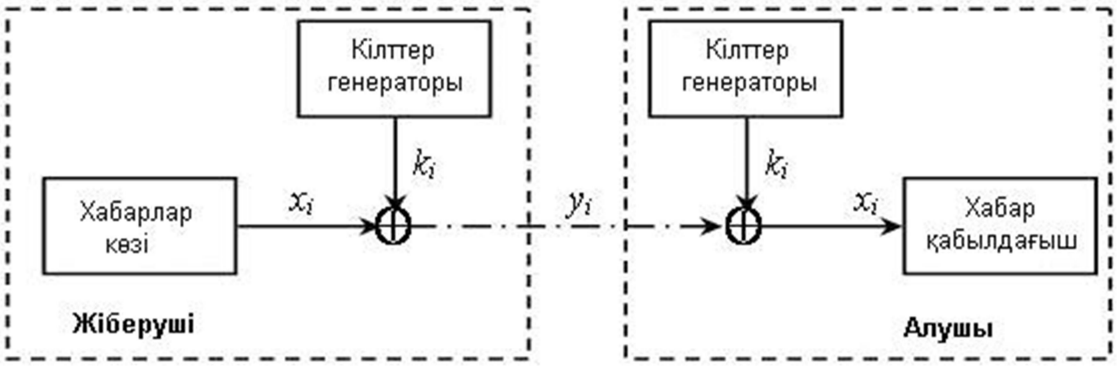
Лекция 7 АҒЫНДЫ ШИФРЛАР ЖӘНЕ КЕЗДЕЙСОҚ ТӘРІЗДЕС САНДАРДЫҢ ГЕНЕРАТОРЛАРЫ

Ағынды шифрлар

Блокты алгоритм белгілі бір ұзындығы бар блоктарды шифрлауға арналған. Бірақ, деректерді блоктармен емес, мысалы, символдар бойынша шифрлаудың қажеті болу мүмкін. **Ағынды шифр (stream cipher)** кіру хабардың биттерін (немесе байттарын) бір-бірден түрлендіріп бір операцияны орындайды. Ағынды шифрлау алгоритмы хабарды бүтін сандар блоктарға бөлуді қажет етпейді, сондықтан ол нақты уақытта істей алады. Сонымен, егер символдар ағыны берілсе, әрбір символ шифрланып бірден беріледі.

Типті ағынды шифрдың жұмысы 7.1 суретте көрсетілген.



Сурет 7.1. Ағынды шифрдың жұмыс принципі

Кілттер генераторы биттер ағының k_i шығарады, олар гамма ретінде пайдаланатын болады. Хабар көзі ашық мәтіннің x_i биттерін генерациялайды, олар гаммамен модулі 2 бойынша қосылады, нәтижесінде шифрланған хабардың y_i биттері алынады:

$$y_i = x_i \oplus k_i, \quad i = 1, 2, \ldots, n$$

Шифрмәтіннен $y_1, y_2, \ldots, y_n$ хабарды $x_1, x_2, \ldots, x_n$ қалпына келтіру үшін тура шифрлау кезіндегідей кілттік тізбекті k_1 генерациялау қажет $y_k, \ldots, k_n$, және ашып оқу үшін

$$x_i = y_i \oplus k_i, \quad i = 1, 2, \ldots, n$$

формуланы пайдалану керек.

Әдетте бастапқы хабар мен кілттік тізбегі тәуелсіз бит ағыны болып табылады. Сонымен, барлық ағынды шифрлар үшін шифрлайтын (және дешифрлайтын) түрлендіруі бірдей болғандықтан, олар тек кілттер генераторын жасау тәсілдерімен ажыратылады. Осыдан, жүйенің қауіпсіздігі кілттер ағын генератордың қасиетіне толық тәуелді болады. Егер кілттер ағын генераторы тек нөлден (немесе бірден) тұратын тізбекті жасап шығарса, онда шифрланған хабар тура бастапқы биттер ағыны сияқты болады. Егер гамма ретінде бір символ (мысалы, сегіз битты) пайдаланса, онда шифрланған хабар бастапқыға ұқсамасада, жүйенің қауіпсіздігі нашар болады. Бұл жағдайда мәтін ұзындығы бойынша

ПКГС ағынды шифрларда кілттер генераторлар ретінде пайдалану мүмкін. Псевдокездейсоқ сандар генераторын пайдалану мақсаты салыстырмалы кішкене кілт ұзындығы бар болғанда «шексіз» кілттік сөзді алу. Псевдокездейсоқ сандар генераторы кездейсоққа ұқсайтын биттер тізбегін жасайды. Шынында, осындай тізбектер белгілі бір ереже бойынша есептеледі және кездейсоқ болмайды, сондықтан олар жіберуші мен алушы жағында да дәл қайталану мүмкін. Егер кілттер генераторы қосылған кезде бірдей биттер тізбегін жасайтын болса, онда осындай жүйені бұзу оңай. Олай болса, ағын кілттер генератордың шығысы кілт функциясы болу керек. Бұл жағдайда шифрлау кезіндегі пайдаланған кілтті білгенде ғана хабарларды ашып оқуға болады.

Криптографиялық мақсатта пайдалану үшін псевдокездейсоқ сандар генератордың келесі қасиеттері болу керек:

1. тізбек периоды өте үлкен болу керек;
2. жасалынатың тізбек нағыз кездейсоққа өте жақын болу керек;
3. түрлі мәндерді жасау ықтималдықатры дәл тең болу керек;
4. тек заңды алушы ғана хабарды ашып оқу үшін, кілттік биттер ағынды k_i алған кезде кейбір құпиялы кілтті пайдалану және еске алу керек.

Айтылған қасиеттер бар болғанда псевдокездейсоқ сандар тізбегі ағынды шифрларда пайдалану мүмкін.

7.3 Псевдокездейсоқ сандардың сызықты конгруэнтті генераторы

Псевдокездейсоқ сандардың генераторы әртүрлі алгоритм бойынша жұмыс істеу мүмкін. Ең қарапайым генератордың біреуі **сызықты конгруэнтті генераторы**, ол келесі k_i есептеу үшін

$$k_i = (a\, k_{i-1} + b) \bmod c,$$

формуланы пайдалайды, мұндағы a, b, c — кейбір константалар, ал k_{i-1} — алдыңғы псевдокездейсоқ сан.

k_1 мәнің алу үшін бастапқы k_0 мәні беріледі.

Мысал ретінде алайық a=5, b=3, c=11 және k_0=1 болсын. Бұл жағдайда жоғары келтірілген формула арқылы біз 0 ден 10-ға дейін мәндер алаламыз (себебі c=11). Тізбектің бірнеше элементтерін есептейік:

$$\begin{aligned} k_1 &= (5\ 1 + 3) \bmod 11 = 8;\\ k_2 &= (5\ 8 + 3) \bmod 11 = 10;\\ k_3 &= (5\ 10 + 3) \bmod 11 = 9;\\ k_4 &= (5\ 9 + 3) \bmod 11 = 4;\\ k_5 &= (5\ 4 + 3) \bmod 11 = 1. \end{aligned}$$

Алынған мәндер (8, 10, 9, 4, 1) кездейсоқ сандарға ұқсайды. Бірақ, келесі мән k_6 қайтадан 8-ге тең болады:

$$k_6 = (5\ 1 + 3) \bmod 11 = 8, \text{ ал } k_7$$

және k_8 мәндері 10 мен 9 сәйкес тең болады:

$$k_7 = (5\ 8 + 3) \bmod 11 = 10;$$
$$k_8 = (5\ 10 + 3) \bmod 11 = 9.$$

Сонымен, біздің псевдокездейсоқ сандар генераторымыз периодтық сандарды 8, 10, 9, 4, 1 жасай отырып қайталанады.

Өкінішке орай, бұл қасиет барлық сызықты конгруэнтті генераторларға сипатты. Негізгі параметрлер мәнің a, b және c өзгертіп, период ұзындығына және k_i мәндеріне әсер етуге болады. Мысалы, c санның өсуі жалпы жағдайда период өсуіне келтіреді. Егер a, b және c параметрлер дұрыс таңдап алынса, онда генератор максимал c периоды бар кездейсоқ сандарды туғызып отырады. Бағдарламалық жүзеге асырғанда, c мәні әдетте 2^b немесе 2^{b-1} тең болып қойылады, мұнда b — сөз ұзындығы, бит.

Псевдокездейсоқ сандардың сызықты конгруэнтті генераторлардың артықшылығы оның оңайлығы және псевдокездейсоқ мәнің алудың жоғары жылдамдығы. Сызықты конгруэнтті генераторлар модельдеу және математикалық статистика есептерді шешу үшін қолданылады, бірақ криптографиялық мақсатта оларды пайдалануға ұсынуға болмайды, себебі криптоталдау мамандары ПКС тізбегін бірнеше мәні арқылы қалпына келтіруді үйренген.

Мысалы, қарсылас k_0, k_1, k_2, k_3 мәндерін анықтай алсын. Онда:

$$k_1 = (a\, k_0 + b) \bmod c$$

$$k_2 = (a\, k_1 + b) \bmod c$$

$$k_3 = (a\, k_2 + b) \bmod c$$

Осы үш теңдеуден тұратын жүйені шешіп, a, b және c табуға болады.

Псевдокездейсоқ сандарды алу үшін квадраттық және кубтық генераторлар:

$$k_i = (a_1^2\, k_{i-1} + a_2 \cdot k_{i-1} + b) \bmod c$$

$$k_i = (a_1^3\, k_{i-1} + a_2^2\, k_{i-1} + a_3\, k_{i-1} + b) \bmod c$$

ұсынған болатын. Бірақ оларды да «алдын ала білу» себебінен криптографияда пайдалануға болмайды.

7.4 Кешігуі бар Фибоначчи әдісі

Псевдокездейсоқ сандарды алудың басқа да схемалары бар.

Кешігуі бар Фибоначчи әдісі (Lagged Fibonacci Generator) — псевдокездейсоқ сандарды генерациялау әдістерінің біреуі. Ол псевдокездейсоқ сандардың жоғары «сапасын» алуға мүмкіндік береді. Фибоначчи датчиктерде нақты сандармен арифметикалық операциялардың жылдамдығы бүтін санды арифметика жылдамдығымен тең болғандықтан, олар жиі пайдаланады.

Фибоначчи датчиктің ең кең таралғанның біреуі келесі рекуррентты формулаға негізделген:

$$k_i = \begin{cases} k_{i-a} - k_{i-b}, & \text{әгәәд } k_{i-a} \geq k_{i-b} \\ k_{i-a} - k_{i-b} + 1, & \text{әгәәд } k_{i-a} < k_{i-b} \end{cases}$$

мұндағы k_i — [0,1] диапазондағы нақты сандар; a, b — оң бүтін сандар, генератор параметрлері. Жұмыс үшін Фибоначчи датчигіне бұрыңғы генерацияланған кездейсоқ сандарды $\max\{a,b\}$ білу қажет. Бағдарламалық жүзеге асыруда генерацияланған кездейсоқ сандарды сақтау үшін a мен b параметрлерге тәуелді белгілі бір жад көлемі қажет болу керек.

Мысал. Кешігуі бар Фибоначчи әдісі арқылы генерацияланатын алғашқы он сандар тізбегін есептейік. Келесі бастапқы мәілметтерде $a = 4$, $b = 1$, k_0=0.1; k_1=0.7; k_2=0.3; k_3=0.9; k_4=0.5 бастаймыз k_5-тен:

$$k_5 = k_1 - k_4 = 0.7 - 0.5 = 0.2;$$

$$k_6 = k_2 - k_5 = 0.3 - 0.2 = 0.1;$$

$$k_7 = k_3 - k_6 = 0.9 - 0.1 = 0.8;$$

$k_8 = k_4 - k_7 + 1 = 0.5 - 0.8 + 1 = 0.7$; $k_9 = k_5 - k_8 + 1 = 0.2 - 0.7 + 1 = 0.5$; $k_{10} = k_6 - k_9 + 1 = 0.1 - 0.5 + 1 = 0.6$; $k_{11} = k_7 - k_{10} = 0.8 - 0.6 = 0.2$;

$$k_{12} = k_8 - k_{11} = 0.7 - 0.2 = 0.5;$$

$k_{13} = k_9 - k_{12} + 1 = 0.5 - 0.5 + 1 = 1$; $k_{14} = k_{10} - k_{13} + 1 = 0.6 - 1 + 1 = 0.6$.

Көрініп тұр, генерацияланған сандар тізбегі кездейсоққа сырттай ұқсайды. Шынында да, зерттеуден белгілі, алынған кездейсоқ сандардың статистикалық қасиеттері жақсы.

Кешігуі бар Фибоначчи әдісі бойынша құрастырылған генераторлар үшін ұсынылған a және b параметрлер бар. Мысалы, зерттеушілер келесі мәндерді ұсынады: (a,b) = (55, 24), (17, 5) немесе (97,33). Алынған кездейсоқ сандардың сапасы a константа мәніне тәуелді: ол неғұрлым үлкен болса, кездейсоқ векторлар сақталынатын кеңістіктің өлшемі соғұрлым жоғары болады. Сонымен бірге, a константаның өсуімен алгоритм пайдаланатын жад көлемі де өседі.

Нәтижесінде (a,b) = (17,5) мәндері қарапайым қосымшалар үшін ұсынылады. (a,b) = (55,24) мәндері көптік криптографиялық алгоритмдарға қанағаттанған сандарды алуға мүмкіндік береді. (a,b) = (97,33) мәндері өте сапалы кездейсоқ сандарды алуға мүмкіндік береді және жоғары өлшемі бар кездейсоқ векторлармен істейтін алгоритмда пайдаланады.

Кешігуі бар Фибоначчи әдісіне негізделген ПКС генераторы криптографияда пайдаланған болатын. Одан басқа, олар математикалық және статистикалық есептерде қолданылады, және де кездейсоқ процестерді модельдеуде. Кешігуі бар Фибоначчи әдісіне негізделген ПКС генераторы танымал Matlab жүйеде пайдаланған.

7.5 BBS алгоритм негізіндегі кездейсоқ сандар генераторы

BBS алгоритмы (авторлар аттарынан — L.Blum, M.Blum, M.Shub) немесе квадратты қалдығы бар генераторы деп аталатын псевдокездейсоқ сандарды генерациялау алгоритмы кең тараған. Криптография мақсаты үшін бұл әдіс 1986 жылы ұсынылған болатын.

Осы әдісте, алдымен екі үлкен жай сандар p мен q таңдап алынады. p мен q сандары модулі 4 бойынша 3-пен салыстырмалы болу керек, яғни p мен q-ны 4-ке бөлгенде бірдей қалдығы 3 алыну керек. Онан әрі $M = p\ q$ саны есептеледі, оны бүтін Блюм саны деп атайды. Сосын M-мен өзара жай (яғни бірден басқа ортақ бөлгіштері жоқ) басқа кездейсоқ бүтін сан x таңдап алынады. Есептейміз $x_0 = x^2 \bmod M$. x_0 – генератордың бастапқы саны деп аталады.

Генератор жұмысының әрбір n қадамында есептеледі $x_{n+1} = x_n^2 \bmod M$. n-ші қадамның нәтижесі x_{n+1} санның бір (әдетте кіші) биты болып табылады. Кейде нәтиже ретінде жұптық битын қабылдайды, яғни элементтің екілік түріндегі бірліктер саны. Егер сан жазуында бірліктер саны жұп болса - жұптық битын 0-ге тең деп алады, тақ болса - жұптық битын 1-ге тең деп қабылдайды.

Мысалы, $p = 11$, $q = 19$ болсын (көз жеткендей, 11 mod 4 = 3, 19 mod 4 = 3). Онда $M = p.q = 11.19 = 209$. M-мен өзара жай x таңдап аламыз: $x = 3$ болсын. Генератордың бастапқы санын x_0 есептейік:

$$x_0 = x^2 \bmod M = 32 \bmod 209 = 9 \bmod 209 = 9.$$

BBS алгоритмы бойынша алғашқы он x_i санды есептейік. Кездейсоқ бит ретінде x_i санның екілік түріндегі жазуында кіші битты аламыз:

$x_1 = 9^2 \bmod 209 = 81 \bmod 209 = 81$ кіші бит: 1
$x_2 = 81^2 \bmod 209 = 6561 \bmod 209 = 82$ кіші бит: 0
$x_3 = 82^2 \bmod 209 = 6724 \bmod 209 = 36$ кіші бит: 0
$x_4 = 36^2 \bmod 209 = 1296 \bmod 209 = 42$ кіші бит: 0
$x_5 = 42^2 \bmod 209 = 1764 \bmod 209 = 92$ кіші бит: 0
$x_6 = 92^2 \bmod 209 = 8464 \bmod 209 = 104$ кіші бит: 0
$x_7 = 104^2 \bmod 209 = 10816 \bmod 209 = 157$ кіші бит: 1
$x_8 = 157^2 \bmod 209 = 24649 \bmod 209 = 196$ кіші бит: 0
$x_9 = 196^2 \bmod 209 = 38416 \bmod 209 = 169$ кіші бит: 1

$$x_{10} = 169^2 \bmod 209 = 28561 \bmod 209 = 137 \text{ кіші бит: } 1$$

Тәжірибелік мақсаты үшін осы әдістің ең қызық қасиеті - тізбектін *n*-ші саның алу үшін барлық x_i санның *n* бұрыңғыларын есептеу қажет емес. Өйткені мына формула бойынша x_n тура алуға болады:

$$x_n = x_0^{2^n \bmod [(p-1)(q-1)]} \bmod M$$

Мысалы, x_{10}-нан тура x_0-ды есептейік:

$$x_{10} = x_0^{2^{10} \bmod [(11-1)(19-1)]} \bmod 209 = x_0^{1024 \bmod 180} \bmod 209 =$$

$$= 9^{124} \bmod 209 = (9^4)^{31} \bmod 209 = 81^{31} \bmod 209 =$$

$$= (81^{15} \bmod 209)(81^{16} \bmod 209) = ((81^3)^5 \bmod 209)((81^4)^4 \bmod 209)$$

$$= (26^5 \bmod 209)(42^4 \bmod 209) = (144 \cdot 104) \bmod 209 = 14976 \bmod 209 = 137$$

Нәтижесінде, шынында да рет-ретімен есептегендей мән аламыз - 137. Есептеу күрделі болып көрінеді, бірақ оларды кішкентай процедура немесе программа түрінде орындап, қажет болғанда пайдалануға болады.

x_n-ды «тура» алу мүмкіндігі BBS алгоритмды ағынды шифрлауда пайдалануға мүмкіндік береді, мысалы, еркін қатынауы бар файлдар үшін немесе деректер қорына жазулары бар файл фрагменттері үшін.

BBS алгоритмның қауіпсіздігі үлкен *M* санды көбейткіштерге жіктеу күрделілігіне негізделген. Егер *M* жеткілікті үлкен болса, оны құпиялы түрде сақтаудың қажеті де жоқ; *M*-ды көбейткіштерге жіктемей ПКС генератордың шығуын ешкім айталмайды. Себебі, $n = pq$ (*p* мен *q* — жай сандар) түрлі сандарды көбейткіштерге жіктеу өте қиын, егер біз тек *n*-ды ғана білсек, ал *p* мен *q* — бірнеше ондаған немесе жүздеген биттен тұратын үлкен сандар болса (бұл факторизациялау деп аталатын есеп).

Одан басқа, BBS генераторы генерацияланған кейбір тізбекті біле отырып, қаскүнем не бұрыңғы не келесі биттерді анықтай алмайды. BBS генераторды сол жақ және оң жақ бағытта алдын ала болжауға болмайды. Бұл қасиеті криптография мақсатына өте пайдалы.

Алгоритмның ең маңызды кемшілігі – аса жылдамды еместігі, сондықтан оны нақты уақыт есептеуде және де, өкінішке орай, ағынды шифрлауда пайдалануға болмайды.

Дегенмен, бұл алгоритм үлкен периоды бар псевдокездейсоқ сандардың шынында жақсы тізбегін бергендіктен, оны шифрлауда кілттерді генерациялау үшін пайдалану жөн.

Негізгі ұғымдар

Stream cipher – ағынды шифр.

BBS алгоритмы – псевдокездейсоқ сандарды генерациялау әдістерінің біреуі. Алгоритм аты авторлар есімдерінен жиналған - L.Blum, M.Blum, M.Shub. Алгоритм криптографияда пайдалану мүмкін. BBS алгоритмы бойынша келесі x_{n+1} санды есептеу үшін пайдаланатын формула: $x_{n+1} = x_n^2 \bmod M$, мұнда $M = pq$ екі үлкен *p* мен *q* жай сандардың көбейтіндісі.

Псевдокездейсоқ сандар генераторы (ПКСГ) – сырттан кездейсоққа ұқсайтын биттер тізбегін жасайтын кейбір алгоритм немесе құрылғы.

Псевдокездейсоқ сандардың **сызықты конгруэнтті генераторы** - ең қарапайым генератордың біреуі, келесі k_i есептеу үшін $k_i = (a.k_{i-1}+b) \bmod c$ формуланы пайдаланады, мұнда *a*, *b*, *c* — кейбір константалар, ал k_{i-1} — алдыңғы псевдокездейсоқ сан.

Кешігуі бар Фибоначчи әдісі - псевдокездейсоқ сандарды генерациялау әдістерінің біреуі. Криптографияда пайдалану мүмкін.

Ағынды шифр - кіру хабардың шифрлауын бір-бірден биттер (немесе байттар) бойынша операцияда орындайтын шифр. Ағынды шифрлау алгоритмы хабарды бүтін сандар блоктарға бөлуді қажет етпейді. Ағынды шифрлар нақты уақытта деректерді шифрлау үшін пайдаланады.

Сұрақтар

1. Блокты шифрдан ағынды шифрдың айырмашылығы неде?
2. Айнымалы ұзындығы бар деректер ағынның шифрлауы қалай орындалады?
3. Қандай сандар «псевдокездейсоқ» деп аталады?
4. Криптографиялық мақсатта пайдалану үшін псевдокездейсоқ сандар генераторында қандай қасиеттері болу керек?
5. Қандай псевдокездейсоқ сандар генераторын Сіз айтып бере аласыз?
6. Псевдокездейсоқ сандар генераторлардың негізгі сипаттамаларын, артықшылықтарын және кемшіліктерін айтып беріңіз.