В.А. Романьков

Введение в криптографию

В.А. Романьков

ВВЕДЕНИЕ В КРИПТОГРАФИЮ

Курс лекций

2-е издание, исправленное и дополненное

Рекомендовано студентам высших учебных заведений

МОСКВА
2012

УДК 519.6(075.8)
ББК 22.19я73
Р69

Рецензенты:
доктор технических наук, профессор *Р.Т. Файзуллин*;
доктор физико-математических наук, профессор *Л.М.Мартынов*

Романьков В.А.

Р23 Введение в криптографию. Курс лекций / В.А. Романьков. — М. : ФОРУМ, 2012. — 240 с. — (Высшее образование).

ISBN 978-5-91134-573-0

В курсе лекций изложены математические основы современной криптографии, описан ряд криптографических схем и протоколов, имеющих важное теоретическое значение и широкое практическое применение. Лекции сопровождаются примерами и задачами.

Адресован студентам, впервые изучающим криптографию. Может также оказаться полезным специалистам в данной области.

УДК 519.6(075.8)
ББК 22.19я73

ISBN 978-5-91134-573-0

Содержание

Предисловие автора ко 2-му изданию

Эта книга основана на записках лекций по курсу "Криптография", читанных автором в Омском государственном университете им. Ф.М. Достоевского и ряде других университетов в течение последних 10-15 лет. Она представляет именно введение в предмет, не загруженное излишними деталями, компактное и информативное. Автор надеется, что ее будущие читатели не только смогут ориентироваться в мире криптографии и ее приложений, но также, если возникнет соответствующее желание или необходимость, быстро перейдут к чтению специальной литературы по криптографии и ее приложениям.

Прежде всего книга ориентирована на студентов, впервые знакомящихся с криптографией. Ее можно рассматривать как учебник, тем более, что текст сопровождается рядом примеров, задач и упражнений. Некоторые из задач широко известны, но большинство являются оригинальными. Автор также надеется, что книга окажется полезной для преподавателей курсов по криптографии и защите информации. Специалисты в области криптографии также смогут найти в ней кое-что интересное.

Книга содержит 21 лекцию и 4 приложения, в которых даются математические и теоретико-информационные основы теории, представляются известные криптографические системы и протоколы, анализируется их криптографическая стойкость. Изложение дополняется задачами и упражнениями. Их число по сравнению с предыдущим изданием книги возросло. Кроме того в настоящем издании исправлены замеченные неточности. Автор благодарен студентам, приславшим свои замечания, в особенности, Ивану Антропову.

По сравнению с 1-м изданием, вышедшем в 2006 г., книга дополнена тремя новыми разделами (лекции 5, 13 и приложение 4),

а также рядом задач и упражнений. Изменения и дополнения коснулись и других разделов. Исправлены замеченные неточности и опечатки. Автор благодарен читателям, оказавшим помощь в их обнаружении, особенно – Артему Голубниченко.

Автор надеется дополнить в самое ближайшее время книгу задачником по криптографии. Кроме этого хотелось бы уделить большее внимание современным направлениям в криптографии, из которых в книге затронута только алгебраическая криптография, основанная на бесконечных группах. Имея в виду эти продолжения, автор призывает своих читателей высказывать замечания по тексту. Автор заранее признателен всем, кто захочет это сделать.

В.А. Романьков
Омский государственный университет им. Ф.М. Достоевского
Институт математики и информационных технологий
romankov@omsu.ru
Омск, 2011 г.

Введение

1. История

Криптография – наука о шифрах. История возникновения и развития криптографии восходит к древнейшим временам. Первые сведения о применении шифров в военном деле связаны с именем спартанского полководца Лисандра, использовавшего для передачи сообщений шифр «Сцитала». Известен «шифр Цезаря», которым пользовался Юлий Цезарь в своей переписке. В Древней Греции был изобретен шифр, именуемый как «квадрат Полибия».

Одна из первых книг по шифровке написана аббатом Трителлием (1462–1516) из Германии. В 1566 г. знаменитый математик Кардано опубликовал метод шифрования, известный как «решетка Кардано». Известны шифры короля Генриха IV, кардинала Ришелье, а также «цифирная азбука», автором которой был Петр Великий. Шифрование на Руси существовало и до Петра, называлось оно тайнописью.

Большое влияние на развитие криптографии оказали работы Клода Шеннона, заложившего основы теории информации. Принято считать, что теория информации как наука родилась в 1948 г. после публикации знаменитой работы Шеннона «Математическая теория связи» (см. [14]).

В настоящее время криптография переживает бурный рост. Необходимость защиты информации, передаваемой по современным сетям, а также огромные возможности ее взлома и искажения, обусловленные развитием компьютерной техники, требуют совершенно нового подхода к защите информации, новых методов шифрования. Замечательно, что при этом используется современная математика.

2. Основные понятия и термины

Шифрование – преобразование текста с целью скрыть его содержание от несанкционированного прочтения.

Криптография – искусство составления шифров и шифропротоколов.

Криптоанализ – наука (и практика ее применения) о методах и способах вскрытия шифров.

Криптология – наука, состоящая из двух ветвей: криптографии и криптоанализа.

Криптосистемой (системой шифрования) называется известный способ шифрования, обладающий сменными элементами – *ключами*. Обычно рассматривают ключ шифрования и ключ дешифрования. Если их необходимо хранить в секрете, то система называется *симметричной*, или *системой с секретными ключами*. Если ключ шифрования открыт, то есть не является секретным, то такие криптосистемы называются *асимметричными*, или *системами с открытым ключом*.

Шифруемый текст мы будем называть *исходным текстом* (термин английского языка – *plaintext*). Результат шифрования называется *шифротекстом* (иногда *шифровкой*, по-английски – *ciphertext*).

Система шифрования включает в себя в качестве основного элемента *функцию шифрования*. Функция шифрования преобразует исходный текст в шифровку. Она зависит от ключа шифрования.

Основное правило шифрования гласит: **исходный текст должен однозначно восстанавливаться по шифротексту**, то есть функция шифрования должна быть обратимой. Способность криптосистемы производить шифровки, трудные для несанкционированного прочтения (*взлома*), называется ее *криптостойкостью*. Стойкие системы таковы, что нахождение обратной к функции шифрования в общем случае – трудная задача. Однако при знании соответствующего секрета – ключа дешифрования – это нахождение эффективно выполнимо.

Мы пришли к определению *односторонней функции* $y = f(x)$, где вычисление y по x осуществляется достаточно просто, а нахождение (какого-либо) x по y – трудная задача.

Уже несколько раз употреблен термин «трудная задача». Пока это неформальное понятие, связанное с возможностью решения задачи в реальное время реальными средствами. Понятие реального зависит как от важности задачи, так и от возможностей пытающегося ее решить.

Всем таким понятиям можно придать формальный смысл, но это остается за рамками предлагаемого курса лекций.

Для шифрования используют *односторонние функции* $y = f(x)$ *с секретом*. Во-первых, функция f должна быть обратимой, во-вторых, обратную к ней функцию f^{-1} можно найти эффективно, зная секрет, как правило, это – ключ дешифрования. Все это нужно для того, чтобы санкционированный пользователь, кому собственно шифровка и предназначена, мог ее прочитать.

Обратим внимание на следующее обстоятельство. В современной криптографии используемая криптосистема считается открытой, то есть метод шифрования не является секретным. При этом ее криптостойкость определяется не только качеством криптосистемы, но и надежностью используемых ключей. Гипотетически считается, что возможный взломщик системы не только умен и образован, но и обладает большими возможностями (финансовыми, мощной оргтехникой и т. п.). Надежные криптосистемы должны обеспечивать защиту от таких взломщиков.

Платформы шифрования

А – черный, белый – Е;
И – красный, У – зеленый.
О – синий; тайну их
скажу я в свой черед.
А. Рембо. Гласные

1. Алфавит

Мы рассматриваем тексты. Тексты записываются в виде конечных последовательностей знаков соответствующего алфавита.

Алфавит может иметь обычный смысл: 33 буквы от А до Я в русском языке или 26 букв от A до Z в английском. Однако в криптографии понятие *алфавит* имеет более широкий смысл. Так называют совокупность знаков, среди которых могут встречаться обычные буквы, цифры, знаки препинания, специальные знаки, обозначающие пробелы, и всевозможные другие знаки. Можно, конечно, называть все это *знаковой системой.*

Текст есть конечный упорядоченный набор знаков. Вовсе не обязательно, чтобы он имел смысл. Есть еще одно понятие – *единица текста.* В простейшем случае единицами текста считают знаки, участвующие в его записи. Однако в качестве единиц текста могут выступать более сложные образования. Например, мы можем считать единицами текста последовательные пары знаков – *диграфы,* или тройки – *триграфы,* или наборы из k знаков – *k-графы.* При этом возникает маленькая неприятность – последний k-граф текста может оказаться неполным. Для этого случая необходимо специальное соглашение. Например, можно дополнить последний k-граф знаками выбранного вида.

Кроме этого, тексты часто разбивают на блоки. *Блок* – это некоторая конечная последовательность единиц текста. Обычно длина блока фиксирована.

2. Оцифровка

Одним из самых простых способов шифрования является *простая замена.* Определяется правило, по которому каждая буква алфавита заменяется на некоторую другую букву. Допускается также замена некоторых букв на самих себя. Основное требование – разные буквы не должны заменяться одинаковыми. Это необходимо для того, чтобы исходный текст определялся по шифровке однозначно. Затем текст переписывается в новых буквах. Такой способ шифрования в наше время не считается надежным. Он подвержен частотному анализу, ведь частоты использования различных букв в осмысленных текстах могут очень сильно различаться. Более подробно мы поговорим об этом в следующих лекциях. Другая слабость простой замены заключается в том, что зная хотя бы часть исходного текста и соответствующую часть зашифрованного, мы получаем информацию, позволяющую определить часть остального текста, ведь замены букв в нем те же самые. Современные методы шифрования разрабатываются таким образом, чтобы частичная расшифровка не влияла существенно на возможность расшифровки всего текста. Докомпьютерные методы шифрования обычно основываются на двух элементах: заменах букв на другие буквы и перестановках букв в тексте. Анализ с использованием компьютеров позволяет легко расшифровывать такие тексты без знания ключей, то есть осуществлять взлом. Для построения современных шифров обычная запись текста в некотором алфавите заменяется на кодированную запись в виде элемента некоторой арифметической или алгебраической системы – платформы шифрования. Перейдем к их описанию.

В криптографии, как правило, работают с оцифрованными текстами. Единицы текста заменяют на числа различной природы. Это могут быть обычные целые числа, элементы колец выче-

тов, конечных полей и т. п. Все это более подробно будет рассматриваться в следующих лекциях.

Оцифровка не является шифрованием. Исходный текст в его записи в данном алфавите должен легко и однозначно восстанавливаться по своему оцифрованному виду.

Приведем стандартные оцифровки обычных алфавитов – русского и английского:

РУССКИЙ

а(0) б(1) в(2) г(3) д(4) е(5) ё(6) ж(7) з(8) и(9) й(10) к(11) л(12) м(13) н(14) о(15) п(16) р(17) с(18) т(19) у(20) ф(21) х(22) ц(23) ч(24) ш(25) щ(26) ъ(27) ы(28) ь(29) э(30) ю(31) я(32)

АНГЛИЙСКИЙ

a(0) b(1) c(2) d(3) e(4) f(5) g(6) h(7) i(8) j(9) k(10) l(11) m(12) n(13) o(14) p(15) q(16) r(17) s(18) t(19) u(20) v(21) w(22) x(23) y(24) z(25)

Текст «ястудентуниверситета» оцифровывается следующим образом: 32, 18, 19, 20, 4, 5, 14, 19, 20, 14, 9, 2, 5, 17, 18, 9, 19, 5, 19, 0.

Текст «iamastudent» – 8, 0, 12, 0, 18, 19, 20, 3, 4, 13, 19.

Оцифровка 15, 13, 18, 11 в русском алфавите восстанавливается как омск.

Отсутствие в выбранном алфавите заглавных букв и знаков, обозначающих пробелы, вынуждает нас писать тексты слитно, употребляя только строчные буквы. Если предполагается использовать как строчные, так и заглавные буквы, это должно быть зафиксировано в оцифровке. Все зависит от выбранного алфавита.

Введем для использования искусственный алфавит **Ы**, приведя оцифровку его знаков:

Ы

a	&	α	?	1	2	7	$\_$
0	1	2	3	4	5	6	$\overline{7}$

Мы видим, что в **Ы** предусмотрен знак пробела «_». Текст «$a\&21\_?$» оцифровывается как $0, 1, 5, 4, 7, 3$.

Допустим теперь, что в алфавите **Ы** мы в качестве единиц текста выберем диграфы – пары знаков. Самый простой способ их нумерации следующий. Считаем, что паре $\alpha, 2 \longleftrightarrow 2, 5$ отвечает запись числа $(2, 5)_8$ в 8-ричной системе счисления. То есть $(2, 5)_8 = 5 + 2 \cdot 8 = 21$. Всего возможных пар $8^2 = 64$. Все они нумеруются двузначными числами в 8-ричной системе счисления от $(0, 0)_8 = 0$ до $(7, 7)_8 = 7 + 7 \cdot 8 = 63$.

Если мы рассмотрим k-граф $(a_{k-1}, a_{k-2}, \ldots, a_0)$, то ему можно сопоставить число $(a_{k-1}, a_{k-2}, \ldots, a_0)_8 = a_0 + a_1 \cdot 8 + a_2 \cdot 8^2 + \ldots + a_{k-1} \cdot 8^{k-1}$.

Для русского алфавита удобно выбрать аналогичную нумерацию в 33-ричной системе.

Например, 3-графу год $\longleftrightarrow 3, 15, 4$ можно сопоставить число $(3, 15, 4)_{33} = 4 + 15 \cdot 33 + 3 \cdot (33)^2 = 3766$. Для английского алфавита желательно взять 26-ричную систему и т. п.

Современная компьютерная техника устроена таким образом, что почти всегда тексты записываются в кодированном виде как бинарные (двоичные) последовательности. Обозначение $m \in \{0, 1\}^*$ означает, что m – бинарная последовательность.

Для использования бинарных последовательностей удобно, чтобы алфавит содержал 2^k знаков. Например, бинарными последовательностями длины 8 записываются байты, число которых $2^8 = 256$.

В алфавите **Ы** в точности $2^3 = 8$ букв, каждой из которых мы сопоставим двоичный номер: $a \longleftrightarrow 000$, $\& \longleftrightarrow 001$, $\alpha \longleftrightarrow 010$, $? \longleftrightarrow 011$, $1 \longleftrightarrow 100$, $2 \longleftrightarrow 101$, $7 \longleftrightarrow 110$, $\_ \longleftrightarrow 111$. Если в алфавите 2^k букв, то каждой из них сопоставляется последовательность из k нулей и единиц.

3. Платформы шифрования

Платформой называется система, используемая для записи текста. Это может быть совокупность бинарных последовательно-

стей, кольцо вычетов, конечное поле, эллиптическая кривая и т. п. Более подробно все это рассматривается в следующих лекциях.

Иногда платформы исходного и зашифрованного текста разные. Это может быть связано с использованием разных алфавитов, может быть вызвано различным выбором единиц текстов или еще какими-нибудь причинами.

В последние годы получило интенсивное развитие новое направление криптографии – *алгебраическая криптография.* Также используется термин *криптография, основанная на теории групп (group-based cryptography).*

Основным отличительным признаком алгебраической криптографии является использование в качестве платформы шифрования абстрактной группы. Как правило, группа берется бесконечная. Можно заметить, что и классическая криптография часто использует в качестве платформы группу. Это может быть мультипликативная группа конечного поля, группа обратимых элементов кольца вычетов, группа точек эллиптической кривой над конечным полем. Мы видим, что все перечисленные группы коммутативны и конечны. Предпринимались и другие попытки использования конечных и матричных групп для шифрования, но систематическое построение алгебраической криптографии началось совсем недавно. Этой области будет посвящена лекция 21.

4. Хэш-функции

Криптографическая *хэш-функция* h – это функция, переводящая произвольную конечную бинарную последовательность в последовательность определенной длины. Требуется, чтобы хэш-функция была односторонней. Используются также хэш-функции со значениями в конечных полях, кольцах вычетов или группах эллиптических кривых.

В криптографии хэш-функции используются для различных целей: образования так называемых дайджестов $h(m)$ для сообщений m, создания образов некоторых данных для их безопасного хранения и т. п. Существуют многочисленные стандарты хэш-функций. В данном курсе мы неоднократно будем упоминать хэш-функции, но специальной лекции им не отводим.

Задачи и упражнения

Задача 1.1. Восстановите первую строку одной из наиболее известных шифровок на русском языке:

«...
Его мы очень смирным знали,
Гроза двенадцатого года,
Но бог помог – стал ропот ниже...»

Задача 1.2. Восстановите начало одного из самых известных монологов на английском языке, зашифрованное простой заменой букв на числа:

$$11, 17\_3, 7\_17, 10\_0, 17, 11\_11, 17\_3, 7\_$$

$$\_11, 21, 23, 11\_6, 8\_11, 21, 7\_20, 5, 7, 8, 11, 6, 17, 0$$

Что помогает при расшифровке? Сколько в английском языке слов из одной буквы? Из двух букв? Объясните, почему зашифрованный текст лучше записывать без пробелов и без знаков препинания.

Задача 1.3. Испанский алфавит состоит из 29 букв. Ниже дается оцифровка начала одного из известнейших стихотворений испанской поэзии:

$$5, 14, 18, 9, 28, 0\_5, 12\_13, 0, 15, 22, 17\_$$

$$\_4, 5\_12, 0\_7, 23, 9, 22, 0, 20, 20, 0\_$$

$$\_21, 5\_20, 17, 14, 18, 5, 15\_12, 0, 21\_2, 17, 18, 0, 21\_$$

$$\_4, 5\_12, 0\_14, 0, 4, 20, 23, 7, 0, 4, 0$$

Можно легко восстановить текст по испанскому алфавиту. Заметим только, что у него есть варианты.

Следующая строфа – перевод этого текста на русский язык, принадлежащий М. Цветаевой:

10_4_6_11_4_4

В ней указано только количество букв в словах. Приведите эти знаменитые строки!

Задача 1.4. Имеется достаточно длинный литературный текст английского языка. Текст зашифрован методом простой замены. Взломщику разрешается узнать кодировку любой буквы по его выбору. Как он может узнать кодировки сразу двух букв? Какие это буквы? Можно ли с большой долей вероятности узнать три буквы, если разрешается выяснить замены только двух букв?

Задача 1.5. Во время 2-й Мировой войны некоторые из американских солдат, чтобы указать место своего пребывания (открыто писать было нельзя – письма просматривались), меняли на конверте второй из инициалов адресата, например, писали вместо "A.B. Smith" – "A.T. Smith". Буква "T" означала первую букву места дислокации. Затем в следующих письмах появлялось "U, N, I, S". В итоге родные должны были все это заметить и прочитать "TUNIS". Объясните, почему все-таки часто это не срабатывало, хотя родные и понимали идею.

Модулярная арифметика

1. Кольцо целых чисел

Множество целых чисел $\mathbf{Z} = \{0, \pm 1, \pm 2, \ldots, \pm k, \ldots\}$ вместе с обычными операциями сложения «+» и умножения «·» образует коммутативное ассоциативное кольцо с 1. Так говорят, потому что наряду с аксиомами, задающими алгебраический объект, называемый кольцом:

1) $(a + b) + c = a + (b + c)$ *(ассоциативность сложения)*,

2) существует элемент 0 (нуль), для которого верно тождество $a + 0 = 0 + a = a$ *(существование нуля)*,

3) для любого элемента a существует противоположный элемент, обозначаемый $(-a)$, такой, что $a + (-a) = (-a) + a = 0$ *(существование противоположного элемента)*,

4) $a + b = b + a$ *(коммутативность сложения)*,

5) $a \cdot (b + c) = a \cdot b + a \cdot c$ *(правая дистрибутивность)*,

6) $(a + b) \cdot c = a \cdot c + b \cdot c$ *(левая дистрибутивность)*,

также верны дополнительные аксиомы:

7) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ *(ассоциативность умножения)*,

8) существует элемент 1 (единица) такой, что верно тождество $a \cdot 1 = 1 \cdot a = a$ *(существование единицы)*,

9) $a \cdot b = b \cdot a$ *(коммутативность умножения)*.

Если мы оперируем с элементами кольца $\mathbf{Z}$, используя операции сложения и умножения, то это означает, что мы оперируем в обычной арифметике целых чисел. В ней можно определить операцию вычитания «$-$», полагая $a - b = a + (-b)$. В ней можно производить деление «:», полагая $a : b = c$ в том и только том случае, если $a = b \cdot c$ (в дальнейшем мы будем для простоты убирать знак умножения «·», записывая, например, последнее

равенство как $a = bc$). Обращаем внимание на то, что деление не всегда определено. Мы не можем делить на 0, мы не можем писать $5 : 7 = \frac{5}{7}$, поскольку оперируем **только** с целыми числами.

Мы пишем $a \vdots b$ (a делится на b) или равносильно $b|a$ (b делит a), если такое деление возможно, то есть существует элемент c такой, что $a = bc$.

В кольце целых чисел $\mathbf{Z}$ всегда выполняется деление с остатком на ненулевой элемент (на нуль нельзя делить даже с остатком). А именно: для любой пары чисел a, b, где $b \neq 0$, найдется единственное число r (остаток) такое, что, во-первых, $a = bc + r$ для некоторого $c \in \mathbf{Z}$, во-вторых, $0 \leq r < |b|$. Как всегда, $|b|$ означает абсолютную величину числа b.

Например: $12 = 5 \cdot 2 + 2$, $-14 = 9 \cdot (-2) + 4$, $15 = 8 \cdot 1 + 7$.

Если, например, $b = 19$, то возможные остатки выглядят так: $0, 1, 2, \ldots, 18$.

Оказывается, что для любого положительного целого числа $n \geq 2$ (модуля) можно определить арифметику остатков на множестве вычетов по модулю n, которое обозначается как $\mathbf{Z}_n$. Эта арифметика (для каждого n своя) находит широкое применение в криптографии.

2. Кольца вычетов

Итак, множество $\mathbf{Z}_n$ для любого положительного целого $n \geq 2$ состоит из элементов $0, 1, 2, \ldots, n-1$, называемых *вычетами*. Операции определяются следующим образом:

1) чтобы вычислить сумму двух вычетов $a + b$, нужно вычислить сумму $a + b$ в кольце $\mathbf{Z}$, а затем в качестве результата взять остаток этой суммы при делении ее на n;

2) аналогично результатом произведения ab двух вычетов считается остаток при делении обычного произведения целых чисел ab на n.

Пример 1. $5 + 7 = 4(\mathrm{mod}8), 13 + 11 = 0(\mathrm{mod}24), 2 \cdot 8 = 1(\mathrm{mod}\,5), 4 \cdot 21 = 6(\mathrm{mod}26), 3 \cdot 6 = 0(\mathrm{mod}18)$.

Определим, что означает *сравнение по модулю*. Полагаем $a = b(\mod n)$, если разность $a-b$ делится на n в кольце $\mathbf{Z}$: $a-b \vdots n$. Другими словами, $a = b(\mathrm{mod} n)$ в том и только том случае, если существует целое число k такое, что выполнено равенство $a = b + nk$. Указанное сравнение позволяет использовать для вычетов разные имена. Например, сравнение $5 = 25(\mathrm{mod}10)$ означает, что вычет 5 может быть записан под другим именем: 25. Мы будем называть обозначения вычетов $0, 1, \ldots, n-1$ *стандартными именами* элементов кольца $\mathbf{Z}_n$.

Примеры показывают, что в модулярной арифметике произведение ненулевых элементов может оказаться нулем, произведение элемента, отличного от 1 и (-1), на другой элемент кольца $\mathbf{Z}_n$ может оказаться равным 1.

Если $ab = 0(\mathrm{mod}n)$, в то время как $a \neq 0$ и $b \neq 0$, то говорят, что a, b *делители нуля*. Если $cd = 1(\mathrm{mod}n)$, то говорят, что c и d *взаимно обратные* и обозначают $d = c^{-1}(\mathrm{mod}n)$ или $c = d^{-1}(\mathrm{mod}n)$.

Если в кольце $\mathbf{Z}_n$ (или в каком-нибудь другом) нет делителей нуля, то говорят, что $\mathbf{Z}_n$ – *область целостности*. Если для какого-нибудь элемента $a \in \mathbf{Z}_n$ существует $a^{-1} \in \mathbf{Z}_n$, то говорят, что a *обратим* в $\mathbf{Z}_n$. Очевидно, что обратима единица и не обратим нуль.

Следующее утверждение очень важно для криптографии.

Предложение 2. *Элемент $0 \neq a \in \mathbf{Z}_n$ обратим тогда и только тогда, когда нод$(a, n) = 1$.*

Доказательство. Пусть a обратим. Это означает, что существует такой вычет b, что выполнено равенство $ab = 1(\mathrm{mod}n)$, то есть существует такое число k, для которого $ab = 1 + nk$. Если бы a, n делились на число $d > 1$ (общий делитель), мы получили бы противоречие в приведенном равенстве: $ab \vdots d, nk \vdots d$, из чего следует, что $1 \vdots d$. Таким образом, взаимная простота чисел a, n является необходимым условием обратимости $a(\mathrm{mod}n)$.

Покажем, что это условие также достаточно. Для этого воспользуемся обобщенным алгоритмом Эвклида. Мы дадим описание алгоритма без доказательств.

Пусть требуется найти $d = \text{нод}(k, n)$. Для этого начинаем деление с остатком до тех пор, пока этот остаток не станет нулем:

$$\begin{aligned} n &= k \cdot t_1 + r_1, \\ k &= r_1 \cdot t_2 + r_2, \\ r_1 &= r_2 \cdot t_3 + r_3, \\ &\dots, \\ r_{s-1} &= r_s \cdot t_{s+1} + r_{s+1}, \\ r_s &= r_{s+1} \cdot t_{s+2} + 0. \end{aligned} \tag{2.1}$$

Значит, $d = r_{s+1}$. Обратите внимание, что на каждом следующем шаге мы делим предыдущий делитель на полученный остаток. Итак, мы показали, как вычисляется $d = \text{нод}(k, n)$.

Покажем теперь, как найти целые числа l, t такие, что выполняется равенство

$$kl + nt = d. \tag{2.2}$$

Добавление этой процедуры к алгоритму Эвклида дает так называемый *обобщенный алгоритм Эвклида*. В формуле (2.1) движемся в обратном направлении, начиная с предпоследнего равенства, которое мы перепишем как

$$d = r_{s-1} - r_s t_{s+1},$$

подставляя значения предыдущих остатков и выполняя формальные преобразования:

$$d = r_{s-1} - (r_{s-2} - r_{s-1} t_s) t_{s+1} = r_{s-1}(1 + t_s t_{s+1}) - r_{s-2} t_{s+1} = \dots,$$

пока не получим равенство вида

$$d = \dots = kl + nt,$$

что и требовалось. Если $d = 1$, то это означает, что

$$kl = 1(\mathrm{mod}n). \tag{2.3}$$

Предложение доказано.

Пример 3. *Вычислим* $17^{-1}(\mathrm{mod}125)$.

$125 = 17 \cdot 7 + 6,$

$17 = 6 \cdot 2 + 5,$

$6 = 5 \cdot 1 + 1,$

$5 = 1 \cdot 5 + 0,$

далее:

$1 = 6 - 5 \cdot 1 = 6 - (17 - 6 \cdot 2) \cdot 1 = 6 \cdot 3 - 17 \cdot 1 =$

$= (125 - 17 \cdot 7) \cdot 3 - 17 \cdot 1 = 125 \cdot 3 - 17 \cdot 22.$

Это означает, что

$17 \cdot (-22) = 1(\mathrm{mod}125).$

Найдем стандартное имя вычета (-22), *то есть вычета, который в сумме с* 22 *дает нуль. Легко видеть, что это* 103. *Значит,* $103 = 17^{-1}(\mathrm{mod}125)$. *Читатель может самостоятельно проверить, что произведение* $17 \cdot 103$ *дает при делении на* 125 *остаток* 1.

Зададим теперь вопрос: сколько всего существует обратимых вычетов $\mathrm{mod}n$? Это число зависит от n и выражается известной в теории чисел *функцией Эйлера* $\varphi(n)$.

Легко перечислить непосредственно все обратимые вычеты относительно небольших модулей. Например, mod 15 обратимы $1, 2, 4, 7, 8, 11, 13, 14$. Ясно, что $1 = 1^{-1}(\mathrm{mod}15)$, но также ясно, что $14 = 14^{-1}(\mathrm{mod}15)$. Действительно, $14 = (-1)(\mathrm{mod}15)$, а $(-1)^2 = 1$. Есть и другие квадраты, равные 1: $4^2 = 1(\mathrm{mod}15)$, $11^2 = 1(\mathrm{mod}15)$. Другие обратимые вычеты составят пары: $2 \cdot 8 = 1(\mathrm{mod}15), 7 \cdot 13 = 1(\mathrm{mod}15)$. Обратим внимание на то, что в кольце $\mathbf{Z}_{15}$ уравнение $x^2 = 1(\mathrm{mod}15)$ имеет в точности четыре решения. В одной из следующих лекций мы к этому (числу решений уравнения $x^2 = 1(\mathrm{mod}n)$)еще вернемся.

Мы видим, что $\varphi(15) = 8$. Ясно, что $\varphi(2) = 1$, $\varphi(3) = 2$, $\varphi(4) = 2, \varphi(5) = 4, \varphi(6) = 2, \varphi(7) = 6, \ldots$. Обратим внимание на тот факт, что для любого простого числа p имеем $\varphi(p) = p - 1$. Действительно, число p не имеет других делителей, кроме 1 и самого себя. Если $0 < k < p$, то k не может делиться на p, значит, нод$(k, p) = 1$. Как было доказано в предложении 2, это означает обратимость k. Обратимы все ненулевые вычеты mod$p$. Коммутативное ассоциативное кольцо с 1, в котором обратимы все ненулевые элементы, называется *полем*. Полям (в основном конечным) будет посвящена лекция 8.

В дальнейшем (для построения системы RSA в лекции 10) нам потребуется точное значение функции Эйлера $\varphi(n)$ для случая, когда $n = pq$ есть произведение двух различных простых чисел:

$$\varphi(n) = (p - 1)(q - 1). \tag{2.4}$$

Вычисление производим следующим образом:

1) выписываем все числа от 1 до $n - 1$;

2) отмечаем все числа, делящиеся на p :

$$p, 2p, ..., (q - 1)p,$$

их в точности $q - 1$;

3) отмечаем все числа, делящиеся на q :

$$q, 2q, ..., (p - 1)q,$$

их в точности $p - 1$;

4) замечаем, что среди отмеченных в 2), 3) чисел нет одинаковых и все не взаимно простые с n числа (меньшие, чем n) отмечены указанным способом;

5) подсчитаем, сколько осталось чисел, взаимно простых с n :

$$(pq - 1) - (q - 1) - (p - 1) = (p - 1)(q - 1),$$

что устанавливает равенство (2.4).

3. Китайская теорема об остатках[1]

Теорема 4. *Предположим, что целые числа* $n_1, n_2, \ldots, n_k \geq 2$ *попарно взаимно просты, то есть для любых* $i \neq j$ *нод*$(n_i, n_j) = 1$. *Тогда система сравнений*

$$\begin{cases} x & = & b_1 (\mathrm{mod} n_1), \\ x & = & b_2 (\mathrm{mod} n_2), \\ & \ldots & , \\ x & = & b_k (\mathrm{mod} n_k) \end{cases} \tag{3.1}$$

всегда имеет решение, то есть существует целое число x, *удовлетворяющее системе (3.1). Более того, решением является любое целое число* y, *сравнимое с* x: $y = x (\mathrm{mod} N)$, *где* $N = n_1 n_2 \ldots n_k$, *и других решений система (3.1) не имеет.*

Доказательство. Полагаем для каждого $i = 1, 2, \ldots, k$

$$N_i = n_1 n_2 \ldots n_{i-1} n_{i+1} \ldots n_k = N/n_i. \tag{3.2}$$

В силу попарной взаимной простоты модулей $n_1, n_2, \ldots, n_k$ число N_i оказывается взаимно простым с $n_i (i = 1, \ldots, k)$. Как установлено в предложении 2, в этом случае N_i обратимо $\mathrm{mod} n_i$, то есть существует такое число $M_i \in \mathbf{Z}$, для которого $N_i M_i = 1 (\mathrm{mod} n_i)$.

Мы утверждаем, что число

$$x = b_1 M_1 N_1 + b_2 M_2 N_2 + \ldots + b_k M_k N_k \tag{3.3}$$

является решением системы (3.1). Действительно, произведем вычисления $\mathrm{mod} n_i$. В этом случае все числа $b_l M_l N_l$ для $l \neq i$ удовлетворяют сравнениям $b_l M_l N_l = 0 (\mathrm{mod} n_i)$, потому что тогда $N_l \vdots n_i$. В то же время по нашему выбору выполняется сравнение $M_i N_i = 1 (\mathrm{mod} n_i)$, а вместе с ним и сравнение $b_i M_i N_i = b_i (\mathrm{mod}$

[1] Считается, что Китайская теорема об остатках была впервые сформулирована и доказана китайским математиком Сунь-Цзе в 100 г. до н. э.

n_i). Значит, $x = b_i(\mathrm{mod}\, n_i)$ для любого $i = 1, 2, \ldots, k$, что и требовалось.

Если $y = x(\mathrm{mod}N)$, то $y = x + qN$ для некоторого $q \in \mathbf{Z}$. Поскольку $N = 0(\mathrm{mod}n_i)$, получаем сравнение $y = x = b_i(\mathrm{mod}\, n_i)$, что верно для любого $i = 1, 2, \ldots, k$. Значит, y также является решением (3.1).

Пусть теперь x, y – два решения системы (3.1). Так как $x = y = b_i(\mathrm{mod}\, n_i)$, что равносильно делимости $x - y \vdots n_i$ для любого i, мы получаем делимость $x - y \vdots N$, поскольку N – произведение попарно взаимно простых чисел $n_1, \ldots, n_k$, а это означает, что $x = y(\mathrm{mod}N)$, что завершает доказательство теоремы.

Из доказательства теоремы легко вывести алгоритм нахождения всех решений системы (3.1).

Пример 5. *Пусть дана система*

$$\begin{cases} x = 5(\mathrm{mod}11), \\ x = 4(\mathrm{mod}13), \\ x = 3(\mathrm{mod}24). \end{cases}$$

В обозначениях из доказательства теоремы 4 имеем $b_1 = 5$, $b_2 = 4, b_3 = 3, n_1 = 11, n_2 = 13, n_3 = 24$.

Далее вычисляем: $N = n_1 n_2 n_3 = 3432$, $N_1 = n_2 n_3 = 312$, $N_2 = n_1 n_2 = 264$, $N_3 = n_1 n_2 = 143$. *Числа* M_1, M_2, M_3 *вычислим, используя обобщенный алгоритм Эвклида.*

$$\begin{array}{c} 312 = 11 \cdot 28 + 4, \\ \downarrow \quad 11 = 4 \cdot 2 + 3, \\ 4 = 3 \cdot 1 + 1. \end{array}$$

Далее:

$$\uparrow 1 = 4 - 3 \cdot 1 = 4 - (11 - 4 \cdot 2) \cdot 1 = 4 \cdot 3 - 11 \cdot 1 =$$

$$= (312 - 11 \cdot 28) \cdot 3 - 11 \cdot 1 = 312 \cdot 3 - 11 \cdot 85,$$

следовательно, $312 \cdot 3 = 1(\mathrm{mod}11)$, $312^{-1} = 3(\mathrm{mod}11)$, $M_1 = 3$;

$$\downarrow \begin{array}{c} 264 = 13 \cdot 20 + 4, \\ 13 = 4 \cdot 3 + 1, \end{array}$$

поэтому

$$\uparrow 1 = 13 - 4 \cdot 3 = 13 - (264 - 13 \cdot 20) \cdot 3 = -264 \cdot 3 + 13 \cdot 61,$$

следовательно, $264 \cdot 10 = 1(\mathrm{mod}13)$, $264^{-1} = 10(\mathrm{mod}13)$, $M_2 = 10$;

$$\downarrow \begin{array}{c} 143 = 24 \cdot 5 + 23, \\ 24 = 23 \cdot 1 + 1, \end{array}$$

откуда

$$\uparrow 1 = 24 - 23 \cdot 1 = 24 - (143 - 24 \cdot 5) \cdot 1 = 24 \cdot 6 - 143 \cdot 1,$$

следовательно, $143 \cdot 23 = 1(\mathrm{mod}24)$, $143^{-1} = 23(\mathrm{mod}24)$, $M_3 = 23$.

Решением рассматриваемой системы сравнений является, например,

$$x = 5 \cdot 312 \cdot 3 + 4 \cdot 264 \cdot 10 + 3 \cdot 143 \cdot 23 = 4680 + 10560 + 9867 = 25107.$$

Задачи и упражнения

Задача 2.1. Используя обобщенный алгоритм Эвклида, найти

$$19^{-1}(\mathrm{mod}26), 131^{-1}(\mathrm{mod}834).$$

Задача 2.2. Используя лишь калькулятор, найти число x, удовлетворяющее уравнению

$$3^x = 5(\mathrm{mod}p),$$

где $p - 1 = 2 \cdot 3 \cdot 101 \cdot 103 \cdot 107^2$.
Упрощенный вариант: $p - 1 = 2 \cdot 3 \cdot 101$, или еще проще: $p - 1 = 2 \cdot 3 \cdot 11$.

Задача 2.3. По Китайской теореме об остатках вычислить x (найти общее решение) для следующей системы сравнений:

$$\begin{cases} x = 5(\mathrm{mod}7), \\ x = 3(\mathrm{mod}11), \\ x = 10(\mathrm{mod}13). \end{cases}$$

Задача 2.4. Используя Китайскую теорему об остатках, вычислить x (найти общее решение) для следующей системы сравнений:

$$\begin{cases} x = 1(\mathrm{mod}15), \\ x = 3(\mathrm{mod}17), \\ x = 3(\mathrm{mod}24), \\ x = 4(\mathrm{mod}19). \end{cases}$$

Существует ли решение, если вместо 19 взять в качестве последнего модуля 18?

Задача 2.5. Сколько существует обратимых матриц порядка 2×2 над кольцом $\mathbf{Z}_6$? Над кольцом $\mathbf{Z}_8$?

Элементы шифрования

Штирлиц достал из книжного шкафа
томик Монтеня, перевел цифры в слова и
соотнес эти слова с кодом, скрытым среди мудрых истин
великого и спокойного французского мыслителя.
Ю. Семенов. Семнадцать мгновений весны

1. Основные понятия

Криптография есть изучение методов передачи текстов в скрытом виде таким образом, что понять содержание текста может только тот получатель, которому этот текст предназначен.

Прежде всего мы должны указать, как записываются тексты, то есть выбрать соответствующую платформу. Этому была посвящена лекция 1, где были введены понятия алфавита, буквы, единицы текста, блока. Было отмечено, что лучше всего работать с оцифрованными текстами, когда единицы текста заменяются на элементы некоторой числовой системы. Например, вместо обычного английского алфавита из 26 букв можно рассмотреть вычеты mod 26. Вместо «cake» мы запишем в этом случае $2, 0, 10, 4$. В русском варианте вместо «поэт» запишем $16, 15, 30, 19$ и т. п.

Кроме возможности записывать тексты, нам необходимо выбрать способ переработки текста в зашифрованный текст. Этот способ может быть представлен как ***функция шифрования***, или ***алгоритм шифрования***. Рассмотрим его более подробно.

2. Алгоритм (функция) шифрования

Пусть m – исходный текст. Возможно также, что m – единица текста или его блок. Функция шифрования E перерабатывает m в шифрованный текст (единицу текста, блок) $c = E(m)$.

Основное правило криптографии гласит: исходный текст должен однозначно восстанавливаться по шифрованному тексту.

Значит, функция E должна быть обратимой. При этом обратить ее может только тот, кому это позволено, – легитимный пользователь. Значит, должен быть какой-то секрет, знание которого позволяет обращать E и узнать который нелегко. Обычно функция E выбирается из множества функций шифрования, определяемого выбранной системой шифрования. Чтобы получить конкретный шифр, нужно выбрать ***ключ шифрования*** e (от слова encryption – шифрование). При этом должен существовать ***ключ дешифрования*** d (от слова decryption – дешифрование).

Если кто-то пытается восстановить исходный текст, не имея на это прав (он не знает ключа d), то его попытки называются *взломом*, или *раскалыванием* текста. Взломщик может пытаться взломать всю систему, узнать все секретные данные – это *взлом системы*. *Криптостойкостью* системы шифрования называется ее защищенность от возможного взлома. Криптостойкость зависит как от правильности выбора способа шифрования, так и от правильности выбора необходимых параметров, в частности ключей.

Функция E, используемая для шифрования, должна быть ***односторонней***. Последнее означает, что вычисление $c=E(m)$ должно быть достаточно простой операцией, а нахождение m по c – трудным делом. Однако необходимость дешифровки для пользователя, которому шифровка предназначена, приводит к более сложному понятию ***односторонней функции с секретом***. Это означает, что в общем случае обратить функцию E (научиться вычислять m по $c = E(m)$) трудно, но при знании соответствующего *секрета* (ключа дешифрования d) легко.

Что в криптографии означает «легко» и что «трудно»? «Легко» означает, что соответствующую операцию компьютер выполняет за разумное время, а «трудно» – что он не может ее выполнить за «реальное» время. Разумность и реальность, в свою очередь, зависят от важности задачи, мощности используемых процессоров и т. п. Эти понятия со временем меняются.

Перейдем к примерам, в которых используются оцифровки русского и английского алфавитов из лекции 1.

3. Шифр замены

По сути дела, в наше время это не шифр в полном смысле слова, а именно элемент шифрования, включаемый в системы шифрования.

Пусть A – некоторый алфавит, состоящий из n букв, $E_\sigma : A \longrightarrow A$ – взаимно-однозначная функция замены, определяемая подстановкой

$$\sigma = \begin{pmatrix} 0 & & n-1 \\ \sigma(0) & \dots & \sigma(n-1) \end{pmatrix}.$$

Здесь $\sigma(0), \sigma(1), \dots, \sigma(n-1)$ – перестановка чисел $0, 1, \dots, n-1$. То есть функция E_σ заменяет в тексте букву с номером i на букву с номером $\sigma(i)$.

Например, подстановка вида $\sigma(i) = i + 13 (\mathrm{mod} 26)$ задает для текстов, записанных с помощью стандартной нумерации английского алфавита, шифр сдвига. Этот шифр переводит слово algorithm $\longleftrightarrow$ $0, 11, 6, 14, 17, 8, 19, 7, 12$ в слово $13, 24, 19, 1, 4, 21, 6, 20, 25 \longleftrightarrow$ nytbevguz.

В общем случае шифр сдвига задается функцией шифрования.

$$c = i + e (\mathrm{mod} n),$$

где e – некоторая константа (ключ шифрования). Дешифровка определяется функцией

$$i = c + d (\mathrm{mod} n),$$

где $d = n - e$.

В общем случае шифр замены (часто говорят «шифр простой замены»), соответствующий подстановке σ, задается функцией

$$c = \sigma(i),$$

дешифровка осуществляется функцией

$$i = \sigma^{-1}(c),$$

где σ^{-1} – обратная к σ подстановка.

Шифр простой замены весьма прост и легко поддается взлому методом статистического анализа, о котором говорится в лекции 4.

Другую возможность использовать в качестве ключа шифрования подстановку предоставляет шифр перестановки, изменяющий не знаки, а места, на которых они расположены.

4. Шифр перестановки

Это блочный шифр. Исходный текст разбивается на блоки одинаковой длины l. Выбирается случайная подстановка

$$\tau = \begin{pmatrix} 1 & 2 & \dots & l \\ \tau(1) & \tau(2) & \dots & \tau(l) \end{pmatrix}.$$

Она действует на каждый блок длины l, переставляя его символы (не изменяя их). Например, подстановка $\tau = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix}$ представляет текст *abcd* как *cadb*.

Криптостойкость шифра перестановки невелика.

5. Шифр Вернама и гаммирование

Шифр Вернама (Vernam) работает с исходным текстом, представленным в виде бинарной последовательности $m = m_1 m_2 \dots m_l$. В качестве ключа шифрования используется бинарная последовательность $k = k_1 k_2 \dots k_l$ той же самой (или большей) длины. Шифрованный текст представляется в виде бинарной последовательности $c = c_1 c_2 \dots c_l$ той же самой длины, где процесс шифрования определяется операцией побитного сложения $\bmod 2$:

$$c_i = m_i \oplus k_i, i = 1, 2, \dots, l. \tag{5.1}$$

Эта операция называется XOR.

Более общим является шифрование методом гаммирования $\mathrm{mod}n$. Обычно n – мощность алфавита, буквы которого занумерованы вычетами $0, \ldots, n-1$. Исходный текст $m = m_1 m_2 \ldots m_l$ представлен вычетами $m_i(\mathrm{mod}n)$, $i = 1, \ldots, l$. Ключ также представляет собой последовательность вычетов $\mathrm{mod}n$ той же (или большей) длины: $k = k_1 k_2 \ldots k_l$. Шифрование, результатом которого является последовательность $c = c_1 c_2 \ldots c_l$ вычетов $\mathrm{mod}n$, осуществляется по правилу

$$c_i = m_i + k_i(\mathrm{mod}n), i = 1, \ldots, l. \tag{5.2}$$

Формула (5.1) есть частный случай формулы (5.2) для $n = 2$.

Ясно, что дешифрование шифрованного текста c осуществляется по формуле

$$m_i = c_i - k_i(\mathrm{mod}n), i = 1, 2, \ldots, l, \tag{5.3}$$

причем в случае $n = 2$ получается формула

$$m_i = c_i \oplus k_i, i = 1, 2, ..., l. \tag{5.4}$$

Если ключ k представляет собой случайную последовательность и используется один раз, то говорят о шифровании *методом одноразового блокнота*. Такое шифрование является абсолютно надежным. Таким образом, из текста m можно получить подходящим выбором ключа любой(!) текст той же самой длины. Любая частичная дешифровка текста (определение части ключа) нисколько не помогает при расшифровке остального текста (определению остатка ключа). Основной недостаток данного метода шифрования – его громоздкость, необходимость предварительного секретного разделения ключа, надежного его хранения, невозможность использования в массовых системах, одноразовость.

Гаммированием называется метод шифрования, который внешне совпадает с формулой (5.2), но при этом ключ k генерируется. Ключ k называется *гаммой*. Поскольку генерирование является

искусственным процессом и осуществляется некоторым алгоритмом, к нему предъявляется ряд требований: выпускная последовательность, играющая роль ключа k, должна быть псевдослучайной, обладать хорошими статистическими свойствами, в случае ее периодичности – иметь большой период и т.п. О некоторых методах генерирования псевдослучайных последовательностей разговор пойдет в лекциях 13 и 14.

6. Четыре возможных режима использования блочных шифров

Многие современные системы шифрования предоставляют блочные шифры. При этом возникает вопрос об их правильном использовании. Давно замечено, что блоки необходимо каким-то образом «сцеплять». Эта идея может быть реализована самыми разными способами. Некоторые из них будут представлены ниже.

Произвольный блочный шифр может быть использован в различных режимах. В США был введен федеральный стандарт, предписывающий использование стандарта шифрования DES (см. приложение 1) в одном из следующих режимов: ECB, CBC, OFB, CFB. Подобные режимы применимы также для других блочных шифров. Рассмотрим их подробнее.

Режим ECB (Electronic Code Book)

Простейший среди всех режимов. Исходный текст m делится на блоки $m_1, m_2, \ldots, m_q$. При необходимости последний из блоков m_q дополняется до полного блока одним из обусловленных в каждом конкретном случае способов. Далее к каждому блоку m_i последовательно применяется шифрование с заданным ключом k: $c_i = E_k(m_i)$. Результатом является разбитый на блоки шифротекст

$$c = c_1 c_2 c_3 \ldots c_q.$$

Режим ECB имеет ряд недостатков. Во-первых, злоумышленник может удалить или добавить какой-нибудь блок c_t или переставить два таких блока.

Могут появиться два одинаковых блока c_i, c_j $(i < j)$, соответствующих одинаковым блокам m_i, m_j исходного текста.

Необходимо предусмотреть защиту от подобных атак.

Режим CBC (Cipher Block Chaining)

При таком режиме в исходный текст, разбитый на блоки $m = m_1 m_2 \dots m_q$, дополняется фиктивный блок $\tilde{m_1}$, и шифрование осуществляется следующим образом:

$c_1 = E_k(m_1 \oplus \tilde{m_1})$, $c_2 = E_k(m_2 \oplus c_1)$, $\dots$, $c_q = E_k(m_q \oplus c_{q-1})$.

При таком методе шифрования ошибка, допущенная в блоке с номером j, распространяется не только на этот блок, но и на все последующие. Одинаковые блоки исходного текста шифруются по-разному.

Режим OFB (Output Feedback)

Основная задача этого режима – адаптировать данный блочный шифр к его поточному использованию. Для этого при размере одного блока n выбирается переменная j $(1 \le j \le n)$. С помощью блочного шифра создается поток ключей, j бит за один раз. Как и прежде, мы разбиваем исходный текст $m = m_1 m_2 \dots m_q$, но на этот раз блоки состоят из j бит. Вначале переменной x_1 присваивается значение $\tilde{m_1}$, а затем вычисляется $y_1 = E_k(x_1)$, $y_2 = E_k(x_2), \dots$; далее полагается $c_i = m_i \oplus e_i$, где e_i – j крайних слева бит для y_j, причем $x_{i+1} = y_i$.

Режим CFB (Cipher Feedback)

Этот режим похож на предыдущий и осуществляется по следующей схеме:

$y_0 = \tilde{m_1}$, $z_i = E_k(y_{i-1})$, $e_i = j$ крайних слева бит блока z_i, $y_i = m_i \oplus e_i$.

Задачи и упражнения

Задача 3.1. Пусть алфавит состоит из 26 букв английского языка. Шифрование осуществляется при помощи подстановки

$$s = (p, h, u, q, l, f, e, r, m, g, y, x, v, s, n, i, b, o, j, c, w, t)(a, k, d)(z).$$

Здесь подстановка определяется циклом, в котором буквы переходят в следующие по написанию, а последняя буква переходит в первую (цикл замыкается). Буква z остается на месте.

Пример: исходный текст – *I am fine* = *iamfine* (мы не учитываем пробелы и разницу между строчными и прописными буквами, а также знаки препинания), шифрованный текст *bkgebir*.

1) Зашифровать текст:

Some people say a man is made out of mud A poor mans made out of muscle and blood Muscle and blood and skin and bone A mind thats weak and the back thats strong Sixteen tons.

2) Расшифровать текст:

btkiarmpumjqyurkwuwukmprmranpmrrpirdmtur
mrpurwukmprmrapukgrnajrnefjtkgkmdbirsrmxek
wrbgrrpgkmdnjetrkdrirnngkmdnjetjrofkdrfjiaji.

Задача 3.2. Алфавит русский (33 буквы). Исходный текст разбит на блоки величины $n = 8$. Буквы внутри каждого блока меняются местами перестановкой

$$s = (1, 8, 2, 7)(4, 6, 5, 3).$$

1) Зашифровать текст:

Скучно на этом свете господа Гоголь.

2) Расшифровать текст:

алмрпотуевиидкчьдубъуичиьнслтактишыувбгазнваонинийсекнсе.

Задача 3.3. Зашифровать текст с помощью гаммирования, переводя все его буквы в бинарные последовательности длины 5:

hidden number is a key = 7 8 3 3 4 13 ....

Выбрать в качестве ключа свой индивидуальный номер, записанный в двоичном исчислении. Индивидуальный номер равен сумме значений букв фамилии (русский язык, 33 буквы, стандартная нумерация).

Задача 3.4. Назовем шифр замены, отвечающий подстановке $\sigma \in \mathbf{S}_n$, *гомоморфным*, если он удовлетворяет тождеству

$$\forall i, j \in \mathbf{Z}_n \ \sigma(i+j) = \sigma(i) + \sigma(j) (\mathrm{mod} n).$$

Сколько существует гомоморфных шифров замены при фиксированном n? Как они устроены?

Лекция 4

Элементы криптоанализа

1. Основные понятия

Криптоанализ – это наука о надежности систем шифрования. В криптоанализе исследуются различные возможности получения скрытой информации. Изучаются возможные способы несанкционированного восстановления исходного текста по его шифровке, нахождения тем или иным способом секретных параметров криптосистемы, вычисление секретного ключа и т. п. В криптоанализе также рассматриваются вопросы возможности подлога, подмены, подделки и т. п. Ценность представляет любая, даже частичная или вероятностная информация.

Попытки взлома или получения какой-либо скрытой информации называются *атаками*. Успешные атаки всегда основаны на предварительном знании. Обычно предполагается, что сама система шифрования известна. Неизвестны только ее секретные параметры и ключи.

Выделим некоторые типы атак, основываясь на возможности взломщика добывать предварительную информацию:

1. Известны соответствующие друг другу исходный текст m и шифрованный текст c: $E_e(m) = c$. Необходимо найти ключ шифрования e или ключ дешифрования d.

Говорят, что это анализ *с полной информацией*.

2. Известен только шифрованный текст c. Требуется прочитать m, найти ключ шифрования e или ключ дешифрования d.

Такая атака называется *пассивной*.

3. Взломщик имеет возможность посылать шифрованные тексты и получать на них ответы. По этим данным он пытается восстановить секретные параметры шифра.

Это – *активная атака.*

Следующий метод является конечно универсальным, но применяется достаточно редко. Это объясняется тем, что хорошие шифры не позволяют производить его за реальное время.

2. Полный перебор

Рассмотрим некоторые возможности взломщика. Допустим, он знает исходный текст m и шифрованный текст $c = E_e(m)$. Предположим, что существует единственный ключ шифрования e, для которого выполнено равенство $c = E_e(m)$. Взломщик может использовать алгоритм полного перебора ключей e. Упорядочив все ключи каким-либо способом: $e_1, e_2, \ldots, e_t$, он перебирает в качестве e последовательно $e_1, e_2, \ldots, e_t$ до тех пор пока не получит равенство $c = E_{e_p}(m)$. Это означает, что найден ключ шифрования $e = e_p$.

Пусть взломщику известен только шифрованный текст c. Он может пытаться найти ключ дешифрования d, перебирая все возможности $d_1, \ldots, d_r$, до тех пор пока не получится такой текст $m = E_{d_p}(c)$, который, во-первых, имеет смысл, а во-вторых, по этому смыслу становится ясно, что это как раз искомый текст. Мы уже видели, что в случае применения алгоритма одноразового блокнота такая атака не имеет шансов на успех.

Недостаток подобных алгоритмов полного перебора состоит в том, что при правильном шифровании такой перебор занимает слишком много времени (сейчас мы, конечно, говорим о компьютерном времени) или же стоит слишком дорого. В современной криптографии речь идет о предельных возможностях. Бывают настолько важные задачи, что даже параллельная работа десятков тысяч мощных процессоров в течение нескольких месяцев может считаться реальной.

3. Частотный анализ

Одним из эффективных методов криптоанализа является *статистический метод* исследования. Возможность применения та-

кого метода считается слабостью используемой криптосистемы. Например, описанный в п. 3 лекции 3 метод замены сохраняет частоты букв используемого алфавита. Эти частоты для разных букв алфавита могут существенно отличаться. Приведем стандартные частоты, с которыми буквы русского и английского алфавитов используются в достаточно длинных литературных текстах. В качестве эталона для английского языка обычно берется классическое произведение Мелвилла «Моби Дик». Разные исследователи частот изучали различные источники, поэтому соответствующие данные могут отличаться, но в общем случае эти отличия незначительны. Далее относительные частоты указаны в процентах:

РУССКИЙ[2]

а(7,96) б(1,67) в(4,71) г(1,87) д(3,07) е(8,90) ё(0,11) ж(1,18) з(1,74) и(6,38) й(0,98) к(3,25) л(4,64) м(3,13) н(6,70) о(11,26) п(2,71) р(3,90) с(5,32) т(6,31) у(2,63) ф(0,18) х(0,73) ц(0,29) ч(1,91) ш(0,82) щ(0,29) ъ(0,03) ы(1,73) ь(2,25) э(0,36) ю(0,60) я(2,39).

АНГЛИЙСКИЙ

a(8,2) b(1,5) c(2,8) d(4,2) e(12,7) f(2,2) g(2,0) h(6,1) i(7,0) j(0,1) k(0,8) l(4,0) m(2,4) n(6,7) o(7,5) p(1,9) q(0,1) r(6,0) s(6,3) t(9,0) u(2,8) v(1,0) w(2,4) x(0,1) y(2,0) z(0,1).

Если в английском шифрованном тексте чаще всего встречается буква j, хотя известно, что шифрование произведено с сохранением частот (например, с помощью замены), то, скорее всего, под j скрывается e, a, i, t, возможно, буква с еще меньшей относительной частотой, но никак не z, q и т. п. При анализе следует

[2]Эти частоты вычислены студентами математического факультета ОмГУ Антоном Розвезевым и Натальей Гафаровой. Анализировался роман Ф.М. Достоевского «Идиот» Суммы частот, приведенные в ряде отечественных книг по криптографии, почему-то не равны 100.

обращать внимание прежде всего на наиболее и наименее частые буквы и пытаться подставлять буквы с похожими стандартными частотами.

Если в текстах используется знак пробела, то он может оказаться наиболее частым, причем разница с частотой появления следующего знака может оказаться значительной. По этой причине тексты часто пишут без пробелов. Можно использовать разные знаки для обозначения пробела, есть и другие приемы. Однако в хороших современных шифрах все устроено гораздо лучше и проблемы такого рода уже не возникают.

Статистический анализ современных шифров гораздо более тонкий. Как правило, он не позволяет сразу получать полную информацию о системе.

Пример 6. *Допустим, что у нас имеется шифровка, полученная шифром простой замены в алфавите русского языка из 32 букв (е не отличается от ё):*

я х ш б ю ф ы л е г к г ы а я е ю х ъ х у б о б щ м н к а к г и я к ф к н к я у к ь ь ш б э я э щ я к э б г б я б н к ь ф б н к ь ю б я э щ у к ь я э и в
ь г к и ф к р к ь щ щ о у к а к г к г м н к я и ю к ф к ш б у и и у х ъ г к ш б у и ш у к с щ щ о у к а г к г м н к б ъ ю к ф к ш б г э х я х р ю о у б ч ф щ и м к с ь б щ м ы ф щ ф е щ м н и р а к о к м б г ш б у и н б ь я б э о ш б у и ы ь ш и о ъ щ а и ю ф б в ф щ ц ф б ь я ы ш э ы ш к м б г ш э щ м н б ь р ы д щ ф ы щ ь ю щ э я ф х р ю и г.

Текст содержит 251 знак. Выпишем те буквы, которые повторяются чаще всего:
к – 31, б – 23, щ – 17, ф – 13, я – 13, ь – 12, ш – 11, ы – 7
и сравним их с наиболее частыми буквами русского языка: о, е, а, и, н, т, с.

Рассмотрев несколько вариантов (полагая, например, к ↔ о, б ↔ е, щ ↔ а, ф ↔ и, ..., а затем производя перестановки вроде

к ↔ а, б ↔ о, щ ↔ е, . . .), обнаруживаем, что при соответствии ъ ↔ с, б ↔ о, щ ↔ е, ы ↔ и, ф ↔ н в следующем тексте:

- - - о - н и - - - а - и - т - - - - - - о - о
е - - а - а - - т а н а - а т - а с
с - о - т - е т а - о - о т о - а
с н о - а с - о т - е - а с т - - - с - а -
н а н а с е е - - а - а - а - - - а т -
- а н а - о - - - - - - - а - о - -
- - а - е е - - а - а - а - - - а
е - - а н а - о - - - т - - - - - о -
н е - - а - с о е - и н е н - е
- - - - - а - а - о - - о - -
- о с т о - - - о - - и с - - -
- е - - - н о - н е - н о с т и
- - и - а - о - - - е - - о с - и
- е н - е с - е - т н - - - - -

появляется кусок с о е (м)и н е н (е) е, что позволяет восстановить соответствие м ↔ д, е ↔ ъ. Также мы находим кусок н е (ц) н о с т и, что позволяет определить, что ц соответствует букве ж. Слово е д (н) а дает н ↔ в.
Далее нетрудно уже прочитать слова и сочетания «загадок», «ее глаза как два» и постепенно восстановить отрывок из знаменитого стихотворения Н. Заболоцкого:

Ты помнишь, как из тьмы былого,
Едва закутана в атлас,
С портрета Рокотова снова
Смотрела Струйская на нас?
Ее глаза – как два тумана,
Полуулыбка, полуплач,
Ее глаза – как два обмана,
Покрытых мглою неудач.
Соединенье двух загадок,
Полувосторг, полуиспуг,

Безумной нежности припадок,
Предвосхищенье смертных мук.

Конечно, этот пример не особо показателен, так как текст довольно короткий. Частотный анализ наиболее эффективен именно для расшифровки длинных текстов, когда отклонения частот появления букв от стандартных частот очень невелики. Расшифровывать указанным способом короткие тексты гораздо труднее.

Существуют самые разные методы криптоанализа. Приведем лишь краткий обзор некоторых из них.

4. Разностный криптоанализ

Этот метод предложен в 1990 г. Э. Бихэмом (E. Biham) и А. Шамиром (A. Shamir). Основным объектом применения метода является стандарт шифрования DES (см. приложение 1). Полного успеха метод не достиг. Обычный DES оказался неприступным. Однако определенные успехи в применении метода были получены при анализе некоторых упрощенных схем DES. Кроме того, были проведены эффективные атаки на ряд других криптоалгоритмов (Lucifer, FEAL, LOKI, Khufu), что позволяет говорить о достаточно широкой применимости метода. В настоящее время известно несколько модификаций разностного криптоанализа: *линейно-разностный анализ, анализ с использованием близких ключей, разностный анализ с ошибками.*

В настоящем курсе мы не приводим описания метода разностного криптоанализа. Сделать это нам не позволяет его громоздкость и необходимость приведения ряда важных понятий и результатов из теории вероятностей. Указанному методу посвящена специальная литература. См. по этому поводу [24].

5. Линейный криптоанализ

Этот метод введен в обращение в 1993 г. японским криптологом М. Матсуи (M. Matsui). Первоначально он подобно разностному криптоанализу был ориентирован на DES. Метод предусмат-

ривает возможность для взломщика генерировать случайные тексты и получать их шифровки, что соответствует активной атаке. Изложение метода заняло бы слишком много места, поэтому, как и в предыдущем пункте, мы отсылаем читателя к специальной литературе.

Известны успешные атаки, проведенные данным методом для обнаружения ключа DES. Например, сам М. Матсуи в 1996 г. программой около 1000 строк, написанной на языках *C* и Ассемблер, провел параллельные вычисления на 12 мощных процессорах. Через 50 суток он вычислил ключ DES.

Задачи и упражнения

Задача 4.1. Пусть некоторый английский текст зашифрован методом простой замены. Предположим, что определено значение, скажем 14, буквы q. Пусть в тексте содержится сочетание 14, 18. Чему скорее всего соответствует 18?

Задача 4.2. Пусть известны шифровки $c_1 = E_{\tau_1}(m), c_2 = E_{\tau_2}(m)$ одного и того же достаточно длинного текста m, полученные шифрами перестановки с одинаковой длиной l блока: $\tau_1, \tau_2 \in \in \mathbf{S}_l$.

Как можно с помощью этой информации эффективно найти l?

Задача 4.3. Пусть имеется исходный текст m и его шифровка, полученная с помощью шифра замены: $c_1 = E_{\sigma_1(m)}$, $\sigma_1 \in \mathbf{S}_n$.

Найти m можно частотным анализом или еще каким-либо способом. Вопрос в том, облегчает ли задачу знание еще одной шифровки того же текста: $c_2 = E_{\sigma_2}(m)$, $\sigma_2 \in \mathbf{S}_n$?

Задача 4.4. Допустим, используется английский алфавит из 26 букв со стандартной нумерацией. Пусть задан гомоморфный шифр замены (см. определение в условии задачи 3.4), соответствующий подстановке $\sigma \in \mathbf{S}_{26}$. Можно ли полностью восстановить σ, зная $\sigma(b)$? Тот же вопрос для $\sigma(c)$. Что можно сказать по этому поводу в общем случае?

Задача 4.5. Допустим, что вместо буквы "A" английского алфавита, частота которой 8, 2%, используется 82 различных знака с равной вероятностью. Вместо "J" частоты 0, 1% – 1 знак и т.д. Таким образом в алфавите появляется не 26, а 260 букв. Но теперь они встречаются в тексте с приблизительно равной вероятностью. К такому тексту применяется простая замена. Покажите, что в этом случае частотный анализ по-прежнему эффективен, только применять его нужно по другому. Каким образом?

Модели систем шифрования

1. Вероятностная модель К. Шеннона

В конце 40-х гг. XX в. К. Шеннон (C. Shannon) опубликовал свои знаменитые работы (см. [14]), в которых он не только положил начало и сформировал основы всей информационной науки, но также представил методологию исследования систем шифрования. Опишем в общих чертах одну из наиболее известных его моделей.

Пусть система шифрования $\mathcal{S}$ построена на некотором конечном алфавите или даже нескольких алфавитах, с которыми связаны следующие множества:

M – множество исходных текстов,

C – множество всех возможных шифрованных текстов,

K – множество всех возможных ключей.

Мы не вдаемся в природу этих множеств. Будем лишь предполагать, что все они конечны.

Допустим, что все указанные множества являются вероятностными пространствами. Это означает, что для любых элементов $m \in M, c \in C, k \in K$ определены вероятности $0 \leq p(m), p(c), p(k) \leq 1$, позволяющие наделить множества M, C, K функциями распределения вероятностей с обычными свойствами. Наличие таких распределений позволяет говорить о случайном выборе элементов или подмножеств в рассматриваемых множествах. Будем называть соответствующее вероятностное пространство *регулярным*, если вероятность любого его элемента отлична от нуля.

Далее предполагается, что для любого ключа $k \in K$ определены функция шифрования

$$E_k : M \to C$$

и функция дешифрования

$$D_k : C \to M.$$

При любом k эти функции должны быть взаимно-обратны, то есть они должны удовлетворять тождествам:

$$E_k \circ D_k = id_M, \tag{1.1}$$

$$D_k \circ E_k = id_C. \tag{1.2}$$

В частности, при любом k функции E_k, D_k сюръективны.

Также определяются функции двух переменных

$$E(m,k) : M \times K \to C,$$

$$D(c,k) : C \times K \to M,$$

задаваемые равенствами

$$E(m,k) = E_k(m), D(c,k) = D_k(c). \tag{1.3}$$

Они связаны тождествами

$$D(E(m,k),k) = m, E(D(c,k),k) = c \tag{1.4}$$

и также сюръективны.

На основе этой базовой модели К. Шеннон развил теорию секретности, получившую в дальнейшем широкое распространение и дополненную последующими исследованиями. Мы затронем только некоторые аспекты этой теории.

Совершенная секретность

Система шифрования называется *совершенной*, если знание шифрованного текста не изменяет вероятностей исходных текстов. Формально это означает, что для любых $m \in M, c \in C$

$$p(m) = p(m|c). \tag{1.5}$$

Здесь, как обычно, $p(m|c)$ означает условную вероятность выбранного текста m при данной шифровке c.

Если определить совместную вероятность $p(m, c)$ и воспользоваться формулами Байеса

$$p(m, c) = p(c)p(m|c) = p(m)p(c|m), \tag{1.6}$$

то получим равносильное определению условие совершенной секретности

$$\forall m \forall c \; p(c) = p(c|m). \tag{1.7}$$

В совершенных системах с получением шифрованного текста c к априорной информации об исходном тексте – его вероятности $p(m)$ – ничего не добавляется. В частности, для любых двух текстов $m_1, m_2 \in M$ и любой шифровки $c \in C$ совпадают суммы вероятностей

$$\sum_{E(m_1,k)=c} p(k) = \sum_{E(m_2,k)=c} p(k). \tag{1.8}$$

Пусть задана система шифрования $\mathcal{S}$, в которой $M = \mathbf{Z}_n^t$, $n \geq 2, t \geq 1$ – пространство всех возможных текстов в алфавите $\mathbf{Z}_n$ фиксированной длины t. Считаем, что функция распределения вероятностей в $\mathcal{S}$ равномерна, то есть

$$\forall m \in M \; p(m) = \frac{1}{n^t}. \tag{1.9}$$

Аналогичным образом строим пространства

$$C = \mathbf{Z}_n^t, K = \mathbf{Z}_n^t.$$

Функции шифрования и дешифрования определяются следующим образом:

$$E_k(m) = m \oplus k(mod n), \tag{1.10}$$

$$D_k(c) = c \ominus k(mod n), \tag{1.11}$$

где сложение и вычитание наборов осуществляется покомпонентно.

Покажем, что такая система шифрования совершенна. Действительно,

$$p(m|c) = \sum_{E(m,k)=c} p(k) = p(k') = \frac{1}{n^t}, \tag{1.12}$$

где $k' = c \ominus m(mod n)$ – единственный ключ, дающий равенство

$$m \oplus k' = c(mod n). \tag{1.13}$$

Отсюда получаем равенство

$$p(m|c) = p(m) = \frac{1}{n^t}. \tag{1.14}$$

Теорема 7. *(Теорема Шеннона). Пусть S – регулярная система шифрования, в которой множества M, C, K имеют одинаковую мощность:*

$$|M| = |C| = |K| = n.$$

Тогда система S является совершенной в том и только том случае, если выполнены следующие условия:

1) для любых текстов $m \in M, c \in C$ существует единственный ключ $k \in K$ такой, что

$$E(m, k) = c;$$

2) распределение вероятностей на пространстве ключей K равномерно:

$$\forall k \in K \; p(k) = \frac{1}{n}.$$

Доказательство. Установим вначале необходимость выполнения условий 1) и 2).

Заметим, что при условии совершенности системы $\mathcal{S}$ для любых $m \in M, c \in C$ найдется единственный ключ $k \in K$ такой, что

$$E(m,k) = c. \tag{1.15}$$

Предположим, что для некоторых m, c такого k не существует. Тогда $p(c|m) = 0$, откуда и из совершенности системы следует, что $p(c) = 0$, что противоречит регулярности системы $\mathcal{S}$.

Если же для некоторого m два различных ключа k_1, k_2 дают одинаковую шифровку $c = E_{k_1}(m) = E_{k_2}(m)$, то в силу равенства $|K| = |C|$ найдется шифровка c', не получаемая из m никаким ключом, что противоречит только что доказанному утверждению.

Аналогично тому, как была выше доказана однозначная разрешимость уравнения (1.15) относительно k, устанавливается однозначная разрешимость этого уравнения относительно m. При этом вместо равенства $|K| = |C|$ нужно использовать равенство $|M| = |C|$.

Установим равномерность распределения вероятностей на пространстве K. Пусть $k_1, k_2 \in K$ – произвольная пара ключей. Зафиксируем любую шифровку $c \in C$. Тогда найдутся два единственным образом определяемых текста m_1, m_2 таких, что

$$c = E(m_1, k_1) = E(m_2, k_2). \tag{1.16}$$

Эти равенства выполняются при единственных k_1, k_2, поэтому

$$p(c) = p(c|m_1) = p(k_1) = p(c|m_2) = p(k_2), \tag{1.17}$$

следовательно, $p(k_1) = p(k_2) = \frac{1}{n}$.

Необходимость выполнения условий 1), 2) установлена.

Покажем их достаточность для совершенности системы $\mathcal{S}$. Для этого вычислим вероятность и условную вероятность относительно произвольных шифровки c и текста m.

$$p(c) = \sum_{E(m,k)=c} p(m)p(k) = \frac{1}{n}\sum p(m) = \frac{1}{n}, \tag{1.18}$$

$$p(c|m) = \sum_{E(m,k)=c} p(k) = \frac{1}{n}. \tag{1.19}$$

Они одинаковы.

Теорема доказана.

Прямоугольная таблица, строки которой соответствуют элементам множества M, а столбцы – элементам множества K, в то время как на пересечении строки m и столбца k стоит шифровка $c = E(m,k)$, называется *таблицей шифрования.*

Следствие 8. *В случае регулярной совершенной системы с условием $|M| = |C| = |K| = n$ таблица шифрования является латинским квадратом.*

Напомним, что латинский квадрат размером $n \times n$ по определению содержит в каждой строке и каждом столбце по одному представителю множества из n элементов.

2. Мера теоретической секретности

Количественной мерой информации является энтропия. Для любого дискретного вероятностного пространства Q с заданным распределением $p(q), q \in Q$ *энтропия* определяется как функция

$$H(Q) = -\sum_{q \in Q} p(q)log(p(q)), \tag{2.1}$$

где log берется по основанию 2. При $x = 0$ считается, что $xlogx = 0$.

В рассмотренной выше модели К. Шеннона мы можем говорить о функциях $H(M), H(C), H(K)$.

В совершенных регулярных системах шифрования при $|M| = |C| = |K| = n$ имеем

$$H(K) = log(n).$$

По аналогии с (2.1) определяются функции

$$H(M|c) = -\sum_{m \in M} p(m|c) log(p(m|c)), c \in C; \tag{2.2}$$

$$H(M|C) = \sum_{c \in C} p(c) H(M|c) = -\sum_{m \in M, c \in C} p(c) p(m|c) log(p(m|c)) =$$

$$= -\sum_{m \in M, c \in C} p(m, c) log(p(m|c)). \tag{2.3}$$

Мы видим, что значение $H(M|c)$ показывает среднюю неопределенность исходного текста при известной шифровке c. Можно также определить функции $H(K|c), c \in C, H(K|C)$ с естественным трактованием.

Важное значение имеет функция

$$I(K|C) = H(K) - H(K|C), \tag{2.4}$$

выражающая среднее количество дополнительной информации о ключах, получаемой из шифровок. Также часто рассматривается функция

$$D(M) = log(|M|) - H(M), \tag{2.5}$$

показывающая разницу между случаем равномерного распределения в пространстве M, когда $H(M) = log(n)$, и рассматриваемым случаем. Величину $D(M)$ называют ***избыточностью языка*** M.

Предположим, что действительные числа $a_i > 0$ $(i = 1, ..., n)$ таковы, что $\sum_{i=1}^{n} a_i = 1$. Тогда для любого набора чисел $x_i > 0$ $(i = 1, ..., n)$ имеет место хорошо известное

Неравенство Йенсена

$$\sum_{i=1}^{n} a_i log(x_i) \leq log(\sum_{i=1}^{n} a_i x_i), \tag{2.6}$$

равенство в котором достигается тогда и только тогда, когда $x_1 = ... = x_n$.

Теорема 9. *Пусть Q - дискретное вероятностное пространство с исходами $q_1, ..., q_n$ и функцией вероятности p. Тогда справедливо неравенство*

$$0 \leq H(Q) \leq log(n).$$

Очевидно, что нижняя граница неравенства достигается на случайной величине, принимающей одно из своих значений с вероятностью 1. *Верхняя граница достигается на случайной величине, распределенной равномерно.*

Доказательство. Вычисляем

$$0 \leq H(Q) = -\sum_{i=1}^{n} p(q_i) log(p(q_i)) =$$

$$= \sum_{i=1}^{n} p(q_i) log(\frac{1}{p(q_i)}) \leq$$

$$\leq log(\sum_{i=1}^{n} (p(q_i) \frac{1}{p(q_i)})) = log(n).$$

Неравенство в преобразованиях следует из неравенства Йенсена.

Теорема 10. *(Теорема Шеннона). Предположим, что* $|M| = |C|$. *Тогда*

$$I(K, C) \leq D(M).$$

Другими словами, количество информации о ключе в шифровке в среднем не превосходит избыточности языка сообщений.

Доказательство. В самом деле,

$$H(C) \leq log(|C|) = log(|M|),$$

поэтому

$$H(K|C) = H(M) + H(K) - H(C) =$$

$$= H(K)log(|M|) - D(M) - H(C) \geq$$

$$\geq H(K) - D(M),$$

откуда

$$D(M) \geq H(K) - H(K|C) =$$

$$= I(K|C).$$

По этой теореме избыточность языка сообщений дает возможность для успешного анализа шифровок, то есть для получения информации о ключе по шифровке.

Задачи и упражнения

Задача 5.1. Пусть $M = C = \mathbf{Z}_n^t$ $(n, t \geq 2)$ – пространства исходных текстов и шифровок соответственно; $K = \mathbf{Z}_n^l$ $(l \geq 2)$ – пространство ключей. Считаем, что вероятности на всех этих пространствах распределены равномерно. В качестве системы шифрования используется система Виженера.

1. Доказать, что при $l \leq t - 1$ такая система не будет совершенной.

2. Что можно сказать о ней, если $t \leq l$?

Задача 5.2. Пусть случайная величина Q принимает не более t значений. Чему равны минимальное и максимальное значения энтропии $H(Q)$ при всех возможных распределениях Q?

Задача 5.3. Пусть X, Y – случайные величины. Справедлива ли оценка

$$H(X|Y) \leq H(X)?$$

Что можно сказать об этой оценке, если величины X, Y независимые?

Задача 5.4. Верно ли, что для любого шифра справеливо равенство

$$H(M|K, C) = 0?$$

Тот же вопрос для соотношения

$$H(K|M) = H(K) + H(M).$$

Простейшие шифры

1. Аффинный шифр

Пусть шифруется оцифрованный текст, единицами которого являются вычеты $\bmod n$. Это означает, что в данном случае платформой шифрования является кольцо вычетов $\mathbf{Z}_n$. Единицами шифрованного текста также служат элементы кольца $\mathbf{Z}_n$.

Итак, пусть $m \in \mathbf{Z}_n$ – произвольная единица исходного текста. Соответствующая ей единица шифрованного текста $c \in \mathbf{Z}_n$ вычисляется по правилу

$$c = am + b(\mathrm{mod} n), \tag{1.1}$$

где $a, b \in \mathbf{Z}_n$ – некоторые параметры. Пару $e = (a, b)$ можно считать ключом шифрования. Параметры ключа не могут быть произвольными. Для обращения функции шифрования

$$E_e(m) = am + b(\mathrm{mod} n)$$

необходима обратимость $a(\mathrm{mod} n)$. Тогда

$$m = a^{-1}c - a^{-1}b(\mathrm{mod} n). \tag{1.2}$$

Как известно из лекции 2, существование $a^{-1}(\mathrm{mod} n)$ равносильно взаимной простоте a, n: нод$(a, n) = 1$.

Итак, параметр a ключа $e = (a, b)$ должен быть взаимно прост с модулем n. На параметр b ограничения не накладываются – b может быть произвольным.

Равенство (1.2) показывает, что дешифрование происходит по той же схеме, что и шифрование. При этом используется ключ дешифрования $d = (a^{-1}, -a^{-1}b)$.

Зная ключ e, легко восстановить ключ d, и наоборот. Поэтому оба ключа должны храниться в секрете.

Криптостойкость аффинного шифра незначительна. Например, зная две соответствующие друг другу пары единиц (m_1, c_1), (m_2, c_2) исходного и шифрованного текста, мы можем пытаться найти параметры a, b, решая систему сравнений:

$$\begin{cases} c_1 &= am_1 + b(\mathrm{mod} n), \\ c_2 &= am_2 + b(\mathrm{mod} n). \end{cases} \tag{1.3}$$

Вычитая одно сравнение из другого, получим

$$c_1 - c_2 = a(m_1 - m_2)(\mathrm{mod} n)$$

и, если нод$(m_1 - m_2, n) = 1$, однозначно вычислим

$$a = (c_1 - c_2)(m_1 - m_2)^{-1}(\mathrm{mod} n),$$

а затем и b. Ясно, что система (1.3) не всегда имеет единственное решение (a, b), но при достаточно большом наборе известных пар $(m_1, c_1), (m_2, c_2), \ldots, (m_k, c_k)$, по-видимому, нахождение (a, b) не составит труда.

Аффинный шифр представляет собой частный случай шифра подстановки, поэтому его можно анализировать, используя частотный анализ.

2. Шифр Хилла

Данный метод шифрования является обобщением предыдущего аффинного шифрования. Л. Хилл (L. Hill) ввел его в обращение в двух своих статьях в 1929 и 1931 гг.

Единицами исходного и шифрованного текстов являются элементы кольца вычетов $\mathbf{Z}_n$. В данном случае текст разбивается на блоки одинаковой длины l. Шифрование осуществляется по правилу

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_l \end{pmatrix} = A \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_l \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{pmatrix}, \tag{2.1}$$

где $\begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_l \end{pmatrix}$ – блок исходного текста, $\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_l \end{pmatrix}$ – блок шифрованного текста,

A – матрица порядка $l \times l$ с элементами из кольца вычетов $\mathbf{Z_n}$, $\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{pmatrix}$ – вектор с элементами из $\mathbf{Z_n}$.

Для однозначного восстановления исходного текста по шифрованному тексту необходимо и достаточно, чтобы матрица A была обратима над $\mathbf{Z_n}$. Как известно, существование обратной матрицы A^{-1} равносильно тому, что определитель $\delta = \det A$ матрицы A обратим $\mathrm{mod} n$, то есть $\mathrm{нод}(\delta, n) = 1$.

Тогда дешифрование происходит по формуле

$$\begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_l \end{pmatrix} = A^{-1} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_l \end{pmatrix} - A^{-1} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{pmatrix}. \qquad (2.2)$$

Пример 11. *Пусть* $l = 2$, $n = 33$, *элементы кольца* $\mathbf{Z_{33}}$ *нумеруют буквы русского алфавита.*

Полагаем $A = \begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix}$, *обращая внимание на то, что определитель* $\delta = detA = 4$ *взаимно прост с* $n = 33$. *Легко вычислить, что* $4^{-1} = 25(\mathrm{mod} 33)$. *Для упрощения вычислений берем* $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Зашифруем текст:

ВОСТОЧНЫЙ КАЗАХСТАН $\longleftrightarrow$

$2, 15.18, 19.15, 24.14, 28.10, 11.0, 8.0, 22.18, 19.0, 14$:

$$\begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix}\begin{pmatrix} 2 \\ 15 \end{pmatrix} = \begin{pmatrix} 32 \\ 27 \end{pmatrix} \longleftrightarrow (я,\ ъ),$$
$$\begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix}\begin{pmatrix} 18 \\ 19 \end{pmatrix} = \begin{pmatrix} 23 \\ 10 \end{pmatrix} \longleftrightarrow (ц,й),$$
$$\begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix}\begin{pmatrix} 15 \\ 24 \end{pmatrix} = \begin{pmatrix} 30 \\ 30 \end{pmatrix} \longleftrightarrow (э,\ э)$$

и т. д. Все вычисления проводятся mod33.

Зашифрованный текст имеет вид:

ЯЪЦЙЭЭДМЯКПЯКХЦЙЫЦ $\longleftrightarrow$

$32, 27.23, 10.30, 30.4, 13.32, 11.16, 32.11, 22.23, 10.28, 23.$

Произведем дешифровку. Вычисляем

$$A^{-1} = 25\begin{pmatrix} 4 & -2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 16 \\ 0 & 25 \end{pmatrix}.$$

Считаем далее:

$$\begin{pmatrix} 1 & 16 \\ 0 & 25 \end{pmatrix}\begin{pmatrix} 32 \\ 27 \end{pmatrix} = \begin{pmatrix} 2 \\ 15 \end{pmatrix} \longleftrightarrow (в,\ о),$$
$$\begin{pmatrix} 1 & 16 \\ 0 & 25 \end{pmatrix}\begin{pmatrix} 23 \\ 10 \end{pmatrix} = \begin{pmatrix} 18 \\ 19 \end{pmatrix} \longleftrightarrow (с,\ т)$$

и т. д., пока не получим исходный текст.

Криптостойкость шифра Хилла также не очень велика. Определить элементы матрицы A и вектора $\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{pmatrix}$ можно, зная $l+1$ пару соответствующих друг другу блоков исходного и шифрованного текста. Это позволит записать систему $l^2 + l$ сравнений от такого же числа неизвестных. Если система не будет однозначно разрешимой, нужно добавить еще несколько сравнений.

Пример 12. *Пусть имеется зашифрованный текст английского языка, единицами которого служат элементы кольца* $\mathbf{Z}_{26}$, *разбитые на блоки длины* $l = 2$:

$m, b.s, r.f, j.s, f.y, t.f, b \longleftrightarrow 12, 1.20, 17.5, 9.20, 5.24, 19.5, 1.$

Предположим, что известен соответствующий исходный текст

$cryptography \longleftrightarrow 2, 17.24, 15.19, 14.6, 17.0, 15.7, 24.$

Попробуем восстановить ключ шифрования, записав соответствующую систему линейных уравнений относительно неизвестных компонент матрицы $A = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix}$ *и вектора* $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$.

Легко видеть, что система распадается на две подсистемы относительно неизвестных $\alpha_{11}, \alpha_{12}, b_1$ *в 1-й и* $\alpha_{21}, \alpha_{22}, b_2$ *во 2-й. Все вычисления идут в кольце* $\mathbf{Z}_{26}$. *Записываем 1-ю подсистему:*

$$\begin{cases} 2\alpha_{11} + 17\alpha_{12} + b_1 = 12, \\ 24\alpha_{11} + 15\alpha_{12} + b_1 = 20, \\ 19\alpha_{11} + 14\alpha_{12} + b_1 = 5, \\ 6\alpha_{11} + 17\alpha_{12} + b_1 = 20, \\ 0\alpha_{11} + 15\alpha_{12} + b_1 = 24, \\ 7\alpha_{11} + 24\alpha_{12} + b_1 = 5. \end{cases}$$

Получилась явно избыточная система, для которой легко находится единственное решение:

$$\alpha_{11} = 2(\mathrm{mod}26), \alpha_{12} = 5(\mathrm{mod}26), b_1 = 1(\mathrm{mod}26).$$

Составив систему относительно $\alpha_{21}, \alpha_{22}, b_2$, *получим единственное решение:*

$$\alpha_{21} = 1(\mathrm{mod}26), \alpha_{22} = 3(\mathrm{mod}26), b_2 = 0(\mathrm{mod}26).$$

Таким образом, шифрование осуществляется с помощью матрицы $A = \begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix} \in GL_2(\mathbf{Z}_{26})$ *и вектора* $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ *над* $\mathbf{Z}_{26}$.

3. Шифр Виженера

В XIX в. большое распространение получил так называемый *метод блокнотного шифрования.* Им пользовались революционеры – народники, шпионы и т. п. Например, у кого-нибудь в блокноте записана безобидная с виду фраза, скажем: «14 июля – Mary's birthday». Возможно, что действительно у некой Мери это был день рождения. А на самом деле данная запись хранила ключ шифрования. Если мы используем принятую нами для примеров нумерацию букв английского алфавита, то Marysbirthday означает $12, 0, 17, 24, 18, 1, 8, 17, 19, 7, 3, 0, 24$. Для шифровки фразы $Iamgoing \longleftrightarrow 8, 0, 12, 6, 14, 8, 13, 6$ производится сложение mod 26 текста с ключом, в роли которого выступает записанная фраза. Получается

$$20, 0, 3, 4, 6, 9,\ 21, 23 \longleftrightarrow UADEGJVX.$$

Как мы видим, в данном случае это обыкновенное гаммирование. Французский криптограф Виженер (Vigenere) предложил использовать ключ такого типа и в тех случаях, когда текст длиннее ключа, накладывая его столько раз, сколько нужно. При этом совсем необязательно, чтобы ключ получался из осмысленной фразы. Более того, это даже нежелательно, так как осмысленность может помочь взломщику шифра.

Возьмем, например, текст (мой вольный перевод знаменитой фразы Тютчева):

A SMOKE OF MOTHERLAND IS SWEET FOR US AND PLEASANT $\leftrightarrow$

$0, 18, 12, 14, 10, 4, 14, 5, 12, 14, 19, 7, 4, 17, 11, 0, 13, 3, 8, 18, 18, 22, 4, 4, 19,$
$5, 14, 17, 20, 18, 0, 13, 3, 15, 11, 4, 0, 18, 0, 13, 19$
и ключ $17, 9, 3, 8$.

Шифровка получается гаммированием mod26:

$pt : 0, 18, 12, 14, 10, 4, 14, 5, 12, 14, 19, 7, 4, 17, 11, 0, 13, 3, 8, 18,$
$key : 17, 9, 3, 8, 17, 9, 3, 8, 17, 9, 3, 8, 17, 9, 3, 8, 17, 9, 3, 8,$

ct : 17, 1, 15, 22, 1, 13, 17, 13, 3, 23, 22, 15, 21, 0, 14, 8, 4, 12, 11, 0,
pt : 18, 22, 4, 4, 19, 5, 14, 17, 20, 18, 0, 13, 3, 15, 11, 4, 0, 18, 0, 13, 19,
key : 17, 9, 3, 8, 17, 9, 3, 8, 17, 9, 3, 8, 17, 9, 3, 8, 17, 9, 3, 8, 17,
ct : 9, 5, 7, 12, 10, 14, 17, 25, 11, 1, 3, 21, 20, 24, 14, 12, 17, 1, 3, 21, 10.

Итак, шифр Виженера получается как повторяющаяся комбинация сдвигов. В общем случае этот шифр не сохраняет частоту встречающихся букв и по этой причине не может напрямую подвергаться статистическому анализу.

Существуют эффективные способы, позволяющие, во-первых, вычислить длину ключа и, во-вторых, определить сдвиги в ключе. Одним из таких способов является тест Казисского.

4. Тест Казисского и определение с его помощью длины ключа в шифре Виженера

Вначале определим понятие *индекса косовпадения* (ci) данного текста. Пусть рассматривается текст m, соответствующий алфавиту, состоящему из n букв. Пусть $l = |m|$ – длина этого текста. Обозначим через l_i число вхождений буквы с номером i в текст m. Тогда индекс косовпадения текста m определяется как

$$ci(m) = \left(\frac{l_1}{l}\right)^2 + \left(\frac{l_2}{l}\right)^2 + \ldots + \left(\frac{l_n}{l}\right)^2 = \frac{1}{l^2} \sum (l_i)^2.$$

Эмпирически проверено, что индекс косовпадения длинных осмысленных английских текстов, таких как «Моби Дик» Мелвила, приблизительно равен $0,065$. При этом, конечно, в тексте оставляют только 26 букв английского алфавита. В то же время абсолютно случайный достаточно длинный текст на 26 буквах, в котором все буквы встречаются приблизительно одинаковое число раз, равен $0,038$. Замечено, что чем «осмысленнее» текст, тем выше его индекс косовпадения. Это обстоятельство как раз и помогает вычислять длину ключа в шифре Виженера.

Алгоритм вычисления длины ключа

Пусть $m = m_1 m_2 m_3 \ldots$ – исходный текст, в котором m_i – его i-я буква, а $c = c_1 c_2 c_3 \ldots$ – его шифровка по Виженеру.

Если применяется обычный сдвиг, то есть длина ключа $|k| = 1$, то должно выполняться равенство $ci(m) = ci(c)$, поскольку изменяются только номера букв, но не числа их вхождений. Так как m – осмысленный (по предположению) текст, то значение $ci(c)$ будет приблизительно равно стандартному значению ci для данного языка. Мы рассматриваем пример обычного английского языка, поэтому $ci(c) \simeq 0,065$. Конечно, вряд ли шифр Виженера будет в общем случае получен ключом длины 1. Поэтому мы последовательно вычисляем следующие индексы косовпадения:

$$ci(c_1 c_2 c_3 \ldots) = d_1,$$
$$ci(c_1 c_3 c_5 \ldots) = d_2,$$
$$ci(c_1 c_4 c_7 \ldots) = d_3,$$
$$\ldots,$$
$$ci(c_1, c_{1+t}, c_{1+2t}, \ldots) = d_t,$$
$$\ldots$$

до тех пор, пока не получим $d_t \simeq 0,065$.

Это может свидетельствовать о том, что длина ключа равна t, хотя и может оказаться ложным следом.

Действительно, если длина ключа равна t, то текст $c_1 c_{1+t} c_{1+2t} \ldots$ будет получен из $m_1 m_{1+t} m_{1+2t} \ldots$ сдвигом, следовательно, сохранит $ci(m_1 m_{1+t} m_{1+2t} \ldots)$, а текст $m_1 m_{1+t} m_{1+2t} \ldots$, в свою очередь, является случайной выборкой осмысленного текста, следовательно, должен сохранить его статистические характеристики, в частности индекс косовпадения.

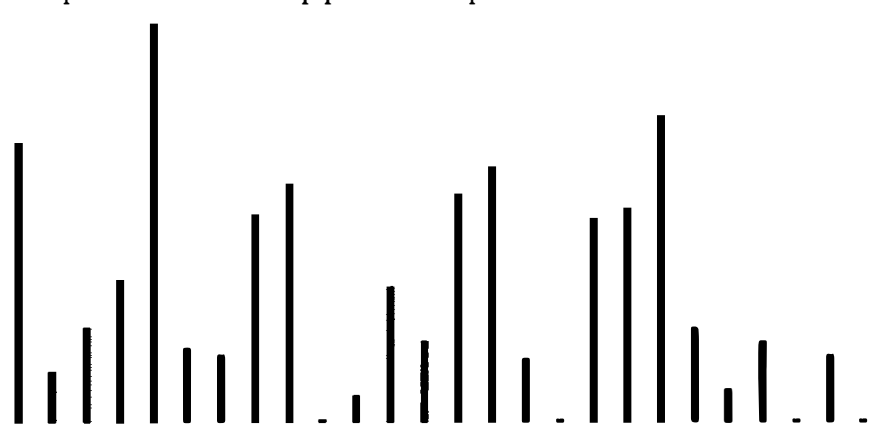
Если индекс косовпадения некоторого языка неизвестен, то использование теста Казисского также возможно. Нужно не сравнивать полученные значения индексов косовпадения со стандартным значением, а смотреть, когда этот индекс резко возрастет. Это может сигнализировать о найденной длине ключа. Конечно, речь идет о расшифровке осмысленных и одновременно достаточно длинных текстов. Впрочем, понятие осмысленности для формальных языков – понятие непростое.

Другим применением теста Казисского является проверка сохранения частот встречающихся букв при шифровании. Пусть c – зашифрованный текст, причем алгоритм шифрования неизвестен. Если известно, что использовался обычный английский алфавит и значение $ci(c)$ близко к $0,065$, то это дает основание полагать, что использовался шифр, сохраняющий частоты. Возможно, что это шифр простой замены. В ситуации, когда значение $ci(c)$ далеко от $0,065$, можно предположить, что использовался шифр, не сохраняющий частоты, или же текст был бессмысленным, или же использовался другой алфавит и т. п. Одним словом, что-то оказалось не так и необходим более глубокий анализ.

Однако вернемся к шифру Виженера. Пусть мы правильно определили длину ключа, равную t. Теперь нужно найти сам ключ.

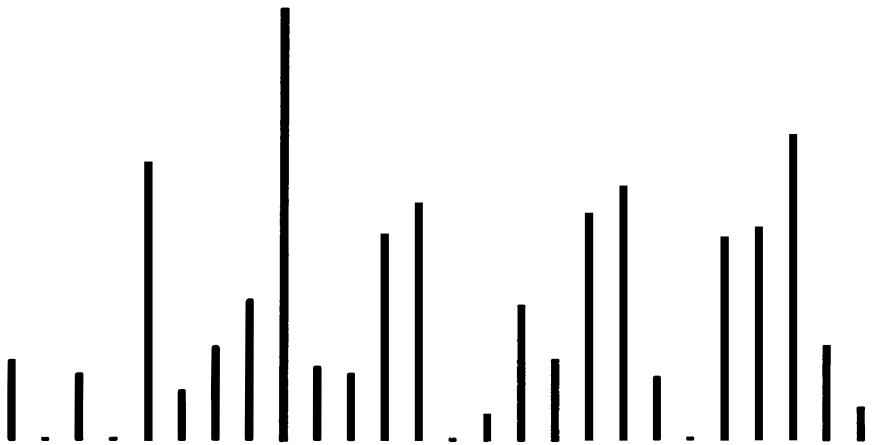
Определение ключа для шифра Виженера

Гистограмма, построенная по стандартным частотам букв в языке (в рассматриваемом ниже примере она соответствует таблице частот букв английского языка из лекции 4), имеет свои отличительные особенности. Они объясняются крайне неравномерным использованием букв в английском языке. Эта неравномерность как раз и позволяет эффективно применять частотный анализ.



Прежде всего обращают на себя внимание «пики», соответствующие буквам $A, E, H, I, N, O, R, S, T$, и «пеньки», соответствующие J, Q, X, Z. При этом некоторые «пики» стоят рядом, даже есть целая тройка: R, S, T. Все вместе дает весьма специфический рельеф.

Если используется сдвиг, скажем, на 4, то картина изменяется циклически:



Мы наблюдаем циклический сдвиг рельефа на 4 единицы. Если не знать величину сдвига, то ее нетрудно восстановить, руководствуясь здравым смыслом.

5. Роторные машины

Можно усовершенствовать шифр Виженера, рассматривая в качестве повторяющегося ключа комбинацию произвольных замен: $\sigma_1, \sigma_2, \ldots, \sigma_t$. Это означает, что единицы исходного текста $m_1 m_{1+t} m_{1+2t} \ldots$ преобразуются в $c_1 = \sigma_1(m_1)$, $c_{1+t} = \sigma_1(m_{1+t})$, $c_{1+2t} = \sigma_1(m_{1+2t}), \ldots$, единицы m_2, m_{2+t}, m_{2+2t} соответственно в $c_2 = \sigma_2(m_2), c_{2+t} = \sigma_2(m_{2+t}), c_{2+2t} = \sigma_2(m_{2+2t}), \ldots$ и т. д.

При взломе такого шифра, как и в случае шифра Виженера, вначале нужно определить длину ключа t. Это можно делать с использованием теста Казисского так же, как в описанном случае. Далее для определения подстановок $\sigma_1, \sigma_2, \ldots, \sigma_t$ можно применить частотный анализ.

Задачи и упражнения

Задача 6.1. Выбрать язык (алфавит), матрицу шифрования шифром Хилла размера не менее чем 3 на 3 и зашифровать фразу *Моя фамилия...* (можно выбрать другую фразу или использовать другой язык).

Задача 6.2. Построить шифр Виженера с длиной ключа не менее чем 5, выбрать фразу длины не менее чем 50 знаков, оцифровать ее и зашифровать, используя построенный шифр. Привести вычисления длины ключа, использующие тест Казисского (т. е. вычислить значения функции Казисского для прореженных текстов). Соответствуют ли вычисления теории?

Задача 6.3. Доказать, что кратное выполнение шифра замены для различных ключей $\sigma_1, \sigma_2 \in \mathbf{S}_n$ (n – мощность алфавита) равносильно однократному шифру замены.

Задача 6.4. Что можно сказать о композиции двух шифров перестановки с ключами $\tau_1 \in \mathbf{S}_m, \tau_2 \in \mathbf{S}_k$?

Задача 6.5. Доказать, что композиция двух шифров Виженера E_{k_1}, E_{k_2} с ключами k_1, k_2 длины l_1, l_2 соответственно снова является шифром Виженера E_{k_3} с ключом k_3. Чему равна длина l_3 ключа k_3?

Задача 6.6. ШИФРОГРАММА № 1:

G H Q T J B S Q R L N T U S F V S Q P W
6 7 16 19 9 1 18 16 17 11 13 19 20 18 5 21 18 16 15 22
Z E Z X S E Y M R V G H Q X G C I O W S
25 4 25 23 18 4 24 12 17 21 6 7 16 23 6 2 8 14 22 18
E E U R L R R Q W E V N S X G A O Z W H
4 4 20 17 11 17 17 16 22 11 21 13 18 23 6 0 14 25 22 7
R C U E D V S F W L U E F L W B R K M K
17 2 20 4 3 21 18 5 22 11 20 4 5 11 22 1 17 10 12 10
D U U X W O E M Y L V F G P S A D P I N

3 20 20 23 22 14 4 12 24 11 21 5 6 15 18 0 3 15 8 13
R L A T A A G D E H V D Q C
17 11 0 19 0 0 6 3 4 7 21 3 16 2

Известно, что эта шифрограмма получена шифром Виженера. Алфавит английский – 26 букв, занумерованных от 0 до 25. Длина ключа неизвестна.

1) Вычислить длину ключа. 2) Найти ключ. 3) Найти зашифрованный текст.

Задача 6.7. ШИФРОГРАММА № 2:

Щ Ц Ы Ж К Ч У Ц Ь С Д Ъ А Ц Б Т И Т К Ф
26 23 28 7 11 24 20 23 29 18 4 17 0 23 1 19 9 19 11 21
Ь Л Г Ъ Ь С Ю П Й Э Х Ч Р Х К Ф Ш П Н Т
29 12 3 27 29 18 31 16 10 30 22 24 17 22 11 21 25 16 14 19
А Э М Н Ю С Н Я Ц Ч П К Ы Ш Х Н Р Х Л Я
0 30 13 14 31 18 14 32 23 24 16 11 28 25 22 14 17 22 12 32
Л Ч Ч Ц О П О Д Ъ С О Х Й Щ Е Л З Л Л Ф
12 24 24 23 15 16 15 4 27 18 15 22 10 26 5 12 8 12 12 21
Ц Ш Т Л Ц С Б О О Ь Ё Р Ь Т О Ю К С Я Щ
23 25 19 12 23 18 1 15 15 29 6 17 29 19 15 31 11 18 32 26
А Ч У Ъ П Х Е Х В Н Ю И Л Р Ш Ц Ь Е Р Р
0 24 20 27 16 22 5 22 2 14 31 9 12 17 25 23 29 5 17 17
Т Е О Х Й Н Щ С Ъ Ё Б Ъ Н И Ц Ц Р Ъ Ы Ч
19 5 15 22 10 14 26 18 27 6 1 27 14 9 23 23 17 27 28 24
Ш Ж С Ц К П Ц Ш Л Я Н И Д Ж Р К У Щ Ь С
25 7 18 23 11 16 23 25 12 32 14 9 4 7 17 11 20 26 29 18
А Э М Ч Ч С Р Л Ш Ц И Р К Ч У А Я С А Т
0 30 13 24 24 18 17 12 25 23 9 17 11 24 20 0 32 18 0 19
У У Н С И Ш О Н П П З Ъ Ч Ы П Х Й Ц Ц С
20 20 14 18 9 25 15 14 16 16 8 27 24 28 16 22 10 23 23 18
Ы Х К Я Х Щ Н Я Е Щ У Ч Ы Ш К Л Т И Щ К
28 22 11 32 22 26 14 32 5 26 20 24 28 25 11 12 19 9 26 11
Е Н У Р Ъ Х Й Э Э Щ Н Я Е Щ Щ Ц А Ж Й Щ
5 14 20 17 27 22 10 30 30 26 14 32 5 26 26 23 0 7 10 26

К Ы Н Р Н И Щ С В Т А Ш Ч С Ъ
11 28 14 17 14 9 23 18 2 19 0 25 24 18 27

Известно, что эта шифрограмма получена шифром Виженера. Алфавит русский – 33 буквы, занумерованные от 0 до 32. Длина ключа неизвестна.

1) Вычислить длину ключа. 2) Найти ключ. 3) Найти зашифрованный текст.

Задача 6.8. Для какого языка индекс косовпадения абсолютно бессмысленного текста больше: для русского, английского или армянского? От чего зависит величина стандартного индекса косовпадения при одинаковом количестве букв в языках?

Группы

1. Аксиомы группы

Группой называется множество G с определенной на нем бинарной операцией, обычно называемой *умножением,* со стандартным обозначением: $c = a \cdot b$, или $c = ab$, удовлетворяющей следующим аксиомам:

1) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ *(ассоциативность)*,

2) существует элемент 1 *(единица),* для которого верно тождество

$a \cdot 1 = 1 \cdot a = a$ *(существование единицы)*,

3) для любого элемента a существует *обратный элемент* a^{-1} такой, что

$a \cdot a^{-1} = a^{-1} \cdot a = 1$ *(существование обратного)*.

Если на группе G выполнено также тождество *коммутативности* $a \cdot b = b \cdot a$, то такая группа называется *коммутативной*, или *абелевой* (в честь норвежского математика Абеля, доказавшего существование уравнений 5-й степени неразрешимых в радикалах).

Абелевы группы часто записываются аддитивно, то есть вместо операции умножения применяется операция сложения. Это в основном объясняется тем, что многие числовые системы и алгебраические структуры используют операцию сложения, относительно которой они являются абелевыми группами, то есть выполняются следующие аксиомы:

1) $(a + b) + c = a + (b + c)$ *(ассоциативность)*,

2) $a + 0 = 0 + a = a$ *(существование нуля)*,

3) $a + (-a) = (-a) + a = 0$ *(существование противоположного элемента)*,

4) $a + b = b + a$ *(коммутативность)*.

Пример 13. *Абелевы аддитивные группы:*
$\langle \mathbf{Z}, + \rangle$ – *группа целых чисел,*
$\langle \mathbf{Q}, + \rangle$ – *группа рациональных чисел,*
$\langle \mathbf{R}, + \rangle$ – *группа вещественных чисел,*
$\langle \mathbf{C}, + \rangle$ – *группа комплексных чисел,*
$\langle \mathbf{Z}_n, + \rangle$ – *группа вычетов* $\mathrm{mod} n$,
$\langle F[x], + \rangle$ – *группа многочленов над полем* F,
$\langle M_n[K], + \rangle$ – *группа матриц над кольцом* K.

Во всех группах примера на их основных множествах, кроме сложения, определена операция умножения. Но когда мы рассматриваем аддитивную группу, мы как бы забываем об остальных операциях. Группа состоит из множества с определенной на нем одной операцией.

Существуют естественные мультипликативные группы, происходящие из числовых и алгебраических систем. В качестве групповой операции берется определенное на основном множестве умножение (остальные операции – сложение и другие в определении не участвуют). Мультипликативные группы не обязательно абелевы.

Пример 14. *Если* K – *ассоциативное кольцо с* 1, *то множество всех его обратимых (по умножению) элементов* K^* *образует группу по умножению, называемую мультипликативной группой кольца* K *(если* $K = F$ *поле, то* F^* – *мультипликативная группа поля).*

Например, $\mathbf{Z}^* = \{1, -1\}$ – мультипликативная группа кольца целых чисел, $GL_n(K)$ – группа невырожденных матриц над ассоциативным кольцом K с 1. Заметим, что группа $GL_n(K)$ при $n \geq 2$ не коммутативна (неабелева).

Если в группе G конечное число n элементов, то говорят, что G – *конечная группа порядка* n: $|G| = n$. В противном случае группу G называют *бесконечной.*

Пусть G – некоторая группа. Если H – группа, содержащаяся в G : $H \subseteq G$, причем операция в H индуцирована операцией в G, то говорят, что H *подгруппа* группы G, и обозначают этот факт как $H \leq G$.

Если $g \in G$ – элемент группы, то все его степени $\{g^k | k \in \mathbf{Z}\} = $ $=< g >$ образуют подгруппу, которая называется *циклической подгруппой, порожденной элементом* g. При этом считается, что $g^0 = 1$. Если $G =< g >$, то говорят, что G – *циклическая группа, порожденная элементом* g.

Если $g^k = 1$ для некоторого целого положительного k, то наименьшее k с указанным свойством называется *порядком элемента* g : $|g| = k$. В этом случае $< g > = \{g^0 = 1, g, g^2, \ldots, g^{k-1}\}$. Если такого k не существует, то подгруппа $< g >$ оказывается бесконечной циклической группой. При этом говорят, что порядок элемента g *бесконечен*. Заметим, что в бесконечной циклической группе $< g >$ равенство степеней $g^k = g^l$ влечет равенство показателей $k = l$. Во всех случаях порядок подгруппы $| < g > |$ равен по определению порядку $|g|$ элемента g.

Пример 15. *Матрица* $a = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ *имеет в группе* $GL_2(\mathbf{Z})$ *порядок* 2, *матрица* $b = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ *в той же группе* $GL_2(\mathbf{Z})$ *имеет бесконечный порядок. Если рассматривать группу* $GL_2(\mathbf{Z}_5)$, *то в ней порядок элемента* $c = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ *(внешне выглядящего как* b *в предыдущем случае) равен* 5.

2. Теоремы Ферма, Эйлера, Лагранжа

Следующие утверждения хорошо известны в теории чисел.

Теорема 16. *(Малая теорема Ферма). Пусть* p *– простое число,* a *– целое число, взаимно простое с* p*:* нод$(a, p) = 1$. *Тогда справедливо сравнение*

$$a^{p-1} = 1(\mathrm{mod} p). \tag{2.1}$$

Теорема 17. *(Теорема Эйлера). Пусть $n \geq 2$ – целое положительное число, $\varphi(n)$ – функция Эйлера. Пусть a – целое число, взаимно простое с n: нод$(a, n) = 1$. Тогда справедливо сравнение*

$$a^{\varphi(n)} = 1(\mathrm{mod} n). \tag{2.2}$$

Для нас важно отметить, что равенства (2.1), (2.2) не только хорошо формулируются на языке теории групп, но представляют собой частные случаи следующей более общей теоремы.

Теорема 18. *(Теорема Лагранжа). Пусть G – конечная группа, $H \leq G$ – ее подгруппа, тогда порядок $|G|$ группы G делится на порядок $|H|$ подгруппы H:*

$$|G| = |H||G : H|.$$

Частное от деления, обозначенное $|G : H|$, называется индексом подгруппы H в группе G.

В частности, порядок $|G|$ группы G делится на порядок $|g|$ любого ее элемента.

Следствие 19. *1) Пусть $n = |G|$ – порядок группы G, $g \in G$ – произвольный элемент. Тогда $g^n = 1$.*

2) Наоборот, пусть $m = |g|$ – порядок элемента $g \in G$ и $g^q = 1$. Тогда $q \vdots m$ (в частности, $n \vdots m$).

Доказательство. 1) Пусть $k = | < g > | = |g|$ – порядок элемента g. По теореме Лагранжа $n = kl, l \in \mathbf{Z}$. Тогда $g^n = (g^k)^l = 1^l = 1$.

2) Разделим с остатком: $q = m \cdot s + r$, $0 \leq r < m$. Тогда $1 = g^q = (g^m)^s \cdot g^r = g^r$, что противоречит при $r > 0$ минимальности m. Значит, $r = 0$ и $q \vdots m$.

Теперь можно объяснить, почему теоремы 16 и 17 (Ферма и Эйлера) являются частными случаями теоремы Лагранжа.

В условиях теоремы 16 рассмотрим группу $\mathbf{Z}_p^*$ обратимых элементов поля $\mathbf{Z}_p$. Как мы знаем, в поле $\mathbf{Z}_p$ обратимы все элементы, кроме 0. Порядок группы $\mathbf{Z}_p^*$ равен $p - 1$. По теореме Лагранжа любой элемент $g \in \mathbf{Z}_p^*$ группы удовлетворяет равенству $g^{p-1} = 1$. На языке сравнений это есть ни что иное, как (2.1). Заметим, что в качестве g можно брать любое число $a \in \mathbf{Z}$, взаимно простое с p. Именно такие числа являются именами элементов группы $\mathbf{Z}_p^*$.

Аналогичные рассуждения подходят и для теоремы 17. Обратимы $\mathrm{mod} n$ только те числа a, которые взаимно просты с n: $\mathrm{нод}(a, n) = 1$. Порядок группы $\mathbf{Z}_n^*$ обратимых элементов равен по определению $\varphi(n)$. Значит, по теореме Лагранжа $a^{\varphi(n)} = 1$, что на языке сравнений записывается как (2.2).

Предложение 20. *Пусть a, b – элементы группы G, имеющие взаимно простые порядки: $|a| = k, |b| = l$: нод$(k, l) = 1$. Тогда из равенства $a^r = b^s$ вытекает, что $r \vdots k, s \vdots l$ и, значит, $a^r = b^s = 1$.*

Доказательство. Предположим, что $a^r = b^s \neq 1$, тогда $\mathrm{нод}(r, k) = d \neq k$, $\mathrm{нод}(s, l) = c \neq l$. Тогда $a^{rk/d} = 1$, но в этом случае $b^{sk/d} = 1$, чего не может быть, так как $\mathrm{нод}(sk/d, l) = c \neq l$.

Предложение доказано.

Предложение 21. *Пусть a, b – элементы абелевой группы A, имеющие взаимно простые порядки: $|a| = k, |b| = l$. Тогда элемент $c = ab$ имеет порядок $|c| = kl$.*

Доказательство. Ясно, что $c^{kl} = (a^k)^l (b^l)^k = 1$. Предположим, что $|c| = m$. Тогда $c^m = a^m b^m = 1$. В этом случае $a^m = (b^{-1})^m$. Так как порядки взаимно обратных элементов одинаковы, $|b^{-1}| =$

$= l$. По предложению 20 получаем равенства $a^m = 1$, $b^m = 1$. По теореме Лагранжа $m \vdots k, m \vdots l$, но тогда в силу взаимной простоты k и l получаем $m \vdots kl$, значит, $m = kl$.

Предложение доказано.

Предложение 22. *Пусть A – конечная абелева группа. Предположим, что элемент $a \in A$ имеет наибольший порядок $|a| = m$. Тогда порядок $|b|$ любого другого элемента b делит m, следовательно, $b^m = 1$.*

Доказательство. Если $A = < a >$ – циклическая группа, то утверждение следует из теоремы Лагранжа. Пусть порядок $k = |b|$ некоторого элемента $b \in A$ не делит m. Тогда в разложениях чисел k, m по степеням простых относительно некоторого p показатель степени для k будет больше, чем для m : $k = p^\alpha k_1$, $m = p^\beta k_2$, где $\beta < \alpha$, нод$(k_i, p) = 1$, $i = 1, 2$. Элемент b^{k_1} имеет порядок $|b^{k_1}| = p^\alpha$. Элемент a^{p^β} имеет порядок k_2. Так как нод$(p, k_2) = 1$, по предложению 21 элемент $a^{p^\beta} b^{k_1}$ имеет порядок $|a^{p^\beta} b^{k_1}| = k_2 p^\alpha > k_2 p^\beta$, что противоречит максимальности m.

Предложение доказано.

Задачи и упражнения

Задача 7.1. Сколько решений имеет уравнение вида $x^2 = b$ в группе $\mathbf{Z}_{21}^*$? Привести все возможные случаи.

Задача 7.2. Будет ли циклической мультипликативная группа $\mathbf{Z}_{24}^*$?

Задача 7.3. Сколько порождающих элементов в мультипликативной группе $\mathbf{Z}_{23}^*$?

Задача 7.4. Доказать, что абелева группа порядка pq, где p, q – различные простые числа, обязательно циклическая. Верно ли это утверждение в случае абелевой группы порядка p^2(p – простое)? Построить пример неабелевой (заведомо нециклической) группы порядка pq, где p, q – также различные простые числа.

Конечные поля

1. Простые конечные поля

Из лекции 2 мы знаем, что в кольце вычетов $\mathbf{Z}_n$ обратимы те и только те элементы m, для которых нод$(m, n) = 1$. Если $n = p$ – простое число, то из приведенного утверждения следует, что в кольце $\mathbf{Z}_p$ обратимы все вычеты, кроме 0. Напомним, что *полем* называется коммутативное ассоциативное кольцо с 1, все ненулевые элементы которого обратимы. Итак, справедлива

Лемма 23. *Кольцо вычетов $\mathbf{Z}_n$ является полем тогда и только тогда, когда $n = p$ – простое число.*

Поля $\mathbf{Z}_p$, p – простое, называются *простыми полями*. Любое конечное поле F содержит некоторое простое поле $\mathbf{Z}_p$. Действительно, $1 \in F$, значит, $2 = 1+1, 3 = 1+1+1, \ldots, n = 1+1+\cdots+1 \in \in F$, но в силу конечности F все такие элементы не могут быть различными. Если, например, $n = n + k$ в F, то $k = 0$. Наименьшее число $k = 0$ в F должно быть простым. Если бы это было не так и $k = rs$, то из обратимости r следует $r^{-1}k = s = r^{-1}0 = 0$ в F, что противоречит минимальности k. Итак, $k = p$ – простое число, которое называется *характеристикой* поля F. Элементы $0, 1, \ldots, p-1$, очевидно, составляют простое поле $\mathbf{Z}_p \subseteq F$.

Элементы поля F характеристики p можно умножать на элементы подполя $\mathbf{Z}_p$. Относительно этой операции и операции сложения в F можно рассматривать поле F как векторное пространство над $\mathbf{Z}_p$. Тогда существует базис $e_1, e_2, \ldots, e_r$, и любой элемент $w \in F$ однозначно записывается в виде $w = \alpha_1 e_1 + \alpha_2 e_2 + \cdots + \alpha_r e_r$, где $\alpha_i \in \mathbf{Z}_p$. Отсюда получаем, что $|F| = p^r$, где $|F|$ обозначает *порядок* (число элементов) поля F. Мы видим, что любое конечное

поле имеет примарный порядок p^r. Из алгебры известно, что поле порядка p^r всегда существует для любого простого p и любого натурального r. Можно доказать, что с точностью до изоморфизма такое поле единственно. Оно обозначается как $\mathbf{F}_{p^r}$. Заметим, что $\mathbf{F}_{p^1} = \mathbf{Z}_p$. Простые поля можно еще определять как поля, не содержащие собственных (т. е. не совпадающих со всем полем) подполей.

Приведенное выше рассуждение справедливо также для бесконечного поля F, в котором $p = 0$ для некоторого минимального натурального (обязательно простого) числа p. В этом случае поле F имеет бесконечную размерность над $\mathbf{Z}_p$. Если же в некотором поле F не выполняется равенство $p = 0$ ни для какого простого p, то можно легко показать, что поле F содержит подполе рациональных чисел $\mathbf{Q}$ – единственное бесконечное простое поле, характеристика которого, а значит и поля F, по определению считается равной 0.

2. Построение конечных полей

Для нужд криптографии важно уметь строить конечные поля, уметь однозначно записывать их элементы и эффективно выполнять основные операции сложения и умножения. Важно также уметь эффективно вычислять обратные для ненулевых элементов. Все это можно делать, что позволяет эффективно использовать конечные поля в качестве платформ для шифрования и криптографических протоколов.

Пусть требуется построить конечное поле $\mathbf{F}_{p^r}$. Для того чтобы это сделать, выберем неразложимый на множители (неприводимый) многочлен $f(x) \in \mathbf{Z}_p[x]$ от одной переменной x с коэффициентами из простого поля $\mathbf{Z}_p$ нужной нам степени r. Напомним, что под неразложимостью $f(x)$ понимается отсутствие представления $f(x) = g(x)h(x)$, где многочлены $g(x), h(x) \in \mathbf{Z}_p[x]$ имеют степени строго меньше, чем степень многочлена $f(x)$. Представление вида $f(x) = \alpha(\alpha^{-1}f(x))$, где $0 \neq \alpha \in \mathbf{Z}_p$, разложением не считается. Всегда можно считать, что коэффициент перед степенью

x^r в записи $f(x)$ равен 1. Из алгебры известно, что существует неприводимый многочлен любой наперед заданной степени r.

В качестве основного множества элементов поля $\mathbf{F}_{p^r}$ выберем множество всех многочленов из $\mathbf{Z}_p[x]$ степени не больше чем $r-1$. Легко видеть, что таких многочленов $g(x) = \alpha_{r-1}x^{r-1} + \alpha_2 x^{r-2} + \\ + \ldots + \alpha_0$ в точности p^r. Многочлены из $\mathbf{F}_{p^r}$ складываются обычным образом. Для того чтобы выполнить умножение $g(x)$ на $h(x)$, вначале берем произведение $k(x) = g(x)h(x)$ в $\mathbf{Z}_p[x]$, а затем вычисляем остаток $r(x)$ при делении $k(x)$ на $f(x)$: $k(x) = f(x)s(x) + r(x)$. Записываем: $g(x)h(x) = r(x)$ или $g(x)h(x) = r(x)(\mathrm{mod} f(x))$. Мы видим, что операции в $\mathbf{F}_{p^r}$ выполняются $\mathrm{mod} f(x)$. На алгебраическом языке $\mathbf{F}_{p^r}$ есть фактор кольцо $\mathbf{Z}_p[x]$ по идеалу, порожденному многочленом $f(x)$. Зачем же все-таки была необходима неразложимость $f(x)$? Указанным выше способом мы определяем операции $\mathrm{mod} f(x)$, не используя требуемой неразложимости. Дело в том, что полем построенное кольцо оказывается только в случае, если $f(x)$ неразложим. Если данное условие выполняется, то любой ненулевой элемент полученного кольца – многочлен $g(x) \in \mathbf{Z}_p[x]$ степени строго меньшей чем r – оказывается взаимно простым с $f(x)$. Используя обобщенный алгоритм Эвклида, мы можем получить представление вида

$$1 = g(x)h(x) + f(x)u(x),$$

что дает равенство в $\mathbf{F}_{p^r}$:

$$1 = g(x)h(x)(\mathrm{mod} f(x)),$$

поэтому

$$h(x) = g(x)^{-1}(\mathrm{mod} f(x)).$$

Если $f(x)$ разложим: $f(x) = t(x)u(x)$, где $t(x), u(x)$ – многочлены степени строго меньшей чем r, то в построенном $\mathrm{mod} f(x)$ кольце имеем равенство $t(x)u(x) = 0$. Ненулевые элементы любого кольца (в данном случае $t(x), u(x)$), произведение которых равно 0, называются *делителями нуля*. В поле не может быть делителей нуля из-за противоречащего равенства

$$t(x)^{-1}(t(x)u(x)) = u(x) = t(x)^{-1}0 = 0.$$

Пример 24. *Построим поле* $\mathbf{F}_{5^2}$. *Для этого выберем многочлен* $f(x) = x^2 + x + 2 \in \mathbf{Z}_5[x]$. *Если бы* $f(x)$ *был разложим, то он представлялся бы в виде* $f(x) = (x+a)(x+b)$. *Элементы* $-a, -b \in \mathbf{Z}_5$ *были бы корнями уравнения* $f(x) = 0$. *Однако мы видим, что* $f(0) = 2, f(1) = 4, f(2) = 3, f(3) = 4, f(4) = 2$, *т. е.* $f(x)$ *не имеет корней в* $\mathbf{Z}_5$. *Поле* $\mathbf{F}_{5^2}$ *состоит из многочленов степени* ≤ 1 *над* $\mathbf{Z}_5 : 0, 1, \ldots, x, x+1, \ldots, x+4, 2x, \ldots, 4x+4$. *Все они, кроме 0, обратимы в* $\mathbf{F}_{5^2}$. *Вычислим, например,* $(3x+1)^{-1}$. *Для этого применим алгоритм Эвклида:*

$$x^2 + x + 2 = (3x + 1)2x + (4x + 2), 3x + 1 = (4x + 2)2 + 2,$$

поэтому

$$1 = (3x + 1)3 - (4x + 2) = (3x + 1)3 - ((x^2 + x + 2) - (3x + 1)2x) =$$

$$= (3x + 1)(2x + 3) - (x^2 + x + 2),$$

значит,

$$1 = (3x + 1)(2x + 3)(\mathrm{mod}(x^2 + x + 2)),$$

$$2x + 3 = (3x + 1)^{-1}(\mathrm{mod}(x^2 + x + 2)).$$

3. Цикличность мультипликативной группы конечного поля

Из алгебры хорошо известно, что многочлен $f(x) \in F[x]$ над полем F имеет корень $\alpha \in F$ тогда и только тогда, когда $f(x)$ делится на $x-\alpha$: $f(x) = (x-\alpha)f_1(x)$, где $f_1(x) \in F[x]$ – многочлен, степень которого на 1 меньше степени многочлена $f(x)$. По этой причине многочлен степени n не может иметь более n различных корней в F.

Заметим, что для многочленов с коэффициентами из кольца K, не являющегося полем, это свойство может оказаться нарушенным.

Например, многочлен x^2-1 имеет в кольце вычетов $\mathbf{Z}_{15}$ четыре корня: $1, 14, 4, 11$.

Для современной криптографии особое значение имеет

Предложение 25. *Мультипликативная группа $\mathbf{F}_{p^r}^*$ конечного поля $\mathbf{F}_{p^r}$ циклическая. Это значит, что существует такой элемент $g \in \mathbf{F}_{p^r}^*$ (он называется порождающим), что все элементы $\mathbf{F}_{p^r}^*$ являются его степенями: $\mathbf{F}_{p^r}^* = \{g, g^2, \ldots, g^{p^r-1} = 1\}$.*

Порождающий элемент g в общем случае не единственен.

Предварим доказательство следующим примером.

Пример 26. *Порождающим мультипликативной группы $\mathbf{F}_7^*$ является элемент $g = 3$: $\mathbf{F}_7^* = \{3^1 = 3, 3^2 = 2, 3^3 = 6, 3^4 = 4, 3^5 = = 5, 3^6 = 1\}$. Заметим, что элемент 2 не будет порождающим, поскольку его степени $2^1 = 2, 2^2 = 4, 2^3 = 1, 2^4 = 2, \ldots$ перечисляют не все элементы $\mathbf{F}_7^*$. Мы видим, что элемент $g \in \mathbf{F}_{p^r}^*$ является порождающим в том и только том случае, если его порядок, то есть наименьшая степень, равная 1, равен $p^r - 1$. В этом случае все элементы $g^1, g^2, \ldots, g^{p^r-1}$ различны, так как равенство $g^m = g^t (m > t)$ дает равенство $g^{m-t} = 1$.*

Доказательство. Пусть $g \in \mathbf{F}_{p^r}^*$ – элемент максимального порядка m. Если $m = p^r - 1$, то группа $\mathbf{F}_{p^r}^*$ состоит из степеней элемента g, поэтому является циклической группой, что и требовалось доказать

Если $m < p^r - 1$, то по предложению 22 из лекции 7 любой элемент $f \in \mathbf{F}_{p^r}^*$ удовлетворяет равенству $f^m = 1$, то есть является решением уравнения $x^m = 1$. Однако в поле уравнение степени m не может иметь более чем m корней. Полученное противоречие доказывает предложение.

4. Оцифровка конечных полей

Для целей криптографии необходимо уметь устанавливать взаимно однозначное соответствие между числами $0, 1, \ldots, p^r - 1$ и элементами конечного поля $\mathbf{F}_{p^r}$.

Проще всего это делается в случае простого поля $\mathbf{F}_{p^1} = \mathbf{Z}_p$, элементы которого (вычеты mod p) имеют стандартные имена $0, 1, \ldots, p-1$.

Пусть $r \geq 2$. Элементы поля $\mathbf{F}_{p^r}$ имеют стандартные однозначные записи как многочлены $f(x) \in \mathbf{Z}_p[x]$ степени не больше чем $r-1$: $f(x) = \alpha_{r-1}x^{r-1} + \alpha_{r-2}x^{r-2} + \ldots + \alpha_0 \longleftrightarrow (\alpha_{r-1}, \alpha_{r-2}, \ldots, \alpha_0)$, где $\alpha_i \in \mathbf{Z}_p$, $i = 0, 1, \ldots, r-1$. Сопоставим вектору $(\alpha_{r-1}, \alpha_{r-2}, \ldots, \alpha_0)$ указанного вида число в p-ичном исчислении

$$(\alpha_{r-1}, \alpha_{r-2}, \ldots, \alpha_0)_p = \alpha_0 + \alpha_1 \cdot p + \alpha_2 \cdot p^2 + \ldots + \alpha_{r-1} \cdot p^{r-1}.$$

Таким способом каждый элемент поля получает свой номер в пределах от 0 до $p^r - 1$. Разные многочлены (элементы поля $\mathbf{F}_{p^r}$) имеют разные номера. Такую оцифровку поля $\mathbf{F}_{p^r}$ в дальнейшем мы будем называть *стандартной.*

Задачи и упражнения

Задача 8.1. Построить конечные поля $\mathbf{F}_{2^4}, \mathbf{F}_{3^3}$ и $\mathbf{F}_{5^2}$, указав соответствующие многочлены $f(x)$. Найти порождающие элементы мультипликативных групп построенных полей. Будет ли элемент x (в каждом случае свой) порождать мультипликативную группу?

Задача 8.2. Доказать, что в любом поле $\mathbf{F}_{p^k}$ содержится единственный (тривиальный) корень p-й степени из 1.

Задача 8.3. Пусть поле $\mathbf{F}_{2^4}$ построено по многочлену $f(x) = x^4 + x^3 + 1 \in \mathbf{Z}_2[x]$.

Перечислить все элементы $a \in \mathbf{F}_{2^4}$, для которых разрешимо уравнение $y^3 = a$.

Задача 8.4. Доказать, что в поле $\mathbf{F}_{2^6}$ для любого $a \in \mathbf{F}_{2^6}$ разрешимо уравнение $y^5 = a$.

Дискретный логарифм

Неосторожный гувернер поторопился объяснить
мне в восемь лет логарифмы, а в одном
из детских английских журналов мне попалась
статейка про феноменального индуса,
который ровно в две секунды мог извлечь
корень семнадцатой степени из такого,
скажем, приятного числа, как
352947114576027513230189734205586617139 2
(кажется, 212, но это неважно).
В. Набоков. Другие берега

1. Определение дискретного логарифма

Напомним, что полем называется ассоциативное коммутативное кольцо с 1, в котором обратимы все ненулевые элементы. Среди всех колец вычетов $\mathbf{Z}_n$ полями являются только кольца вычетов $\mathbf{Z}_p$ по простым модулям p. Они называются простыми конечными полями. Любое конечное поле имеет примарный порядок p^r, p – простое число (характеристика поля). Оно обязательно содержит подполе $\mathbf{Z}_p$. Для каждого примарного числа p^r существует единственное с точностью до изоморфизма поле данного порядка, которое обозначается как $\mathbf{F}_{p^r}$.

Для построения конечного поля $F = \mathbf{F}_{p^r}$ данного порядка p^r нужно найти неприводимый многочлен $f(x) \in \mathbf{Z}_p[x]$ степени r. Такой многочлен всегда существует. Тогда в качестве F можно взять фактор кольцо $F = \mathbf{Z}_p[x]/$ ид$(f(x))$, где ид$(f(x))$ (идеал, порожденный многочленом $f(x)$) состоит из всех многочленов, делящихся на $f(x)$. Его элементы – многочлены степеней $\leq r-1$ над $\mathbf{Z}_p$, сложение и умножение осуществляется $\bmod f(x)$.

Мы видим, что элементы поля записываются эффективно как: $\alpha_0 + \alpha_1 x + \ldots + \alpha_{r-1} x^{r-1}$, $\alpha_i \in \mathbf{Z}_p, i = 0, ..., r-1$.

Напомним также, что мультипликативная группа F^* любого конечного поля F порядка p^r является циклической группой порядка $p^r - 1$.

Это означает, что в поле F есть такой элемент $g \in F$, что все элементы: $g, g^2, \ldots$ до степени $p^r - 1$, равной 1, различны и составляют все ненулевые элементы поля F, то есть мультипликативную группу F^*. Такие элементы g (их всегда достаточно много) называются *порождающими элементами мультипликативной группы* F^*, или *примитивными элементами поля* F.

Диффи и Хеллман [21] рассмотрели функции вида $\delta : \mathbf{Z} \longrightarrow \longrightarrow F^*, \delta(x) = g^x$ (здесь F – конечное поле, g – порождающий элемент мультипликативной группы F^*). Если

$$g^x = f \in F^*, \tag{1.1}$$

то x называется *дискретным логарифмом элемента* f *по основанию* g. Мы знаем, что уравнение $g^x = f$ разрешимо для любого элемента $f \in F^*$, причем всегда существует единственное решение в пределах $0 \leq x \leq p^r - 2$. Задача нахождения x по f (g считается заданным) известна как *проблема дискретного логарифма*. Она считается вычислительно трудной. То есть функция $\delta(x) = g^x$ может рассматриваться как односторонняя (one-way) функция и поэтому может использоваться (используется и очень широко) для конструирования различных шифросхем.

2. Некоторые протоколы, основанные на трудности нахождения дискретного логарифма

Данный протокол разделения секретного ключа (числа) между абонентами предложен в упомянутой выше знаменитой работе Диффи и Хеллмана.

Протокол 27. (Диффи – Хеллмана (Diffie – Hellman))

Пусть известно большое конечное поле F и порождающий элемент g мультипликативной группы F^*. Двум абонентам A и B требуется выбрать секретным образом какое-нибудь случайное большое число (секретный ключ), используя указанные данные в качестве платформы. Переписка между абонентами по этому поводу происходит по открытому каналу связи (например, через Интернет). Никакой предварительной договоренности абоненты не имеют. Сам выбор поля F и порождающего элемента g осуществляется по открытому каналу, и поэтому данные F, g являются открытыми. Мы считаем все элементы поля F перенумерованными, как это объяснено в лекции 8.

Процесс разделения ключа организуется следующим образом. Вначале происходит

Установка

Абоненты A и B по открытой сети договариваются о выборе поля F и порождающего элемента g.

Далее происходит генерация абонентами случайных (секретных) чисел и передача данных по открытой сети.

Генерация случайных чисел и передача данных по сети

Абонент A выбирает случайным образом натуральное число x, вычисляет $y = g^x$ и сообщает результат абоненту B. Тот, в свою очередь, выбирает случайным образом число z, вычисляет $u = g^z$ и сообщает результат абоненту A. Заметим, что все вычисления выполняются в поле F. Если, например, $F = \mathbf{Z}_p$ – простое поле, то эти вычисления ведутся mod$p$. *Передаваемые по открытой сети значения y, u обязательно имеют стандартные имена вычетов mod$p$, то есть $0 \leq y, u \leq p-1$. В случае произвольного конечного поля F имена передаваемых данных также стандартны (см. лекцию 8).*

Разделение секретного ключа

Абонент A, зная x и u, вычисляет элемент $u^x = g^{zx}$ и записывает его стандартным именем. Абонент B, зная y и z, вычисляет $y^z = g^{xz}$, также присваивая ему стандартное имя. В результате каждый из них узнает элемент g^{xz} и соответствующее ему число q – его номер при стандартной (или какой-нибудь другой) нумерации элементов поля F. Таким способом они «разделили» секретное число q.

Несанкционированный пользователь, перехватив g^x или g^z или то и другое, не сможет восстановить g^{xz} простым способом. Он должен решить либо проблему нахождения x по $y = g^x$, либо проблему нахождения z по $u = g^z$, либо придумать еще какой-нибудь способ вычисления g^{xz}.

Остается открытым вопрос: упрощает ли знание элементов g^x, g^z нахождение элемента g^{xz} или эта проблема в общем случае эквивалентна проблеме нахождения дискретного логарифма? Хотя формулировка вопроса довольно проста, ответить на него очень сложно.

Существуют и другие протоколы и системы шифрования, использующие понятие дискретного логарифма и основанные на трудности его нахождения. Например, Эль Гамалю принадлежит идея, которую в общих чертах можно сформулировать следующим образом.

Пусть A требуется передать сообщение v абоненту B. Элемент v можно считать принадлежащим полю F. Как замечено выше, A и B могут обменяться секретным для других элементом – ключом w. Тогда A может передать сообщение в виде $v + w$ или в виде $v \cdot w$. Абонент B расшифрует его, либо вычитая w, либо деля на w. В данном случае мы просто используем уже разделенный ключ. Приведем еще два широко известных протокола.

Протокол 28. (Масси – Омуры (Massey – Omura))

Впервые такой протокол был предложен Эль Гамалем, использовавшим в качестве платформы произвольное простое конечное поле. Вот почему он иногда называется протоколом Эль Гамаля. Масси и Омура распространили протокол на произвольные конечные поля.

Установка

Абоненты A и B по открытой сети договариваются о выборе простого числа p и большого конечного поля $F = \mathbf{F}_{p^r}$ порядка p^r. Эти данные открыты и являются платформой протокола.

Генерация случайных чисел и передача данных по сети

Абонент A имеет секретное сообщение m, представленное в виде числа $0 \leq m \leq p^r - 1$. Этому числу соответствует элемент поля с тем же обозначением. Далее он случайным образом выбирает число $0 < x \leq p^r - 2$, взаимно простое с $p^r - 1$: нод$(x, p^r - 1) = 1$. Одновременно он вычисляет $x^{-1}(\mathrm{mod}\,(p^r - 1))$. Наконец, A вычисляет стандартное имя элемента m^x.

Аналогично абонент B, который должен получить от A секретное сообщение m, выбирает случайное число $0 < z \leq p^r - 2$, взаимно простое с $p^r - 1$: нод$(z, p^r - 1) = 1$. Одновременно он вычисляет $z^{-1}(\mathrm{mod}\,(p^r - 1))$.

Далее абонент A передает по сети стандартное имя элемента m^x.

Абонент B вычисляет стандартное имя элемента m^{xz} и посылает его обратно абоненту A.

Абонент A вычисляет элемент $m^{xzx^{-1}} = m^z$ и посылает его стандартное имя абоненту B.

Расшифровка абонентом B сообщения m

Абонент B вычисляет элемент $(m^z)^{z^{-1}} = m$.

Протокол 29. (Эль Гамаля (ElGamal))

Этот протокол также придуман Эль Гамалем, что закрепилось в его названии.

Установка

Все абоненты некоторой системы по открытой сети договариваются о выборе большого конечного поля $F = \mathbf{F}_{p^r}$ порядка p^r и элемента большого порядка $g \in F^*$. Желательно, но не обязательно, чтобы элемент g порождал мультипликативную группу F^*. Эти данные открыты. Мультипликативная группа F^* вместе с элементом g является платформой протокола.

Каждый из абонентов, пусть это будет B, выбирает секретное натуральное число b и вычисляет стандартное имя элемента g^b. Значение g^b открыто для других пользователей системы. Оно позволяет им передавать засекреченные сообщения, прочитать которые может только знающий значение b, то есть только абонент B. Каких-то особых секретных договоренностей между различными абонентами не требуется.

Генерация случайных чисел и передача данных по сети

Абонент A хочет передать секретное сообщение $m \in F$ абоненту B. Он знает значение g^b, играющее в данном случае роль открытого ключа. Абонент A выбирает случайным образом натуральное число $1 \leq k \leq p^r - 2$. Послание, которое он передает по сети абоненту B, имеет вид: (g^k, mg^{bk}).

Расшифровка абонентом B сообщения m

Абонент B знает секрет b. Вначале он вычисляет $(g^k)^b = g^{bk}$, а затем $m = (mg^{bk})(g^{bk})^{-1}$.

Сделаем несколько замечаний к приведенным методам. Мы используем два факта – эффективную запись элементов поля F и эффективную вычислимость возведения в степень и умножения в

поле F. Это позволяет передавать сообщение. С другой стороны, используемая запись передаваемых элементов не дает возможности восстановить соответствующую степень элемента g (он фигурирует во всех протоколах) группы F^*, то есть решить проблему дискретного логарифма. Мы могли бы использовать вместо F^* любую другую циклическую группу G. Но обычно циклическая группа задается как набор степеней: $G = \{1, g, g^2, \ldots, g^{n-1}\}$, $|G| = n$, а умножение в ней записывается формулой: $g^k g^l = g^{k+l}$, где $k + l$ берется $\mathrm{mod} n$. При такой форме записи проблема дискретного логарифма решается очевидным образом – графически, соответствующий логарифм просто написан как показатель. Итак, выбор объекта, в котором происходит шифрование – группы, поля и тому подобного, еще обусловливается формой записи его элементов.

Слабости протоколов Диффи – Хеллмана, Масси – Омуры и Эль Гамаля

Основная слабость заключается в том, что в данных протоколах не предусмотрена процедура аутентификации. Протоколы не защищены от так называемой атаки «человек посередине», когда взломщик выдает себя за одного из корреспондентов.

3. Атаки на дискретный логарифм

Пусть $\mathbf{F}_q$ – конечное поле порядка $q = p^r$. Пусть $q - 1 = \prod_p p^{\alpha_p}$ – разложение числа $q - 1$ в произведение степеней простых чисел.

Предположим, что все простые делители числа $q - 1$ малы. В этом случае примарное число q называется гладким. При таком предположении существует следующий достаточно быстрый алгоритм нахождения дискретного логарифма в $\mathbf{F}_q^*$.

Алгоритм Сильвера – Полига – Хеллмана (Silver – Pohlig – Hellman)

Пусть g – порождающий элемент $\mathbf{F}_q^*$.

Вначале для каждого простого делителя p числа $q-1$ вычислим все корни p-й степени из $1: r_{p,j} = g^{j(q-1)/p}, j = 0, 1, \ldots, p-1$. Поскольку элемент g является порождающим элементом мультипликативной группы $\mathbf{F}_q^*$, все выписанные элементы различны. Легко проверить, что они являются корнями степени p из $1: r_{p,j}^p = g^{j(q-1)} = 1, j = 0, ..., p-1$ по Малой теореме Ферма. В любом поле решений уравнения степени p (в данном случае уравнения $z^p = 1$) не более чем p, значит, наш список полон.

Этот шаг проделывается один раз. Его результат – списки значений $r_{p,j}, j = 0, ..., p-1$, сохраняются. Обозначим такой список как $\sqrt[p]{1}$.

Рассмотрим проблему нахождения дискретного логарифма x:

$$g^x = y$$

при различных значениях $y \in F_q^*$.

Заметим, что нам достаточно найти для любого p, делящего $q-1$, вычет $x(p) = x(\mathrm{mod} p^{\alpha_p})$. Полученные значения $x(p)$ позволяют записать все сравнения $x = x(p)(\mathrm{mod} p^{\alpha_p})$ и вычислить x по Китайской теореме об остатках.

Будем использовать единственную p-ичную запись числа $x(p)(\mathrm{mod} p^{\alpha_p})$.

Предположим, что

$$x(p) = x_0 + x_1 p + x_2 p^2 + \ldots + x_{\alpha_p - 1} p^{\alpha_p - 1} (\mathrm{mod} p^{\alpha_p}),$$

$$0 \le x_i < p.$$

Для того чтобы вычислить x_0, подсчитаем вначале $y^{q-1/p}$. В результате мы получим корень p-й степени из 1 в $\mathbf{F}_q^*$. Так как $y = g^x$, то $y^{q-1/p} = g^{x(q-1)/p} = g^{x_0(q-1)/p} = r_{p,x_0}$.

Сравниваем r_{p,x_0} с элементами $r_{p,j}$, где $0 \le j < p$. Находим такое значение j, что $r_{p,x_0} = r_{p,j}$. Значит, $x_0 = j$.

Для того чтобы вычислить x_1, заменим y на

$$y_1 = y \cdot g^{-x_0} = g^{x_0 + x_1 p + \ldots + x_{\alpha-1} p^{\alpha-1}} \cdot g^{-x_0} = g^{x_1 p + \ldots + x_{\alpha_p - 1} p^{\alpha_F - 1}}.$$

Новый элемент y_1 имеет дискретный логарифм, равный

$$x - x_0 = x_1 p + \ldots + x_{\alpha_p - 1} p^{\alpha_p - 1} (\mathrm{mod} p^{\alpha_p}).$$

Ясно, что $y_1^{q-1/p} = 1$. Далее:

$$y_1^{q-1/p^2} = g^{(x-x_0)(q-1)/p^2} = g^{(x_1 + x_2 p + \ldots)(q-1)/p} = g^{x_1 (q-1)/p} = r_{p,x_1}.$$

Сравниваем r_{p,x_1} с элементами $\{r_{p,j}\}$. Находим $j : r_{p,x_1} = r_{p,j}$. Полагаем $x_1 = j$. Продолжаем этот процесс, находим значение

$$x = x(p) = x_0 + x_1 p + \ldots + x_{\alpha_p - 1} p^{\alpha_p - 1} (\mathrm{mod} p^{\alpha_p}).$$

Далее записываем систему сравнений:

$$x = x_p (\mathrm{mod} p^{\alpha_p})$$

для всех простых делителей p числа $q - 1$ и находим x, используя алгоритм, представленный в доказательстве Китайской теоремы об остатках (см. лекцию 2).

Пример 30. *Вычислим* $\log_2 28$ *в поле* $\mathbf{F}_{37}$.

$37 - 1 = 36 = 2^2 \cdot 3^2$.

Вначале считаем, что $p = 2, \alpha_2 = 2$. *Выпишем* $\sqrt[2]{1}$, *для чего считаем*

$$g^{q-1/2} = 2^{18} = 36 (\mathrm{mod} 37),$$

следовательно, $\sqrt[2]{1} = \{r_{2,0} = 1, r_{2,1} = 36\}$.

Заметим, что $\sqrt[2]{1}$ *имеет вид* $\sqrt[2]{1} = \{r_{2,0} = 1, r_{2,1} = -1\}$ *в любом поле* F.

Вычисляем $x = x(2) (\mathrm{mod} 2^2)$ *в виде*

$$x(2) = x_0 + x_1 \cdot 2 (\mathrm{mod} 2^2), 0 \leq x_0, x_1 \leq 1.$$

Для этого считаем

$$y^{q-1/2} = 28^{36/2} = 28^{18} = 1 (\mathrm{mod} 37).$$

Значит, $x_0 = 0$, *так как* $1 = r_{2,0}$.

Считаем далее:

$$y^{q-1/4} = 28^{36/4} = 28^9 = 36(\mathrm{mod}37),$$

следовательно, $x_1 = 1$, *так как* $36 = r_{2,1}$.

Таким образом,

$$x = x(2) = 2(\mathrm{mod}2^2).$$

Теперь перейдем к рассмотрению случая, когда $p = 3, \alpha_3 = 2$. *Выпишем* $\sqrt[3]{1}$,, *для чего считаем:*

$$g^{q-1/3} = 2^{36/3} = 2^{12} = 26(\mathrm{mod}37),$$

$$g^{2(q-1)/3} = 2^{24} = 10(\mathrm{mod}37),$$

следовательно, $\sqrt[3]{1} = \{r_{2,0} = 1, r_{2,1} = 26, r_{2,2} = 10\}$.

Вычисляем $x = x(3)(\mathrm{mod}3^2)$ *в виде*

$$x(3) = x_0 + x_1 \cdot 3(\mathrm{mod}3^2), 0 \leq x_0, x_1 \leq 2.$$

Для этого считаем

$$y^{q-1/3} = 28^{36/3} = 28^{12} = 26(\mathrm{mod}37).$$

Значит, $x_0 = 1$, *так как* $26 = r_{3,1}$.

Продолжаем вычисления:

$$y_1 = y \cdot 2^{-x_0} = 28 \cdot 2^{-1} = 14(\mathrm{mod}37),$$

$$y_1^{q-1/3^2} = 14^{36/9} = 14^4 = 10(\mathrm{mod}37).$$

Значит, $x_1 = 2$, *так как* $10 = r_{3,2}$.

Таким образом,

$$x = x(3) = 1 + 2 \cdot 3 = 7(\mathrm{mod}3^2).$$

Составляем систему

$$\begin{cases} x & = & 2(\mathrm{mod}4), \\ x & = & 7(\mathrm{mod}9). \end{cases}$$

Единственным решением системы в пределах $0 \leq x \leq 36$ *будет* $x = 34$.

Итак,

$$2^{34} = 28(\mathrm{mod}37), log_2 28 = 34.$$

Задачи и упражнения

Задача 9.1. Вычислить методом Сильвера – Полига – Хеллмана дискретный логарифм элемента 25 в поле $\mathbf{F}_{41}$ относительно порождающего элемента $g = 7$.

Задача 9.2. Построить поле $\mathbf{F}_{7^3}$. Найти порождающий элемент g его мультипликативной группы $\mathbf{F}_{7^3}^*$. Вычислить дискретный логарифм $log_g f$ по основанию g элемента $f = 5x^2 + 2x + 6$ методом Сильвера – Полига – Хеллмана.

Задача 9.3. Построить поле $\mathbf{F}_{3^3}$. Найти порождающий элемент g его мультипликативной группы $\mathbf{F}_{3^3}^*$. Вычислить дискретный логарифм $log_g f$ по основанию g элемента $f = x^2 + 2x + 2$ методом Сильвера – Полига – Хеллмана.

Задача 9.4. Построить поле $\mathbf{F}_{3^4}$. Найти порождающий элемент g его мультипликативной группы $\mathbf{F}_{3^4}^*$. Вычислить дискретный логарифм $log_g f$ по основанию g элемента $f = x^3 + 2x^2 + 2x + 2$ методом Сильвера – Полига – Хеллмана.

Задача 9.5. Построить поле $\mathbf{F}_{43}$. Найти порождающий элемент g его мультипликативной группы $\mathbf{F}_{43}^*$. Вычислить дискретный логарифм $log_g f$ по основанию g элемента $f = 11$ методом Сильвера – Полига – Хеллмана.

Задача 9.6. Продемонстрировать версию алгоритма Сильвера – Полига – Хеллмана вычисления дискретного логарифма в поле $\mathbf{F}_{17}$ для случая $3^x = 15(mod 17)$.

Задача 9.7. Построить конкретный протокол Диффи – Хеллмана с платформой $\mathbf{F}_{27}^*$.

Задача 9.8. Пусть дано простое конечное поле $\mathbf{Z}_{43}$. Предположим, что g – порождающий элемент его мультипликативной группы. Сколько всего имеется порождающих элементов и как их можно получить из g?

Задача 9.9. Предлагается следующее упрощение протокола Масси-Омуры. При выбранном конечном поле F Алиса осуществляет передачу сообщения $m \in F$ Бобу. Она выбирает секретный элемент $a \in F^*$ и передает элемент ma. Боб выбирает секретный элемент $b \in F^*$ и передает элемент mab. Алиса умножает на a^{-1} и передает mb. Боб вычисляет m, умножая полученное сообщение на b^{-1}. Какова основная слабость протокола? Можно ли его улучшить, заменив платформу F на кольцо вычетов $\mathbf{Z}_n$?

Криптосистема с открытым ключом Ривеста – Шамира – Адлемана

1. Идея шифрования с открытым ключом

Речь пойдет о введенной в рассмотрение в 1977 г. криптосистеме RSA (опубликовано в [26]), названной так в честь ее создателей: Rivest, Shamir, Adleman. Алгоритм шифрования в системе RSA является первым из алгоритмов шифрования с открытым ключом. Так называют алгоритмы, в которых из двух ключей – шифрования и дешифрования – секретным является только ключ дешифрования. Идея криптографии с открытым ключом впервые появилась в знаменитой короткой заметке Диффи и Хеллмана [21]. Прошло уже 30 лет, однако криптосистема RSA по-прежнему остается наиболее популярной из всех криптосистем с открытым ключом.

Итак, в чем же состоит идея шифрования с открытым ключом, что необходимо для построения соответствующей криптосистемы и в чем заключается ее преимущество перед классическими криптосистемами, в которых как ключ шифрования, так и ключ дешифрования являются секретными?

Вначале напомним, что называется односторонней функцией. Пусть $f : A \to B$ функция, область определения A и область значений B которой могут быть произвольными. Пусть выполнены следующие условия:

1) вычисление значения $b = f(a)$ для любого аргумента $a \in A$ выполняется достаточно просто;

2) нахождение по данному элементу $b \in B$ такого элемента a, что верно равенство $b = f(a)$, в общем случае – достаточно трудная задача.

Тогда мы называем f *односторонней функцией.*

Для целей криптографии слово «просто» по отношению к вычислению $b = f(a)$ из условия 1) означает, что существует компьютерная программа, которая осуществляет такое вычисление за реальное время. Понятие «реальное время», в свою очередь, зависит от важности решаемой задачи, мощности используемого процессора и т. п. Противоположный смысл имеют слова «трудная задача» из условия 2).

Для системы шифрования с открытым ключом e необходима односторонняя взаимно однозначная функция $f_e : A \to B$. Однако этого недостаточно. Необходима выполнимость еще одного свойства:

3) существует такой секрет, при знании которого нахождение аргумента a из уравнения $f(a) = b$ становится достаточно простой задачей.

Действительно, обладатель секрета (обычно это секретный ключ d дешифрования) – это адресат шифровки, кому она собственно предназначена. Для него процесс дешифрования должен быть реальным.

Поначалу кажется, что условия 2) и 3) противоречат друг другу. Это одна из главных причин столь позднего с исторической точки зрения появления систем шифрования с открытым ключом. Односторонние функции, удовлетворяющие кроме условий 1) и 2) еще и условию 3), называются односторонними функциями с секретом. Их не так-то просто найти. Хорошие примеры можно перечислить на пальцах. В дальнейшем мы еще вернемся к их обзору.

Поговорим о преимуществах. Они очевидны. Криптосистема шифрования с открытым ключом позволяет передавать секретные сообщения от одного корреспондента к другому без предварительного обмена секретными данными. Публикация открытых ключей по типу телефонной книги делает это не только возможным, но и достаточно широко применимым. Одним из первых использований такой системы RSA, о которой мы подробно поговорим далее, являлось создание на ее основе цифровой (электрон-

ной) подписи, что, в свою очередь, позволило выпустить первые электронные карты. Другая область применения – протоколы распределения ключей и аутентификации. В настоящее время все это широко используется в финансовой сфере, электронной торговле и т. п.

2. Криптосистема RSA

Приведем основные этапы работы RSA.

Установка

Вначале выбираются два достаточно больших простых числа p и q и вычисляется модуль $n = pq$. Числа p, q секретны (сверхсекретны), модуль n открыт. Далее вычисляется функция Эйлера $\varphi(n) = (p-1)(q-1)$, выбирается число e такое, что нод$(e, \varphi(n)) = 1$. Параметр e служит открытым ключом шифрования. Значение $\varphi(n)$ держится в секрете. Взаимная простота e и $\varphi(n)$ позволяет однозначно вычислить число d такое, что

$$e \cdot d = 1 (\mathrm{mod} \varphi(n)).$$

Говоря об однозначности, мы имеем в виду выполнение неравенства $1 \leq d \leq \varphi(n) - 1$. Другими словами можно сказать, что d – обратный элемент к e в группе $\mathbf{Z}^*_{\varphi(n)}$. Значение d является секретным ключом дешифрования.

Шифрование

В качестве платформы для единиц исходного текста выберем кольцо вычетов $\mathbf{Z}_n$. Единицами шифрованного текста также будут элементы кольца $\mathbf{Z}_n$. Единица шифротекста c получается из единицы исходного текста m по следующему правилу:

$$c = m^e (\mathrm{mod} n). \tag{2.1}$$

Обращаем внимание на то, что полученный вычет c записывается стандартным именем, то есть $0 \leq c \leq n-1$.

Дешифрование

Для дешифрования c достаточно знать значение секретного ключа d. Дешифрование осуществляется следующим образом:

$$m = m^{ed} (\mathrm{mod} n). \tag{2.2}$$

Объясним, как получилось только что приведенное равенство. Пусть вначале единица m взаимно проста с n. Тогда $m \in \mathbf{Z}_n^*$, то есть является элементом мультипликативной группы $\mathbf{Z}_n^*$ кольца вычетов $\mathbf{Z}_n$, порядок которой равен $\varphi(n)$. По теореме Эйлера $m^{\varphi(n)} = 1$. Равенство $ed = 1 (\mathrm{mod} \varphi(n))$ равносильно существованию числа $k \in \mathbf{Z}$, для которого $ed = 1 + \varphi(n)k$. Вычисляем:

$$c^d = m^{ed} = m(m^{\varphi(n)})^k = m \cdot 1^k = m (\mathrm{mod} n).$$

Может показаться с первого взгляда, что условие нод$(m, n) = 1$ является необходимым при RSA-шифровании. В ряде учебников по криптографии это условие признается необходимым, но на самом деле это требование излишне. Равенство $m = m^{ed} (\mathrm{mod} n)$ справедливо для любого m. Действительно, если $m \vdots n$, то оно принимает вид $0 = 0^{ed} (\mathrm{mod} n)$, что очевидно верно. Рассмотрим случай, когда m делится на один из множителей, скажем p, числа n и не делится на другой. Пусть, например, $m = p^t m_1$, где нод$(m_1, p) = 1$. Мы уже знаем, что $m_1^{ed} = m_1 (\mathrm{mod} n)$. Поэтому достаточно проверить справедливость сравнения

$$p^{ed} = p (\mathrm{mod} n).$$

Так как $n = pq$ – произведение различных простых чисел, это сравнение равносильно системе

$$\begin{cases} p^{ed} = p (\mathrm{mod} p), \\ p^{ed} = p (\mathrm{mod} q). \end{cases}$$

Здесь первое сравнение очевидно, а второе уже установлено. Можно шифровать любое значение m!

Пример 31. (Пример конкретного RSA-шифра)

Пусть мы работаем с 26-буквенным алфавитом английского языка $A - Z$, в котором буквы занумерованы по порядку начиная с $0 : A \leftrightarrow 0, B \leftrightarrow 1, \cdots, Z \leftrightarrow 25$. Считаем, что единицами исходного текста служат блоки из 3-х букв – 3-графы, а единицами шифрованного текста – 4-графы. Покажем, как осуществляется оцифровка единиц текста, на которые разбивается исходный текст. Найдем, например, численное значение слова YES. При нашей нумерации $Y \leftrightarrow 24, E \leftrightarrow 4, S \leftrightarrow 18$. Считаем, что набор $(24, 4, 18)$ – это запись искомого ключа в 26-ричной системе, то есть $YES \leftrightarrow x = 24 \cdot (26)^2 + 4 \cdot 26 + 18 = 16346$. Подобным образом цифруется каждый 3-граф. Затем мы переходим к шифровке полученных значений. Для этого выберем модуль n как произведение простых чисел $p = 281, q = 167; n = pq = 46927$. Имеем $\varphi(n) = (p-1)(q-1) = 46480$. Выберем случайное число e взаимно простое с $\varphi(n)$, скажем, $e = 46377$. Найдем d из условия $ed = 1(mod(\varphi(n)) : d = 18953$. Модуль (открытый) $n = 46927$, значение функции Эйлера $\varphi(n) = 46480$, открытый ключ шифрования $e = 46377$, секретный ключ дешифрования $d = 18953$.

Для шифровки сообщения, например $YES \longleftrightarrow x = 16346$, нужно вычислить $c = x^e (\mathrm{mod} n)$. Затем, если это необходимо, записать буквенное выражение. Для дешифровки нужно вычислить $c^d (\mathrm{mod} n)$.

Замечание о выборе ключа дешифрования

При выборе ключа дешифрования d мы руководствовались теоремой Лагранжа, согласно которой для любого $m \in \mathbf{Z}_n^*$ имеем $m^{\varphi(n)} = 1(\mathrm{mod} n)$. В данном случае величина $\varphi(n)$ является порядком мультипликативной группы $\mathbf{Z}_n^*$. Создается впечатление, что полученный таким способом d является единственным по $\mathrm{mod} n$ ключом, дешифрующим все возможные тексты. Это впечатление обманчиво. Если элемент $m \in \mathbf{Z}_n^*$ имеет порядок g, то есть $m^g = 1(\mathrm{mod} n)$ и g – наименьшее натуральное число с указанным свойством, то расшифровать текст $c = m^e (\mathrm{mod} n)$ можно

путем возведения c в степень f такую, что $ef = 1(\bmod g)$. Доказательство по сути то же самое, только необходимо всюду заменить $\varphi(n)$ на g. В этом случае величина f может оказаться существенно меньше, чем d. Из сказанного можно заключить, что расшифровку текстов m малого порядка в группе $\mathbf{Z}_n^*$ можно осуществить подбором f. Конечно, f зависит от m. Но мы можем ввести в рассмотрение функцию $\tau(n)$, равную по определению наименьшему общему кратному всех порядков элементов из $\mathbf{Z}_n^*$. Поскольку любой порядок f по теореме Лагранжа делит $\varphi(n)$, величина e остается взаимно простой с $\tau(n)$. Ключ $\bar{d}$, вычисленный из равенства $e\bar{d} = 1(\mathrm{mod}\tau(n))$, дешифрует любую шифровку $c = m^e(\mathrm{mod}n)$. Равенство $c^{\bar{d}} = m(\mathrm{mod}n)$ устанавливается точно так же, как и раньше. Заметим, что $\tau(n)$ всегда строго меньше, чем $\varphi(n)$. Действительно, легко заметить, что $\tau(n)$ делит $\delta = \text{нок}(p-1, q-1)$, так как очевидные равенства $m^\delta = 1(\mathrm{mod}p), m^\delta = 1(\mathrm{mod}q)$ влекут в силу взаимной простоты чисел p и q равенство $m^\delta = 1(\mathrm{mod}n)$. Из-за того, что оба числа $p-1$ и $q-1$ четные, уже величина $\tau(n)$ строго меньше, чем $\varphi(n)$. Если же числа $p-1$ и $q-1$ имеют общие большие делители, $\tau(n)$ может оказаться существенно меньше, чем $\varphi(n)$, что может привести к значительной разнице между d и $\bar{d}$.

Итак, мы закончили криптографический этап построения криптосистемы RSA. Остается уделить время криптоанализу этой системы, то есть выяснить ее криптостойкость, объяснить, как правильно выбирать участвующие в ее установке параметры, и т. п.

3. Сложные задачи, обеспечивающие криптостойкость RSA

Во-первых, следует сказать о сложности разложения модуля n на множители. В настоящее время не известно ни одного эффективного алгоритма, позволяющего в общем случае находить множители разложения $n = pq$. Впрочем, нет и доказательства его несуществования. Если бы такой алгоритм был найден, система RSA сразу же вышла бы из употребления. Действительно, знание

p, q позволяет получить функцию Эйлера $\varphi(n) = (p-1)(q-1)$, а затем алгоритмом Эвклида вычислить секретный ключ d как обратный элемент к открытому ключу e в группе $\mathbf{Z}^*_{\varphi(n)}$.

Впрочем, для вычисления ключа d достаточно знания одной лишь функции Эйлера $\varphi(n) = (p-1)(q-1) = pq - (p+q) + 1 = n - (p+q) + 1$. Но в этом случае нам становится известным как произведение $n = pq$, так и сумма $p + q = n - \varphi(n) + 1$. Это, конечно, дает возможность найти p и q.

Отсюда вытекает, что умение разлагать на множители модуль $n = pq$ равносильно умению вычислять каким-то образом функцию Эйлера $\varphi(n)$. И то, и другое опять же приводит к полному рассекречиванию системы RSA.

Посмотрим, что еще позволило бы рассекретить систему RSA. Допустим, например, что мы знаем эффективный способ вычисления всех корней второй степени из 1 в группе $\mathbf{Z}^*_n$. Во-первых, заметим, что нахождение двух таких корней не составляет труда: $y_1 = 1, y_2 = -1 = n - 1 (\mathrm{mod} n)$. Попробуем доказать, что всего таких корней четыре.

Перечислим вычеты $\mathrm{mod} n$, сравнимые с 1 и -1 $\mathrm{mod} p$:

$$1, 1+p, 1+2p, \ldots, 1+(q-1)p;$$

$$-1, -1+p, -1+2p, \ldots, -1+(q-1)p.$$

Аналогично выпишем все вычеты $\mathrm{mod} n$, сравнимые с 1 и -1 $\mathrm{mod} q$:

$$1, 1+q, 1+2q, \ldots, 1+(p-1)q;$$

$$-1, -1+q, -1+2q, \ldots, -1+(p-1)q.$$

Если вычет $\mathrm{mod}\ n$ сравним с $\pm 1 (\mathrm{mod}\ p)$ и одновременно с $\pm 1 (\mathrm{mod} q)$, то его квадрат сравним с $1 (\mathrm{mod} n)$. Верно и обратное утверждение. Итак, мы должны найти в выписанных наборах одинаковые элементы. Равенства $1 + pi = 1 + qj, -1 + pi = -1 + qj$, где $i = 0, 1, 2, \ldots, (q-1); j = 0, 1, 2, \ldots, (p-1)$, предоставляют нам два уже известных решения: $y_1 = 1, y_2 = -1$. Равенства

вида $1+pi=-1+qj, -1+pi=1+qj$ при тех же значениях параметров i,j дают еще два решения: y_3, y_4. Действительно, первое из них эквивалентно системе сравнений

$$\begin{cases} pi &= -2(\mathrm{mod}q), \\ qj &= 2(\mathrm{mod}p), \end{cases}$$

для которой $i=-p^{-1}2(\mathrm{mod}q), j=2q^{-1}(\mathrm{mod}p)$. В свою очередь, вся система имеет решение по Китайской теореме об остатках. Решение y_3, как легко видеть, единственное стандартное $\mathrm{mod}n$, так как все решения отличаются слагаемым, кратным $n=pq$. Из последнего равенства мы получим еще одно решение $y_4=-y_3$.

Итак, мы установили, что всего корней второй степени из 1 $\mathrm{mod}n$ ровно четыре: $y_1=1, y_2=-1, y_3, y_4=-y_3$. Если кто-то умеет их вычислять, он запишет сравнение

$$y_3^2-y_1^2=(y_3-1)(y_3+1)=0(\mathrm{mod}n).$$

Поскольку $y_3 \neq \pm 1(\mathrm{mod}n)$, возникающее при некотором целом s равенство

$$(y_3-1)(y_3+1)=sn$$

позволяет найти $d_1=$нод(y_3-1,n), $d_2=$нод(y_3+1,n) и получить разложение $n=d_1d_2$.

Из всего сказанного выше можно сделать вывод, что умение вычислять все четыре корня второй степени из 1 $\mathrm{mod}n$ равносильно умению разлагать модуль на множители: $n=pq$. В одну сторону, когда разложение получается через корни, мы это только что продемонстрировали. В другую все очень просто: записываем $\mathrm{mod}n$ наборы корней из 1 $\mathrm{mod}p$ и $\mathrm{mod}q$, как это делалось выше, и решаем возникающие сравнения вида $1+pi=-1+qj, -1+pi=1+qj$, для $i=1,2,\ldots,q-1; j=1,2,\ldots,p-1$. При этом, естественно, нужно знать разложение $n=pq$.

Мы перечислили ряд задач, эффективное решение которых привело бы к рассекречиванию системы RSA в общем случае. Пока такое решение не найдено.

4. Правильный выбор параметров

Очевидно, что исходные параметры установки криптосистемы RSA – простые числа p и q – должны быть большими. В настоящее время считается, что, кроме этого требования, необходимо, чтобы разность $p-q$ (при $p>q$) также была бы достаточно большой. Более подробно мы поговорим об этом в лекции 12, посвященной разложению больших составных чисел на множители. Отметим также, что из Замечания о выборе ключа дешифрования, приведенного выше, следует, что числа $p-1, q-1$ должны иметь большие различные простые делители, а нод$(p-1, q-1)$ должен быть небольшим.

Обратимся к вопросу: как не нужно выбирать параметры RSA в системах многочисленных пользователей? Отметим здесь наиболее очевидные рекомендации.

Во-первых, модули n_1, n_2 различных пользователей должны быть взаимно простыми: нод$(n_1, n_2) = 1$. Действительно, пусть, например, $n_1 = pq_1, n_2 = pq_2$, где $q_1 \neq q_2$. Тогда нод$(n_1, n_2) = p$, и это значение, а значит и разложения: $n_1 = pq_1, n_2 = pq_2$, может вычислить любой знающий открытые данные n_1, n_2. Менее очевидно объяснение для случая совпадения модулей: $n_1 = n_2$. Если совпадают ключи: $e_1 = e_2$ и $d_1 = d_2$, то такие пользователи просто напросто неразличимы с внешней стороны, но они знакомы со всеми секретами друг друга, что, конечно, следует учитывать. Предположим теперь, что $e_1 \neq e_2$, значит, $d_1 \neq d_2$. Друг для друга пользователи, о которых идет речь, полностью открыты, поскольку знание обоими функции Эйлера $\varphi(n)$ позволяет по e_1 вычислить d_1, а по e_2 найти d_2.

Посмотрим, что можно рассекретить извне. Для этого предположим, что указанные пользователи получили одно и то же сообщение m:

$$c_1 = m^{e_1}, c_2 = m^{e_2} (\mathrm{mod} n).$$

Извне можно вычислить $t_1 = e_1^{-1} (\mathrm{mod}\, e_2)$, что эквивалентно равенству вида $t_1 e_1 = 1 + l e_2$ в Z. Зная $t_1 e_1 - 1$ и e_2, можно вычислить l.

Тогда

$$c_1^{t_1}c_2^{-l} = m^{e_1t_1}m^{-le_2} = m^{1+le_2}m^{-le_2} = m(\mathrm{mod} n).$$

Значит, извне можно прочитать сообщение m.

Все вышесказанное позволяет предположить, что различные пользователи $1, 2, \ldots$ некоторой системы имеют попарно взаимно простые модули RSA: $n_1, n_2, \ldots$. Допустим, что k пользователей выбрали один и тот же открытый ключ шифрования $e = k$ (такое легко может произойти, если выбираются небольшие значения для $e_1, e_2, \ldots$). Предположим, что все пользователи получили сообщение m, тогда:

$$\begin{cases} c_1 = m^k \pmod{n_1}, \\ c_2 = m^k \pmod{n_2}, \\ \qquad\qquad \ldots, \\ c_k = m^k \pmod{n_k}, \end{cases}$$

где для простоты мы считаем пользователей ключом $e = k$ первыми. Так как нод$(n_i, n_j) = 1$ при $i \neq j$, можно воспользоваться Китайской теоремой об остатках и вычислить такое число $0 \leq s \leq \leq n_1n_2 \ldots n_k$, для которого имеют место сравнения:

$$\begin{cases} c_1 = s \pmod{n_1}, \\ c_2 = s \pmod{n_2}, \\ \qquad\qquad \ldots, \\ c_k = s \pmod{n_k}. \end{cases}$$

Заметим, что $0 \leq m < n_i$ для любого i. Значит, $0 \leq m^k \leq \leq n_1n_2 \ldots n_k$. Все другие возможные для s значения отличаются друг от друга не менее чем на $n_1n_2 \ldots n_k$. Отсюда получаем, что $s = m^k$ как целые числа. Число m находится простым извлечением из s корня степени k, что в $\mathbf{Z}$ делается достаточно легко.

5. Битовая стойкость RSA

Предположим, что существует эффективный способ определения по любой шифровке

$$c = m^e (\mathrm{mod} n)$$

четности исходного текста m, относительно которого мы, как всегда, предполагаем, что $0 \leq m < n$. Если записать число m в двоичной системе как

$$m = \varepsilon_0 + \varepsilon_1 2 + ... \varepsilon_k 2^k,$$

где $\varepsilon_i \in \{0, 1\}, i = 1, ..., k$, то легко видеть, что его четность определяется значением младшего бита ε_0. При $\varepsilon_0 = 0$ число m четное, при $\varepsilon_0 = 1$ – нечетное. Ясно, что эффективный способ вычисления m в целом позволяет эффективно вычислить ε_0. Оказывается, и это уже не столь очевидно, что верно также обратное утверждение: эффективный способ вычисления четности любого текста по его шифровке дает эффективный способ вычисления произвольного фиксированного текста. Дадим его описание.

Прежде всего заметим, что в силу нечетности числа n для любого $0 \leq m < n$ значение стандартного имени числа $2m$ является четным в том и только том случае, если $m \in [0, n/2)$. Действительно, если m находится в обозначенной левой половине интервала $[0, n)$, то число $2m < n$, очевидно, четное. В противном случае $2m = k + n$, где k – его стандартное имя, очевидно, нечетное.

Полагаем теперь

$$c_1 = 2^e c = (2m)^e (\mathrm{mod} n).$$

Шифровка c_1 легко вычисляется в силу открытости ключа шифрования e. По предположению, эффективно определяется четность стандартного имени текста $2m$. В случае четности пишем

$$0 \leq m < \frac{n}{2},$$

в случае нечетности –

$$\frac{n}{2} < m \leq n-1.$$

Таким образом эффективно определяется принадлежность текста левой или правой целочисленной половине интервала $[0, n)$.

Дальнейший процесс строится аналогичным образом. Если, например, на предыдущем шаге выяснено включение $m \in [0, n/2)$, то стандартное имя текста $4m$ будет четным, если только $m \in \in [0, n/4)$. Это определяется формированием шифровки

$$c_2 = 2^{2e}c = (2^2 m)^e (\mathrm{mod} n).$$

Если, напротив, на предыдущем шаге оказалось, что $m \in (n/2, n-1]$, то рассматриваются подинтервалы $(n/2, 3n/4)$ и $(3n/4, n-1]$. Стандартное имя текста $4m$ будет четным, если только $m \in (n/2, 3n/4)$. Данный процесс легко формализуется, но мы рассмотрим его работу на конкретном примере. Заметим, что число m определяется примерно через $log_2 m$ шагов. При этом можно исключить шаги, когда последовательное деление интервала на два подинтервала может реально его не разделить. Это также видно на примере.

Пример 32. *Пусть* $n = 3 \cdot 7 = 21, \varphi(n) = 2 \cdot 6 = 12, e = 5, c = 3$. *Рассмотрим шифровку*

$$m^5 = 3(\mathrm{mod} 21).$$

Легко видеть, что $m = 12$. *Покажем, как определить* m *указанным выше процессом.*

Вычислим значения

$$c_i = 2^{i5}c = (2^i m)^5 (\mathrm{mod} 21)$$

для $i = 1, ..., 7$. Каждый раз смотрим на четность стандартного имени числа $2^i m$ (вычисления идут $\mathrm{mod} n$):

$$2m = 3, 2^2m = 6, 2^3m = 12, 2^4m = 3, 2^5m = 6, 2^6m = 12, 2^7m = 3,$$

что дает следующий набор четностей: н, ч, ч, н, ч, ч, н.

Последовательно получаем неравенства:

$$11 \leq m \leq 20, 11 \leq [3n/4] = 15, 11 \leq m \leq [5n/8] = 13,$$

$$\mathbf{11} \leq m \leq [19n/32] = \mathbf{12}, 11 \leq m \leq [38n/64] = 12,$$

$$\mathbf{12} \leq m \leq [77n/128] = 12.$$

Результат: $m = 12$. Мы видим, что в двух случаях интервалы повторились. Это происходит только на небольших интервалах. Поэтому можно считать, что необходимо проделать примерно $[log_2 n]$ шагов. Либо исключить проверки в случаях, если подинтервал совпадет с интервалом.

6. Семантическая секретность

В лекции 5 рассматривалась модель К. Шеннона, в которой множества исходных текстов M, шифровок C и ключей K являются вероятностными пространствами. Важное значение в этой модели имеют условные вероятности $p(m|c)$ выбора исходного текста m по данной шифровке c. Совершенный шифр требует равенства $p(m) = p(m|c)$, что означает отсутствие дополнительной информации о распределении в пространстве M при известной шифровке c.

Если рассматривается система шифрования с открытым ключом e, например, система RSA, в которой $c = m^e (\mathrm{mod} n)$, то всегда можно проверить для данного текста m' выполнимость равенства $c = m'^e (\mathrm{mod} n)$. Раз так, мы уже не можем говорить об отсутствии дополнительной информации при данной шифровке.

Однако и в этом случае рассматриваются различные модели, связанные с вычислительной секретностью. Опишем один из возможных подходов.

Будем считать, что некоторая машина (алгоритм) A определяет по натуральному параметру k тексты из пространства M: $A(k) = m$. Пусть имеется функция f, определенная на пространстве M. Допустим, что существует некоторый алгоритм B, определяющий по шифровке $c = E(m)$ и открытому ключу e значение $f(m)$ с определенной вероятностью $p_B(k)$. Это означает, что алгоритм вычисляет по $c = E(m)$ и открытому ключу e значение x, и если $f(m) = x$, то B принимает значение 1, если нет – то 0. Вероятности 1 и 0 определены. Система шифрования называется *семантически секретной*, если для любого такого алгоритма B найдется подобный алгоритм C, но уже вычисляющий без знания шифровки c, для которого разность вероятностей правильного вычисления значения функции f пренебрежимо мала по отношению к k. Например, выполняется неравенство

$$|p(B, f) - p(C, f)| \leq \frac{1}{k^{100}}.$$

Здесь $p(B, f)$ означает вероятность правильного вычисления значения функции f алгоритмом B аналогично для C.

Часто в подобную схему добавляют требование полиномиальной вычислимости относительно задействованных алгоритмов.

Задачи и упражнения

Задача 10.1. Пусть $n = pq$ – модуль системы RSA. Пусть e – открытый ключ шифрования. Предположим, что Боб знает порядок элемента e в группе $\mathbf{Z}^*_{\varphi(n)}$ и этот порядок равен k. Как Боб может дешифровать сообщение $m^e(mod n)$ и даже вычислить изначально секретный ключ d?

Задача 10.2. (Возможность использования общего модуля и распределения ключей RSA доверительным Центром.) Иногда RSA распределяется Центром, который, зафиксировав модуль $n = pq$, раздает пользователям ключи (e_i, d_i), $i = 1, ..., k$. Параметры p, q держатся Центром в секрете. Показать, что любой пользователь системы может определить все ключи d_i других пользователей за полиномиальное время.

Задача 10.3. (Неподвижные точки при шифровании RSA.) При шифровании в системе RSA $(n = pq, e, d)$ возможны совпадения шифровки c и исходного текста m. Покажите, что число совпадений равно $(1+$ нод$(p-1, e-1))(1+$нод $(q-1, e-1))$. Какой вывод можно сделать для правильного выбора параметров p, q, e?

Задача 10.4. Пусть $n = pq$ – модуль системы RSA, e – открытый ключ шифрования. Предположим, что существует эффективный способ вычисления по шифровке любого текста $m \in \mathbf{Z}_n$ второго младшего бита ε_0 его двоичной записи. Можно ли аналогично тому, что было показано в пункте 5 данной лекции, определить по шифровке произвольный фиксированный текст m? Можно ли сделать то же самое, если мы умеем эффективно определять бит $\varepsilon_i, i \in \mathbf{N}$, любого текста m?

Задача 10.5. Пусть в системе RSA выполнено равенство $c^f = m(\mathrm{mod} n)$, где $c = m^e(\mathrm{mod} n)$ – шифровка. Можно ли утверждать, что $f = d$, где d – ключ дешифрования?

Задача 10.6. Алиса вычисляет (за деньги) аргумент m RSA-функции шифрования. То есть она получает при открытых данных (n, e) значение $c = m^e (\mathrm{mod} n)$ и после платежа сообщает обратившемуся величину m. Боб хочет расшифровать $c = m^e (\mathrm{mod}\ n)$, но он не желает, чтобы в процессе расшифровки Алиса узнала m. Описать, как Боб может организовать обращение к Алисе, чтобы в результате получить m, не раскрывая его значения для Алисы.

Задача 10.7. Допустим, что в системе RSA передана не только шифровка $c = m^e (\mathrm{mod} n)$, но и сообщение вида $\tilde{c} = m^{2^k} (\mathrm{mod}\ n)$. Может ли третье лицо прочитать сообщение m?

Задача 10.8. Модуль RSA равен $n = 24637$, открытый ключ $e = 3$. Вычислить ключ дешифрования d.

Простые числа

1. Потребность в простых числах

В криптографии простые числа встречаются буквально на каждом шагу. Для выбора конкретного шифра криптосистемы RSA необходимо иметь два достаточно больших различных простых числа p и q, по которым вычисляется открытый модуль $n = pq$. Числа p, q являются секретными, поэтому они должны выбираться правильно: с одной стороны, случайным образом, с другой – в нужных пределах и с определенными ограничениями на выбор. Дискретный логарифм используется в самых различных криптосистемах и протоколах. В качестве платформы для дискретного логарифма часто берутся мультипликативные группы $\mathbf{Z}_p^*$ простых полей $\mathbf{Z}_p$, элементами которых являются вычеты по простому модулю p. Выбор платформы, таким образом, – это выбор достаточно большого простого числа p, на которое также обычно накладывается ограничение. Если в качестве платформы используется мультипликативная группа $\mathbf{F}_{p^r}^*$ конечного поля порядка p^r, то само оно строится известным образом, исходя из простого поля $\mathbf{Z}_p$.

Существует три основных способа выбора простого числа p: поиск его в какой-либо базе данных (такие существуют, и достаточно объемные), генерация p каким-либо образом или выбор p последовательным рассмотрением кандидатов – больших нечетных чисел. Мы остановимся на двух последних способах. Тем более, что они хорошо известны и достаточно эффективны.

2. Проверка на простоту

Допустим, нам необходимо выяснить, является ли большое нечетное число n простым. Конечно, можно сразу предположить,

что n не делится на небольшие простые числа: $3, 5, 7, 11$, для которых существуют известные правила делимости. Можно продолжить этот ряд, добавив еще обозримое множество простых, так как проверка на делимость числа n на данное число m выполняется компьютером достаточно просто. Это число m может быть и большим. Проблема в том, что возможных простых делителей у числа n достаточно много.

Легко видеть, что в возможном разложении $n = ab$ один из множителей не превосходит $\sqrt{n}$. Итак, первым шагом достаточно грубой проверки n на простоту является выяснение, делится ли n на достаточно небольшие простые числа и, возможно, на некоторые избранные числа.

Следующий шаг связан с проверкой на псевдопростоту. Эта проверка тоже достаточно поверхностная, польза ее состоит в том, что с ее помощью убираются из рассмотрения некоторые «явно» составные числа n.

Согласно Малой теореме Ферма для любого простого p и любого числа a, взаимно простого с p: $\text{нод}(a, p) = 1$, выполнено сравнение

$$a^{p-1} = 1 (\mathrm{mod} p), \tag{2.1}$$

что, как мы видим в лекции 7, следует из более общей теоремы Лагранжа. Отсюда мы получаем первый – самый простой тест на вероятную простоту.

Определение 33. *Число n называется псевдопростым относительно взаимно простого с ним числа b: $\text{нод}(b, n) = 1$, если выполнено сравнение*

$$b^{n-1} = 1 (\mathrm{mod} n). \tag{2.2}$$

Можно, конечно, считать что $1 \leq b \leq n - 1$. Заметим, что если n действительно является простым числом, то оно выдержит тест для любого числа $1 \leq b \leq n - 1$. Значит, нарушение (2.2) хотя бы для одного числа b указанного вида влечет непростоту n. Если для достаточно большого выбора чисел $\overline{b} = \{b_1, b_2, \ldots, b_k\}$,

нод$(b_i, n) = 1 (i = 1, \ldots, k)$ мы обнаружим, что n псевдопростое относительно каждого из них, то мы можем предположить вероятную простоту числа n. Может показаться, что вероятность такого суждения тем больше и ближе к 1, чем больше набор тестовых чисел $b_1, b_2, \ldots, b_k$ (мы считаем их различными). Однако такое суждение не может быть подкреплено точными оценками. Дело в том, что существуют составные числа n псевдопростые по отношению к любому $1 \leq b \leq n-1$, если нод$(b, n) = 1$. Такие числа называются *числами Кармайкла*. Сравнительно недавно [16] удалось установить, что чисел Кармайкла бесконечно много.

Предложение 34. *Нечетное число n будет числом Кармайкла, если и только если выполнены следующие условия:*

1) $n = p_1 p_2 \ldots p_k$,

где p_i - различные простые (нечетные) числа, то есть n не делится на квадрат простого числа;

2) $p_i - 1$ делит $n - 1$ для любого $i = 1, \ldots, k$.

Доказательство. Докажем достаточность условий 1), 2), так как они позволяют находить числа Кармайкла и проверять на это свойство большие нечетные числа n.

Итак, пусть $n-1 = (p_i - 1)l_i$ $(i = 1, \ldots, k)$. Пусть число $1 \leq b \leq n-1$ взаимно просто с n: нод$(b, n) = 1$. Тогда по Малой теореме Ферма

$$b^{p_i - 1} = 1 (\mathrm{mod} p_i) \tag{2.3}$$

для всех $i = 1, \ldots, k$. Тогда

$$b^{n-1} = b^{(p_i-1)l_i} = 1 (\mathrm{mod} p_i) \tag{2.4}$$

для всех $i = 1, \ldots, k$. Последнее означает, что $b^{n-1} - 1$ делится на $p_1, \ldots, p_k$. Но в этом случае $b^{n-1} - 1$ делится также на произведение $n = p_1 \ldots p_k$, то есть выполнено (2.2).

Пример 35. *Числом Кармайкла является:*

$$n = 561 = 3 \cdot 11 \cdot 17.$$

Замечание 36. *Число Кармайкла не может быть произведением двух различных простых чисел. Действительно, если* $n = pq$ *(*p, q *– простые,* $p < q$ *), то* $n - 1 = pq - 1 = (p-1)(q-1) + (p-1) + (q-1)$. *Так как* $q - 1$ *не является делителем* $p - 1$ *(оно просто больше), то и* $n - 1$ *не делится на* $q - 1$.

Но мы еще не доказали, что число n, делящееся на квадрат простого числа, скажем, на p^2, не может быть числом Кармайкла. Легко вычисляется значение функции Эйлера $\varphi(p^2) = p(p-1)$. Заметим, что группа $\mathbf{Z}^*_{p^2}$ циклическая. Если f – порождающий элемент циклической группы $\mathbf{Z}^*_p$, то в качестве порождающего элемента циклической группы $\mathbf{Z}^*_{p^2}$ можно взять либо $g = f$, либо $g = f(p+1)$. Доказательство оставляем читателю.

Итак, пусть g – порождающий элемент циклической группы $\mathbf{Z}^*_{p^2}$. Пусть m – произведение всех простых делителей числа n, отличных от p. Согласно Китайской теореме об остатках существует число b, удовлетворяющее системе сравнений

$$\begin{cases} b = g(\mathrm{mod} p^2), \\ b = 1(\mathrm{mod} m). \end{cases} \tag{2.4}$$

Тогда b так же, как и g, является порождающим группы $\mathbf{Z}^*_{p^2}$. Заметим, что нод$(b, n) = 1$, что следует из (2.4). Если выполнено требование псевдопростоты $b^{n-1} = 1(\mathrm{mod} n)$, то верно сравнение $b^{n-1} = 1(\mathrm{mod}\, p^2)$. Порядок элемента b в группе $\mathbf{Z}^*_{p^2}$ равен порядку группы $p(p-1)$. В этом случае $n-1$ должно по теореме Лагранжа делиться на число $p(p-1)$, что очевидно невозможно.

Предложение 34 доказано.

Итак, псевдопростота не гарантирует простоты на вероятностном уровне. И все из-за чисел Кармайкла!

Требуются другие способы проверки на простоту.

Остановимся на одном из наиболее известных вероятностных тестов на простоту.

Тест Миллера – Рабина (Miller – Rabin)

Этот тест основан на следующих соображениях. Пусть для некоторого числа $1 \leq b \leq n-1$ взаимно простого с тестируемым на простоту большим нечетным числом n: нод$(b,n) = 1$ выполнено равенство $b^{n-1} = 1 \pmod{n}$, то есть n – псевдопростое по отношению к b. Если такое равенство не выполнено, проверять нечего – n составное. Пусть

$$n - 1 = 2^s t, \tag{2.5}$$

где $s \geq 1, t$ нечетное. Если либо выполнено равенство $b^t = 1 (\mathrm{mod}\ n)$, либо найдется $0 \leq r \leq s-1$ такое, что $b^{2^r t} = -1 (\mathrm{mod} n)$, то n называется сильно псевдопростым по отношению к b.

Основание

Согласно Малой теореме Ферма для простого n в равенстве

$$b^{n-1} - 1 = (b^t - 1)(b^t + 1)(b^{2t} + 1)(b^{2^2 t} + 1) \ldots (b^{2^{s-1} t} + 1) = 0 (\mathrm{mod}\, n) \tag{2.6}$$

хотя бы одна из скобок равна $0 (\mathrm{mod} n)$.

Действительно, при простом n кольцо $\mathbf{Z}_n$ является полем, а в поле, как известно, нет делителей нуля: если $ab = 0$, то либо $a = 0$, либо $b = 0$. Это свойство очевидным образом продолжается на любое число множителей. Но тогда одна из скобок в (2.6) должна равняться 0, что приводит к определению сильной псевдопростоты.

Предложение 37. *Если n – нечетное составное число, то n является сильно псевдопростым по отношению к не более чем 25 % чисел $1 \leq b \leq n-1$: нод$(b,n) = 1$.*

Мы не приводим доказательства этого утверждения ввиду громоздкости и насыщенности его техническими деталями (см., например, [23]).

Замечание. Из практики известно, что для проверки на простоту большого нечетного числа n тестом Миллера – Рабина нет

необходимости перебирать большое количество элементов b. Подсчитано, например, что существует единственное разложимое на множители нечетное число $n \leq 2,5 \cdot 10^{10}$, а именно: $n = 3215031751$, которое выдерживает тест для $b = 2, 3, 5, 7$.

Существуют также другие вероятностные алгоритмы проверки на простоту больших натуральных чисел. Сравнительно недавно индийские математики М. Аграваль, Н. Кайал и Н. Саксена (M. Agrawal, N. Kayal and N. Saxena) [15] представили алгоритм, определяющий простоту натурального числа за полиномиальное время. Пока этот алгоритм, являющийся несомненно выдающимся теоретическим результатом, не нашел сколько-нибудь удовлетворительного практического применения. Все дело в том, что полиномиальность алгоритма еще не означает его практической применимости, ведь при реализации часто сказывается, что алгоритм работает очень медленно, создается большая база данных для его использования и т. п.

3. Построение больших простых чисел

Следующая теорема дает возможность эффективно строить большие простые числа.

Теорема 38. *Пусть n – нечетное число, $n - 1 = qr$, где q – нечетное простое число, r – четное число. Допустим, существует такое число a, что выполняются следующие условия:*

$$a^{n-1} = 1 (\mathrm{mod} n), \tag{1}$$

$$нод(a^{n-1/q} - 1, n) = 1. \tag{2}$$

Тогда для любого простого делителя p числа n выполнено сравнение $p = 1 (\mathrm{mod} 2q)$. В частности, $p \geq 2q + 1$.

Доказательство.

Условие 1) равносильно тому, что разность $a^{n-1}-1$ делится на n, значит, $a^{n-1}-1$ делится на p, что дает сравнение

1') $a^{n-1}=1(\mathrm{mod}p)$.

Условие 2) влечет, в частности, что верно утверждение

2') $a^{n-1/q}\neq 1(\mathrm{mod}p)$.

С другой стороны, по Малой теореме Ферма выполнено

3) $a^{p-1}=1(\mathrm{mod}p)$.

Переходя к группе $\mathbf{Z}_p^*$, получаем:

$$a^{n-1/q}\neq \mathbf{1}, a^{n-1}=1, a^{p-1}=1. \qquad (3.1)$$

Неравенство $a^{n-1/q}\neq 1$ вместе с равенством $(a^{n-1/q})^q=a^{n-1}=$ $=1$ показывает, что порядок элемента $a^{n-1/q}$ равен q. Значит, по теореме Лагранжа порядок элемента a делится на q. Отсюда и из равенства $a^{p-1}=1$ по теореме Лагранжа получаем: $p-1\vdots q$. Очевидно, что $p-1\vdots 2$, поэтому $p-1\vdots 2q$, что означает требуемое сравнение $p=1(\mathrm{mod}2q)$.

Теорема доказана.

Покажем теперь, как это утверждение можно использовать для построения больших простых чисел. Для этого приведем

Предложение 39. *Если выполнены условия предыдущей теоремы и кроме этого имеет место неравенство*

$$r\leq 4q+2,$$

то n – простое число.

Доказательство.

Пусть n – составное, тогда n делится на произведение хотя бы двух простых чисел p_1 и p_2 (возможно, одинаковых). Тогда, как доказано в теореме, $p_1, p_2\geq 2q+1$.

Произведем вычисления:

$$(2q+1)^2=4q^2+4q+1\leq n=qr+1\leq q(4q+2)+1=4q^2+2q+1,$$

противоречие.

Предложение доказано.

Алгоритм построения простых чисел

1) Стартуем с простого нечетного числа $q = q_1$.
2) Случайным образом выбираем четное r:

$$q \leq r \leq 4q + 2.$$

3) Полагаем

$$n = qr + 1.$$

4) Выбираем случайным образом число $a = a_1$ в пределах $1 < a < n-1$ и проверяем выполнимость условий (1), (2) из теоремы 38. Если $a = a_1$ не удовлетворяет этим условиям, то берем другое случайное число $a = a_2$. Так повторяем достаточное число раз: $a = a_1, a_2, \ldots, a_k$, пока не найдем подходящее значение a.

Если это удалось сделать, то n – простое. Полагаем $q = q_2 = n$ и повторяем построение, начиная с шага 1). Так делаем до тех пор, пока не получим достаточно большое простое число.

Если при большом числе проб для a так и не удалось выполнить условия 1), 2) теоремы 38, то изменяем r и повторяем все снова.

Замечание 40. *Предположим, что построенное число n – действительно простое. Зададимся вопросом, какова вероятность нахождения числа a с заданными свойствами* (1), (2) *из теоремы 38. Во-первых, отметим, что условие* (1) $a^{n-1} = 1 (\mathrm{mod} n)$ *будет автоматически выполнено по Малой теореме Ферма. В данном случае (при простом n) условие* (2) *нод*$(a^{n-1/q} - 1, n) = 1 \sim$ $\sim$ *нод*$(a^r - 1, n) = 1$ *выполнено в том и только том случае, если $a^r \neq 1 (\mathrm{mod} n)$. В поле $\mathbf{Z}_n$ уравнение $x^r = 1$ имеет не более r корней, один из которых равен 1, а второй $n-1$. Поэтому на промежутке $1 < a < n-1$ имеется не более $r-2$ чисел r, для которых $a^r = 1$ в поле $\mathbf{Z}_n$. Это означает, что вероятность выбора такого a не больше, чем* $\frac{r-2}{n-3} \sim \frac{r}{qr} = \frac{1}{q}$.

Заметим также, что таким образом построенное простое число n будет больше чем q^2, поскольку $q \leq r$ и $n = qr + 1$. При последовательном построении простых чисел $q_1, q_2, \ldots$ они растут не менее, чем квадратично.

Замечание 41. *Зададимся другим вопросом: насколько реально найти простое число $n = qr + 1$ при указанных ограничениях $q \leq r \leq 4q + 2$ выбора четного r.*

Прежде всего заметим, что по знаменитой теореме Дирихле прогрессия $n = 2qt+1$ ($t = 0, 1, 2, 3, \ldots$) содержит бесконечно много простых чисел. Нас интересуют простые числа n указанного вида с возможными малыми параметрами $t = 1, 2, \ldots$. Если справедлива обобщенная гипотеза Римана, то наименьшее простое число в указанной последовательности не превосходит $c(\varepsilon)q^{2+\varepsilon}$ при любом $\varepsilon > 0$ ($c(\varepsilon)$ – константа, зависящая от ε). Опыт вычислений показывает, что простые числа в указанной последовательности встречаются довольно часто и близко к ее началу. Заметим также, что по известной в теории чисел гипотезе Крамера $p_{n+1} - p_n = 0(\ln^2 p_n)$ (здесь p_n обозначает n-е по порядку простое число). Примерно то же самое следует из обобщенной гипотезы Римана.

Пример 42. *Следующие последовательности простых чисел построила имеющаяся у меня программа, написанная одним из студентов:*
11, 353, 372769, 181860575417,
8913023801957890501 4857 *(23 знака);*
7, 197, 9613**7**, 19404292081,
733929752109109478339 *(21 знак).*

Так же легко она построила простое число
971094454039605311359192814875471851126 47209784567
(50 знаков).

Работала она на обычном процессоре Pentium 4 в течение нескольких секунд. При этом я не стремился к рекордам.

Задачи и упражнения

Задача 11.1. Математики в Древнем Китае считали, что число n является простым тогда и только тогда, когда n делит $2^n - 2$.

Доказать, что данное условие действительно является необходимым для простоты числа n. Более того, простое число n обязано делить разность $k^n - k$ при любом k.

Привести пример, показывающий его недостаточность. Другими словами, представить составное число n, тем не менее делящее $2^n - 2$.

Задача 11.2. Пусть p – простое число. Доказать, что

$$(p-1)! = -1 (\mathrm{mod} p).$$

Этот результат известен как теорема Вильсона.

Задача 11.3. Пусть n – составное число, отличное от 4. Доказать, что

$$(n-1)! = 0 (\mathrm{mod} n).$$

Задача 11.4. Пусть $a^n - 1$ – простое число. Доказать, что в этом случае $a = 2$ и n – тоже простое число.

Простые числа вида $a^p - 1, p$ простое, называются *числами Мерсенна.*

Задача 11.5. Пусть $a^n + 1$ – простое число. Доказать, что a четно и n является степенью 2.

Простые числа вида $2^{2^t} + 1$ называются *числами Ферма.*

Разложимость целых чисел на множители

1. Представление целого составного числа в виде разности квадратов

Пусть n – положительное целое нечетное число. Пусть $n = ab$ – произведение натуральных чисел, где для определенности $a \geq b \geq 2$. Тогда $n = t^2 - s^2 = (t+s)(t-s) = ab$, где $a = t+s$, $b = t-s$. Мы видим, что любому разложению $n = ab$ соответствует представление в виде разности квадратов $n = t^2 - s^2$ и наоборот. Получить последнее представление часто бывает легче, чем искать разложение в обычном виде. Эту идею высказывал еще Ферма.

Пример 43. *Пусть требуется факторизовать* $n = 4559$. *Легко вычислить, что* $[\sqrt{n}] = 67$ $(67^2 = 4489 < n < 68^2 = 4624)$.

Попытаемся найти представление вида $n = t^2 - s^2$, *беря в качестве* t *числа:* $[\sqrt{n}] + 1 = 68, [\sqrt{n}] + 2 = 69, [\sqrt{n}] + 3 = 70, \ldots$. *Имеем:*

$$68^2 - n = 4624 - 4559 = 65$$

(не подходит, так как не является полным квадратом, поскольку $8^2 = 64 < 65 < 9^2 = 81$*);*

$$69^2 - n = 4761 - 4559 = 202$$

(не подходит, так как не является полным квадратом, поскольку $14^2 = 196 < 202 < 15^2 = 225$*);*

$$70^2 - n = 4900 - 4559 = 341$$

(не подходит, так как не является полным квадратом, поскольку $18^2 = 324 < 341 < 19^2 = 361$*);*

$$71^2 - n = 5041 - 4559 = 482$$

(не подходит, так как не является полным квадратом, поскольку $21^2 = 441 < 482 < 22^2 = 484$*);*

$$72^2 - n = 5184 - 4559 = 625 = 25^2,$$

значит,

$$n = 4559 = 72^2 - 25^2 = 97 \cdot 47.$$

В рассмотренном примере разложение получилось почти сразу. Это произошло из-за того, что число t ненамного превосходит $[\sqrt{n}]$. Можно сказать и по-другому: множители a, b сравнительно мало отличаются друг от друга, а число s мало.

Иногда, a близко не к b, а к kb для некоторого множителя k. В этом случае $kn = (ka)(kb)$ – разложение с близкими множителями при малом k. Тогда ищем представление $t^2 - kn = s^2$, полагая последовательно $t = [\sqrt{kn}] + 1, [\sqrt{kn}] + 2, \ldots$, пока не получим $t^2 - kn$ как полный квадрат.Отсюда можно заключить, что для некоторого s имеем $kn = t^2 - s^2 = (t + s)(t - s)$. Шанс получить разложение n на множители заключается в том, что нод$(n, t + s)$ или нод$(n, t - s)$ будет отличен от n и 1.

Пример 44. *Пусть требуется факторизовать* $n = 223027$*, обладая дополнительной информацией о близости одного из его делителей* a *к числу* $5b$*, где* b *– другой делитель.*

Ищем представление $t^2 - 5n = s^2$*, полагая последовательно* $t = [\sqrt{5n}] + 1, [\sqrt{5n}] + 2, \ldots$*, пока не получим* $t^2 - kn$ *в виде* s^2*.*

Далее:

$t_1 = [\sqrt{5n}] + 1 = 1056,$

$t_1^2 - 5n = 1115136 - 1115135 = 1,$

$5n = 1056^2 - 1^2 = 1057 \cdot 1055.$

Мы видим, что нод$(1057, 5) = 1$*. Значит,* $n \vdots 1057$*. Подставляем значение* n *и получаем разложение* $n = 223027 = 1057 \cdot 211$.

2. Метод фактор-баз Ферма

Этот метод разложения на множители больших нечетных составных чисел n использует следующую неформальную идею. Если мы найдем совершенно разными способами два квадрата t^2, s^2, сравнимых $\bmod n$: $t^2 = s^2 (\bmod n)$, что равносильно выполнению равенства $t^2 = s^2 + nl$ для некоторого целого l, то мы получаем шанс найти разложение n на множители из равенства $t^2 - s^2 = (t+s)(t-s) = nl$, вычисляя $d_1 = \text{нод}(t+s, n)$, $d_2 = \text{нод}(t-s, n)$. Удача будет сопутствовать нам, когда для некоторого $i = 1, 2$ получится $d = d_i \neq 1, n$. Разложение $n = d \cdot b$, где $b = n/d$, будет искомым.

Слова «совершенно разными способами» могут иметь различный смысл, один из которых реализуется в методе фактор-баз, к описанию которого мы сейчас приступаем.

Проведем небольшую подготовку.

Определение 45. *Наименьшим абсолютным вычетом (least absolute residue) числа a по нечетному модулю n называется число b из интервала $I_n = \left[-\frac{n-1}{2}, \frac{n-1}{2}\right]$ с условием: $b = a (\bmod n)$. Так как длина интервала I_n равна n, наименьший абсолютный вычет всегда существует и единственен для любого a. Обозначим его $b = a \bmod n|$.*

В этом определении мы заменяем стандартные имена вычетов $0, 1, \ldots, n-1 \bmod n$ другими стандартными именами: $-\frac{n-1}{2}$, $-\frac{n-1}{2}+1, \ldots, 0, \ldots, \frac{n-1}{2}-1, \frac{n-1}{2}$. Конечно, можно привести аналогичное определение и для четного модуля, убирая одно из крайних значений: $-\frac{n}{2}$ или $\frac{n}{2}$, но для наших целей оно не понадобится.

Определение 46. *Если $B = \{-1, p_1, p_2, \ldots, p_k\}$ – некоторое множество целых чисел, состоящее из -1 и простых чисел $p_1, p_2, \ldots, p_k$, то B-числом называется такое целое число b, что разложение $|b|$ в произведение степеней простых чисел содержит множители из B (другими словами: «b полностью разлагается в произведение множителей из B» или «b не имеет простых делителей, не входящих в B»).*

Определение 47. *Квадрат целого числа b^2 называется B-квадратом относительно нечетного модуля n, если его наименьший абсолютный вычет $b^2 \bmod n$ является B-числом.*

Теперь все готово к описанию метода. Предположим, что нам нужно разложить на множители нечетное составное число n.

Установка

Вначале выбирается фактор-база $B = \{-1, p_1, p_2, \ldots, p_k\}$, состоящая из -1 и некоторых (различных) простых чисел $p_1, p_2, \ldots, p_k$.

Выбранные фактор-база и модуль n позволяют однозначно сопоставить любому B-квадрату b^2 относительно n целочисленный вектор $\chi(b^2) = (\varepsilon, \gamma_1, \gamma_2, \ldots, \gamma_k)$, где $\varepsilon \in \{0, 1\}$, компоненты γ_i однозначно определяются из разложения $b^2 \bmod n = (-1)^{\varepsilon} p_1^{\gamma_1} p_2^{\gamma_2} \ldots p_k^{\gamma_k}$. Вектор $\chi(b^2)$ назовем четным, если все его компоненты четные. Набор B-квадратов $b_1^2, b_2^2, \ldots, b_l^2$ назовем четным, если сумма $\sum_{i=1}^{l} \chi(b_i^2)$ – четный вектор (первые компоненты складываются mod2, остальные – как целые числа). В этом случае произведение $c = \prod_{i=1}^{l} b_i^2 \bmod n$ является полным квадратом, сравнимым с произведением $d = \prod_{i=1}^{l} b_i^2 \bmod n$. Обозначив $d = t^2, c = s^2$ $(t, s \in Z)$, получим сравнение $t^2 = s^2 (\mathrm{mod} n)$, дающее шанс разложить n на множители.

Алгоритм фактор-базы

1) Последовательно выбираем квадраты: $b_1^2, b_2^2, \ldots, b_k^2$, оставляя только те из них, которые являются B-квадратами. Для этого вычисляем наименьшие абсолютные вычеты $b_i^2 \bmod n$, $i = 1, 2, \ldots, k$ и проверяем, являются ли они B-числами. Определяем векторы $\chi(b_1^2), \chi(b_2^2), \ldots, \chi(b_k^2)$.

2) Находим четный поднабор B-квадратов $b_{i_1}^2, b_{i_2}^2, \ldots, b_{i_q}^2$ и соответствующие числа $d = t^2 = \prod_{i=1}^{q} b_{i_j}^2$, $c = s^2 = \prod_{j=1}^{q} b_{i_j}^2 \bmod n$, связанные сравнением $t^2 = s^2 (\mathrm{mod} n)$.

3) Вычисляем $d_1 = \text{нод}(t+s, n)$, $d_2 = \text{нод}(t-s, n)$ и если хотя бы для одного $d = d_i$ ($i = 1$ или 2) $d \neq 1, n$ находим разложение $n = d \cdot b$.

Если в 3) нас постигла неудача, находим другой четный поднабор в 2) и повторяем попытку. Если повторение 2) и 3) не дает результата, ищем другой набор B-квадратов в 1). Если это также не дает результата (после достаточно большого числа попыток), то меняем фактор-базу в установке.

Замечание 48. *Разложение больших нечетных составных чисел на множители – это высокое искусство. В приведенном выше алгоритме мы не оцениваем размеры фактор-базы и множества выбранных B-квадратов. Для уяснения подобных деталей необходимо обращаться к специальной литературе (для начала посмотреть [23]).*

Более того, основные выдающиеся результаты в данном направлении получены с использованием метода квадратичного решета, обобщающего и наводящего определенный порядок в идеях метода фактор-баз. В настоящих лекциях мы не затрагиваем этот метод, достойный гораздо более подробного и тщательного изучения, чем мы можем сейчас себе позволить.

Пример 49. *Пусть требуется факторизовать число* $n = 23299$. *Выберем в качестве фактор-базы множество* $B = \{-1, 2, 3, 5, 7, 11\}$. *Будем брать в качестве квадратов числа* $b_i^2 = ([\sqrt{n}] + i)^2 = (152+i)^2$, $i = 1, 2, \ldots$, *оставляя для дальнейшего рассмотрения B-квадраты:*

$b_1^2 = 153^2 = 23409$, $b_1^2 \bmod n = 110 = 2 \cdot 5 \cdot 11$,
$b_2^2 = 154^2 = 23716$, $b_2^2 \bmod n = 417 = 3 \cdot 139$ *(не подходит)*,
$b_3^2 = 155^2 = 24025$, $b_3^2 \bmod n = 726 = 2 \cdot 3 \cdot 11^2$,
$b_4^2 = 156^2 = 24336$, $b_4^2 \bmod n = 1037$ *(не подходит)*,
$b_5^2 = 157^2 = 24649$, $b_5^2 \bmod n = 1350 = 2 \cdot 3^3 \cdot 5^2$.

Выпишем соответствующие B-квадратам векторы:

$\chi(b_1^2) = (0, 1, 0, 1, 0, 1)$,
$\chi(b_3^2) = (0, 1, 1, 0, 0, 2)$,

$\chi(b_5^2) = (0,1,3,2,0,0)$.

В качестве четного поднабора полученных B-квадратов берем b_3^2, b_5^2. *Вычисляем:*

$d = b_3^2 b_5^2 = (155 \cdot 157)^2 = (24335)^2$,

$c = b_3^2 | \bmod n| \cdot b_5^2 | \bmod n| = (2 \cdot 3^2 \cdot 5 \cdot 11)^2 = (990)^2$.

Далее получаем равенство

$(24335 + 990)(24335 - 990) = 25325 \cdot 23345 = n \cdot l$

для некоторого $l \in \mathbf{Z}$.

Вычисляем

нод$(23345, 23299) = 23$.

Получаем разложение

$n = 23299 = 23 \cdot 1013$.

Пример 50. *Попытаемся факторизовать число* $n = 1829$, *беря в качестве* b_i *целые числа вида* $[\sqrt{nk}]$ *и* $[\sqrt{nk}] + 1$, $k = 1, 2, \ldots$ *такие, что наименьшие абсолютные вычеты* $b_i^2 | \bmod n|$ *являются произведениями (возможно, со знаком «минус») простых чисел, не превышающих* 19. *Это означает, что мы фиксируем в качестве фактор-базы множество* $B = \{-1, 2, 3, 5, 7, 11, 13, 17, 19\}$.

Запишем полученные данные:

$b_1^2 = [\sqrt{n}]^2 = (42)^2 = 1764$, $b_1^2 | \bmod n| = -65 = (-1) \cdot 5 \cdot 13$,

$b_2^2 = ([\sqrt{n}] + 1)^2 = (43)^2 = 1849$, $b_2^2 | \bmod n| = 20 = 2^2 \cdot 5$,

$b_3^2 = ([\sqrt{2n}])^2 = (60)^2 = 3600$, $b_3^2 | \bmod n| = -58 = (-1) \cdot 2 \cdot 29$,

(не подходит),

$b_4^2 = ([\sqrt{2n}] + 1)^2 = (61)^2 = 3721$, $b_4^2 | \bmod n| = 63 = 3^2 \cdot 7$,

$b_5^2 = ([\sqrt{3n}])^2 = (74)^2 = 5476$, $b_5^2 | \bmod n| = -11$,

$b_6^2 = ([\sqrt{3n}] + 1)^2 = (75)^2 = 5625$, $b_6^2 | \bmod n| = 138 = 3 \cdot 2 \cdot 23$,

(не подходит),

$b_7^2 = ([\sqrt{4n}])^2 = (85)^2 = 7225$, $b_7^2 | \bmod n| = -91 = (-1) \cdot 7 \cdot 13$,

$b_8^2 = ([\sqrt{4n}] + 1)^2 = (86)^2 = 7396$, $b_8^2 | \bmod n| = 80 = 2^4 \cdot 5$.

Выпишем соответствующие B-квадратам векторы:

$\chi(b_1^2) = (1,0,0,1,0,0,1,0,0)$,

$\chi(b_2^2) = (0,2,0,1,0,0,0,0,0)$,

$\chi(b_4^2) = (0,0,2,0,1,0,0,0,0)$,

$\chi(b_5^2) = (1,0,0,0,0,1,0,0,0)$,

$\chi(b_7^2) = (1,0,0,0,1,0,1,0,0),$

$\chi(b_8^2) = (0,4,0,1,0,0,0,0,0).$

Мы видим, что b_2^2, b_8^2 *– четный поднабор.*

Вычисляем:

$d = b_2^2 \cdot b_8^2 = (43 \cdot 86)^2 = (3698)^2,$

$c = b_2^2 | \bmod n | \cdot b_8^2 | \bmod n | = (2^3 \cdot 5)^2 = (40)^2,$

значит,

$(3698 + 40)(3698 - 40) = 3738 \cdot 3658 = nl$

для некоторого $l \in \mathbf{Z}$.

Вычисляем:

нод$(3738, 1829) = 1,$

нод$(3658, 1829) = 1829,$

что свидетельствует о неудаче в разложении n.

Заметим, что $b_1^2, b_2^2, b_4^2, b_7^2$ *также является четным поднабором. Вычисляем*

$d = (42 \cdot 43 \cdot 61 \cdot 85)^2 = (9364110)^2,$

$c = (2 \cdot 3 \cdot 5 \cdot 7 \cdot 13)^2 = (2730)^2.$

Можно упростить выражения, заменяя значения в скобках на сравнения с ними $\bmod n$*:*

$d_1 = (1459)^2,\ c_1 = (901)^2.$

Тогда

$(1459 + 901)(1459 - 901) = 2360 \cdot 558.$

Вычисляем:

нод$(2360, 1829) = 59,$

нод$(558, 1829) = 31,$

что дает разложение

$n = 1829 = 59 \cdot 31.$

Задачи и упражнения

Задача 12.1. Разложить на множители числа

$$n_1 = 21894583143407671,$$

$$n_2 = 317481472366756346287.$$

Задача 12.2. Найти простое число $p \geq 2^{100}$, такое, что $q = (p-1)/2$ – тоже простое.

Такие числа называются числами Софи Жермен.

Задача 12.3. Разложить на множители числа

$$n_1 = 57398617605606738780974553655371817744968877084487616807685915353903 24,$$

$$n_2 = 30004823012610422335343801599588688229540363958714487162565428757720061053669162994262636443794703 39,$$

$$n_3 = 16604589036844609947075611165473677273146067100305915193876385419636008124704444102982413426026365453 7.$$

Алгоритмы генерации псевдослучайных последовательностей

1. Равномерно распределенная случайная последовательность (РРСП)

Случайная последовательность вычетов $x_i \in \mathbf{Z}_n, i = 1, ..., t, ...$ называется *равномерно распределенной*, если выполнены следующие свойства.

(C_1) Для любого натурального числа k и произвольных значений индексов $1 \leq t_1 < ... < t_k$ случайные величины $x_{t_1}, ..., x_{t_k} \in \in \mathbf{Z}_n$ независимы в совокупности.

(C_2) Для любого натурального t случайная величина $x_t \in \mathbf{Z}_n$ распределена равномерно:

$$P(x_t = m) = \frac{1}{n}$$

для любого $m \in \mathbf{Z}_n$.

Важным следствием свойства (C_1) с точки зрения теории информации является «непредсказуемость» любого значения x_t при известных значениях $x_{t_1}, ..., x_{t_k}$, если $t \neq t_j, j = 1, ..., k$. Действительно, из совокупной независимости случайных величин $x_{t_1}, ...,$ x_{t_n}, x_t следует, что знание значений первых k величин не изменяет вероятностей значений величины x_t. Кроме того, из свойства (C_1) следует, что любая подпоследовательность РРСП также является РРСП. Можно привести еще ряд следствий свойств (C_1) и (C_2) как вероятностного, так и информационного характера. Смотрите по этому поводу [14]. Там же можно найти описания известных тестов проверки последовательности на свойство быть РРСП. Впрочем, еще лучше обратиться к классической монографии Д. Кнута [6].

2. Генерация псевдослучайных последовательностей

Псевдослучайной называется последовательность $x_i \in \mathbf{Z}_n, i = 1, ..., t, ...$, вычисляемая по известному рекуррентному соотношению и по своим статистическим свойствам сходная с РРСП. Предполагается компьютерное построение последовательностей, и поскольку современные компьютеры используют реализованные в программах детерминированные алгоритмы, мы не можем говорить о построении случайных последовательностей, а только псевдослучайных. Мы рассмотрим ряд известных генераторов псевдослучайных последовательностей.

Линейные и мультипликативные конгруэнтные генераторы

Пусть множеством возможных значений последовательности является $\mathbf{Z}_n$. *Линейный конгруэнтный генератор (ЛКГ)* определяется рекуррентным соотношением

$$x_{t+1} = ax_t + b(mod n),$$

где $x_0 \in \mathbf{Z}_n$ – *начальное значение (seed)*, $0 \neq a \in \mathbf{Z}_n$ – *мультипликатор*, $b \in \mathbf{Z}_n$ – *приращение*. Если $b = 0$, то получаем *мультипликативный конгруэнтный генератор (МКГ)*. Данные генераторы заведомо дают периодические последовательности с периодом, не превышающим n. Последнее очевидно, поскольку при $x_{t_i} = x_{t_j}$ получаем $x_{t_i+l} = x_{t_j+l}, l = 1, 2, ...$

Генератор псевдослучайных последовательностей, основанный на системе RSA

Установка

Прежде всего выбирается система RSA, как это объяснено в Лекции 10. Для работы генератора нам достаточно знать открытые параметры: модуль n и открытый ключ e.

Алгоритм генерации

1) Выбирается случайный вычет $x_0 \in \mathbf{Z}_n$.

2) Для $i = 1, 2, \ldots$ последовательно проводятся вычисления по формуле

$$x_i = x_{i-1}^e (mod n).$$

3) Вычисляются младшие биты z_i вычетов $x_i \in \mathbf{Z}_n, i = 1, 2, \ldots$.

Результат

Бинарная последовательность

$$z_1, z_2, \ldots.$$

Замечание 51. *1) Для вычетов x_i, $i = 0, 1, \ldots$ используются их стандартные имена $0, 1, \ldots, n-1$.*

2) Как правило, заранее определяется число l элементов выпускной последовательности, которая в этом случае выглядит как

$$z_1, z_2, \ldots, z_l.$$

Секретность данного процесса имеет то же самое основание, что и система RSA: трудность разложения на множители больших чисел.

Генератор псевдослучайных последовательностей Микали – Шнорра (Micali – Schnorr)

Установка

Выбирается система RSA, причем для работы генератора, как и в описанном выше случае, требуется лишь знание открытых параметров: модуля n и открытого ключа e. При этом, однако, дополнительно требуется выполнение неравенства $80e \leq N$, где

$N = \lfloor log_2 n \rfloor + 1$ (длина n в битах). Полагается $k = \lfloor N(1 - 2/e) \rfloor$ и $r = N - k$.

Алгоритм генерации

1) Выбирается случайная последовательность $x_0 \in \{0,1\}^*$ битовой длины r (seed).

2) Псевдослучайная последовательность длины kl порождается в соответствии со следующими формулами: $(i = 1, 2, ..., l)$:

2.1) $y_i = x_{i-1}^e (mod n)$;
2.2) x_i – последовательность r старших битов y_i;
2.3)z_i – последовательность k младших битов y_i.

3) Выпускная последовательность имеет вид:

$$z_1 || z_2 || ... || z_l,$$

где || обозначает операцию дописывания (конкатенации).

Замечание 52. *1) Алгоритм Микали – Шнорра более экономичен, чем алгоритм, описанный перед ним, поскольку в нем после одного экспоненцирования выходит больше битов, чем в том алгоритме. Например, при* $e = 3$ *и* $N = 1024$ *получаем* $k = 341$, *то есть одно экспоненцирование дает* 341*-битовую часть выпускной последовательности. Более того, каждое экспоненцирование требует только одного модулярного возведения в квадрат* 683*-битового числа и одного модулярного умножения.*

2) Секретность алгоритма Микали – Шнорра основывается на более сильном, чем трудность разложения больших чисел на множители, допущении следующего вида: распределение $x^e (mod n)$ *для случайной* r*-битовой последовательности* x *не различимо полиномиальными по времени статистическими тестами от равномерного распределения целых в интервале* $[0, n-1]$.

Генератор BBS псевдослучайных последовательностей

Оператор введен в рассмотрение Мануэлем Блюмом, Ленорой Блюм и Майклом Шубом (M. Blum, L. Blum, M. Shub) в 1983 году.

Установка

Выбирают большие случайные (различные) простые числа p, q, сравнимые с $3(mod4)$, вычисляется модуль $n = pq$.

Алгоритм генерации

1) Выбирается случайное число $1 \leq s \leq n-1$, взаимно простое с n (seed). Вычисляется

$$x_0 = s^2(modn).$$

2) Псевдослучайная последовательность длины l вычисляется в соответствии со следующими формулами $(i = 1, ..., l)$:

2.1. $x_i = x_{i-1}^2(modn)$;
2.2. z_i – наименьший бит x_i.

3) Выпускная последовательность имеет вид:

$$z_1, ..., z_l.$$

Замечание 53. *Секретность алгоритма базируется на трудности задачи извлечения квадратного корня $modn$, где $n = pq$ – произведение больших различных простых чисел, что равносильно задаче разложимости числа n на множители (см. лекцию 10).*

Задачи и упражнения

Задача 13.1. Объяснить, почему при установке BBS-генератора требуется, чтобы простые числа p, q были сравнимы с $3(mod 4)$.

Задача 13.2. Вывести формулу для общего члена последовательности, порождаемой ЛКГ

$$x_{t+1} = ax_t + b(\mathrm{mod} n).$$

Задача 13.3. Выяснить, когда ЛКГ из предыдущей задачи порождает последовательность максимального периода.

Задача 13.4. Пусть задан МКГ

$$x_{t+1} = ax_t(\mathrm{mod} n).$$

Предположим, что нод$(x_0, n) = 1$. Каково возможное максимальное значение периода выпускной последовательности?

Поточные криптосистемы

1. Общие понятия

Поточные криптосистемы зашифровывают исходный текст, работая последовательно с его буквами, как правило, из бинарного алфавита $\{0, 1\}$. В отличие от блочных криптосистем, использующих постоянные ключи для больших блоков букв, поточные криптосистемы пользуются меняющимися ключами. Как правило, поточные криптосистемы более быстрые при работе на компьютерах, они используются, например, в тех случаях, когда помещение текущей информации в буфер ограничено, когда вероятны ошибки при передаче информации и т. п.

Существуют обширные знания о поточных криптосистемах, они широко используются в практике, но очень незначительно (в отличие от блочных криптосистем) отражаются в открытой литературе. Можно, тем не менее, ожидать, что в ближайшие годы поточные криптосистемы будут интенсивно развиваться.

Поточные криптосистемы могут использовать как симметрические, так и открытые ключи. В данной лекции мы ограничимся рассмотрением только одного из возможных методов генерации ключей.

Напомним, что криптосистема Вернама использует в качестве платформ бинарные последовательности. Она выглядит следующим образом: $c_i = m_i \oplus k_i, i = 1, 2, \ldots$, где m_i – единица исходного текста, k_i – единица поточного ключа (keystream), представляющие из себя также буквы бинарного алфавита, c_i – единица шифрованного текста, получающаяся сложением mod2 единицы исходного текста и ключа.

Замечание 54. *Очевидно, что поточный ключ $k_1, k_2, \ldots$ в криптосистеме Вернама должен быть не короче, чем исходный*

текст. Секретность ключей криптосистемы абсолютная в том смысле, что знание только шифрованного текста не позволяет восстановить исходный текст, более того, частичная расшифровка текста не добавляет сведений о том, как можно расшифровать остальной текст.

Недостатки системы также очевидны – громозкий поточный ключ, трудность в управлении и передаче ключей.

2. Синхронные поточные криптосистемы

Так называются системы, в которых ключ $k = k_1, k_2, \ldots$ генерируется независимо от исходного и шифрованного текстов. Процесс шифровки может быть описан следующими преобразованиями:

$$\sigma_{i+1} = f(\sigma_i, k),$$
$$k_i = g(\sigma_i, k),$$
$$c_i = h(k_i, m_i),$$

где σ_0 – начальное состояние системы, определяемое только по ключу k, f – функция, определяющая последующие состояния, g – функция, генерирующая поточный ключ $k_1, k_2, \ldots$, h – функция шифрования, комбинирующая единицы исходного текста m_i и единицы поточного ключа k_i, выдающая в качестве значений единицы шифрованного текста c_i, $i = 1, 2, \ldots$.

Замечание 55. *1) Необходимость синхронизации.*

В поточной синхронной криптосистеме оба участника должны действовать синхронно, использовать единый ключ и оперировать с одной и той же позицией. Если теряются или, наоборот, добавляются единицы шифрованного текста, то процесс снова необходимо синхронизировать.

2) Нераспространение ошибок.

Если какие-то единицы шифрованного текста изменяются (но не исчезают или добавляются), то эта ошибка не влияет на расшифровку остального текста.

3) Обнаружение активных атак.

Из 1) следует, что добавление или выбрасывание единиц шифрованного текста немедленно приводит к десинхронизации и может быть замечено принимающим.

3. Самосинхронизирующиеся поточные криптосистемы

Такие системы определяются следующими преобразованиями:

$$\sigma_i = (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}),$$

$$k_i = g(\sigma_i, k),$$

$$c_i = h(k_i, m_i),$$

где, $\sigma_0 = (c_{-t}, c_{-t+1}, \ldots, c_{-1})$ – начальное (открытое) состояние, k – ключ, g – функция, производящая текущий ключ, h – функция шифрования.

Как и раньше, k_i – единицы поточного ключа, m_i – единицы исходного текста, c_i – единицы шифрованного текста.

Замечание 56. *1) Самосинхронизация.*

Так как дешифровка зависит только от ограниченного числа единиц шифрованного текста, то если какие-то из них исчезли или добавились, через некоторое время это уже не будет играть роли.

2) Ограниченное распространение ошибок.

Ошибки распространяются, но только на ограниченное число шагов.

3) Трудность обнаружения ошибок.

Несколько труднее обнаружить ошибки, чем в предыдущей системе (в синхронных поточных криптосистемах), поскольу они проявляются не сразу.

4. Линейный регистр сдвига с обратной связью (Linear feedback shift register – LFSR)

LFSR часто используются при генерации поточных ключей. Для этого есть следующие причины:

1) LFSR хорошо встраивается в «железо»,

2) он может производить последовательности большого периода с хорошими статистическими свойствами,

3) он алгебраичен, хорошо поддается изучению.

Описание линейного регистра с обратной связью

Определим LFSR длины L состояний (или элементов задержки): $0, 1, \ldots, L-1$, каждое из которых может принимать (input), хранить (store) один бит информации, а также часов (clock), контролирующих прохождение данных. В течение единицы времени происходят следующие события:

4.1) содержание s_0 состояния 0 выпускается и становится частью выпускной последовательности;

4.2) содержание s_i состояния i переходит в содержание состояния $i-1$ для всех $1 \leq i \leq L-1$;

4.3) появляется новое содержание состояния $L-1$ (feedback bit) s_{L-1}, являющееся суммой mod2 содержаний некоторого фиксированного подмножества состояний $\{0, 1, \ldots, L-1\}$.

Например, пусть $c_i, i = 0, 1, \ldots, L-1$ – метки соответствующих состояний. Определим s_j как сумму mod2 содержаний i-х состояний, для которых $c_{L-i} = 1$.

Определим *связующий многочлен* $c(D) = 1 + c_1 D + \ldots + c_L D^L \in$ $\in Z_2[D]$, который называется несингулярным, если $c_L = 1$, сингулярным, если $c_L = 0$.

Пусть $[s_{L-1}, s_{L-2}, \ldots, s_1, s_0]$ – вектор начальных содержаний, соответствующих состояниям LFSR.

Предложение 57. *Если вектор начальных содержаний зафиксирован, то выпускная последовательность* $s = s_0, s_1, \ldots, s_{L-1}, s_L, \ldots$ *при* $j \geq L$ *единственным образом определяется формулой:*

$$s_j = c_1 s_{j-1} + c_2 s_{j-2} + \ldots + c_L s_{j-L} \pmod 2.$$

Пример 58. *Пусть* $L = 4$, $[s_3, s_2, s_1, s_0] = [0, 1, 0, 1]$, $c(D) = 1 + D + D^3 + D^4$, *то есть* $c_1 = 1, c_2 = 0, c_3 = 1, c_4 = 1$.

Тогда $s_4 = s_3 + s_1 + s_0 = 0 + 0 + 1 = 1$, $s_5 = s_4 + s_2 + s_1 = 1 + 1 + 0 + 0$, *следовательно, выпускная последовательность имеет вид*

$$1, 0, 1, 0, 1, 0, 1, 0, \ldots .$$

Период этой последовательности равен 2.

Пример 59. *Пусть* $L = 4$, $[s_3, s_2, s_1, s_0] = [0, 1, 1, 0]$, $c(D) = 1 + D + D^4$, *то есть* $c_1 = 1, c_2 = 0, c_3 = 0, c_4 = 1$.

Тогда $s_4 = s_3 + s_0 = 0 + 0 = 0$, $s_5 = s_4 + s_1 = 0 + 1 + 1$, *и далее:* $s_6 = 0, s_7 = 0, s_8 = 0 \ldots$.

Можно проверить, что данная последовательность имеет максимально возможный период 15, *поскольку первое повторение содержаний регистра происходит, когда появляется строка* $[s_{18}, s_{17}, s_{16}, s_{15}] = [0, 1, 1, 0]$.

Факты о периодичности и статистике выпускных последовательностей LFSR

Неприводимый многочлен $f(D) \in \mathbf{F}_{p^r}[D]$ называется *примитивным,* если элемент D порождает группу $\mathbf{F}^*_{p^r}$.

1. Если $c(D) \in \mathbf{Z}_2[D]$ неприводим, то каждый из $2^L - 1$ ненулевых векторов начальных содержаний $[s_{L-1}, s_{L-2}, \ldots, s_1, s_0]$ несингулярного LFSR дает выпускную последовательность с периодом, равным $\min N$, для которого $c(D)$ делит $1 + D^N$.

2. Если $c(D)$ к тому же примитивен, то каждый из $2^L - 1$ ненулевых векторов $[s_{L-1}, s_{L-2}, \ldots, s_1, s_0]$ дает выпускную последовательность максимально возможного периода $2^L - 1$. Вот почему $LFSR =< L, c(D) | c(D)$ – примитивен $>$ называется еще LFSR максимальной длины.

3. Пусть выпускная последовательность S получена с использованием примитивного многочлена, то есть она имеет максимально возможный период. Пусть $1 \leq k \leq L-1$, $\bar{S}$ – подпоследовательность (без пробелов) в S длины $2^L + k - 2$. Тогда любая ненулевая последовательность длины k появляется в точности 2^{L-k} раз как подпоследовательность (без пробелов) в $\bar{S}$. Нулевая появляется $2^{L-k} - 1$ раз. Другими словами, распределение в последовательности S почти равномерное.

Объяснение

Необходимо объяснить, почему такое число N существует. Мы докажем, что в любом случае многочлен $c(D)$ делит в кольце $\mathbf{Z}_2[D]$ многочлен $D^{2^L-1} + 1$ (значит, N существует, причем $N \leq 2^L - 1$). Так как $c(D)$ неприводимый степени L, он определяет поле $\mathbf{F}_{2^L}$ (см. лекцию 8). Порядок мультипликативной группы $\mathbf{F}^*_{2^L}$ равен $2^L - 1$. По теореме Лагранжа многочлен $D \in \mathbf{F}^*_{2^L}$ удовлетворяет соотношению $D^{2^L-1} = 1$, что на языке многочленов равносильно равенству вида

$$D^{2^L-1} = 1 + c(D)f(D) \tag{4.1}$$

для некоторого многочлена $f(D) \in \mathbf{Z}_2[D]$. Последнее означает, что $c(D)$ делит многочлен $D^{2^L-1}+1$. Заметим, что число N в точности равно порядку $|D|$ элемента D в мультипликативной группе $\mathbf{F}^*_{2^L}$.

По определению примитивность D равносильна тому, что порядок D в группе $\mathbf{F}^*_{2^L}$ в точности равен $2^L - 1$. Как замечено выше, это дает максимально возможный период $2^L - 1$ выпускной последовательности.

Задачи и упражнения

Задача 14.1. Показать, что многочлен $f(D) = D^4 + D^3 + D^2 + D + 1 \in \mathbf{Z}_2[D]$ неприводим, но не примитивен. Чему равен период выпускной последовательности LFSR с таким связующим многочленом $f(D)$?

Задача 14.2. Проверить, будет ли многочлен $f(D) = D^4 + D + 1 \in \mathbf{Z}_2[D]$ неприводимым и примитивным.

Задача 14.3. Показать, что многочлен $f(D) = D^4 + D^3 + 1 \in \mathbf{Z}_2[D]$ неприводим. Будет ли он примитивным? Чему равен период выпускной последовательности LFSR с таким связующим многочленом?

Задача 14.4. Проверить, что неприводимый многочлен $f(D) = D^5 + D^2 + 1 \in \mathbf{Z}_2[D]$ является примитивным. Убедиться, что в качестве связующего многочлена он дает выпускную последовательность максимального периода 31, выбрав вектором начальных содержаний $[1, 1, 1, 1, 1]$.

Задача 14.5. Построить LFSR максимального периода с 6-ю состояниями.

Задача 14.6. Следующая шифровка получена шифром Вернама, в котором ключ генерируется LFSR с длиной регистра 7:

$$0, 1, \mathbf{1}, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, \mathbf{0}, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1,$$

$$0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1.$$

Предположим, что известно начало исходного текста

$$1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0.$$

Найти весь исходный текст. Что можно было бы сказать при длине регистра 8?

Идентификация и аутентификация

Усы, лапы и хвост – вот мои документы!
Кот Матроскин.
Э. Успенский. Трое из Простоквашино

1. Основные понятия

Аутентификацией называется процедура, с помощью которой проверяющий устанавливает законность идентификации пользователя в информационной системе.

Результатом аутентификации является принятие или отклонение поступившего от пользователя требования.

Итак, пользователь должен доказать свою правомочность, а проверяющий должен проверить правильность доказательства.

Требования к протоколу аутентификации:

- пользователь должен иметь возможность доказать законность своей идентификации, проверяющий должен иметь возможность проверить это;
- вероятность отклонения правильного доказательства и вероятность принятия неправильного доказательства должны быть пренебрежимо малы;
- процедура аутентификации должна быть устойчива к подбору, подлогу и подделке.

Протокол аутентификации должен быть протоколом реального времени, отвечающим на вопросы: кто, где и когда?

Идентификация базируется на одном из следующих признаков или их комбинации:

1) знание чего-либо (пароля, PIN, секретного ключа, доказательства некоторого результата и т. п.);

2) возможность предъявления чего-либо (магнитной или смарт-карты, паспорта, иного документа);

3) возможность проверки индивидуальных особенностей (отпечатков пальцев, подписи, голоса, структуры сетчатки глаза, ДНК и т. п.).

Конечно, имеется связь процедуры аутентификации с цифровой подписью. Главное отличие в том, что цифровая подпись, как правило, не связана с реальным временем.

Различают сильную и слабую аутентификацию. Пароли, PIN – это слабая аутентификация. Перейдем к их рассмотрению.

Фиксированные пароли

Обычно пароль – это слово из 6, 10 или более букв, знаков. Пароль является разделенным секретом между пользователем и системой. Основная задача – как хранить пароль.

Возможны следующие способы.

1. Пароли хранятся в специальном файле, защищенном от чтения и записи.

2. Пароли хранятся как образы для некоторой односторонней функции. Файл должен быть защищен от записи.

Существуют определенные правила: ограничение пароля на длину, требование использовать знаки разной природы, ограничение по числу или времени использования паролей.

Применение в UNIX

Файл паролей содержит одностороннюю функцию, хранит образ при этой функции известного текста, сам пароль используется как ключ для этой функции. В качестве такой функции выступает слегка модифицированный стандарт шифрования DES (см. приложение 1).

При вводе пользователем пароля система генерирует на его основе ключ шифрования и шифрует известный текст, а затем

сравнивает шифровку с первоначальной шифровкой, полученной в момент образования пароля.

PINs и passkeys

PIN или personal identification number, используется как фиксированный пароль, обычно совместно с каким-либо физическим носителем – пластиковой или телефонной картой. Введение (сообщение) правильного PINa требуется при использовании карты. PIN обычно короткий. Иногда применяются дополнительные меры, например, конфискация карты, если PIN трижды набран неправильно.

Passkey называется технология, при которой пароль пользователя перерабатывается односторонней функцией в ключ шифрования.

Одноразовые пароли

Имеется общий список допустимых пар паролей. Пользователь выбирает одну из пар и сообщает системе. Система понимает, что правильный пароль – это второй элемент пары. Каждая пара используется только один раз.

2. Многократная идентификация

Одной из наиболее известных схем многократной идентификации является

Схема Лампора (Lamport)

Установка

Пусть имеется пользователь A системы C. Пусть имеется односторонняя функция H.

Протокол

1. Пользователь A выбирает число t возможных идентификаций и заказывает у системы случайное число w.

2. Начальный разделенный секрет $H^t(w) = w_0$ передается системе C. Система C включает счетчик, полагая $i(A) = 1$.

3. При i-й идентификации A посылает C сообщение $(A, i, w_i = H^{t-i}(w))$.

4. C проверяет, что $i = i(A)$ и что $w_{i-1} = H(w_i)$. Если оба равенства верные, C принимает пароль, изменяя значения счетчика $i(A)$, добавляя к нему 1 и сохраняя w_i вместо w_{i-1}.

Ясно, что пользователь A сможет осуществить только t идентификаций.

Объяснение

После того как произойдет t идентификаций, система будет хранить значение $H^0(w) = w$. Счетчик будет показывать $i(A) = t$. Для последующей идентификации пользователю A необходимо передать системе C такое число v, для которого выполняется равенство $H(v) = w$. Поскольку функция H односторонняя, при правильной работе A не может найти такое v. Здесь же скрыто объяснение того, кто должен выбирать начальное число w. Это должна делать система C, ведь A может взять произвольное v и положить $w = H(v)$, а впоследствии обмануть систему на одну лишнюю идентификацию. При желании он может с самого начала зарезервировать за собой любое число l возможных обманов, положив $w = H^l(v)$.

Конечно, числа в протоколе могут быть любой природы. Важно только наличие односторонней функции H.

Заметим также, что счетчик идентификаций играет вспомогательную роль. По сути, ему не надо ничего считать, он только должен указывать на пользователя A, чтобы система C могла правильно вычислять и сверять данные.

3. Доказательства с нулевым разглашением

Свойство нулевого разглашения означает, что пользователь доказывает знание чего-либо таким образом, что по доказательству невозможно установить само знание.

Одним из наиболее известных доказательств такого сорта является следующий протокол.

Протокол Фиата – Шамира (Fiat – Shamir)

Установка

Центр T, которому доверяют все пользователи, выбирает открытый модуль $n = pq$, равный произведению различных больших секретных простых чисел p, q. Произвольный пользователь A выбирает секретное число $1 \leq s \leq n-1$, взаимно простое с n: нод$(s, n) = 1$. Далее A вычисляет $v = s^2 (\mathrm{mod} n)$ и регистрирует v в центре T в качестве открытого данного.

Идентификация

Проводится следующий трехшаговый интерактивный процесс, в результате которого пользователь A идентифицирует себя для пользователя B, доказывая знание s. Далее приводятся передаваемые данные:

1. A выбирает случайное число r, вычисляет его квадрат

$$x = r^2 (\mathrm{mod} n)$$

и передает x для B.

2. B случайным образом (с вероятностью $1/2$) выбирает число $e \in \{0, 1\}$ и передает его A.

3. A вычисляет

$$y = r \cdot s^e (\mathrm{mod} n)$$

и передает его B.

Проверка правильности идентификации

Пользователь B в случае, если $y \neq 0$, проверяет равенство

$$y^2 = x v^e (\mathrm{mod} n).$$

Если оно верно, идентификация считается правильной. В случае $y = 0$ доказательство отвергается. Так как s взаимно просто с n, равенство $y = 0(\mathrm{mod} n)$ возможно только в случае, если $r = 0(\mathrm{mod} n)$. Заметим, что, как обычно, вычеты $\mathrm{mod} n$ фигурируют под своими стандартными именами.

Объяснение

Если $e = 0$, то $y = r(\mathrm{mod} n)$. В этом случае проверка дает равенство

$$y^2 = r^2 = x = x \cdot v^0(\mathrm{mod} n).$$

Если $e = 1$, то $y = rs(\mathrm{mod} n)$, тогда

$$y^2 = r^2 s^2 = x \cdot v(\mathrm{mod} n),$$

что приводит к правильному выводу.

Криптостойкость протокола основывается на трудности вычисления квадратных корней из произвольных элементов $\mathrm{mod} n$, где $n = pq$ – произведение различных больших секретных чисел (см. по этому поводу лекцию 10).

Нулевое разглашение

Если $e = 0$, то B получает $y = r(\mathrm{mod} n)$, что никак не зависит от секрета s. В случае $e = 1$ пользователь B получает $y = rs(\mathrm{mod} n)$, что в силу произвольности r не позволяет ему вычислить s. Как было сказано выше, вычисление s по его квадрату $s^2 = v(\mathrm{mod} n)$ является трудной задачей.

Возможный обман

Допустим, что A вовсе не A, а некоторый злоумышленник C, пытающийся выдать себя за A. При этом ему, конечно, секрет s неизвестен. Может ли он обмануть B? Может, если угадает случайный ответ e.

Если он угадает, что $e = 0$, то ему не нужно знать s, он посылет $y = r(\mathrm{mod} n)$, где r может быть выбрано им самим при посылке

$x = r^2 (\mathrm{mod} n)$. Если он не угадает, то ему нужно будет послать $y = rs (\mathrm{mod} n)$, что невозможно.

Но как он может обмануть, угадав ответ $e = 1$? Очень просто. Он берет произвольное r, но посылает в качестве x не $r^2 (\mathrm{mod}\ n)$, а $x_1 = r^2 \cdot v^{-1} (\mathrm{mod} n)$. Получив ответ $e = 1$, он в качестве y посылает $y_1 = r (\mathrm{mod} n)$. Пользователь b вычисляет $(y_1)^2 = r^2 = = x_1 v (\mathrm{mod} n)$ и видит верное равенство. Однако если угадать не удается, то C получает ответ $e = 0$, и тогда ему нужно послать такое значение r_1, что $(r_1)^2 = x_1 = r^2 s^{-2} (\mathrm{mod} n)$, то есть ему нужно знать $rs^{-1} (\mathrm{mod} n)$, а значит и $s (\mathrm{mod} n)$.

Мы видим, что обмануть можно всегда с вероятностью $1/2$. Обычно этот протокол проводится в t раундов, где t зависит от уровня требований надежности. Таким образом, вероятность обмана $\frac{1}{2^t}$ можно сделать сколь угодно малой.

Задачи и упражнения

Задача 15.1. В сети каждый пользователь A имеет свой открытый алгоритм шифрования E_A и секретный алгоритм дешифрования D_A. Сообщение m от A к B посылается в формате $(E_B(m), A)$. Адрес A говорит B, от кого пришло сообщение. Получатель B извлекает из сообщения m и автоматически посылает обратно по указанному адресу A сообщение $(E_A(m), B)$ в том же формате.

Покажите, что третий пользователь C, который может перехватывать сообщения и посылать их в правильном формате, способен извлечь сообщение m.

Задача 15.2. Показать, что изменение формата на $E_B(E_B(m), A), A)$ и автоматического ответа $E_A(E_A(m), B), B)$ не делает коммуникацию безопасной.

Электронные подписи

Вы не Достоевский, – сказала гражданка,
сбиваемая с толку Коровьевым.
– Ну, почем знать, почем знать, – ответил тот.
– Достоевский умер, – сказала гражданка,
но как-то не очень уверенно.
– Протестую! – горячо воскликнул Бегемот. –
Достоевский бессмертен.
М. Булгаков. Мастер и Маргарита

1. Основные требования

Электронные (цифровые) подписи удостоверяют подлинность электронных документов. Необходимо, чтобы они удовлетворяли всем требованиям, предъявляемым к обычным подписям на бумажных носителях. В то же время особенности выполнения электронных подписей – запись их в файлы – выдвигают к ним новые требования. Главным отличием электронной подписи от обычной является ее зависимость от подписываемого документа.

Перечислим основные требования, которым должна удовлетворять электронная подпись:

- *проверяемость* – любой пользователь информационной системы должен иметь возможность проверить правильность подписи;

- *неотклоняемость* – должна быть предусмотрена возможность доказательства (в том числе юридического) того, что подпись поставлена именно лицом, выпустившим соответствующий документ;

- *уникальность* – подпись должна быть присуща одному лицу, подделка должна быть трудной задачей;

- *неотъемлемая часть документа* – это означает невозможность перенесения подписи на другой документ;

- *защита целостности документа* – его содержание не может быть изменено, заменено и т. п.

Иногда требование уникальности формулируют как «невозможность подделки». Мы заменяем его на «подделка должна быть трудной задачей», где понятие «трудная задача» понимается как обычно в криптографии. «Невозможность» на 100 % обеспечить нельзя. Последние 2 требования свидетельствуют о неразделимости документа и подписи. Отметим в связи с этим, что существуют подписи *добавленные (added)*, для проверки правильности которых необходимо знание подписанного документа, и *восстанавливающие (recover)*, где из подписи воспроизводится сам документ.

2. Основные обозначения

В системе, предусматривающей электронную подпись, должны быть выделены следующие элементы:

M – пространство документов;

M_s – пространство дайджестов документов;

S – пространство подписей;

$R : M \longrightarrow M_s$ – инъективная функция, сопоставляющая документу его дайджест;

$R^{-1} : ImR \longrightarrow M$ – обратная к R функция;

$h : M \longrightarrow M_s$ – односторонняя функция (обычно это хэш-функция).

Часто подпись ставится не под самим документом $m \in M$, а под его дайджестом $R(m) \in M_s$.

3. Подпись на базе RSA

Установка

Каждый пользователь A некоторой информационной системы устанавливает систему шифрования RSA, выбирая открытые данные (n_A, e_A), где n_A – модуль системы, e_A – открытый ключ шифрования, и вычисляет секретный ключ дешифрования d_A, известный только пользователю A: $e_A d_A = 1(\mathrm{mod}\varphi(n_A))$.

Для пользователя A определяются соответствующие пространства $M = M_s = S = \mathbf{Z}_n$ и некоторая функция $R : \mathbf{Z}_n \longrightarrow \mathbf{Z}_n$ (часто в качестве R используется сдвиг $\mathrm{mod}n$).

Алгоритм подписи

1) Имеется документ $m \in \mathbf{Z}_n$;
2) Вычисляется дайджест

$$\tilde{m} = R(m) \in \mathbf{Z}_n;$$

3) Вычисляется подпись

$$s = \tilde{m}^{d_A}(\mathrm{mod}n_A).$$

Алгоритм проверки подписи

1) Берутся открытые данные (n_A, e_A);
2) Вычисляется

$$s^{e_A} = \tilde{m}^{d_A e_A} = \tilde{m}(\mathrm{mod}n_A);$$

3) Восстанавливается

$$m = R^{-1}(\tilde{m}).$$

Объяснение

Поясним равенство 2) из алгоритма проверки подписи (все то же самое, что и в RSA). Так как $e_A d_A = 1(\mathrm{mod}\varphi(n_A))$, для

некоторого $k \in \mathbf{Z}$ имеем $e_A d_A = 1 + k\varphi(n_A)$. Тогда $\tilde{m}^{d_A e_A} = \tilde{m}^{1+k\varphi(n_A)} = \tilde{m}(\tilde{m}^{\varphi(n_A)})^k = \tilde{m}(\mathrm{mod} n_A)$. Последнее равенство подробно объяснено в лекции 10.

Мы видим, что данная подпись является восстанавливающей. В ее правильности мы убеждаемся при правильном построении документа m, из содержания которого должно быть видно, что подпись s – правильная.

Обращаем внимание на выполнение всех требований к электронной подписи. Об этом можно говорить только при правильной установке системы RSA (см. лекцию 10). Поставить подпись в этом случае может только A, никто другой не знает и не может вычислить секретный ключ d_A. Заметим, что процедура постановки подписи противоположна процедуре шифрования. Действительно, зашифровать текст для A может каждый, а поставить свою подпись может только A. Зато проверить ее может каждый, что, собственно, и требуется.

Пример 60. *Пусть* $p = 11$, $q = 7$, *тогда* $n = 77$, $\varphi(n) = 60$. *Пусть* $e = 13$, *тогда* $d = 37$. *Предположим, что* $m = 2$, $\tilde{m} = R(m) = m + 1 = 3$. *Подписью для* $\tilde{m}$ *будет* $\tilde{m}^d = 3^{37} = 31(\mathrm{mod}\ 77)$. *Проверяем:* $\tilde{m} = 31^{13} = 3(\mathrm{mod} 77), m = 3 - 1 = 2$.

4. Возможные атаки на подпись в системе RSA

В лекции 10 рассмотрены вопросы криптостойкости RSA. Ясно, что они имеют прямое отношение к надежности соответствующей подписи. Мы не будем повторять уже сказанное. Отметим только некоторые особенности, присущие подписи в системе RSA и не связанные напрямую со стойкостью самой системы.

Использование мультипликативности

Пусть мы отказались от использования функции R (приняли ее равной тождественной функции). Тогда из двух подписей $s_1 = m_1^{d_A}(\mathrm{mod} n_A)$, $s_2 = m_2^{d_A}(\mathrm{mod} n_A)$ можно сформировать новую правильную подпись $s_1 s_2 = m_1^{d_A} m_2^{d_A} = (m_1 m_2)^{d_A}(\mathrm{mod} n_A)$.

Если взять функцию R такую, что $R(m_1 m_2) \neq R(m_1)R(m_2)$ (например, нетривиальный сдвиг), то подобная подделка оказывается невозможной. Цель функции R – нарушить свойство мультипликативности подписи.

Подпись на зашифрованном документе

Допустим, что пользователь A желает не только снабдить документ m подписью s, но и зашифровать сообщение с использованием открытых данных (n_B, e_B) другого пользователя B. Имеются две возможности – сначала зашифровать, затем подписать или сделать все то же самое, но в другом порядке.

Подпись на зашифрованном документе $s = (R(m^{e_B}) (\mathrm{mod} n_B)))^{d_A} (\mathrm{mod} n_A)$ имеет очевидную слабость, которую обычно стараются не допускать. Третье лицо, обозначим его C, может вначале «снять» подпись абонента A, использовав открытый ключ e_A:

$$s^{e_A} = (R(m^{e_B})(\mathrm{mod} n_B))^{d_A e_A} (\mathrm{mod} n_A) = R(m^{e_B})(\mathrm{mod} n_B)(\mathrm{mod} n_A),$$

а затем заменить ее своей подписью

$$s_1 = (R(m^{e_B})(\mathrm{mod} n_B))^{d_C} (\mathrm{mod} n_C).$$

Таким образом можно поставить свою подпись на чужой документ.

Недостатки второй схемы, когда документ вначале подписывается, а затем шифруется, не столь очевидны. Их лучше всего показать на примере.

Пример 61. *Пусть* $n_A = p_A q_A = 8387 \cdot 7499 = 62894113$, $e_A = 5$, $d_A = 37726937$, $n_B = p_B q_B = 55465219$, $e_B = 5$, $d_B = 44360237$.

Пусть R *– тождественное отображение,* $m = 1368797$. *Тогда*

1) $s = m^{d_A} = 59847900 (\mathrm{mod} n_A)$,

2) $c = s^{e_B} = 38842235 (\mathrm{mod} n_B)$.

Пользователь B *проверяет правильность подписи:*

1) $\hat{s} = c^{d_B} = 4382681 (\mathrm{mod} n_B)$,

2) $\hat{m} = \hat{s}^{e_A} = 543835568 (\mathrm{mod} n_A)$.

Обнаруживается, что $\hat{m} \neq m$, хотя все делалось по протоколу. Причиной этого является то, что $n_A > s > n_B$. Мы знаем, что шифровать можно только те сообщения, которые не превосходят соответствующего модуля.

Самый простой способ устранения возможной ошибки указанного типа заключается в том, что пользователи применяют различные модули (следовательно, и системы RSA) для шифрования и подписи. При этом любой модуль для подписи должен быть меньше любого модуля для шифрования.

5. Схема базовой электронной подписи Эль Гамаля

Установка

Выбирается большое простое число p. В качестве платформы используется простое конечное поле $\mathbf{Z}_p$.

Кроме этого, необходима односторонняя функция $h\colon \{0,1\}^* \longrightarrow \mathbf{Z}_p$, сопоставляющая бинарным последовательностям элементы выбранного поля. Часто в роли h выступает хэш-функция.

Также необходимо знать порождающий элемент g мультипликативной группы поля $\mathbf{Z}_p^*$.

Случайно выбранное число $1 \leq a \leq p-2$ играет роль долгосрочного ключа.

По указанным данным вычисляется открытое значение $y = g^a$.

Данные (p, g, y) открыты.

Алгоритм подписи

Пусть m – документ, представляющий собой бинарную последовательность произвольной длины.

1) Случайным образом выбирается число $1 \leq k \leq p-2$, для которого $\text{нод}(k, p-1) = 1$, играющее роль сессионного ключа;

2) Условие взаимной простоты k и $p-1$ позволяет однозначно вычислить $1 \leq l \leq p-2$ такое, что $kl = 1(\mathrm{mod}(p-1))$;

3) Вычисляется 1-й элемент подписи

$$r = g^k;$$

4) Вычисляется 2-й элемент подписи

$$s = l(h(m) - ar)(\mathrm{mod}(p-1)).$$

Подписью под документом m объявляется пара (r, s).

Пояснение

Элементы поля $\mathbf{Z}_p$, в частности r, записываются своими стандартными именами. Поэтому $1 \leq r \leq p-1$. Элементы, вычисленные $\mathrm{mod}(p-1)$, в частности s, также записываются своими стандартными именами. Поэтому $0 \leq s \leq p-2$.

Проверка подписи

1) Необходимо знать открытые данные (p, g, y);

2) Проверяем: $1 \leq r \leq p-1$, $1 \leq s \leq p-2$, в противном случае подпись неверна (конечно, случай $s = 0$ возможен, что не допускается волевым распоряжением, устранить недоразумение можно, заменив k, а вместе с ним и r);

3) Вычисляем $v_1 = y^r r^s$;

4) Вычисяем $h(m)$ и $v_2 = g^{h(m)}$;

5) Подпись принимается, если оказывается что $v_1 = v_2$, в противном случае она считается неверной.

Объяснение проверки

Вычисляем:

$$v_1 = y^r r^s = g^{ar} g^{kl(h(m)-ar)} = g^{ar} g^{h(m)-ar} = g^{h(m)} = v_2.$$

При этом использовались равенства $g^{p-1} = 1$, $kl = 1 + (p-1)t$ для некоторого $t \in Z$.

Пример 62. *Выберем* $p = 31, g = 3, a = 5$, *тогда* $y = 3^5 = 26$. *Пусть* n *сопоставляет бинарной последовательности из* ≤ 4 *знаков ее значение как записи числа в* 2*-ичной записи. Например, документу* $m = 1011$ *сопоставляет число* $1+1\cdot 2+0\cdot 2^2+1\cdot 2^3 = 11$*:* $h(m) = 11$.

Пусть $k = 7$, *тогда* $l = 13$.

Вычисляем элементы подписи:

$r = 3^7 = 17,$

$s = 13(11 - 5 \cdot 17) = 13(11 - 25) = -13 \cdot 14 = 13 \cdot 16 = 28(\mathrm{mod}30).$

Проверка подписи

1) $1 \leq r \leq 30 \longleftarrow 1 \leq 17 \leq 30$ *(верно),*

$0 \leq s \leq 29 \longleftarrow 0 \leq 28 \leq 29$ *(верно);*

2) $v_1 = 26^{17} \cdot 17^{28} = 3^{25} \cdot 3^{16} = 3^{41} = 3^{11} = 13.$

$v_2 = 3^{11} = 13.$

Слабости алгоритма

Посмотрим, что произойдет, если в качестве ключей a, k будут выбираться крайние возможные значения этих параметров.

Пусть, например, $a = 1$. Тогда $y = g$ и по виду этого открытого значения легко восстанавливается долгосрочный ключ a. То же самое можно сказать и тогда, когда параметр a имеет небольшие значения или же, наоборот, имеет значение, ненамного отличающееся от $p - 2$. Заметим, что знание a позволяет подделывать подпись на любом документе.

Если значение k близко к крайним возможностям, то по виду $r = g^k$ легко вычислить k, что позволяет подделывать подпись для данной сессии. Более того, знание k позволяет из $s = l(h(m)-$ $-ar)(\mathrm{mod}(p-1))$ вычислить $ks = kl(h(m) - ar) = (h(m) - ar)\cdot$ $\cdot(\mathrm{mod}(p-1))$. Затем можно получить $a = (ks - h(m)))(-r)^{-1}\cdot$ $\cdot(\mathrm{mod}(p-1))$. Тогда можно будет подделывать все подписи. Отсюда вывод: к выбору сессионных ключей необходио относиться столь же серьезно, как и к выбору долгосрочного ключа.

Заметим, что в общем случае вычисление a по y, или k по r – это вычисление дискретного логарифма. Поэтому в общем случае при правильном выборе p – это трудная задача.

Обычным возражением на отмеченные слабости является следующее утверждение: при большом p вероятность выбора значений, близких к крайним, пренебрежимо мала. Это так, но иногда злоумышленник находится в самой системе и выбирает параметр не случайным образом. Поставив слабую подпись, он впоследствии может доказать возможность ее постановки без знания секрета, то есть уйти от юридической ответственности.

Базовая схема Эль Гамаля лежит в основе таких стандартов, как российский стандарт на электронную подпись ГОСТ Р 34.10-94 и стандарт США DSS (Digital Signature Standard) (см. приложение 2).

Задачи и упражнения

Задача 16.1. Рассмотрим следующее упрощение схемы Эль Гамаля. Пусть p – простое число, g – порождающий элемент мультипликативной группы $\mathbf{Z}_p^*$. Эти данные открыты.

Пользователь выбирает секретное число $1 \leq a \leq p-1$ и открывает значение $y = g^a (\mathrm{mod} p)$.

К сообщению $m \in \{0,1\}^*$ применяется известная хэш-функция, дающая значение $h = h(m) \in \mathbf{Z}_p$. Все, как в алгоритме Эль Гамаля.

Допустим, что нод$(h, p-1) = 1$. Этого можно добиться, внеся необходимые коррективы в случае невыполнения. Мы не уточняем, как это можно сделать.

При генерации подписи пользователь вычисляет значение

$$z = h^{-1} a (\mathrm{mod}(p-1)).$$

В качестве подписи под документом m предлагается использовать g^z.

Прверка правильности подписи элементарна:

$$(g^z)^h = g^{h^{-1}ha} = g^a (\mathrm{mod} p).$$

Подпись принимается, если полученное значение совпадет с y.

Объяснить, почему упомянутая схема неприемлема.

Электронные платежи

1. Основные понятия

Изобретателем технологии электронных платежей является Д. Шаум (D. Chom) – основатель и исполнительный директор фирмы DigiCash и одновременно признанный специалист в области криптографии.

DigiCash разработала и запатентовала криптографическую технологию безопасных электронных платежей. Компания Master Card International получила от DigiCash разрешение на использование этой технологии для создания исторически первых электронных карточек Master Card. Это позволило компании Master Card International захватить лидерство в области электронных платежей.

Электронные деньги – это лишь информация о реально существующих средствах.

Нас сейчас интересует только то, что содержит данная информация и с помощью каких криптографических средств она защищена. В этом смысле мы не обращаем внимание ни на защищенность материальных носителей этой информации (электронных карточек, других возможных носителей), ни на тонкости обращения электронных платежей в финансовой сфере.

Конечно, электронные платежи существенно развились за последние 20–30 лет. Они стали многообразными по своим функциональным возможностям. Придумано также немало криптографических протоколов для защиты электронных платежей. Мы не ставим своей целью обозреть все это. Наша задача гораздо более определенная – на основе технологии, лежащей в основе первых электронных платежей с использованием Master Card, выделить

основные принципы их криптографической защиты от несанкционированных пользователей. Далее мы употребляем термины «карточка», «купюра», но это делается только для облегчения изложения. Прежде всего нас интересует информация и ее защищенность в электронных платежах.

2. Необходимые элементы электронной карточки

Электронные карточки, впервые выпущенные компанией Master Card International, были дебетными. Каждая реальная карточка соответствовала определенному номиналу. Далее этот номинал расходовался владельцем карточки без пополнения. Противоположный вариант – кредитная карточка, для которой определялась сумма кредита. Возможны всякие другие варианты. Их отличия лежат в чисто финансовой сфере взаимоотношений банка-эмитента, выпустившего карточку, банков со счетами владельца карточки и тех, кому он по ней платит за покупки (продавцов), самого владельца и продавцов. Нас эти взаимоотношения не интересуют, поэтому мы оставляем одно понятие – «номинал». Карточка должна содержать соответствующую информацию. Будем говорить, что карточка должна содержать номинал. Это во-первых.

Во-вторых, карточка должна иметь номер, благодаря которому банк-эмитент имеет возможность следить за тем, чтобы платежи по данной карточке не вышли за рамки ее номинала.

В-третьих, пожалуй, это самое главное – карточка должна быть информационно защищена. Необходимо оградить записанную на ней информацию от подделки, подлога, замены, использования несанкционированным пользователем, клонирования и т. п. Напомним, что мы оставляем за кадром защищенность носителя информации и соответствующих сетей. Защищенность карточки обеспечивается электронной подписью банка-эмитента.

Перечислим основные требования к деньгам, в том числе и электронным:

- невозможность подделки,

- невозможность вторичного использования уже истраченной суммы (выхода за рамки номинала),

- неотслеживаемость платежей.

С номиналом все достаточно очевидно. Спросим себя: зачем нужен номер? Мы уже давали естественный ответ на этот вопрос: номер нужен для обеспечения второго требования. Если это так, то банку-эмитенту нужно знать только то, сколько еще остается на карточке неизрасходованных средств. Более того, ему даже это не нужно знать. Достаточно знать, хватает ли оставшихся средств на очередной платеж. Банк должен фиксировать номер карточки для правильного снятия со счета сумм платежей.

Посмотрим на третий пункт требований. Реальные бумажные деньги в процессе своего обращения не несут информации о своем владельце. Только в редких случаях, когда производятся какие-то оперативные мероприятия, допускаются специальные метки купюр или переписываются их номера. В регулярной ситуации деньги безлики. Выпуск электронных денег, удовлетворяющих этому требованию, был в свое время большим достижением. Требование обеспечивалось так называемой «слепой» подписью. Ее также придумал Д. Шаум, что ставили ему в заслугу даже в большей мере, чем изобретение технологии электронных платежей. К созданию последней почти одновременно с ним пришли другие, а вот до «слепой» подписи в то время никто, кроме Д. Шаума, не додумался.

Необходимость в выполнении третьего пункта требований значительно возросла тогда, когда население США практически полностью оказалось «под колпаком» налоговых и других спецслужб. Все платежи в связи с широким распространением карточек стали прозрачными. Вот тогда-то и заработала «слепая» подпись.

К сожалению, в настоящее время контролирующие органы США снова вошли в силу. На большинстве электронных карточек красуются фамилии их владельцев. Так называемые кредитные истории пользователей активно учитываются и, кажется, проти-

востоять этой бюрократической машине невозможно. Я уже не говорю о тех странах, где технологии электронных платежей такое требование не предусматривают вовсе.

3. Технология электронного платежа

Участники

1. Банк-эмитент C, осуществляющий эмиссию электронных денег;

2. Клиент A, заказывающий у C электронную карточку (или другое право электронного платежа);

3. Продавец B, имеющий возможность принимать электронные платежи данной системы.

Конечно, «продавцов» в системе обычно довольно много, мы рассматриваем одного, так как остальные функционируют по той же схеме. Кроме перечисленных основных участников могут быть вспомогательные. Например, банки, в которых открыты функционирующие в системе счета участников. Они не бязательно совпадают с банком C. Банк C осуществляет специальные операции, взаимодействуя с ними. Эта чисто финансовая сторона дела нас не интересует. Для простоты считаем, что функционирующие в системе счета всех участников открыты в C.

Заказ электронной карточки

Клиент A посылает в банк C полуфабрикат электронной карточки с ее номером и своей подписью. Банк C стирает подпись клиента и ставит свою. При этом он снимает со счета клиента A указанную там сумму, равную номиналу карточки (теперь она на карточке), и возвращает карточку со своей электронной подписью.

Электронный платеж

При покупке товара (услуг) клиентом A у продавца B происходит следующее:

– B проверяет через C подлинность карточки и наличие на ней достаточных средств для оплаты покупки;

– при выполнении приведенных выше условий B отдает товар клиенту A, параллельно с этим банк C кладет на счет продавца B соответствующую сумму, снимая ее с электронной карточки A.

Пояснение

Проверка подлинности карточки осуществляется одним из следующих способов:

– если продавец B имеет централизованную систему проверки, заключающуюся в постоянной связи с C, он передает в C данные карточки – подпись банка C и номер карточки;

– если система проверки подлинности автономна, то соответствующая связь между B и C осуществляется периодически. Данные передаются либо в определенное время, либо по мере их накопления. В этом случае есть риск вторичных платежей (выхода за номинал) со стороны A. Зато этот способ более дешевый.

Если платеж по карточке осуществляется впервые, то банк C заносит ее номер в базу данных, фиксируя остаток средств на карточке. В последующие платежи банк находит в базе данных номер карточки и производит соответствующие расчеты, вычисляя остаток средств на карточке. Если платеж превосходит остаток, банк не позволяет его произвести.

Обратим внимание на то, что первоначально банк может не знать номера, поэтому ему неизвестен и владелец.

4. Master Card

Приведем исторически первый пример описанной выше технологии, реализованный в карточках, из названия.

Протокол заведения электронной карточки

Установка

Банк C заводит RSA – систему шифрования. Для этого выбираются два больших различных простых числа p и q, вычисляется

модуль $n = pq$ и значение функции Эйлера $\varphi(n) = (p-1)(q-1)$. Выбирается открытый ключ шифрования $1 \leq e \leq \varphi(n) - 1$, взаимно простой с $\varphi(n)$: нод$(e, \varphi(n)) = 1$. Вычисляется секретный ключ дешифрования d такой, что $ed = 1(\mathrm{mod}\varphi(n))$. Кроме этого фиксируется односторонняя функция f: $\mathbf{Z}_n \longrightarrow \mathbf{Z}_n$.

Номинал

В первых Master Cards номинал взаимно-однозначно соответствовал модулю n (банк выбирал несколько различных, но фиксированных модулей).

Номер

Клиент A выбирает случайное число $x \in \mathbf{Z}_n$ (номер купюры) и вычисляет значение $f(x)$.

Полуфабрикат карточки

Клиент A выбирает случайное число $r \in \mathbf{Z}_n^*$, то есть число $1 \leq r \leq n-1$ взаимно простое с n: нод$(r, n) = 1$. Последнее условие гарантирует существование обратного r^{-1} к r mod$n$: $rr^{-1} = 1(\mathrm{mod}n)$. Далее клиент A вычисляет

$$y = f(x)r^e(\mathrm{mod}n),$$

где r^e называется затемняющим множителем. Все вычеты mod n записываются стандартными номерами. Значение y передается банку C.

Электронная карточка с копиркой

Банк C вычисляет:

$$z = y^d = f(x)^d r^{ed} = f(x)^d r(\mathrm{mod}n)$$

и передает это значение клиенту A. При этом C осуществляет все соответствующие банковские операции, упомянутые в описании технологии.

Снятие копирки

Клиент A, получив значение z, вычисляет

$$z \cdot r^{-1} = f(x)^d (\mathrm{mod} n).$$

Теперь он получает карточку с номером x и электронной подписью банка $f(x)^d$.

Примечание

Дополнительно к данному протоколу используются процедуры аутентификации A для C и наоборот.

Протокол транзакции платежа

Проверка подлинности карточки

Продавец B, получив от A электронную карточку, посылает информацию $(f(x)^d, x)$ банку C. Банк C проверяет равенство $(f(x)^d)^e = f(x)(\mathrm{mod} n)$. Если оно выполнено, карточка правильная.

Заметим, что такую проверку при автономной системе осуществляет сам продавец B (точнее, его кассовый аппарат). Ведь для этого достаточно знать открытый ключ e и известную одностороннюю функцию f.

Проверка достаточности средств на карточке

Банк C ищет номер x в базе данных. Если его там нет (первый платеж по данной карточке), то он заносится в базу данных с соответствующим остатком. Если есть, то с карточки снимается сумма платежа и фиксируется новый остаток. Конечно, если платеж превышает остаток, сделка не разрешается.

Как уже отмечалось, в настоящее время существует много различных вариантов технологии электронных платежей. Существуют способы образования карточки, при которых банк C даже не знает, сколько на ней средств. Ведь банку нужно знать, не сколько средств, а хватает ли их на очередной платеж.

К сожалению, все большее распространение получает другая тенденция, когда банк имеет возможность получать полную информацию о платежах конкретного клиента, записывая ее в так называемую кредитную историю.

Что дальше? Когда-то мечтой были электронные карточки. Сам их вид эволюционировал – от обыкновенных магнитных до смарт-карт, являющихся миникомпьютерами. Появились другие виды электронных платежей, которые все более популярны. Потребности в криптографической защите возросли многократно. В Майкрософте все новые технологии должны быть согласованы с отделом криптографии. Появилась новая мечта – о личных деньгах. Как они должны выглядеть – серьезный вопрос. Возможно, что и эта фантастическая идея будет реализована. А, может быть, есть еще какой-нибудь путь? Пока мы не можем точно ответить на подобные вопросы.

Задачи и упражнения

Задача 17.1. Привести схему заказа электронной карточки, в которой фиксируется имя владельца. Предполагается защищенность такой фиксации.

Задача 17.2. Привести схему электронных платежей, при которой банк-эмитент не владеет полной информацией об остатке средств на карточке. Он может лишь судить о достаточности остатка при очередном платеже.

Управление ключами

Assume the cipher is known
to attackers –
– all security is in the key.
Kerchoffs' Principle

1. Основные понятия

В криптографических методах защиты информации ключам отводится особая роль. Обычно предполагается, что система шифрования известна, значит, стойкость шифра определяется целиком и полностью стойкостью ключей. В симметричных системах шифрования и основанных на них протоколах необходима стойкость как ключа шифрования, так и ключа дешифрования. В асимметричных системах, в которых ключ шифрования открыт, стойкость определяется невозможностью подделки или подбора ключа дешифрования.

Задачи управления ключами заключаются в правильном выполнении следующих операций, связанных с жизнью ключей:

– генерация ключей,
– хранение ключей,
– распределение ключей,
– замена ключей,
– уничтожение ключей.

В настоящее время ключи обычно подразделяются на долгосрочные и сессионные (краткосрочные). Кроме этого, существуют понятия мастер-ключей, являющихся главными ключами серверов, ключей шифрования файлов и сеансовых ключей, предназначенных для шифрования передаваемых данных.

Ключ может выглядеть как бинарная последовательность, число, вычет или набор соответствующих последовательностей,

чисел или вычетов. Возможны и существуют более сложные образования, которые можно считать ключами.

В настоящее время принято считать надежными ключи разрядности не менее 80 бит для симметричных систем и не менее 768 бит для асимметричных систем. Эти оценки, конечно, условны. В основном они учитывают возможности подбора путем полного перебора всех возможных ключей.

Например, при разрядности ключа криптосистемы симметричного шифрования в 64 бита (объем ключевого пространства 2^{64}) компьютер, выполняющий 10^6 переборов ключей в секунду, потратит на проверку всех возможных ключей 5–6 тысяч компьютерных лет.

Современная вычислительная техника способна перебирать 56-битные ключи. В 1999 г. удалось вскрыть шифр RSA с ключом разрядности 512 бит. Работа по разложению модуля n шифра на сомножители: $n = pq$ велась в течение 7 месяцев с привлечением для параллельных вычислений ресурсов 292 мощных процессоров. В настоящее время успехи в данном направлении обуславливаются как развитием вычислительной техники, так и достижениями в области анализа.

2. Распределение ключей

Одним из наиболее известных протоколов распределения сеансовых ключей является протокол Цербер (Kerberos – трехголовый пес с хвостом и гривой из змей, который, согласно древнегреческой мифологии, охраняет вход в подземное царство). Протокол разработан в MIT (Массачусетский институт технологий).

Предполагается, что в массовой информационной системе пользователей $A, B, \ldots$ имеется центральный сервер C, задачей которого является генерация и снабжение ключами пользователей для их сообщения между собой. Центр C пользуется доверием всех пользователей системы.

Протокол Цербер (Kerberos)

Установка

Предполагается, что каждый пользователь имеет долгосрочный секретный ключ, разделенный между ним и сервером C. Пользователь A имеет ключ K_{AC}, пользователь B – K_{BC} и т. д.

Распределение ключей

Генерация и распределение сеансовых ключей k между пользователями осуществляется следующим интерактивным многошаговым процессом. Обозначения $E_{K_{AC}}, E_k \ldots$ означают, соответственно, шифрование ключом K_{AC}, ключом k и т. д.

(1) $$A \longrightarrow C$$

A порождает метку N и посылает серверу C информацию: A, B, N, что означает следующее: A желает иметь сообщение с B, метка N ставится, чтобы выделить данное обращение A к C от других обращений.

(2) $$A \longleftarrow C$$

C передает A шифровку $E_{K_{AC}}(k, N, L, B)$, которую A способен прочитать, в ней содержатся указанные в скобке данные k, N, L, B, где L – время жизни сессионного ключа k. Кроме этого, C передает билет (ticket) $E_{K_{BC}}(k, A, L)$, который A прочитать не может.

(3) $$A \longrightarrow B$$

A передает B ticket $E_{K_{BC}}(k, A, L)$ и аутентификатор (autentificator) $E_k(A, T_A, A^*)$, которые B может прочитать: вначале билет, а затем с помощью k аутентификатор. Символ T_A обозначает локальное время, A^* – вспомогательный ключ для возможных рабочих сообщений между A и B.

(4) $$A \longleftarrow B$$

B проверяет, что A один и тот же, что локальное время T_A находится в рамках L и передает A сообщение $E_k(T_A, B^*)$, в котором B^* означает еще один вспомогательный ключ, для тех же целей, что и A^*.

Замечание 63. *1) Использование локального времени предполагает его синхронизацию.*

2) Время L позволяет многократно аутентифицировать A для B без дополнительных обращений к C.

3) Первые версии протокола Цербер предполагали использование DES для шифрования E. В дальнейшем было разрешено использование любой симметричной системы шифрования.

Задачи и упражнения

Задача 18.1. Предположим, что в некоторой структуре используется система шифрования, в которой, как, например, в DES (см. приложение 1), ключи K представляют собой бинарные последовательности фиксированной длины l (в DES эта длина равна 56). Предположим, что ключи хранятся в зашифрованном с использованием системы RSA виде, как

$$C = K^e (\mathrm{mod} n).$$

Здесь, как обычно, $n = pq$ – модуль системы RSA, e – открытый ключ шифра. Ключ K при этом рассматривается как число в двоичной записи. Показать, что с большой вероятностью ключ K может быть раскрыт путем перебора примерного объема $2^{l/2}$. Откуда появилась дополнительная слабость? Какое свойство натуральных чисел может быть использовано?

Эллиптические кривые

1. Определения и основные факты

Пусть F – произвольное поле характеристики, не равной 2 или 3. Пусть $f(x) = x^3+ax+b$ $(a, b \in F)$ – многочлен, взаимно простой со своей формальной производной $f'(x) = 3x^2 + a$. Это означает, что многочлен $f(x)$ не имеет кратных корней. Последнее условие равносильно неравенству для дискриминанта $\Delta = 4a^3 + 27b^2 \neq 0$. Эллиптической кривой над полем F, определяемой многочленом $f(x)$, называется множество точек $(x, y) \in F^2$, удовлетворяющих уравнению

$$y^2 = x^3 + ax + b \tag{1.1}$$

вместе с формальной точкой, обозначаемой 0 и называемой «точкой на бесконечности».

Если поле F имеет характеристику 2 (мы рассмотрим только случай конечного поля $F = \mathbf{F}_{2^r}$ порядка 2^r), то эллиптическая кривая над F определяется аналогично как множество точек $(x, y) \in F^2$, удовлетворяющих уравнению вида

$$y^2 + xy = x^3 + ax^2 + b \qquad (a, b, c \in F), \tag{1.2}$$

где $b \neq 0, a = 0$ или $\sum_{i=0}^{r-1} a^{2^i} = 1$.

Иногда эллиптические кривые над полем характеристики 2 определяются еще уравнением вида

$$y^2 + cy = x^3 + ax + b \qquad (a, b \in F) \tag{1.3}$$

также вместе с «точкой на бесконечности» 0 и с некоторыми ограничениями на коэффициенты. В криптографии такие кривые практически не используются. Мы этот случай не рассматриваем.

Условие на существование кратных корней у правых частей (1.2), (1.3) не накладывается.

Заметим, что уравнения указанного вида являются каноническими. Нужно понимать определение в том смысле, что существует система координат, в которой уравнение имеет указанный вид.

Наконец, в случае поля F характеристики 3 эллиптическая кривая определяется уравнением вида

$$y^2 = x^3 + ax^2 + bx + c \qquad (a, b, c \in F), \tag{1.4}$$

где многочлен, стоящий в правой части, не имеет кратных корней. Как и в рассмотренных выше случаях, выделяют формальную точку 0 кривой «на бесконечности».

Пример 64. *Пусть $K = \mathbf{F}_7$. Рассмотрим кривую E, определенную уравнением $y^2 = x^3 + 2x + 3$. Заметим, что E – эллиптическая кривая. Действительно, нод$(x^3 + 2x + 3, 3x^2 + 2) = 1$. Это видно из следующих вычислений по алгоритму Эвклида:*

$$x^3 + 2x + 3 = (3x^2 + 2) \cdot 5x + (6x + 3),$$

$$3x^2 + 2 = (6x + 3)(4x + 5) + 1.$$

Можно также просто вычислить дискриминант: $\Delta = 5 (\mathrm{mod}\ 7)$.

Замечание 65. *Над полем $\mathbf{F}_5$ кривая $y^2 = x^3 + 2x + 3$ уже не эллиптична, так как нод$(f(x), f(x)') = x + 1$ и $\Delta = 0 (\mathrm{mod} 5)$.*

Нас в первую очередь интересуют эллиптические кривые над конечными полями. Полезно отметить, что эллиптические кривые занимают в математике важное место, так как естественно появляются в самых различных ее областях. Достаточно хорошо изучены эллиптические кривые над полями комплексных, вещественных и рациональных чисел, известные своими приложениями в математическом анализе, геометрии, теории чисел. Мы, конечно, не затрагиваем и даже не упоминаем многие известные

результаты об эллиптических кривых. Наша цель – представить в лаконичном виде необходимые факты из теории эллиптических кривых для криптографии.

Пусть E – эллиптическая кривая над произвольным полем F. Ее важнейшим свойством является возможность определения на множестве точек E операции сложения $+$, относительно которой E становится абелевой группой. Если характеристика поля нечетна, это делается следующим образом.

1. Формальная точка 0 играет роль нуля группы Это означает, что для любой точки $P \in E$ имеем $P+0 = 0+P = P$. В частности, $-0 = 0$.

В пунктах 2–5 мы обозначаем через P, Q произвольные точки E, отличные от 0.

Рассмотрим вначале случай поля F характеристики, отличной от $2, 3$.

2. Легко видеть, что вместе с точкой $P = P(x, y) \in E$ существует точка $-P = -P(x, -y) \in E$. Полагаем $P + (-P) = = (-P) + P = 0$.

3. Известно и легко проверяется, что для пары точек $P(x_1, y_1)$, $Q(x_2, y_2)$ с различными первыми координатами: $x_1 \neq x_2$ прямая $l = l(P, Q)$, через них проходящая, пересекает E не более, чем в трех точках. Пусть l пересечет E еще в точке $R = R(\tilde{x_3}, \tilde{y_3})$. Возможен, правда, случай касания l к E в точке P, тогда полагается $R = P$, или касания в точке Q, тогда $R = Q$. Сумма $P + Q$ равна по определению точке $-R = -R(x_3, y_3)$, где $x_3 = \tilde{x_3}, y_3 = -\tilde{y_3}$.

4. Если $Q = -P$, то есть первые координаты точек Q, P одинаковы, а вторые отличаются знаком, то по определению $P+Q = = Q + P = 0$.

5. Наконец, в случае равенства точек $P = Q$ в качестве l берется касательная к E в точке P, ищется точка R ее пересечения с E и полагается $2P = P + P = -R$, где, как и раньше, точка $-R$ отличается от точки R знаком второй координаты. Если вдруг

оказалось, что P – точка двойного касания к E прямой l, то полагается $R = P$ и далее, как определялось раньше, $2P = -P$, то есть $3P = 0$.

Пример 66. *Рассмотрим кривую E над полем $\mathbf{F}_5$, заданную уравнением $y^2 = x^3 + 2x + 1$. Легко проверить, что нод$(x^3 + 2x + 1, 3x^2 + 2) = 1$, следовательно, кривая E – эллиптическая.*

Непосредственное перечисление дает все точки E: 0, $P_1(0,1)$, $-P_1(0,4)$, $P_2(1,2)$, $-P_2(1,3)$, $P_3(3,2)$, $-P_3(3,3)$.

Попробуем вычислить сумму P_1 и $(-P_2)$. Сначала запишем уравнение прямой $l = P(P_1, -P_2)$, проходящей через эти точки. Общее уравнение прямой $y = \alpha x + \beta$ дает систему

$$\begin{cases} 1 = \beta, \\ 3 = \alpha + \beta. \end{cases}$$

Отсюда получаем $\alpha = 2$, $\beta = 1$; $l : y = 2x + 1$. Решая систему

$$\begin{cases} y = 2x + 1, \\ y^2 = x^3 + 2x + 1, \end{cases}$$

получаем уравнение $x^3 + x^2 + 3x = 0$, откуда $x_1 = 0$, что соответствует точке P_1, $x_2 = 1$, что соответствует точке P_2, $x_3 = 3$, что соответствует точке $R = P_3$. Значит, $P_1 + P_2 = -P_3$.

Мы не приводим вывод соответствующих формул для сложения точек эллиптической кривой. Запишем только их окончательный вид.

Если $P(x_1, y_1)$, $Q(x_2, y_2)$ – точки кривой E, отличные от 0 и имеющие различные 1-ые координаты: $x_1 \neq x_2$, то координаты их суммы $R(x_3, y_3) = P(x_1, y_1) + Q(x_2, y_2)$ вычисляются по формулам:

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - (x_1 + x_2),$$

$$y_3 = -y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3). \qquad (1.5)$$

Координаты точки $R(x_3, y_3) = 2P(x_1, y_1)$ вычисляются по формулам:

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1,$$

$$y_3 = -y_1 + \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3), \qquad (1.6)$$

где, конечно, $y_1 \neq 0$. Если $y_1 = 0$, то $2P(x_1, y_1) = 0$ по определению. Формулы для сложения точек с противоположными по знаку первыми координатами очевидны – сумма равна 0.

Случай поля $F = F_{2^r}$ особый. Точка $-P$ в нем для точки $P(x_1, y_1)$ имеет по определению координаты $-P(x_1, x_1 + y_1)$.

Если $P + Q = R \neq 0$, то координаты точки $R(x_3, y_3)$ определяются по координатам точек $P(x_1, y_1), Q(x_2, y_2)$ согласно следующим формулам.

Если $x_1 \neq x_2$, то

$$x_3 = (\frac{y_1 + y_2}{x_1 + x_2})^2 + \frac{y_1 + y_2}{x_1 + x_2} + a + x_1 + x_2,$$

$$y_3 = (\frac{y_1 + y_2}{x_1 + x_2} + 1)x_3 + \frac{y_1 x_2 + y_2 x_1}{x_1 + x_2}.$$

Если $x_1 = x_2$, то есть $P = Q$, то

$$x_3 = (\frac{x_1^2 + y_1}{x_1})^2 + \frac{x_1^2 + y_1}{x_1} + a,$$

$$y_3 = (\frac{x_1^2 + y_1}{x_1} + 1)x_3 + x_1^2.$$

Мы не приводим здесь формулы для эллиптических кривых над полем F характеристики 3.

Замечание 67. *Важно отметить, что группа $G = G(E)$ эллиптической кривой над любым полем F абелева по определению. Однако, если $F = \mathbf{F}_q (q = p^r)$ – конечное поле, она не обязана быть*

циклической (в отличие от группы $\mathbf{F}_q^*$). Можно доказать, что группа G в этом случае либо циклическая, либо является прямой суммой двух циклических групп. Она однозначно представляется в виде прямой суммы $G = \mathbf{Z}_{n_1} \oplus \mathbf{Z}_{n_2}$, где $\mathbf{Z}_n$ обозначает циклическую группу порядка n, n_2 делит n_1, более того, n_2 делит $q-1$.

Теперь мы поговорим о порядке $|G|$ этой группы, очевидно, ограниченном сверху числом $2q+1$.

Оценка $|G| \leq 2q+1$ слишком завышена. Хорошее приближение дает следующая теорема.

Теорема 68. *(Теорема Хассе). Пусть $N = |G|$ - порядок группы $G = G(E)$, тогда*

$$t = |N - (q+1)| \leq 2\sqrt{q}. \tag{1.7}$$

Кривая E называется *суперсингулярной*, если $t^2 = 0, q, 2q, 3q$ или $4q$. В противном случае она называется *не суперсингулярной*. Если q есть степень 2 и E суперсингулярная, то порядок группы $G = G(E)$ нечетен, а если не суперсингулярная, то четен. Согласно теореме Ватерхауза [28] при простом q для любого числа t такого, что $|t| \leq 2\sqrt{q}$, существует по крайней мере одна эллиптическая кривая E, определенная над $\mathbf{F}_q$, для которой $N = |G(E)| = q+1-t$. Если q является степенью 2, то для любого нечетного t такого, что $|t| \leq 2\sqrt{q}$, существует по крайней мере одна не суперсингулярная эллиптическая кривая E, определенная над $\mathbf{F}_q$, для которой $N = |G(E)| = q+1-t$.

2. Оцифровка эллиптических кривых

Как говорилось в самом начале курса, обычная работа идет с оцифрованными текстами. В лекции 8 было объяснено, как цифруются элементы конечного поля. Если мы хотим использовать для шифрования эллиптические кривые (более точно – группы эллиптических кривых), мы должны уметь оцифровывать эти кривые. К сожалению, до сих пор неизвестен хороший способ это делать определенно. Однако существует вполне приемлемый метод

вероятностной оцифровки, к изложению которого мы сейчас приступаем.

Вероятностный метод оцифровки эллиптической кривой

Предположим, что любая единица m текста ограничена неравенствами $0 \le m < M$. Выберем поле $\mathbf{F}_q$, для порядка которого выполнено неравенство $q > Mk$ при достаточно большом фиксированном числе k (на практике обычно достаточно брать k между 20 и 50). Числа от 1 до Mk записываются в виде $mk + j$, где $1 \le j \le k$. Устанавливается взаимно однозначное соответствие между ними и Mk первыми (по стандартной оцифровке $\mathbf{F}_q$) элементами поля $\mathbf{F}_q$. Напомним, что каждое число t от 1 до Mk можно записать в p-ичном выражении $t = (\alpha_{r-1}, \dots \alpha_1\alpha_0)_p$, затем взять многочлен $\alpha_0+\alpha_1 x+\ldots+\alpha_{r-1}x^{r-1} \in \mathbf{Z}_p[x]$, представляющий элемент поля $\mathbf{F}_q$, и считать его соответствующим числу t.

Таким образом для данного m мы получим k элементов поля $\mathbf{F}_q$, отвечающих числам $mk+1, mk+2, \dots, mk+k$. Последовательно берем в качестве x эти элементы. Для каждого из них пытаемся найти $y \in \mathbf{F}_q$ из уравнения (1.1)

$$y^2 = f(x) = x^3 + ax + b$$

(это для случая, если $p \ne 2, 3$, для исключительных случаев используем другие уравнения (1.2)–(1.4)). Если мы найдем y такое, что $y^2 = f(x)$, мы полагаем $P_m = (x, y)$ – точка, соответствующая m. Таким образом мы сопоставим (инъективно) всем возможным единицам текста точки эллиптической кривой E.

Легко видеть, что $m = [x - 1)/k]$ (целая часть), что позволяет по элементу поля $\mathbf{F}_q$ (точнее, его номеру) однозначно и эффективно восстанавливать m.

Замечание 69. *Поскольку $f(x)$ является полным квадратом примерно в 50 % случаев, вероятность неуспеха в сопоставлении m точки P_m не превосходит $\frac{1}{2^k}$. Известно более точное утверждение: $f(x)$ является квадратом $\sim$ в $N/2q$ случаях.*

Задачи и упражнения

Задача 19.1. Проверить, что уравнение

$$y^2 = x^3 + 2x + 1$$

задает эллиптическую кривую E над полем $\mathbf{Z}_5$.

Найти все элементы группы $G(E)$. Привести таблицу сложения этих элементов. Какова алгебраическая структура этой группы? Будет ли она циклической?

Задача 19.2. Построить эллиптические кривые над полями $\mathbf{F}_{2^4}, \mathbf{F}_{13}, \mathbf{F}_{5^2}$, привести таблицы сложения для соответствующих групп. Какова их алгебраическая структура?

Лекция 20

Криптография, основанная на эллиптических кривых

1. Аналог возведения в степень

Из лекции 9 известно, как можно использовать мультипликативную группу конечного поля $\mathbf{F}_q^*$ $(q = p^r)$ в качестве платформы для построения криптосистемы. Наиболее популярными являются схемы, в основе которых лежит сложность разрешения проблемы дискретного логарифма. Целью настоящей лекции является построение криптосистемы, аналогичной вышеупомянутой, но основанной на сложности похожей проблемы для абелевой группы $G = G(E)$, соответствующей эллиптической кривой E над конечным полем F.

Итак, пусть E – эллиптическая кривая, определенная над полем $F = \mathbf{F}_q$ $(q = p^r)$, $G = G(E)$ – ее группа. В данном случае G – конечная абелева группа, которую мы записываем с аддитивной операцией. Таким образом, возведению в степень k в группе $\mathbf{F}_q^*$ теперь соответствует вычисление элемента kP, где P – точка на E.

Замечание 70. *Пусть $N = |E|$ – число точек на E. Тогда N – также порядок группы G: $N = |G|$. Значит, $NP = 0$ для любой точки $P \in E$. Поэтому можно считать, что* $0 \leq k \leq N - 1$.

Из предыдущей лекции известно, как можно эффективно сопоставлять единицам исходного текста точки эллиптической кривой E. Будем иметь это в виду.

2. Дискретный логарифм над эллиптической кривой

Определение 71. *Если E – эллиптическая кривая над полем $F = \mathbf{F}_q$ $(q = p^r)$, а $B \in E$ – фиксированная точка кривой, то проблемой вычисления дискретного логарифма относительно B на E называется проблема решения уравнения вида $xB = P$ для произвольной точки $P \in E$. Будем писать: $x = \log_B P$, хотя такое значение x даже при ограничении $0 \leq x \leq N - 1$ может оказаться неоднозначным.*

Конечно, при неудачном выборе точки B эта проблема может получиться довольно простой. Например, это происходит, если B как элемент группы $G = G(E)$ имеет малый порядок. Если группа G циклическая, то в качестве B желательно брать ее порождающий элемент. Тогда при ограничении $0 \leq x \leq N - 1$ число $x = \log_B P$ определяется однозначно. В противном случае (когда G – прямое произведение двух циклических или B не порождает группу G) логарифм $x = \log_B P$ может и не существовать.

Тем не менее, похоже, что проблема вычисления дискретного логарифма для эллиптических кривых в общем случае более неподатлива, чем соответствующая проблема для мультипликативных групп конечных полей. Сильная техника, развитая в теории полей, в данной ситуации неприменима. Например, известны эффективные алгоритмы вычисления дискретных логарифмов в группах $\mathbf{F}_{2^r}^*$, если, конечно, r не слишком велико. В то же время даже для полей $\mathbf{F}_{2^r}$ с относительно небольшими значениями r дискретный логарифм на эллиптических кривых, определенных над этими полями, трудно вычислим в общем случае. Поля $\mathbf{F}_{2^r}$ наиболее удобны для компьютеров, вот почему шифрование с использованием эллиптических кривых над ними столь популярно.

Замечание 72. *Долгое время не существовало субэкспоненциальных алгоритмов нахождения дискретных логарифмов над эллиптическими кривыми, если только порядок группы $G = G(E)$*

делился на достаточно большое простое число. Однако в самом конце XX в. Менезес, Окамото и Ванстоун (Menezes, Okamoto, Vanstone) [24], используя спаривание Вейля, указали вложение G в мультипликативную группу некоторого расширения поля $\mathbf{F}_{q^k} \supseteq \mathbf{F}_q$ $(q = p^r)$, для которого проблема вычисления дискретного логарифма в G сводится к аналогичной проблеме в $\mathbf{F}^*_{q^{kr}}$. Однако, по существу, единственными эллиптическими кривыми, для которых k мало, являются так называемые суперсингулярные кривые, среди которых наиболее известны кривые вида:

$$E : y^2 = x^3 + ax, p = 1(\mathrm{mod}4),$$

$$E : y^2 = x^3 + b, p = 1(\mathrm{mod}3).$$

Большинство эллиптических кривых суперсингулярностью не обладает, для них данное сведение задачи не облегчает.

3. Аналог алгоритма Диффи – Хеллмана генерации ключа

Допустим, что два корреспондента A и B договариваются о генерации секретного ключа, который затем может быть использован для шифровки и дешифровки в классической криптосистеме или для каких-нибудь других целей. Сначала они выбирают открытым способом конечное поле $\mathbf{F}_q$ $(q = p^r)$. Затем строят случайным образом эллиптическую кривую E над $\mathbf{F}_q$. Далее выбирается случайная точка $C \in E$. Все перечисленные данные открыты. Можно считать, что они передаются по открытой сети. Предполагается, что порядок элемента C группы $G = G(E)$ достаточно большой. Для генерации ключа A выбирает секретное натуральное число a, вычисляет точку $aC \in E$ и передает ее координаты корреспонденту B. Тот, в свою очередь, выбирает секретное число b, вычисляет точку $bC \in E$ и передает (открыто) ее координаты корреспонденту A. Секретный ключ, который они генерируют для дальнейшего использования, есть $abC = baC \in E$. Он получается умножением сообщения bC на a корреспондентом A и

умножением сообщения aC на b корреспондентом B. Возможно, что ключом будет служить не пара координат точки $abC \in E$, а ее первая координата или комбинация координат, что, конечно, уже не принципиально.

Возможный перехватчик должен восстановить a по aC или b по bC. Для этого ему необходимо решить проблему вычисления дискретного логарифма, что, как уже отмечалось, является трудной задачей. В данном случае можно повторить уже сказанное ранее о слабостях алгоритма из-за его неавторизованности.

4. Аналог протокола Масси – Омуры

Пусть E – известная эллиптическая кривая известного порядка N над полем $\mathbf{F}_q$ $(q = p^r)$. Каждый из множества корреспондентов выбирает число e такое, что нод$(e, N) = 1$, и находит число d такое, что $ed = 1(\mathrm{mod} N)$.

Допустим, что корреспондент A (числа которого e_A, d_A соответственно) хочет переслать корреспонденту B (числа которого e_B, d_B соответственно) послание m, закодированное точкой $P = P_m \in E$. Для этого он вначале вычисляет и передает B точку $e_A P_m$, получает в ответ вычисленную тем точку $e_B e_A P_m$, вычисляет и передает точку $d_A e_B e_A P_m = e_B P_m$, что позволяет корреспонденту B получить P_m (а значит, и m) умножением: $d_B e_B P_m = P_m$.

5. Аналог протокола Эль Гамаля

Пусть E – эллиптическая кривая над полем $\mathbf{F}_q$ $(q = p^r)$, $C \in E$ – ее фиксированная точка. Каждый из пользователей выбирает секретное число e и открыто публикует точку eC. Чтобы послать сообщение m корреспонденту B (число которого e_B), корреспондент A посылает ему пару точек $(kC, P_m + k(e_B C))$ кривой E. Для прочтения сообщения B умножает kC на e_B и вычитает результат из $P_m + k(e_B C)$ (снимает маску). В итоге он получает P_m (то есть сообщение m). Во всех вычислениях k – случайное число, используемое один раз.

Задачи и упражнения

Задача 20.1. Привести примеры протоколов Диффи – Хеллмана и Масси – Омуры для эллиптических кривых над полями $\mathbf{F}_5, \mathbf{F}_{13}, \mathbf{F}_{2^4}$.

Задача 20.2. Пусть E – эллиптическая кривая над полем $\mathbf{F}_{2^4}$, заданная уравнением $y^2 + y = x^3$. Показать, что для любой точки $P \in E$ имеет место равенство $3P = 0$. Это означает, что группа $G(E)$ имеет период 3.

Сколько точек содержит группа $G(E)$? Какова ее алгебраическая структура?

Лекция 21

Алгебраическое шифрование

1. Основные понятия

Сравнительно недавно, лет 7–8 назад, появилось новое направление в криптографии, отличительной особенностью которого является использование в качестве платформ бесконечных некоммутативных групп. Еще не окончательно оформилось само направление и не устоялось его название. Часто используют термин «криптография, базирующаяся на группах» («group based cryptography»). Я предпочитаю применять термин «алгебраическая криптография», полагая, что в скором времени к группам в качестве платформ добавятся другие алгебраические системы. Основы этого направления изложены в монографии А. Мясникова, В. Шпильрайна и А. Ушакова [26].

Новое направление открывает новые возможности и ставит новые задачи.

Очень важен выбор класса абстрактных групп, используемых в качестве платформы. Интуитивно ясно, что эти группы должны быть наделены каноническими формами записи их элементов. По-видимому, процесс переписки произвольного элемента, записанного в виде группового слова от порождающих элементов, в каноническую форму должен быть эффективным. Криптостойкость должна основываться на каких-то трудноразрешимых задачах в этих группах.

Настоящая лекция только слегка затрагивает обозначенное направление в криптографии. По сути дела, мы приводим одну схему, использующую в качестве платформ группы кос Артина. В настоящее время это весьма популярная тема. Группы кос Артина достаточно хорошо изучены, в них можно эффективно выполнять вычисления разного толка. В то же время существуют

трудноразрешимые проблемы, дающие возможность построения стойких криптосистем.

Данная лекция предполагает довольно существенное знакомство с абстрактными группами.

2. Группы кос Артина

Группы, о которых идет речь, определяются своими конечными представлениями вида

$$\mathbf{B}_n =< \sigma_1, \ldots, \sigma_{n-1} : \sigma_j\sigma_i = \sigma_i\sigma_j \qquad \text{при } |i-j| \geq 2,$$

$$\sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \qquad \text{при } |i-j| = 1 >,\ n \geq 2. \tag{2.1}$$

Очевидно, что естественное отображение группы $\mathbf{B}_n$ в группу $\mathbf{B}_{n+1}$, сопоставляющее порождающим элементам $\sigma_1, \ldots \sigma_{n-1}$ точно так же обозначенные порождающие элементы группы $\mathbf{B}_{n+1}$, является вложением. Таким образом, группы $\mathbf{B}_2, \mathbf{B}_3, \ldots, \mathbf{B}_n, \ldots$ образуют индуктивную возрастающую систему групп. Группа $\mathbf{B}_2$ – бесконечная циклическая группа, группы $\mathbf{B}_n$ при $n \geq 3$ уже некоммутативны.

Существуют также геометрические определения групп кос, о которых мы здесь подробно не говорим. Из литературы о группах кос упомянем монографии [8; 18].

В группах кос можно указать однозначные формы записи их элементов. Для этого вначале индуктивно определим элементы:

$$\Delta_1 = 1, \Delta_{m+1} = \Delta_m\sigma_m\sigma_{m-1}\ldots\sigma_1.$$

Назовем косу (то есть элемент группы кос) положительной, если ее можно записать в качестве полугруппового слова от порождающих элементов (не включающего в запись обратных к порождающим). Пусть $\mathbf{B}_n^+$ означает полугруппу положительных кос. Оказывается, что любую косу $b \in \mathbf{B}_n$ можно однозначно записать в виде $b = \Delta_n^k \cdot c$, где $k \in \mathbf{Z}$ – максимально возможный показатель, а элемент c принадлежит $\mathbf{B}_n^+$. Существует очевидный гомоморфизм группы $\mathbf{B}_n$ в группу подстановок $\mathbf{S}_n$. Оказывается,

что можно выбрать такие элементы $b_1, \ldots, b_l \in \mathbf{B}_n^+$, $l = n!$, образы которых взаимнооднозначно соответствуют элементам группы $\mathbf{S}_n$. Такие косы называются простыми. Каждая коса $b \in \mathbf{B}_n$ допускает единственное разложение вида $b = \Delta_n^k b_{i_1} \ldots b_{i_t}$, где $k \in \mathbf{Z}$ максимально, $b_{i_1}, \ldots, b_{i_t}$ – простые косы. Значит, канонической формой элемента b можно считать набор $(k, b_{i_1}, \ldots b_{i_t})$. Важно отметить, что геометрическая структура, связанная с каноническими формами, является автоматной. Это, в частности, означает, что процесс переписки в каноническую форму не более, чем квадратичен.

3. Схема Аншель – Аншеля – Голдфилда (L. Anshel, M. Anshel, G. Goldfeld)

В работе [17] предложена следующая схема генерации ключа.

Установка

Корреспонденты A и C выбирают (открыто) группу $\mathbf{B}_n$. Кроме этого они выбирают (также открыто) два набора элементов: $a_1, \ldots, a_k \in \mathbf{B}_n$, $c_1, \ldots, c_l \in \mathbf{B}_n$.

Алгоритм

Корреспондент A выбирает секретный элемент $u = u(a_1, \ldots, a_k)$ и вычисляет набор элементов $c_1' = u^{-1}c_1u, \ldots, c_l' = u^{-1}c_lu$. При этом A публикует (открыто посылает корреспонденту C) нормальные формы элементов $c_1', \ldots, c_l'$.

Корреспондент C аналогичным образом выбирает секретный элемент $v = v(c_1, \ldots, c_l)$, вычисляет нормальные формы элементов $a_1' = v^{-1}a_1v, \ldots, a_k' = v^{-1}a_kv$ и посылает их в открытом виде корреспонденту A.

Далее A вычисляет нормальную форму элемента $w_A = u^{-1}u(a_1', \ldots, a_k') = u^{-1}v^{-1}uv = [u, v]$.

В свою очередь, C вычисляет нормальную форму элемента $w_C = v^{-1}v(b_1', \ldots, b_l') = v^{-1}u^{-1}vu = [v, u] = [u, v]^{-1}$.

Секретный ключ, известный теперь обоим корреспондентам, есть элемент $[u, v]$.

Криптостойкость системы основана на трудноразрешимой задаче нахождения по нормальной форме элемента вида $x^{-1}dx$ и элементу d сопрягающего элемента x. Иногда такую проблему называют «проблемой сопряженности». Но это не совсем точно. Классическая проблема сопряженности для группы G есть поиск ответа на вопрос: сопряжены ли два произвольных элемента $g, f \in G$? В данном случае мы заранее знаем, что элементы a_i' и a_i (c_j' и c_j) сопряжены ($i = 1, \ldots, k$; $j = 1, \ldots, l$). Необходимо найти сопрягающий элемент.

О трудности данной проблемы поиска сопряженного элемента можно говорить только при правильном выборе как элементов $a_1, \ldots a_k$, $b_1, \ldots, b_l$, так и элементов u, v. Ряд математиков верят, что эта проблема вычислительно трудна в общем случае. Автор скорее придерживается скептического взгляда на возможность использования групп кос в качестве платформ для шифрования. Во всяком случае, в предложенных ранее вариантах. Более подходящими для этой цели он считает бесконечные нильпотентные и разрешимые группы с хорошими каноническими формами элементов (например, свободные нильпотентные и свободные разрешимые группы достаточно высоких степеней).

К сожалению, рамки предлагаемого курса позволяют только обозначить данное направление.

Существуют и другие возможности развития криптографии, в том числе связанные с развитием физики (квантовое шифрование). Какие из них будут реализованы, покажет время. Автор же заканчивает этот курс лекций, сопровождая его четырьмя приложениями и списком литературы.

Задачи и упражнения

Задача 21.1. Привести конкретный пример разделения ключа в схеме Аншелm – Аншеля – Голдфилда.

Задача 21.2. Объяснить, почему использование вместо группы кос $\mathbf{B}_n$ свободной группы F_n в системе Аншель – Аншеля – Голдфилда не обладает достаточной криптостойкостью.

Приложение 1

Стандарт шифрования DES

1. История создания

В конце 60-х гг. XX столетия фирма IBM инициировала научно-исследовательский проект в области компьютерной криптографии. Одним из лидеров в проекте стал Хорст Фейстель (Horst Feistel), до этого принимавший участие во многих прикладных проектах, в частности, в разработке для ВВС США системы «свой-чужой». Под его руководством был разработан принцип шифрования, названный его именем. В 1971 г. был создан алгоритм шифрования LUCIFER, проданный банку Ллойда в Лондоне для применения в системе оборота наличных денег. Шифр использовал 128-битный ключ, перерабатывающий 64-битные блоки исходного текста в 64-битные блоки шифрованного текста.

В дальнейшем группа разработчиков под руководством Уолтера Тачмана (Walter Tuchman) и Карла Мейера (Carl Meyer) усовершенствовала LUCIFER. Поэтому, когда в 1973 г. Национальное бюро стандартов США (тогда NBS, позднее – NIST) объявило конкурс на создание общегосударственного стандарта шифрования, компания IBM смогла представить на него вариант, разработанный Тачманом и Мейером. В результате он стал победителем, в 1977 г. был утвержден в качестве государственного стандарта шифрования DES (Data Encryption Standard).

Появление DES – событие значительной важности. Кроме того, что это был первый такой стандарт, получивший вскоре самое широкое распространение в мире, это был первый продукт такого рода, описание которого стало доступно широкой аудитории. Споры относительно DES не утихли до сих пор, как нет и полной ясности в истории его появления. Дело в том, что при разработке проекта его исполнители обратились в Национальное секретное

агенство (NSA) – организацию гораздо более мощную, чем всем известные ЦРУ и ФБР вместе взятые. Долгое время вплоть до 1990-х гг. она отрицала само свое существование. Эксперты NSA внесли ряд изменений в проект, главное из которых – замена 128-битного ключа на 56-битный. Кроме этого, они внесли некоторые изменения в ядро алгоритма – так называемые S-таблицы. Ряд независимых криптографов обвиняли NSA в том, что таким образом они нашли тайную возможность дешифровки DES без знания ключа. До сих пор это обвинение не снято, не подтверждено и не опровергнуто.

Публикация алгоритма DES стала беспрецедентным событием в истории криптографии. Никогда ранее продукт, одобренный NSA, не публиковался в открытой печати. Возможно, что это было следствием недопонимания со стороны NSA, считавшей, что как и раньше, все будет реализовано «в железе», то есть аппаратно. В дальнейшем NSA признало появление DES, а точнее выход из-под контроля, как одну из самых больших своих ошибок.

Как бы то ни было, DES вышел в мир, как джинн из бутылки, и быстро стал популярным. За 30 лет своего существования алгоритм ничем себя не скомпрометировал. 56-битный ключ был слабым с самого начала. При этом он был насильно навязан разработчикам. Современные параллельные вычисления на мощных процессорах сделали DES уязвимым только по этой причине. На практике в основном используется тройной DES, речь о котором пойдет далее.

2. Описание DES

Алгоритм шифрования блочный, он перерабатывает 64-битные блоки исходного текста в 64-битные блоки шифрованного текста. Оба текста записываются в бинарном алфавите. В процессе шифрования используется задаваемый с самого начала 64-битный ключ K. На самом деле 8 бит ключа K не оказывают никакого воздействия на процесс шифрования, они считаются разделительными (это биты с номерами, кратными 8: 8, 16, 24, ..., 64).

По существу, используется только 56 бит ключа K. В процессе шифрования задействованы: S-таблицы, расширение E, перестановка P, начальная перестановка IP. Из ключа K генерируется 16 раундовых ключей $K_1, K_2, \ldots, K_{16}$, каждый по 48 бит. Процесс шифрования включает 16 раундов, использующих последовательно ключи $K_1, K_2, \ldots, K_{16}$.

3. Генерация раундовых ключей

Вначале определим числа v_i $(i = 1, \ldots, 16)$, полагая $v_i = 1$ для $i = 1, 2, 9, 16$ и $v_i = 2$ в остальных случаях.

Далее берем 64-битный исходный ключ $K = k_1 k_2 \ldots k_{64}$ и применяем к нему перестановку $PC1$:

$$\left(\begin{array}{ccccccc} 57 & 49 & 41 & 33 & 25 & 17 & 9 \\ 1 & 58 & 50 & 42 & 34 & 26 & 18 \\ 10 & 2 & 59 & 51 & 43 & 35 & 27 \\ 19 & 11 & 3 & 60 & 52 & 44 & 36 \\ == & == & == & == & == & == & == \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 \\ 7 & 62 & 54 & 46 & 38 & 30 & 22 \\ 14 & 6 & 61 & 53 & 45 & 37 & 29 \\ 21 & 13 & 5 & 28 & 20 & 12 & 4 \end{array}\right).$$

Выходом будет 56-битная последовательность

$$K' = k_{57} k_{49} \ldots k_9 k_1 \ldots k_{44} k_{36} || k_{63} k_{55} \ldots k_{12} k_4,$$

по виду которой можно догадаться, как действует перестановка $PC1$. Ее можно назвать «перестановка с убыванием», поскольку биты $k_8, k_{16}, \ldots, k_{64}$ с номерами, кратными 8, исчезают, а остальные биты меняют порядок следования. Исчезнувшие биты уже не появятся, значит, от них ничего не зависит.

Для произвольной последовательности символов $abc \ldots xyz$ определим циклические сдвиги:

$$abc \ldots xyz \hookleftarrow 1 = bc \ldots yza,$$

$$abc\ldots xyz \leftarrow 2 = c\ldots yzab.$$

Для $i = 1, 2, \ldots, 16$ вычисляем раундовые ключи $K_i = C_iD_i$, где C_i – 24-битная левая, D_i – 24-битная правая половина ключа K_i. Для этого вначале вычислим заготовки ключей $K'_i = C'_iD'_i$, $i = 0, 1, ..., 16$, в которых C'_i, D'_i также обозначают левую и правую (28-битные) половины соответственно.

Полагаем $K'_0 = K' = C'_0D'_0$ и далее: $C'_i = C'_{i-1} \leftarrow v_i$, $D'_i = D'_{i-1} \leftarrow v_i$, и наконец: $K_i = C_iD_i = PC2(C'_i, D'_i), i = 1, ..., 16$, где $PC2$ – перестановка со сжатием:

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

Итак, мы получили 16 раундовых 48-битных ключа $K_1, K_2, \ldots$ $\ldots, K_{16}$.

4. Алгоритм шифрования

Ввод: 64-битный блок исходного текста $M = m_1, \ldots, m_{64}$.

1-й шаг. Начальная перестановка исходного текста

$$IP(m_1, \ldots, m_{64}) = L_0R_0,$$

где $L_0 = m_{58}\ldots m_8$, $R_0 = m_{57}\ldots m_7$ – левая и правая половины получившейся последовательности соответственно, IP – начальная перестановка:

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

2-й шаг (16 раундов). Основные преобразования

Начинаем с L_0R_0. Допустим, что уже построена последовательность $L_0R_0, ..., L_{i-1}R_{i-1}$, где, как обычно, L_{i-1} – левая, а R_{i-1} – правая половины элемента последовательности.

Далее пользуемся следующим правилом:

$$L_i = R_{i-1},$$

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

где $\oplus$ – операция побитного сложения mod2,

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i)).$$

Для вычисления $f(R_{i-1}, K_i)$ вначале применяется перестановка с расширением E:

$$\begin{pmatrix} 32 & 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 9 & 10 & 11 & 12 & 13 \\ 12 & 13 & 14 & 15 & 16 & 17 \\ 16 & 17 & 18 & 19 & 20 & 21 \\ 20 & 21 & 22 & 23 & 24 & 25 \\ 24 & 25 & 26 & 27 & 28 & 29 \\ 28 & 29 & 30 & 31 & 32 & 1 \end{pmatrix}.$$

На выходе получаем 48-битную последовательность, которую складываем операцией побитного сложения $\oplus$ с раундовым ключом K_i.

Результат – 48-битная последовательность разбивается на восемь 6-битных блоков $B_1B_2\dots B_8$. По каждому блоку $B_i = b_1b_2b_3b_4b_5b_6$ определяем два числа: $r = 2b_1 + b_6$ и $c = 8b_2 + 4b_3 + 2b_4 + b_5$, лежащие в пределах: $0 \le r \le 3$, $0 \le c \le 15$. Полагаем: r – номер строки в таблице S_i (соответствующей блоку B_i, $i = 1, 2, \dots 8$), c – номер столбца в S_i. На пересечении этих строки и столбца стоит число $0 \le q \le 15$, которое мы записываем в 2-ичной системе как $q = (q_4q_3q_2q_1)_2 = 8q_4 + 4q_3 + 2q_2 + q_1$. Получаем блок $Q_i = q_4q_3q_2q_1$. Результатом S-преобразования последовательности блоков $B_1B_2\dots B_8$ будет последовательность блоков $Q_1Q_2\dots Q_8$, имеющая 32 бита.

Таблица S имеет вид:

S_1:

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	14	4	13	1	2	15	11	8
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(0)	3	10	6	12	5	9	0	7
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1)	0	15	7	4	14	2	13	1
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(1)	10	6	12	11	9	5	3	8
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(2)	4	1	14	8	13	6	2	11
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(2)	15	12	9	7	3	10	5	0
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(3)	15	12	8	2	4	9	1	7
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(3)	5	11	3	14	10	0	6	13

S_2 :

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	15	1	8	14	6	11	3	4
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(0)	9	7	2	13	12	0	5	10
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1)	3	13	4	7	15	2	8	14
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(1)	12	0	1	10	6	9	11	5
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(2)	0	14	7	11	10	4	13	1
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(2)	5	8	12	6	9	3	2	15
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(3)	13	8	10	1	3	15	4	2
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(3)	11	6	7	12	0	5	14	9

,

S_3 :

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	10	0	9	14	6	3	15	5
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(0)	1	13	12	7	1	4	2	8
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1)	13	7	0	9	3	4	6	10
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(1)	2	8	5	14	12	11	15	1
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(2)	13	6	4	9	8	15	3	0
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(2)	11	1	2	12	5	10	14	7
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(3)	1	10	13	0	6	9	8	7
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(3)	4	15	14	3	11	5	2	12

,

S_4 :

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	7	13	14	3	0	6	9	10
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(0)	1	2	8	5	11	12	4	15
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1)	13	8	11	5	6	15	0	3
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(1)	4	7	2	12	1	10	14	9
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(2)	10	6	9	0	12	11	7	13
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(2)	15	1	3	14	5	2	8	4
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(3)	3	15	0	6	10	1	13	8
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(3)	9	4	5	11	12	7	2	14

S_5 :

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	2	12	4	1	7	10	11	6
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(0)	8	5	3	15	13	0	14	9
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1)	14	11	2	12	4	7	13	1
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(1)	5	0	15	10	3	9	8	6
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(2)	4	2	1	11	10	13	7	8
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(2)	15	9	12	5	6	3	0	14
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(3)	11	8	12	7	1	14	2	13
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(3)	6	15	0	9	10	4	5	3

,

S_6 :

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	12	1	10	15	9	2	6	8
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(0)	0	13	3	4	14	7	5	11
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1)	10	15	4	2	7	12	9	5
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(1)	6	1	13	14	0	11	3	8
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(2)	9	14	15	5	2	8	12	3
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(2)	7	0	4	10	1	13	11	6
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(3)	4	3	2	12	9	5	15	10
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(3)	11	14	1	7	6	0	8	13

,

S_7 :

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	4	11	2	14	15	0	8	13
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(0)	3	12	9	7	5	10	6	1
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1)	13	0	11	7	4	9	1	10
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(1)	14	3	5	12	2	15	8	6
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(2)	1	4	11	13	12	3	7	14
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(2)	10	15	6	8	0	5	9	2
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(3)	6	11	13	8	1	4	10	7
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(3)	9	5	0	15	14	2	3	12

,

S_8 :

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	13	2	8	4	6	15	11	1
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(0)	10	9	3	14	5	0	12	7
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(1)	1	15	13	8	10	3	7	4
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(1)	12	5	6	11	0	14	9	2
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(2)	7	11	4	1	9	12	14	2
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(2)	0	6	10	13	15	3	5	8
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(3)	2	1	14	7	4	10	8	13
	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
(3)	15	12	9	0	3	5	6	11

.

Остается применить перестановку P:

$$\begin{pmatrix} 16 & 7 & 20 & 21 \\ 29 & 12 & 28 & 17 \\ 1 & 15 & 23 & 26 \\ 5 & 18 & 31 & 10 \\ 2 & 8 & 24 & 14 \\ 32 & 27 & 3 & 9 \\ 19 & 13 & 30 & 6 \\ 22 & 11 & 4 & 25 \end{pmatrix}.$$

Итак, после сложения $L_{i-1} \oplus f(R_{i-1}, K_i)$ мы определим обе половины $L_i R_i$.

3-ий шаг. Замена частей

Полученную на 16 раунде последовательность $L_{16}R_{16}$ меняем на $R_{16}L_{16}$.

4-ый шаг. Обратная перестановка

Полагаем

$$C = (IP)^{-1}(R_{16}L_{16}),$$

что есть шифровка исходного блока M.

4. Дешифрование

Процесс дешифрования полностью совпадает с процессом шифрования, только при этом раундовые ключи используются в обратном порядке: $K_{16}, ..., K_1$. Заметим, что это те самые раундовые ключи, которые применялись при шифровании.

Итак, вначале необходимо применить начальную подстановку IP:

$$IP(C) = IP(IP^{-1}(R_{16}L_{16})) = R_{16}L_{16}.$$

Далее мы переводим правую половину (в данном случае L_{16}) налево, получая $L_{16} = R_{15}$ (как было в процессе шифрования). Для вычисления правой половины мы используем формулу $R_{16} \oplus f(L_{16}, K_{16})$, то есть добавляем операцией $\oplus$ к левой половине R_{16} значение $f(L_{16}, K_{16}) = P(S(E(L_{16}) \oplus K_{16})) = P(S(E(R_{15}) \oplus \oplus K_{16})) = R_{16} \oplus L_{15}$. Последнее равенство следует из равенства

$$R_{16} = L_{15} \oplus P(S(E(R_{15}) \oplus K_{16}))$$

простым сложением операцией $\oplus$ с L_{15}. Но тогда правая часть полученной последовательности оказывается равной

$$R_{16} \oplus (R_{16} \oplus L_{15}) = L_{15},$$

а сама последовательность приобретает вид

$$R_{15}L_{15}.$$

Итак, мы перешли за один раунд от $R_{16}L_{16}$ к $R_{15}L_{15}$. При этом вид операций P, S, E не играл никакой роли. Продолжая этот процесс, мы за 16 раундов перейдем к последовательности

$$R_0L_0,$$

а затем применим замену частей и получим последовательность

$$L_0R_0 = IP(M).$$

Применив подстановку IP^{-1}, мы получим исходный текст M.

Процесс дешифрования закончен.

5. Основные свойства DES и его криптостойкость

Существует много желательных характеристик для блочных шифров. Каждый бит шифрованного текста должен зависеть от каждого бита исходного текста (речь, конечно, идет не о полных текстах, а о соответствующих друг другу блоках). Не должно быть статистических зависимостей между исходным текстом и шифрованным текстом. Изменение одного бита исходного текста или ключа должно изменять каждый бит с вероятностью близкой к $1/2$. Эмпирически DES удовлетворяет всем перечисленным свойствам. Говорят, что DES обладает «лавинным эффектом», поскольку изменение одного бита в исходном тексте или ключе вызывает «лавину» изменений в шифрованном тексте.

При длине ключа 56 бит имеется $2^{56} \sim 7,2 \cdot 10^{16}$ всевозможных ключей. При нынешних возможностях использования параллельных вычислений на мощных процессорах такой перебор достижим в реальное время. Если процессор выполняет 10^5 операций в секунду и используется тычяча таких процессоров (чипов), то для такого перебора ключей DES понадобится примерно 110 суток. Поскольку успех может прийти где-то в середине перебора, можно говорить примерно о 50-ти сутках. Таким образом, DES, по существу использующий ключ в 56 бит, практически стойким считаться не может. В этой связи естественно рассмотреть возможность кратного применения преобразований. Рассмотрим этот вопрос более подробно.

6. Кратное использование DES

Пусть E_K обозначает произвольное преобразование DES с ключом $K \in \{0,1\}^*$, который мы с самого начала считаем 56-битным: $K \in \{0,1\}^{56}$. Пусть M обозначает множество всех бинарных блоков $m \in \{0,1\}^{64}$. Любое преобразование E_K можно считать элементом группы $\mathbf{S}_M$ всех подстановок множества M, порядок которой равен $|\mathbf{S}_M| = 2^{64}!$. Множество всех DES-преобразований

$$D = \{E_K | K \in \{0,1\}^{56}\}$$

содержит вместе с каждым элементом E_K обратный к нему, который мы обозначаем $E_{K^{-1}}$.

Обозначим через $G = \text{гр}(D)$ подгруппу группы $\mathbf{S}_M$, порожденную всеми DES-преобразованиями.

В ряде работ были найдены нижние границы порядка $|G|$ группы G. Вначале Д. Копперсмит (D. Coppersmith) [20] рассмотрел произведение $E_{10} = E_1 \circ E_0$ двух DES-преобразований с ключами, состоящими из всех единиц и всех нулей с естественными обозначениями. Он вычислил для 33 различных блоков $m_1, ..., m_{33} \in \in \{0,1\}^{64}$ наименьшие числа $k_1, ..., k_{33}$ такие, что

$$E_{10}^{k_i}(m_i) = m_i, \ i = 1, ..., 33.$$

Легко видеть, что числа k_i должны быть делителями порядка $|E_{10}|$ элемента $E_{10} \in \mathbf{S}_M$ и, значит, по теореме Лагранжа – делителями порядка $|G|$ группы G. Отсюда, в частности, следует, что порядок $|G|$ не меньше, чем нок$(k_1, ..., k_{33})$. Д. Копперсмит получил таким способом неравенство

$$|G| \geq 10^{277}.$$

Отсюда вытекает, что множество D, порядок которого 2^{56} значительно меньше, чем 10^{277}, не замкнуто относительно суперпозиции. Обычно говорят, что G не является группой. Это, конечно, так, но уже потому, что в D нет единицы. Утверждение о

незамкнутости относительно суперпозиции первоначально не было очевидным. Полученное следствие можно усилить, заметив, что если при некотором l выполняется неравенство $2^{56l} < 10^{277}$, то в принципе допустимо использовать l-кратный DES, поскольку он не сводится к $(l-1)$-кратному. Неравенство, полученное Д. Копперсмитом, было значительно усилено в работе [19], авторы которой рассмотрели еще 295 дополнительных блоков и получили еще 295 чисел $k_{34}, ..., k_{328}$. Вычислив их наименьшее общее кратное, они вывели неравенство

$$|G| \geq 1,94 \cdot 10^{2499}.$$

Конечно, на практике нет нужды брать большую кратность DES. Обычно используют так называемый тройной DES, шифруя текст M трижды:

$$C = E_{K_3}(E_{K_2}(E_{K_1}(M))).$$

Впрочем, чаще всего берут одинаковые первый и третий ключи: $K_3 = K_1$. Считается, что таким образом достигается эффект 128-битного (или около того) ключа шифрования.

Остается, правда, еще один вопрос: а почему бы не использовать двойной DES? Попробуем объяснить это, не вдаваясь в вычислительные детали.

При обычном DES поиск ключа K при известных исходном M и шифрованном C текстах заключается в переборе K и проверке равенства $E_K(M) = C$. При двойном этот процесс сводится к перебору ключей K_1, K_2 и проверке равенств $E_{K_1}(M) = D_{K_2}(C)$, что не столь существенно отличается от предыдущего перебора. Другое дело – перебор, соответствующий равенству $E_{K_2}(E_{K_1}(M)) = D_{K_1}(C)$, времени на него требуется уже существенно больше.

По-видимому, использование четверного DES уже не дает существенного преимущества по сравнению с тройным.

7. Еще раз о криптостойкости DES

Говоря об анализе DES, мы не учитываем целый ряд обстоятельств. Есть сведения, говорящие о том, что NSA и другие подобные организации построили специальные компьютеры, вычисляющие ключи DES всего за несколько часов. При этом используются накопленные статистические сведения. Возможно, что существует «тайный вход». Принимаются во внимание часто используемые методы генерации ключей и т. п. Можно утверждать, что в реальном использовании DES еще менее безопасен, чем в теоретическом плане. Впрочем, то же самое можно сказать и о любом другом методе шифрования. Но для DES эти слабости в основном объясняются малым размером ключей.

Несмотря на появление в начале XXI в. нового стандарта шифрования AES (см. приложение 3), стандарт DES продолжает использоваться, в частности, как составная часть различных протоколов.

Задачи и упражнения

Задача П1.1. Пусть $D \subseteq \mathbf{S}_M$ – подмножество элементов группы $\mathbf{S}_M$ всех подстановок некоторого конечного множества M. Для любой пары элементов $d \in D$ и $m \in M$ определим наименьшее число $k = k(d, m)$, для которого $d^k(m) = m$. Доказать, что k делит порядок $|d|$ элемента $d \in \mathbf{S}_M$.

Задача П1.2. Проверить по одной из S-таблиц следующие свойства:

1) при фиксированных крайних битах 6-битового входа центральные 4-битовые блоки находятся во взаимно-однозначном соответствии с 4-битовыми выходами;

2) если входы отличаются только одним битом, то выходы отличаются не менее чем двумя битами.

Приложение 2

Стандарты электронной подписи

1. DSS

В августе 1991 г. произошло историческое событие: Национальный институт стандартов и технологий (NIST) США принял алгоритм электронной подписи DSA (Digital Signature Algorithm), зафиксировав его в качестве стандарта DSS (Digital Signature Standard), известного также как FIPS 186 (Federal Information Processing Standard 186). Это был первый стандарт электронной подписи в мировой практике.

Алгоритм DSA основан на базовом протоколе электронной подписи Эль Гамаля, рассмотренном в лекции 16. Вместе с тем он имеет ряд существенных особенностей, отличающих его от базового протокола.

В DSS используется хэш-функция $h = SHA - 1$ (Secure Hash Algorithm – 1), переводящая произвольную бинарную последовательность в кольцо вычетов $\mathbf{Z}_q$, где q – некоторый простой параметр, участвующий в установке алгоритма. DSS производит приложенную электронную подпись любого документа m, представленного в виде бинарной последовательности: $m \in \{0,1\}^*$.

Установка

Выбирается простое число q:
$2^{159} < q < 2^{160}$.
Выбирается целое неотрицательное число t:
$0 \le t \le 8$,
и простое число p:
$2^{511+64 \cdot t} < p < 2^{512+64 \cdot t}$,
причем q делит $p - 1$: $q | p - 1$.

Выбирается порождающий элемент f единственной циклической подгруппы $C = C_q$ (порядка q) группы $\mathbf{Z}_p^*$.

Для этого берется случайным образом элемент $g_1 \in \mathbf{Z}_p^*$ и вычисляется его степень $f_1 = g_1^{p-1/q}$. Если $f_1 \neq 1$, то полагается $f = f_1$. Если $f_1 = 1$, то берется следующий случайный элемент $g_2 \in \mathbf{Z}_p^*$, и процесс повторяется. Так делается до тех пор, пока мы не получим элемент f, очевидно, имеющий порядок q.

Выбирается долгосрочный секретный ключ a – случайное число, удовлетворяющее неравенствам:

$1 \leq a \leq q - 1$.

Вычисляется

$y = f^a (\mathrm{mod} p)$.

Все перечисленное позволяет сформировать набор открытых данных (p, q, f, y) и секретный ключ a.

Замечание 73. *Пусть известен порождающий элемент g циклической группы $\mathbf{Z}_p^*$. Тогда в качестве f можно взять элемент $f = g^{p-1/q}$. Ясно, что $f^q = f^{p-1} = 1$. Значит, его порядок делит q. Так как q – простое число, оно и является этим порядком. Это рассуждение показывает существование элемента f с требуемыми свойствами. Знание порождающего элемента g в стандарте не предполагается, поэтому поиск f организован описанным выше способом.*

Заметим также, что все элементы любой циклической подгруппы C порядка q являются в поле $\mathbf{Z}_p$ корнями из 1 степени q. Но в поле не может быть более чем q корней. Значит, C состоит в точности из всех корней данной степени, следовательно, такая подгруппа единственна.

Алгоритм подписи

Выбирается сессионный секретный ключ k – случайное число, удовлетворяющее неравенствам:

$1 \leq k \leq q - 1$.

Вычисляется 1-й элемент r электронной подписи:

$r = (f^k(\mathrm{mod}p))(\mathrm{mod}q)$.

Вычисляется $l = k^{-1}(\mathrm{mod}q)$.

Вычисляется 2-й элемент s электронной подписи:

$s = l(h(m) + ar)(\mathrm{mod}q)$.

Все перечисленное позволяет сформировать электронную подпись (r, s) под документом m.

Замечание 74. *Все значения под знаками* $\mathrm{mod}p$ *и* $\mathrm{mod}q$ *имеют стандартные имена. Выражение* $(b(\mathrm{mod}p))(\mathrm{mod}q)$ *означает, что вначале берется стандартное имя* b_p *для* $b(\mathrm{mod}p)$, *а затем стандартное имя* $b_p(\mathrm{mod}q)$, *которое и есть* $(b(\mathrm{mod}p))(\mathrm{mod}q)$.

Проверка правильности подписи

Необходимо знать открытые данные (p, q, f, y), функцию h и документ m.

Проверяется выполнение неравенств:

$0 < r < q$ и $0 < s < q$;

если хотя бы одно из них не выполнено, то подпись считается неправильной (значения $r = 0$ и $s = 0$ возможны; если $r = 0$, следует изменить сессионный ключ k; если $s = 0$, следует также изменить k, что приведет к другим значениям r и s).

Вычисляется

$w = s^{-1}(\mathrm{mod}q)$ и $h(m)$.

Вычисляется

$u_1 = w \cdot h(m)(\mathrm{mod}q)$ и $u_2 = r \cdot w(\mathrm{mod}q)$.

Вычисляется

$v = (f^{u_1} y^{u_2}(\mathrm{mod}p))(\mathrm{mod}q)$.

Подпись считается правильной, если

$v = r$.

Объяснение

Если (r, s) – правильная подпись под документом m, то $h(m) = -ar + ks (\mathrm{mod} q)$.

Умножая обе части полученного сравнения на w, имеем сравнение $wh(m) + arw = k (\mathrm{mod} q)$.

Это есть $u_1 + au_2 = k (\mathrm{mod} q)$. Возводим f и y в соответствующие степени:

$(f^{u_1} y^{u_2} (\mathrm{mod} p))(\mathrm{mod} q) = (f^k (\mathrm{mod} p))(\mathrm{mod} q)$.

Следовательно, $v = r (\mathrm{mod} q)$, что и требовалось установить.

Пример 75. *Для иллюстрации мы приводим алгоритм при малых параметрах $p = 31$, $q = 5$, не удовлетворяющих ограничениям стандарта. То же самое касается функции h, мы полагаем $h(m) = 4$.*

Легко проверить, что $g = 3$ – порождающий элемент группы $\mathbf{Z}_{31}^*$, поэтому можно взять $f = g^{p-1/q} = 3^6 = 16$. Элемент f порождает в $\mathbf{Z}_{31}^*$ циклическую подгруппу $C = C_5$ порядка 5.

Возьмем в качестве долгосрочного ключа $a = 2$, а в качестве сессионного $k = 3$. Тогда $y = f^a = 16^2 = 8 (\mathrm{mod} 31)$, $l = k^{-1} = 2 (\mathrm{mod} 5)$, $r = (f^k (\mathrm{mod} p))(\mathrm{mod} q) = (16^3 (\mathrm{mod} 31))(\mathrm{mod} 5) = 4 (\mathrm{mod} 31)(\mathrm{mod} 5)$, $s = l(h(m) + ar) = 2 \cdot (4 + 2 \cdot 4) = 4 (\mathrm{mod} 5)$.

Проведем проверку:

$w = 4 (\mathrm{mod} 5)$, $h(m) = 4$, $u_1 = 1 (\mathrm{mod} 5)$, $u_2 = 4 \cdot 4 = 1 (\mathrm{mod} 5)$, $v = (16^1 \cdot 8^1 (\mathrm{mod} 31))(\mathrm{mod} 5) = (4 (\mathrm{mod} 31))(\mathrm{mod} 5)$.

Подпись правильная, так как $v = r (\mathrm{mod} 5)$: $4 = 4 (\mathrm{mod} 5)$.

Если говорить о слабых местах DSS, то, во-первых, можно повторить все сказанное по аналогичному поводу касательно электронной подписи Эль Гамаля (см. лекцию 16). Во-вторых, нам бы хотелось отметить некоторую нелогичность в построении DSS, приводящую, как нам представляется, к дополнительным слабостям алгоритма.

Замечание 76. *Основная работа идет в циклической подгруппе C группы $\mathbf{Z}_p^*$. Тем не менее, вряд ли стоит называть C платформой шифрования. Все из-за того, что используется двойной модуль $(\mathrm{mod} p)(\mathrm{mod} q)$. При вычислениях r и v его можно было бы*

избежать, в результате чего заключительное равенство выглядело бы как $v = r(\mathrm{mod}p)$. *То, что мы приводим* $\mathrm{mod}q$ *значения* s, u_1, u_2, k, l, a, *объяснимо, эти параметры фигурируют в показателях степеней элемента* f *порядка* q. *Заметим, что слагаемое* ar *в выражении для* s *можно было бы успешно заменить слагаемым* ac, *где* c – *случайное число, удовлетворяющее неравенствам* $1 \leq c \leq q-1$. *Такая замена не позволяет вычислить долгосрочный ключ* a *по сессионному ключу* k.

Например, в рассмотренном выше примере мы получим значения $y = 8(\mathrm{mod}31)$, $r = 4(\mathrm{mod}31)$, *а значения* $w, h(m), u_1, u_2, s, k, l, a$ *останутся без изменения. Заключительное равенство будет выглядеть как*

$$r = v = 16^1 \cdot 8^1 = 4(\mathrm{mod}31),$$

что свидетельствует о правильности подписи.

При таком варианте нам не нужно было бы требовать выполнения неравенства $r \neq 0(\mathrm{mod}q)$, оно было бы выполнено автоматически. Легче было бы добиваться неравенства $s = l(h(m)+ac) \neq 0(\mathrm{mod}q)$, изменяя параметр c и не трогая уже вычисленного элемента подписи r. Также можно было бы не изменять сессионный ключ k.

2. ГОСТ Р 34.10-94.

Российский стандарт электронной подписи, о котором идет речь, принят в 1994 г. Он также основан на базовом алгоритме электронной подписи Эль Гамаля и внешне похож на DSS. При внимательном рассмотрении обнаруживается ряд особенностей.

В рассматриваемом стандарте используется хэш-функция h из соответствующего стандарта ГОСТ Р 34.11–94, отображающая произвольную бинарную последовательность $m \in \{0,1\}^*$ в $\mathbf{Z}_q$. Исходный текст представим в виде бинарной последовательности.

Установка

Выбирается простое число q:
$2^{254} < q < 2^{256}$.

Выбирается простое число p в одном из следующих диапазонов:

$2^{509} < p < 2^{512}$ или $2^{1020} < p < 2^{1024}$,

причем q делит $p-1$: $q|p-1$.

Выбирается порождающий элемент f единственной циклической подгруппы $C = C_q$ порядка q группы $\mathbf{Z}_p^*$.

Для этого берется случайным образом элемент $g_1 \in \mathbf{Z}_p^*$ и вычисляется его степень $f_1 = g_1^{p-1/q}$. Если $f_1 \neq 1$, то полагается $f = f_1$. Если $f_1 = 1$, то берется следующий случайный элемент $g_2 \in \mathbf{Z}_p^*$, и процесс повторяется. Так делается до тех пор, пока мы не получим элемент f, очевидно, имеющий порядок q.

Выбирается долгосрочный секретный ключ a – случайное число, удовлетворяющее неравенствам:

$0 \leq a \leq q-1$.

Вычисляется

$y = (f^a(\mathrm{mod} p))(\mathrm{mod} q)$.

Все перечисленное позволяет сформировать набор открытых данных (p, q, f, y) и секретный ключ a.

Алгоритм подписи

Выбирается сессионный секретный ключ k, удовлетворяющий неравенствам:

$0 \leq k \leq q-1$.

Вычисляется 1-ый элемент r электронной подписи:

$r = (f^k(\mathrm{mod} p))(\mathrm{mod} q)$.

Вычисляется 2-ой элемент s электронной подписи:

$s = (kh(m) + ar)(\mathrm{mod} q)$.

Все перечисленное позволяет сформировать электронную подпись (r, s) под документом m.

Замечание 77. *Все значения под знаками* $\mathrm{mod} p$ *и* $\mathrm{mod} q$ *имеют стандартные имена, как это объяснено выше.*

Проверка правильности подписи

Проверяется выполнение неравенств:

$0 < r < q$ и $0 < s < q$;

если хотя бы одно из них не выполнено, то подпись считается неправильной (значения $r = 0$ и $s = 0$ возможны; если это случилось, то выбирается новый сессионный ключ k).

Находится

$w = h(m)^{-1}(\mathrm{mod} q)$.

Это можно сделать, если только $h(m) \neq 0(\mathrm{mod} q)$. Если все-таки $h(m) = 0(\mathrm{mod} q)$, то $h(m)$ присваивается значение 1.

Вычисляется значение

$v = (f^{ws} f^{-awr}(\mathrm{mod} p))(\mathrm{mod} q)$.

Подпись считается правильной, если

$v = r$.

Объяснение

Если (r, s) – правильная подпись, то

$v = (f^{ws} f^{-awr}(\mathrm{mod} p))(\mathrm{mod} q) =$

$= (f^{w(kh(m)+ar)-awr}(\mathrm{mod} p))(\mathrm{mod} q) =$

$= (f^{k+arw-awr}(\mathrm{mod} p))(\mathrm{mod} q) =$

$= f^{k}(\mathrm{mod} p))(\mathrm{mod} q) = r.$

Что касается слабостей приведенного алгоритма, то в нем слегка усугубились слабости, отмеченные в базовом алгоритме электронной подписи Эль Гамаля. Напомним, что там речь шла о крайних значениях ключей a, k. В данном варианте возможны даже нулевые значения, что, конечно, недопустимо. Можно повторить то, что говорилось об использовании двойных модулей $(\mathrm{mod} p)$ mod q в DSS. Но самое большое нарекание вызывает рекомендация замены $h(m) = 0(\mathrm{mod} q)$ на $h(m) = 1$. В этом случае нарушается основное требование на электронную подпись – ее зависимость от документа. Думается, что подобного варианта можно было легко избежать, чего разработчики по какой-то причине не сделали.

Задачи и упражнения

Задача П2.1. Пусть $G = C_n$ – циклическая группа конечного порядка n, q – делитель n. Доказать, что в группе G существует единственная подгруппа $H = C_q$ порядка q.

Данное свойство используется при построении платформ описанных выше стандартов электронной подписи.

Задача П2.2. Предположим, что Алиса использует в системе DSA один и тот же «случайный» сессионный ключ k для подписи двух различных документов m_1 и m_2. Пусть эти подписи (r_1, s_1) и (r_2, s_2) соответственно. Предположим, что потенциальный взломщик Оскар знает как документы m_1, m_2, так и данные подписи. Показать, как Оскар может восстановить секретный ключ a.

Задача П2.3. Анна пользуется "детской" версией DSS с простым параметром $p = 43$, порождающим элементом $f = 21$ порядка 7. Случайно она подписывает сразу два документа, используя один и тот же сессионный ключ k. Хэшированные значения $h(m_1), h(m_2)$ для этих документов равны 2 и 3, соответственно. Подписи $(2, 1)$ и $(2, 6)$, соответственно. Определить долгосрочный ключ a.

Приложение 3

Стандарт шифрования AES

1. Конкурс

В 1997 г. Национальный институт стандартов и технологии (NIST) США объявил, наконец, новый конкурс на стандарт шифрования. Предполагаемый новый стандарт AES (Advanced encryption standard) должен был заменить DES, существующий с 1974 г.

Считается, что основной слабостью DES является малая длина ключа (56 бит), иногда также говорят о неудобстве реализации DES на современных процессорах, недостаточном его быстродействии и т. п. В то же время следует отметить, что DES выдержал многолетнее испытание в самом главном – криптостойкости. За более чем 30 лет интенсивного криптоанализа этой системы не было найдено (во всяком случае не было обнародовано) эффективных методов его вскрытия, отличных от полного перебора ключа.

NIST сформулировал следующие требования к новому стандарту шифрования:

- открытость его публикации,
- использование симметричного блочного шифра, допускающего размеры ключей в 128, 192 или 256 бит,
- возможность аппаратной и программной реализации,
- незапатентованность,
- возможность качественного анализа стойкости, стоимости, гибкости, реализуемости в смарт-картах достаточно простого вида.

В последнем требовании под стойкостью понимается способность шифра противостоять различным методам криптоанализа, статистическая безопасность, стойкость к атаке методом полного перебора с учетом прогнозируемого роста вычислительных возможностей. Стоимость зависит от быстродействия, удобства программной и аппаратной реализации, достаточно низких требований к объему памяти, а также сравнительной простоты алгоритмов. Под гибкостью понимается способность вырабатывать ключи разной длины, работать в разных средах, возможность выполнения дополнительных функций (например, хэширования). Предполагалось, что новый стандарт может работать на простейших смарт-картах с объемами памяти: 256 байт RAM, 2000 байт ROM.

Фактически требовалось, чтобы новый стандарт превосходил тройной DES по всем параметрам.

В конкурсе приняли участие 15 проектов, разработанных криптографами 12 стран. В финальной стадии конкурса оказалось 5 проектов: MARS, TWOFISH, RC(6) (США), RIJNDAEL (Бельгия), SERPENT (Великобритания, Израиль, Норвегия).

Проект MARS создан в IBM, один из его разработчиков – Копперсмит (Coppersmith) – был в свое время участником разработки DES. Проект RC(6) представлен фирмой RSA-lab, одним из его авторов является Ривест (Rivest). Проект SERPENT создан Андерсоном, Бихэмом и Кнудсеном (Anderson, Biham, Knudsen). Проект TWOFISH основан на существовавшем тогда и довольно известном методе шифрования BLOWFISH, одним из авторов этих систем является известный криптограф Шнайер (Schneier).

Наконец, проект RIJNDAEL создан бельгийскими криптографами Дайменом и Риеменом (Daemen, Rijmen). Он и стал победителем конкурса и, следовательно, стандартом AES, утвержденным в 2000 г.

2. Общее описание AES

Алгоритм представляет собой блочный шифр с варьируемыми размерами ключа, блоков текстов и количеством раундов шифрования.

Один блок исходного текста записывается в виде таблицы байтов с числом строк 4 и числом столбцов 4 ($N_b = 4$), 6 ($N_b = 6$) или 8 ($N_b = 8$), соответствующим длине блока 128, 192 или 256 бит. Байты рассматриваются как элементы поля $F = \mathbf{F}_{2^8}$ порядка $2^8 = 256$.

Поле F строится как фактор-кольцо кольца многочленов $\mathbf{Z}_2[x]$ от одной переменной x с коэффициентами из простого поля $\mathbf{Z}_2$ относительно неразложимого многочлена

$$f(x) = x^8 + x^4 + x^3 + x + 1.$$

Так как $f(x)$ неразложим в $\mathbf{Z}_2[x]$ (мы не проверяем здесь это свойство, оставляя это в качестве упражнения), все многочлены степени $\leq 7 \bmod f(x)$ образуют поле F. При этом многочлену $g(x) = g_0 + g_1 x + g_2 x^2 + \ldots + g_7 x^7$ соответствует бинарная запись байта $g_7, g_6, \ldots, g_2, g_1, g_0$, то есть последовательность из 8 бит. Заметим, что байты можно не только (побитно) складывать, но и умножать, а также вычислять для ненулевых байтов обратные к ним.

Пример 78. *1) Для того чтобы найти произведение байтов* $(0,1,1,0,1,1,1,0) \cdot (1,0,0,1,1,0,1,1)$, *запишем их в виде многочленов и выполним вначале умножение в кольце* $\mathbf{Z}_2[x]$*:*

$(x^6+x^5+x^3+x^2+x)\cdot(x^7+x^4+x^3+x+1) = x^{13}+x^{12}+x^9+x^5+x.$

Затем вычислим остаток при делении полученного многочлена на $f(x)$*:*

$x^{13} + x^{12} + x^9 + x^5 + x = (x^8 + x^4 + x^3 + x + 1) \cdot (x^5 + x^4) + (x^7 + x^6 + x^5 + x^4 + x)$,

значит, произведением будет байт $(1,1,1,1,0,0,1,0)$.

2) Чтобы найти обратный к байту $(1,1,0,1,1,0,1,0)$, *запишем его в виде многочлена* $g(x) = x^7+x^6+x^4+x^3+x$ *и применим обобщенный алгоритм Эвклида:*

$x^8 + x^4 + x^3 + x + 1 = (x^7 + x^6 + x^4 + x^3 + x)(x + 1) + (x^6 + x^5 + x^4 + x^2 + 1)$,

$x^7 + x^6 + x^4 + x^3 + x = (x^6 + x^5 + x^4 + x^2 + 1)x + (x^5 + x^4)$,

$x^6+x^5+x^4+x^2+1=(x^5+x^4)(x)+(x^4+x^2+1),$
$x^5+x^4=(x^4+x^2+1)(x+1)+(x^3+x^2+x+1),$
$x^4+x^2+1=(x^3+x^2+x+1)(x+1)+(x^2),$
$x^3+x^2+x+1=(x^2)(x+1)+(x+1),$
$x^2=(x+1)(x)+x,$
$x+1=(x)(1)+1.$

Обратный ход:

$1=(x+1)-x=(x+1)-(x^2-(x+1)x)=$
$=(x+1)(x+1)-x^2=(x+1)((x^3+x^2+x+1)-x^2(x+1))-x^2=$
$=(x+1)(x^3+x^2+x+1)-x^2\cdot x^2=$
$=(x+1)(x^3+x^2+x+1)-x^2((x^4+x^2+1)-(x^3+x^2+x+1)(x+1))=$
$=(x^3+x^2+x+1)(x^3+x^2+x+1)-(x^4+x^2+1)\cdot x^2=$
$=(x^3+x^2+x+1)((x^5+x^4)-(x^4+x^2+1)(x+1))-(x^4+x^2+1)\cdot x^2=$
$=(x^3+x^2+x+1)(x^5+x^4)-(x^4+x^2+1)(x^4+x^2+1)=$
$=(x^3+x^2+x+1)(x^5+x^4)-((x^6+x^5+x^4+x^2+1)-(x^5+x^4)\cdot$
$\cdot x)(x^4+x^2+1)=$
$=(x^5+x^4)(x^5+x^2+1)-(x^6+x^5+x^4+x^2+1)(x^4+x^2+1)=$
$=((x^7+x^6+x^4+x^3+x)-(x^6+x^5+x^4+x^2+1)x)(x^5+x^2+1)-$
$-(x^6+x^5+x^4+x^2+1)(x^4+x^2+1)=$
$=(x^7+x^6+x^4+x^3+x)(x^5+x^2+1)-(x^6+x^5+x^4+x^2+1)\cdot(x^6+$
$+x^4+x^3+x^2+x+1)=$
$=(x^7+x^6+x^4+x^3+x)(x^5+x^2+1)-(f(x)-(x^7+x^6+x^4+x^3+$
$+x)(x+1))\cdot$
$\cdot(x^6+x^4+x^3+x^2+x+1)=(x^7+x^6+x^4+x^3+x)(x^7+x^6+x^2)$
$-f(x)\cdot(x^6+x^4+x^3+x^2+1),$

следовательно,

$g(x)^{-1}=(x^7+x^6+x^2)(\mathrm{mod}\, f(x)),$

что можно записать как $(1,1,0,0,0,1,0,0)$.

3. Описание алгоритма

Итак, расматривается один блок (мы берем $N_b = 4$) вида

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix}, \tag{3.1}$$

в котором a_{ij} – байты. Читается блок по столбцам сверху вниз. Далее выполняются следующие преобразования.

Замена байтов – SubBytesTransformation

Каждый байт a_{ij} заменяется вначале на a_{ij}^{-1}, если $a_{ij} \neq 0$; и остается без изменений, если $a_{ij} = 0$.

Пусть $a_{ij}^{-1} = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \ldots + \alpha_7 x^7$.
Тогда a_{ij}^{-1} заменяется на $b_{ij} = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_7 x^7$, где

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}. \tag{3.2}$$

На выходе имеем блок

$$\begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix}. \tag{3.3}$$

Циклический сдвиг 3-х последних строк – ShiftRowsTransformation

Три строки в блоке с номерами 1, 2, 3 сдвигаются циклически влево на величины $1, 2, 3$, соответственно.

В первоначальном варианте шифра Rijndael величины сдвигов при числе столбцов в блоке $N_b = 8$ отличались от приведенных выше, но в стандарте AES эта разница исчезла. Также нужно заметить, что в стандарте величины N_b и N_k обозначаются одинаково – N_k.

Перемешивание столбцов (отсутствует в последнем раунде) – MixColumnsTransformation

Столбцам ставятся в соответствие многочлены степени ≤ 3 над полем $F = \mathbf{F}_{2^8}$ по следующему правилу:

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \longleftrightarrow a_3y^3 + a_2y^2 + a_1y + a_0. \tag{3.5}$$

Рассматривается фиксированный многочлен

$$g(y) = g_3y^3 + g_2y^2 + g_1y + g_0, \tag{3.6}$$

где $g_0 = x, g_1 = 1, g_2 = 1, g_3 = x + 1 \in F$.
Многочлен вида (3.5) умножается на фиксированный многочлен (3.6) $\mathrm{mod}(y^4 + 1)$. Результат записывается как

$$b_3y^3 + b_2y^2 + b_1y + b_0 \longleftrightarrow \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}, \tag{3.7}$$

полученный столбец заменяет исходный столбец из левой части (3.5). Данная процедура проделывается с каждым столбцом массива текста.

Остается только объяснить, что указанное преобразование обратимо. Это необходимо сделать, поскольку многочлен $y^4 + 1 = (y+1)^4$ разложим. Если бы многочлен $g(y)$ имел общий множитель с $y^4 + 1$ ненулевой степени, он обладал бы корнем 1. Легко проверить, что это не так.

Наложение ключа – AddRoundKeyTransformation

Длина ключа (в байтах) должна совпадать с количеством байтов в тексте. Вычисленный раундовый ключ k_i накладывается операцией XOR на текст.

На выходе мы получаем новый текст, с которым снова проделываем те же операции, пока не пройдем все раунды шифрования. Выполняется 10 раундов при $N_b = 4$, 12 при $N_b = 6$ и 14 при $N_b = 8$.

4. Генерация раундовых ключей

Исходный ключ K так же, как и текст, записывается в виде таблицы из 4-х строк и 4-х ($N_K = 4$), 6-ти ($N_K = 6$) или 8-ми ($N_K = 8$) столбцов, в клетках которой записаны байты. Размеры выбираются независимо от выбора размеров таблиц текста.

Итак, исходный ключ K состоит из N_K 4-байтовых столбцов (слов). Далее мы расширяем K по следующим правилам.

Расширенный ключ есть последовательность 4-байтовых слов $\{W[i], i = 0, 1, 2, \ldots\}$. Первые N_K слов – это в точности исходный ключ K .

Далее при $N_K \leq 6$ полагаем:

$$W[i] = W[i-1] \oplus W[i - N_K],$$

если $i \geq N_K, i \neq l \cdot N_K$ $(l = 1, 2, 3, \ldots)$.

Для i, кратных N_K $(i = lN_K)$, полагаем

$$W[i] = F(T(W[i-1])) \oplus W[i - N_K] \oplus C(i/N_K), \qquad (4.1)$$

где T – циклический сдвиг байтовой последовательности на один байт, F обозначает операцию замены байтов, как в шифровании, $C(t)$ есть 4-байтовый вектор

$$(RC[t], 0, 0, 0),$$

первый байт которого вычисляется по закону

$$RC[t] = t \cdot RC[t-1], RC[0] = 1 (t = 1, 2, 3, \ldots). \qquad (4.2)$$

Если $N_K \geq 6$, то правила те же самые с одним отличием: если $i - 4$ кратно N_K, то

$$W[i] = F(W[i-1] \oplus W[i - N_K]). \qquad (4.3)$$

Задачи и упражнения

Задача П3.1. Доказать неприводимость многочлена $f(x) = x^8 + x^4 + x^3 + x + 1 \in \mathbf{Z}_2[x]$, по которому строится поле $\mathbf{F}_{2^8}$ в системе AES.

Задача П3.2. Пусть $g(y) \in \mathbf{Z}_2[y]$ – многочлен, определенный формулой (3.6). Указать явный вид многочлена $g'(y) \in \mathbf{Z}_2[y]$ такого, что

$$g(y)g'(y) = 1 (\mathrm{mod}\, (y^4 + 1)).$$

Другими словами, дать явный вид многочлена $g(y)^{-1} \bmod (y^4 + 1)$.

Приложение 4

Практическое использование криптографии

1. Общие сведения

В настоящее время область использования криптографических средств защиты информации уже достаточно широка. Она включает в себя Интернет, финансовые системы, мобильную телефонию и т. п. Потребность в соответствующих разработках велика. В этом направлении работает очень большое число различных организаций. Производимая ими программная и аппаратная продукция в значительной своей части не выдерживает критики. Совсем непросто бывает разобраться в том, какая продукция предоставляет удовлетворительную или даже хорошую защиту, а какая может привести к серьезным неприятностям.

Для этого есть немало объективных причин, некоторые из которых мы попытаемся осветить в данном приложении. Стоить заметить, что мы всего лишь обозначаем важную проблему практического использования криптографии. Детальный анализ даже основных проблем занял бы слишком много места.

Мы приведем краткие сведения о наиболее известном пакете PGP прикладных программ криптографической защиты, который, по-видимому, и на сегодняшний день остается наиболее используемым продуктом данного вида.

Далее мы воспользуемся рядом замечательных высказываний известного криптографа Брюса Шнайера (Bruce Schneier), которые в афористичной форме как нельзя лучше помогают разглядеть отличительные признаки недоброкачественной продукции на рынке криптографических средств защиты информации.

Наконец, мы представим несколько примеров успешного взлома достаточно известной криптографичекой продукции.

2. Пакет PGP

В начале 1990-х гг. известный программист Фил Циммерман (Philip Zimmerman) создал пакет программ для криптографической защиты файлов и сообщений от несанкционированного прочтения. Выдержав непростые испытания, в числе которых значится попытка обвинения его в нарушении закона США о зашифрованной корреспонденции 1991 г., а именно: в свободном распространении PGP через Интернет, Ф. Циммерман основал в 1996 г. компанию Pretty Good Privacy, успешно работающую и в настоящее время.

Имеется уже достаточно много версий PGP. Все они используют известные алгоритмы и протоколы шифрования, цифровой подписи, аутентификации и т. п.

Особо отметим, что PGP свободно и бесплатно распространяется через Интернет.

3. Афоризмы Брюса Шнайера о криптографической продукции

Основная неприятность, связанная с плохой системой защиты, состоит в том, что она выглядит точно так же, как и хорошая система защиты.

Б. Шнайер именует рекламу плохой криптографической защиты «Ханаанским бальзамом». Так в старину называли шарлатанские снадобья, сулящие избавление от всех недугов.

Создатели подобной продукции в значительной своей части просто недостаточно компетентны, они действительно считают, что их продукция хороша. При этом они часто пытаются реализовать действительно хорошие алгоритмы, но делают это абсолютно неумело. Есть, впрочем, много и таких «специалистов», которые просто пытаются, как говориться, «срубить денег».

Псевдоматематическое кудахтанье.

Малограмотные производители часто считают, что язык математики не имеет никакого конкретного смысла. Поэтому, подобно

знахарям-шарлатанам, они сопровождают свою рекламу бессмысленными наборами псевдоматематических терминов. Б. Шнайер приводит следующий перл: «Encryptor 4.0 использует уникальный алгоритм инкрементального сдвига базы собственной разработки». Впрочем, иногда такое «кудахтанье» носит явно антиматематический характер: «Так как длина ключа и ключевая структура изменяются и так как механизм шифрования не использует никаких математических алгоритмов, восстановление исходных данных невозможно, и подбор также невозможен».

Некоторые математики при беглом взгляде на криптографию решают, что она очень проста, и хороший алгоритм шифрования можно создать прямо из того, чем они сейчас занимаются.

Это заблуждение. Криптография представляет собой сложную область деятельности со своими терминами, методами, разработками. Продукция должна производиться в комплексе, она обязана быть достаточно простой в употреблении. Есть еще немало требований, нарушение которых является серьезным минусом.

Если разработчик скрывает используемые алгоритмы, значит, его продукция, скорее всего, недоброкачественна.

Есть немало примеров взлома такой продукции. Например, это произошло с широко разрекламированным продуктом компании GenioUSA, то же самое случилось с одной из первых версий протокола телефонии GSM и т. п.

Некоторые компании высказывают в своих заявлениях ужасающее невежество.

Например, компания TriStrata утверждала про свой алгоритм шифрования: «Так как криптографическая схема TriStrata очень проста и имеет очень низкую вычислительную сложность, клиентская часть может размещаться на самых разных системах – от сервера до ноутбука». Но дело в том, что практически все хорошие протоколы и алгоритмы шифрования требуют очень мало

ресурсов и могут уместиться на ноутбуке, быть реализованы на смарт-карте и т. д.

Предлагая ключи нелепой длины, компании также проявляют свою некомпетентность.

Иногда заводят речь о ключах даже в миллион бит. Абсурд, полный перебор 256-битовых ключей в наше время уже почти запределен. В асимметричных системах ключи, конечно, должны быть длиннее, но более 2048 бит использовать также не имеет смысла.

4. Примеры практического взлома криптографической защиты

В октябре 1998 г. в Омске злоумышленник взломал систему защиты электронных карт «Золотая корона». Эта операция была произведена с картами типа Solaic. Ошибка разработчика заключалась в том, что защищающие карту ключи могли быть извлечены из общения карты с платежными терминалами. Другой слабостью этих карт было использование 320-битового ключа системы шифрования RSA, который из-за малой длины не мог уже считаться стойким. Для справедливости отметим, что взлом произошел тогда, когда сама фирма уже переходила к использованию карт нового типа. Это свидетельствует о том, что разработчики предвидели подобное событие. Другими словами, случилось то, что должно было случиться.

На международной конференции CRYPTO, проходившей в 2003 г. в Санта-Барбаре (США), израильский криптограф Э. Бихэм со своими учениками Э. Барканом и Н. Келлером представили доклад, в котором описывалась серьезная брешь в системе защиты протокола GSM, работающего в телефонии. Они показали, что эта брешь может быть эффективно использована для проведения различных атак – от прослушивания разговоров до подделки SMS или клонирования телефонов.

Необходимо заметить, что еще раньше, в 1999 г., американский хакер М. Бриссен, известный в сети Интернет как Lucky Green,

полностью восстановил засекреченные алгоритмы криптографической защиты GSM: A3/A8 (аутентификация) и A5 (шифрование). Было показано, что слабости имеет не только экспортный вариант шифрования A5/2, но и предположительно сильный A5/1. Однако, по общему мнению, следует признать, что новые атаки, предложенные израильскими криптографами, оказались куда более опасными. Вкратце они могут быть охарактеризованы следующим образом:

– алгоритм A5/2 легко взламывается за реальное время на основе одной лишь шифровки;

– более сильные алгоритмы A5/1 и A5/3 можно вскрыть, используя активную атаку.

Интересно, что вскрытие A5/2 возможно еще до начала разговора, поскольку слабость вносит код исправления ошибок, применяемый к сигналу еще до зашифрования. Этот код вносит избыточность, что, собственно, и позволяет провести атаку.

Интересна также предложенная технология атаки на A5/1 и A5/3, в ходе которой вынуждается переход на A5/2 с теми же сеансовыми ключами, что позволяет их определить атакой первого вида, а затем уже с их помощью довершить остальное.

Описанные атаки стали возможными из-за наличия в разработках теоретических, технологических и организационных ошибок. А ведь протокол GSM в мире используют около миллиарда телефонов. Сама компания GSM была первой из тех, кто стал серьезно относиться к секретности применяемых протоколов. Она же одной из первых признала, что засекречивание алгоритмов, лежащих в основе протоколов, вовсе не является панацеей от бед, скорее, наоборот, приводит к дополнительным проблемам.

Послесловие

О чем же говорит эта книга? Я задаю вопрос прежде всего самому себе. При написании я старался дать общее представление о криптографии, сопровождая изложение описанием алгоритмов, протоколов, схем, иллюстрируя примерами. Выбирались алгоритмы, получившие широкое распространение и проверенные многочисленными приложениями.

В то же время это было всего лишь введение в мир криптографии, причем введение краткое. Естественно, что многое осталось в стороне. Я практически не касался теории сложности, даже не упомянул о булевых функциях, обошелся без формального определения односторонней функции и т. п. Много фактов приведено без доказательств. Эти жертвы принесены сознательно. Мне хотелось дать компактное представление о криптографии, не загромождая его деталями. Относительно доказательств рекомендую книгу Г.П. Агибалова [1]. Есть и другая литература – монографии и статьи по криптографии, в которой отражены различные ее аспекты, описаны приложения (см., например, [4; 6; 10]).

Могу ли я считать свою цель достигнутой? Судить, конечно, читателю. Я считаю, что, несмотря на все многообразие средств и методов, протоколов, схем и алгоритмов, современная криптография основана на небольшом круге идей. Они, как правило, просты и остроумны. Им находится ясное математическое обоснование. Должен все-таки констатировать, что теоретически современная криптография построена в основном «на песке». Если кто-то сможет доказать равенство $P = NP$, это будет означать принципиальную теоретическую невозможность построения надежных шифров. Действительно, равенство $P = NP$ на языке криптографии означает, что полиномиальная расшифровка со знанием секрета (секретного ключа) влечет существование полиномиальной расшифровки без знания этого секрета. Вряд ли, конечно, такой возможный теоретический удар поколеблет практическую криптографию. Ведь знание существования алгоритма

еще не является знанием того, как он выглядит. Кроме того, полиномы бывают разные, иные вовсе не обеспечивают реальности вычислений. Если же вдруг окажется, что справедливо ожидаемое неравенство $P \neq NP$, картина будет выглядеть оптимистично не только с практической, но и с теоретической стороны. Одним словом, криптография будет жить еще долго. Вероятно, что она будет меняться. Время покажет.

Автор

Литература

1. *Агибалов Г.П.* Избранные теоремы начального курса криптографии: учебное пособие. – Томск: Изд-во науч.-техн. лит-ры, 2005. – 113 с.

2. *Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.* Основы криптографии: учебное пособие. – М.: Гелиос АРВ, 2002. – 480 с.

3. Введение в криптографию / под общ. ред. В.В. Ященко. – М.: МЦНМО: ЧеРо, 1999. – 272 с.

4. *Иванов М.А.* Криптография. Криптографические методы защиты информации в компьютерных системах и сетях. – М.: Кудиц-образ, 2001. – 363 с.

5. *Кан Д.* Война кодов и шифров. История четырех тысячелетий криптографии. – М.: Рипол классик, 2004. – 526 с.

6. *Кнут Д.* Искусство программирования для ЭВМ. – М.: Мир, Т. 1. – 1976; Т. 2. – 1977; Т. 3. – 1978.

7. *Мао В.* Современная криптография: теория и практика. – М.: Вильямс, 2005. – 768 с.

8. *Марков А.А.* Основы алгебраической теории кос // Труды матем. ин-та АН СССР. 16 (1945).

9. *Романьков В.А.* Введение в криптографию: метод. указ. – Усть-Каменогорск: Изд-во ВКГУ, 2003. – 43 с.

10. *Н. Смарт.* Криптография. - М.: Техносфера, 2005. – 525 с.

12. *В. Столлингс.* Криптография и защита сетей. Принципы и практика. М.; СПб.; Киев: Вильямс, 2001. — 669 с.

13. *Фомичев В.М.* Дискретная математика и криптология. – М.: Диалог-МИФИ, 2003. - 397 с.

14. *Харин Ю.С., Берник В.И., Матвеев Г.В., Агиевич С.В.* Математические и компьютерные основы криптологии: учебное пособие. – Минск: Новое знание, 2003. – 382 с.

15. *Шеннон К.* Работы по теории информации и кибернетике. – М.: ИЛ, 1963. – 830 с.

16. *Agrawal M., Kayal N., Saxena N.* Primes is in P. Annals of Math., 169 N2 (2004), 781–793.

17. *Alford W.R., Granville A., Pomerance C.* There are infinitely many Carmichael numbers. Annals of Math., 140 N 3 (1994), 703–722.

16. *Anshel I., Anshel M., Goldfeld G.* An algebraic method for public-key cryptography. Math. Res. Lett., 6 N3–4 (1999), 287–291.

18. *Birman J.S.* Braids, links, and mapping class groups. Annals Math. Stud., 82 (1974).

19. *Campbell K.W., Wiener M.J.* DES is not a group. Lect. Notes in Computer Science. Advances in Cryptology – Crypto'92, New York: Springer-Verlag, 512–520.

20. *Coppersmith D.* The real reason for Rivest's phenomenon. Lect. Notes in Computer Science. Advances in Cryptology. – Crypto '85, New York: Springer-Verlag, 535–536.

21. *Diffie W., Hellman M.E.* New directions in cryptography. IEEE Transaction Information Theory, 22 N 6 (1976), 644–654.

22. *ElCamal.* A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transaction on Informatoin Theory, 1985.

23. *Koblitz N.* A course in number theory and cryptography. Graduate texts in math. 114. – New-York: Springer-Verlag, 1994.

24. *Menezes A., Okamoto T., Vanstone S.A.* Reducing elliptic curve logarithms to logarithms in a finite field. In: Proc. of the 23rd Annual ACM Symp. on Theory of Computing, 1991, New Orlean, USA, ACM 1991, STOC 1991, 80–89.

25. *Menezes A., Oorschot P.C., Vanstone S.A.* Handbook of Applied Cryptography, CRC Press, 1966.

26. *Myasnikov A., Shpilrain V., Ushakov A.* Group-based cryptography. Advanced courses in mathematics CRM Barselona. – Basel-Boston-Berlin: Birkhäuser, 2008. 183 p.

27. *Rivest R.L., Shamir A., Adleman L.* A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21 N 2 (1978), 120–126.

28. *Waterhouse W.* Abelian varieties over finite fields. Ann. Sci. École Norm. Sup., 4^e série, 2 (1969), 521-560.

Романьков Виталий Анатольевич

Введение в криптографию

Курс лекций

Редактор *Л.А. Милинская*
Корректор *О.Н. Картамышева*
Компьютерная верстка *С.О. Смерчинская*
Оформление серии *П. Родькина*

Подписано в печать 07.08.2011. Формат 60×90/16.
Гарнитура «Таймс». Усл. печ. л. 15,0. Уч.-изд. л. 15,6.
Печать офсетная. Бумага офсетная. Тираж 700 экз.
Заказ № 608

Издательство «ФОРУМ»
101990, Москва — Центр, Колпачный пер., д. 9а
Тел./факс: (495) 625-32-07, 625-52-43
E-mail: forum-knigi@mail.ru

Отдел продаж издательства «ФОРУМ»:

101990, Москва — Центр, Колпачный пер., д. 9а
Тел./факс: (495) 625-52-43
E-mail: forum-ir@mail.ru
www.forum-books.ru

Книги издательства «ФОРУМ»
вы также можете приобрести:

Отдел продаж «ИНФРА-М»
127282, Москва, ул. Полярная, д. 31в
Тел.: (495) 380-05-40 (доб. 252)
Факс: (495) 363-92-12

Отдел «Книга-почтой»
E-mail: podpiska@infra-m.ru;
books@infra-m.ru

Отпечатано с готовых диапозитивов
в ООО "Типография "ПОЛИМАГ"
127247, Москва, Дмитровское шоссе, 107