Ашық кілтті криптографияның идеялары мен әдістерін жақсы түсінуге көмектесетін үш мәселені қарастырыңыз. Мұның бәрі

тапсырмалардың практикалық маңызы зор.

Бірінші міндет-парольдерді компьютерде сақтау. Біз білеміз,

желідегі әрбір пайдаланушының өзінің құпия паролі бар. Кезінде

желіге кіру пайдаланушы өзінің атын (құпия емес) көрсетеді, содан кейін

құпия сөзді енгізеді. Мәселе мынада: егер сіз парольді сақтасаңыз

компьютер дискісінде, Хауа оны оқи алады, содан кейін рұқсатсыз кіру үшін қолдана алады (бұл әсіресе оңай,

егер Хауа осы желінің жүйелік әкімшісі болып жұмыс істесе). Сондықтан

құпия сөздерді компьютерде сақтауды ұйымдастыру қажет

мұндай "бұзу" мүмкін емес еді.

Екінші міндет радиолокаторлар мен Әуе қорғанысы жүйесінің пайда болуымен туындады. Ұшақ шекараны кесіп өткенде, радиолокатор пароль сұрайды. Егер пароль дұрыс болса, онда ұшақ "өз", әйтпесе — "бөтен". Бұл жерде келесі мәселе туындайды: құпия сөз Ашық арна (ауа ортасы)арқылы берілуі керек,the қарсылас барлық келіссөздерді тыңдап, дұрыс парольді біле алады. Содан кейін "бөтен" ұшақ сұралған жағдайда қайталанадыбұрын ұсталған" дұрыс " құпия сөз локаторға жауап ретінде қабылданбайды.

Үшінші тапсырма алдыңғы тапсырмаға ұқсас және қашықтан қол жетімді компьютерлік желілерде пайда болады, мысалы, өзара әрекеттесу кезінде

банк және Клиент. Әдетте, сеанстың басында банк клиенттен аты-жөнін, содан кейін құпия сөзді сұрайды, бірақ Хауа парольді біле алады,

байланыс желісі ашық болғандықтан.

Бүгінгі таңда бұл мәселелердің барлығы криптографиялық әдістерді қолдану арқылы шешіледі. Осы мәселелердің барлығын шешу маңызды мәселелерге негізделген

бір жақты функция ұғымдары (one-way функциясы).

2.1. Фон және негізгі идеялар 13

Анықтама 2.1. Функция берілсін

y=f(x), (2.1)

кері функция бар X (x € X) ақырлы жиынында анықталған

x = f^{-1} (y). (2.2)

Егер есептеу формула бойынша болса, Функция бір жақты деп аталады (2.1) - аз уақытты қажет ететін қарапайым тапсырма және есептеу

БҚ (2.2) - күрделі міндет, ол есептеу ресурстарының массасын тартуды талап етеді, мысалы, қуатты суперкомпьютердің $10^{6} - 10^{10}$ жыл жұмыс істеуі.

Бұл анықтама, әрине, бейресми. Бір жақты функцияның қатаң анықтамасын [26, 28] табуға болады, бірақ

біздің мақсаттарымыз жоғарыда айтылғандар жеткілікті.

Бір жақты функцияның мысалы ретінде келесіні қарастырыңыз:

y=a^{x}mod p, (2.3)

мұндағы p-кейбір жай сан (яғни онсыз бөлінетін сан

қалдық тек өзіне және бірлікке), ал x - {1, 2, жиынының бүтін саны . . . , p − 1}. Кері функция белгіленеді

x=$\log_a y$ mod p (2.4)

және дискретті логарифм деп аталады.

2.4) бойынша есептеу қиындықтарын қамтамасыз ету үшін

қазіргі уақытта ең жақсы заманауи компьютерлерді пайдалану үшін 512 биттен асатын сандар қолданылады. Іс жүзінде жиі

басқа біржақты функциялар қолданылады, мысалы, хэш функциялары деп аталады, олар айтарлықтай қысқа

шамамен 60-120 биттік сандармен (олар 8-тарауда қарастырылады).

Алдымен біз (2.3) есептеуді жеткілікті жылдам орындауға болатындығын көрсетеміз. Санды есептеу мысалынан бастайық a^{16}modp. Біз жаза аламыз

$$a^{16} \mathrm{mod} p = \left(\left(\left(a^2 \right)^2 \right)^2 \right)^2 \mathrm{mod} p,$$

яғни бұл функцияның мәні a·a· "аңғал" нұсқасымен 15-тің орнына тек 4 көбейту амалында есептеледі. . .a. Осының негізінде жалпы алгоритм.

Алгоритмді сипаттау үшін шаманы енгіземіз

t=log2 x

log2 x бүтін бөлігі (бұдан әрі барлық логарифмдер екілік болады, сондықтан

Бұдан әрі біз 2) негізді көрсетпейміз. Сандарды есептеу

$$a, \quad a^2, \quad a^4, \quad a^8, \quad \ldots, \quad a^{2^t} \qquad (\mathrm{mod}\ p). \qquad (2.5)$$

(2.5) қатарында әр Сан алдыңғы санды p модуліне көбейту арқылы алынады.

екілік санау жүйесіндегі x:

x = $(x_t x_{t-1}$−1 . . . $x_1 x_0)$2.

Содан кейін y=a^xmod p санын келесідей есептеуге болады

$$y = \prod_{i=0}^{t} a^{x_i \cdot 2^i} \ \mathrm{mod}\ p \qquad (2.6)$$

(барлық есептеулер p модулі бойынша жүргізіледі).

Мысал 2.1. 3^{100} mod 7 есептеу қажет болсын. Бізде бар

t =log 100 = 6. Қатардың сандарын есептеңіз (2.5):

$$\begin{array}{ccccccc} a & a^2 & a^4 & a^8 & a^{16} & a^{32} & a^{64} \\ 3 & 2 & 4 & 2 & 4 & 2 & 4 \end{array} \qquad (2.7)$$

Көрсеткішті екілік санау жүйесінде жазамыз:

100 = (1100100)2

және (2.6) формула бойынша есептеулер жүргіземіз:

$$\begin{array}{ccccccccccc} a^{64} & & a^{32} & & & & a^4 & & & & \\ 4 & \cdot & 2 & \cdot & 1 & \cdot & 4 & \cdot & 1 & = & 4 \end{array} \qquad (2.8)$$

Бізге тек 8 көбейту операциясы қажет болды (есептеу үшін 6

жолдар (2.7) және 2 (2.8)). ut

Жалпы жағдайда мыналар әділ

2.1. Фон және негізгі идеялар 15

Бекіту 2.1 (есептеудің күрделілігі туралы (2.3)). Сипатталғандай есептеу кезінде көбейту операцияларының саны (2.3)

әдіс 2 log x-тен аспайды.

Д О к а з а т е л с т в о . (2.5) қатарының сандарын есептеу үшін t көбейту қажет, (2.6) бойынша y есептеу үшін t артық емес

көбейту (2.1 мысалды қараңыз). Шарттан t =log x

, log x≤ log x екенін ескере отырып, дәлелденген мәлімдеменің әділдігі туралы қорытынды жасаймыз. ut

З а м е ч а н және Е. болашақта көрсетілгендей, в тұрғызу кезінде

p Модулінің дәрежесі тек X < p көрсеткіштерін қолданудың мағынасы бар.

Бұл жағдайда көбейту операцияларының Саны деп айта аламыз

есептеу кезінде (2.3) 2 log p-ден аспайды.

Кері функцияны есептеудің бірдей тиімді алгоритмдері (2.4) белгісіз екенін ескеру маңызды. "Нәресте қадамы, алыптың қадамы" деп аталатын есептеу әдістерінің бірі (2.4) егжей-тегжейлі сипатталатын болады бөлім. 3.2. Бұл әдіс 2 ретті қажет етеді

√p операциялары.

Үлкен p кезінде (2.3) функциясы кері функцияны есептеу үшін пайдаланылса, шынымен бір жақты екенін көрсетейік

"нәресте қадамы, алыптың қадамы" әдісі. Біз келесі нәтижені аламыз (кесте. 2.1).

Таблица 2.1. **Количество умножений для вычисления прямой и обратной функции**

Количество десятичных знаков в записи p	Вычисление (2.3) ($2\log p$ умножений)	Вычисление (2.4) ($2\sqrt{p}$ умножений)
12	$2 \cdot 40 = 80$	$2 \cdot 10^6$
60	$2 \cdot 200 = 400$	$2 \cdot 10^{30}$
90	$2 \cdot 300 = 600$	$2 \cdot 10^{45}$

Егер біз 50-100 модульдерді қолдансақ

Ондық сандар," түзу " функциясы тез есептеледі, ал кері мәні іс жүзінде есептелмейді. Мысалы, 10-14 секундта екі 90 таңбалы санды көбейтетін суперкомпьютерді қарастырайық.

(қазіргі компьютерлер үшін бұл әлі қол жетімді емес). Есептеу үшін (2.3) мұндай компьютер қажет болады

$$T = 600 \cdot 10^{-14} = 6^{-12} \cdot 10 \text{ сек.},$$

ал есептеу үшін (2.4) —

$$T = 600 \cdot 10^{-14} = 6^{-12} \cdot 10 \text{ сек.},$$

яғни 10 22 жылдан астам. Біз кері функцияларды есептеуді көреміз

сандардың ұзындығы шамамен 90 ондық болғанда іс жүзінде мүмкін емес

және параллель есептеулер мен компьютерлерді пайдалану

желілер жағдайды айтарлықтай өзгертпейді. Қарастырылған мысалда

біз кері функция 2 үшін есептеледі деп ойладық

√p операциялары. Қазіргі уақытта дискретті логарифмді есептеудің "жылдам" әдістері де белгілі, бірақ жалпы сурет бірдей —

оларда талап етілетін операциялардың саны 2 log p-ден көп.

осылайша, (2.3) функциясы шынымен біржақты, бірақ ескертумен деп айтуға болады. Кері функцияны (2.4) "түзу" сияқты жылдам есептеу мүмкін доказстігін ешкім дәлелдеген жоқ.

Барлығын шешу үшін бір жақты функцияны (2.3) қолданыңыз

осы бөлімнің басында сипатталған үш тапсырма, алайда, кез-келген басқа да қолдануға болатындығын ұмытпайды

бір жақты функция.

Құпия сөздерді компьютер жадында сақтаудан бастайық. Мәселені шешу парольдердің мүлдем сақталмайтындығына негізделген! Дәлірек айтқанда,

желіде тіркелу пайдаланушы өзінің аты мен паролін тереді; рұқсат етіңіз,

мысалы, оның аты - "жеміс", ал пароль — "өрік". Компьютер

"өрік" сөзін x санының екілік жазбасы ретінде қарастырады және (2.3) есептейді, мұндағы a және p — екі құпия емес сандар, тіпті бәріне белгілі. Осыдан кейін компьютер жадында іске қосылады

жұп (аты, y), мұндағы y (2.3) бойынша есептеледі x = құпия сөз. Барлығы үшін

жұпты енгізгеннен кейін осы пайдаланушының қосымша кірістері ("жеміс",

"өрік"), компьютер есептейді (2.3) уновтың Жаңа мәні

x = "өрік" және жадта сақталған бұрын есептелген y мәнімен салыстырады . Егер унов жадта сақталғанға сәйкес келсе

берілген атқа сәйкес келетін y, содан кейін бұл заңды пайдаланушы.

Әйтпесе, бұл Хауа.

Хауа X-ті y-ден табуға тырысуы мүмкін . Алайда біз мұны көрдік

қазірдің өзінде 90 таңбалы сандармен бұл үшін 10 22-ден астам қажет болады

жылдар. Осылайша, ұсынылған парольді сақтау жүйесі өте сенімді және қазіргі уақытта көптеген нақты әлемде қолданылады

операциялық жүйелер.

Екінші мәселенің шешімін қарастырыңыз (Әуе қорғанысы және ұшақ). Сіз жасай аласыз

келесі әдісті қолданыңыз. Әрбір" өзінің " ұшағына Әуе қорғанысы жүйесі мен ұшқышқа, дәлірек айтқанда борттық компьютерге белгілі құпия атау беріледі. Мысалы, ұшақтардың біріне рұқсат етіңіз

сұңқардың Құпия атауы берілген және бұл ұшақ жақындап келеді

шекарасы 2005 жылғы 01 ақпанда сағат 12-де.45 мин. содан кейін ұшақтың борттық компьютері шекараға жақындағанға дейін сөз жасайды

сұңқар 05 02 01 12 45

(аты жыл ай күн сағат минуттар).

Басқаша айтқанда, ұшақтағы компьютер мен Әуе қорғанысы станциясы құпия сөзге ағымдағы уақыт туралы мәліметтерді қосады және алынған сөзді x саны ретінде қарастыра отырып, y = a есептейді

X mod p, қайда

а және р құпия емес. Содан кейін ұшақ Әуе қорғанысы станциясының у санын хабарлайды.

Станция өзі есептеген у санын ұшақтан алынған санмен салыстырады. Егер есептелген және алынған мәндер сәйкес келсе, онда ұшақ

"өз"деп танылады.

Қарсылас бұл жүйені бұза алмайды. Шынында да, бірге

бір жағынан, ол сұңқардың құпия сөзін білмейді және мүмкін емес

оны у бойынша табыңыз, өйткені Х-ті у бойынша есептеу, айталық, 10 22 алады

жылдар. Екінші жағынан, ол жай ғана у көшіре алмайды және оны болашақта жауап ретінде пайдалана алмайды, өйткені қиылысу уақыты

шекаралар ешқашан қайталанбайды және келесі у мәндері болады

түпнұсқадан өзгеше.

"Әуе қорғанысы мәселесін" шешудің қарастырылған нұсқасы ұшақта және локаторда сағатты дәл синхрондауды қажет етеді. Бұл мәселе

оңай шешіледі. Мысалы, навигация қызметі уақыт белгілерін үнемі ашық түрде жібереді (уақыт құпия емес),

барлық ұшақтар мен локаторлар бұл белгілерді сағаттарын синхрондау үшін пайдаланады. Бірақ одан да нәзік мәселелер бар. Уақыт белгісі

барлық есептелген мәндер үшін х сөзіне қосылады

у әр түрлі болды және қарсылас оларды қайта қолдана алмады.

Алайда, қарсылас ағымдағы минут ішінде у-ді бірден қайталауға тырысуы мүмкін. Бұл мүмкіндікті қалай болдырмауға болады? Бұл

бірінші сұрақ. Тағы бір қиындық ұшақ 45-ші минуттың соңында у санын жіберіп, локатор қабылдайтын жағдайда туындайды

оның басында 46-шы. Біз оқырманға осы мәселелерді шешудің нұсқасын ұсынуға мүмкіндік береміз.

Егер біз "әуе қорғанысы мәселесін" шешудің тағы бір әдісі мүмкін

қосымша ашық деректер арнасын пайдаланыңыз

ұшаққа локатор. Жоғарыда айтылғандай, әр" өз " ұшағы мен локаторы олар ауыстырылмайтын құпия сөзді (сұңқар сияқты) біледі. Мақсатты анықтағаннан кейін локатор оған кездейсоқ пайда болған санды жібереді

а ("қоңырау"). Ұшақ у = а есептейді

X mod p, мұндағы х-құпия

сөз ("Сұңқар"), және Y санын локаторға айтады. Локатор бірдей есептеулерді шығарады және есептелген у мен қабылданған салыстырады.

Бұл схемада сағатты синхрондаудың қажеті жоқ, бірақ бұрынғыдай қарсылас у санын қайталай алмайды , өйткені локатор әр уақытта әр түрлі қоңыраулар жібереді (а). Бір қызығы, бұл тапсырма,

тарихи тұрғыдан алғанда, біржақты функциялар бірінші болып шешілді.

Үшінші мәселе дәл осылай шешіледі және парольді қалыптастырудың екі әдісі де қолданылады және қолданылады

нақты желілік хаттамалар.