17) келтірілді.

Енді, RSA шифрлау жүйесімен жұмыс істеудің тағы бір мысалын келтірелік [1]. Айталық, біздің хабарымыз мынадай болсын: АСУ.

1. p және q (p≠q) екі құпия жәй сандардың мәндерін таңдау қажет. Айталық, p=17 және q=31 болсын.

2. n=p∗q=17*31=527 есептеледі.

3. Берілген формула бойынша Эйлер функциясын есептеу керек:

$$\varphi(\mathrm{n}) = f(p,q) = (p-1)(q-1) = (17-1)(31-1) = 480.$$

4. $e \cdot d = k \cdot f(p,q) + 1$ шартын қанағаттандыратын, іріктеп таңдау әдісін қолдана отырып e, k және d мәндерін есептеу қажет. Ашық (e) және құпия (d) кілттері өзара жай сандар болуы керек. Біздің жағдайымызда e=7, k=5, d=343.

Деректерді (n, e) ашық кілтімен шифрлау үшін мынадай әрекеттер орындалады:

Бастапқы M мәтіні блоктарға бөлініп, олардың M_i әрқайсысы 0 мен (n-1) аралығында жатқан сандардың тізбегі ретінде көрсетімделеді. Ол үшін АСУ сөзінің әріптері бес өлшемді екілік векторлармен кодтау керек.

Қазақ әліпбиінде А, С және У әріптері мынадай нөмірлі орындарда орналасқан: А әрпі 1-нөмірлі орында, С әрпі 23-орында, У әрпі 25-орында.

Сонда, А=1=(00001), С=23=(10111), У=25=(11001).

Олай болса, АСУ = (000011011111001) түрінде жазылады. Берілген 0…526 (n-нің мәніне сәйкес) аралықты ескере отырып, былайша бөле аламыз:

АСУ = (0000110111110), (01) = (M_1 = 446, M_2 = 1).

Алынған сандар тізбегін $C = E_k(M_i) = M_i^e (\mathrm{mod} n)$ формуласы бойынша шифрлау керек:

$$C_1 = E_k(M_1) = M_1^e (mod\, n) = 446^7 (mod\, 527) = 489.$$

$$C_2 = E_k(M_2) = M_2^e (mod\, n) = 1^7 (mod\, 527) = 1$$

Құпия (n, d) кілтімен осы деректерді кері шифрлау үшін $D_k(C_i) = C_i^d (\mathrm{mod}\, n)$ формуласы бойынша мынадай есептеуді орындау қажет:

$$D_k(C_1) = C_1^d (mod\, n) = 489^{343} (mod 527) = 446$$

$$D_k(C_2) = C_2^d (mod\, n) = 1^{343} (mod 527) = 1.$$

Әріптік жазбаға қайта оралып, кері шифрлағаннан кейін бастапқы АСУ сөзін алатын боламыз.

2.2.2. Полиг-Хеллман шифрлау сұлбасы

Полиг-Хеллман шифрлау сұлбасы RSA шифрлау сұлбасына ұқсас [31]. Ол симметриялық емес алгоритм түріне жатады. Өйткені, шифрлауға және кері шифрлауға әр түрлі кілттер пайдаланылады. Сонымен бірге, бұл сұлбаны ашық кілтті криптожүйелер сыныбына жатқызуға болмайды. Себебі, шифрлау және кері шифрлау кілттерінің бірін, екіншісі арқылы оңай есептеп табуға болады. Сондықтан, екі кілтті де (шифрлау және кері шифрлау) құпия түрде ұстау қажет.

С криптограммасы мен Р ашық мәтіні келесі өрнектермен анықталады: $C=P^e(\mathrm{mod}\ n)$, $P=C^d(\mathrm{mod}\ n)$, мұнда $e*d\equiv 1$.

RSA алгоритмінен бұл сұлбаның ерекшелігі – n саны екі қарапайым үлкен сандар арқылы анықталмайды; n саны құпия кілттің бір бөлігі болып қалады. Егер кім де кім е және n сандарының мәнін білсе, онда ол d мәнін есептеп, тауып алады. Егер е немесе d мәндерін білмесе, онда $e = \log_P C\ (\mathrm{mod}\ n)$ мәнін есептеуге мәжбүр болады. Бұл аса қиын мәселе екені белгілі.

2.2.3. Эль-Гамаль шифрлау сұлбасы

Эль-Гамаль криптожүйесі (1985 ж.) – Диффи-Хеллманның ашық кілттер әдісінің бір нұсқасы болып табылады. Іске асырылу тұрғысынан қарағанда RSA және Эль Гамаль (Taher ELGamal) стандарттарының арасында айтарлықтай айырмашылық жоқ. Олардың тек криптобериктілік жағынан елеулі өзгешіліктері бар. Эль Гамаль жүйесінің криптобериктілігі мынаған сүйенген: бүтін санның дәрежесін есептеп шығару жеңіл, бірақ белгілі бір сан алу үшін, берілген санды дәрежелеу керек болатын дәреже көрсеткішін табу қиын. Эль Гамаль шифрлау сұлбасы – шифрлау үшін де, цифрлық қолтаңба қою үшін де пайдаланылады [1, 3, 31, 34, 37].

Шекті өрістегі дискреттік логарифмдерді есептеу негізіндегі жүйе

Эль Гамаль шифрлау сұлбасының бұл түрі – шекті өрістегі дискреттік логарифмдерді есептеу күрделігіне негізделген.

Кілттер жұбын (ашық кілт - құпия кілт) генерациялау үшін алдымен үлкен мәнді P қарапайым саны және G үлкен бүтін саны (G<P) таңдап алынады. P және G сандары, пайдаланушылар арасында ашық таратылуы мүмкін. Содан соң кез келген кездейсоқ бүтін x саны (x<P) таңдап алынады. x саны – *құпия кілт* болып табылады. Әрі қарай $Y=G^x \bmod P$ есептеледі. Ал Y саны – *ашық кілт* болып саналады.

M хабарын шифрлау үшін k кездейсоқ бүтін саны (1<k<P-1) таңдап алынады. Бұл санды таңдағанда k және (P-1) сандары өзара қарапайым болу керек. Одан соң мына сандарды есептейді:

$a = G^k \bmod P$, $b = Y^k M \bmod P$.

(a, b) сандардың жұбы – шифрмәтін болып есептеледі. Шифрланған мәтіннің ұзындығы, бастапқы ашық M мәтін ұзындығынан екі есе екенін байқауға болады. (a, b) шифрмәтінді кері шифрлау үшін мынадай есептеу жүргізіледі:

$$M = b/a^x \bmod P. \qquad (2.7)$$

$a^x \equiv G^{kx} \bmod P$ және $b/a^x \equiv Y^k M/a^x \equiv G^{kx} M/G^{kx} \equiv M \pmod P$ болғандықтан, (2.7) өрнегі шынайы болады.

Мысал. P=17, G=2, құпия кілт x=5 деп алсақ, онда:

$Y = G^x \bmod P = 2^5 \bmod 17 = 32 \bmod 17 = 15$.

Сонымен, ашық кілттің мәні Y=15. “ӘКЕ” деген хабарды шифрлайық. M хабары M_1=2, M_2=13 және M_3=8 болады. Кездейсоқ сан ретінде k=3 деп аламыз. НОД(k, P-1)=1 екеніне көз жеткізелік. Шынында да, НОД(3, 16)=1. Енді a және b сандар жұбын есептейміз:

$a = G^k \bmod P = 2^3 \bmod 17 = 8 \bmod 17 = 8$,

$b_1 = Y^k M_1 \bmod P = 15^3 * 2 \bmod 17 = 6750 \bmod 17 = 1$.

$b_2 = Y^k M_2 \bmod P = 15^3 * 13 \bmod 17 = 43875 \bmod 17 = 15.$

$b_3 = Y^k M_3 \bmod P = 15^3 * 8 \bmod 17 = 27000 \bmod 17 = 4.$

Сонда мынадай шифрланған мәтін аламыз:

$(a, b_1) = (8, 1); (a, b_2) = (8, 15); (a, b_3) = (8, 4).$

Енді осы шифрмәтінді х құпия кілтін қолданып, кері шифрлаймыз:

$M_1 = b_1/a^x \bmod P = 1/8^5 \bmod 17.$

$M_2 = b_2/a^x \bmod P = 15/8^5 \bmod 17.$

$M_3 = b_3/a^x \bmod P = 4/8^5 \bmod 17.$

Осы өрнектерді мынадай түрде жаза аламыз:

$8^5 * M_1 \equiv 1 \bmod 17$ немесе $32768 * 2 = 1 \bmod 17;$ $M_1=2$

$8^5 * M_2 \equiv 15 \bmod 17$ немесе $32768 * 13 \equiv 15 \bmod 17;$ $M_2=13$

$8^5 * M_3 \equiv 4 \bmod 17$ немесе $32768 * 8 \equiv 4 \bmod 17;$ $M_3=8$

Кері шифрланған хабар $M_1=2$, $M_2=13$, $M_3=8$ екенін табамыз. Әдетте, Р ретінде үлкен мәнді бүтін қарапайым сан (512 ÷ 1024 бит) пайдаланылады.

Эллипстік қисықтар негізіндегі жүйе

Дискреттік логарифмдеуге негізделген кез келген криптожүйені эллипстік қисыққа жеңіл ауыстыруға болады. Жүйені құрудың негізгі қағидаты – $y=g^x \bmod p$ операциясын, Y=[x]G mod p операциясымен ауыстыру. Х дәрежесінің көрсеткіші q модулі бойынша келтірілген. q нүктелер жиынының қуатын қамтитын эллипстік қисықта [x] көбейткішімен де осы секілді әрекет орындалады, айырмашылығы тек мынада: у – сан, ал Y – нүкте, әдетте нүктеден санға ауысу талап етіледі. Мұндай ауысудың қарапайым тәсілі – нүкте абсциссасын қолдану [34].

Эллипстік қисықты қолдану мысалы ретінде Эль-Гамаль шифрын қарастырамыз.

Эллипстік қисықтағы Эл-Гамаль шифры

Желінің пайдаланушылары үшін $E_p(a,b)$ жалпы эллипстік қисық пен осы қисықтың G нүктесі таңдап алынсын делік: G,

[2]G, [3]G,…., [q]G түрлі нүктелер және q жәй саны үшін [q]G=Õ (Õ нүктесі эллипстік қисықтағы операцияларда нөл міндетін атқарады және онда мынадай шарттық белгі қолданылған: мәселен, R=P+P=[2]P) [34].

Әрбір U пайдаланушы c_U ($0<c_U<q$) санын таңдайды да, оны өзінің құпия кілті ретінде сақтайды. Одан кейін нәтиже пайдаланушының ашық кілті болып табылатын $D_U=[c_U]G$ қисықтың нүктесін есептейді. Қисықтың параметрлері мен ашық кілттердің тізімі желінің барлық пайдаланушыларына жіберіледі.

Мәселен, A пайдаланушы, B пайдаланушыға хабар жібергісі келді делік. Хабар $m<p$ саны түрінде көрсетілген болсын. A пайдаланушы мынадай әрекеттер тізбегін орындайды:

1) k кездейсоқ санын таңдайды, $0<k<q$;
2) $R = [k]G$, $P = [k]D_B = (x,y)$ мәндерін есептейді;
3) $e = mx \bmod p$ шифрлайды;
4) B пайдаланушыға (R,e) шифрмәтінін жібереді.

B пайдаланушы (R,e) шифрмәтінін алғаннан кейін:

1) $Q = [c_B]R = (x,y)$ мәнін есептейді;
2) кері шифрлайды (яғни, $m' = ex^{-1} \bmod p$).

Q нүктесінің x координатасы қаскөй үшін құпия болып қала береді, себебі ол k санын білмейді. Қаскөй k санын R нүктесінен есептеуге әрекет жасап көруі мүмкін. Бірақ, ол үшін оған қисықтағы дискреттік логарифмдеу мәселесін шешуге тура келеді, ал бұл мүмкін емес.

Бұл хаттаманы m саны ретінде (блоктық немесе ағындық шифр кезінде) құпия кілтті жіберу үшін қолдануға болады. Бұл жағдайда қисықтың параметрін log q саны, шифр кілтінің ұзындығынан шамамен екі есе артық болатындай етіп таңдаған жөн.

2.3. Хэш-функциялар мен цифлық қолтаңбалар

2.3.1. Хэштеу функциялары

Хэш-функция (hash function) – әр түрлі ұзындығы бар M хабарды (қатарды) тұрақты ұзындығы бар биттер тізбегіне түрлендіретін функция. Хэш-функцияның мәні, хабардың барлық символдарына тәуелді және хэш-мәні өзгермейміндей

етіп хабарды өңдеу-түзету мүмкін емес: хабарға енгізілген кез келген өзгерту, хэш-функция арқылы өңдеу барысында, оның шығысындағы хабардың өзгеруіне әкеледі. Хэш-функциялардың бір бағыттық қасиеті – оларды жіберушінің цифрлық қолтаңбасы бастапқы, күйде сақталған жалған (өзгертілген) хабар қалыптастыруға мүмкіндік бермейді [3, 22, 24, 31, 34, 37].

Message Digest алгоритмдері

Message Digest алгоритмдер тобына – MD2, MD4 және MD5 хэш-функциялары жатады. Бұл алгоритмдерді 1989 жылдары Ривест әзірлеген. Олардың бәрі кез келген ұзындықты мәтінді 128-биттік сигнатураға түрлендіреді [3, 37].

MD4 алгоритмінде мәтінді 512 модулі бойынша 448 битке тең ұзындыққа дейін толықтыру, мәтіннің 64 биттік көрсетімдегі ұзындығын қосып, 512-биттік блоктарды, Damgard-Merkle процедурасымен өңдеу көзделген. Осының нәтижесінде MD4 алгоритмінің шықпасында бастапқы тізбектің 128 биттік ұзындығы бар "сығындысы" алынады. MD4 алгоритмі 32 разрядты аппараттық тұғырнама үшін оңтайландырылған, әрі жылдам жұмыс істейді. Бұл алгоритмде әрбір 512 биттік блок әр түрлі үш циклға қатысады. MD4 алгоритмінде үш 32 биттік айнымалылармен істейтін үш функция қатысады. 32 биттік айнымалылардың бастапқы мәндері: H_1=′674523016′ H_2=′efcdab89′, H_3=′98badcfe′, H_4=′10325476′. Алгоритмнің жұмысы барысында ағымдағы (H_1, H_2, H_3, H_4) хэш-күйлер қадағаланып отырады [3, 37].

MD5 алгоритмде әрбір блок әр түрлі төрт циклға қатысатын болды. Сондықтан, MD5 алгоритмі MD4 алгоритміне қарағанда баяу жұмыс істейді. Енді ***MD5 хэштеу функциясын*** қарастырамыз. Ұзындығы b бит хабар бар деп есептелік. Мұндағы b – кез келген теріс емес бүтін сан. Хабардың биттері $m_0, m_1, \ldots, m_{(b-1)}$ тәртіпте жазылатын болсын [3]. Хабардың үйірткісін (орамын) есептеу үшін мынадай бес қадам орындалады.

1-қадам. Толтыру биттерін толықтыру. Хабар, оның ұзындығы 512 модулі бойынша 448-бен салыстырмалы

болатындай етіліп толықтырылады. Толықтыру (кеңейту) былайша жүргізіледі: хабарға алдымен 1-ге тең 1 бит қосылады, ал қалған биттер – нөлмен толтырылады. Сонымен, қосымша биттердің саны 1÷512 аралығында болады.

2-қадам. Ұзындығын толықтыру. b хабары (толықтыру биттері қосылғанға дейінгі) ұзындығының 64 биттік көрсетімі, алдыңғы қадамның нәтижесіне қосылып жазылады. Егер хабар ұзындығы 2^{64}-тен артық болса, онда көрсетімнің тек кіші 64 биті (екі 32 разрядтық сөз ретінде) пайдаланылады. Осы операциядан кейін хабардың ұзындығы 512 битке, сондай-ақ 16 (32 разрядтық) сөзге дәл еселі болып шығады. Осылайша, алынған хабардың сөздері M[0…N-1] деп белгіленеді.

3-қадам. Үйірткінің арашығын инициализациялау. Төрт (A, B, C, D) сөзден тұратын арашық (buffer), хабардың үйірткісін есептеу үшін қолданылады. Бұл регистрлер мынадай 16-лық мәндермен толтырылады: A=′01234567′, B=′89abcdef′, C=′fedcba98′, D=′76543210′.

4-қадам. Хабарды 16 сөздік блоктар түрінде өңдеу. Әрқайсысының аргументі мен нәтижесі 32 биттік сөздер болып табылатын 4 көмекші функцияны анықтаймыз.

$$F(X,Y,Z) = XY \vee \neg(X)Z \qquad G(X,Y,Z) = XZ \vee Y\neg(Z)$$

$$H(X,Y,Z) = X \oplus Y \oplus Z \qquad I(X,Y,Z) = Y \oplus (X \vee \neg(Z))$$

Бұл қадамда синус функциясының негізінде құрылған 64 сөзден тұратын T[1…64] кесте пайдаланылады. T[*i*] – кестенің *i*-ші элементі. Бұл элемент 4294967296∗abs(sin(*i*)) санының бүтін бөлігіне тең (бұл жерде *i*-дің өлшемі радиан).

5-қадам. Шығу. Хабардың үйірткісі A, B, C, D регистрлерінде сақталады, яғни, A-ның кіші байтынан басталып, D-ның үлкен байтына дейін орналасады.

SHA алгоритмі

SHA (Secure Hash Algorithm – қауіпсіз хэштеу алгоритмі) хэш-функциясының SHA-1, SHA-256, SHA-384 және SHA-512 сияқты түрлері бар. Олардың шықпасында бастапқы тізбектің,

тиесілі 160 биттік, 256 биттік, 384 биттік және 512 биттік ұзындығы бар хэш-мән алынады [31, 37].

Ұзындығы 2^{64} биттен кем М хабарды енгізгенде SHA-1 алгоритмі 160 биттік шықпа хабарын жасап шығарады. SHA алгоритмі былай жұмыс істейді. Ең алдымен бастапқы М хабары 512 битке еселенетіндей етіп толықтырылады. Хабарды қосымша толтыру былай жүзеге асырылады: алдымен 1 қосылады, одан соң 512 еселенген биттерден 64 биті кем болатын хабар құрастыру үшін қанша 0 керек болса, сонша нөл қосылады. Ең соңында бастапқы хабардың ұзындығының 64-биттік көрсетімі қосылады.

A = 6 7 4 5 2 3 0 1
B = E F C D A B 8 9
C = 9 8 B A D C F E
D = 1 0 3 2 5 4 7 6
E = C 3 D 2 E 1 F 0

Бес 32-биттік (оналтылық санау жүйесінде берілген) айнымалылар алынады. Осыдан соң алгоритмнің бас циклы басталады. Мұнда хабардағы 512 биттік блоктар бірінен соң бірі кезекпен өңделеді. Алғашқы бес А, В, С, Д, Е айнымалылар басқа a, b, c, d, e айнымалыларға көшіріледі: a=A, b=B, c=C, d=D, e=E.

Бас цикл әрқайсысында 20 операциясы бар төрт циклдан тұрады. Әрбір операция бес a, b, c, d, e айнымалылардың үшеуінің сызықтық емес функциясын жүзеге асырады, одан соң ығыстыру мен қосуды орындайды.

Бас цикл аяқталған соң a, b, c, d, e мәндері А, В, С, Д, Е мәндерімен қосылады, содан соң алгоритм деректердің келесі 512 биттік блогын өңдеуге кіріседі. Ең соңында, нәтиже А, В, С, Д, Е мәндерінің конкатенциясы түрінде қалыптастырылады.

ГОСТ Р34.11-94 хэш-функциясы

ГОСТ Р34.11-94 деп аталатын стандарт кез келген тізбектелген екілік символдарға арналған хэш-функцияны есептейтін алгоритм мен процедураны анықтайды. Бұл стандарт ГОСТ 28147-89 блоктық шифрлау алгоритмі негізінде жасалған. Негізінде, 64 биттік блогы мен 256 биттік кілті бар басқа блоктық шифрлау алгоритмін де пайдалануға болады. Бұл хэш-функция 256-биттік хэш-мәнді қалыптастырады.

H_i f(M_i, H_{i-1}) қысу функциясы (M_i және H_{i-1} операндтары 256 биттік шама) былайша анықталады:

1. Төрт K_i =1, ..., 4 шифрлау кілті генерацияланады. Ол M_i, H_{i-1} және C_j тұрақтыларын сызықтық араластыру арқылы жүзеге асырылады.

2. Әрбір K_i кілтін H_{i-1} сөзінің 64-биттік h_i ішсөздерін қарапайым ауыстыру режімінде шифрлау үшін пайдаланады: $S_i=E_{K_i}(h_i)$. Ұзындығы 256 биттік соңғы нәтиже S_4, S_3, S_2, S_1 тізбесі S уақытша айнымалыда сақталады.

3. S_i, M_i және H_{i-1} үшеуінің сызықтық араластыру функциясы болғанмен, H_i мәні күрделі болып келеді.

M хабарының ең соңғы хэш-мәнді есептеген кезде, оның өзара байланысқан айнымалылар мәні есепке алынады: H_n - соңғы хабар блогының хэш-мәні; Z - барлық хабар блоктарын екі модулі бойынша қосқан кезде алынатын бақылау қосындысының мәні; L - хабар ұзындығы.

Осы үш айнымалы мен хабардың толықтырылған соңғы блогы M' ең ақырғы хэш-мәнге былайша біріктіріледі: $H = f(Z \oplus M', f(L, f(M', H_n)))$.

2.3.2. Цифрлық қолтаңба алгоритмдері

Цифрлық қолтаңба хабардың өзіне және қол қоюшыға ғана мәлім құпия кілтке тәеулді болатын сан. Цифрлық қолтаңба мынадай негізгі үш мәселені шешуге мүмкіндік береді: хабар жіберуші жақты аутентификациялау, хабардың тұтастығын қадағалау және нақты хабарды қолтаңбалау фактісінен бас тарту мүмкіндігін жоққа шығару [1, 3, 34, 37].

Қазіргі уақытта цифрлық қолтаңба сұлбаларын жасауда бірнеше келісім бар. Оларды мынадай үш топқа бөлуге болады [1]: ашық кілтті шифрлау жүйелері негізіндегі сұлбалар (мәселен, RSA цифрлық қолтаңбасы); арнайы әзірленген есептеу мен қолтаңба тексеру алгоритмдері бар сұлбалар (мәселен, Фиат-Шамир және Эль-Гамаль цифрлық қолтаңбалары); симметриялық шифрлау жүйелері негізіндегі сұлбалар (мәселен, Диффи-Лампорт цифрлық қолтаңбасы).

RSA цифрлық қолтаңбасы

RSA алгоритмінің жұмысы, үлкен сандарды жіктеу мен дискретті логарифмдерді есептеу қиындықтарына негізделген. Ол деректерді шифрлау режімінде де, цифрлық қолтаңба режімінде де жұмыс істей алады [1, 3, 34, 37].

Құжатты RSA алгоритмінің көмегімен қолтаңбалау қажет болғанда алдымен оның параметрлері, осы алгоритмнің шифрлау режіміндегі сияқты таңдап алынады.

Мысал. "Келісілген" деген хабардың цифрлық қолтаңбасын қалыптастыру керек. Содан кейін алынған қолтаңбаны RSA алгоритмінің көмегімен тексеру қажет. Қабылданған шартты белгілер: N – модуль, e – ашық кілт, d – жабық кілт, p және q – кездейсоқ сандар. Хэш-функциясы ретінде хабардың барлық әріптерінің N модулі бойынша саналған қосындысы алынған. Қолданылатын параметрлердің мәні:

$p=3,\ q=17,\ N=p\cdot q=3\cdot 17=51;\ \varphi=(p-1)(q-1)=32;\ e=3;\ e\cdot d=1 \bmod \varphi,\ d=11.$

M хабарын сәйкестік кестесінің (2.1-сурет) көмегімен сандар ретінде көрсетелік. Сонда ("Келісілген") хабарымыз мынадай түрде жазылады: M=13,7,15,37,23,37,15,4,7,17.

M хабарының қолтаңбасын қалыптастыру:

1. Хэш-функциясының мәнін табамыз:

$$m=h(M)=\sum_{i=1}^{10} M_i \pmod{51} = 175 \pmod{51} = 22 \cdot$$

2. Құпия кілттің d=11 мәнін және m=22 хэш-мәнін қолдана отырып, цифрлық қолтаңбаны есептейміз:

$$S = m^d \pmod{N} = 22^{11} \pmod{51} = 28.$$

3. Сонда S қолтаңбасы бар M хабарының түрі мынадай болып шығады:

(M, S)=(13, 7, 15, 37, 23, 37, 15, 4, 7, 17, 28).

Қолтаңбаны тексеру алгоритмі:

1. Хэш-функциясының мәнін есептейміз:

$$m' = h(M') = \sum_{i=1}^{10} M_i \pmod{N} = 175 (\bmod 51) = 22 \cdot$$

2. e=3 ашық кілтті пайдаланып, хэш-функцияның мәнін табамыз:

$$m'' = S^e \pmod{51} = 28^3 \pmod{51} = 21952 \bmod 51 = 22.$$

3. Тексеру үшін m′ және m′′ мәндерін салыстырамыз. Бұл жағдайда m′=m′′=22 болғандықтан, қолтаңба қабылданады.

Эль-Гамаль цифрлық қолтаңбасы

Эль-Гамаль цифрлық қолтаңбасы шекті өрісте логарифм мәнін есептеудің күрделілігіне негізделген [1, 34].

p – жәй сан, ал α – Z_p өрісінің примитивтік элементі болсын.

$1 \le a \le p-2$ аралығында *a* кездейсоқ санын таңдап, $\beta=\alpha^a \bmod p$ мәнін есептейміз. Сонда, *a* саны құпия кілт, ал (p, α, β) жиыны – ашық кілт болып саналады. M хабары үшін қолтаңба келесі алгоритм көмегімен есептеледі:

1. r кездейсоқ бүтін санды таңдау, $1 \le r \le p-2$.
2. $\gamma=\alpha^r \bmod p$ мәнін есептеу.
3. x=M үшін $\delta = (x - \alpha\gamma)r^{-1} \bmod (p-1)$ мәнін есептеу.
4. M хабарының қолтаңбасы ретінде (γ, δ) жұбы алынады.

Қолтаңбаны тексеру алгоритмі $\beta^\gamma \gamma^\delta \equiv \alpha^x \pmod{p}$ салыстыруын тексеруді көздейді. Егер салыстыру дұрыс болса, онда қолтаңба қабылданады, ал егер қате болса – қабылданбайды.

Цифрлық қолтаңбаның мұндай сұлбасының негізгі артықшылығы – бір құпия кілттің көмегімен, көптеген хабардың цифрлық қолтаңбасын дайындау мүмкіндігі. Хабарды кері шифрлау үшін көрсеткіштік теңдеулердің шешімін (оның ішінде Z_p өрісінде логарифм мәнін) табумен байланысты күрделі математикалық есепті шешу қажет болады.

Екі нәрсені ескерту керек. Біріншісі – r санын таңдауға қатысты. Бұл сан қолтаңба есептелгеннен кейін r саны бірден жойылуы керек. Шын мәнінде де, r саны мен қолтаңба мәні белгілі болған жағдайда, a құпия кілтті есептеу қиындық тудырмайды:

$a= (x - r\delta)\gamma^{-1} \bmod (p - 1)$.

Сондай-ақ, r саны шын мәнінде кездейсоқ болуы және құпия кілттің бір мәнін қолдану арқылы алынған түрлі қолтаңбалар үшін қайталанбауы керек. Олай болмаған жағдайда, *a* құпия кілтін есептеп алуға тағы да мүмкіндік туады.

Екінші ескерту – алгоритмнің үшінші қадамында қолтаңбаны есептеу кезінде x ретінде M хабардың өзін емес, x=h(M) үйірткісін қолданған дұрыс екендігіне байланысты. Бұл қолтаңбаның белгілі мәнін қолдана отырып, хабарды іріктеп алу мүмкіндігінен қолтаңбалау сұлбасын қорғайды. Осындай іріктеудің бірнеше тәсілдері бар. Мысалы, $0<i<p-1$, $0<j<p-1$, $(j, p-1)=1$ шарттарын қанағаттандыратын i, j сандарын таңдап алсақ және $\gamma = \alpha^i \beta^j \bmod p$, $\delta = -\gamma j^{-1} \bmod (p-1)$, $x = -\gamma i j^{-1} \bmod (p-1)$ деп алатын болсақ, онда M=x хабары үшін (γ, δ) жұбы дұрыс цифрлық қолтаңба болатынына оңай көз жеткізуге болады.

Эль-Гамаль цифрлық қолтаңбасының сұлбасы, басқа көптеген қолтаңба сұлбаларын құру үлгісі ретінде қолданыс тапты. Олардың бәрінің негізінде салыстырудың мынадай түрі жатыр:

$$\alpha^A \beta^B \equiv \gamma^C (\bmod p).$$

Мұндағы (A, B, C) үштігі ±x, ±δ және ±γ сандарының біреуіне сәйкес келеді. Мәселен, Эль-Гамаль қолтаңбасының бастапқы сұлбасында A=x, B=-γ және C=δ, ал DSS стандартында A=x, B=γ және C=δ, сондай-ақ ГОСТ Р34.10-94 стандартында A=-x, B=δ және C=γ мәндері қолданылады.

Эль-Гамал сұлбасында (γ, δ) сандар жұбын (γ mod q, δ mod q), сандар жұбына ауыстыру арқасында қолтаңба ұзындығын қысқарту мүмкіндігі бар. Мұндағы q саны – (p-1) санының қарапайым бөлгіші. Бұл кезде тексеру (салыстыру) теңдігінің түрі мынадай болады:

$$(\alpha^A \beta^B \bmod p) \bmod q \equiv \gamma^C (\bmod q).$$

Яғни, p модулі бойынша тексеру теңдігі, q модулі бойынша тексеру теңдігіне өзгертіледі. DSS цифрлық қолтаңба стандартында дәл осылай жасалған.

DSS стандарты

Цифрлық қолтаңба DSA (Digital Signature Algorithm) алгоритмін кейде цифрлық қолтаңба DSS (Digital Signature Standard) стандарты деп атайды. Бұл стандартқа сәйкес электрондық цифрлық қолтаңба (ЭЦҚ) мына үш алгоритмдердің біреуі бойынша әзірленеді: DSA, ANSI X9.31 (RSA DSA) немесе ANSI X9.63 (EC-DSA). Соңғы кезде эллипстік қисықтар тобын қолданатын алгоритмдер кеңінен таралған. Олар EC-DSA (эллипстік қисықтар негізіндегі цифрлық қолтаңба алгоритмдері) деп белгіленеді. DSA алгоритмінің бұл нұсқасы басқаларына қарағанда тезірек жұмыс істейді және оның көмегімен алынған қолтаңба қысқарақ болады [1, 3, 34, 37].

DSA алгоритмі. Оның параметрлері Эль-Гамаль жүйесінің параметрлеріне ұқсас болып келеді.

1) ***Параметрлер таңдау***. p, q және g сандарын таңдап алу. Бұл жерде: p – қарапайым сан ($2^{l-1} < p < 2^l$), l – 64-ке еселі және $512 \le l \le 2048$; q – 160 биттік p-1 санының қарапайым бөлгіші ($2^{159} < q < 2^{160}$); g – q дәрежелі топтың элементі, $g=h^{(p-1)/q}$, $1<h<p-1$ және $h^{(p-1)/q} >1$. Аталған үш сандар ашық деректер болып табылады.

Құпия x кілті ($0<x<q$) таңдалады және қолтаңбаны тексеруге арналған $y=g^x(\mathrm{mod}\ p)$ ашық кілті есептеледі.

2) ***Цифрлық қолтаңбаны әзірлеу.*** M хабарынан h(m) хэш-функцияның мәні есептеледі. Ол үшін SHA-1 алгоритмі пайдаланылады. Хэш-функцияның мәнінің ұзындығы 160 битке тең. Одан кейін жіберуші кездейсоқ немесе жалғанкездейсоқ k мәнін ($0<k<q$) таңдап алады, $k^{-1}(\mathrm{mod}\ q)$ мәнін есептейді де, R мен S мәндерін санап шығарады:

$$R = g^k (\mathrm{mod} p)(\mathrm{mod} q); \qquad S = k^{-1}(h(m) + xr)(\mathrm{mod} q).$$

Осы (R, S) жұбының мәндері M хабарының электрондық қолтаңбасы болады. Бұдан кейін k мәні жойылады.

3) ***Цифрлық қолтаңбаны иландыру.*** Қабылданған M хабарын тексеру теңдеуінің түрі мынадай:

$$R \equiv g^{h(m_1)\cdot s^{-1}} \cdot y^{r\cdot s^{-1}} (\mathrm{mod} p)(\mathrm{mod} q)$$

Мысал [37]. q=13, p=4q+1=53, g=16. x=3 және y=g^3 (mod p) = 15 болсын. Егер хэш-мәні h=5 болатын хабарды қолтаңбалау керек болса, онда алдымен эферлік кілт k=2 таңдап алынып, R және S есептеледі

R=(g^k (mod p)) (mod q) = 5, S=(h+xR)/k (mod q) = 10.

Жіберушінің қолтаңбасын тексеру үшін қабылдаушы A, B, V есептейді:

A=h/S (mod p) = 7, B=R/S (mod p) = 7, V=($g^A y^B$ (mod p)) (mod q) =5.

Сөйтіп, R=V болғандықтан, қолтаңба шынайы деп саналады.

EC-DSA. Эллипстік қисықтар негізіндегі EC-DSA цифрлық қолтаңба алгоритмдерінің жұмыс істеу (қарапайым) мысалын келтірелік [37]. Эллипстік қисықтар негізі ретінде E: $Y^2=X^3+X+3$ алынған. Оның элементтерінің саны Q=197, жасаушының координатасы P – (1, 76). Құпия кілті деп x=29 алынса, онда ашық кілті мынаған тең болады:

Y=[x]P=[29](1,76)=(113, 191).

Егер құпия кілттің иесі хэш-мәні H(M)=68 болатын хабарды қолтаңбалағысы келеді десек, онда ол эфемерлік кілтті k=153 деп алып, R мәнін есептейді.

R = x-коор.([k]P) = x-коорд.([153](1, 76])) = x-коорд. ((185, 35)) = 185.

Одан кейін S мәні есептеледі.

S = (h(M))+x·R)/k (mod Q) = (68+29·185)/153 (mod 197) = 78.

Сонымен, хабармен бірге жіберілетін қолтаңба болып (R, S)=(185, 78) жұбы алынады.

Қолтаңбаны тексеру үшін A, B және Z мәндері есептеледі.

A = h(M)/S (mod Q) = 68/78 (mod 197) = 112,

B = R/S (mod Q) = 185/78 (mod 197) = 15.

Z = [A]P+B[Y] = [112](1, 76) + [15](113,191) =

= (111, 60) + (122, 140) = (185, 35).

Қолтаңба шынайы деп саналады, себебі R = 185 = x-коорд.(Z).

ГОСТ Р34.10-94 стандарты

Американдық стандартқа қарағанда, бұл ресейлік стандартта хэш-функция мәнінің ұзындығы үлкейтілген (қақтығыс ықтималдығын азайту үшін) және генератор-элементтің дәрежесі өсірілген (құпия кілтті ашу кезінде дискреттік логарифм есебін шешуді қиындату үшін). Бұл электрондық (цифрлық) қолтаңба жүйесінің құрамына екі процедура кіреді: қолтаңба дайындау және оны тексеру. Қолтаңба екі бүтін саннан тұрады. Олар төменде келтірілген ережеге сәйкес есептеледі [3, 34].

Жүйенің параметрлері болып саналатын p, q және a сандары – құпия болып саналмайды. Хабарға қол қою процедурасы кезінде генерацияланатын бүтін k саны құпия болып, әрі қолтаңба дайындалып біткеннен кейін міндетті түрде жойылуы қажет.

Мынадай шартты белгілер қабылданған: p – қарапайым сан, $2^{509}<p<2^{512}$ немесе $2^{1020}<p<2^{1024}$; q – қарапайым сан, $2^{254}<q<2^{256}$ және (p-1) үшін бөлуші болып табылады; a – бүтін сан, $1<a<(p-1)$ және $a^q(\mathrm{mod}\ p)=1$; k – бүтін сан, $0<k<q$; x – қолтаңбаны дайындауға керек құпия кілт, $0<x<q$; y – қолтаңбаны тексеруге керек ашық кілт, $y=a^x(\mathrm{mod}\ p)$.

Қолтаңба дайындау процедурасы мына қадамдардан тұрады:

1) M хабарының хэш-функциясының h=h(m) мәнін есептеу. Оның мәні нөлге тең болған жағдайда, қолтаңба дайындалмайды;

2) k бүтін санын ($0<k<q$) қалыптастыру;

3) $r^1=a^k(\mathrm{mod}\ p)$ және $r=r^1(\mathrm{mod}\ q)$ мәндерін есептеу. Егер r=0 болса, онда 2-қадамға қайта оралып, k-нің басқа мәнін алу керек;

4) x құпия кілтін пайдалана отырып, $s=(kh+xr)(\mathrm{mod}\ q)$ мәнін есептеу. Егер s=0 болса, онда 2-қадамға қайтып оралу, ал басқа жағдайда – алгоритмнің жұмысын аяқтау.

5) Қолтаңбаланған (M, r, s) хабар алынады.

Тек қабылдаушыда хабар жіберушінің ашық кілті болған жағдайда ғана цифрлық қолтаңбаны тексеруге болады. Тексеру теңдеуінің түрі мынадай:

$$r \equiv (a^{s \cdot h(m_1)^{-1}} \cdot y^{-r \cdot h(m_1)^{-1}} (\mathrm{mod} p))(\mathrm{mod} q).$$

Қолтаңбаны тексеру былайша жүзеге асырылады.

1. 0<s<q және 0<r<q шарттарын тексеру. Егер осы шарттардың біреуі немесе екеуі де орындалмаса, онда қолтаңба жарамсыз деп саналады.
2. Қабылданған M_1 хабарының $h(m_1)$ хэш-функциясын есептеу.
3. $\nu = (h(m_1))^{q-2} (\mathrm{mod}\ q)$ мәнін есептеу.
4. $z_1 = \nu s (\mathrm{mod}\ q)$ және $z_2 = \nu (q-r) (\mathrm{mod}\ q)$ мәндерін есептеу.
5. $u = (a^{z1} y^{z2} (\mathrm{mod}\ p)) (\mathrm{mod}\ q)$ мәнін есептеу.
6. r=u шартын тексеру.

Егер r және u мәндері бір-біріне тең болса, онда қабылданған хабардағы қолтаңба нағыз жіберушінікі болғаны және тасымалдану кезінде хабардың тұтастығы бұзылмаған деп саналады. Басқа жағдайда, қолтаңба жарамсыз болып саналады.

ГОСТ Р34.10-2001 стандарты

Бұл стандарт ГОСТ Р34.10-94 стандартының орнына қолдану үшін жасалған. Бұл стандартта a және b коэффициенттері немесе J(E) шамасы арқылы берілген эллипстік қисық E қолданылады [3, 34].

$$\begin{cases} a = 3k (\mathrm{mod}\ p), \\ b = 2k (\mathrm{mod}\ p), \end{cases} \qquad k = \frac{J(E)}{1728 - J(E)} (\mathrm{mod}\ p), \qquad J(E) \neq 0,1728$$

$$J(E) = 1728 \frac{4a^3}{4a^3 + 27b^2} (\mathrm{mod}\ p).$$

Қабылданған шартты белгілер: p – эллипстік қисықтың модулі, қарапайым сан, $p > 2^{255}$; m – бүтін сан, эллипстік қисықтың нүктелер тобының дәрежесі; q – қарапайым сан,

эллипстік қисықтың нүктелер тобының иштобы дәрежесі. Мына шартты қанағаттандыруы керек:

$$\begin{cases} m = ng, n \in Z, n \geq 1 \\ 2^{254} < g < 2^{256}; \end{cases}$$

Мұндағы: Q – k нүктенің еселігі, k∈Z, егер P нүктесіне Q=kP орындалса; (x_p,y_p) - qP=0 теңдігі орындалатын P нүктенің координатасы.

Әрбір пайдаланушының жеке екі кілті болады. Құпия кілт d (0<d<q) және ашық кілт – qP=Q теңдігі орындалатын (x_q,y_q) координатасы бар Q.

ЭЦҚ параметрлері келесі шарттарды қанағаттандыруы керек:

– $p^t \neq 1 (\text{mod } q)$ барлық бүтін t(1, 2, ..., B (B≥31);

– m(p және J(E) ≠ 0 немесе 1728.

$\overline{h}$ =(α_{255}, ..., α_0) екілік векторына $\alpha = \sum_{i=0}^{255} \alpha_i 2^i$ саны сәйкес қойылады.

M хабары үшін қолтаңба мына алгоритм бойынша дайындалады.

1) M хабарының хэш-функциясын есептеу: $\overline{h}$ = h(M).

2) Жалғанкездейсоқ k санын генерациялау (0<k<q).

3) Эллипстік қисықтың C=kP нүктесін анықтау.

4) $r=x_c \pmod q$ мәнін есептеу. Бұл жерде x_c - C нүктесінің x-координатасы. Егер r=0 болса, онда қайтадан 2-қадамға оралу.

5) $s=(rd+k\overline{h}) \pmod q$ мәнін есептеу. Егер s=0 болса, онда қайтадан 2-қадамнан бастау.

6) r және s сандар жұбымен M хабары қолтаңбаланады (M, r, s).

M хабарының қолтаңбасын тексеру үшін Q кілтін қолдана отырып, келесі қадамдарды орындау керек.

1. Қабылданып алынған қолтаңбадағы r мен s сандары үшін егер 0<r<q және 0<s<q теңсіздіктері орындалса, онда келесі қадамды орындау. Басқа жағдайда – қолтаңба жарамсыз.

2. Қабылданған M хабардың хэш-функциясын есептеу: h= h(M).

3. $z_1=sh^{-1} \pmod q$ және $z_2=-r\ h^{-1} \pmod q$ мәндерін есептеу.

4. Эллипстік қисықтың $C=z_1P+z_2Q$ нүктесін есептеу және $R=x_c(\mathrm{mod}\ q)$ деп алу. Бұл жерде x_c – C нүктесінің x-координатасы.

5. Егер R=r болса, онда қолтаңба жарамды, ал басқа жағдайда, жарамсыз болып табылады.

2.4. Кванттық криптография

Кванттық криптография – оптикалық арналармен құпия ақпарат тасымалдау тәсілі. Бұл тәсілде деректер жеке фотондар арқылы кодталады [7, 8, 15, 25, 26, 30, 35, 36]. Бұл жерде Гейзенбергтің қағидаты тікелей қолданылады. Осы қағидатқа сәйкес, кванттық жүйеде өлшеу жүргізуге бағытталған әрекет, оның күй-жайын бұрмалауға әкеледі және осындай өлшеу нәтижесінде алынған ақпарат, өлшеп бастауға дейінгі күй-жайға толық сай келмейді. Бөгеттердің кванттық байланыс арнасындағы ақпаратқа тиісу, міндетті түрде оған бөгеуілдер енгізуге әкеледі. Сондықтан қаскөй, тасымалданып жатқан ақпаратқа маңызды бұрмалау енгізбей, арнаны тыңдауға мүмкіндігі жоқ. Ресми абоненттер осы бұрмалаулар деңгейі бойынша бұзу фактісін әрқашан анықтай алады.

Осындай технология – дәстүрлі шифрлау алгоритм-деріндегі құпия кілттерді тасымалдауға өте ыңғайлы. Таратушы жақ кілтті генерациялап, оны кванттық арна арқылы қабылдаушы жаққа жібереді. Ашық арна ретінде кәдімгі телефон және радиобайланыс желілері немесе жергілікті есептеу желілері, сондай-ақ стандартты жұмыс режімінде істейтін талшық-оптикалық байланыс желісі (link) қолданылады.

Қолданыстағы, бұрыннан бар бірде-бір шифрлау жүйесі, сенімділігі жағынан кванттықпен салыстыруға келмейді. Мұндағы ақпарат бірлі-жарым үйектелген (поляризацияланған) фотондар серияларын алмастыру арқылы генерацияланатын кілттермен шифрланады. Үйектелген немесе «байланысқан» фотондарды жолай ұстау, теория жүзінде де мүмкін емес. Себебі, кез келген жолай ұстау әрекеті – оның кванттық қалып-

күйінің бүлінуіне әкеліп соғады (бұл кванттық физиканың іргелі заңдылықтарының бірі).

Кванттық криптографияда кілт туралы ақпаратты кодтау үшін, жарықтың жалғыз ғана фотонын қолдану көзделген. Өз кезегінде, бұл фотон талшық-оптикалық кәбіл бойымен қабылдаушыға жіберіледі.

Кванттық криптография – бір түйіннен екінші түйінге бағытталатын фотондардың кванттық күйлерін бақылау арқылы, байланыс желісін қорғауға мүмкіндік береді. Ақпаратты жасырын жолай ұстауға бағытталған кез келген іс-әрекет, оның физикалық тасуыштарының (фотондардың) кванттық күйлерінің өзгеруіне әкеледі. Нәтижесінде ақпараттың бұрмаланған фактісі белгілі болып, теріс іс-әрекет әшкереленеді.

Дегенмен, кванттық криптография қауіпсіздікті 100% қамтамасыз етуге кепілдік бере алмайды. Себебі, қаскөйлер аппараттық қамтама құрылмасының ерекшеліктерін пайдалануы мүмкін. Бәрібір, шифрлаудың кванттық технологиясы, деректер тасымалдау кезінде қолданылатын (негізінде біржақты функция жатқан) криптография жүйелеріне қарағанда әлдеқайда қауіпсіз.

Кванттық криптографияның дамуын тежеп тұрған шешілмеген мәселелердің бірі – фотондық сигнал деградациясының күштілігі салдарынан, талшықты-оптика бойымен тасымалдау қашықтығының шектеулігі (120 км, бүгінгі таңда - 50 км). Осы мәселені шешудің бір жолы – сигналды ауа немесе жерсеріктері арқылы жіберу, ал бұл өз кезегінде кванттық криптографияны континентаралықтық етуге мүмкіндік берер еді. Сонымен, кванттық криптографиялық жүйелерді қолдану – қауіпсіздікті қазіргі қолданыстағы криптографиялық жүйелермен салыстырғанда, елеулі жоғарылатуға мүмкіндік береді.

Солтүстік Батыс университеті (Иллинойс штаты, АҚШ) ғалымдары мен Массачусетс штатындағы BBN Technologies компаниясының бірігіп әзірлеген жұмыстары нәтижесінде, толық кванттық шифрлауды қамтитын алғашқы толыққанды деректер тасымалдау желісі құрылды**.** Жаңа желіде фотондар шоғырын үйектеуге (поляризациялауға) негізделген AlphaEta

жүйесінде кванттық шифрлау тәсілі, сондай-ақ шифрлау кілтін жіберудің кванттық жүйесі қолданылған. Кванттық криптографияның арқасында желі, деректерді жолай ұстап қалу мүмкіндігіне беріктілігімен ерекшеленеді. AlphaEta жүйесінде шифрлаудың дәстүрлі алгоритмдерімен қатар, кванттық механика негізіндегі физикалық қағидаттары да қолданылады.

Деректер тасымалдау кезінде AlphaEta кванттық жүйесі, шифрлаудың үнемі жаңартылып отырылатын кванттық кілтін қолданады. Желінің жұмыс істеу қабілетін көрсету үшін SONET талшық-оптикалық желінің ұзындығы 9 км-ге тең екі түйін арасындағы үзіндісі пайдаланылды (бірі – Гарвард университетінде, екіншісі – BBN Technologies компаниясының орталық кеңсесінде орналасқан). Шифрланған деректерді тасымалдау жылдамдығы 155 Мбит/с-ті құрады, ұзындығы 1 Кбиттік кванттық кілт, әрбір 1 сек сайын жаңартылып тұрды.

Іс жүзінде мынадай кванттық криптографиялық хаттамалар жиі қолданылады: BB84 (Чарльз Беннет және Жиль Брассар), B92, E91 (Экерт), тығыз кодтайтын кванттық хаттама (Чарльз Беннет және Стивен Визнер), BB84 (4+2) – BB84 классикалық хаттаманың жаңартылмасы, алты күй-жай хаттамасы (Брюсс), Гольденберг-Вайдман хаттамасы, т.б. [8, 36].

Кванттық криптографиялық жүйелердің негізгі даму бағыттары:

– талшық-оптикалық жүйелер (кванттық криптографияның талшық-оптикалық желілерді қорғаумен айналысатын саласы);

– сымсыз кванттық криптография;

– кванттық компьютер.

Бүгінгі кезде кванттық криптография – талшық-оптикалық байланыс желілері (ТОБЖ) бойынша тасымалданатын ақпаратты қорғау саласында ең көп практикалық қолданыс тапқан. Себебі, ТОБЖ-нің оптикалық талшықтары фотондардың бұрмалаусыз тасымалдауын қамтамасыз етеді. Фотондар көзі ретінде, лазерлік диодтар қолданылады. Одан кейін жарық сигналының қуаты біраз (дәлірек айтсақ, бір импульске шаққандағы фотондардың орта саны бірден біршама азырақ болатындай деңгейге дейін) әлсіретіледі. Қабылдауыш жақта

фотондар санау режімінде жұмыс істейтін, көшкіндік фотодиодтар қолданылады. Қабылдауыш жеке бөлшегінде осындай фотодиод қолданылатын ТОБЖ арқылы ақпарат тасымалдау жүйелері – кванттық оптикалық байланыс арналары (КОБА) деп аталады.

Кванттық оптикалық байланыс арналарында ТОБЖ-мен салыстырғанда, ақпарат тасымалдау жылдамдығы онша жоғары емес (секундына килобиттен бірнеше мегабитке дейін). Сондықтан, көп жағдайда кванттық криптографиялық жүйелер, жоғары жылдамдықты деректер ағынын шифрлау құралдарында пайдаланатын кілттерді тарату үшін қолданылады.

Кванттық ақпараттану – хабар жіберу мен есептеулердің (кванттық байланыс арналары, кванттық криптография, кванттық компьютер сияқты) жаңа әдістерін жүзеге асыру үшін, кванттық жүйелерді пайдалануға негізделген ғылым мен технологияның жаңа, қарқынды дамып келе жатқан саласы.

Кванттық арналардың классикалықпен салыстырғандағы негізгі артықшылығы – қорғау деңгейі сапасының жоғарылығы болып табылады: кванттық арна абсолюттік қорғанышты қамтамасыз ете алады, себебі онда жүйеге кіруге бағытталған кез келген іс-әрекет бірден аңғарылып қалады (кванттық байланыс арнасын бұзуға болады, бірақ оны ашу мүмкін емес).

Екі күйі бар және 1 бит ақпаратты тасымалдай алатын кванттық жүйе – ***кубит (qubit)*** деп аталады [7].

Кванттық компьютрді құру мүмкіндігі жөнінде алғашқы идеяны Фейнман ұсынған болатын. Компьютердің кванттық бөлігі n кубиттерден тұрады. Олардың әрқайсысына резонанстық сыртқы айнымалы өріс импульстары іріктеулі әсер етуі мүмкін. Өріс генераторларын қосу және олардың берілген кубитке сәулеленуінің мекендетілуі, классикалық компьютерлердің басқаруымен жүзеге асырылады. Деректерді енгізу және алгортмнің орындалуы, бір және екі кубиттік вентильдерді қолдану арқылы іске асырылады. Алгортм аяқталған соң, есептеу нәтижесі кубиттердің ақырғы кванттық күйіне жазылады. Нәтижені «оқу» үшін, (бір немесе бірнеше) кубиттердің қалып-күйлеріне кванттық өлшеу жүргізу қажет.

XXI ғасырдағы компьютер әлеміндегі кванттық компьютерлердің орнын былайша анықтауға болады: олар бұрыннан бар компьютер әлемін қолданыстан ығыстырып шығармайды, қайта оны толықтырады. Кванттық компьютерлерді, тек олар есепті тездетуге мүмкіндік беретін жағдайларда ғана қолдану керек.

Қазір бүкіл әлемде, жұмыс істеу қағидаты, кванттық механикаға негізделген құрылғы – кванттық компьютер құру жобалары жасалуда. Мұндай компьютер – классикалық компьютерде орындау мүмкін емес операцияларды жүзеге асыра алады (дәлірек айтсақ, орындауға мүмкіндік бар, бірақ оларды орындауға кететін уақыт көлемі өте үлкен болуы мүмкін). Екіншіден, талшықоптика бойынша фотондарды тасымалдаудың барынша үлкен ара қашықтығы 150 км-ді құрайды. Ара қашықтықты алдағы уақытта одан әрі ұзарту үшін, ретранслятор қажет. Кванттық компьютер осындай ретранслятор міндетін атқара алады.

Джон фон Нейман (John von Neumann) ұсынған фон-неймандық сәулетте әдетте, есептеу машинасы төрт бөліктен тұрады: жадылар, енгізу/шығару жүйелері (I/O), арифметикалық-логикалық блок (ALU) және басқару жүйелері. Кванттық жағдайда фон-неймандық сәулет құрамдасының бір бөлігі – енгізу/шығару жүйесін жүзеге асыру жөнінде ғана әңгіме етуге болады. Қалған құрамдас бөліктерінің кванттық аналогтары жоқ. Сонымен, кванттық жағдайда «компьютер» сөзінің үйреншікті мазмұнынан, тек «ақпаратты оқу» мүмкіндігі ғана қалады.

Қазіргі таңда әлемде кванттық компьютер құрастыруға бағытталған бірнеше технология пайда болды. Оптикалық технологиямен жұмыс істеу ыңғайлы, бірақ бір мезгілде кубиттер жұбымен (фотондар арасындағы әлсіз байланыстың салдарынан) жұмыс істеу айтарлықтай қиын. Айзек Чуан (Isaac Chuang) тобының ЯМР (ядерлік магнитті резонанс) технологиясы бойынша жасалған кванттық компьютер – бір молекуладан тұрды, ал кубиттер – оған кіретін атомдардың (жеті дана) ядросы болды. Джозефсон технологиясындағы басты

мәселе – жеке тұраулы кубит құру, ал олардың санын ұлғайту, оншалықты күрделі мәселе емес.

Дегенмен, бұл технологиялардың барлығы кванттық компьютердің мына сәулетіне бағытталған: жүйені абсолюттік кванттық бірлестіктің жанаспайтын «шиеленіскен» күй-жайында сақтау кезінде, жекелеген кубиттерге тізбекті және өте дәл әрекет ету қажет. Кубиттердің саны көбейген сайын, бұл шарттардың орындалуы қиындай түсуде. Сондықтан, дәлдікке және сыртқы орта әсерінен оқшаулауға қойылатын талаптары қатаң емес басқа сәулеттер, белсенді түрде іздестірілуде.

Соның бірі – адиабатикалық кванттық компьютер (АКК) идеясы. Есептің бастапқы деректері кубиттер жиынының (ең аз мүмкін болатын энергияны қамтитын) бастапқы күйімен кодталады. Сонан соң жүйені ақырын өзгерте бастайды. Және минимальды энергияның осы күйі («негізгі күй») де ақырын өзгере бастайды, бірақ үнемі минимальды мүмкін болатын энергиялы күйде қала береді. Ақыры процесс негізгі күйдің осындай пішінүйлесіміне бейімделіп алады да, жауапты кодтайды. Канадалық D-Wave компаниясы өздерінің Orion деп аталатын кванттық компьютері үшін дәл осы сәулетті таңдап, оны 2007 жылдың қаңтарында әлемге жариялады.

Orion бір мезгілде 64 мың операцияны орындай алады. Кез келген кванттық компьютердің негізгі бөлігі – кванттық регистр болып табылады. Кванттық регистр – белгілі бір мөлшерлі кубиттердің жиыны. Ақпаратты компьютерге енгізер алдында регистрдің барлық кубиттері негізгі базистік күйге келтірілуі керек (инициализациялау операциясы деп аталады). Одан кейін әрбір кубитке (сыртқы электромагниттік өріс импульстері немесе басқа жолмен) іріктеулі әсер етеді және регистр тұтастай базистік күйдің суперпозициясына көшеді.

Orion кванттық компьютерінде регистр 16 кубиттан тұрады. Ақпарат, кванттық логикалық операциялар тізбегін орындайтын кванттық процессормен өңделеді. Компьютер шығысындағы ақпаратты түрлендіру нәтижесі, күй-жайдың жаңа суперпозициясы болып есептеледі. Оны алдағы уақытта

пайдалануға жарамды болу үшін, әрі қарай түрлендіруге болады.

Бұл кванттық компьютер қазіргі қолданылып жүрген компьютерлермен бәсекеге түсе алмайды. Себебі ол бастапқы ақпарат көлемі үлкен және айнымалылар саны көп тапсырмаларды шешуге арналған. Мұндай есептерге криптография жүйелері мен деректерді қапуіпсіз тасымалдау, биология және медицинада, кванттық жүйелерді үлгілеу, түрлі процестерді оңтайландыру сияқты мәселелер жатады.

2004 жылы Кембриджде (АҚШ, Массачусетс штаты) әлемде бірінші рет кванттық криптографияны қамтитын **компьютерлік желі** пайдалануға берілді. Quantum Net (Qnet) жүйесі қазіргі кезде Дүниежүзілік өрмек және Интернет пайдаланушыларымен өзара әрекеттесуге қабілеті бар 6 серверден тұрады. Qnet жүйесін тәжрибелік эксплуатациялау кезінде деректер, кванттық желіге ұзындығы 10 км-лік қарапайым талшық-оптикалық кәбіл бойымен жіберілді. Литиидің ниобат кристалдарынан жасалған бағдарламаланатын оптикалық ауыстырып-қосқыштар Qnet жүйесінде бағдарғылауыш міндетін атқарады. Және олар фотондар қозғалысын тиісті талшықоптикаға бағыттап отырады [25].

3. СТЕГАНОГРАФИЯ

Бұрынғы заманнан бастап адамдар бір-біріне хат жіберу арқылы, өзара ақпарат алмастырумен шұғылданған. Хатты балауызбен, ал соңғы кезде сургучтік жеке мөрмен жапсырған. Олар осы күнге дейін поштамен хат алысу құпияларын сақтауға көмектеседі.

Ақпаратты әр түрлі тәсілдермен жасыруға болады, мысалы оны шифрлап тастау. Бірақ бұл тәсіл (криптография) кезінде қаскөй, сіздің қандай болмасын бір құпия хабар жіберіп жатқаныңызды біледі, бірақ оны оқи алмайды. Басқа (стеганография деп аталатын) тәсілде – тек қана хабар жасырылып қоймай, сонымен қатар тасымалдау фактісінің өзі де жасырылады.

Стеганография (steganography) – байланыс бар екендігінің өзін жасыратын байланыс ұйымдастыру әдісі. Стеганография әдістері ендірілген (қоса салынған) құпия жолдаулар бар екендігіне күдік тудырмайтындай етіп, оларды зиянсыз хабарлар құрамына ендіруге мүмкіндік береді.

"Стеганография" сөзі грек тілінен (steganographic) аударғанда – "құпия жазу" (steganos – құпия; graphy – жазу) дегенді білдіреді. Оған көрінбейтін сиялар, микрофотосуреттер, таңбалардың шартты түрде орналасуы, құпия арналар мен өзгермелі жиіліктер негізінде жасалған байланыс құралдары, т.б. байланыс құпия құралдарының біразы жатады.

Стеганография – ерте заманғы Геродот уақытынан бері белгілі [16]. Ежелгі Грекияда жолдаулар тақтайшалардың бетіне жазылып, содан кейін тақтайша балауызбен қапталған. Сонымен, жазу балауыздың астында қалып, көрінбейді.

Ал Қытайда хатты жібек мата жолағына жазған. Хабарды жасыру үшін хат мәтіні бар жолақтан шарик жасалып, ол балауызбен жабылған, содан соң шабарман оны жұтып қоятын болған.

Орта ғасырда алғашқы рет шифрлар мен стеганографиялық әдістер бірге қолданыла бастаған. XV ғасырда Тритемиус (1462–1516) криптография және стеганографиямен шұғылданған,

хабарды жасырын тасымалдаудың әр түрлі әдістерін сипаттаған. 1499 жылы бұл жазбалар "Steganographia" деп аталатын кітап болып шыққан.

Симпатикалық (түссіз), жасырын хабарлар, дәл бағытталған сәулемен тасымалдауға және тек арнайы физикалық немесе химиялық әсерден кейін ғана көрінетін түссіз сиялармен жазуға мүмкіндік беретін, арнайы техникалық құралдардың көмегімен істелінуі мүмкін. Осылайша, жасырылған (бүркеме) хабарларды – ***құпия жазу*** (немесе жұмбақ, сырлы жазу) деп атайды.

Стеганография тарихында фотографиялық шағын нүктелер ерекше орын алған. Шағыннүкте (микронүкте) – мөлшері типографиялық нүктедей өте кішкентай фотография, бірақ үлкейткен кезде стандарттық баспа беттің айқын бейнесін береді. Осындай бір немесе бірнеше нүктелер, кәдімгі хатқа жабыстырылған. Оларды табу қиын, сондай-ақ олардың үлкен көлемді ақпарат тасымалдау қабілеттілігі бар. Мәтіннің кәдімгі нүктесіндей мөлшерлі осындай бір шағын нүктеде құжаттардың жүздеген беттері орналасуы мүмкін, әрі оны орта форматты кітап ішінен табу өте қиын болды. Қазір, электрондық микроскоптан басқа, оптикалық құралдар көмегімен оны оқуға болмайтындай етіп, дәл осылай ұсақ мәтін жазып қоюдың ешқандай техникалық қиындығы жоқ.

Көптеген байланыс арналарында бөгеуілдер ағыны, шифрланатын құпия ақпарат үлесінен біршама артық болады. Сондықтан, шифрланған құпия биттерді (шуылдың аздап артуы сияқтандырып) кәдімгі жәй хабарлар арасына жасырады. Шифрқұжаттарды жасыру өте қарапайым: себебі олардың байланыс арналарындағы әдеттегі шуыл немесе бөгеуілдерден ешқандай айырмашылығы жоқ. Егер әдеттегі құпия жазу жеңіл оқылатын болса, онда шуылға немесе шатасуларға ұқсастырып жасырылған шифрланған хабардың құпия жазуын табу мүлде мүмкін емес.

Техникалық революция және ақпараттық технологиялар заманында компьютерлік технологиялар, стеганографияны дамытуға және жетілдіруге мүмкіндік берді. Сөйтіп, ақпарат

қорғау саласында жаңа бағыт (*компьютерлік стеганография*) пайда болды. Компьютерлік стеганографияның мақсаттары: ақпаратты жасырын тасымалдаудың сенімді стеганографиялық әдістерін әзірлеу және стеганографиялық жүйелердің ашылуға беріктілігін бағалау.

Қазіргі уақытта компьютерлік стеганография – ақпараттық жүйелерде ақпарат тасымалдау және сақтау фактісін жасырудың негізгі құралы болып табылады. Оны криптографиямен бірге қолдану, пайдаланушыға деректер қорғауды жоғары деңгейде ұйымдастыруға мүмкіндік береді.

Ақпарат жасырудың негізгі екі әдісі бар [4, 5, 9, 12]. Олардың біреуі – *ең аз мәнді биттерді ауыстыру әдісі* немесе LSB (Least Significant Bits) әдісі. Бұл әдіс цифрланған бейнелерде немесе аудио- және бейнефайлдарда жиі кездесетін дискреттендіру қателіктерін қолдану негізінде жасалған. Осы қателік – санның мәндік разрядтарының ең кішісіне тең. Сондықтан, кіші биттерді өзгерту – көп жағдайда бейнеде маңызды өзгерістер тудырмайды және көзге де көрінбейді. Бұл қазір ең көп таралған, бірақ беріктілігі аз әдіс.

Хабар ендірудің көп тараған, әйгілі басқа бір әдісі – деректер форматтарының ерекшеліктерін пайдалану әдісі. Мұнда деректер жоғалуы (шығыны) болатын қысу қолданылады (мысалы, JPEG). Бұл әдісте – қысылған бейне сапасын кең аралықта өзгертуге мүмкіндік бар (яғни, бұрмалау себебін анықтау қиындатылады). Сондықтан бұл әдіс (LSB әдісіне қарағанда), геометриялық түрлендіруге және тасымалдау арнасын іздеп-табуға көбірек төзімді.

Қазіргі заманғы стеганография

Компьютерлік желілер мен мультимедиа құралдар саласындағы жетістіктер, телекатынас арналарымен деректер тасымалдау қауіпсіздігін қамтамасыз етуге және оларды жарияламаған мақсаттарға қолдануға арналған жаңа әдістер әзірлеуге мүмкіндік тудырды. Цифрлау құрылғыларының табиғи дәлсіздіктері мен аналогтық бейне немесе аудио

сигналдың артықшылықтарын ескере отырып, бұл әдістер хабарларды компьютерлік файлдарда (контейнерлерде) жасыруға мүмкіндік береді.

Қазіргі компьютерлік стеганографияда файлдың екі негізгі түрі бар: *хабар* (немесе жасырылатын хабар) – жасырылуға тиісті кез келген файл, *контейнер* – ішінде хабар жасыру үшін қолданылатын файл. Ақпарат тасымалдау фактісін белгісіз ету үшін, хабар-файлды ерекше түрде контейнер-файлмен араластырады. Бұл контейнер, барынша нақты және күдік тудырмайтындай болып көрінуі керек. Контейнерге құпия ақпараттың орналастырылуы, оның негізгі қасиеттерін бұзбауға тиісті. Ал хабарды шығарып алу үшін араластыру жүргізілген алгоритмді білу қажет.

Цифрлық стеганография мына бағыттарды қамтиды [5, 12]:

1) жасырын тасымалдау мақсатымен ақпарат ендіру;

2) цифрлық су таңбаларын (ЦСТ) ендіру (watermarking);

3) идентификациялық нөмірлерді ендіру (fingerprinting);

4) бастамаларды ендіру (captioning).

Ендірілген *деректерді жасыру* – контейнерге күрделі талаптар қояды: контейнер мөлшері, ендірілетін деректер мөлшерінен бірнеше есе үлкен болуы керек.

Цифрлық су таңбалар (ЦСТ) немесе цифрлық ерекше белгілерін (ЦЕБ), көбінесе ақпаратты көшірмелеу мен рұқсатсыз пайдаланудан қорғау үшін, яғни цифрлық бейнелердің, фотографиялардың немесе өнердің басқа да цифрланған туындыларының авторлық немесе мүліктік құқықтарын қорғау үшін қолданады.

Өндірушінің *идентификациялық нөмірлерін ендіру* технологиясының, ЦСТ технологиясымен біраз ұқсастығы бар. Айырмашылығы – қорғалатын ақпараттың әрбір көшірмесінің өзіндік бірегей нөмірі болады («саусақтардың іздері»). Бұл идентификациялық нөмір, өндірушіге өз өнімінің тағдырын қадағалап отыруға мүмкіндік береді.

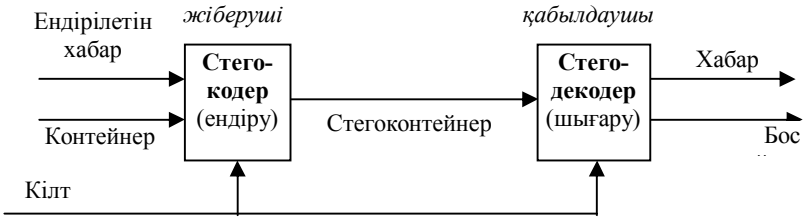
Көрінбейтін *бастамаларды ендірудің* мақсаты – әр түрлі көрсетімделген ақпаратты біртұтас ретінде сақтау. Мәселен, медициналық суреттерге қолтаңба қою, т.б. Бастамалар

цифрлық бейнелердің электрондық үлкен қоймаларында бейнелерді, дыбыстық және бейнефайлдарды таңбалауға пайдаланылады.

Терминдер және анықтамалар

1996 жылы Information Hiding: First Information Workshop конференциясында бірыңғай терминологияны қолдану ұсынылып, негізгі терминдер келісілген.

Стеганографиялық жүйе немесе ***стегожүйе*** (steganographic system) – ақпарат тасымалдаудың жасырын арнасын қалыптастыруға қолданылатын құралдар мен әдістердің жиынтығы. 3.1-суретте стегожүйенің негізгі элементтері көрсетілген [5, 9, 32, 33].



3.1-сурет. Стеганографиялық жүйенің үлгісі

Деректер ретінде кез келген ақпарат (мәтін, хабар, бейне, т.б.) қолданылады. Хабар ретінде мәтін немесе бейне, сондай-ақ аудиодеректер бола алатындықтан, бұдан әрі жасырылатын ақпаратты жалпы түрде – ***хабар*** деп атаймыз.

Кез келген стеганографиялық жүйе мына талаптарға сай болуы керек:

- Контейнердің қасиеттері, көзбен шолып бақылау кезінде өзгерістерді айқындауға болмайтындай етіліп, өзгертілуі керек. Бұл ендірілетін хабарды жасырудың сапасын анықтайтын талап: стегохабардың байланыс арнасымен бөгетсіз өтуін қамтамасыз ету үшін, ол ешнәрсемен өзіне шабуылшының назарын аудармауға тиісті.

- Стегохабар (стегоконтейнер) бұрмалауларға орнықты болуға тиісті, оның ішінде жаман ойлыға да.
- Ендірілетін хабардың тұтастығын сақтау үшін қатені жөндейтін кодты қолдану қажет.

Контейнер (контент, content, container, carrier) – құпия хабарды жасыруға арналған кез келген ақпарат. Контейнердің екі түрі болады. Контейнер немесе табиғи контейнер (немесе *бос контейнер*) – ішінде жасырын ақпараты жоқ контейнер (яғни, ендірілген хабары жоқ контейнер). *Стего* (стегоконтейнер немесе толтырылған контейнер, stego, stego-container) – ішінде жасырын ақпараты бар (яғни ішінде ендірілген хабары бар) контейнер.

Контейнерді таңдау, стеганографиялық жүйелердің сенімділігі мен жасырылған хабарды тасымалдау фактісін анықтау мүмкіндігіне зор ықпалын тигізеді.

Ұзындығы бойынша контейнерлер үздіксіз (ағындық) және шектелген (бекітілген) ұзындықты болып екі түрге бөлінеді. Ағындық контейнердің ерекшелігі – оның басын немесе соңын анықтау мүмкін емес. Сондай-ақ, кейінгі шуылдық биттердің қандай екендігін алдын ала біліп қоюға мүмкіндік жоқ. Қабылдайтын жақ үшін ең қиын мәселе – жасырылған хабардың басталатын жерін анықтау. Егер ағындық контейнерде уақытүйлесімдіру сигналдары болса немесе дестенің шекарасы белгіленсе, онда жасырын хабар бірден олардың біреуінен кейін басталады.

Бекітілген ұзындықты контейнерлерді қолданған кезде, жіберуші алдын ала файл мөлшерін біледі және жасырушы биттерді қолайлы жалғанкездейсоқтық тәртіпте таңдай алады. Бірақ, бекітілген ұзындықты контейнерлердің көлемі шектеулі болғандықтан, кейде ендірілетін хабар файл-контейнердің ішіне сыймай қалуы мүмкін. Осыған қарамастан, тәжірибе жүзінде көбінесе бекітілген ұзындықты контейнерлер қолданылады.

Контейнерлердің мынадай вариантты болуы мүмкін:

– контейнерді стеганографиялық жүйенің өзі генерациялайды;

– генерацияланған көп контейнерлер арасынан бір контейнер іріктеліп алынады. Бұл нұсқада алдымен көптеген балама контейнерлер генерацияланады, содан кейін олардың ішінен хабарды жасыруға жарайтын ең қолайлысы таңдап алынады. Оңтайлы контейнер таңдау кезіндегі қойылатын негізгі талап – контейнердің табиғилығы;

– контейнер сырттан келіп түседі. Бұл жағдайда контейнер таңдау мүмкіншілігі жоқ, хабар жасыру үшін бірінші кездескен кез келген контейнер алынады.

Ендірілген (жасырылған) хабар (стегохабар, embedded data, hidden data, message) – құпия түрде жіберілетін хабар (яғни, контейнерге ендірілетін хабар).

Стегокодер (stego-coder) – белгілі бір алгоритм бойынша контейнерге хабар ендіруді қамтамасыз ететін құрылғы.

Стегодетектор, стегодекодер (stego-detector, stego-decoder) – стего-контейнерден хабарды шығарып алатын немесе іздеу алгоритмінің көмегімен стего-контейнер ішінен хабарды іздеп-табатын құрылғы.

Стеганографиялық арна немесе ***стегоарна*** – стего тасымалдауға арналған арна.

Стеганографиялық кілт (стегоключ немесе кілт) – деректерді түрлендіру алгоритмінің кейбір параметрлерінің құпия, нақтылы күй-жағдайы. Бұл күй-жағдай берілген түрлендіру алгоритмінің мүмкін болатын барлық жиынтығының ішінен, тек бір түрлендіруді таңдап алуды қамтамасыз етуі керек. Сонымен, стегоключ – ақпарат жасыруға арналған құпия кілт. Қорғаныш деңгейлерінің санына байланысты (мысалы, алдын ала шифрланған хабарды ендіру) стегожүйеде бір немесе бірнеше стегокілт болуы мүмкін.

Криптографиядағы сияқты стегокілттің түріне байланысты стеганографиялық жүйелер екіге бөлінеді: құпия кілтті және ашық кілтті.

Құпия кілтті стеганографиялық жүйеде – бір кілт қолданылады. Бұл кілт құпия хабарлар алмастырып бастағанға дейін анықталуы керек немесе қорғалған арнамен жіберілген болуы тиіс.

Ашық кілтті стеганографиялық жүйеде хабарды ендіру мен шығарып алу үшін (біреуін, екіншісі арқылы есептеп табуға болмайтындай) әр түрлі кілттер қолданылады. Сондықтан, біреуін (ашық кілтті) қорғалмаған байланыс арнасымен жіберуге болады.

Стеганографиялық қорғау (қорғаныш) – деректерді стеганографиялық түрлендіру арқылы хабарды қорғау.

Стеганографиялық түрлендіру – хабарды контейнерде жасыру арқылы деректерді түрлендіру.

Стегообъект – хабарды объектіге енгізу нәтижесінде алынған объект (файл). Мәселен, мәтіндік файлға хабарды енгізу кезінде стегомәтін, ал бейнеге хабарды енгізу кезінде – стегобейне, т.с.с. алынады

Хабарды жасыру – белгілі бір талаптарға сай, хабар мен контейнерді стегоконтейнерге түрлендіру үрдісі.

Хабарды шығарып алу – стегоконтейнерді хабарға түрлендіру үрдісі.

Цифрлық ерекше белгі (ЦЕБ, digital watermark), цифрлық су таңбалары – арнайы мақсатпен бейнені қалқалайтын (бүркемелейтін) көзге көрінбейтін ақпарат. Мысалы, цифрлық ерекше белгі – авторлық құқық таңбасы рөлін, цифрлық қолтаңба, т.б. міндетін атқаруы мүмкін.

Стеганография әдістері

Қазіргі уақытта компьютерлік стеганография әдістері екі негізгі қағидатқа негізделген [4, 5, 9, 12, 33]:

– файлдардың немесе деректер алмастыру хаттамаларының форматтарының арнайы қасиеттерін пайдалану негізінде ақпаратты жасыру;

– мультимедиалық (дыбыстық және визуальдық) деректердің артықтылығын пайдалану негізінде ақпаратты жасыру.

Файлдар немесе хаттамалардың форматтарының арнайы қасиеттерін пайдалану негізінде ақпаратты жасыратын стеганография әдістерінің кемшіліктері – олардың жасырындылығы, дұшпанның тек жасыру алгоритмін білмеуіне

ғана негізделгендігі. Оларға қарағанда мультимедиалық деректердің артықшылығы негізінде жасалған стеганографиялық жүйелердің ерекшеліктері – жасыру қағидаты толық белгілі және алдын ала зерттеу үшін ашық, бірақ жүйенің беріктілігі тек құпия кілтке ғана тәуелді. Осыған орай дыбыстық және визуальдық (көзбен шолу) ақпараттың артықшылығын пайдалану, стеганографияның негізгі және ең көп таралған бағыты болып табылады.

1) Компьютерлік форматтардың арнайы қасиеттерін пайдалану негізінде жасалған әдістер

Бұл әдістер деректердің артықшылығын пайдаланбай, тек осы деректерді көрсетімдеудің компьютерлік форматтарының арнайы қасиеттерін ғана пайдалану негізінде жұмыс істейді. Бұл форматтардың арнайы қасиеттері – жасырылатын хабарды тікелей тыңдаудан, қараудан немесе оқудан қорғау керектігін ескере отырып таңдалады. Форматтардың арнайы қасиеттерінің түріне байланысты стеганографиялық әдістерді мынадай топтарға бөлуге болады:

1.1) Деректердің компьютерлік форматтарының арнайы қасиеттерін пайдаланатын әдістер. Бұған – деректердің компьютерлік форматтарының (кеңейту үшін қосалқыланған) өрістерін қолданатын әдістер жатады. Кеңейту өрістері көптеген мультимедиялық форматтарда болады, олар тек нөлдік ақпаратпен толтырылады да, бағдарлама оларды ескермейді. Сондықтан, осындай өрістерді хабар жасыру үшін пайдалануға болады.

1.2) Мәтіндік файлдарды арнайы бөліп-белгілеу (форматтау) әдістері. Бұл топқа мына әдістер кіреді:

– сөздердің, сөйлемдердің, азат жолдарының белгілі бір ығысуын пайдаланатын әдістер. Бұл әдістер сөздердің арасына қосымша бос орындар қосу арқылы, сөйлемдегі жолдар мен сөздердің орналасуын өзгерту негізінде жасалады;

– әріптердің белгілі бір позицияларын таңдайтын әдістер. Акроөлең – осы әдістің бір түрі. Мысалы, жасырын хабар өлеңнің әрбір жолының бастапқы әріптерінен құрастырылады;

– форматтардың экранда көрсетілмейтін өрістерінің арнайы қасиеттерін пайдаланатын әдістер. Бұл әдістерде нұсқамалар мен сілтемелер ұйымдастыруға арналған арнайы "көрінбейтін", жасырын өрістер қолданылады (мәселен, қара қарыппен қара фонда жазу);

– иілгіш дискілердің (тегеріштердің) пайдаланылмайтын орындарында (мәселен, нөлдік жолшықта) хабар жасыру әдістері;

– еліктететін функцияларды (mimic-function) қолданатын әдістер. Бұл әдіс – мәтіндерді генерациялауға негізделген және акроөлеңнің жалпыламасы болып табылады. Құпия хабардың орнына, осы хабарды жасыратын саналы басқа бір мәтін генерацияланады;

– бастаманының идентификациялайтын файлын жоятын әдістер. Бұл әдісте жасырылатын хабар шифрланып, алынған шифрқұжатта идентификациялайтын бастама алынып тасталады (тек шифрланған деректер ғана қалады). Қабылдаушы жақ хабар тасымалданатыны жайында алдын ала біледі және жетпейтін (жіберуші жақта алынып тасталған) бастама, оның өзінде бар болады. Бұл әдіс жеңіл жүзеге асырылатындықтан, ол көптеген құралдарда (мәселен, White Noise Storm, S-Tools бағдарламаларында) қолданылады;

– қателерді түзетуші кодтарды қолданатын әдістер, т.б.

2) Дыбыстық және визуальдық ақпараттың артықшылығын пайдаланатын әдістер

Дыбыстық және визуалдық (көзбен шолу) ақпараттың артықшылығын пайдалану, негізгі және ең болашағы зор бағыт болып саналады.

Компьютерлік форматтардың арнайы қасиеттерін қолдану әдістеріне қарағанда артықшылықты пайдаланатын әдістерде, стеганографиялық жүйе және оның жүзеге асырылуының егжей-тегжейі алдын ала жақсы белгілі, тек жасыру жолдарында ғана хабардың биттерін орналастыру ережесі белгісіз деп саналады. Бұл әдістерге – цифрлық фотографияның, цифрлық дыбыс пен цифрлық бейненің артықшылығын пайдалану әдісі жатады.

Цифрлық фотографиялар, цифрлық музыка, цифрлық бейне – кеңістіктегі және/немесе мезгілдегі өзгеше кездерде күшейтілгендікті (күшейе түскендікті) кодалайтын сандардың матрицаларымен көрсетімделеді. Цифрлық фотография – кадрдың белгілі бір нүктелеріндегі түстердің күшейтілгендігін көрсететін сандардың матрицасы. Цифрлық дыбыс – уақыт өту барысында дыбыстық сигналдың күшейтілгендігін көрсетімдейтін сандардың тізбегі.

Физикалық өрістерді, сигналдарды цифрлау, сондай-ақ мультимедиялық деректерді синтездеу кезінде екілік сандар (*цифрлық санаулар*) қолданылады. Осы цифрлық санаулардың кіші разрядтарында өте аз мөлшерде (дыбыс пен бейненің ағымдағы параметрлері жайындағы) пайдалы ақпарат сақталады. Оларды қосымша ақпаратпен толтыру, түйсіну (қабылдау) сапасына көп әсер етпейді. Сол себептен, бұл разрядтарды ақпарат жасыру үшін пайдалануға болады.

Цифрлық ерекше белгілерді ендіру үшін күрделірек әдістер қолданылады.

Компьютерлік стеганографияның қолданылуы

Компьютерлік стеганография әдістері мынадай қолданбалық мақсаттарда қолдануға арналған [33]:

1) ақпаратты рұқсаттыз қол жеткізуден қорғау;

2) авторлық құқықтарды қорғау;

3) шын сияқты терістеу (мойындамау) мүмкіншілігі;

4) түсініктемелерді мультимедиялық файлдарға енгізу.

♦Стеганографиялық әдістердің негізгі міндеті – *рұқсаттыз қол жеткізуден ақпаратты қорғау*. Бұл әдістер, желілік қорлардың мониторинг жүйелеріне қарсы әрекет жасауға бағытталған және жасырын ақпаратты байланыс арналарымен тасымалдау немесе цифрлық деректер тасығышында сақтау кезінде қорғауға арналған.

♦Стеганографияның басқа бір міндеті – *авторлық құқықты қарақшылықтан қорғау* болып келеді. Қазіргі заманғы компьютерлік жүйелер дыбыстық және бейнедеректерді оңай көшірмелеуге мүмкіндік береді. Бұл деректерге авторлық құқық

таралуы мүмкін. Сондықтан, цифрлық фотографияларға, цифрлық бейнелерге (авторлық құқық объектілеріне) арнайы тамға (label) қойылады. Мұндай тамғалар көзге көрінбейді, бірақ арнайы бағдарламалық қамтаманың көмегімен ажыратуға-тануға болады.

Зияткерлік меншікті қорғау – цифрлық тамғалардың «цифрлық ерекше белгілер» (digital watermarking) және «саусақтардың цифрлық іздері» (fingerprinting) деп аталатын түрлерін енгізуді көздейді. Олар авторлық туындыларды пайдаланудың заңдылығын немесе заңға қайшылығын дәлелдеуге мүмкіндік береді.

Саусақтардың цифрлық іздері – әр түрлі тапсырыс берушілер үшін анықталған (жасырын тіркеу нөміріне ұқсас), объект көшірмесіне енгізілген тамғалар.

Цифрлық ерекше белгілер (цифрлық су таңбалар) – мультимедиялық деректердің ішіне жасырылған авторлық құқық тамғасы. Графикалық және аудио файлдарда авторлық құқықтарды қорғауға арналған көзге көрінбейтін ерекше белгілер ендіріледі. Файлға орналастырылған осындай цифрлық ерекше белгілерді, арнайы бағдарламалардың көмегімен ажыратып алуға болады. Мұндай арнайы бағдарламалар файлдан біраз пайдалы ақпарат шығарып береді: файлдың қашан құрылғандығы, кім авторлық құқықтарға ие, оның авторымен қалай қарым-қатынас жасауға болады.

Цифрлық ерекше белгілер қалыптастыруға арналған жүйелерде, таржолақты сигнал болып табылатын, тамға ендіру қағидаты қолданылады. Бұл әдіс екі алгоритмнің және олардың жаңғыртылған түрлерінің көмегімен жүзеге асырылады. Бірінші жағдайда, деректер ақпараттық сигналды (тасығышты) сандардың жалғанкездейсоқтық тізбегімен фазалық модуляциялау арқылы жасырылады. Екінші жағдайда - жиіліктер диапозоны бірнеше арналарға бөлінеді де тасымалдау осы арналардың арасында жүргізіледі. Тамға ендіру кезінде бастапқы бейнеге біршама қосымша шуыл қосылады. Бірақ, сигналда әрқашан шуылдар болатындықтан, тамға ендіру себебінен оның аздап өсуі, көзге көрінетіндей бұрмалау

бермейді. Сонымен қатар, тамға бастапқы бейненің бәр жеріне шашыраңқыланатындықтан, оны жою үшін тамғаны іздеп-табу қиын болады.

♦Қазіргі заманғы стеганографияның тағы бір бағыты – *шын сияқты терістеу мүмкіншілігі*. Бұл жерде файлдардың стеганографиялық жүйесі, маңызды ақпаратты амалсыз жариялаудан қорғауды қамтамасыз ететін қабілеті бар тетік. Осындай тетік (тіпті егер қаскүнем жүйе қорларына толық қол жеткізе алған болса да) жеке файлдардың бар болуын шындыққа ұқсайтындай етіп, теріске шығаруға мүмкіндік береді.

♦Стеганография – *түсініктемелерді мультимедиялық файлдарға енгізу* үшін де қолданылады. Қазіргі уақытта түсініктемелер медициналық суреттерде, мультимедиялық дерекқорлар мен жарнамалық хабарландыруларда қолданылып жүр.

3.1-кестеде ашық стеганографиямен шешілетін негізгі мәселелер, жүзеге асыру технологиялары және қолданылу салалары берілген. Кестеде көрсетілгендей, қойылған мәселелерді шешуді қамтамасыз ететін негізгі технологиялар – жасырын байланыс және ЦСТ технологиялары болып табылады [5].

3.1-кесте

Ашық стеганографиямен шешілетін негізгі мәселелер

Мәселе	**Шешу технологиясы және жолдары**	**Жүзеге асырылу мысалдары**	**Қолданылу салалары**
Рұқсатсыз қатынас құрудан құпия ақпаратты қорғау	Жасырын ақпаратты жалпы қол жеткізерлік мультимедиалық ақпаратқа ендіру	Цифрланған дыбыстың (44100 Гц, 8 бит, стерео) 1 секунды 5 бет мәтіндік ақпаратты жасыруға мүмкіндік береді.	Әскери, т.б. салаларда қолдану, сондай-ақ криптографияны пайдалануға болмайтын жағдайларда қолдану

3.1-кестенің жалғасы

Мониторинг және желілік ресурстарды басқару жүйелерін орай өту	Өнеркәсіптік шпионажға қарсы әрекетке бағытталған стеготәсілдер компьютерлік желілердегі ақпаратты бақылауға қарсы тұруға мүмкіндік береді.	Internet Explorer браузері негізінде жұмыс істейтін Camera/Shy утилиті. Ол LSB стеганографиялық техникасын және 256-разрядтық кілтті AES шифрлау алгоритмін қолданады. Сондай-ақ, бұл бағдарлама хабарды gif-файлдарда жасыруға және HTML-парақтарда жасырын ақпараты бар графикалық бейнелердің бар-жоқтығын автоматты түрде сканерлеуге мүмкіндік береді	Бұл бағдарлама желіаралық экрандарды орай өтуге арналған
Бағдарламалық қамтаманы түрін өзгертіп далдалау	БҚ қолдану шектеулі болған жағдайда, ол стандарттық бағдарламаға далдаланылуы немесе мультимедиа файлдарында жасырылған болуы мүмкін	Арнайы редакторлар, дыбыстық қолдаулар, жарнама, т.б. қолданылады	Бағдарламалық қамтамаға көпдеңгейлік рұқсат етілген қатынас құру мүмкіндігі қамтамасыз етіледі
Зияткерлік меншікті көшірмелеуге байланысты авторлық құқықты қорғау және	Цифрлық су таңбалары (ЦСТ) және идентификациялық нөмірлер (ИН) технологиялары	ЦСТ қорғалатын объектіге ендіріледі және ол көрінетін де, сондай-ақ көрінбейтін де	ЦСТ және ИН технологиялары электрондық тасуыштарды көшірмелеуден қорғау

3.1-кестенің жалғасы

аутентификациялау	қолданылады	болуы мүмкін. Оның құрамында аутентикалық кода, меншік иесі туралы ақпарат және басқару ақпараты болады. ИН-дің ЦСТ-нан айырмашылығы - әрбір көшірменің өзіндік ИН-і болады	мақсатында және электрондық коммерцияда, дауыстық поштада, бейнебақылау жүйелерінде ақпаратты рұқсатсыз қолданудан қорғау мақсатында қолданылады
Құжаттарды жасырын аңдатпалау және деректер (ақпарат) банкін оңтайландыру	ЦСТ және ИН технологиялары қолданылады	Тек емдеуші дәрігерге ғана қолжетерлік электрондық медициналық құжат	Медицинада, картографияда, мультимедиалық деректер банкінде құжаттарды жасырын аңдатпалау, сондай-ақ олардағы қажетті ақпаратты іздеу үшін қолданылады

3.2-кестеде деректерді стеганографиялық жабудың бағдарламалық құралдарының мысалдары, олардың мүмкіндіктері, жұмыс істеу қағидаттары, жағымсыз жақтары және жұмыс істей алатын операциялық жүйелері келтірілген [5].

3.2-кесте

Деректерді стеганографиялық жабудың бағдарламалық құралдары

Аталымы	Мүмкіндіктері	Жұмыс істеу қағидаты	Кемшіліктері	Операциялық жүйе
OutGuess	JPEG бейнелерде деректерді жасыру	Бейне блоктарының кіші биттерінде деректерді жасыру		UNIX-тәрізді операциялық жүйелер
Gifshuffle	GIF форматтағы графикалық файлдарда деректерді жасыру	Палитрадағы түстер тәртібін өзгерте отырып ақпаратты жасыру	Контейнер өлшеміне тәуелсіз, жасырылатын хабар көлемінің аздығы	UNIX-тәрізді операциялық жүйелер
Steganos	BMP, DIB, VOC, WAV, ASCII графикалық файлдарында ақпаратты жасыру	Бейне элементтерінің кіші биттерін ауыстыру арқылы ақпаратты жасыру	Контейнерлердің ескірген пішімдерін (форматтарын) қолдану .	MS-DOS және Windows операциялық жүйелері
Invisible Secrets 2002	PCM, WAV, BMP, DIB, VOC, HTML, ASCII форматындағы аудиодеректерге деректер енгізу	Контейнер элементтерінің кіші биттерін ауыстыру арқылы, сондай-ақ ASCII форматындағы қатар соңына бос аралықтар қосу арқылы ақпаратты жасыру	Жасырылатын ақпараттың контейнер бойынша үлестірілуінің жоқтығы, контейнердің жарамдылығын алдын ала талдаудың болмауы	Windows операциялық жүйелері
Steganos for Windows			Контейнерлердің ескірген форматтарын қолдану	
Hide4PGP			Контейнердің жарамдылығын алдын ала талдаудың болмауы	
StegoWav	PCM (WAV) форматындағы аудиодеректерге деректер енгізу	Контейнер элементтерінің кіші биттерін ауыстыру арқылы ақпаратты жасыру	Дамымайтын өнім, контейнерлердің ескі форматтарын қолдану	Windows операциялық жүйелері
Steaghan	Графикалық BMP- және дыбыстық WAV-файлдарда деректерді жасыру			UNIX-тәрізді операциялық жүйелер
S-Tools	PCM (WAV) форматындағы аудиодеректерге деректер енгізу		Контейнерлердің ескірген форматтарын қолдану	Windows операциялық жүйелері
MP3Stego	MP3 форматындағы аудиодеректерге деректер енгізу	Аудиоағындарды қысу кезеңінде жиіліктік коэффициенттерді - кванттау кезінде жұптықты өзгерту тәсілінің көмегімен ақпаратты жасыру	Белсенді қарсыластың шабуылына (жасырын деректерді жою) төзімсіздігі	Windows және UNIX операциялық жүйелері
UnderMP3 Cover		Қате MP3 кадрларын контейнерге енгізу арқылы ақпаратты жасыру	Бейбелсенді стеготалдауға беріктігінің төмендігі	UNIX-тәрізді операциялық жүйелер
Covert_TCP	Бағдарлама TCP_IP бастамасын басқаралы және әрбір файлмен бірге, бір жасырын байт жіберіп тұрады	Жасырын деректерді жіберу, деректер дестелерімен де, қызметтік дестелермен де жүзеге асырылады		Linux операциялық жүйелері
SecurEngine	Мәтіндік файлдарға, графикалық файлдарға (BMP, JPG) және аудиофайлдарға (WAV) деректер енгізу			Windows операциялық жүйелері

Қазіргі уақытта, жоғарыда көрсетілгендермен қатар, мынадай стеганографиялық бағдарламалар да қолданылады: Gif-It-Up (Win95, деректерді Gif-файлдарда жасырады), EZStego (Java-қолданба, GIF және PICT форматтары үшін LSB әдісі), Contraband (Win95, BMP форматы), Steganography Tools 4 (ақпаратты DEA, MPJ2, DES, TripleDES, NSEA алгоритмдерімен шифрлайды. Одан кейін оны графикалық және дыбыстық файлдарда, сондай-ақ дискілер секторларында жасырады), Gif-It-Up 1.0 for Windows, Steganos for Win 95, Contraband, Max File Encryption, INFO Hider v1.0, StegoMagic, SecurEngine, Steganos Security Suite 2006, Steganos Privacy Suite 2008, Masker 7.0, TrueCrypt, т.б.

S-Tools – MS Windows платформасына арналған өнім. Бұл бағдарлама gif және bmp форматты бейнелерде, сондай-ақ wav форматты аудио файлдарында кез келген файлды жасыруға мүмкіндік береді. Жасыруға жататын файл алдын ала DES, 3-DES немес IDEA сияқты симметриялық кілтті криптографиялық алгоритмдердің біреуімен шифрланады. Сөйтіп, S-Tools бағдарламасында стеганография да, криптография да қолданылады. Бағдарлама терезесіне алдымен тасығыш-файл апарылады. Содан соң, осы файлдың ішіне кез келген форматты файл салынады да, құпиясөз енгізіледі, сөйтіп шифрлау алгоритмі таңдап алынады. Бағдарламаның жұмысынан кейін таңғаларлық нәтиже алынады: графикалық файл сырттай еш өзгеріссіз қалады (тек кейбір жерде түс реңктері өзгереді), дыбыстық файлда да айтарлықтай еш өзгертулер болмайды.

Соңғы жылдары компьютерлік желілердің, интернет желісінің және мультимедиалық технологиялардың қарқынды дамуына байланысты, авторлық құқықты және цифрлық түрде көрсетілетін зияткерлік меншікті қорғау өзекті мәселеге айналып отыр. Осыны ескере отырып, қазіргі уақытта көшірмелеуден қорғау және аутентификацияны қамтамасыз ету мәселелерін шешу үшін ұйымдастырушылық-заңдық шаралармен қатар, цифрлық су таңбалар (ЦСТ) технологиялары қолданылады (3.3-кесте) [5].

3.3-кесте

Цифрлық су таңбалары технологиясының негізгі ерекшеліктері

ЦСТ технологиясы	**ЦСТ ерекшеліктері**
Робастылы (РЦСТ)	Сыртқы әсерлерге орнықтылығы жоғары
Осал (ОЦСТ)	Толған контейнерді болар-болмас модификациялау кезінде-ақ бұзылады. Сигналдарды аутентификациялау үшін қолданылады. ЭЦҚ қарағанда ОЦСТ контентті біршама модификациялауға мүмкіндік береді. Тек контейнерді модификациялау фактісін көрсетіп қана қоймай, осы өзгерістің түрі мен орнын да көрсетеді
Жартылай осал (ЖОЦСТ)	Кейбір әсерлерге қатысты орнықты, ал басқаларына – орнықсыз болып келеді. ЖОЦСТ, мәселен бейнені қысуға мүмкіндік бергенімен, одан үзінді қиып алуға немесе оған үзінді қосуға мүмкіндік бермейді

Цифрлық су таңбаларды **(**ЦСТ) бейнеге (кескінге) ендіру алгоритмдеріне мыналар жатады [5]:

– *аддитивтік алгоритмдер.* Мәселен, Гаусс заңы бойынша үлестірілген (ұзындығы 1000 санды қамтитын) жалғанкездейсоқ сандар тізбегі деп аталатын алгоритм. Ол дискреттік косинустық түрлендірудің (ДКТ) ең үлкен 1000 коэффициенттерін модификациялау негізінде жұмыс істейді. Екінші бір мысал - Гаусс заңы бойынша үлестірілген (ұзындығы модификацияланатын коэффициенттер санына сәйкес), жалғанкездейсоқ нақты сандар тізбегі деп аталатын алгоритм. Бұл алгоритм бес деңгейлік вейвлет-түрлендірудің барлық ішдиапазондарындағы мәндік коэффициенттерді модификациялау негізінде жұмыс істейді. Сондай-ақ, кешендік вейвлет-түрлендірудің ең үлкен 1000 коэффициенттерін модификациялауға негізделген, Биполярлық жалғанкездейсоқ сандар массиві деп аталатын алгоритм, т.б.;

– *ЦСТ мен контейнерді біріктіру негізіндегі алгоритмдер*. Мәселен, бастапқы бейненің бірдеңгейлік декомпозияцияларының барлық коэффициенттерін модификациялау негізінде жұмыс істейтін, қара-ала логотип алгоритмі, т.б.;

– скалярлық кванттауды пайдаланатын алгоритмдер. Мәселен, бастапқы бейненің екі деңгейлік вейвлет-түрлендіруіндегі ЖЖ-ТЖ және ТЖ-ЖЖ аумақтарын модификациялау әдісін қолданатын бинарлық бейне алгоритмі, т.б.;

– *фракталдық түрлендіруді пайдаланатын алгоритмдер*. Мәселен, **б**астапқы бейнені, биттер қатарын қолданатын алгоритмдер.

Стеганография үрдісін бірнеше кезеңге бөлуге болады [11].

1. Ақпараттық файлды таңдау. Стеганография үрдісінің бірінші кезеңінде жабылуға (жасырылуға) тиісті файл таңдап алынады. Бұл файлды – *ақпараттық файл* деп те атайды.

2. Файл-контейнерді таңдау. Стеганография үрдісінің екінші кезеңінде, ақпаратты жабу (жасыру) үшін қолданылатын файл таңдап алынады. Бұл файл *файл-контейнер* деп те аталады. Ақпарат жасыру үшін файл-контейнер жадысының көлемі, ақпараттық файл жадысының көлемінен шамамен 8 есе артық болуы керек.

3. Стеганографиялық бағдарламаны (өнімді) таңдау. Стеганография үрдісінің үшінші кезеңінде стеганографиялық бағдарлама таңдап алынады. Мәселен, ең көп таралған S-Tools бағдарламасы gif, bmp және wav форматтарында кез келген файлдарды жасыруға мүмкіндік береді.

4. Файлды кодтау. Ақпараттық файл, файл-контейнер және стеганографияның бағдарламалық қамтамасы таңдап алынған соң, жаңа файлды құпиясөз көмегімен қорғау қажет.

5. Жасырылған хабарды электрондық пошта бойынша жіберу және оны кері кодтау. Бұл кезең аты айтып тұрғандай, жасырын хабарды электрондық пошта арқылы жіберу, алынған хабарды кері ашу үрдісін білдіреді.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР

1. *Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.* Основы криптографии: Учеб. пособие. – М.: Гелиос АРВ, 2002. - 480 с.

2. *Анин Б.Ю.* Защита компьютерной информации. - СПб.: БХВ – С.-Петербург, 2000.-384 с.

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

4. *Барсуков В.С., Романцов А.П.* Компьютерная стеганография вчера, сегодня, завтра. http://kiev-security.org.ua/box/12/80.shtml

5. *Барсуков В.С., Шувалов А.В.* Еще раз о стеганографии – самой современной из древнейших наук. "Специальная Техника", № 2, 2004 г.

6. *Бурдаев О.В., Иванов М.А., Тетерин И.И.* Ассемблер в задачах защиты информации/ Под ред. И.Ю. Жукова - М.: КУДИЦ-ОБРАЗ, 2002. - 320 с.

7. *Валиев К.А.* Квантовая информатика: квантовые компьютеры, связь и криптография, ФТИАН.

8. *Виноградова Л., Виноградов К.* Защита информации: квантовая криптография. http://www.citforum.ru/security/cryptography/quant_crypto

9. *Генне О. В.* Основные положения стеганографии. Защита информации. Конфидент, № 3, 2000, ООО "Конфидент".

10. *Герасименко В.А.* Защита информации в автоматизированных системах. ч. 1, 2. - М.: Высшая школа, 1994.

11. *Голубев В.* Компьютерная стеганография – защита информации или инструмент преступления ?. Crime-research.ru, LexusSPb, 09/2007.

12. *Грибунин В.Г., Оков И.Н., Туринцев И.В.* Цифровая стеганография. (Серия “Аспекты защиты”) - М.: СОЛОН-Пресс, 2002. - 272 с.

13. *Гундарь К.Ю., Гундарь А.Ю., Янишевский Д.А.* Защита информации в компьютерных системах. - К.: Корнейчук, 2000. - 152 с.

14. *Дмитриев В.И.* Прикладная теория информации. - М.: Высшая школа, 1989. - 320 с.

15. *Евангели А.* Квантовая криптография. PC Week/RE, № 43, 2003.

16. *Жельников В.* "Криптография от папируса до компьютера" -М.: ABF, 1996

17. *Задирака В.К., Абдукаликов А.* Элементы современной криптографии и методы банковской защиты.- Алматы, 1999.

18. *Зензин О.С., Иванов М.А.* Стандарт криптографической защиты – AES. Конечные поля. – М.: КУДИЦ-ОБРАЗ, 2002. – 176 с.

19. Компьютер пресс, 03/2002, 03/2003.

20. Қазақстан Республикасының ұлттық қауіпсіздігі туралы Заңы. - Алматы: Жеті жарғы, 1998.

21. Қазақстан Республикасының мемлекеттік құпиялар туралы Заңы. - Алматы: Дәнекер, 2000.

22. Қазақстан Республикасының электрондық құжат және электрондық цифрлық қолтаңба туралы Заңы. – Астана, 2003.

23. *Манаев М., Петренко С.* Технологии защиты информации в Интернете. Специальный справочник. - СПб.: Питер, 2002. - 848 с.

24. *Нечаев В.И.* Элементы криптографии. Основы теории защиты информации. - М.: ВШ, 1999.

25. Первая компьютерная сеть защищена на квантовом уровне, 2004, http://www.cnews.ru/news/top/index.shtml

26. Первый квантовый компьютер: подробности, 2007, http://www.cnews.ru/news/top/index.shtml

27. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. /Пер. с англ./ - М.: Мир, 1976. - 594 с.

28. *Пшенин Е.С.* Теоретические основы защиты информации: Учеб. пособие. - Алматы: КазНТУ, 2000. -125 с.

29. *Расторгуев С.П.* Программные методы защиты информации в компьютерах и сетях. - М.: Яхтсмен, 1993. - 188 с.

30. Расширение возможности беспроводной квантовой криптографии, 2007. http://www.securitylab.ru.

31. *Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф.* Защита информации в компьютерных системах и сетях. - М.: Радио и связь, 2001. - 328 с.

32. *Романцов А.П.* Криптография и стеганография / Под. ред. А.В. Петракова. – М.: РИО МТУСИ, 2002. -80с.

33. *Романцов А.П.* Стеганографическая защита цифровыми водяными знаками. Учеб. пособие /Под ред. А.В. Петракова. – М.: РИО МТУСИ, 2003. -68с.

34. *Рябко Б.Я., Фионов А.Н.* Криптографические методы защиты информации: Учеб. пособие. – М.: Горячая линия-Телеком, 2005. - 229 с.

35. *Санько С.* Квантовая криптосистема на CeBIT', 2002, . http://www.kv.minsk.by/index.htm.

36. *Слепов Н.* Квантовая криптография: передача квантового ключа. Проблемы и решения // Электроника: Наука, Технология, Бизнес. №2. 2006.

37. *Смарт Н.* Криптография. – М.: Техносфера, 2005. -528 с.

38. *Тұрым А.Ш.* Ақпараттану және есептеу техникасының түсіндірме сөздігі. Ы. Алтынсарин атындағы Қазақтың білім академиясының Республикалық баспа кабинеті, Алматы, 2000, 192 б.

39. *Шеннон К.* Работы по теории информации и кибернетике/ Пер. с англ. под ред. Р.Л.Добрушина и О.Б.Лупанова. - М.: ИЛ, 1963. с.244 - 332.

МАЗМҰНЫ

Оқулық басылым

Тұрым Асқын Шамұлы
Мұстафина Бақытжан Мұхамеджанқызы
Шайқұлова Ақтоты Әлиқызы

АҚПАРАТ ЖАБУ НЕГІЗДЕРІ

Оқу құралы

РББ бастығы *З.А. Ғұбайдуллина*
Редакторы *К.Мүптекеқызы*
Компьютерде беттеген *Л.Т. Қасжанова*

Басуға 26. 12. 2011 ж. қол қойылды

Таралымы 100 дана. Пішімі 60х84 1/16. № 1 баспаханалық қағаз.
Көлемі 10,3 оқу-баспа табақ. Шартты б.т. 9,6.Тапсырыс №
Бағасы келісімді.

Қ.И.Сәтбаев атындағы
Қазақ ұлттық техникалық университетінің басылымы,
Ақпараттық-баспа орталығы, Алматы, Ладыгин көшесі, 32

ISBN 978-601-228-275-7