Лекция 11 АШЫҚ КІЛТІ БАР КРИПТОГРАФИЯЛЫҚ АЛГОРИТМДЕР ЖӘНЕ ОЛАРДЫ ПАЙДАЛАНУ

1 RSA алгоритмы

Негізгі мәліметтер

Ашық кілті бар шифрлау алгоритмы **RSA** XX ғасырдың 70-ші жылдары ұсынылған болатын. Оның аты авторлар фамилиясының бірінші әріптерінен жиңалған: Р.Ривест (R.Rivest), А.Шамир (A.Shamir) және Л.Адлеман (L.Adleman). RSA алгоритмы криптографиялық жүйелерде ең кең тараған және жиі қолданылытын ассиметриялық алгоритмы болып табылады.

Алгоритмның негізінде мына факт жатыр: үлкен санды жай көбейткіштерге жіктеу өте күрделі есеп. RSA криптографиялық жүйе сандар теориясындаға келесі екі фактіге негізделеді:

1. сан жай ма деп тексеру есебі аса қиын емес;
2. $n = pq$ (p мен q — жай сандар) түрлі сандарды көбейткіштерге жіктеу есебі өте қиын, егер біз тек n-ды ғана білетін болсак, ал p мен q — ұлкен сандар болса (мұны факторизациялау есебі деп атайды).

RSA алгоритмы шифрлаудың блокты алгоритмы, мұнда шифрланған және шифрланбаған деректер 0 мен n - 1 аралығында кейбір n үшін бүтін сандар түрде берілу керек.

Шифрлау

Сөйтіп алгоритмды қарастырайық. Абонент A шифрланған хабарды абонент $Б$-ға бергісі келеді. Бұл жағдайда абонент $Б$ екі кілт дайындау керек (ашық кілт; жабық кілт) және өзінің ашық кілтің пайдаланушы A-ға жібереді.

Бірінші кезең ашық пен жабық кілтті жасау (генерациялау). Ол үшін алдымен екі үлкен жай сан таңдап алынады P және Q. Сосын көбейтіндісі есептеледі N:

$$N = PQ.$$

Осыдан кейін көмекші сан анықталады f:

$$f = (P - 1)(Q - 1).$$

Сонан соң кездейсоқ таңдап алынады $d < f$ саны және ол f–пен өзара жай болу керек.

Онан әрі e санды табу керек, ол үшін

$$ed \bmod f = 1.$$

Сандар d және N пайдаланушының ашық кілті болып табылады, ал e мәні – жабық кілт.

Сонымен, бұл кезеңде пайдаланушының қолында кестеде көрсетілген ақпарат болу керек:

	Ашық кілт	Жабық кілт
Жүйе пайдаланушысы	N, d	e

Пайдаланушы $Б$ пайдаланушы A-дан шифрланған хабар алғысы келгендіктен, онда пайдаланушы $Б$ өзінің ашық кілтің (d, N) пайдаланушы A-ға жіберу керек. P мен Q сандардың енді қажеті жоқ, бірақ оларды ешкімге айтпаған жөн; ең жақсысы оларды жалпы ұмыту.

Осымен кілттерді дайындау кезеңі аяқталады және деректерді шифрлау үшін RSA негізгі протоколын пайдалануға болады.

Екінші кезең – деректерді шифрлау. Егер абонент *А* кейбір деректерді *Б* абонентқа жібергісі келсе, ол өзінің хабарын цифрлық түрге келтіріп және оны m_1, m_2, m_3,
... блоктарға бөледі, мұнда $m_i < N$. Шифрланған хабар c_i блоктан тұрады.

Абонент *А* өз хабарының әрбір блогын, пайдаланушы *Б*-ның ашық параметрлерін пайдаланып,

$$c_i = m_i^d \bmod N$$

формула бойынша шифрлайды, және шифрланған хабарды $C=(c_1, c_2, c_3, ...)$ ашық арна арқылы жібереді.

Шифрланған хабарды алған абонент *Б* хабардың барлық блоктарын

$$m_i = C^e \bmod N$$

формула бойынша ашып оқиды.

Барлық ашып оқылған блоктар тура *А* пайдаланушыдан шыққандай болады.

Барлық хабарларды ұстап алған және ашық ақпаратты білетін қаскүнем, *P* мен *Q* үлкен мәндері болғанда, бастапқы хабарды табалмайды.

11.2 Алгоритм бойынша есептеу алгоритмы

Пайдаланушы *А* хабарды пайдаланушы *Б*-ға бергісі келеді. Бұл жағдайда алдымен пайдаланушы *Б* ашық және жабық кілттерді дайындау керек. Мысалы, ол келесі параметрлерді таңдап алсын:

$$P = 3,\ Q = 11,\ N = 3*11 = 33.$$

Онда

$$f = (P - 1)(Q - 1) = (3-1)(11-1) = 20.$$

Сосын пайдаланушы *Б f*-пен ортақ бөлгіші жоқ кез келген *d* санды таңдайды (бұл сосын шифрланған хабарды бір мәнді қалпына келтіру үшін қажет болады). $d = 13$ болсын. Бұл сан ашық кілттің құрауышыларынын бірі болады.

Онан әрі *е* санды табу керек, оны хабарды ашып оқу үшін жабық кілт ретінде пайдалануға болады. *е* мәні

$$ed \bmod f = 1$$

ара қатынасқа қанағаттандырылу керек.

f–ң кіші мәндері үшін *е*–ны таңдау әдісі арқылы табуға болады. Жалпы жағдайда *е*– ны іздеу үшін жалпыланған Евклид алгоритмын пайдалануға болады. Біздің жағдайда келеді $e=17$ (тексереміз: 13*17 mod 20 = 221 mod 20 = 1).

Енді пайдаланушы *Б* өз жабық кілтің 17 есте қалдыру керек, ашық кілтті (13, 33) *А* пайдаланушыға жіберу керек және $P=3$ мен $Q=11$ сандарды жою керек.

Ашық кілтті (13, 33) алған пайдаланушы *А*, $N=33$ тең болғанын көріп бастапқы хабарды үш блокқа бөледі, әрбіреунің мәні *N*–нан кем. Мысалы, үш блок болсын $m_1=8$, $m_2=27$, $m_3=5$. Сосын пайдаланушы *А* әрбір блокты шифрлайды:

$$c_1 = 8^{13} \bmod 33 = 17$$
$$c_2 = 27^{13} \bmod 33 = 15$$
$$c_3 = 5^{13} \bmod 33 = 26.$$

Үш блоктан (17, 15, 26) тұратын шифрланған хабар *Б* пайдаланушыға жіберіледі. Ол өз жабық кілтін $e=17$ және $N=33$ пайдаланып, хабарды ашып оқиды:

$$m_1 = 17^{17} \bmod 33 = 8$$
$$m_2 = 15^{17} \bmod 33 = 27$$
$$m_3 = 26^{17} \bmod 33 = 5.$$

Сонымен, абонент *А*-дан алынған хабарды абонент *Б* ашып оқыды.

11.3 RSA алгоритмның тәжірибелік пайдалануының сұрақтары

RSA алгоритмға негізделген ашық шифрлау кең тараған шифрлау PGP пакетте, Windows операциялық жүйеде, әртүрлі Интернет-браузерлерде, банктік компьютерлік жүйелерде қолданылады. Одан басқа, әртүрлі ашық кілті бар шифрлаудың және цифрлық қолды жасаудың халықаралық стандарттары RSA-ны негізгі алгоритм ретінде пайдаланады.

Шифрлаудың жоғары сенімділігін қамтамасыз ету үшін мынау қажет: модуль ретінде шығатын *N* саны өте үлкен болу керек – бірнеше жүздеген немесе мыңдаған бит. Тек бұл жағдайда ғана, ашық параметрі бойынша, жабық кілтті анықтау мүмкін емес деп күтуге болады. Мысалы, 1995 жылы 500-таңбалы модулі үшін RSA шифрын ашып алды. Ол үшін Интернет желі көмегімен мыңнан астам компьютер бірлесіп жұмыс істеді.

RSA-ның авторлары *N* модульдің келесі параметрлерін пайдалануға ұсыныс берді (2004 жылы): 768 бит – жеке меншік адамдар үшін; 1024 бит – коммерциялық ақпарат үшін; 2048 бит - өте құпиялы ақпарат үшін. Олардың ұсыныстарынан біраз уақыт өтті, сондықтан, қазіргі пайдаланушылар кілт мөлшерін өсіруіне назар аудару керек. Бірақ, кілт мөлшері неғұрлым үлкен болса, соғұрлым жүйенің жұмысы баяу болады. Сондықтан, кілт мөлшерін қажетсіз өсіре беру мағынасыз.

Кілт мөлшерімен RSA-ның тағы бір жүзеге асыру аспектісі байланысты – есептеуіш. Алгоритмды пайдалануда есептеулер қажетті кілттерді жасау кезінде де және шифрлау/дешифрлау кезінде де. Мұнда кілт мөлшері неғұрлым үлкен болса, соғұрлым есептерді орындау қиын болады. Аса үлкен сандармен жұмыс істеу үшін ұзын арифметика аппаратын пайдалану керек. Көп жүздеген биттен тұратын сандар көпшілік микропроцессорлардың регистрларына кіре алмайды, сондықтан оларды бөліктеп өңдеу керек. Бұл кезде шифрлау мен дешифрлау да үлкен бүтін санды *N* модулі бойынша бүтін дәрежеге көтеруді қосады. Тура есептерде аралық мәндер тым айтқысыз болар еді. Есептеу процесті оңайлату үшін, үлкен сандармен жұмыс істеуге арнайы алгоритмдер пайдаланады, олар модульды арифметиканың қасиеттеріне және дәрежеге көтеруді оптимизациялауға негізделген.

RSA алгоритмы бағдарламалық та аппараттық та түрде жүзеге асырылады. Көптеген әлемдік фирмалар шифрлауда RSA алгоритмын орындайтын мамандырылған микросхемалар шығарады. Бағдарламалық жүзеге асырулар аппараттықтан бірталай баяу. RSA-ның бағдарламалық шифрлау артықшылықтарына жатады: параметрлерді икемділік күйге келтіру мүмкіндігі, программалық пакеттерге интеграциялау мүмкіндігі. Толығымен айтқанда, RSA-ның бағдарламалық та аппараттық та жүзеге асыруы, симметриялық алгоритммен салыстырғанда (мысалы ГОСТ 28147-89), орындауға шамамен мың есе көп уақыт талап етеді.

RSA алгоритмы электронды цифрлық қол қоюды құрастыруға және де кілттермен алмасу үшін пайдалану мүмкін. Электронды қол қоюды құрастыру мүмкіндігінің себебі – бұл жүйеде құпиялы да ашық та кілттер тең құқылы. Кілттердің әрбіреуі *d* немесе *e* шифрлау үшін де дешифрлау үшін де пайдалану мүмкін. Бұл қасиет ашық кілті бар криптожүйелердің бәрінде орындалмайды.

11.4 Диффи-Хеллман алгоритмы

Негізгі мәліметтер

Осы алгоритмның алғашқы жариялауы XX ғасырдың 70-ші жылдары Диффи және Хеллман мақаласында шықты. Диффи-Хеллман алгоритмы хабарларды шифрлау немесе электронды қол қоюды құрастыру үшін пайдаланбайды. Оның міндеті – *кілттерді үлестіру*. Ол екі не одан көп пайдаланушыларға делдалсыз кілттермен

алмасуға мүмкіндік береді, сосын бұл кілт симметриялық шифрлау үшін қолданылады. Бұл қорғалған арна арқылы жіберілетін құпиялы кілтсіз ақпаратты қорғауға мүмкіндік беретін алғашқы криптожүйе болатын. Диффи мен Хеллман ұсынған кілттерді үлестірудің ашық схемасы криптографияда кәдімгідей төңкеріс жасады, себебі классикалық криптографияның негізгі проблемасын шешті - кілттерді үлестіру проблемасын.

Алгоритм дискретты логарифмды есептеу операциясының күрделігіне негізделген. Бұл алгоритмде есептеулер кейбір үлкен жай санның P модулі бойынша жүргізіледі. Алдымен арнайы түрде P–дан кіші кейбір натурал сан A таңдап алынады. Егер біз X мәнді шифрлайтын болсақ, онда есептейміз

$$Y = A^X \bmod P.$$

X-ты біле отырып Y–ты есептеу оңай. Y–тан X-ты есептеудің кері есебі жеткілікті күрделі болады. X экспонентасы Y–тын дискретты логарифмы деп аталады. Сонымен, дискретты логарифмның есептеу күрделігін біліп, Y санды ашық түрде қандай да байланыс арна арқылы жіберуге болады, өйткені P–нын үлкен модулінде бастапқы X мәнін табу мүмкін емес деп айтуға болады. Кілтті құрастыру үшін Диффи-Хеллман алгоритмы осы математикалық фактісіне негізделген.

Жалпы кілтті құрастыру

Екі пайдаланушы, оларды шартты түрде пайдаланушы 1 және пайдаланушы 2 деп атайық, симметриялық шифрлау алгоритмы үшін жалпы кілт құрастырғысы келеді. Басында олар үлкен жай сан P және кейбір арнайы сан A ($1<A<P$-1) таңдап алу керек. Мұнда [1, 2, ..., P-l] интервалындаға барлық сандар A mod P –ң түрлі дәрежелері ретінде көрсетілу мүмкін болсын. Осы сандар жүйенің барлық абоненттеріне белгілі болу керек және ашық түрде таңдалынады. Бұл жалпы параметрлер деп аталсын.

Сосын 1-ші пайдаланушы X_1 ($X_1<P$) сан таңдайды, оны кездейсоқ сан датчикі көмегімен генерациялау жөн болады. Бұл 1-ші пайдаланушының жабық кілті болады, оны құпиялы түрде сақтау қажет. Жабық кілттің негізінде пайдаланушы 1

$$Y_1 = A^{X_1} \bmod P$$

санды есептейді, оны ол екінші абонентке жібереді.

Екінші пайдаланушы да осындай түрімен X_2 генерациялайды және есептейді

$$Y_2 \neq^{X_2} \bmod P$$

Бұл мәнді пайдаланушы 2 бірінші пайдаланушыға жібереді.

Осыдан кейін кестеде келтірілген ақпарат пайдаланушылардың қолында болу керек:

	Ортақ параметрлер	Ашық кілт	Жабық кілт
Пайдаланушы 1	P, A	Y_1	X_1
Пайдаланушы 2		Y_2	X_2

Y_1 и Y_2 сандардан және өз жабық кілттерінен абоненттердің әрбіреуі симметриялық шифрлау сеансы үшін ортақ құпиялы кілт Z құрастыра алады. Бірінші пайдаланушы мұны былай істеу керек:

$$Z = (Y_2)^{X_1} \bmod P$$

Одан басқа мұны ешкім істей алмайды, өйткені X_1 саны құпиялы. Екінші пайдаланушы өз жабық кілтін және өз абонентының ашық кілтін пайдаланып, дәл сол санды Z алу мүмкін:

$$Z = (Y_1)^{X_2} \bmod P$$

Егер ортақ құпиялы кілтті құрастыру протоколы дұрыс орындалса, абоненттердің екеуінде де Z мәндері бірдей шығу керек. Айта кетейік, қарсылас

құпиялы сандарды X_1 және X_2 білмей Z санды есептей алмайды. X_1 мен X_2 білмей қарсылас тек ашық түрде берілетің P, A, Y_1 және Y_2 пайдаланып Z-ты есептеуге тырысу мүмкін. Ортақ кілтті құрастыру қауіпсіздігі Диффи-Хеллман алгоритмда мұндай: жай санның модулі бойынша экспонентаны есептеуге салыстырмалы оңай болса да, дискретты логарифмды есептеу өте қиын. Үлкен жүздеген және мындаған биты бар жай сандар үшін бұл есеп шешілмейтін деп саналады, себебі орасан зор есептеу ресурстарды жұмсайды.

Пайдаланушылар 1 және 2 деректерді шифрлау мен дешифрлау үшін Z мәнің құпиялы кілт ретінде пайдалану мүмкін. Осы жолмен кез келген қос абоненттер өздеріне ғана белгілі құпиялы кілтті есептей алады.

11.5 Алгоритм бойынша есептеу мысалы

Интернет арқылы шифрланған хабарлармен алмасқысы келетін екі абонент, кезекті сеанс үшін құпиялы кілтті құрастырғысы келді. Оларда келесі ортақ параметрлер болсын:

$$P = 11, A = 7.$$

Абоненттер құпиялы сан X таңдайды және оған сәйкес ашық сан Y есептейді.

Таңдалған сандар:

$$X_1 = 3, X_2 = 9.$$

Есептейміз

$$Y_1 = 7^3 \bmod 11 = 2,$$
$$Y_2 = 7^9 \bmod 11 = 8.$$

Сосын пайдаланушылар ашық кілттерін Y_1 мен Y_2 айырбастайды. Осыдан кейін пайдаланушылардың әрбіреуі ортақ құпиялы кілт есептей алады:

$$\text{пайдаланушы 1: } Z = 8^3 \bmod 11 = 6.$$
$$\text{пайдаланушы 2: } Z = 2^9 \bmod 11 = 6.$$

Енді оларда ортақ кілті 6 бар, ол байланыс арна арқылы жіберілмейді.

11.6 Диффи-Хеллман алгоритмның тәжірибелік пайдалануының сұрақтары

Диффи-Хеллман алгоритмы дұрыс істеу үшін, яғни протоколға қатысатын екі пайдаланушы бірдей Z сан алу үшін, есептеуде пайдаланатын A санды дұрыс таңдау керек.

А санында келесі қасиеттер болу керек:

барлық

$$A \bmod P, A^2 \bmod P, A^3 \bmod P, \ldots, A^{P-1} \bmod P$$

сияқты сандар әртүрлі болу керек және 1 ден (P-1)-ге дейін диапазонда бүтін оң мәндерінен тұру керек. Тек осындай жағдайда түрлі бүтін сан $Y<P$ және A мәні үшін бір ғана X экспонентаны табуға болады

$$Y = A^X \bmod P, \qquad \text{мұнда } 0 \leq X \leq (P\text{-}1)$$

Еркін берілген P-да A параметрді таңдау есебі қиын болу мүмкін, себебі ол P-1 санды жай көбейткіштерге жіктеуімен байланысты.

Тәжірибеде келесі жолды қолдануға болады. Жай P санды таңданғанда $P = 2q + 1$ теңдік орындалу керек, мұндағы q – жай сан. Онда A ретінде түрлі санды алуға болады, ол үшін

$$1 < A < P\text{-}1 \qquad \text{және} \qquad A^q \bmod P\text{-}1$$

теңсіздіктер орын алу керек.

Келісімді A мен P параметрлерді іріктеп алу үшін біраз уақыт қажет, бірақ бұл байланыс жүйенің жұмысын тежелмейді. Бұл параметрлер толық пайдаланушылар тобына ортақ болып табылады. Олар әдетте Диффи-Хеллман протоколын қолданалатын пайдаланушылар тобын жасағанда бір рет таңдап алынады, және жұмыс барысында өзгермейді. Ал жабық кілттер мәнің жиі өзгерте отырып, кездейсоқ тәріздес сандар генераторы көмегімен таңдау керек.

Айта кетейік, бұл алгоритм барлық ассиметриялық алгоритмдер секілді, «man-in-the-middle» («адам ортада») деп аталатын шабуылды шыдамайды. Егер қарсылас қолына түскен хабардан басқа оларды ауыстырай алса, онда ол қатысушылардың ашық кілттерін ұстап алып, өзінің қос ашық және жабық кілттерін жасап, және қатысушылардын әрбіреуіне өзінің ашық кілтін жіберу мүмкін. Осыдан кейін әрбір қатысушы кілтті есептейді, ол кілт басқа қатысушымен емес, қарсыласпен ортақ болады.

11.7 Эль-Гамаль алгоритмы

Негізгі мәліметтер

1985 жылы Эль-Гамаль (T.ElGamal) ұсынған ассиметриялық алгоритм әмбебап. Ол барлық үш негізгі есептерді шешуге пайдалану мүмкін: деректерді шифрлау үшін, цифрлық қолды құрастыру үшін және ортақ кілтті келістіру үшін. Одан басқа, парольді тексеру үшін, хабардың пара-парлығын дәлелдеу үшін алгоритмды түрлендіруге болады. Бұл алгоритмның қауіпсіздігі, Диффи-Хеллман алгоритмындай, дискретты логарифмның есептеу күрделігіне негізделген. Бұл алгоритм абоненттердің ортақ құпиялы кілтін құрастыру үшін Диффи-Хеллман схемасын пайдаланады және сосын хабарды осы кілтке көбейтіп ол шифрланады.

Шифрлау кезінде де, цифрлық қолды құрастыру кезінде де әрбір пайдаланушыға қос кілттер жасау қажет. Ол үшін, Диффи-Хеллман схемасындай, кейбір үлкен жай сан P және сан A таңдап алынады (A-ның әртүрлі дәрежелері P модулі бойынша әртүрлі сандар болып табылады). P мен A сандар ашық түрде беріліп, желінің барлық абоненттеріне ортақ болу мүмкін.

Сосын топтың әрбір абоненты өзінің құпиялы санын X_i, $1<X_i<P-1$ таңдайды, және
оған сәйкес ашық санды Y_i есептейді: $Y_i = A^{X_i} \bmod P$. Сонымен, әрбір пайдаланушы жабық X_i және ашық Y_i кілт жасай алады.

Жүйенің қажетті параметрлері кестеде келтірілген:

	Ортақ параметрлер	Ашық кілт	Жабық кілт
Пайдаланушы 1	P, A	Y_1	X_1
…		…	…
Пайдаланушы i		Y_i	X_i

Шифрлау

Енді қай түрде деректер шифрлайтының қарастырып шығайық. Шифрлауға арналған хабар, бір сан немесе P-дан кіші сандар жиынтығы түрінде берілу керек. Пайдаланушы 1 пайдаланушы 2-ге m хабарды бергісі келсін. Онда іс-әрекет тізбегі мұндай болу керек:

1. Бірінші пайдаланушы P-1 мен өзара жай кездейсоқ k санды таңдайды және есептейді сандарды:

$$r = A^k \bmod P, \qquad e = m \cdot Y_2^k \bmod P$$

мұндағы Y_2 – пайдаланушы 2-нің ашық кілті. *k* саны құпиялы түрде сақталынады.

2. Қос сандар (*r*, *e*) шифромәтін деп аталады және екінші пайдаланушыға жіберіледі.

3. Екінші пайдаланушы (*r*, *e*)-ны алып, хабарды дешифрлау үшін есептейді:

$$m = e \cdot r^{P-1-X_2} \bmod P$$

мұндағы X_2 – пайдаланушы 2-нің жабық кілті. Нәтижесінде ол бастапқы *m* хабарды алады.

Егер *P*, *A*, Y_2, *r*, *e* қаскүнемнің қолына түссе немесі біліп қойса, онда оларға сүйеніп ол *m* хабарды ашып оқи алмайды. Өйткені *m* хабарды шифрлау үшін бірінші пайдаланушы таңдап алған *k* параметрді қарсылас білмейді. Қандай да болса әдіспен *k*-ны есептеу мүмкін емес, себебі бұл дискретты логарифмдеу есебі. Демек, қаскүнем *m* мәнін де есептей алмайды, өйткені *m* оған белгісіз санға көбейтілген болатын. Қарсылас хабарды алушының (екінші абонент) іс-әрекеттерін де қайталай алмайды, себебі ол жабық кілтті X_2 білмейді (Y_2-ге негізделіп X_2-ны есептеу — ол да дискретты логарифмдеу есебі).

Ұқсас алгоритмы бойынша деректердің үлкен көлемін симметриялық шифрлау үшін пайдаланатын кілттің келісуін де жүргізуге болады. Одан әрі, тәжірибеде Эль-Гамаль алгоритмын үлкен хабарды тікелей шифрлау үшін емес, сессияның ортақ кілтін келістіру үшін пайдалану жөн. Бұл үлкен модулі бойынша дәрежеге көтеру және көбейту операцияларға байланысты. RSA мен Диффи-Хеллман алгоритмында сияқты, операциялар үлкен жүздеген немесе мындаған биты бар сандармен жүргізіледі. Сондықтан үлкен хабарлардың шифрлауы өте баяу жасалынады.

Шифрлаудың мысалы

Интернет арқылы шифрланған хабарлармен алмасқысы келетін екі абонентте келесі ортак параметрлер болсын:

$$P = 11, A = 7.$$

Одан басқа, 1 мен 2 пайдаланушыларда қос жабық және ашық кілттері бар болсын:

Пайдаланушы 1: жабық кілт $X_1 = 3$, ашық кілт $Y_1 = 7^3 \bmod 11 = 2$,

Пайдаланушы 2: жабық кілт $X_2 = 9$, ашық кілт $Y_2 = 7^9 \bmod 11 = 8$.

Бірінші абонент екіншіге хабар бергісі келеді. Ол үшін бірінші абонент кілттерді үлестіру орталығынан екінші абоненттың ашық кілтін $Y_2 = 8$ сұрайды. Енді ол өз хабарын шифрлай алады, бұл хабар сандық түрде мұндай болсын $m = 9$.

Бірінші абонент кездейсоқ түрде *k* санды таңдайды, мысалы $k = 7$. *k* саны *P*-1 мен өзара жай болу керек. $k = 7$ мәнінде *P*-1=10 мәнімен ортақ бөлгіштері жоқ, демек, ол бізге келеді. Бірінші абонент өз хабарын шифрлайды мына формулалар бойынша:

$$r = A^k \bmod P = 7^7 \bmod 11 = 6$$

$$e = m\, Y_2^k \bmod P = 9\, 8^7 \bmod 11 = 7$$

Қос сандар (6, 7) шифромәтін болып табылады және екінші пайдаланушыға жіберіледі. Екінші пайдаланушы (6, 7) алып және өз жабық кілтін $X_2 = 9$ пайдаланып, хабарды ашып оқу үшін есептейді

$$m = e \cdot r^{P-1-X_2} \bmod P = 7 \cdot 6^{11-1-9} \bmod 11 \quad 7 \cdot 6^1 \bmod 11$$

9 .

Нәтижесінде ол шынында да бастапқы хабарды *m* алады.

11.8 Эллиптикалық қисықтарға негізделген криптографиялық жүйелер

1985 жылы американ ғалымдары Н.Коблиц (Neal Koblitz) және В.Миллер (Victor Miller) ашық кілті бар криптожүйелер үшін эллиптикалық қисықтар теориясын пайдалануға ұсынды. Ал 1998 жылдан криптографиялық есептерді шешу үшін

эллиптикалық қисықтардың пайдалануы АҚШ стандартына енгізілді ANSI X9.62 және FIPS 186-2. 2001 жылы ұқсас стандарт Ресейде де қабылданды ГОСТ Р34.10-2001.

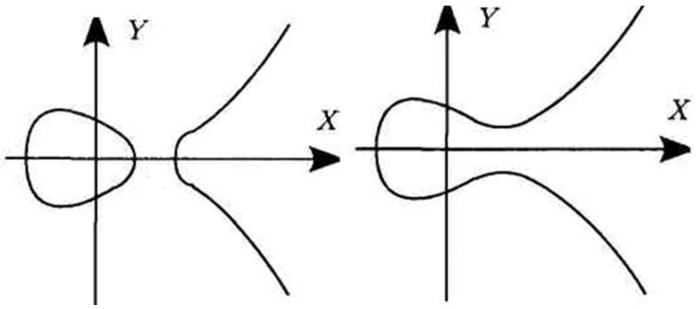
Эллиптикалық қисықтарға негізделген криптожүйелердің негізгі артықшылығы, басқа ассиметриялық криптожүелерге қарағанда, өндеуге және есептеуге жұмсалған бірдей шығындарда, оның жоғары криптоберіктігі. Себебі, эллиптикалық қисықтарда кері функциялардың есептеуі, дискретты логарифмды есептеуге (Диффи-Хеллман және Эль-Гамаль алгоритмдары) немесе факторизациялау есебін шешуге (RSA алгоритмы) қарағанда өте күрделі. Нәтижесінде, бірдей беріктік деңгейге жету үшін, мысалы RSA алгоритмда 1024 битты модулі қажет, ал эллиптикалық қисықтарға негізделген жүйелерде модуль мөлшері 160 бит ғана болады.

Эллиптикалық қисықтарды пайдалануымен криптографиялық жүйелердің құруының негізгі принциптерін тұжырымдайық.

Криптографияда мына теңдеумен анықталатын жазықтықтағы эллиптикалық қисықтар қолданылады:

$$Y^2 = X^3 + aX + b \bmod p,$$

мұндағы p – кейбір үлкен жай сан; a және b – константалар. a мен b параметрлердің әртүрлі мәндеріндегі эллиптикалық қисықтын графигі 11.1 суретте көрсетілген.



Сурет 11.1. Эллиптикалық қисықтар графиктерінің вариантты

Эллиптикалық қисықтардың пайдалану принципі мынадай. Пайдаланушылар тобы үшін ортақ эллиптикалық қисық E және оның үстінде кейбір нүкте G таңдап алынады. Пайдаланушының жабық кілті ретінде кейбір бүтін сан c болып табылады, ал ашық кілті –

Е қисықтағы нүкте D, ол c санды пайдаланып композицияның арнайы түрлендіру нәтижесінде алынады. Қисықтың параметрі мен абоненттердің ашық кілттер тізімі әдеттегідей желінің барлық пайдаланушыларына беріледі. Пайдаланушылардың ашық және жабық кілттері, алгоритм міндетіне байланысты, шифрлау мен дешифрлау операцияларды орындау үшін пайдаланылады.

Эллиптикалық қисықтар көмегімен танымал ашық кілті бар протоколдардың көбі жүзеге асырылу мүмкін. Дискретты логарифмдеуге негізделген кез келген криптожүйені эллиптикалық қисықтарға түсіру оңай. Мысалы, $y = g_x \bmod p$ түрлі математикалық операцияны эллиптикалық қисықтар математикалық аппаратының операцияларымен кілтті құрастыру Диффи-Хеллман алгоритмында немесе цифрлық қолды есептеу Эль-Гамаль алгоритмында ауыстыруға болады. Нәтижесінде сол баяғы алгоритмдер алынады, бірақ басқа математикалық операциялармен.

Эллиптикалық қисықтардың математикалық аппараттын күрделігіне қарамастан, қажетті есептерді жеткілікті тез жүзеге асыратын тиімді есептеуіш әдістер бар.

11.9 Ассиметриялық шифрлау алгоритмды пайдалануда мүмкін болатын шабуылдар

Шабуыл «адам ортада»

Ашық кілті бар қарапайым шифрлау протоколды еске түсірейік. Егер пайдаланушы

А пайдаланушы *Б*-ға құпиялы хабар берейін десе, ол пайдаланушы *Б*-дан ашық кілт $U_Б$ алу керек және осы ашық кілтпен өз хабарын шифрлайды. Шифрланған хабар кез келген байланыс арна арқылы жіберілу мүмкін, мысалы, электронды поштамен. Пайдаланушы *А*-дан хабарды алғаннан кейін пайдаланушы *Б* оны өзінің жабық кілтімен $R_Б$ ашып оқиды. Шифрланған хабарлармен алмасудың осындай процедурасы қолға түскен ашық кілттер және шифрланған хабарлар бойынша қарсыласқа бастапқы хабарды ашуға мүмкіндік бермейді. Бұл бір жақты функцияның қасиеттерімен қамтамасыз етіледі.

Бірақ осындай схема «man-in-the-middle» (**«адам ортада»**) типті шабуылға берікті болалмайды. Қаскүнем хабарларды ғана ұстап алудан басқа, оларды ауыстара алады, яғни белсенді шабуыл жасай алады делік. Бұл қазіргі беру желілерде әбден мүмкін, мысалы Интернетте, онда ақпарат бір пайдаланушыдан басқаға көптік аралық тораптар арқылы беріледі. Қаскүнем ретінде, мысалы желінің жүйелік әкімшісі болу мүмкін. Осындай бұзушы пайдаланушылардың хабарларын ұстап алудан басқа, оларды жою немесе өзінікімен ауыстыру мүмкін. Ол өзін байланыс қатысушылардын орнына қою мүмкін. «Man-in-the-middle» шабуыл былай өткізілу мүмкін:

1. Пайдаланушы *Б* пайдаланушы *А*-ға өз ашық кілтін $U_Б$ жібереді. Қарсылас бұл кілтті ұстап алып, оны сақтайды және өзінің ашық кілтімен $U_К$ ауыстырады.

2. Пайдаланушы *А* өз хабарын *М* алынған ашық кілтпен $U_К$ шифрлайды (ол абонент *Б*-ның ашық кілтін пайдаланып отырмын деп ойлайды) және шифрланған хабарды пайдаланушы *Б*-ға жібереді.

3. Қаскүнем бұл хабарды ұстап алады, өзінің жабық кілтімен $R_К$ дешифрлайды, оқыйды немесе ауыстырады, сосын пайдаланушы *Б*-ның ашық кілтімен шифрлайды және пайдаланушы *Б*-ға жібереді.

Осыған ұқсас қаскүнем пайдаланушы *Б*-ның жауаптарын оқу үшін пайдаланушы *А*-ның да ашық кілтін ұстап алады. Нәтижесінде бұзушы абоненттердің барлық корреспонденциясын оқып (өзгерте) алады. Пайдаланушылар *А* және *Б* оны сезбеу де мүмкін.

Тәжірибеде «man-in-the-middle» шабуылға қарсы бірнеше тәсілдер құрастырған. Олардың біреуі мынадай: әрбір шифрланған хабарды бір-бірімен байланысты екі бөлшекке бөлу. Хабардың бөлшектері рет-ретімен жіберіледі және жеке ашып оқылмайды. Бұл хабарлармен алмасу протоколы мына түрде болу мүмкін:

1. Пайдаланушылар *А* мен *Б* ашық кілттерін айырбастайды.
2. Пайдаланушы *А* өз хабарын пайдаланушы *Б*-ның ашық кілтімен шифрлайды және жартысын пайдаланушы *Б*-ға жібереді.
3. Пайдаланушы *Б* өз хабарын пайдаланушы *А*-ның ашық кілтімен және жартысын пайдаланушы *А*-ға жібереді.
4. Пайдаланушы *А* шифрланған хабардын екінші жартысын пайдаланушы *Б*-ға

жібереді.

5. Пайдаланушы *Б* алынған хабардын жартыларын қосады және өзінің жабық кілтімен дешифрлайды. Сосын өзінің шифрланған хабарының екінші жартысын пайдаланушы *А*-ға жібереді.
6. Пайдаланушы *А* пайдаланушы *Б*-дан алынған хабардын жартыларын қосады және өзінің жабық кілтімен дешифрлайды.

Протоколды жүзеге асыру үшін хабарды екі бөлшекке бөлу процесі әртүрлі тәсілмен жүргізілу мүмкін, мысалы, әрбір тақ байты бірінші хабарға енгізіледі, ал әрбір жұп байты – екінші бөлшекке.

«Адам ортада» шабуылдан басқа тәсілмен де құтылуға болады, мысалы, жіберілетің ашық кілттерге арнайы куаландыру орталықтың цифрлық қолдарын қосу.

Таңдалған ашық мәтін негізіндегі шабуыл

Осындай шабуыл орын алады, егер криптоталдаушы оған берілген «мәтін-шифрлама» жұптан басқа, оған керек мәтіндерді құрастырып оларды шифрлай алса.

Таңдалған ашық мәтін арқылы шабуыл жасауды былай түсіндіруге болады. Ортақ құпиялы кілтті келістіру үшін біз ассиметриялық алгоритм F пайдаланайық. Абоненттің біреуі басқаға 64-битты сеанстық кілтті K жіберсін, ол басқа абоненттың ашық кілтімен y шифрланған $C=F(K, y)$. Қаскүнем цифрланған хабарды C ұстап алып, оны жабық кілті x болмағандықтан дешифрлай алмайды. Бірақ, бұзушы былайда істей алады: K-ң сәйкес мәнің таңдап алуға тырысу. Ол үшін барлық мүмкін болатын ашық мәтіннің 64-битты амалдарын ашық кілтпен y шифрлау қажет және C мен салыстыру. Бұл мүмкін, себебі y пен C мәні ашық түрде берілген болатын. Осындай шабуылдың қауіп-қатері өте актуалды, егер бастапқы хабарлардың саны аса көп емес болса.

Осындай шабуылдан құтылу үшін рандомизировалған (немесе ықтималдық) алгоритмды және ашық кілті бар ЭЦҚ құрастыруды пайдаланады. Осындай алгоритм бірдей хабарды, бірдей кілт болғанда, әр сайын түрліше шифрлайды, өйткені кейбір кездейсоқ элементті пайдаланады. Рандомизировалған ашық кілті бар алгоритмның мысалы – Эль-Гамаль алгоритмы және ГОСТ Р34.10 бойынша ЭЦҚ құрастыру алгоритмы.

Таңдалған ашық мәтін негізіндегі шабуылдан құтылудың басқа варианты – шифрланатын хабарға кейбір қосымша «кездейсоқ» ақпаратты қосу, мысалы, уақыт белгісін.

Негізгі терминдер

RSA алгоритмы – ашық кілті бар шифрлау алгоритмы. Алгоритм аты авторлар фамилиясының бірінші әріптерінен жиңалған: Р.Ривест (R.Rivest), А.Шамир (A.Shamir) және Л.Адлеман (L.Adleman). RSA алгоритмы үлкен сандарды факторизациялау есебінің күрделігіне негізделген. RSA алгоритмы криптографиялық жүйелерде ең кең тараған және жиі қолданылытың ассиметриялық алгоритмы болып табылады.

Диффи-Хеллман алгоритмы - ашық кілті бар шифрлау алгоритмы. Бұл алгоритм дискретты логарифмды есептеуінің күрделігіне негізделген. Диффи-Хеллман алгоритмы симметриялық шифрлауда пайдаланатын кілттерді үлестіру үшін пайдалану мүмкін.

Эль-Гамаль алгоритмы - ашық кілті бар шифрлау алгоритмы, дискретты логарифмның есептеу күрделігіне негізделген. Эль-Гамаль алгоритмы деректерді шифрлау үшін, цифрлық қолды құрастыру үшін және ортақ кілтті келістіру үшін пайдалану мүмкін. Бұл алгоритм абоненттердің ортақ құпиялы кілтін құрастыру үшін Диффи-Хеллман схемасын пайдаланады және сосын осы кілтке көбейтіліп хабар шифрланады.

Шабуыл «адам ортада» (ағыл. «man-in-the-middle») – криптография термины, бұл жағдайда шабуылшы хабарларды оқып өзгерте алады. Ал хабармен алмасатын абоненттер оны (байланыс арнасында шабуылшының бар болғаның) сезбейді.

Эллиптикалық қисықтарға негізделген криптожүйелер – математикалық аппарат ретінде жазықтықтағы эллиптикалық қисықтардың қасиеттерін пайдаланатын ашық кілті бар алгоритмдар тобы.

Сұрақтар

1. RSA алгоритмы қандай мақсаттар үшін қолданылу мүмкін?
2. RSA алгоритмын пайдалануымен шифрлау процесті бейнелеңіз.
3. Диффи-Хеллман алгоритмы қандай мақсаттар үшін қолданылу мүмкін?
4. Диффи-Хеллман алгоритмын пайдалану кезінде іс-әрекет тізбегін бейнелеңіз.
5. Эль-Гамаль алгоритмы қандай мақсаттар үшін қолданылу мүмкін?
6. Эль-Гамаль алгоритмын пайдалану кезінде іс-әрекет тізбегін бейнелеңіз.
7. Ашық кілті бар шифрлау алгоритмдарды пайдалану кезінде қандай шабуылдар болу мүмкін?