Лекция 3 ЖАБЫҚ КІЛТІ БАР БЛОКТЫ ШИФРЛАРДЫ ҚҰРУ ПРИНЦИПТЕРІ

3.1 Композициялық шифрдың ұғымы

Бірнеше қатарынан пайдаланған қарапайым шифрлардың комбинациясы (мысалы, орын ауыстыру немесе орнына қою) нәтижесінде **құрастырылған** (**композициялық) шифры** деп аталатын одан күрделі түрлендіруді береді. Бұл шифрдың криптографиялық мүмкіндіктері, жеке орын ауыстыруға немесе орнына қоюға қарағанда, артығырақ болады.

2-ші сабақтағы мысалды қайтадан еске түсірейік, онда тіркелген периоды бар орын ауыстыру әдіспен шифрлау жүргізіледі. Орын ауыстыру периоды d=6 болсын, ал кілт K тең 436215. Бұл мынаны көрсетеді: әрбір алты символдан тұратын блокта төртінші символ бірінші орынға тұрады, үшінші – екінші орынға, алтыншы – үшіншіге және т.б. Таңдап алынған кілт көмегімен СИГНАЛ сөзді шифрлайық:

Қарсыласқа шифрлау әдісі белгілі деп, ал кілтті ол білмейді деп санайық. Егер қарсыластың қолына НГЛИСА хабар түссе, оған ең көбі 720 әрекет қажет болады (толық іріктеп алу әдісті пайдаланғанда). әрбір вариантқа 1 секунд кетсін, онда барлық 720 әрекетке 12 минут қана жұмсалады. Сонымен, ең көбі 12 минуттан кейін қарсылас біздің кілтті біле алады және барлық хабарларды ашып оқиды. Компьютер көмегімен кілтті іздегенде одан да аз уақыт қажет болады.

Біздің шифрдың криптоталдау есебін қалай күрделендіруге болады? Орын ауыстыру периодтың мөлшерін үлкейтуге болады, яғни ішінде символдар ауыстырылатын блокты, мысалы мың белгіге дейін. Бірақ, біріншіден, жүздеген және мыңдаған белгілерді іріктеп алу қазіргі компьютерде бір минуттың ішінде орындалады, екіншіден, бұл жағдайда мың символға дейін кілт ұзындығы да өседі. Осындай кілтті есте қалдыруға да пайдалануға да қиын.

Басқа жолмен барып көрейік, алты символдан тұратын блокта орын ауыстырудын алдында, Цезарь әдісі бойынша қарапайым орнына қоюды қолданылайық. Цезарь әдісіндегі кілтті k_1 ($1 \le k_1 \le 31$), ал орын ауыстыру кілтін k_2 деп белгілейік. Онда ортақ кілт K = (k_1, k_2). Сонымен, егер K = (5, 436215) болса, онда алдымен кілті 5 бар Цезарь әдісі бойынша символдар ауыстырылады, сосын алты символды әрбір блокта 436215 кілтпен орын ауыстыру жүргізіледі. Екі кезеңмен СИГНАЛ сөзді шифрлайық:

1-ші кезең (ауыстыру): *ÑÈÃÍÀË* $\xrightarrow{k1=5}$ *ÖÎÈÒÅÐ*

2-ші кезең (орын ауыстыру): *ÖÎÈÒÅÐ* $\xrightarrow{k\,2=436215}$ *ÒÈÐÎÖÅ*

Былай да жазуға болады:

$$СИГНАЛ \xrightarrow{K=(5,436215)} ТИРОЦЕ$$

Біздің жағдайда Цезарь шифрындағы мүмкін болатын кілттер саны тең 31, сондықтан қолданылған құрастырылған шифрда мүмкін кілттердің жалпы вариантар саны (кілттер кеңістігі) тең 31·720=22320. Сонымен, шынында да алынған құрастырылған шифры жеке орындалған ауыстыру мен орын ауыстыру шифрлардан күштірек.

Статистикалық әдіспен криптоталдауды қиындату үшін біздің құрастырылған шифрды бір кілтпен екі рет пайдалануға болады:

Шифрлау циклы 1:

1-ші кезең (ауыстыру): $ÑÈÃÍÀË \xrightarrow{k1=5} ÖÎÈÒÅÐ$

2-ші кезең (орын ауыстыру): $ÖÎÈÒÅÐ \xrightarrow{k\,2=\ 436215} ÒÈÐÎÖÅ$

Шифрлау циклы 2:

1-ші кезең (ауыстыру): $ÒÈÐÎÖÅ \xrightarrow{k1=5} ×ÎÕÓÛË$

2-ші кезең (орын ауыстыру): $×ÎÕÓÛË \xrightarrow{k\,2=\ 436215} ÓÕËÎ×Û$

Екі қатарынан орындалған шифрлау циклы нәтижесінде СИГНАЛ сөзі УХЛОЧЫ – ға айналды. Мұнда шифрдың кілттер кеңістігі өзгермеді, бірақ бастапқы мәтіннің статистикалық заңдылықтары күштірек жасырынды.

Нақты шифрларда да белгілер блоктарына бірнеше қарапайым операциялардың комбинациясы пайдаланылады. Криптоберіктікті күшейту үшін бұл операциялар циклды түрде бірнеше рет орындалады, мұны раунд немесе қадамдар деп атайды. Шифрдың беріктігіне әсер етеді блок мөлшері, кілт мөлшері, шифрлау раундтар саны. Қазіргі жабық кілті бар шифрлар тек екілік деректерді өңдейді, сондықтан оларда кәдімгі ауыстыру мен орын ауыстырудан басқа кейбір екілік сандар үшін арнайы операциялар қолданылады.

Симметриялық шифрлау алгоритмы бастапқы мәтінді блоктармен немесе ағындармен өңделу мүмкін. Осыған байланысты симметриялық шифрлаудың блокты және ағынды алгоритмдарын ажыратады. Мәтін блогы теріс емес бүтін сан ретінде немесе

бірнеше тәуелсіз теріс емес бүтін сандар ретінде қарастырылады. Блок ұзындығы әрқашан екінің дәрежесіне тең деп алынады, мысалы, 64, 128, 256 бит.

3.2 Симметриялық шифрлаудың блокты алгоритмында пайдаланатын операциялар

Симметриялық шифрлаудың алгоритмдарының көбінде пайдаланылатын операцияларды қарап шығайық. Қандай да болса ақпарат, мысалы бейне немесе мәтін, екілік түрде ұсынылу мүмкін.

Жиі пайдаланатын операциялардың біреуі – модулі 2 бойынша биттық қосу операциясы, ол XOR немесе деп белгіленеді. Модулі 2 бойынша қосуда операндтар разряд ретімен өңделеді. Нәтиженің разрядына бір қойылады, егер операндатардың сәйкес разрядында бірліктер саны тақ болса. Мысалы, модулі 2 бойынша екі 16-разрядты санды қосайық:

Разряд номірі	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Операнд 1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Операнд 2	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Модулі 2 бойынша қосынды	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1

Осы операцияда өте ыңғайлы қасиеті бар: модулі екі бойынша азайту қосындымен бірдей, сондықтан операндтардың біреуі басқа операндтың қосындысына қосу арқылы табылу мүмкін.

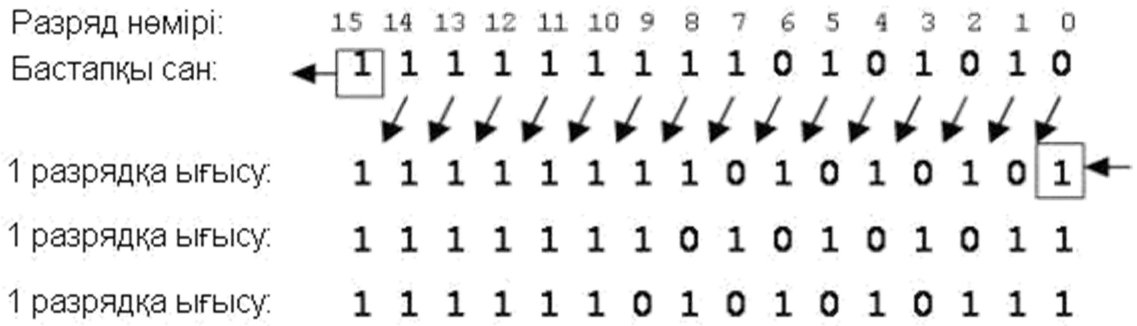
Және де блокты шифрлау алгоритмда жиі қолданылады модулі 2^{32} немесе модулі 2^{16} бойынша қосу операциясы. Бұл операция кәдімгі екілік сандардың қосуы, нәтиженің разрядын жоғары 32-ші немесе 16-шы разрядқа тасымалдауын еске алмасақ. Мысалы, модулі 2^{16} бойынша екі 16-разрядты санды қосайық:

Разряд номірі	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Операнд 1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Операнд 2	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Модулі 2^{16} бойынша қосынды (1)	0	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1

15-ші разрядтан тасымалдау, мысалда жақшадағы бір ретінде белгіленген, әрі қарай пайдаланбайды және сол себептен лақтырып тасталынады.

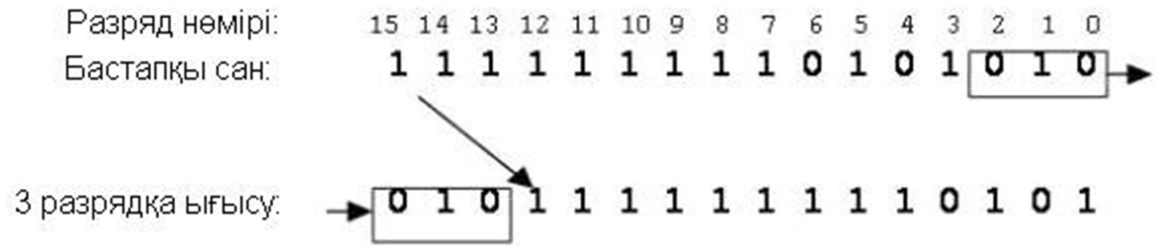
Циклдық ығысу бит тізбегін бірнеше разрядқа солға немесе оңға қарай жылжытады. Ығысу операциясын орындағанда, екілік сан туннельдің бір жағынан жорғалап кіріп жатқан және екінші жағынан жорғалап шығып жатқан ұзын құртқа ұқсайды. Солға қарай циклдық ығысуда сол жақтан кестеден сыртқа шығатын биттер оң жағынан босаған орындарға жазылады. Оңға қарай циклдық ығысуда барлық биттер оңға жылжыйды, ал орын жетпегендіктер тізбектін құйрығына жазылады. Мысалы, екілік санды 3 разрядқа солға қарай циклдық ығыстырайық. Ол үшін екілік цифрларды солға қарай 1 разрядқа ығыстыра отырып үш рет қайталап жазамыз және 15-ші разрядтан шығатын белгілерді нольдік орнына тасымалдаймыз.

3 разрядқа солға қарай циклдық ығысу (←)



Осыған ұқсас оңға қарай циклдық ығысу да орындалады. Мысалы, 3 разрядқа оңға қарай ығысуда бастапқы санның нөлдік, бірінші және екінші биттері разрядты тордан шығып кетеді және жадында сақталады, қалған барлық биттер оңға қарай 3 позицияға жылжыйды, сосын жадтағы цифрлер он үшінші, он төртінші және он бесінші орынға жазылады.

3 разрядқа оңға қарай циклдық ығысу (→)



Кестелік орнына қоюды орындаған кезде бит тобы басқа биттер тобына қайтарылады. Бұл операцияда екілік деректердің бір блогы белгілі ереже немесе кесте бойынша басқа блокпен ауыстырылады. Мысалы, үш екілік цифрдан тұратын әрбір топты басқа үш цифрдан тұратын топқа ауыстыруға болады мына кесте бойынша:

Кіру	**Шығу**
000	011
001	101
010	000
011	111
100	010
101	110
110	001
111	100

Егер «Кіру» және «Шығу» бағандағы әрбір мәнің екілік емес, ондық түрде жазатын болсақ, онда осы ауыстыру кестесін қысқаша жазуға болады, мысалы былай:

```
0->3, 1->5, 2->0, 3->7, 4->2, 5->6, 6->1, 7->4
```

Осындай жазудағы бірінші цифр кірудегі мәнің көрсетеді, ал екіншісі – шығудағы. Егер кіру мәндері өсу бойынша реттелсе, онда бірінші цифрды жазбай, тек сәйкес шығу мәндерін жазуға болады:

3, 5, 0, 7, 2, 6, 1, 4.

Яғни 3-битты блоктың мәні үшін ауыстыру ретінде ауыстыру кестеден реттік нөмірі ауыстырылатын блоктың мәніне тең болатын элемент таңдап алынады.

Егер төрт екілік цифрдан тұратын топтарды ауыстыру қажет болса, онда ауыстыру кестеде 16 мәні болу керек. Жалпы жағдайда *n*-битты блоктар үшін ауыстыру кестеде 2^n элемент болу керек.

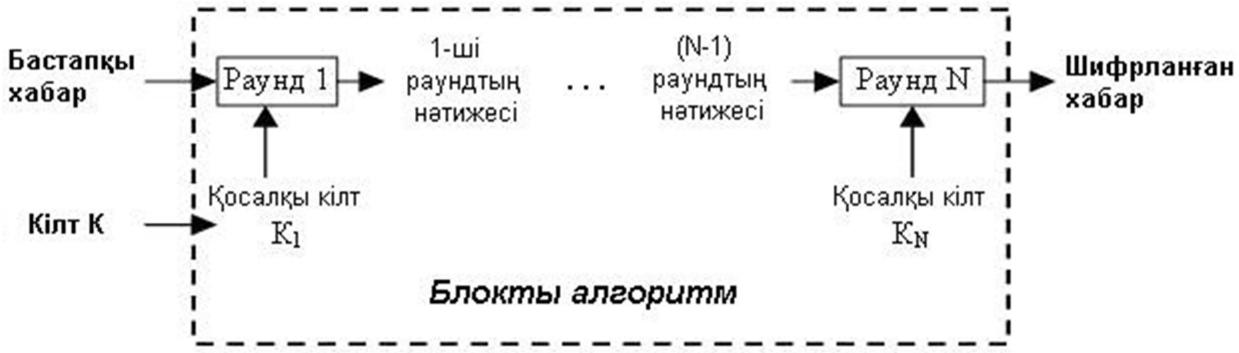
Кестелік орнына қоюды әдебиетте кейде *S*-блокты немесе *S*-box пайдалануымен ауыстыру деп атайды (*S* әрпі ағылшын сөзден substitution – ауыстыру алынған).

Орын ауыстыру операция көмегімен хабардың биттері қайталап реттеледі. Орын ауыстыруды тағы permutation немесе *P*-блогы деп атайды.

3.3 Симметриялық шифрлаудың блокты алгоритмының құрылымы

Сонымен, симметриялық шифрлау алгоритмда жиі пайдаланады модулі 2 бойынша қосу, модулі 2^{16} немесе 2^{32} бойынша қосу, циклдық ығысу, орнына қою (ауыстыру) және орын ауыстыру операциялар.

Бұл операциялар алгоритмда *N* рет циклдық қайталанып, раунд немесе қадамдарды құрады. Әрбір раунд үшін бастапқы мәліметтер: бұрынғы раундтың шығуы және белгілі алгоритмы бойынша ортақ шифрлау кілттен *K* алынған кілті болып табылады. Раунд кілті қосалқы кілт K_i деп аталады. Нәтижесінде блокты шифрлау алгоритмы келесі түрде келтірілу мүмкін (сур. 3.1):



Сурет 3.1. Симметриялық шифрлаудың блокты алгоритмының құрылымы

Блокты шифрлау алгоритмдар екілік деректерге қолданылады. Жалпы жағдайда блокты шифрлаудың процедурасы ашық мәтіннің *n*-битты блогын шифрланған мәтіннің *k*-битты блогына түрлендіреді. *n* ұзындығы бар блоктар саны тең 2^n. Түрлендіру қайтымды болу үшін, осындай блоктардың әрбіреуі өз бірегей шифрланған мәтіннің блогына түрлендірілу қажет. Блок ұзындығы әрқашан екінің дәрежесіне тең болып таңдап алынады, мысалы 64, 128, 256 бит.

3.4 Блокты шифрлау алгоритмға қойылатын талаптар

Қазіргі блокты шифрлау алгоритмға қойылатын негізгі талаптарды қарастырайық:

1. Алгоритм беріктіктің жоғары деңгейін қамсыздандыру қажет және бұл беріктік алгоритмның жасырынып сақталуына негізделмеу керек.

2. Бастапқы хабардың азын-аулақ өзгертуі шифрланған хабардың елеулі өзгеруіне келтіру керек, бірдей кілтті пайдаланғанда да.

3. Таңдаған мәтін бойынша шабуылдарға алгоритм ойдағыдай қарсы тұру керек, яғни көп жұптарды (шифрланған хабар, шифрланбаған хабар) білгенде де кілтті анықтай алмауға.

4. Шифрлау алгоритмы қандай да болса түрлі талаптарды ұсынатын платформада жүзеге асырылу болу керек. Ең жылдамды қосымшалар үшін арнайы аппаратура пайдаланылады. Бұған қарамастан, бағдарламалық жүзеге асыруы да жиі қолданылады. Сондықтан, алгоритм әмбебап микропроцессорларда тиімді бағдарламалық жүзеге асыруға да мүмкіндік беру керек. Және де микроконтроллер мен басқа орта мөлшерлі процессорларда жұмыс істеу керек.

5. Алгоритм микропроцессорларда тиімді болатын қарапайым операцияларды пайдалану керек, яғни шығарылған немесе, қосуды, кестелік орнына қоюды, модулі бойынша көбейтуді. Айнымалы ұзындығы бар ығысулыр, бит бойынша ауыстырулар немесе шартты тасымалдаулар пайдаланбау керек.

6. Алгоритм шифрлау мен дешифрлауды орындауға арналған мамандандырылған аппаратурада тиімді жүзеге асырылу керек, яғни электронды құрылғылар түрінде алгоритмның жүзеге асырылуы үнемді болу қажет.

7. Шифрлау алгоритмы көптеген қосымшаларда қолданбалу керек. Деректер файлдарын немесе деректердің үлкен ағындарын шифрлағанда, кездейсоқ биттердің белгілі санын жасаған кезде алгоритм тиімді болу керек. Және де оның бір жақты хеш-функцияны құрастыру үшін пайдалану мүмкіндігі болу қажет. Хэш-функция – бұл еркін ұзындығы бар жол үшін кейбір бүтін мәнді немесе кейбір тұрақты ұзындығы бар басқа жолды есептейтің математикалық немесе басқа функция.

8. Алгоритм кодты жазу үшін жеңіл болу керек, сонда бағдарламалық қателіктер мүмкіндігі азаяды. Сонымен бірге бұл талдауға мүмкіндік береді және алгоритмның жабығын азайтады.

9. Алгоритм қажетті ұзындығы бар кез келген биттер жолын мүмкін кілт ретінде пайдалануға рұқсат беру керек (бұл кілттердің жазық кеңістігі деп аталады). Криптоталдауды оңайлататың «осал» кілттер болмау керек.

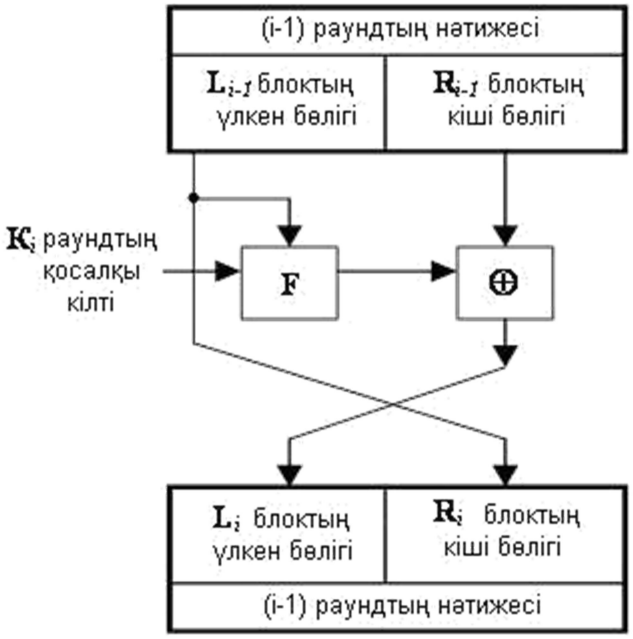
10. Алгоритм түрлі қауіпсіздік деңгейлері үшін түрін оңай өзгерте алу керек және минимал мен максимал талаптарға қанағаттандырылу болу қажет.

3.5 Фейштель желісі

3.1 суретте блокты шифрлау алгоритмның жалпы құрылымы келтірілген болатын. Деректердің түрлендіруі раундта немесе шифрлау қадамдарында жасалынатыны әрине түсінікті. Барлық алгоритмды орындаған кезде сенімді шифрланған деректерді алу үшін бір раундта қандай іс-әрекеттер жасау керек?

Блокты шифрдың құрастыру принциптерін зерттеуге үлкен үлес американ ғалымы Х.Фейштель (Horst Feistel) қосты. Ол, мысалы, IBM фирманың шифрлау жүйесін «Люцифер» құрастыруына қатысқан болатын. Фейштель, қазір Фейштель желісі деп аталатын, құрылымды ұсынды. Фейштель желілері кең таралған, өйткені бір жағынан олар симметриялық шифрлауға қойылатын барлық талаптарға қанағаттандырылады, ад екінші жағынан жеткілікті қарапайым және пайдалануда ыңғайлы.

Фейштель желі бойынша ұйымдастырылған раундтың келесі құрылымы бар. Кіру блогы бірнеше бірдей ұзындығы бар бөлшектерге бөлінеді. Бұл бөлшектер *бұтақ* деп аталады. Мысалы, блок ұзындығы 64 бит болса, 32 битты екі бұтақ пайдаланады. Бұтақтар жеке өңделеді, сосын барлық бұтақтарды солға қарай циклдық ығыстырады. Екі бұтағы бар жағдайда әрбір раундтың 3.2 суретте көрсетілгендей құрылымы болады.



Сурет 3.2. Фейштель желінің *i*-шы раунды

F функциясы *құрастырушы* деп аталады. Әрбір раундта бір бұтақ үшін *F* функция есептеледі және *F* нәтиженің басқа бұтақпен бит бойлы «модулі 2 бойынша қосынды» операциясы орындалады. Осыдан кейін бұтақтар орындарын айырбастайды. Раундтар саны түрлі алгоритмы үшін әртүрлі болу мүмкін. Кейбір алгоритмда 8 ден 32-ге дейін раунд ұсынады, басқаларда – одан артық. Толығымен айтқанда, раундтар санның өсуі алгоритмның криптоберіктігінің өсуіне әкеледі. Сол себептен Фейштель желісі кең таралу, өйткені криптоберіктікті күшейту үшін тек алгоритмды өзгертпей раундтар саның көбейту керек.

Соңғы уақытта көбінесе 128-битты блоғы үшін төрт бұтақты Фейштель желісі пайдаланылады. Бұтақтар саның көбейтуі 32-разрядты сөздерге байланысты.

Фейштель желі негізінде құрастырылған блокты алгоритмда негізгі операция құрастырушы *F* функцияны есептеу. Бұл функция нәтижені есептеу үшін раундтың қосалқы кілтін және кіру блогының бір бұтағын пайдаланады. Шифрлау жүйелері бір бірінен ерекшеленеді *F* функцияның анықтауымен.

Негізгі ұғымдар

Құрастырылған (композициялық) шифр - бірнеше қатарынан пайдаланған қарапайым шифрлардың комбинациясы нәтижесінде деректердің криптографиялық түрлендіруі.

Кілт – хабарларды шифрлауға және ашып оқуға қажетті ақпарат.

Шифр – бастапқы құпиялы хабарды қорғау үшін оның алдын ала келіскен түрлендіру тәсілдерінің жиынтығы.

Жабық кілті бар шифрлау (симметриялық шифрлау) – деректерді қайтымды түрлендіру әдістер, оларда бірдей кілт пайдаланады. Бұл кілтті ақпарат алмасудың екі жағы да қарсыластан жасырыну түрде сақтау керек.

Сұрақтар

1. Қандай шифр құрастырылған немесе композициялық шифр деп аталады?
2. Блокты шифрлау алгоритмның беріктігіне қандай факторлар әсер етеді?
3. Блокты шифрлау алгоритмда қандай қарапайым операциялар қолданылады?
4. Ағымды шифрлардан блокты шифрлау алгоритмның ерекшелігі неде?

5. Шифрлау алгоритмның «раунды» дегенді қалай түсінесіз?
6. Блокты шифрлау алгоритмға қойылатын талаптар қандай?
7. Неге блокты шифрлау алгоритмда қарапайым және түсінікті құрылым болу керек?
8. Шифрлау алгоритмның «жоғары криптоберіктігі» талапты қалай түсінесіз?
9. Фейштель желісі деген не?

Әдебиеттер:

1. Бабаш А.В. Информационная безопасность. Лабораторный практикум: Учебное пособие / А.В. Бабаш, Е.К. Баранова, Ю.Н. Мельников. - М.: КноРус, 2013. - 136 с.
2. Гафнер В.В. Информационная безопасность: Учебное пособие / В.В. Гафнер. - Рн/Д: Феникс, 2010. - 324 с.
3. Громов Ю.Ю. Информационная безопасность и защита информации: Учебное пособие / Ю.Ю. Громов, В.О. Драчев, О.Г. Иванова. - Ст. Оскол: ТНТ, 2010. - 384 с.
4. Ефимова Л.Л. Информационная безопасность детей. Российский и зарубежный опыт: Монография / Л.Л. Ефимова, С.А. Кочерга. - М.: ЮНИТИ-ДАНА, 2013. - 239 с.
5. Партыка Т.Л. Информационная безопасность: Учебное пособие / Т.Л. Партыка, И.И. Попов. - М.: Форум, 2012. - 432 с.