Министерство транспорта и связи Украины
Государственная администрация связи
Одесская национальная академия связи им. А. С. Попова

А. В. Онацкий, Л. Г. Йона

**Защита информации
в телекоммуникационных системах и сетях**

***Модуль 2 Криптографические методы защиты информации
в телекоммуникационных системах и сетях***

АСИММЕТРИЧНЫЕ МЕТОДЫ ШИФРОВАНИЯ

Учебное пособие
по направлениям подготовки студентов
6.050903 Телекоммуникаций
6.170102 Системы технической защиты информации
6.050901 Радиотехника

Под редакцией проф. Н. В. Захарченко

УТВЕРЖДЕНО
методическим советом
факультета ТКС
Протокол № 3
от 24.11.2009 г.

Одесса – 2010

УДК 003.26

План НМИ 2009 г.

Онацкий А. В., Йона Л. Г. Асимметричные методы шифрования. – Модуль 2 Криптографические методы защиты информации в телекоммуникационных системах и сетях: Учеб. пособие / Под ред. **Н. В. Захарченко** – Одесса: ОНАС им. А. С. Попова, 2010. – 148 с.

Изложены основные подходы и методы современной криптографии для решения задач, возникающих при обработке, хранении и передаче информации в системах телекоммуникаций.

Рассмотрены методы шифрования с открытыми ключами, цифровой подписи, основные криптографические протоколы и хэш-функции, криптосистемы на эллиптических кривых. Изложение теоретического материала ведется с использованием математического аппарата из теории чисел. Подробно описаны алгоритмы, лежащие в основе международных стандартов. Приведены упражнения, необходимые при проведении практических занятий.

ОДОБРЕНО
на заседании кафедры
информационной безопасности
и передачи данных
Протокол № 3
от 23.10.2009 г.

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Дисциплина Н3.16 – Защита информации в телекоммуникационных системах и сетях – включена в учебный план просветительно-профессиональной программы подготовки бакалавров по направления 6.050903 – Телекоммуникации.

Цель учебной дисциплины – формирование у студентов базовых знаний по проблеме защиты информационных ресурсов в системах телекоммуникаций и сетях от нарушения ее конфиденциальности, целостности и доступности.

Курс базируется преимущественно на дисциплинах: Н2.01 – Высшая математика; Н3.01 – Теория электрических цепей и сигналов; Н3.03 – Основы схемотехники; Н3.04 – Вычислительная техника и микропроцессоры; Н3.02 – Теория электрической связи; Н3.14 – Системы документальной электросвязи; Н3.11 – Телекоммуникационные и информационные сети, из которых студенты должны знать: теорию вероятности и математической статистики, элементы дискретной математики и комбинаторики, виды первичных сигналов электросвязи и их математическое описание, амплитудно-частотные и фазовые спектры, частотные характеристики электрических цепей, аналоговые и цифровые компоненты, принципы построения микропроцессорных систем, основные понятия теории информации, основные характеристики линий передачи, характеристики и общие принципы функционирования систем коммутации и систем передачи электросвязи.

Дисциплина состоит из двух модулей:

модуль 1 – Архитектура систем защиты информации: основы законодательной и нормативно-правовой базы Украины, системный подход к решению проблемы защиты информации, угрозы информации в телекоммуникационных системах, критерии информационной безопасности (лекций – 16 ч.; практических занятий – 8 ч.; лабораторных работ – 8 ч.; самостоятельная работа – 16 ч.; всего – 48 ч.);

модуль 2 – Криптографические методы защиты информации в телекоммуникационных системах и сетях: типы и классификация алгоритмов шифрования, общие принципы построения симметричных и асимметричных криптосистем, электронная цифровая подпись, решение проблем аутентификации в телекоммуникационных системах (лекций – 16 ч.; практических занятий – 8 ч.; лабораторных работ – 8 ч.; самостоятельная работа – 24 ч.; всего – 56 ч.).

Структура модуль 2

№	Тематика и содержание лекционного курса	Литература
1	Защита информации при передаче по каналам связи. Терминология. Типы и классификация алгоритмов шифрования. Шифры замены и перестановки. Принцип шифрования по методу Вермана. Условия стойкости шифров.	Л1, Л2, Л3, Л4
2	Криптосистемы с открытым ключом. Понятие односторонней функции, односторонней функции с секретом. Модель криптосистемы с открытым ключом. Криптоалгоритм Меркле–Хеллмана.	Л1, Л2, Л4, Л5
3	Система Idempotent Elements. Шифр Шамира. Стандарты асимметричного шифрования RSA, Рабина, Эль-Гамаля.	Л1, Л2, Л4, Л5
4	Стойкость RSA. Однонаправленные хэш-функции. Алгоритм цифровой подписи RSA. Принципы управления ключевой системой. Генерирование, хранение и распределение ключей. Алгоритм Диффи–Хеллмана.	Л1, Л2, Л4, Л5
5	Общие принципы построения симметричных криптосистем. Математические операции, используемые в симметричных криптосистемах Основы архитектуры современных симметричных криптосистем. Блочные алгоритмы шифрования. Шифры на основе сети Фейстеля.	Л1, Л2, Л3, Л4
6	Криптосистема DES. Схема алгоритма шифрования DES. Режимы работы алгоритма DES.	Л1, Л2, Л3, Л4
7	Стандарт шифрования ГОСТ 28147–89. Схема алгоритма шифрования ГОСТ 28147–89. Режимы работы стандарта шифрования ГОСТ 28147–89.	Л1, Л2, Л3, Л4
8	Стандарт шифрования IDEA. Реализация сети Фейстеля в стандарте шифрования IDEA.	Л1, Л2, Л3, Л4

Литература

1 **Захарченко Н. В.** и др. Развитие криптографии и ее место в современном обществе. Ч. 1. Классические методы шифрования: Учеб. пособие / Н. В. Захарченко, Л. Г. Йона, Ю. В. Щербина, А. В. Онацкий – Одесса: ОНАС им. А. С. Попова, 2003. – 95 с.

2 **Кисель В. А., Захарченко Н. В.** Основы криптографии: Учеб. пособие. – Одесса: УГАС им. А. С. Попова, 1997. – 48 с.

3 **Защита информации** в системах телекоммуникации: Учеб. пособие. Под ред. В. Л. Банкета. – Одесса: УГАС им. А. С. Попова, 1997. – 96 с.

4 **Горохов С. М., Йона Л. Г., Онацький О. В.** Сучасні криптографічні системи: Навч. посібник / Під. ред. М. В. Захарченка, – Одеса: ОНАЗ ім. О. С. Попова, 2007. – 152 с.

5 Данное учебное пособие.

Перечень практических занятий модуля 2

Темы занятий	Кол-во часов
Элементы теории чисел. Решение задач на криптоалгоритм Меркле–Хеллмана и систему Idempotent Elements	2
Решение задач на алгоритм Шамира, RSA, Рабина, Эль-Гамаля, Диффи–Хеллмана	2
Изучение криптосистемы DES. Режимы работы криптосистемы DES	2
Изучение криптосистемы ГОСТ 28147–89. Режимы работы ГОСТ 28147–89	2

Перечень лабораторных занятий модуля 2

Наименование лабораторных работ	Кол-во часов
Исследование шифра “Двойной квадрат”	2
Исследование асимметричного алгоритма шифрования RSA	2
Исследование алгоритма генерации ключа методом Диффи–Хеллмана	2
Исследование алгоритма создания цифровой подписи на основе RSA	2

Перечень знаний и умений, которые должен приобрести студент в процессе изучения материала

Уметь использовать нормативно-правовую базу Украины в области защиты информации, положения, инструкции, техническую документацию и рекомендовать мероприятия по закрытию возможных каналов утечки информации в телекоммуникационных системах и сетях.

Под руководством ведущего специалиста выполнять расчеты необходимых параметров систем технической защиты информации в системах и сетях связи.

ВВЕДЕНИЕ

Неуклонно возрастаю многообразие и сложность проблем информационной безопасности, возникающих в ходе активного развития информационных технологий. Современные решения многих проблем защиты информации немыслимы без использования криптографических методов.

Во всем многообразии проблем обеспечения информационной безопасности, решаемых при помощи криптографических методов и средств, задача обеспечения целостности и достоверности передаваемой информации представляется на сегодняшний день одной из самых острых. С учетом современных требований к информационно-телекоммуникационным системам эта задача все чаще и чаще превращается в серьезную проблему. Особенно актуальна она в финансовой сфере, поскольку для надежного функционирования электронной платежной системы необходимым условием является сохранение всех документов целостности и достоверности.

Асимметричная криптография, изобретенная и развившаяся за последние два десятилетия прошлого века, заняла за это время почти такое же положение, как и блочное симметричное шифрование. Асимметричное шифрование, или, как его еще называют, шифрование на открытом ключе представляет собой совершенно иную идеологию, уверенно занявшую свою нишу среди систем защиты информации.

Концепция криптографии с открытым ключом была предложена Уитфилдом Диффи (Whitfield Diffie)[1] и Мартином Хеллманом (Martin Hellman)[2], и, независимо, Ральфом Меркле (Ralph Merkle). Еще в 40–е годы прошлого столетия К. Шеннон[3] предложил строить шифр таким образом, чтобы задача его вскрытия была эквивалентна решению некой математической задачи, требующей объема вычислений, недоступного для современных компьютеров. Впервые идею К. Шеннона представили Диффи и Хеллман на Национальной компьютерной конференции (National Computer Conference) в Нью-Йорке 1976 году, и через несколько месяцев в печати появилась их основополагающая работа “New Directions in Cryptography” (Новые направления в криптографии). Эта работа не

---

[1] **Уитфилд Диффи** (род. в 1944 г. в США). В 1965 году окончил Массачусетский технологический институт. К началу 70–х годов, проработав в нескольких местах (заслуженный инженер компании Sun Microsystems), стал одним из нескольких полностью независимых экспертов по безопасности, свободным криптографом. Оглядываясь назад, можно сказать, что он был первым шифрпанком.

[2] **Мартин Хеллман** (род. в 1945 г. в США). В 70–е годы прошлого столетия стал профессором Стэнфордского университета в Калифорнии. В 1974 г. познакомился с У. Диффи и вместе они приступили к изучению проблемы распределения ключей, а через некоторое время к ним присоединился **Ральф Меркле**. В 1976 г. Диффи, Хеллман и Меркле произвели переворот в мире криптографии.

[3] **Клод Элвуд Шеннон** (Claude Elwood Shannon, 1916–2001) – американский математик и электротехник, один из создателей математической теории информации, в значительной мере предопределил результатами своих исследований развитие общей теории дискретных автоматов. Клода Шеннона называют “отцом теории информации”. Как говорил сам Шеннон, работа в области криптографии подтолкнула его к созданию теории информации.

только существенно изменила криптографию, но и привела к появлению и бурному развитию новых направлений в математике. Она положила начало криптографии с открытым ключом и теории криптографических протоколов.

На сегодняшний день асимметричное шифрование применяется для идентификации и аутентификации пользователей, защиты каналов передачи данных от навязывания ложных данных, защиты электронных документов от копирования и подделки.

Одно из новых направлений криптографии с открытыми ключами – системы на эллиптических кривых. Эллиптические кривые давно изучались в математике, но их использование в криптографических целях было впервые предложено Коблицем (Need Koblitz) и Миллером (Victor Miller) в 1985 году. С 1998 года использование эллиптических кривых для решения криптографических задач, таких как цифровая подпись, было закреплено в стандартах США ANSI X9.62 и FIPS 186–2, а в 2001 году стандарт ГОСТ Р 34.10–2001 был принят в России. В 2002 году в Украине принят стандарт цифровой подписи, основанной на эллиптических кривых ДСТУ 4145–2002, а в 2003 году – законы "Про електронний цифровий підпис" и "Про електронні документи та електронний документообіг".

Основное достоинство криптосистем на эллиптических кривых состоит в том, что, по сравнению с "обычными" криптосистемами, они обеспечивают существенно более высокую стойкость при равной трудоемкости или, наоборот, существенно меньшую трудоемкость при равной стойкости. Это объясняется тем, что для вычисления обратных функций на эллиптических кривых известны только алгоритмы с экспоненциальным ростом трудоемкости, тогда как для обычных систем предложены субэкспоненциальные методы. В результате уровень стойкости, который достигается, скажем, в RSA при использовании 1024-битовых модулей, в системах на эллиптических кривых реализуется при размере модуля 160 бит, что обеспечивает более простую как программную, так и аппаратную реализацию.

Криптографические алгоритмы с открытым ключом используют математический аппарат из теории чисел. Мы рассмотрим необходимый минимум из этой теории – классические теоремы Ферма, Эйлера, Уилсона, алгоритм Евклида и ряд других определений и теорем. Читатели, знакомые с теорией чисел, могут непосредственно перейти к разделу два.

1 ЭЛЕМЕНТЫ ТЕОРИИ ЧИСЕЛ

Приведенных в данном разделе сведений из теории чисел будет достаточно для описания криптографических алгоритмов с открытым ключом.

Теория чисел занимается изучением свойств целых чисел. *Целыми* называются не только числа натурального ряда 1, 2, 3, … (положительные целые), но также нуль и отрицательные целые –1, –2, –3, …

В данном пособии будет употребляться обозначение множества {..., –3, –2, –1, 0, 1, 2, 3, ...} целых чисел буквой *Z*. Множество целых чисел {1, 2, 3, ...} называется *множеством натуральных чисел* и стандартно обозначается буквой *N*.

Многие целые числа можно представить как произведение меньших чисел. Важные характеристики и отношения целых чисел могут быть получены на основе анализа их структуры с точки зрения составляющих множителей.

1.1 Делимость

Определение Целое число a есть *кратное* числа b, если $a = bm$ для некоторого целого числа m. Ненулевое целое число b делит целое число a, что обозначается как $b \mid a$, если a есть кратное b. Целое число b, которое делит целое число a, называется *делителем* числа a.

По определению, $9 \mid 27$, поскольку $27 = 9 \cdot 3$, но 5 не делит 12, так как не существует целого числа m такого, что $12 = 5 \cdot m$. Целые числа 1, 2, 3, 4, 6 и 12, и только они, являются положительными делителями числа 12. Целые числа –1, –2, –3, –4, –6 и –12 также являются делителями числа 12.

Теорема 1 (алгоритм деления) Для положительных целых чисел a и b существуют единственные неотрицательные целые числа q и r, где $0 \le r < b$ такие, что $a = bq + r$. Такие целые числа r и q называются, соответственно, *остатком* и *частным* от деления a на b.

Если $a < b$, то q должно быть равно 0, чтобы выполнялось $a = bq + r$, где $0 \le r < b$ и $q \ge 0$. Например, для $a = 4$ и $b = 7$ алгоритм деления дает $q = 0$ и $r = a = 4$, так что $4 = 7 \cdot 0 + 4$.

В силу единственности q и r, если можно получить q и r каким-нибудь другим способом, при условии $a = bq + r$, $0 \le r < b$ и $q \ge 0$, эти q и r должны совпадать с теми, которые существуют согласно теореме 1.

Поскольку любой положительный делитель положительного целого числа n не может быть больше самого этого числа, все положительные делители числа n можно найти, перебирая все целые числа от 1 до n и проверяя, не делят ли они n. Так, например, мы определили, что положительными делителями числа 12 являются числа 1, 2, 3, 4, 6 и 12. Точно так же можно показать, что положительными делителями числа 90 являются 1, 2, 3, 5, 6, 9, 10, 15, 18, 30, 45 и 90.

Проверка показывает, что числа 1, 2, 3 и 6 являются делителями как 12, так и 90. Числа 1, 2, 3 и 6 называются общими делителями чисел 12 и 90. Более того, 6 есть наибольший из этих общих делителей. Обратим внимание, что об-

щие делители 1, 2, 3 и 6 являются также делителями наибольшего общего делителя 6. В контексте наибольших общих делителей рассматриваются только положительные делители.

Определение Положительное целое число d называется *общим делителем* чисел a и b, если $d \mid a$ и $d \mid b$.

Определение Положительное целое число d называется *наибольшим общим делителем* чисел a и b, если

а) $d \mid a$ и $d \mid b$;

б) из $c \mid a$ и $c \mid b$ следует $c \mid d$.

Наибольший общий делитель чисел a и b обозначается через НОД (a, b).

Теорема 2 Если d и c – наибольшие общие делители целых чисел a и b, то $c = d$; иначе говоря, существует единственный общий делитель для целых положительных чисел a и b.

Теорема 3 Наибольший общий делитель положительных целых чисел a и b существует. Такой наибольший общий делитель может быть записан в виде

$$ua + vb$$

для некоторых целых чисел u и v. Кроме того, наибольший общий делитель – это наименьшее положительное целое число такого вида.

Теорема 4 Если $a = bq + c$, то НОД (a, b) = НОД (b, c); другими словами, каждый делитель a и b является делителем b и c и обратно.

Теорема 5 Если a, b и c – целые числа, НОД (a, b) = 1 и $a \mid bc$, то $a \mid c$.

Определение Если наибольший общий делитель a и b равен 1, то числа a и b называются *взаимно простыми*.

Например, числа 6, 10, 15 ввиду (6, 10, 15) = 1 – взаимно простые. Числа 8, 13, 21 ввиду (8, 13) = (8, 21) = (13, 21) = 1 – попарно простые. Числа 15 и 27 не являются взаимно простыми.

Непосредственно из определения следует, что если a и b – взаимно простые числа, то существуют целые числа u и v такие, что $au + bv = 1$.

Заметим, что если a и b – положительные целые числа, то ab кратно и a и b. Если рассматривать множество всех чисел, кратных a и b, то, согласно принципу полного упорядочения существует наименьшее кратное чисел a и b. Если c – наименьшее кратное чисел a и b, а d – второе кратное чисел a и b, то $c \mid d$. Иначе говоря, существуют q и r такие, что $d = qc + r$, где $r < c$. Поскольку и a и b делят числа d и c, они также делят r. Но тогда r было бы кратным a и b, которое меньше, чем c, что приводит к противоречию. Таким образом, мы приходим к следующим определениям.

Определение Положительное целое число m называется *общим кратным* целых чисел a и b, если $a \mid m$ и $b \mid m$.

Определение Положительное целое число m называется *наименьшим общим кратным* целых чисел a и b, если

а) $a \mid m$ и $b \mid m$; б) из $a \mid n$ и $b \mid n$, то $m \mid n$.

Наименьшее общее кратное чисел a и b будем обозначать НОК (a, b).

Упражнения

1 Найти положительные делители каждого из следующих чисел:

а) 54; б) 63; в) 72; г) 73; д) 74.

2 Для положительных целых чисел a и b найти неотрицательные целые числа q и r, где $0 \le r < b$ таковы, что $a = bq + r$.

а) $a = 54$, $b = 27$; б) $a = 47$, $b = 47$; в) $a = 93$, $b = 17$; г) $a = 43$, $b = 8$.

3 Для положительных целых чисел a и b найти НОД (a, b), НОК (a, b), если они определены.

а) $a = 54$, $b = 27$; б) $a = 12$, $b = 16$; в) $a = 33$, $b = 1$; г) $a = 6$, $b = 15$.

4 Доказать, что для взаимно простых чисел a и b и заданного числа n существуют целые числа x и y такие, что $ax + by = n$.

1.2 Алгоритм Евклида

Один из способов вычисления наибольшего общего делителя двух чисел – использование алгоритма Евклида[4].

Теорема 6 (алгоритм Евклида) Пусть даны два числа – a и b; $a \ge 0$, $b \ge 0$, считаем, что $a > b$. Находим ряд равенств:

$$
\begin{array}{ll}
a = b\,q_1 + r_1 & 0 \le r_1 < b \\
b = r_1\,q_2 + r_2 & 0 \le r_2 < r_1 \\
r_1 = r_2\,q_3 + r_3 & 0 \le r_3 < r_2 \\
r_2 = r_3\,q_4 + r_4 & 0 \le r_4 < r_3 \\
\cdots\cdots\cdots\cdots & \cdots\cdots\cdots \\
r_{n-3} = r_{n-2}\,q_{n-1} + r_{n-1} & 0 \le r_{n-1} < r_{n-2} \\
r_{n-2} = r_{n-1}\,q_n + r_n & 0 \le r_n < r_{n-1} \\
r_{n-1} = r_n\,q_{n+1} & r_{n+1} = 0
\end{array}
$$

заканчивающийся, когда получаем некоторое $r_{n+1} = 0$. Тогда r_n – наибольший общий делитель чисел a и b.

Последнее неизбежно, так как ряд $b, r_1, r_2, \ldots$, как ряд убывающих целых, не может содержать более чем b положительных. Имеем: $b > r_1 > r_2 > \ldots > r_n \ge 0$, следовательно процесс оборвется максимум через b шагов.

---

[4]**Евклид** (ок. 300 г. до н. э.) – древнегреческий математик. Основное сочинение Евклида называется “Начала”, которое состоит из тринадцати книг. VII–IX книги посвящены теории чисел. Алгоритм вычисления наибольшего общего делителя двух чисел изложен в IX книге.

Пример 1.1 Пусть $a = 525$, $b = 231$. Найти НОД. Применим алгоритм Евклида:

$$525 = 231 \cdot 2 + 63;$$
$$231 = 63 \cdot 3 + 42;$$
$$63 = 42 \cdot 1 + 21;$$
$$42 = 21 \cdot 2.$$

Получаем последний положительный остаток $r_3 = 21$. Таким образом, НОД (525, 231) = 21.

Пример 1.2 Пусть $a = 1234$, $b = 54$. Найти НОД.

$$1234 = 54 \cdot 22 + 46;$$
$$54 = 46 \cdot 1 + 8;$$
$$46 = 8 \cdot 5 + 6;$$
$$8 = 6 \cdot 1 + 2;$$
$$6 = 2 \cdot 3.$$

Последний ненулевой остаток равен 2, поэтому НОД (1234, 54) = 2.

С помощью алгоритма Евклида можно находить числа u и v из Z (теорема 3) таких, что $r_n = au + bv = (a, b)$. Действительно, из цепочки равенств имеем:

$$r_n = r_{n-2} - r_{n-1} q_n = r_{n-2} - (r_{n-3} - r_{n-2} q_{n-1}) q_n = \ldots = au + bv = (a, b)$$

(идем по цепочке равенств снизу вверх, выражая из каждого следующего равенства остаток и подставляя его в получившееся уже к этому моменту выражение).

Пример 1.3 Выразить НОД (85, 34) в виде $85u + 34v$.

$$85 = 34 \cdot 2 + 17;$$
$$34 = 17 \cdot 2 + 0.$$

Таким образом, НОД (85, 34) = 17 и НОД (85, 34) = $17 = 85 \cdot 1 + 34(-2)$.

Пример 1.4 Выразить НОД (252, 580) в виде $252u + 580v$.

$$580 = 252 \cdot 2 + 76;$$
$$252 = 76 \cdot 3 + 24;$$
$$76 = 24 \cdot 3 + 4;$$
$$24 = 4 \cdot 6 + 0.$$

Обратная подстановка дает

$$4 = 76 - 24 \cdot 3 = 76 - [252 - 76 \cdot 3] \cdot 3 = 76 \cdot 10 + 252 \,(-3) =$$
$$= [580 - 252 \cdot 2] \cdot 10 + 252(-3) = 580 \cdot 10 + 252 \,(-23).$$

Пример 1.5 Выразить НОД (252, 576) в виде $252u + 576v$.

$$576 = 252 \cdot 2 + 72;$$
$$252 = 72 \cdot 3 + 36.$$

После обратной подстановки получаем

$$36 = 252 - 72 \cdot 3 = 252 - [576 - 252 \cdot 2]3 = 252 \cdot 7 + 576(-3).$$

Теорема 7 Пусть заданы целые числа a и b, не равные нулю, тогда числа $a/\text{НОД}(a, b)$ и $b/\text{НОД}(a, b)$ являются взаимно простыми, т. е.

$$\text{НОД}\left(\frac{a}{\text{НОД}(a,b)}, \frac{b}{\text{НОД}(a,b)}\right) = 1.$$

Использование наибольшего общего делителя оказывается полезным при нахождении решений уравнений вида $ax + by = c$.

Определение Диофантовым уравнением первой степени называется уравнение вида $ax + by = c$ с целыми коэффициентами a, b, c решаемое на множестве целых чисел.

Теорема 8 Уравнение $ax + by = c$, где a, b и c – целые числа, имеет целочисленное решение (т. е. существуют целые числа x и y такие, что $ax + by = c$) тогда и только тогда, когда c делится на НОД (a, b). Если c делится на НОД (a, b), то решение $ax + by = c$ имеет вид

$$x_0 = \frac{uc}{\text{НОД}(a,b)}; \; y_0 = \frac{vc}{\text{НОД}(a,b)},$$

где u и v – любые решения уравнения НОД $(a, b) = au + bv$.

Пример 1.6 Найти решение уравнения $85x + 34y = 51$.

Находим НОД $(85, 34) = 17$ и $85 \cdot 1 + 34(-2) = 17$. Поэтому решение имеет вид

$$x_0 = \frac{uc}{\text{НОД}(a,b)} = \frac{1 \cdot 51}{17} = 3;$$

$$y_0 = \frac{vc}{\text{НОД}(a,b)} = \frac{(-2)51}{17} = -6.$$

Для проверки вычисляем

$$ax_0 + by_0 = 85 \cdot 3 + 34(-6) = 255 + (-204) = 51.$$

Иной способ построения решения состоит в непосредственном использовании уравнения $au + bv = \text{НОД}(a, b)$. Поскольку $u = 1$, $v = -2$, то

$$a(1) + b(-2) = 17,$$

умножаем на 3, получаем

$$a(3) + b(-6) = 51.$$

Замечаем, что при $x = 5$, $y = -11$

$$85 \cdot 5 + 34(-11) = 425 + (-374) = 51.$$

Приходим к выводу, что может существовать более одного решения.

Пример 1.7 Найти решение уравнения $252x + 580y = 20$.

Находим, НОД (252, 580) = 4 и $252(-23) + 580 \cdot 10 = 4$. Умножая каждое слагаемое на 5, получаем

$$252\ (-115) + 580 \cdot 50 = 20,$$

следовательно $x = -115$, $y = 50$ являются решением.

Теорема 9 Если a и b – ненулевые целые числа и (x_0, y_0) – решение уравнения $ax + by = c$, тогда любое другое решение (x, y) имеет вид

$$x = x_0 + \frac{b}{d}t; \quad y = y_0 - \frac{b}{d}t,$$

где t – произвольное целое число, а d = НОД (a, b).

Возвращаясь к примерам, можно записать общее решение уравнения:

$85x + 34y = 51$ имеет вид $x = 3 + 2t$ и $y = -6 - 5t$;
$252x + 580y = 20$ имеет вид $x = -115 + 145t$ и $y = 50 - 63t$.

Рассмотрим примеры использования непрерывных дробей для решения простейших диофантовых уравнений и сравнений первой степени.

Предположим, что $\frac{P_k}{Q_k}$ – последняя подходящая дробь в представлении непрерывной дробью рационального числа $\frac{a}{b}$, где НОД $(a, b) = 1$. Тогда $a = P_k$, $b = Q_k$. Известны рекуррентные соотношения

$$P_k = q_k P_{k-1} + P_{k-2};$$
$$Q_k = q_k Q_{k-1} + Q_{k-2};$$
$$P_{-1} = 1;\ P_0 = q_1;$$
$$Q_{-1} = 0;\ Q_0 = 1,$$

используя которые и находим одно решение диофантова уравнения $ax - by = 1$:

$$x_0 = (-1)^{k-1}\ Q_{k-1};\ \ y_0 = (-1)^{k-1}\ P_{k-1}.$$

Остальные решения имеют вид

$$x = (-1)^{k-1}\ Q_{k-1} + bt, \quad y = (-1)^{k-1}\ P_{k-1} + at, \quad t \in Z.$$

Пример 1.8 Решить диофантово уравнение $31x - 23y = 11$.

Поскольку 11 делится на НОД (31, 23) = 1, решение существует. Заполняем таблицу:

k	–1	0	1	2	3
q_k	–	1	2	1	7
P_k	1	1	3	4	31
Q_k	0	1	2	3	23

Значит, $k = 3$, $\frac{P_2}{Q_2} = \frac{4}{3}$. Находим решение:

$$x = (-1)^2 \cdot 11 \cdot 3 + 23t = 33 + 23t;$$
$$y = (-1)^2 \cdot 11 \cdot 4 + 31t = 44 + 31t,$$

где $t \in Z$.

Проверка:

$$31 \cdot (33 + 23t) - 23 \cdot (44 + 31t) =$$
$$= 31 \cdot 33 + 31 \cdot 23t - 23 \cdot 44 - 23 \cdot 31t = 31 \cdot 33 - 23 \cdot 44 = 11.$$

Пример 1.9 Решить диофантово уравнение $655x - 155y = 700$.

Поскольку 700 делится на НОД (655, 155) = 1, решение существует. Разделим левую и правую часть на 5, получим $131x - 23y = 140$. Заполняем таблицу:

k	–1	0	1	2	3	4
q_k	–	5	1	2	3	2
P_k	1	5	6	17	57	131
Q_k	0	1	1	3	10	23

Значит, $k = 4$, $\frac{P_3}{Q_3} = \frac{57}{10}$. Находим решение:

$$x = (-1)^3 \cdot 140 \cdot 10 + 23t = -1400 + 23t;$$
$$y = (-1)^3 \cdot 140 \cdot 57 + 131t = -7980 + 131t,$$

где $t \in Z$.

Проверка:

$$131 \cdot (-1400 + 23t) - 23 \cdot (-7980 + 131t) =$$
$$= 131 \cdot (-1400) + 131 \cdot 23t + 23 \cdot 7980 - 23 \cdot 131t =$$
$$= 131 \cdot (-1400) + 23 \cdot 7980 = 140.$$

Упражнения

1 Задан алгоритм деления $a = bq + r$. Найти q и r для указанных ниже значений a и b:

а) $a = 75$, $b = 8$; б) $a = 102$, $b = 5$; в) $a = 81$, $b = 9$; г) $a = 16$, $b = 25$.

2 Найти наибольший общий делитель для следующих пар чисел:

а) $a = 621$, $b = 437$; б) $a = 822$, $b = 436$; в) $a = 289$, $b = 377$.

3 Найти наименьшее общее кратное для пар чисел из упражнения 2.

4 Для приведенных ниже пар чисел найти u, v и d такие, что $au + bv = d$, где d – наибольший общий делитель чисел a и b:

а) $a = 83$, $b = 17$; б) $a = 361$, $b = 418$; в) $a = 216$, $b = 324$.

5 Доказать, что для целых чисел a и b, не равных нулю, $a/\text{НОД}(a,b)$ и $b/\text{НОД}(a,b)$ взаимно простые, т. е.

$$\text{НОД}\left(\frac{a}{\text{НОД}(a,b)},\ \frac{b}{\text{НОД}(a,b)}\right)=1.$$

6 Решить диофантово уравнение при помощи подходящих дробей:

а) $43x - 111y = 87$; б) $39x - 111y = 89$; в) $41x - 111y = 87$.

1.3 Простые числа

Определение Целое число, большее 1, называется *простым*, если оно не имеет положительных делителей, кроме 1 и самого себя. Положительное целое число, большее 1, называется *составным*, если оно не является простым.

В настоящее время составлены таблицы всех простых чисел, не превосходящих 50 миллионов, далее известны только отдельные их представители. В приложении А представлена таблица простых чисел, не превышающих 2200. В криптографии используют большие простые числа (512 бит и больше).

Среди первых 10-ти положительных целых чисел имеется четыре простых числа: 2, 3, 5 и 7. Целые числа $4 = 2 \cdot 2$, $6 = 2 \cdot 3$, $8 = 2 \cdot 4$, $9 = 3 \cdot 3$ и $10 = 2 \cdot 5$ являются составными. Итак, если $n = rs$, где $1 < r < n$ и $1 < s < n$, то n – составное число. По определению, целое число 1 не является ни простым, ни составным. Число 2 – единственное четное простое число. Определить, является ли небольшое целое число простым, пытаясь разделить его на меньшие простые числа, сравнительно легко, так как количество возможных вариантов невелико. Однако вопрос о том, является ли простым большое целое число, может оказаться достаточно сложным. Следующая теорема показывает, что существует бесконечно много простых чисел.

Теорема 10 Существует бесконечно много простых чисел.

Доказательство. Допустим, что существует только конечное число простых чисел, например, $p_1, p_2, ..., p_k$. Рассмотрим целое число $(p_1p_2...p_k) + 1$. Предположим, что p_r – некоторое простое число и $p_r \mid ((p_1p_2...p_k) + 1)$. Но тогда $p_r \mid (p_1p_2...p_k)$, откуда следует, что $p_r \mid 1$, а это приводит к противоречию, так как $p_r > 1$. Следовательно, $(p_1p_2...p_k) + 1$ – простое число, что, в свою очередь, также является противоречием, так как этого числа нет среди указанной конечной совокупности простых чисел. Таким образом, наше предположение о том, что существует конечное число простых чисел, ложно, поэтому простых чисел должно быть бесконечно много.

Поскольку разложение целых чисел на простые множители является важной задачей, необходимо иметь быстрый и простой способ определения, является ли данное положительное целое число простым или составным. Следующая теорема показывает, что для проверки простоты числа необходимо определить только некоторые из его возможных делителей.

Теорема 11 Если положительное целое число n является составным, то n имеет простой делитель p такой, что $p^2 \le n$.

Доказательство. Пусть p – наименьший простой делитель числа n. Если n раскладывается на множители r и s, то $p \le r$ и $p \le s$. Следовательно $p^2 \le rs = n$.

Например, чтобы определить, является ли $n = 521$ простым, необходимо рассмотреть только простые числа, которые меньше или равны 22, потому что $22^2 = 484$, а $23^2 = 529$. Простые числа, меньшие или равные 22 – это 2, 3, 5, 7, 11, 13, 17 и 19. Проверяя каждое из них, находим, что ни одно из них не делит 521. Поэтому 521 является простым числом в силу предыдущей теоремы.

Как покажет следующая теорема, простые числа образуют множество своего рода строительных блоков для целых чисел.

Теорема 12 Каждое положительное целое число либо равно 1, либо простое, либо может быть записано как произведение простых чисел.

Целое число 37 – простое. Целое число $1554985071 = 3 \cdot 3 \cdot 4463 \cdot 38713$ – произведение четырех простых чисел, два из которых совпадают.

Теорема 13 (основная теорема арифметики) Любое положительное целое число больше чем 1 либо является простым, либо может быть записано в виде произведения простых чисел, причем это произведение единственно с точностью до порядка сомножителей.

Например, $n = 39\,616\,304 = 2 \cdot 13 \cdot 7 \cdot 2 \cdot 23 \cdot 13 \cdot 2 \cdot 13 \cdot 2 \cdot 7 =$
$= 2 \cdot 2 \cdot 2 \cdot 2 \cdot 7 \cdot 7 \cdot 13 \cdot 13 \cdot 13 \cdot 23$

представляют собой два разложения на множители числа n; однако в каждом из произведений одно и то же простое число использовано одинаковое количество раз. Различается только порядок записи простых чисел. Фактически имеется 12600 различных способов разложения на множители числа n с использованием 10 простых сомножителей, однако каждое такое разложение содержит ровно четыре двойки, две семерки, три множителя равных 13, и один равный 23. Обычно простые множители группируют, используя экспоненциальную запись. Например,

$$n = 2^4 \cdot 7^2 \cdot 13^3 \cdot 23^1.$$

Теорема 14 Каждое положительное целое число, большее 1, может быть записано единственным образом с точностью до порядка в виде $q_1^{k(1)} q_1^{k(2)} \dots q_1^{k(n)}$, где $k(1), k(2) \dots k(n)$ – положительные целые числа.

Теперь понятно, почему 1 не входит во множество простых чисел. В противном случае теорема о единственности разложения на простые множители была бы неверна. Если разложение на простые множители известно, то простые числа, формирующие разложение на простые множители любого делителя этого числа, образуют подмножество соответствующего множества для делимого.

Теорема 15 Если $a = p_1^{a(1)} p_2^{a(2)} p_3^{a(3)} \dots p_k^{a(k)}$ и $b = p_1^{b(1)} p_2^{b(2)} p_3^{b(3)} \dots p_k^{b(k)}$, где p_i – простые числа, которые делят либо a, либо b, и некоторые показатели

степени могут быть равны 0. Пусть $m(i) = \min[a(i), b(i)]$ и $M(i) = \max[a(i), b(i)]$ для $0 \le i \le k$. Тогда

$$\text{НОД}\ (a, b) = p_1^{m(1)} p_2^{m(2)} p_3^{m(3)} \dots p_k^{m(k)}$$

и

$$\text{НОК}\ (a, b) = p_1^{M(1)} p_2^{M(2)} p_3^{M(3)} \dots p_k^{M(k)}.$$

Например, $a = 195\,000$ и $b = 10\,435\,750$. Разложение на простые множители чисел a и b имеет вид $a = 2^3 \cdot 3^1 \cdot 5^4 \cdot 13^1$ и $b = 2^1 \cdot 5^3 \cdot 13^3 \cdot 19^1$. Получаем

$$\begin{aligned}\text{НОД}\ (195\,000, 10\,435\,750) &= 2^{\min(3,1)}3^{\min(1,0)}5^{\min(4,3)}13^{\min(1,3)}19^{\min(0,1)} = \\ &= 2^1 \cdot 3^0 \cdot 5^3 \cdot 13^1 \cdot 19^0 = 2^1 \cdot 5^3 \cdot 13^1 = 3250;\\ \text{НОК}\ (195\,000, 10\,435\,750) &= 2^{\max(3,1)}3^{\max(1,0)}5^{\max(4,3)}13^{\max(1,3)}19^{\max(0,1)} = \\ &= 2^3 \cdot 3^1 \cdot 5^4 \cdot 13^3 \cdot 19^1 = 626\,145\,000.\end{aligned}$$

Теорема 16 Если a и b – положительные целые, то НОД $(a, b) \cdot$ НОК $(a, b) = ab$.

Пример 1.10 Найти НОК (91, 203).

Сначала определим НОД (91, 203), воспользовавшись алгоритмом Евклида, а затем разделим на него произведение чисел 91 и 203. Поскольку НОД (91, 203) = 7, находим

$$\text{ÍÎÊ}\quad (91, 203) = \frac{91 \cdot 203}{7} = 2639.$$

Упражнения

1 Разложить каждое из следующих целых чисел на простые множители:

а) 728; б) 1599; в) 4899; г) 131; д) 523.

2 Используя теоремы данного раздела для нахождения НОД и НОК следующих чисел:

а) $a = 162$, $b = 12$; б) $a = 71$, $b = 23$; в) $a = 72$, $b = 30$; г) $a = 75$, $b = 99$.

3 Если a и b – простые числа, следует ли отсюда, что $a^2 + b^2$ – простое число?

4 Два простых числа a и b называются *числами-близнецами*, если разность между ними равна 2, т. е. $a + 2 = b$. Например, 3 и 5 являются числами-близнецами. Найти три другие пары чисел-близнецов.

5 Является ли среднее арифметическое двух простых чисел-близнецов простым числом?

1.4 Метод выделения множителей Ферма

Следующая теорема является основой алгоритма разложения на простые множители, названого *методом выделения множителей Ферма*. Метод используется для определения того, является ли число простым.

Теорема 17 Целое нечетное число $n > 1$ не является простым тогда и только тогда, когда существуют неотрицательные целые числа p и q такие, что $n = p^2 - q^2$.

Очевидно, если n можно представить как разность квадратов двух неотрицательных целых чисел, скажем, $n = p^2 - q^2$, тогда $n = (p - q)(p + q)$. Поскольку $p - q > 1$, то также $p + q > 1$ и n не является простым числом.

Наоборот, если $n = rs$, где $r \geq s > 1$, тогда n можно представить как $((r + s)/2)^2 - ((r - s)/2)^2$. Поскольку n нечетное, r и s также являются нечетными, следовательно $r + s$ и $r - s$ – четные числа. Положив $p = (r + s)/2$ и $q = (r - s)/2$, находим, что p и q – целые неотрицательные числа и $p - q = s > 1$. При $n = 1$ положим $p = 1$, а $q = 0$.

Применение этого метода состоит в попытке найти целые числа p и q такие, что $n = p^2 - q^2$ или, что эквивалентно, $p^2 = n + q^2$, или $q^2 = p^2 - n$. Если используется первое уравнение, полагаем $q = 1, 2, ...$ до тех пор, пока $n + q^2$ не станет полным квадратом. Если до значения $q = (n - 1)/2$ полный квадрат не достигнут, рассмотрим ситуацию, когда $q = (n - 1)/2$, что дает $n + q^2 = ((n + 1)/2)^2$ и приводит к разложению n на множители. Поскольку q имеет вид $(r - s)/2$, где r и s – делители n, то очевидно, что q не может превысить $(n - 1)/2$. Следовательно, если до значения $q = (r - 1)/2$ полный квадрат не достигнут, число n является простым.

При использовании второго уравнения, т. е. $q^2 = p^2 - n$, берем в качестве m наименьшее целое число такое, что $m^2 > n$, и последовательно полагаем $p = m, m + 1, ...$ до тех пор, пока $p^2 - n$ не станет полным квадратом. Как и прежде, q не может превысить $(n - 1)/2$, так что если до значения $p = (n + 1)/2$ полный квадрат не получен, число n является простым. Преимущество использования второго соотношения состоит в том, что проверке на полный квадрат подлежит меньшее количество чисел.

Пример 1.11 Рассмотрим применение записи $p^2 = n + q^2$ для проверки, является ли простым число $n = 527$.

Рассмотрим $q = 1, 2, ..., (n - 1)/2$:

q	$n + q^2$
1	$527 + 1 = 528$
2	$527 + 4 = 531$
3	$527 + 9 = 536$
4	$527 + 16 = 543$
5	$527 + 25 = 552$
6	$527 + 36 = 563$
7	$527 + 49 = 576 = (24)^2$

Итак, $n = 527$ является составным и его делители легко вычисляются:

$$527 = (24)^2 - 7^2 = (24 - 7)(24 + 7) = 17 \cdot 31.$$

Упражнение

Воспользовавшись методом выделения множителей Ферма, определить, являются ли следующие числа простыми.

а) 1001; б) 1349; в) 4851; г) 1079; д) 8051; е) 7931; ж) 567.

1.5 Сравнения

Определение Пусть $a, b \in Z$, $p \in N$. Говорят, что число a сравнимо с b по модулю p, если $a - b$ при делении на p дают одинаковые остатки.

Запись выглядит так:

$$a \equiv b \pmod{p}.$$

Число a сравнимо с b по модулю p тогда и только тогда, когда $a - b$ делится на p нацело.

$$\frac{a-b}{p} = k\,.$$

Очевидно, это, в свою очередь, бывает тогда и только тогда, когда найдется такое целое число k, что

$$a = b + pk.$$

Сравнимость a с b по модулю p означает, что a и b представляют один и тот же элемент в кольце Z_p.

Понимание отношения $a \equiv b \pmod{p}$, при котором сравнимые между собой числа считаются в известном смысле равными, не отличными друг от друга, можно дать наглядную интерпретацию, представленную на рис. 1.1, используя периодичность, которая свойственна распределению сравнимых между собой чисел в натуральном ряду.

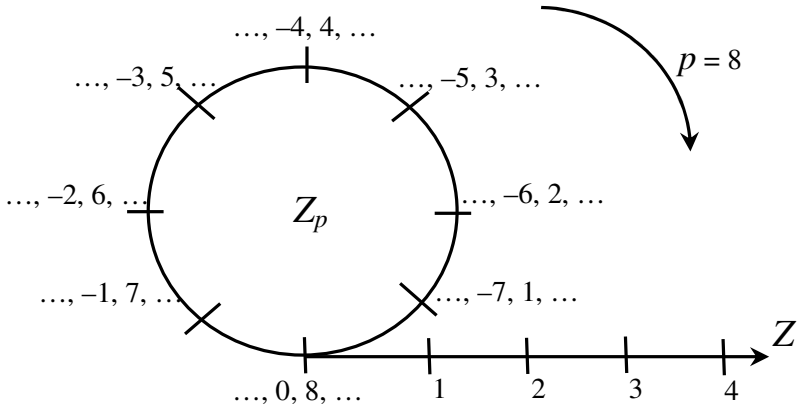

Рисунок 1.1

Пример 1.12

$17 \equiv 2 \pmod 5$, так как $17 = 2 + 5 \cdot 3$;

$-22 \equiv 1 \pmod{23}$, так как $-22 = 1 - 23 \cdot 1$;

$14 \equiv 0 \pmod 7$, так как $14 = 0 + 7 \cdot 2$;

$-5 \pmod 3 \equiv -2 \pmod 3 \equiv 1 \pmod 3 \equiv 4 \pmod 3$.

Если $p = mn$, где m и n – любые целые положительные числа, то выражение $a \equiv b \pmod{p}$ можно записать в виде $a \equiv b \pmod{m}$ и $a \equiv b \pmod{n}$.

Пример 1.13 $16 \equiv 1 \pmod{15}$, следуют $16 \equiv 1 \pmod{3}$ и $16 \equiv 1 \pmod{5}$.

Модульная арифметика во многом подобна обычной арифметике. Так, она тоже коммутативна, ассоциативна и дистрибутивна. Кроме того, приведение каждого промежуточного результата по модулю p дает такой же результат, что и выполнение всего вычисления с последующим приведением конечного результата по модулю p:

$$(a + b) \bmod p \equiv [(a \bmod p) + (b \bmod p)] \bmod p;$$
$$(a - b) \bmod p \equiv [(a \bmod p) - (b \bmod p)] \bmod p;$$
$$(ab) \bmod p \equiv [(a \bmod p)(b \bmod p)] \bmod p;$$
$$[a(b + c)] \bmod p \equiv [((ab) \bmod p) + ((ac) \bmod p)] \bmod p.$$

Из правила умножения следует правило возведения в степень:

$$a^r \equiv (a^{mn}) \bmod p \equiv (a^m \bmod p)^n \bmod p,$$

где $r = mn$.

Пример 1.14 Вычислить

$3^{50} \pmod 8 \equiv (3^2 \bmod 8)^{25} \bmod 8 \equiv (9 \bmod 8)^{25} \bmod 8 \equiv 1^{25} \pmod 8 \equiv 1$.

Определение Пусть p – положительное целое число. Множество всех классов эквивалентности по модулю p обозначается Z_p и называется *множеством классов вычетов по модулю p*.

Классы вычетов по модулю p представляют собой новые объекты. Они являются классами эквивалентности. Элементы каждого класса эквивалентности сравнимы между собой по модулю p. Например, пусть $p = 3$. Имеется три класса эквивалентности по модулю 3, так что множество

$$Z_3 = \{[0], [1], [2]\}$$

содержит три элемента. Элементы Z_3 – классы эквивалентности и, следовательно, множества. Эти три множества содержат 0, 1 и 2. В каждом из этих классов эквивалентности все элементы сравнимы между собой по модулю 3, так что $a \equiv b \pmod 3$ тогда и только тогда, когда a и b принадлежат одному и тому же классу эквивалентности – классу вычетов. Таким образом,

$$[0] = \{..., -6, -3, 0, 3, 6, 9, ...\};$$
$$[1] = \{..., -8, -5, -2, 1, 4, 7, ...\};$$
$$[2] = \{..., -7, -4, -1, 2, 5, 8, ...\}.$$

Определение Пусть Z_p – множество классов вычетов по модулю p. Для любого заданного целого числа m существует целое число r такое, что $0 \le r \le p - 1$ и $[m] = [r]$ или $m \equiv r \pmod p$. При этом говорят, что $[[m]]_p = r$.

Пример 1.15 Для $p = 5$ получаем $Z_5 = \{[0], [1], [2], [3], [4]\} = \{[r]: 0 \le r \le 4\}$.

На множестве Z_p можно определить операции сложения и умножения. Если $[a]$ – класс вычетов по модулю p, содержащий a, и $[b]$ – класс вычетов

по модулю p, содержащий b, то сложение и умножение определим соотношениями

$$[a] \oplus [b] = [a + b] = [[a + b]]_p;$$
$$[a] \otimes [b] = [a \cdot b] = [[a \cdot b]]_p,$$

где сложение и умножение в центре и справа осуществляется между целыми числами, а сложение и умножение по левую сторону выполняется между классами эквивалентности.

Пример 1.16 Для $p = 5$ получаем $Z_5 = \{[0], [1], [2], [3], [4]\}$

$[2] \oplus [4] = [2 + 4] = [6] = [1]$, поскольку $6 \equiv 1 \pmod 5$;
$[2] \otimes [4] = [2 \cdot 4] = [8] = [3]$, поскольку $8 \equiv 3 \pmod 5$.

Вычисляя суммы и произведения, для классов вычетов по модулю 5 можно создать таблицы “суммы” и “произведения”.

$[a] \oplus [b]$	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[1]	[2]	[3]	[4]
[1]	[1]	[2]	[3]	[4]	[0]
[2]	[2]	[3]	[4]	[0]	[1]
[3]	[3]	[4]	[0]	[1]	[2]
[4]	[4]	[0]	[1]	[2]	[3]

$[a] \otimes [b]$	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]	[3]	[4]
[2]	[0]	[2]	[4]	[1]	[3]
[3]	[0]	[3]	[1]	[4]	[2]
[4]	[0]	[4]	[3]	[2]	[1]

Определение Если $b \equiv r \pmod p$ для положительного целого числа p, то говорят, что r есть *вычет* числа b по модулю p.

Полная система вычетов по модулю p есть множество $S = \{r_1, r_2, ..., r_p\}$, где пересечение множества S с каждым классом вычетов по модулю p содержит одно целое число, т. е. S содержит одного и только одного представителя из каждого такого класса вычетов. Полная система вычетов $\{0, 1, 2, ..., (p - 1)\}$ называется *первичной системой вычетов*. Если b – целое число, $b \equiv r \pmod p$ и $0 \le r \le p - 1$, то этот единственный первичный вычет по модулю p обозначается через $r = [[b]]_p$. *Приведенная система вычетов* по модулю p есть подмножество полной системы вычетов, состоящее только из тех целых чисел, которые являются взаимно простыми с p, т. е. $\{r : r \in S$ и НОД $(r, p) = 1\}$.

Полная система вычетов получается путем выбора одного целого числа из каждого класса вычетов [0], [1], ..., $[p - 1]$ множества Z_p. Например, для $p = 6$ $\{24, 7, –58, 40, 113\}$ – полная система вычетов по модулю 6, так как

$$
\begin{aligned}
24 &\equiv 0 \bmod 6, \quad \text{поэтому} \quad 24 \in [0];\\
7 &\equiv 1 \bmod 6, \quad \text{поэтому} \quad 7 \in [1];\\
-58 &\equiv 2 \bmod 6, \quad \text{поэтому} \quad -58 \in [2];\\
15 &\equiv 3 \bmod 6, \quad \text{поэтому} \quad 15 \in [3];\\
40 &\equiv 4 \bmod 6, \quad \text{поэтому} \quad 40 \in [4];\\
113 &\equiv 5 \bmod 6, \quad \text{поэтому} \quad 113 \in [5].
\end{aligned}
$$

Очевидно, что $\{0, 1, 2, 3, 4, 5\}$ является полной (и притом первичной) системой вычетов по модулю 6. По определению, $[[24]]_6 = 0$ и $[[-58]]_6 = 2$.

Теорема 18 Если p – положительное целое число, $\{r_1, r_2, \ldots r_p\}$ – полная система вычетов по модулю p и a – некоторое целое число, то $a \equiv r_k \pmod p$ для одного и только одного k, $1 \le k \le p$.

Элементы полной системы вычетов $\{24, 7, -58, 15, 40, 113\}$ и первичной (полной) системы вычетов $\{0, 1, 2, 3, 4, 5\}$ по модулю 6, которые являются взаимно простыми с $p = 6$, содержат в себе, соответственно, множества $\{7, 113\}$ и $\{1, 5\}$. Поэтому оба последних множества являются приведенными системами вычетов по модулю 6. При этом говорят, что множество $\{1, 5\}$ является первичной приведенной системой вычетов по модулю 6.

Теорема 19:

а) если $a \equiv b \pmod p$ и $c \equiv d \pmod p$, то $a + c \equiv (b + d) \bmod p$ и $ac \equiv bd \pmod p$;

б) если $ac \equiv bd \pmod p$ и НОД $(c, p) = 1$, то $a \equiv b \pmod p$;

в) если $a \equiv b \pmod p$, то $a^m \equiv b^m \pmod p$ для всех целых положительных чисел m;

г) если $a \equiv b \pmod{mp}$, то $a \equiv b \pmod m$ и $a \equiv b \pmod p$;

д) для $c \ne 0$ соотношение $ac \equiv bd \pmod p$ имеет место тогда и только тогда, когда $a \equiv b\left(\bmod \dfrac{p}{\text{НОД}\,(c, p)}\right)$;

е) если $a \equiv b \pmod m$, $a \equiv b \pmod p$ и НОД $(m, p) = 1$, то $a \equiv b \pmod{mp}$.

Теорема 20 Сравнение $ax \equiv c \pmod p$ имеет решением целое число x тогда и только тогда, когда НОД $(a, p) \mid c$. Все целочисленные решения имеют вид

$$x = x_0 + \frac{tp}{\text{НОД}\,(a, p)},$$

где t – любое целое число, и для x_0 существует такое y_0, что (x_0, y_0) является решением уравнения $ax + py = c$.

Теорема 21 Если НОД $(a, p) \mid c$, то $ax \equiv c \pmod p$ имеет конечное число различных решений по модулю p. Эти решения имеют вид

$$x_0 + \frac{tp}{\text{НОД}\,(a, p)} \quad \text{по модулю} \quad p \equiv \left[\left[x_0 + \frac{tp}{\text{НОД}\,(a, p)}\right]\right]_p$$

для $t = 1, 2, 3, \ldots,$ НОД (a, p), где для x_0 существует такое y_0, что (x_0, y_0) является решением уравнения $ax + py = c$.

Пример 1.17 Решить сравнение $35x \equiv 14 \pmod{84}$.

Находим НОД (35, 84) = 7 и проверяем $7 \mid 14$, тогда сравнение имеет ровно семь различных решений по модулю 84, которые имеют вид

$$x_0 + \frac{84t}{7} = x_0 + 12t,$$

для $t = 1, 2, 3, \ldots, 7$ и (x_0, y_0) является решением $35x + 84y = 14$, которое равносильно

$$5x + 12y = 2.$$

Проверка дает в качестве решения $x_0 = -2$ и $y_0 = 1$. Семь различных решений по модулю 84 имеют следующий вид:

t	$x_0 + 12t$
1	$-2 + 12 \cdot 1 = 10$
2	$-2 + 12 \cdot 2 = 22$
3	$-2 + 12 \cdot 3 = 34$
4	$-2 + 12 \cdot 4 = 46$
5	$-2 + 12 \cdot 5 = 58$
6	$-2 + 12 \cdot 6 = 70$
7	$-2 + 12 \cdot 7 = 82$

Когда НОД $(a, p) = 1$, существует единственное решение сравнения: $ax \equiv c \pmod{p}$.

Пример 1.18 Решить сравнение $6x \equiv 7 \pmod{55}$.

Находим НОД (6, 55) = 1 и, очевидно, $1 \mid 7$. Поэтому существует только одно решение по модулю 55, которое имеет вид

$$x_0 + \frac{tp}{\text{НОД}\ (a, p)} = x_0 + \frac{1 \cdot 55}{1} = x_0 + 55,$$

где (x_0, y_0) является решением уравнения $ax + py = c$ или $6x + 55y = 7$. Для нахождения x_0 и y_0 начнем перебор с возвратом по алгоритму Евклида:

$$55 = 6 \cdot 9 + 1;$$
$$6 = 1 \cdot 6 + 0.$$

$55 \cdot 1 + 6(-9) = \text{НОД}\ (6, 55) = 1$. Умножаем каждое слагаемое на 7, получаем

$$55 \cdot 7 + 6(-63) = 7,$$

так что $x_0 = -63$ и $x = -63 + 55 = -8$, тогда $x = -8 + 55 = 47$.

Пример 1.19 Решить сравнение $623x \equiv -406 \pmod{84}$.

Число 623 больше, чем модуль сравнения 84, а –406 – отрицательно. Поскольку разыскиваются решения по модулю 84, выбираем целые числа в диапазоне 0, 1, 2, …, 83, так как они являются возможными остатками при делении на 84 и простейшими представителями классов эквивалентности, порожденных сравнимостью по модулю 84. Используя алгоритм деления, получаем

$$623 = 84 \cdot 7 + 35, \quad \text{или} \quad 623 \equiv 35 \pmod{84};$$
$$-406 = 84 \cdot (-5) + 14, \quad \text{или} \quad -406 \equiv 14 \pmod{84}.$$

Таким образом,

$$35x \equiv 14 \pmod{84}$$

равносильно исходному $623x \equiv -406 \pmod{84}$.

Решение сравнения $35x \equiv 14 \pmod{84}$ было уже найдено (см. пример 1.17).

Определение Если $ax \equiv b \pmod{p}$, то

$$x \equiv (-1)^{k-1} P_{k-1}\, b \pmod{p},$$

где $\frac{p}{a} = \frac{P_k}{Q_k}$ – k-ая подходящая дробь.

Пример 1.20 Решить сравнение $18x \equiv 11 \pmod{23}$, используя подходящие дроби.

Используем алгоритм Евклида для нахождения НОД (23, 18)

$$23 = 18 \cdot 1 + 5;$$
$$18 = 5 \cdot 3 + 3;$$
$$5 = 3 \cdot 1 + 2;$$
$$3 = 2 \cdot 1 + 1;$$
$$2 = 1 \cdot 2\ .$$

Значит $k = 5$; $q_1 = 1$; $q_2 = 3$; $q_3 = 1$; $q_4 = 1$; $q_5 = 2$.

Находим $P_1 = 1$; $P_2 = 3 \cdot 1 + 1 = 4$; $P_3 = 1 \cdot 4 + 1 = 5$; $P_4 = 1 \cdot 5 + 4 = 9$. Тогда

$$x \equiv (-1)^{5-1} \cdot 9 \cdot 11 \pmod{23} \equiv 99 \pmod{23} \equiv 7.$$

Определение Числа a и x являются *обратными* по модулю p, если

$$ax \equiv 1 \pmod{p}.$$

Например, $3 \cdot 4 \equiv 1 \pmod{11}$; $2 \cdot 8 \equiv 1 \pmod{15}$; $17 \cdot 3 \equiv 1 \pmod{25}$.

Определение Число x является *дискретным логарифмом* числа b, если выполняется равенство

$$a^x \equiv b \pmod{p}.$$

Пример 1.21 Найти x, если $2^x \equiv 6 \pmod{11}$.

Вычислим последовательность степеней 2 по модулю 11:

$2^1 \pmod{11} \equiv 2$; $2^2 \pmod{11} \equiv 4$; $2^3 \pmod{11} \equiv 8$; $2^4 \pmod{11} \equiv 5$;
$2^5 \pmod{11} \equiv 10$; $2^6 \pmod{11} \equiv 9$; $2^7 \pmod{11} \equiv 7$; $2^8 \pmod{11} \equiv 3$;
$2^9 \pmod{11} \equiv 6$; $2^{10} \pmod{11} \equiv 1$.

Далее последовательность повторяется. Таким образом, $x = 9$.

Определение Число a называется *квадратом* (квадратичным вычетом) по модулю p, если существует число x и выполняется равенство

$$x^2 \equiv a \pmod{p}.$$

Например, число 5 является квадратом по модулю 11, так как уравнение $x^2 \equiv 5 \pmod{11}$ имеет решение при $x = 4$.

Определим некоторые свойства квадратов.

1) Пусть p – простое число и $l = (p - 1)/2$, тогда, если a – квадрат по модулю p, то $a^l \equiv 1 \pmod{p}$; если a – не квадрат по модулю p, то $a^l \equiv -1 \pmod{p}$.

2) Пусть $n = pq$. Число a является квадратом по модулю n, когда a – квадрат по модулю p и a – квадрат по модулю q.

3) Пусть $n = pq$, где p и q – простые числа. Удалим из ряда чисел от 1 до n те, которые делятся на p и на q. Разделим оставшиеся $(p - 1)(q - 1)$ числа на четыре группы: квадраты по модулю n, квадраты по модулю p, квадраты по модулю q и числа, не являющиеся квадратами по указанным модулям. Числа, не являющиеся квадратами по модулю p и q, называются *псевдоквадратами*.

Пример 1.22 Пусть $p = 5$, $q = 7$, тогда $n = 35$ и $(5 - 1)(7 - 1) = 24$. Рассмотрим все числа от 1 до 35, удалим из них 5, 10, 15, 20, 25, 30, 35, 7, 14, 21, 28, которые делятся на 5 и 7. Оставшаяся часть распадается на четыре группы по шесть элементов в каждой:

квадраты по модулю 35:	1, 4, 9, 11, 16, 29;
квадраты по модулю 5:	6, 19, 24, 26, 31, 34;
квадраты по модулю 7:	2, 8, 18, 22, 23, 32;
псевдоквадраты:	3, 12, 13, 17, 27, 33.

Упражнения

1 Вычислить:

а) $(12 \cdot 30) \bmod 9$; б) $(16 \cdot 18) \bmod 17$; в) $343 \pmod{19}$; г) $2401 \pmod{19}$.

2 Вычислить:

а) $4^{10} \pmod{14}$; б) $3^{15} \pmod{10}$; в) $11^7 \pmod{13}$; г) $7^{17} \pmod{8}$.

3 Найти квадраты по модулю 5, 11, 55 и псевдоквадраты, если $p = 5$, $q = 11$.

4 Найти решение следующих сравнений:

а) $4x \equiv 3 \pmod{7}$; б) $17x \equiv 3 \pmod{15}$; в) $20x \equiv 8 \pmod{33}$;
г) $24x \equiv 6 \pmod{81}$; д) $91x \equiv 26 \pmod{169}$; ж) $23x \equiv 1 \pmod{36}$.

5 Найти x, если

а) $6^x \equiv 5 \pmod{13}$; б) $3^x \equiv 5 \pmod{16}$; в) $2^x \equiv 2 \pmod{5}$; г) $9^x \equiv 3 \pmod{23}$.

6 Доказать, что если a – целое нечетное число, то $a^2 \equiv 1 \pmod{8}$.

1.6 Символы Лежандра и Якоби

Символ Лежандра, обозначается $L(a, p)$, определен, если a – любое целое число $a \in Z$, а p – простое число, $p > 2$. Символ Лежандра равен 0, 1 или –1.

$L(a, p) = 0$, если a делится на p;
$L(a, p) = 1$, если a – квадратичный вычет по модулю p;
$L(a, p) = -1$, если a – квадратичный невычет по модулю p.

Значение $L(a, p)$ можно рассчитать по формуле

$$L(a, p) = (a^{(p-1)/2}) \bmod p.$$

Символ Якоби, обозначается $J(a, n)$, представляет собой обобщение символа Лежандра на составные модули. Символ Якоби определен для любого целого значения a и любого нечетного целого значения n:

если n – простое число, то $J(a, n) = 0$, если a делится на n;

если n – простое число, то $J(a, n) = 1$, если a – квадратичный вычет по модулю n;

если n – простое число, то $J(a, n) = -1$, если a – квадратичный невычет по модулю n;

если n – составное число, то $J(a, n) = J(a, p_1)\ldots J(a, p_m)$, где $p_1\ldots p_m$ – разложение n на простые множители.

Следующий алгоритм позволяет рекурсивно рассчитать символ Якоби:

$J(0, n) = 0$;
$J(1, n) = 1$;
$J(a \cdot b, n) = J(a, n) \cdot J(b, n)$;
$J(a, n) = J((a \bmod n), n)$;
$J(a, b_1 \cdot b_2) = J(a, b_1) \cdot J(a, b_2)$;
$J(2, n) = 1$, если $(n^2 - 1)/8$ четное и -1 – в противном случае.

Символ Якоби нельзя использовать для определения того, является ли a квадратичным вычетом по модулю n (кроме случая, когда n – простое число). Если $J(a, n) = 1$ и n – составное число, то утверждение, что a является квадратичным вычетом по модулю n, не обязательно истинно. Например:

$$J(7, 143) = J(7, 11) \cdot J(7, 13) = 1,$$

однако не существует таких целых чисел x, что $x^2 \equiv 7 \pmod{143}$.

Упражнение

Вычислить:

а) $L(7, 13)$; б) $J(2, 25)$; в) $J(7, 17)$; г) $J(6, 143)$.

1.7 Китайская теорема об остатках

Рассмотрим системы сравнений:

$$x \equiv a_1 \pmod{p_1};$$
$$x \equiv a_2 \pmod{p_2};$$
$$\vdots$$
$$x \equiv a_n \pmod{p_n};$$

где числа p_i – попарно взаимно простые. Требуется найти целое число x, которое при делении на p_i дает остаток a_i, если НОД $(p_i, p_j) = 1$ при $i \neq j$.

Еще в древние времена люди рассматривали системы сравнений и успешно их решали. Очень часто ставились задачи на устный счет наподобие следующей. Представьте, что группа обезьян пытается разложить по кучкам груду кокосовых орехов. Если обезьяны разложат орехи в кучки по пять штук, то останется четыре ореха. Если разложат в кучки по четыре, останутся три ореха. Кучки по семь дадут остаток два. Кучки по девять – остаток шесть. Каково минимально возможное количество орехов?

Если x – возможное количество орехов в кучке, тогда наличие четырех в остатке при раскладке в кучки по пять штук можно выразить как

$$x \equiv 4 \pmod{5}.$$

Аналогично, другие условия имеют вид

$$x \equiv 3 \pmod{4};$$
$$x \equiv 2 \pmod{7};$$
$$x \equiv 6 \pmod{9}.$$

Наименьшее целое положительное число x, удовлетворяющее четырем сравнениям, и является искомым решением. Решение таких задач дает следующая теорема.

Теорема 22 (китайская теорема об остатках) Пусть $p_1, p_2, \ldots, p_n$ – попарно взаимно простые числа, т. е. НОД $(p_i, p_j) = 1$ для всех i и j, меньших или равных n, где $i \neq j$. Тогда система сравнений

$$x \equiv a_1 \pmod{p_1};$$
$$x \equiv a_2 \pmod{p_2};$$
$$\vdots$$
$$x \equiv a_n \pmod{p_n},$$

имеет решение, равное целому числу $p_1 p_2 \ldots p_n$. Далее, если

$$M_j = \frac{\prod_{i=1}^{n} p_i}{p_j}$$

и z_j – решение сравнения $M_j z_j \equiv a_j \pmod{p_j}$ для каждого j, тогда решение имеет вид

$$x = \left[\left[\sum_{j=1}^{n} M_j z_j\right]\right]_{p_1 p_2 \cdots p_n}.$$

Доказательство. Пусть x определено согласно теореме. Тогда при любом k, $1 \le k \le n$,

$$x = \left[\left[\sum_{j=1}^{n} M_j z_j\right]\right]_{p_1 p_2 \cdots p_n},$$

так что

$$x \equiv \sum_{j=1}^{n} M_j z_j \left(\mathrm{mod} \prod_{i=1}^{n} p_i\right) \equiv \sum_{j=1}^{n} M_j z_j \,(\mathrm{mod}\, p_k) \equiv M_k z_k \,(\mathrm{mod}\, p_k) \equiv a_k \,(\mathrm{mod}\, p_k),$$

поэтому x удовлетворяет n сравнениям, $x \equiv a_k \,(\mathrm{mod}\, p_k)$ при $1 \le k \le n$. Если x' также удовлетворяет n сравнениям, тогда

$$x - x' \equiv 0 \,(\mathrm{mod}\, p_i) \quad \text{при} \quad 1 \le i \le n.$$

Поскольку НОД $(p_i, p_j) = 1$ при $i \ne j$, получаем

$$x \equiv x' \left(\mathrm{mod} \prod_{i=1}^{n} p_i\right).$$

Пример 1.23 Найти решение системы сравнений

$$x \equiv 1 \ (\mathrm{mod}\ 4);$$
$$x \equiv 7 \ (\mathrm{mod}\ 11).$$

Поскольку числа 4 и 11 – взаимно простые, существует целое число, а именно 10, такое, что $4 \cdot 10 \equiv 7 \ (\mathrm{mod}\ 11)$, и существует целое число 3 такое, что $11 \cdot 3 \equiv 1 \ (\mathrm{mod}\ 4)$. Следовательно, $4 \cdot 10 + 11 \cdot 3 = 73$, которое сравнимо с 29 по модулю 44, удовлетворяет обоим вышеприведенным сравнениям.

Пример 1.24 Найдем ответ на вопрос об обезьянах и орехах, решая систему сравнений

$$x \equiv 4 \ (\mathrm{mod}\ 5);$$
$$x \equiv 3 \ (\mathrm{mod}\ 4);$$
$$x \equiv 2 \ (\mathrm{mod}\ 7);$$
$$x \equiv 6 \ (\mathrm{mod}\ 9).$$

Имеем

$$M_1 = 4 \cdot 7 \cdot 9 = 252;$$
$$M_2 = 5 \cdot 7 \cdot 9 = 315;$$
$$M_3 = 5 \cdot 4 \cdot 9 = 180;$$
$$M_4 = 5 \cdot 4 \cdot 7 = 140.$$

Поскольку числа 5 и 252 – взаимно простые, существует целое число z_1 такое, что $252\, z_1 \equiv 4 \ (\mathrm{mod}\ 5)$ или $2\, z_1 \equiv 4 \ (\mathrm{mod}\ 5)$, или $z_1 \equiv 2 \ (\mathrm{mod}\ 5)$. Следовательно, z_1 может быть равно 2.

Поскольку числа 4 и 315 – взаимно простые, существует целое число z_2 такое, что $315\, z_2 \equiv 3 \pmod 4$ или $3\, z_2 \equiv 3 \pmod 4$. Следовательно, z_2 может быть равно 1.

Поскольку числа 7 и 180 – взаимно простые, существует целое число z_3 такое, что $180\, z_3 \equiv 2 \pmod 7$ или $5\, z_3 \equiv 2 \pmod 7$. Следовательно, z_3 может быть равно 6.

Поскольку числа 9 и 140 – взаимно простые, существует целое число z_4 такое, что $140\, z_4 \equiv 6 \pmod 9$ или $5\, z_4 \equiv 6 \pmod 9$. Следовательно, z_4 может быть равно 3.

Получаем

$$x \equiv (2 \cdot 252 + 1 \cdot 315 + 6 \cdot 180 + 3 \cdot 140) \mod (5 \cdot 4 \cdot 7 \cdot 9),$$

или $x \equiv 2319 \pmod{1260}$ и $x = 1059$ – наименьшее положительное целочисленное решение.

Упражнение

Решить системы сравнений:

а) $x \equiv 9 \pmod{12}$; $x \equiv 6 \pmod{25}$.

б) $x \equiv 3 \pmod 4$; $x \equiv 5 \pmod 9$.

в) $x \equiv 2 \pmod{13}$; $x \equiv 5 \pmod{21}$.

г) $x \equiv 5 \pmod 7$; $x \equiv 12 \pmod{15}$; $x \equiv 18 \pmod{22}$.

д) $x \equiv 7 \pmod{17}$; $x \equiv 9 \pmod{13}$; $x \equiv 3 \pmod{12}$.

ж) $x \equiv 5 \pmod 9$; $x \equiv 3 \pmod{11}$; $x \equiv 4 \pmod 5$.

1.8 Функция Эйлера

Пусть $p = n_1^{\alpha_1} n_2^{\alpha_2} \ldots n_k^{\alpha_k}$ – разложение на простые множители числа p. Каждый положительный делитель числа p либо равен 1, либо делится на p_i при некотором i, и каждое целое число, взаимно простое с p, не имеет ни одного из указанных чисел в качестве делителя. Некоторые свойства числа p зависят от количества целых чисел s, $1 \le s \le p$, не содержащих ни одно из n_i в качестве делителя.

Определение Пусть $\varphi(p)$ – количество положительных целых чисел, меньших p и взаимно простых с p, т. е. $\varphi(p)$ – количество приведенных вычетов по модулю p. Функция φ называется *тотиент-функцией Эйлера*[5], или *функцией Эйлера*.

[5] Рассматриваемая нами функция названа в честь **Леонарда Эйлера** (Leonard Euler, 1707–1783), перу которого принадлежит большее количество математических трудов. Творческое наследие Эйлера могло бы составить более 75 объемных томов. Ему принадлежат открытия практически во всех областях математики. Только в теории чисел – более 140 оригинальных работ, включая доказательства целого ряда малых теорем Ферма. Он считается основоположником топологии, а также целых разделов математического анализа. Престижную премию Парижской Академии наук, присуждаемую раз в два года, Эйлер получал 12 раз. Многие из ныне существующих систем математических обозначений введены Эйлером.

Например,

$$\varphi(1) = 1; \quad \varphi(5) = 4; \quad \varphi(9) = 6;$$
$$\varphi(2) = 1; \quad \varphi(6) = 2; \quad \varphi(10) = 4;$$
$$\varphi(3) = 2; \quad \varphi(7) = 6; \quad \varphi(11) = 10;$$
$$\varphi(4) = 2; \quad \varphi(8) = 4; \quad \varphi(12) = 4.$$

Любое положительное целое число p может быть выражено с помощью положительных целых чисел, не превосходящих и взаимно простых с каждым делителем числа p. Например, $6 = 2 \cdot 3$ имеет четыре делителя: 1, 2, 3 и 6.

$$\varphi(1) + \varphi(2) + \varphi(3) + \varphi(6) = 1 + 1 + 2 + 2 = 6.$$

Это свойство сформулировано в следующей теореме.

Теорема 23 (Гаусса) Если p – положительное целое число, то

$$\sum_{d|p} \varphi(d) = p,$$

где d – положительные делители числа p.

Пример 1.25 Пусть $p = 12$. Делителями 12 являются 1, 2, 3, 4, 6 и 12. Значения функции Эйлера

$$\varphi(1) + \varphi(2) + \varphi(3) + \varphi(4) + \varphi(6) + \varphi(12) = 1 + 1 + 2 + 2 + 2 + 4 = 12.$$

Для иллюстрации приведенной выше теоремы в частном случае $d = 1, 2, 3, 4, 6$ и 12 находим, что соответствующие значения n/d равны, соответственно $n/d = 12, 6, 4, 3, 2$ и 1, так что две упомянутые суммы равны.

Теперь перейдем к способам вычисления $\varphi(p)$ для любого целого положительного числа p. Решить указанную задачу помогут три следующие теоремы.

Теорема 24 Если числа m и n – взаимно простые, то

$$\varphi(mn) = \varphi(m)\,\varphi(n).$$

Например, пусть $m = 8$ и $n = 15$. Тогда $\varphi(8) = 4$, поскольку только 1, 3, 5 и 7 – положительные целые числа, которые меньше 8 и взаимно простые с 8. Также $\varphi(15) = 8$, поскольку только 1, 2, 4, 7, 8, 11, 13 и 14 – положительные целые числа, которые меньше 15 и взаимно простые с 15. Следовательно,

$$\varphi(120) = \varphi(8)\,\varphi(15) = 32,$$

что можно проверить непосредственно. В соответствии с утверждением теоремы 24, говорят, что φ – мультипликативна относительно взаимно простых множителей.

Теперь покажем, как вычислять $\varphi(p)$, когда p представляет собой степень единственного простого числа.

Теорема 25 Если p – простое число, то $\varphi(p^k) = p^k - p^{k-1}$.

Пример 1.26 Вычислить функцию Эйлера

$$\varphi(49) = \varphi(7^2) = 7^2 - 7^{2-1} = 49 - 7 = 42.$$

Следствие. Целое положительное число p является простым тогда и только тогда, когда $\varphi(p) = p - 1$.

Теорема 26 Если p – целое положительное число с разложением на простые множители вида

$$p = n_1^{\alpha_1} n_2^{\alpha_2} \ldots n_t^{\alpha_t},$$

то

$$\varphi(p) = \prod_{i=1}^{t} \left[n_i^{\alpha_{i-1}} (n_i - 1) \right] = p \prod_{i=1}^{t} \left(1 - \frac{1}{n_i} \right),$$

где n_i – все простые числа, являющиеся делителями числа p.

Пример 1.27 Вычислить функции Эйлера

$$\varphi(5) = 5 - 1 = 4;$$
$$\varphi(30) = \varphi(2 \cdot 3 \cdot 5) = (2-1)(3-1)(5-1) = 8;$$
$$\varphi(40) = 40\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{5}\right) = 40\left(\frac{1}{2}\right)\left(\frac{4}{5}\right) = 16;$$
$$\varphi(60) = 60\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{5}\right) = 60\left(\frac{1}{2}\right)\left(\frac{2}{3}\right)\left(\frac{4}{5}\right) = 16;$$
$$\varphi(405) = \varphi(81)\,\varphi(5) = 54 \cdot 4 = 216.$$

$p = 39\,616\,304 = 2^4 \cdot 7^2 \cdot 13^3 \cdot 23^1$, тогда

$$\varphi(39\,616\,304) = 2^3 (2-1)\, 7^1 (7-1) \cdot 13^2 (13-1) \cdot 23^0 (23-1) =$$
$$= 8 \cdot 1 \cdot 7 \cdot 6 \cdot 169 \cdot 12 \cdot 1 \cdot 22 = 14\,990\,976.$$

Теорема 27 Если целое число p больше 2, то $\varphi(p)$ – четное.

Теорема 28 (Уилсона) Целое положительное число p является простым тогда и только тогда, когда $(p-1)! \equiv -1 \pmod{p}$.

Например, пусть $p = 5$. Тогда $(5-1)! = 4! = 24 \equiv -1 \pmod 5$. Заметим, что в теореме говорится о том, что произведение $(p-1)!$ не может быть сравнимо с -1, если число p не является простым. Посредством теоремы можно проверять простоту числа p, устанавливая, справедливо ли сравнение $(p-1)! \equiv -1 \pmod{p}$. Однако такой критерий не используется для больших значений p, так как вычисление $(p-1)! \pmod{p}$ практически нецелесообразно.

Определение Пусть $ax \equiv b \bmod p$, где $p > 1$; $(a, p) = 1$. Тогда

$$x \equiv (ba^{\varphi(p)-1}) \bmod p.$$

Пример 1.28 Решить сравнение $6x \equiv 7 \pmod{55}$.
Находим НОД (6, 55) = 1. Тогда

$$x \equiv (7 \cdot 6^{\varphi(55)-1}) \bmod 55.$$

Находим $\varphi(55) = \varphi(11 \cdot 5) = (11-1)(5-1) = 40$. Тогда

$$x \equiv \left(7 \cdot 6^{39}\right) \bmod 55 \equiv 7 \cdot \left(6^{32} \cdot 6^4 \cdot 6^2 \cdot 6\right) \bmod 55 \equiv \left(7 \cdot 36 \cdot 31 \cdot 36 \cdot 6\right) \bmod 55 \equiv 47.$$

Упражнения

1 Вычислить значение функции Эйлера $\varphi(p)$ для $p = 46, 96, 1823, 2025, 2231$.

2 Решить сравнение $23x \equiv 1 \pmod{36}$.

3 Доказать, что, если число p – простое и $p > 2$, то $(p-2)! \equiv 1 \pmod{p}$.

4 Построить таблицу значений $\varphi(p)$ при $13 \le p \le 50$.

1.9 Порядок целого числа

Теорема 29 (Эйлера) Если p – целое положительное число и НОД $(a, p) = 1$, то $a^{\varphi(p)} = 1 \pmod{p}$.

Целое число a называют *первообразным корнем* по модулю p. Например, при $a = 3$, $p = 4$ имеем $\varphi(4) = 2$, так что $3^2 = 9 \equiv 1 \pmod 4$.

Если в теореме 28 p – простое число, то каждое целое положительное число, которое меньше p, является взаимно простым с p, так что $\varphi(p) = p - 1$. Таким образом, как частный случай, справедлива следующая теорема.

Теорема 30 (малая теорема Ферма) Если p – простое число, то для каждого такого целого числа a, $0 < a < p$, имеем $a^{p-1} \equiv 1 \pmod{p}$.

Например, если $p = 7$, то $p - 1 = 6$. В таком случае шестая степень каждого целого положительного числа, которое меньше $p = 7$, должна быть сравнима с 1 по модулю 7:

$$1^6 = 1 \equiv 1 \pmod 7;$$
$$2^6 = 64 \equiv 1 \pmod 7;$$
$$3^6 = 729 \equiv 1 \pmod 7;$$
$$4^6 = 4096 \equiv 1 \pmod 7;$$
$$5^6 = 15625 \equiv 1 \pmod 7;$$
$$6^6 = 46656 \equiv 1 \pmod 7;$$
$$7^6 = 117649 \equiv 1 \pmod 7.$$

Утверждение, обратное малой теореме Ферма, неверно. Например, $3^{90} \equiv 1 \pmod{91}$, однако $91 = 7 \cdot 13$ – составное число. С другой стороны, если p – целое положительное число и $0 < a < p$ таково, что $a^{p-1} \not\equiv 1 \pmod{p}$, тогда p не может быть простым. Таким образом, малая теорема Ферма содержит частичный критерий простоты числа, поскольку с ее помощью можно показать, что целое положительное число не является простым без определения нетривиального делителя числа p. Составные положительные числа n таковы, что $a^{n-1} \equiv 1 \pmod{n}$ для некоторого a, $1 < a < n$, до определенной степени схожи с простыми числами; по этой причине такого рода составное число n называют *псевдопростым*

числом по основанию a. Таким образом, число $n = 91$ – псевдопростое по основанию $a = 3$. Однако, если выбрать $a = 2$, то получим $2^{90} \not\equiv 1 \pmod{91}$; т. е. число $n = 91$ не является псевдопростым по основанию 2. Итак, 91 – псевдопростое число по основанию 3, но не является псевдопростым по основанию 2.

Теорема 31 Если p и q – простые числа, причем $p \neq q$ и k – произвольное целое число, то

$$a^{k\varphi(pq)+1} \bmod (pq) \equiv a .$$

Пример 1.29 Возьмем $p = 5$, $q = 7$. Тогда $pq = 35$, а функция Эйлера – $\varphi(35) = 4 \cdot 6 = 24$. Рассмотрим случай $k = 2$, т. е. будем возводить числа в степень $2 \cdot 24 + 1 = 49$. Получим $9^{49} \bmod 35 \equiv 9$, $23^{49} \bmod 35 \equiv 23$.

Утверждение. Пусть p и q – два различных простых числа ($p \neq q$). Тогда

$$\varphi(pq) = (p - 1)(q - 1).$$

Доказательство. В ряду $1, 2, ..., pq - 1$ невзаимно простыми с pq будут числа

$$p, 2p, 3p, ..., (q - 1)p$$

и

$$q, 2q, 3q, ..., (p - 1)q.$$

Всего таких чисел будет $(p - 1) + (q - 1)$. Следовательно, количество чисел, взаимно простых с pq, будет $pq - 1 - (p - 1) - (q - 1) = pq - q - p + 1 = (p - 1)(q - 1)$.

Определение Пусть p – целое положительное число и a – целое число такое, что НОД $(a, p) = 1$. *Порядком* числа a по модулю p называется наименьшее целое положительное число k такое, что $a^k \equiv 1 \pmod{p}$. Это число обозначается через $\operatorname{ord}_n a$.

Теорема 32 Пусть p – целое положительное число, НОД$(a, p) = 1$ и $k = \operatorname{ord}_p a$. Тогда

а) если $a^m \equiv 1 \pmod{p}$, где m – целое положительное число, то $k \mid m$;

б) $k \mid \varphi(p)$;

в) для целых r и s, $a^r \equiv a^s \pmod{p}$ тогда и только тогда, когда $r \equiv s \pmod{k}$;

г) никакие два из целых чисел $a, a^2, a^3, ..., a^k$ не являются сравнимыми по модулю k;

д) если m – целое положительное число, то порядок числа a^m по модулю n равен $\dfrac{k}{\text{НОД}(k,m)}$;

е) порядок числа a^m по модулю p равен k тогда и только тогда, когда числа m и k – взаимно простые.

Пример 1.30 Пусть $p = 14 = 2 \cdot 7$, так что $\varphi(14) = (2 - 1)(7 - 1) = 6$. Первичная приведенная система вычетов для $p = 14$ есть множество $\{1, 3, 5, 9, 11, 13\}$. Рассмотрим приведенную ниже таблицу вычетов для степеней числа $a = 5$,

m	$[[a^m]]_p$	m	$[[a^m]]_p$
1	5	8	11
2	11	9	13
3	13	10	9
4	9	11	3
5	3	12	1
6	1	13	5
7	5	–	–

из которой видно, что после $m = 6$ идет повторение одной и той же схемы. Таким образом, $k = \text{ord}_{14} 5 = 6$. Для $m = 12$ $a^m = 5^{12} \equiv 1 \pmod{14}$ и $k \mid m$, что согласуется с теоремой 32 (а). Также $\text{ord}_{14} 5 \mid \varphi(14)$, поскольку $6 \mid 6$ [теорема 32 (б)]. Кроме того, $2 \equiv 8 \equiv 14 \equiv 2 \pmod 6$ и $5^2 \equiv 5^8 \equiv 5^{14} \equiv 11 \pmod{14}$ [теорема 32 (в)]. Согласно таблице, никакие два из чисел $5^1, 5^2, 5^3, 5^4, 5^5$ и 5^6 не являются сравнимыми по модулю 14. Поскольку $\text{ord}_p b \mid \varphi(p)$ для любого целого числа b и $\varphi(p) = 6$ для $p = 14$, порядок каждого b в $\{1, 3, 5, 9, 11, 13\}$ можно легко подсчитать, как это было сделано для $a = 5$. Порядок числа b по модулю 14 имеют вид

b	$\text{ord}_p b$
1	1
3	6
5	6
9	3
11	3
13	2

Если $m = 4$, то $5^4 \equiv 9 \pmod{14}$, но $\text{ord}_{14} 5 / \text{НОД}(\text{ord}_{14} 5, 4) = 6 / \text{НОД}(6, 4) = 3$. Согласно таблице порядков, $\text{ord}_{14} 5^4 = 3$ [теорема 32 (д)].

Только $b = 3$ и $b = 5$ имеют порядок 6 по модулю 14. Показателями степени m в таблице, приведенной выше, определяющими значение a^m, которое сравнимо либо с числом 3, либо с числом 5, являются $m = 1, 5, 7, 11$ и 13. Это только такие значения m, которые являются взаимно простыми с числом $p = 14$ [теорема 32(е)].

Теорема 33 Если НОД $(a, p) =$ НОД $(b, p) = 1$ и $\text{ord}_p a$ является взаимно простым с $\text{ord}_p b$, то $\text{ord}_p (ab) = (\text{ord}_p a)(\text{ord}_p b)$.

Пример 1.31 Если $p = 11$, то все приведенные вычеты являются взаимно простыми с p. Таблица порядков по модулю 11 имеет вид

Вычет	Порядок	Вычет	Порядок
1	1	6	10
2	10	7	10
3	5	8	10
4	5	9	5
5	5	10	2

Если $a = 3$ и $b = 10$, то $ab = 30 \equiv 8 \pmod{11}$. Таким образом, $\text{ord}_{11} (ab) = \text{ord}_{11} 30 = \text{ord}_{11} 8 = 10 = (\text{ord}_{11} 3)(\text{ord}_{11} 10)$, т. е. НОД (3, 11) = НОД (10, 11) = 1,

$\operatorname{ord}_{11} 3 = 5$ и $\operatorname{ord}_{11} 10 = 2$ – взаимно простые. Заметим, что, если $a = 3$ и $c = 7$, $\operatorname{ord}_{11} 3 = 5$ не является взаимно простым с $\operatorname{ord}_{11} 7 = 10$. В этом случае $\operatorname{ord}_{11} (ac) = \operatorname{ord}_{11} 21 = \operatorname{ord}_{11} 10 = 2 \neq (\operatorname{ord}_{11} 3)(\operatorname{ord}_{11} 7) = 50$.

Результаты, полученные в теремах 32 и 33, приводят к формулировке критерия, названного критерием простоты числа Лукаса.

Теорема 34 (Лукаса) Если p – целое положительное число и существует такое целое число a, что

$$a^{p-1} \equiv 1 \pmod{p}$$

и

$$a^{\frac{p-1}{n}} \not\equiv 1 \pmod{p}$$

для каждого простого числа n, которое делит $p - 1$, тогда p – простое число.

Для того чтобы использовать критерий Лукаса для проверки p, необходимо уметь раскладывать на множители число $p - 1$, что само по себе может представлять трудности. Более того, требуется находить соответствующее a. Целое число a, введенное в теореме 34, называется *примитивным корнем* числа p. Используя критерий с числом $a = 7$, можно показать, что *число Мерсенна*[6] $p = 2^{31} - 1$ является простым, поскольку $p - 1 = 2 \cdot 3^2 \cdot 7 \cdot 11 \cdot 31 \cdot 151 \cdot 331$.

Если НОД $(a, p) = 1$ и число p – простое, теорема Ферма утверждает, что $a^{p-1} \equiv 1 \pmod{p}$. Ее обобщение, теорема Эйлера, для любого положительного числа p дает $a^{\varphi(p)} = 1 \pmod{p}$. Исключая эти и некоторые другие случаи, вычисление a^e по модулю p или, более точно, вычисление $[[a^e]]_p$, т. е. остатка от деления a^e на p, для большого значения e может представлять значительные трудности, поскольку само вычисление a^e в таких случаях и деление его на p практически нецелесообразно. Найти значение $[[a^e]]_p$ можно следующим образом. Для $e = [b_m\, b_{m-1} \ldots b_1\, b_0]$ (двоичная запись числа e) начинаем с

$$p_m = [[a]]_p.$$

Затем для $k = m - 1, m - 2, \ldots, 2, 1$ и 0 вычисляем

$$p_k = \begin{cases} \left[\left[p_{k+1}^2 \right]\right]_p, & \text{если} \quad b_k = 0; \\ \left[\left[p_{k+1}^2\, a \right]\right]_p, & \text{если} \quad b_k = 1. \end{cases}$$

Окончательным результатом является $p_0 = [[a^e]]_p$. Более подробно, начиная с $p_m = [[a]]_p$, получаем следующее произведение p_k, возводя в квадрат предыдущее произведение и приводя полученное по модулю p, когда $b_k = 0$; возводя в квадрат предыдущее произведение, умножая его на a и приводя полученное по модулю p, когда $b_k = 1$.

---

[6] Число Мерсенна – числа вида $M_n = 2^n - 1$, где n – натуральное число. Названы в честь французского математика **Марен Мерсенна** (Marin Mersenne, 1588–1648), одного из основоположников Парижской Академии наук, друга Декарта и Ферма. Числа Мерсенна играют важную роль в теории чисел, криптографии и генераторах псевдослучайных чисел.

Пример 1.32 Вычислить $[[3^{103}]]_{41}$.

Поскольку $103 = 2^6 + 2^5 + 2^2 + 2^1 + 1 = 1100111$, получаем

k	b_k	$p_k = \left[\left[p_{k+1}^2 \, a^{b_k} \right]\right]_p$
6	1	$3 \pmod{41} \equiv 3$
5	1	$(3^2 \cdot 3) \bmod 41 \equiv 27$
4	0	$27^2 \pmod{41} \equiv 729 \equiv 32$
3	0	$32^2 \pmod{41} \equiv 1024 \equiv 40$
2	1	$(40^2 \cdot 3) \bmod 41 \equiv 4800 \equiv 3$
1	1	$(3^2 \cdot 3) \bmod 41 \equiv 27$
0	1	$(27^2 \cdot 3) \bmod 41 \equiv 2187 \equiv 14$

Поэтому $[[3^{103}]]_{41} = 14$. Используя сравнимость по модулю 41, получаем

$$3^{10} = (3^5)^2 \bmod 41 \equiv 38^2 \bmod 41 \equiv 9;$$
$$3^{50} = (3^{10})^5 \bmod 41 \equiv 9^5 \bmod 41 \equiv 9;$$
$$3^{103} = (3^{50} \cdot 3^{50} \cdot 3^3) \bmod 41 \equiv (9 \cdot 9 \cdot 27) \bmod 41 \equiv 14.$$

Упражнения

1 Вычислить:

а) $[[37]]_4$; б) $[[149]]_{27}$; в) $[[8!]]_6$; г) $[[48]]_{23}$; д) $[[3^{275}]]_{100}$.

2 Определить $\operatorname{ord}_n a$ для $1 \le a \le n-1$, если

а) $n = 9$; б) $n = 20$; в) $n = 27$.

3 Показать при помощи критерия Лукаса, что следующие числа являются простыми:

а) $p = 37$; б) $p = 199$; в) $p = 547$.

1.10 Вычисления в конечных полях

Поле F есть множество, на котором определены операции сложения и умножения, удовлетворяющие требованиям: ассоциативности, коммутативности, дистрибутивности, существования аддитивного нуля и мультипликативной единицы, аддитивных обратных и мультипликативных обратных для всех элементов за исключением нуля. Конечное поле $F(p)$ с конечным числом p элементов играет важную роль в криптографии. В общем случае число элементов $p = q^n$, где q – некоторое простое число и $n \ge 1$. Такие конечные поля называют *полями Галуа*[7] и обозначают $GF(q^n)$ или $GF(q)$ при $n = 1$. Многие криптосистемы базируются на полях Галуа $GF(q)$, где q – большое простое число.

[7] **Галуа Эварист** (Galois Evariste, 1811–1832), французский математик начала XIX века. Труды по теории алгебраических уравнений положили начало развитию современной алгебры. С идеями Галуа связаны такие ее важнейшие понятия, как группа, поле и др. Научное наследие Галуа – небольшое количество весьма кратко написанных работ, из-за новизны идей, не понятых при жизни Галуа. Оставшиеся после преждевременной смерти Галуа работы опубликованы были в 1846 году.

Если q – простое число, то число $a \in [1, q-1]$ является взаимно простым с q, и поэтому обратный элемент a^{-1} имеет единственное значение. Тем самым однозначно определяется операция деления.

Обозначим через $GF^*(q)$ множество всех ненулевых элементов поля $GF(q)$. Некоторый элемент g из $GF^*(q)$ называют *образующим*, или *порождающим элементом* $GF^*(q)$, если для всех a из $GF^*(q)$, найдется такое целое x, что $g^x \equiv a \pmod q$. Всего имеется $\varphi(q-1)$ образующих элементов g. Число x называют дискретным логарифмом элемента a по модулю q. Вычисление дискретных логарифмов – трудно решаемая задача, как и разложение на множители.

Пример 1.33 Поле Галуа $GF(5)$ имеет элементы 0, 1, 2, 3, 4 и описывается таблицами сложения и умножения:

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

×	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Еще один тип поля Галуа, используемый в криптографии, основывается на арифметике по модулю неприводимых многочленов степени n, чьи коэффициенты – целые числа по модулю q, где q – простое. Они имеют элементы, которые описываются многочленами степени не выше $n-1$ в форме

$$a(X) = a_{n-1}X^{n-1} + \ldots + a_1 X + a_0.$$

Каждый элемент $a(X)$ является вычетом по модулю $P(X)$, где $P(X)$ – неприводимый многочлен степени n (т. е. $P(X)$ нельзя разложить на сомножители – многочлены степени меньше n).

Арифметические действия над коэффициентами a_i выполняются по модулю q, а наивысшая степень X равна $n-1$, так как выполняется приведение по модулю многочлена $P(X)$, имеющего старшую степень n.

Особый интерес представляют поля $GF(2^n)$. Здесь коэффициентами a_i являются 0 и 1. Поэтому многочлен $a(X)$ степени не выше $n-1$ можно представить как вектор из n двоичных цифр: $a_{n-1}a_{n-2}\ldots a_1 a_0$.

Каждый из n-битовых векторов соответствует конкретному элементу поля $GF(2^n)$. Например, поле Галуа $GF(2^3)$ имеет элементы:

Многочлены	Двоичная форма
0	000
1	001
x	010
$x+1$	011
x^2	100
x^2+1	101
x^2+x	110
x^2+x+1	111

Организация вычислений в полях Галуа предполагает знание некоторых свойств многочленов и их корней в двоичном поле $GF(2)$. Кратко приведем некоторые из них:

Свойство 1 Ненулевые элементы поля $GF(2^n)$ являются корнями обобщенного многочлена $X^{2^n-1}+1$.

Свойство 2 Каждый многочлен $P(X)$ степени n, неприводимый над полем $GF(2)$, является делителем двучлена $X^{2^n-1}+1$, и каждый делитель двучлена $X^{2^n-1}+1$, неприводимый над полем $GF(2)$, имеет степень, равную n и менее.

Свойство 3 Все элементы поля $GF(2^n)$ можно получить как совокупность остатков от деления $100\ldots00$ на неприводимый многочлен $P(X)$, входящий в разложение двучлена $(X^{2^n-1}+1)$. Эти остатки – корни двучлена $(X^{2^n-1}+1)$, т. е. обращают его в нуль. Число остатков равно (2^n-1).

Свойство 4 В поле $GF(2^n)$ существует примитивный элемент α, такой, что каждый ненулевой элемент поля $GF(2^n)$ может быть представлен как некоторая степень α, т. е. мультипликативная группа $GF(2^n)$ является циклической.

Пример 1.34 Определение элементов α_i поля $GF(2^4)$. Согласно свойству 1, ненулевые элементы поля $GF(2^4)$ являются корнями обобщенного двучлена $(X^{2^4-1}+1)=(X^{15}+1)$. Двучлен $(X^{15}+1)$ можно представить в виде произведения неприводимых многочленов-сомножителей:

$$X^{15}+1=P(X^1)\,P(X^2)\,P_1(X^4)\,P_2(X^4)\,P_3(X^4),$$

где $P(X^1)=X+1$; $P(X^2)=X^2+X+1$; $P_1(X^4)=X^4+X^3+1$;
$P_2(X^4)=X^4+X^3+1$;
$P_3(X^4)=X^4+X^3+X^2+X+1$.

В соответствии со свойством 3, вычислим элементы α_i поля $GF(2^4)$ как совокупность остатков от деления $100\ldots00$ на неприводимый многочлен $P_1(X^4)=X^4+X+1$.

Делят на $P_1(X^4)=X^4+X+1\leftrightarrow 10011$ единицу с возрастающим числом нулей, т. е. делят одночлены X^j, где $j=0, 1, 3, \ldots$, на многочлен (X^4+X+1). Степени одночленов X^0, X^1, X^2, X^3 меньше степени многочлена $P_1(X^4)$, поэтому первые четыре остатка от деления на $P_1(X^4)$ равны делимым, т. е. одночленам X^0, X^1, X^2, X^3. Для одночлена $X^4\leftrightarrow 10000$ получаем остаток

$$\begin{array}{rl} & 1\ 0\ 0\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline & 0\ 0\ 1\ 1 \end{array} \quad \begin{array}{|l} 1\ 0\ 0\ 1\ 1 \\ \hline \\ \leftrightarrow X^4 \end{array}$$

Для одночлена $X^5\leftrightarrow 100000$ получаем остаток

$$\begin{array}{rl} & 1\ 0\ 0\ 0\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline & 0\ 1\ 1\ 0 \end{array} \quad \begin{array}{|l} 1\ 0\ 0\ 1\ 1 \\ \hline \\ \end{array} \leftrightarrow X^5$$

Схема вычисления остатков

$$\begin{array}{rl} & 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^4 \leftrightarrow & 0\ 0\ 1\ 1\ 0\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^7 \leftrightarrow & 1\ 0\ 1\ 1\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^8 \leftrightarrow & 0\ 1\ 0\ 1\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{10} \leftrightarrow & 0\ 1\ 1\ 1\ 0\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{12} \leftrightarrow & 1\ 1\ 1\ 1\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{13} \leftrightarrow & 1\ 1\ 0\ 1\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{14} \leftrightarrow & 1\ 0\ 0\ 1\ 0 \\ \oplus & 1\ 0\ 0\ 1\ 1 \\ \hline X^{15} = X^0 \leftrightarrow & 0\ 0\ 0\ 1 \end{array} \quad \begin{array}{|l} 1\ 0\ 0\ 1\ 1 \\ \hline \\ \end{array}$$

Вычисленные остатки и нулевые элементы $\alpha_0 - \alpha_{14}$ поля Галуа $GF(2^4)$ сведены в таблицу

X^i	Остаток	α_i
X^0	0001	α_0
X^1	0010	α_1
X^2	0100	α_2
X^3	1000	α_3
X^4	0011	α_4
X^5	0110	α_5
X^6	1100	α_6
X^7	1011	α_7
X^8	0101	α_8
X^9	1010	α_9
X^{10}	0111	α_{10}
X^{11}	1110	α_{11}
X^{12}	1111	α_{12}
X^{13}	1101	α_{13}
X^{14}	1001	α_{14}

Поле Галуа $GF(2^4)$ построено как поле многочленов с коэффициентами 0 и 1 по модулю неприводимого многочлена: $P_1(X^4) = X^4 + X + 1 \leftrightarrow 10011$.

В поле Галуа $GF(2^n)$ определены четыре алгебраические операции. Операции сложения и вычитания выполняются как операции поразрядного сложения по модулю 2; операция умножения элементов поля выполняется как умножение соответствующих многочленов с приведением по модулю неприводимого многочлена $P(X)$, т. е. многочлена, по модулю которого построены элементы поля $GF(2^n)$.

Пример 1.35 $\alpha_5 = 0110$, $\alpha_6 = 1100$, $\alpha_5 + \alpha_6 = 1010$, так как

$$\begin{array}{rcccc} & 0 & 1 & 1 & 0 \\ \oplus & 1 & 1 & 0 & 0 \\ \hline & 1 & 0 & 1 & 0 \end{array}$$

Чтобы выполнить деление элемента b на элемент a в поле $GF(2^n)$ по модулю $P(X)$, сначала находят обратный элемент $a^{-1}(\mathrm{mod}\, P(X))$, а затем вычисляют:

$$ba^{-1}(\mathrm{mod}\, P(X)).$$

Каждый двоичный вектор длиной n, исключая 0, является взаимно простым с неприводимым многочленом $P(X)$ независимо от значения $P(X)$. Поэтому число вычетов, взаимно простых с $P(X)$, $\varphi(P(X)) = 2^n - 1$ (расширение функции Эйлера для многочленов). Поэтому

$$a^{-1} \equiv a^{\varphi(P(X))-1}(\mathrm{mod}\, P(X)) \equiv a^{2^n-2}(\mathrm{mod}\, P(X)).$$

Пример 1.36 $\alpha_{14} = 1001$, $\alpha_{14}\,\alpha_{14} = \alpha_{14}^2 = \alpha_{13}$ ïî $\mathrm{mod}\, P_1(X^4) \leftrightarrow 10011$.

$$\begin{array}{rcccccccc} & & & & & \times & 1 & 0 & 0 & 1 \\ & & & & & & 1 & 0 & 0 & 1 \\ \hline & & & & & & 1 & 0 & 0 & 1 \\ \oplus & & & & 1 & 0 & 0 & 1 & & \\ \hline & 1 & 0 & 0 & 0 & 0 & 0 & 1 & & \end{array}$$

$$\begin{array}{rcccccc|ccccc} \oplus & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ & 1 & 0 & 0 & 1 & 1 & & & & & & & \\ \hline & \alpha_{13} \leftrightarrow & 1 & 1 & 0 & 1 & & & & & & & \end{array}$$

Пример 1.37 Пусть $a = 100$ и $P(X) = 1011$ в поле $GF(2^3)$.

$$a^{-1} = 100^{2^3-2}\ (\mathrm{mod}1011) = 100^6\ (\mathrm{mod}1011) = 100^2 \cdot 100^4\ (\mathrm{mod}1011).$$

$100^2(\mathrm{mod}\,1011) = 10000 \oplus 10110 = 110$

или

$$\begin{array}{rcccc|cccc} \oplus & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ & 1 & 0 & 1 & 1 & & & & & \\ \hline & & & 1 & 1 & 0 & & & & \end{array}$$

$100^4 \pmod{1011} = 110^2 \pmod{1011} = 010$

$$
\begin{array}{rcr}
 & \times & \begin{array}{ccc} 1 & 1 & 0 \\ 1 & 1 & 0 \end{array} \\
\hline
 & & \begin{array}{ccc} 0 & 0 & 0 \end{array} \\
 & \begin{array}{ccc} 1 & 1 & 0 \end{array} & \\
\begin{array}{ccc} 1 & 0 & 0 \end{array} & & \\
\hline
\begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \end{array} & &
\end{array}
$$

или

$$
\begin{array}{rl}
\oplus & \begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \end{array} \quad \Big| \begin{array}{cccc} 1 & 0 & 1 & 1 \end{array} \\
 & \begin{array}{cccc} 1 & 0 & 1 & 1 \end{array} \quad \Big| \\
\hline
 & \begin{array}{ccc} 0 & 1 & 0 \end{array}
\end{array}
$$

$100^2 \cdot 100^4 \pmod{1011} = 110 \cdot 010 \pmod{1011} = 1100 \pmod{1011} = 111$

или

$$
\begin{array}{rl}
\oplus & \begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \quad \Big| \begin{array}{cccc} 1 & 0 & 1 & 1 \end{array} \\
 & \begin{array}{cccc} 1 & 0 & 1 & 1 \end{array} \quad \Big| \\
\hline
 & \begin{array}{ccc} 1 & 1 & 1 \end{array}
\end{array}
$$

Таким образом, $a^{-1} = 111$.

Проверка: $a = 100$, $a^{-1} = 111$; $P(X) = 1011$; $a\,a^{-1} = 110 \cdot 100 = 11100$.

$$
\begin{array}{rl}
\oplus & \begin{array}{ccccc} 1 & 1 & 1 & 0 & 0 \end{array} \quad \Big| \begin{array}{cccc} 1 & 0 & 1 & 1 \end{array} \\
 & \begin{array}{cccc} 1 & 0 & 1 & 1 \end{array} \quad \Big| \\
\hline
\oplus & \begin{array}{cccc} 1 & 0 & 1 & 0 \end{array} \\
 & \begin{array}{cccc} 1 & 0 & 1 & 1 \end{array} \\
\hline
 & \begin{array}{ccc} 0 & 0 & 1 \end{array}
\end{array}
$$

т. е. $a\,a^{-1} \pmod{1011} = 1$.

Достоинства вычислений в поле $GF(2^n)$:

1 все элементы поля Галуа имеют конечный размер, деление элементов не имеет каких-либо ошибок округления;

2 сложение и вычитание элементов поля $GF(2^n)$ не требует деления на модуль;

3 алгоритмы вычислений в поле $GF(2^n)$ допускают параллельную реализацию;

4 для поля $GF(2^n)$ обычно применяют в качестве модуля трехчлен $P(X^n) = X^n + X + 1$.

Длинная строка нулей между коэффициентами при X^n и X обеспечивает более простую реализацию быстрого умножения (с приведением по модулю). Трехчлен $P(X^n)$ должен быть неприводимым и примитивным. Трехчлен $P(X^n) = X^n + X + 1$ является примитивным для следующих значений n ($n < 1000$):

1, 3, 4, 6, 9, 15, 22, 28, 30, 46, 60, 63, 127, 153, 172, 303, 471, 532, 865, 900.

2 КРИПТОСИСТЕМЫ С ОТКРЫТЫМ КЛЮЧОМ

2.1 Односторонние функции

В работе “New Directions in Cryptography” Диффи и Хеллман предложили принципиально новый способ организации секретной связи без предварительного обмена ключами, так называемое шифрование с открытым ключом. При этом для зашифровывания и расшифровывания используются разные ключи, и знание одного из них не дает практической возможности определить второй. В результате ключ зашифровывания может быть открытым без потери стойкости шифра, и лишь ключ расшифровывания должен держаться получателем в секрете, поэтому криптосистемы с открытым ключом называют *асимметричными* (*несимметричными*) криптосистемами.

Базовым понятием криптографии с открытым ключом является понятие односторонней функции (one-way function). По заданному аргументу $x \in X$ легко вычислить значение этой функции $F(x)$, в то же время определение x из $F(x)$ трудновычислимо, т. е. нет алгоритма для решения этой задачи с полиномиальным временем работы. Теоретически x по известному значению $F(x)$ можно найти всегда, проверяя по очереди все возможные значения x до тех пор, пока соответствующее значение $F(x)$ не совпадет с заданным. Однако практически при значительной размерности множества X такой подход неосуществим.

Односторонней функцией называется функция $F(x)$: $X \to Y$, $x \in X$, обладающая двумя свойствами:

существует полиномиальный алгоритм вычисления значений $y = F(x)$;

не существует полиномиального алгоритма инвертирования функции $F(x) = y$.

Полиномиальным будем называть алгоритм, выполнение которого заканчивается не более чем за $p(n)$ шагов, где n – размер входной задачи, измеряемый, как правило, количеством символов текста, описывающего эту задачу.

Заметим, что до сих пор не доказано существование односторонних функций. Использование их в качестве основы асимметричных алгоритмов шифрования допустимо только до тех пор, пока не найдены эффективные алгоритмы, выполняющие нахождение односторонних функций за полиномиальное время.

Примером кандидата на звание односторонней функции является модульное возведение в степень, т. е. функция $F(x) \equiv a^x \bmod p$, где a – примитивный элемент поля $GF(p)$; p – большое простое число. То, что эта функция может быть эффективно вычислена даже при разрядности параметров в несколько сотен знаков, можно показать на примере: a^{25} можно вычислить с помощью шести операций умножения (умножением считается и возведение в квадрат). Число 25 в двоичной системе счисления записывается как 11001, так что $25 = 2^4 + 2^3 + 2^0$.

Поэтому:

$$a^{25} \bmod p \equiv (a^{16}a^8a) \bmod p \equiv ((((a^2\, a)^2\,)^2\,)^2\, a) \bmod p.$$

Задача вычисления функции, обратной модульному возведению в степень, называется *задачей дискретного логарифмирования*. На сегодняшний день неизвестно ни одного эффективного алгоритма вычисления дискретных логарифмов больших чисел.

Односторонняя функция в качестве функции зашифровывания неприменима, так как, если $F(x)$ – зашифрованное сообщение x, то никто, в том числе и законный получатель, не сможет восстановить x. Обойти эту проблему можно с помощью односторонней функции с секретом (one-way trapdoor function). Например, функция E_k: $X \to Y$, имеющая обратную функцию D_k: $Y \to X$, однако узнать обратную функцию только по E_k без знания секрета k невозможно.

Односторонней функцией с секретом k называется функция E_k: $X \to Y$, зависящая от параметра k и обладающая тремя свойствами:

при любом k существует полиномиальный алгоритм вычисления значений $E_k(x)$;

при неизвестном k не существует полиномиального алгоритма инвертирования E_k;

при известном k существует полиномиальный алгоритм инвертирования E_k.

Функцию E_k можно использовать для зашифровывания информации, а обратную ей функцию D_k – для расшифровывания, так как при всех $x \in X$ справедливо

$$D_k\,(E_k(x)) = x.$$

При этом подразумевается, что тот, кто знает, как зашифровывать информацию, вовсе не обязательно должен знать, как расшифровывать ее. Так же как и в случае с односторонней функцией, вопрос о существовании односторонних функций с секретом открыт. Для практической криптографии найдено несколько функций – кандидатов на звание односторонней функции с секретом. Для них второе свойство не доказано, однако известно, что задача инвертирования эквивалентна решению трудной математической задачи.

Применение односторонней функций с секретом в криптографии позволяет:

организовывать обмен шифрованными сообщениями с использованием только открытых каналов связи, т. е. отказаться от секретных каналов связи для предварительного обмена ключами;

включать при вскрытии шифра сложную математическую задачу и тем самым повышать стойкость шифра;

решать новые криптографические задачи, отличные от шифрования (электронная цифровая подпись и др.).

Стойкость большинства современных асимметричных алгоритмов базируется на двух математических проблемах, которые на данном этапе являются трудновычисляемыми:

дискретное логарифмирование в конечных полях;

факторизация больших чисел.

Поскольку на сегодняшний день не существует эффективных алгоритмов решения данных задач либо их решение требует привлечения больших вычислительных ресурсов или временных затрат, эти математические задачи нашли широкое применение в построении асимметричных алгоритмов.

2.2 Модель криптосистемы с открытым ключом

Несимметричные криптосистемы предполагают наличие двух ключей: открытого, предназначенного для зашифровывания передаваемого сообщения, и закрытого, с помощью которого получатель расшифровывает принятую криптограмму.

Несекретный ключ может передаваться по открытому каналу. Его знание не дает злоумышленнику возможности получить доступ к информации, содержащейся в сообщении.

На рис. 2.1 приведена структурная схема криптосистемы с открытым ключом.

Генератор ключевой пары выдает пару ключей (K_1, K_2) в зависимости от начальных условий (НУ), известных только получателю сообщения. Открытый ключ K_1 передается отправителю по незащищенному каналу связи. Отправитель зашифровывает сообщение M, используя ключ K_1. Шифртекст C передается по незащищенному каналу связи получателю.

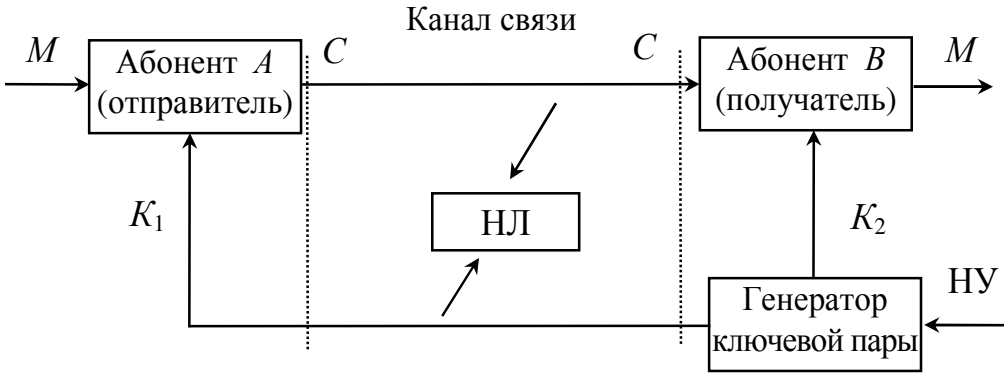

Рисунок 2.1 – Структурная схема криптосистемы с открытым ключом

Получатель расшифровывает криптограмму (восстанавливая исходное сообщение), используя секретный ключ K_2.

Несанкционированное лицо (НЛ) имеет доступ к незащищенным каналам и поэтому может перехватить криптограмму C и открытый ключ K_1. Более того, оно может владеть алгоритмом шифрования. Единственное, чем оно не владеет – ключом K_2.

Наиболее известные системы с открытым ключом:

– ранцевая криптосистема Меркле–Хеллмана (knapsack cryptosystem);

– криптосистема RSA;

– криптосистема Эль-Гамаля (ElGamal Cryptosystem);

– криптосистема Диффи–Хеллмана;

– криптосистема, основанная на свойствах эллиптических кривых (Elliptic Curve Cryptosystem);

– электронно-цифровая подпись.

2.3 Криптоалгоритм Меркле–Хеллмана

Криптосистема была разработана Ральфом Меркле и Мартином Хеллманом в 1978 году [10] и стала первым алгоритмом шифрования с открытым ключом широкого назначения. Криптосистема Меркле–Хеллмана относится к ранцевым алгоритмам. Вначале алгоритм обеспечивал только шифрование сообщений, но, позднее, Ади Шамир модифицировал соответствующую криптосистему для поддержки средств цифровой подписи. Безопасность ранцевых алгоритмов опирается на известную математическую задачу об укладке ранца (рюкзака). В основе алгоритма лежит идея шифрования сообщения на основе решения серии задач по укладке рюкзака.

Пусть задана последовательность элементов a_i, входящих во множество A, которая удовлетворяет условию, в соответствии с которым каждый очередной элемент имеет вес, превышающий сумму весов всех остальных элементов, т. е.

$$a_i = \sum_{j=1}^{i-1} a_j .$$

Такая последовательность называется *быстрорастущей*.

Рассмотрим задачу. Определить подмножество элементов S, состоящее из элементов a_i, принадлежащих множеству A, сумма которых равна T, при условии, что выполняется требование $a_i = \sum_{j=1}^{i-1} a_j$.

$$\sum_{i \in S} a_i = T .$$

Пример 2.1 Пусть множество A задано следующим набором элементов a_i, удовлетворяющих условию быстрорастущей последовательности

1, 4, 7, 13, 27, 55, 150, 310, 623.

Требуется определить те элементы заданного множества, сумма которых составляет 1002.

Будем анализировать элементы множества A, начиная с последнего.

Число 623 < 1002 является элементом подмножества S, поскольку без него составить данное подмножество нельзя.

Число 310 также входит во множество S, поскольку

$$1002 – 623 = 379 > 310$$

и без него также нельзя получить число 1002.

Число 150 не входит в подмножество S, поскольку

$$1002 – (623 + 310) = 69 < 150.$$

Число 55 входит в подмножество S, поскольку

$$1002 – (623 + 310) = 69 > 55.$$

Число 27 не входит в подмножество S, поскольку

$$1002 - (623 + 310 + 55) = 14 < 27.$$

Число 13 входит в подмножество S, поскольку

$$1002 - (623 + 310 + 55) = 14 > 13.$$

Число 7 не входит в подмножество S, поскольку

$$1002 - (623 + 310 + 55 + 13) = 1 < 7.$$

Число 4 не входит в подмножество S, поскольку

$$1002 - (623 + 310 + 55 + 13) = 1 < 4.$$

И, наконец, число 1 входит в подмножество S, поскольку

$$1002 - (623 + 310 + 55 + 13) = 1.$$

Таким образом, подмножество S будет включать числа

$$\{1, 13, 55, 310, 623\}.$$

Задача вычисления суммы подмножества S для быстрорастущей последовательности классифицируется как легкая.

В общем виде она может быть сформулирована следующим образом. Пусть $\vec{A} = (a_1, a_2, \ldots, a_n)$ и $\vec{B} = (b_1, b_2, \ldots, b_n)$ – два вектора. Их скалярное произведение определяется

$$\vec{C} = \vec{A} \times \vec{B} = \sum_{i=1}^{n} a_i b_i .$$

Нахождение $\vec{C}$ по заданным $\vec{A}$ и $\vec{B}$ не связано с какими бы то ни было трудностями, тогда как задача восстановления $\vec{B}$ по заданным $\vec{C}$ и $\vec{A}$ относится к разряду трудновычислимых. Обычно элементы b_i вектора $\vec{B}$ в криптографических алгоритмах принимают значения $\{1, 0\}$. В этом векторе единицы располагаются на позициях, соответствующих элементам, включаемым во множество S.

Суть криптографического алгоритма, основанного на задаче "*о наполнении рюкзака*", сводится к следующему.

Получатель информации выбирает начальный вектор

$$\vec{W} = (w_1, w_2, \ldots, w_n),$$

элементы которого w_i, где $i = 1, 2, \ldots, n$, удовлетворяют требованию быстрорастущей последовательности.

Затем получатель выбирает простое число p, значение которого больше суммы w_i, и некоторое число r (не обязательно простое), удовлетворяющее условию $r \le p - 1$.

После этого он формирует открытый ключ $\vec{K} = (k_1, k_2, \ldots, k_n)$ по правилу

$$k_i \equiv (w_i r) \bmod p, \qquad i = 1, 2, \ldots, n.$$

Значения элементов ключа передаются отправителю сообщения по открытому каналу связи. Значения начального вектора $\vec{W}$, а также чисел p и r содержатся получателем сообщения в секрете.

Отправитель разбивает шифруемое сообщение $\vec{M}$ на блоки размером по n символов.

$$\vec{M} = (m_1, m_2, \ldots, m_n),$$

где $m_i = \{1,\ 0\},\ i = 1, 2, \ldots, n$.

Используя ключ $\vec{K}$, отправитель зашифровывает сообщение по правилу

$$\vec{C} = \vec{K} \times \vec{M} = \sum_{i=1}^{n} k_i m_i .$$

Затем шифрограмма передается отправителем по открытому каналу получателю.

Получатель решает уравнение

$$re \equiv 1 \pmod{p}$$

относительно e, иначе говоря, находит обратное значение для числа r по модулю p, после чего преобразует полученную шифрограмму по правилу

$$C' \equiv (\vec{C}\, e) \bmod p.$$

C' можно привести к виду

$$C' \equiv \left(\vec{C}\ e\right) \bmod p \equiv \left(\sum_{i=1}^{n} k_i m_i e \right) \bmod p \equiv \left(\sum_{i=1}^{n} m_i w_i re \right) \bmod p \equiv \sum_{i=1}^{n} w_i m_i .$$

Далее решается задача определения суммы подмножества.

Пример 2.2 Пусть $n = 5$. Получатель выбирает начальный вектор

$$\vec{W} = (2, 4, 7, 15, 29).$$

Для выбранного набора чисел выполняется требование быстрорастущей последовательности. Получатель выбирает $p = 59$ и $r = 40$. С учетом выбранных величин формируется ключ

$$k_1 \equiv (2 \cdot 40) \bmod 59 \equiv 21;$$
$$k_2 \equiv (4 \cdot 40) \bmod 59 \equiv 42;$$
$$k_3 \equiv (7 \cdot 40) \bmod 59 \equiv 44;$$
$$k_4 \equiv (15 \cdot 40) \bmod 59 \equiv 10;$$
$$k_5 \equiv (29 \cdot 40) \bmod 59 \equiv 39.$$

Далее открытый ключ $\vec{K} = (21, 42, 44, 10, 39)$ передается отправителю по незащищенному каналу.

Отправитель передает сообщение

$$\vec{M} = (1, 1, 0, 1, 1),$$

предварительно зашифровав его по правилу

$$\vec{C} = 21 \cdot 1 + 42 \cdot 1 + 44 \cdot 0 + 10 \cdot 1 + 39 \cdot 1 = 112.$$

Предварительно решив уравнение

$$40e \equiv 1 \pmod{59}$$

относительно e

$$e \equiv \left(40^{\varphi(59)-1}\right) \bmod 59 \equiv 40^{57} \left(\bmod 59\right) \equiv \left(40^{32} \cdot 40^{16} \cdot 40^{8} \cdot 40\right) \bmod 59 \equiv$$
$$\equiv 15\left(29\left(41 \cdot 40\left(\bmod 59\right)\right) \bmod 59\right) \bmod 59 \equiv \left(15\left(29 \cdot 47\right) \bmod 59\right) \bmod 59 \equiv$$
$$\equiv \left(15 \cdot 6\right) \bmod 59 \equiv 31$$

и определив, что его значение равно 31, получатель вычисляет значение

$$C' \equiv (112 \cdot 31) \bmod 59 \equiv 50.$$

Определив значение C', получатель решает простую задачу о содержимом рюкзака, анализируя элементы начального вектора, начиная с последнего:

$$\begin{aligned} 50 > 29 \quad &\rightarrow \quad m_5 = 1; \\ 50 - 29 = 21 > 15 \quad &\rightarrow \quad m_4 = 1; \\ 21 - 15 = 6 < 7 \quad &\rightarrow \quad m_3 = 0; \\ 6 > 4 \quad &\rightarrow \quad m_2 = 1; \\ 6 - 4 = 2 = 2 \quad &\rightarrow \quad m_1 = 1. \end{aligned}$$

Таким образом, полученное сообщение имеет вид

$$\vec{M} = (1, 1, 0, 1, 1).$$

Упражнение

Зашифровать и расшифровать с помощью алгоритма Меркле–Хеллмана:

а) $n = 6$; $p = 197$; $r = 75$; $\vec{W} = (3, 7, 11, 24, 47, 101)$; $\vec{M} = (1, 0, 0, 1, 1, 0)$.
б) $n = 7$; $p = 397$; $r = 75$; $\vec{W} = (3, 7, 11, 24, 47, 101, 195)$; $\vec{M} = (1, 0, 0, 1, 1, 0, 1)$.
в) $n = 5$; $p = 89$; $r = 57$; $\vec{W} = (2, 7, 10, 21, 43)$; $\vec{M} = (1, 0, 0, 1, 1)$.
г) $n = 6$; $p = 173$; $r = 92$; $\vec{W} = (2, 7, 10, 21, 43, 85)$; $\vec{M} = (1, 1, 0, 1, 1, 1)$.

2.4 Система Idempotent Elements

Система Idempotent Elements (IE) является модификацией системы Меркле–Хеллмана. Получатель выбирает начальный вектор $\vec{W} = (p_1, p_2, \ldots, p_n)$, где $p_1, p_2, \ldots, p_n$ – различные простые числа, и вычисляет n равносильных элементов $e_1, e_2, \ldots, e_n$ (idempotent elements):

$$\begin{array}{llll} e_1 \equiv 1 \pmod{p_1}, & e_1 \equiv 0 \pmod{p_2}, & \ldots, & e_1 \equiv 0 \pmod{p_n}; \\ e_2 \equiv 0 \pmod{p_1}, & e_2 \equiv 1 \pmod{p_2}, & \ldots, & e_2 \equiv 0 \pmod{p_n}; \\ \ldots & \ldots & \ldots & \ldots \\ e_n \equiv 0 \pmod{p_1}, & e_n \equiv 0 \pmod{p_2}, & \ldots, & e_n \equiv 1 \pmod{p_n}. \end{array}$$

Получатель выбирает простое число q, такое, что $q > \sum_{i=1}^{n} e_i$, и произвольное число r. Формируется открытый ключ $K = (k_1, k_2, \ldots, k_n)$ по правилу

$$k_i \equiv (e_i r) \bmod q, \qquad i = 1, 2, \ldots, n.$$

Отправитель произвольно разбивает сообщение $M = (m_1, m_2, \ldots, m_n)$ на два, M_1 и M_2, так, что каждый элемент m_i входит один раз либо в M_1, либо в M_2. Зашифровывается сообщение по формуле

$$C = \left| \sum_{M_1} k_i m_i - \sum_{M_2} k_i m_i \right|.$$

Для расшифровывания сообщения получатель решает уравнение $r\,e \equiv 1 \pmod{q}$ и вычисляет C' и C'' по формулам

$$C' \equiv (C\,e) \bmod q;$$
$$C'' = N - C',$$

где $N = \prod_{i=1}^{n} p_i$. Сообщение скрывается в числах C' и C'', поэтому получатель представляет их в векторной форме:

$$\vec{C}' = (C' \bmod p_1,\ C' \bmod p_2,\ \ldots,\ C' \bmod p_n);$$
$$\vec{C}'' = (C'' \bmod p_1,\ C'' \bmod p_2,\ \ldots,\ C'' \bmod p_n).$$

Искомое сообщение M получается из векторов $\vec{C}'$ и $\vec{C}''$, элементы которых равны $1 \pmod{p_i}$, $-1 \pmod{p_i}$ или $0 \pmod{p_i}$, заменой $1 \pmod{p_i}$ и $-1 \pmod{p_i}$ на 1 и $0 \bmod p_i$ на 0.

Пример 2.3

Пусть $n = 5$, $\vec{M} = (1, 1, 0, 1, 1)$ и $\vec{W} = (2, 3, 5, 7, 11)$.

Элементы e_1, e_2, e_3, e_4, e_5 определяются из уравнений (по теореме о “китайском остатке”):

$e_1 \equiv 1 \pmod 2$,	$e_1 \equiv 0 \pmod 3$,	$e_1 \equiv 0 \pmod 5$,	$e_1 \equiv 0 \pmod 7$,	$e_1 \equiv 0 \pmod{11}$,	$e_1 = 1155$.
$e_2 \equiv 0 \pmod 2$,	$e_2 \equiv 1 \pmod 3$,	$e_2 \equiv 0 \pmod 5$,	$e_2 \equiv 0 \pmod 7$,	$e_2 \equiv 0 \pmod{11}$,	$e_2 = 1540$.
$e_3 \equiv 0 \pmod 2$,	$e_3 \equiv 0 \pmod 3$,	$e_3 \equiv 1 \pmod 5$,	$e_3 \equiv 0 \pmod 7$,	$e_3 \equiv 0 \pmod{11}$,	$e_3 = 1386$.
$e_4 \equiv 0 \pmod 2$,	$e_4 \equiv 0 \pmod 3$,	$e_4 \equiv 0 \pmod 5$,	$e_4 \equiv 1 \pmod 7$,	$e_4 \equiv 0 \pmod{11}$,	$e_4 = 330$.
$e_5 \equiv 0 \pmod 2$,	$e_5 \equiv 0 \pmod 3$,	$e_5 \equiv 0 \pmod 5$,	$e_5 \equiv 0 \pmod 7$,	$e_5 \equiv 1 \pmod{11}$,	$e_5 = 210$.

Выбираем простое число q из условия $q > \sum_{i=1}^{5} e_i$:

$$\sum_{i=1}^{5} e_i = 1155 + 1540 + 1386 + 330 + 210 = 4621.$$

Выбираем простое $q = 4649$ и произвольное число $r = 3475$, с помощью которых вычисляем элементы открытого ключа

$$k_1 \equiv (1155 \cdot 3475) \bmod 4649 \equiv 1538;$$
$$k_2 \equiv (1540 \cdot 3475) \bmod 4649 \equiv 501;$$
$$k_3 \equiv (1386 \cdot 3475) \bmod 4649 \equiv 4635;$$
$$k_4 \equiv (330 \cdot 3475) \bmod 4649 \equiv 3096;$$
$$k_5 \equiv (210 \cdot 3475) \bmod 4649 \equiv 4506.$$

Разобьем $\vec{M}$ на M_1 и M_2, $M_1 = (m_1, m_3) = (1, 0)$ и $M_2 = (m_2, m_4, m_5) = (1, 1, 1)$. Зашифровываем сообщение

$$C = \left|\left(m_1k_1 + m_3k_3\right) - \left(m_2k_2 + m_4k_4 + m_5k_5\right)\right| = \left|k_1 - \left(k_2 + k_4 + k_5\right)\right| =$$
$$= \left|1538 - \left(501 + 3096 + 4506\right)\right| = \left|1538 - 8103\right| = 6565.$$

Для расшифровывания сообщения решаем уравнение $3475\ e \equiv 1 \pmod{4649}$ и находим $e = 4550$.

Вычисляем

$$C' \equiv (6565 \cdot 4550) \bmod 4649 \equiv 925;$$
$$N = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 = 2310;$$
$$C'' = 2310 - 925 = 1385.$$

Представим в векторной форме:

$\vec{C}' = (925 \bmod 2, 925 \bmod 3, 925 \bmod 5, 925 \bmod 7, 925 \bmod 11) = (1, 1, 0, 1, 1);$
$\vec{C}'' = (1385 \bmod 2, 1385 \bmod 3, 1385 \bmod 5, 1385 \bmod 7, 1385 \bmod 11) = (1, -1, 0, -1, -1).$

После замены в $\vec{C}''$ –1 на 1 получаем исходное сообщение $M = (1, 1, 0, 1, 1)$.

Практическая реализация. Для последовательности из пяти элементов решить задачу ранцевым алгоритмом нетрудно. Пригодные для практического использования рюкзаки должны содержать не менее 250 элементов. Длина каждого элемента быстрорастущей последовательности должна находиться в диапазоне между 200 и 400 битами, а длина модуля должна быть от 100 до 200 битов. Для получения этих значений в практических реализациях используют генераторы случайных последовательностей.

Вскрывать подобные рюкзаки “в лоб” бесполезно. Если даже компьютер сможет проверять миллион вариантов в секунду, проверка всех возможных вариантов рюкзака потребует свыше 10^{46} лет [8].

Стойкость ранцевого метода. Криптосистему, основанную на задаче об укладке рюкзака, взломали не миллион компьютеров, а пара криптографов. Шамир и Циппел [11] обнаружили слабые места в преобразовании, что позволило восстановить быстрорастущую последовательность рюкзака из нормальной. После взлома схемы Меркле–Хеллмана было предложено множество других систем на основе алгоритма укладки рюкзака (Graham–Shamir, Lu–Lee, Goodman–McAuley, Niemi и другие), но все они были проанализированы и взломаны с использованием одних и тех же криптографических методов.

Упражнение

Зашифровать и расшифровать с помощью системы Idempotent Elements:

а) $n = 4$; $W = (5, 7, 11, 13)$; $M = (1, 1, 0, 1)$.
б) $n = 5$; $W = (2, 5, 7, 13, 17)$; $M = (1, 0, 0, 1, 0)$.
в) $n = 5$; $W = (2, 7, 11, 13, 17)$; $M = (1, 0, 0, 1, 0)$.
г) $n = 7$; $W = (2, 3, 5, 7, 11, 13, 17)$; $M = (1, 1, 1, 1, 0, 1, 0)$.

2.5 Алгоритм Шамира

Шифр, предложенный Ади Шамиром (A. Shamir)[8], позволяет организовать обмен секретными сообщениями по открытой линии связи для лиц, которые не имеют защищенных каналов и секретных ключей.

Пусть есть два абонента A и B, соединенные линией связи (рис. 2.2).

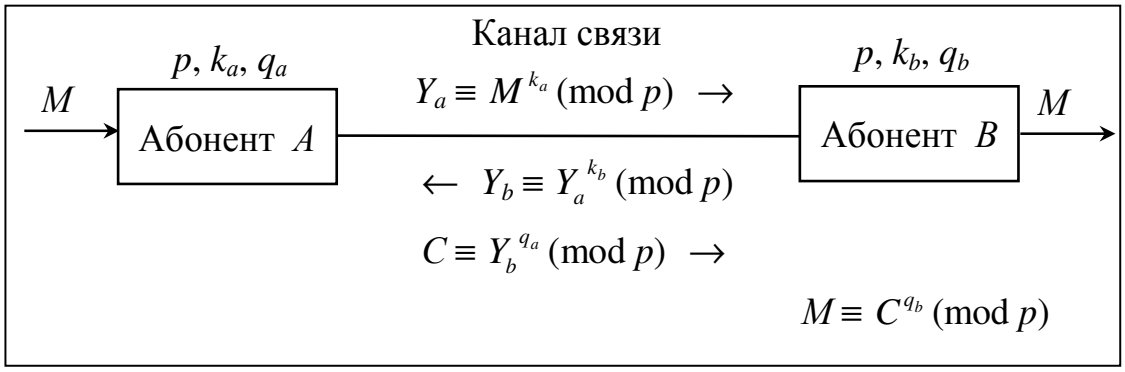

Рисунок 2.2 – Алгоритм Шамира

Сторона A хочет передать сообщение M абоненту B так, чтобы никто не узнал его содержание. Абонент A выбирает случайное большое простое число p и открыто передает его абоненту B. Затем A выбирает два числа – k_a и q_a, такие, что

$$k_a\, q_a \equiv 1 \bmod (p-1).$$

Числа k_a и q_a являются секретными. Абонент B тоже выбирает два секретных числа k_b и q_b, такие, что

$$k_b\, q_b \equiv 1 \bmod (p-1).$$

После выбора чисел сторона A передает свое сообщение M, используя трехступенчатый протокол. Если $M < p$ (M рассматривается как число), то сообщение M передается сразу; если же $M > p$, то сообщение представляется в виде $M = m_1, m_2, ..., m_i$, где все $m_i < p$, и затем передаются последовательно $m_1, m_2, ..., m_i$. При этом для шифрования каждого m_i лучше выбирать случайно новые пары (k_a, q_a) и (k_b, q_b) – в противном случае надежность системы понижается. В настоящее время такой шифр используется преимущественно для передачи чисел, например секретных ключей, значения которых меньше p. Таким образом, мы будем рассматривать только случай $M < p$. Дадим описание протокола.

1 Абонент A вычисляет число Y_a и по открытой линии связи пересылает абоненту B

$$Y_a \equiv M^{k_a} \pmod p.$$

[8] **Ади Шамир** (род. в 1952 г., Тель-Авив, Израиль) **–** ученый в области теории вычислительных систем, лауреат премии Тьюринга. Вместе с Ривестом и Адлеманом разработал криптографический алгоритм с открытым ключом RSA, а также внес большой вклад в развитие дифференциального криптоанализа.

2 Абонент B, получив Y_a, вычисляет число Y_b и по открытой линии связи пересылает абоненту A

$$Y_b \equiv Y_a^{k_b} \pmod{p}.$$

3 Сторона A вычисляет число C и передает его стороне B:

$$C \equiv Y_b^{q_a} \pmod{p}.$$

4 Сторона B, получив число C, вычисляет сообщение M:

$$M \equiv C^{q_b} \pmod{p}.$$

Справедливость последнего пункта утверждения вытекает из следующей цепочки равенств:

$$M \equiv C^{q_b} \pmod{p} \equiv \left(Y_b^{q_a}\right)^{q_b} \bmod p \equiv \left(Y_a^{k_b}\right)^{q_a q_b} \bmod p \equiv \left(M^{k_a}\right)^{k_b q_a q_b} \bmod p \equiv$$
$$\equiv M^{k_a k_b q_a q_b} \pmod{p} \equiv M^{(k_a q_a k_b q_b) \bmod (p-1)} \pmod{p} \equiv M.$$

Пример 2.4 Пусть абонент A хочет передать абоненту B сообщение $M = 10$. Сторона A выбирает $p = 23$, $k_a = 7$ (НОД (7, 22) = 1) и вычисляет $q_a = 19$:

$$q_a \equiv \left(7^{\varphi(22)-1}\right) \bmod (23-1) \equiv 7^9 \left(\bmod 22\right) \equiv \left(7^6 \cdot 7^2 \cdot 7\right) \bmod 22 \equiv \left(15 \cdot 5 \cdot 7\right) \bmod 22 \equiv 19.$$

Аналогично, сторона B выбирает параметры $k_b = 5$ и вычисляет $q_b = 9$:

$$q_b \equiv \left(5^{\varphi(22)-1}\right) \bmod (23-1) \equiv 5^9 \left(\bmod 22\right) \equiv \left(5^6 \cdot 5^2 \cdot 5\right) \bmod 22 \equiv \left(5 \cdot 3 \cdot 5\right) \bmod 22 \equiv 9.$$

Переходим к протоколу Шамира:

1 $Y_a \equiv 10^7 \pmod{23} \equiv \left(10^4 \cdot 10^2 \cdot 10\right) \bmod 23 \equiv \left(18 \cdot 8 \cdot 10\right) \bmod 23 \equiv 14.$

2 $Y_b \equiv 14^5 \pmod{23} \equiv \left(14^2 \cdot 14^2 \cdot 14\right) \bmod 23 \equiv \left(12 \cdot 12 \cdot 14\right) \bmod 23 \equiv 15.$

3 $C \equiv 15^{19} \pmod{23} \equiv \left(15^{16} \cdot 15^2 \cdot 15\right) \bmod 23 \equiv \left(16 \cdot 18 \cdot 15\right) \bmod 23 \equiv 19.$

4 $M \equiv 19^9 \pmod{23} \equiv \left(19^6 \cdot 19^2 \cdot 19\right) \bmod 23 \equiv \left(2 \cdot 16 \cdot 19\right) \bmod 23 \equiv 10.$

Таким образом, абонент B получил передаваемое сообщение $M = 10$.

Шифр Шамира полностью решает задачу обмена сообщениями, закрытыми для прочтения, в случае, когда абоненты могут пользоваться только открытыми линиями связи. Однако при этом сообщение пересылается трижды от одного абонента к другому, что является недостатком.

Упражнение

Для шифра Шамира с заданными параметрами p, k_a, k_b найти недостающие параметры и описать процесс передачи сообщения M от абонента A к B.

а) $p = 19$; $k_a = 5$; $k_b = 7$; $M = 4$. б) $p = 23$; $k_a = 15$; $k_b = 7$; $M = 6$.
в) $p = 19$; $k_a = 11$; $k_b = 5$; $M = 10$. г) $p = 23$; $k_a = 9$; $k_b = 3$; $M = 17$.

2.6 Стандарт асимметричного шифрования RSA

Вскоре после появления ранцевого алгоритма Меркле–Хеллмана был создан первый полноценный алгоритм с открытым ключом, который можно использовать и для шифрования и для создания цифровых подписей – алгоритм RSA. Алгоритм RSA назван в честь трех изобретателей – Рона Ривеста[9], Ади Шамира и Леонарда Адлемана[10] и был предложен 1978 году [12, 13]. Разработчикам данного алгоритма удалось эффективно воплотить идею односторонних функций с секретом. Стойкость RSA базируется на сложности факторизации больших целых чисел. Открытый и закрытый ключи являются функциями двух больших простых чисел разрядностью 100…200 десятичных цифр и даже больше. Восстановление открытого текста по шифртексту и открытому ключу равносильно разложению числа на два больших простых множителя. Многие годы алгоритм RSA противостоит многочисленным попыткам криптографического вскрытия. Криптоанализ ни доказывает, ни опровергает безопасность алгоритма RSA, тем самым обосновывая степень доверия к алгоритму. В 1993 году алгоритм RSA был принят в качестве стандарта (PKCS # 1: RSA Encryption Standard).

Алгоритм RSA. Случайным образом выбираются два больших простых числа p и q. Рассчитывается произведение $n = p\,q$.

Вычисляется функция Эйлера $\varphi(n) = (p-1)(q-1)$.

Случайным образом выбирается простое число e – ключ зашифровывания, которое удовлетворяет условиям $e < \varphi(n)$; $(e, \varphi(n)) = 1$.

Вычисляется число d – ключ расшифровывания, которое является обратным числу e, т. е.

$$e\,d \equiv 1 \pmod{\varphi(n)}.$$

Пара чисел (e, n) делается открытым ключом и помещается в общедоступный справочник, а числа p, q держатся в секрете, d – секретный ключ.

При шифровании сообщение M сначала разбивается на цифровые блоки размерами меньше n, т. е. если p и q являются 100-разрядными простыми числами, то n будет содержать около 200 разрядов и каждый блок сообщения m_i

---

[9] **Рональд Линн Ривест** (Ronald L. Rivest, род. в 1947 г., Скенектади, Нью-Йорк) – американский специалист по криптографии. Имеет звание *профессора имени Эндрю и Эрны Витерби по компьютерным наукам* на факультете электротехники и компьютерных наук (EECS) и состоит в штате кафедры CSAIL в Массачусетском технологическом институте. Ривест – один их авторов алгоритма RSA. Изобрел такие симметричные алгоритмы шифрования как RC2, RC4, RC5 и принимал участие в разработке RC6. Буквы "RC" означают "шифр Ривеста" (*Rivest Cipher*), или, неформально, "код Рона" (*Ron's Code*). Помимо RC, он автор хэш-функций MD2, MD4, MD5, MD6.

[10] **Леонард Макс Адлеман** (Leonard Adleman, род. в 1945 г., в Калифорнии, вырос в Сан-Франциско), поступил в Калифорнийский университет, где получил степени бакалавра по математике в 1968 г. и доктора философии по электротехнике и компьютерным наукам в 1976 г. Адлеман – американский ученый-теоретик в области компьютерных наук, профессор компьютерных наук и молекулярной биологии в университете Южной Калифорнии. Известен как соавтор системы шифрования RSA.

должен иметь около 200 разрядов в длину. Зашифрованное сообщение C будет состоять из блоков c_i той же самой длины. Формула зашифровывания

$$C \equiv M^e \pmod{n}.$$

Расшифровывание обеспечивается операцией возведения в степень d по модулю n принятого шифртекста C

$$M \equiv C^d \pmod{n},$$

так как

$$C^d \pmod{n} \equiv M^{ed} \pmod{n} \equiv (M^{k(p-1)(q-1)+1}) \bmod n \equiv (M\, M^{k(p-1)(q-1)}) \bmod (pq) \equiv M.$$

Все вышесказанное сведено в табл. 2.1.

Таблица 2.1 – Алгоритм RSA

Открытый ключ: n – произведение двух простых чисел p и q (p и q должны храниться в секрете); e – ключ зашифровывания, число взаимно простое с функцией Эйлера $(e, \varphi(n)) = 1$ и $e < \varphi(n)$.
Закрытый ключ: $d \equiv e^{-1} \pmod{\varphi(n)}$ – ключ расшифровывания.
Зашифровывание: $C \equiv M^e \pmod{n}$.
Расшифровывание: $M \equiv C^d \pmod{n}$.

Пример 2.5 Дано $p = 11$, $q = 5$, $M = 15$.

Вычисляем $n = 11 \cdot 5 = 55$.

Определяем функцию Эйлера $\varphi(55) = (11–1)(5–1) = 40$.

Выбираем ключ зашифровывания $e = 7$, который удовлетворяет условиям $7 < 40$; НОД $(7, 40) = 1$.

Определяем d – ключ расшифровывания – из уравнения

$$7d \equiv 1 \pmod{40}.$$

Рассмотрим несколько способов нахождения d.

Способ 1 Для решения уравнения $7d \equiv 1 \pmod{40}$ используем алгоритм Евклида

$$\begin{aligned} 40 &= 7 \cdot 5 + 5; \\ 7 &= 5 \cdot 1 + 2; \\ 5 &= 2 \cdot 2 + 1; \\ 2 &= 1 \cdot 2 + 0. \end{aligned}$$

Обратная подстановка дает

$$\begin{aligned} 1 &= 5 – 2 \cdot 2 = 5 – (7 – 5 \cdot 1)2 = 5 \cdot 3 + 7(–2) = \\ &= (40 – 7 \cdot 5)3 + 7(–2) = 40 \cdot 3 + 7(–17). \end{aligned}$$

Поскольку $–17 \equiv 23 \pmod{40}$, то $d = 23$.

Способ 2 Выражение $7d \equiv 1 \pmod{40}$ представим в виде

$$7d + 40\,k = 1.$$

Для решения диофантова уравнения используем алгоритм Евклида

$$\begin{array}{r} 40 = 7 \cdot 5 + 5; \\ 7 = 5 \cdot 1 + 2; \\ 5 = 2 \cdot 2 + 1; \\ \hline 2 = 1 \cdot 2 + 0. \end{array} \quad \uparrow$$

Составляем ряд l_i коэффициентов в обратном порядке вычислений (показано стрелкой), последняя строка не учитывается; получаем

$$l_1 = 2, \quad l_2 = 1, \quad l_3 = 5.$$

Составляем новый ряд h_i коэффициентов, первое значение которого всегда равно единице: $h_1 = 1$; второе значение ряда h_i равно первому значению ряда l_1; $h_2 = l_1$. Остальные значения ряда h_i вычисляются по формуле

$$h_i = l_{i-1}\, h_{i-1} + h_{i-2} \quad \text{при} \quad i \geq 3.$$

Последние два значения ряда h_i определяют значения коэффициентов k и d'. Получаем

$$h_1 = 1, \quad h_2 = 2, \quad h_3 = 3, \quad h_4 = 17.$$

Получаем $d' = -17$, $k = 3$.

$$7(-17) + 3 \cdot 40 = 1.$$

Поскольку $-17 \pmod{40} \equiv 23$, то $d = 23$.

$$7 \cdot 23 \equiv 1 \pmod{40}.$$

Способ 3 Для нахождения d воспользуемся определением: если $a\,x \equiv b \pmod{p}$, то $x \equiv \left(ba^{\varphi(p)-1}\right) \bmod p$.

Подставим значения

$$d \equiv \left(7^{\varphi(40)-1}\right) \bmod 40 \equiv 7^{15} \left(\bmod 40\right) \equiv \left(7^8 \cdot 7^4 \cdot 7^2 \cdot 7\right) \bmod 40 \equiv \left(1 \cdot 1 \cdot 9 \cdot 7\right) \bmod 40 \equiv 23\,.$$

Зашифровываем сообщение M:

$$C \equiv 15^7 \pmod{55} \equiv (15^6 \cdot 15) \bmod 55.$$

Находим

$$15^2 \pmod{55} \equiv 225 \pmod{55} \equiv 5;$$
$$15^6 \pmod{55} \equiv (15^2 \bmod 55)^3 \bmod 55 \equiv 5^3 \pmod{55} \equiv 15.$$

Тогда

$$C \equiv (15 \cdot 15) \bmod 55 \equiv 5.$$

Расшифровываем сообщение C:

$$M \equiv 5^{23} \pmod{55} \equiv (5^{16} \cdot 5^4 \cdot 5^2 \cdot 5) \bmod 55 \equiv (5 \cdot 20 \cdot 25 \cdot 5) \bmod 55 \equiv 15.$$

Пример 2.6 Зашифруем аббревиатуру RSA. Для этого буквы R, S и A закодируем пятимерными двоичными векторами, воспользовавшись двоичной записью их порядковых номеров в английском алфавите: R = 18 = 10010; S = 19 = 10011; A = 1 = 00001. Теперь представим данное сообщение в виде последовательности чисел, содержащихся в интервале $\overline{0,526}$. Получаем представление

$$\text{RSA} = (100101001), (100001) = (M_1 = 297, M_2 = 33).$$

Пусть $p = 17$ и $q = 31$, вычисляем $n = 17 \cdot 31 = 527$; □ (527) = (17 –1)(31 – 1) = 480. Выбираем $e = 7$ и определяем $d = 343$. Зашифровываем сообщение M_1 и M_2:

$$C_1 \equiv 297^7 \pmod{527} \equiv 474;$$
$$C_2 \equiv 33^7 \pmod{527} \equiv 407.$$

В итоге получаем шифртекст $C = (474, 407)$.

Расшифровываем сообщение C:

$$M_1 \equiv 474^{343} \pmod{527} \equiv 297;$$
$$M_2 \equiv 407^{343} \pmod{527} \equiv 33.$$

Возвращаясь к буквенной записи, получаем после расшифровывания RSA.

Пример 2.7 Предположим, что мы решили использовать 8-битовый код ASCII для символов с размером блока, равного символу или букве. Нам потребуется $n \geq 2^8 = 256$. Пусть $p = 41$ и $q = 73$, так что $n = 41 \cdot 73 = 2993$; □ (2993) = $40 \cdot 72 = 2880$. Пусть $e = 217$, так что НОД (217, 2880) = 1. Используя алгоритм Евклида для решения сравнения $217\,d \equiv 1 \pmod{2880}$, получаем $d = 1513$. В таком случае слово MONEY, зашифрованное с использованием RSA-метода с ключом $(n, e) = (2993, 217)$, имеет вид

i	Блок	M_i	C_i
1	M	77	1537
2	O	79	79
3	N	78	1246
4	E	69	1529
5	Y	89	235

Как видно из примера, имеется число $M_2 = C_2 = 79$, не поддающееся шифровке по алгоритму RSA, что является недостатком. Количество чисел от 1 до n, не поддающихся зашифровыванию, определяется по формуле

$$N = (1 + \text{НОД}\,(d - 1, p - 1))(1 + \text{НОД}\,(d - 1, q - 1)).$$

Упражнение

Зашифровать сообщение и расшифровать криптограмму с помощью алгоритма RSA.

Сообщение $M = 15$.

№ варианта	p	q	e	№ варианта	p	q	e
1	7	7	9, 23, 26	16	7	7	9, 11, 26
2	5	11	16, 33, 35	17	5	11	16, 27, 35
3	5	13	24, 35, 36	18	5	13	24, 37, 36
4	5	17	16, 48, 49	19	5	17	16, 48, 27
5	5	19	18, 32, 49	20	5	19	18, 32, 47
6	7	11	25, 33, 37	21	7	11	23, 25, 33
7	7	13	23, 33, 38	22	7	13	23, 33, 39
8	7	17	18, 35, 36	23	7	17	18, 37, 54
9	7	19	15, 31, 48	24	7	19	15, 25, 48
10	11	13	18, 26, 113	25	11	13	18, 26, 47
11	11	17	36, 88, 131	26	11	17	36, 88, 37
12	11	19	36, 103, 123	27	11	19	36, 53, 123
13	13	17	68, 92, 133	28	13	17	68, 92, 35
14	13	19	46, 59, 96	29	13	19	46, 133, 96
15	17	19	33, 68, 131	30	17	19	33, 68, 241
Примечание: необходимо выбрать только одно число e из трех предложенных.							

2.7 Стойкость RSA

Стойкость алгоритма RSA полностью зависит от трудоемкости решения проблемы разложения на множители больших чисел. С технической точки зрения это не корректно. Утверждение, что безопасность RSA зависит от проблемы разложения на множители больших чисел, является гипотетическим. Никто и никогда не доказал математически, что для восстановления M по C и e нужно разложить n на множители. Не исключено, что может быть открыт совершенно иной способ криптоанализа RSA. Но если этот новый способ позволит криптоаналитику получить d, он также может быть использован для разложения на множители больших чисел.

Рассмотрим *"лобовой метод"* вскрытия системы RSA, который заключается в нахождении числа d, мультипликативного обратного e по модулю $\varphi(n)$. Это легко сделать, если известны числа p и q. Следовательно, решив задачу разложения на сомножители целого числа n, можно дешифровать систему RSA. Для того чтобы усложнить задачу разложения n на простые сомножители, числа p и q должны выбираться случайным образом и иметь достаточно большие значения. Кроме того, числа p и q не должны быть слишком близкими друг к другу. Но такой лобовой взлом менее эффективен, чем даже попытка разложить n на множители.

Покажем возможность использования близости значений p и q. Без ограничения общности можно считать, что $p > q$. Для величин $x = (p+q)/2$ и $y = (p-q)/2$ справедливо отношение $x^2 - y^2 = n$. Для того чтобы найти разложения n на простые сомножители, достаточно выбрать целые числа x и y. Перебирая в порядке возрастания варианты $x > \sqrt{n}$, легко найти решения, так как $x = (p+q)/2$ будет близким к $\sqrt{n}$ при условии, что p и q близки. В итоге находим: $p = x + y$, $q = x - y$.

Пример 2.8 Пусть $n = p\,q = 851$. Воспользуемся описанным выше способом, чтобы разложить n на простые сомножители p и q. Так как $\sqrt{851} \approx 29{,}17$, выбираем $x = 30$, вычисляем $30^2 - 851 = 49$ и находим решения $x = 30$, $y = 7$. Отсюда $p = 30 + 7 = 37$, $q = 30 - 7 = 23$.

Атака при использовании общего модуля. Возможна реализация RSA, в которой всем пользователям раздается одинаковый модуль n, но каждому передается отдельное значение показателей степени e и d. К сожалению, такая реализация работать не будет. Наиболее очевидная проблема заключается в следующем. Пусть одно и то же сообщение когда-нибудь зашифровывалось разными показателями (с одним и тем же модулем) и эти два показателя – взаимно простые числа (как это обычно и бывает). Тогда открытый текст может быть раскрыт даже при отсутствии каких-либо сведений об одном из ключей расшифровывания [14].

Пусть M – это открытый текст сообщения, e_1 и e_2 – два ключа зашифровывания, n – общий модуль. Шифртекстами сообщения являются:

$$C_1 \equiv M^{e_1} \pmod{n};$$
$$C_2 \equiv M^{e_2} \pmod{n}.$$

Криптоаналитику известны n, e_1, e_2, C_1 и C_2. Вот как он определяет сообщение M. Так как e_1 и e_2 – взаимно простые числа, то с помощью расширенного алгоритма Евклида можно найти r и s, для которых $r\,e_1 + s\,e_2 = 1$.

Считая s отрицательным (здесь или r или s должно быть отрицательным; положим, что отрицательным будет s), снова следует воспользоваться расширенным алгоритмом для вычисления C_2^{-1}. Затем:

$$\left(C_2^{-1}\right)^{-s} C_1^{r} \equiv M \pmod{n}.$$

Существуют два других, более тонких способа вскрытия систем такого типа. Один использует вероятностный метод для разложения n на множители. Другой является детерминированным алгоритмом вычисления секретного ключа без разложения модуля на множители [15].

Пример 2.9 Пусть $n = 49$; $e_1 = 23$; $e_2 = 11$; $C_1 = 29$ и $C_2 = 8$. Так как e_1 и e_2 – взаимно простые числа НОД (11, 23) = 1, то из уравнения $23\,r + 11\,s = 1$ с помощью расширенного алгоритма Евклида находим $r = 1$, $s = -2$ (можно $r = 12$, $s = -25$) и определяем $C_2^{-1} = 43$.

Вычисляем M из выражения $M \equiv (43^2 \cdot 29^1) \bmod 49 \equiv 15$. Можно проверить правильность определения M:

$$C_1 \equiv 15^{23} \pmod{49} \equiv 29; \quad C_2 \equiv 15^{11} \pmod{49} \equiv 8.$$

Пример 2.10 Пусть $n = 2993$; $e_1 = 217$; $e_2 = 197$; $C_1 = 235$ и $C_2 = 1200$. Так как e_1 и e_2 – взаимно простые числа, то из уравнения $217\,r + 197\,s = 1$ находим $r = 69$, $s = -76$. Определяем $C_2^{-1} = 2352$.

Вычисляем M из выражения $M \equiv (2352^{76} \cdot 235^{69}) \bmod 2993 \equiv 89$. Проверим правильность определения M:

$$C_1 \equiv 89^{217} \pmod{2993} \equiv 235; \quad C_2 \equiv 89^{197} \pmod{2993} \equiv 1200.$$

Метод бесключевого чтения RSA. Криптоаналитику известны открытый ключ (e, n) и шифртекст C. Он подбирает число i, для которого выполняется следующее соотношение: $C^{e^i} \pmod{n} \equiv C$. Далее просто проводит i раз дешифрование на открытом ключе перехваченного шифртекста (это выглядит следующим образом: $(((C^e)^e)^e \ldots)^e \bmod n \equiv C^{e^i} \pmod{n}$. Найдя такое i, криптоаналитик вычисляет $C^{e^{i-1}} \pmod{n}$ (т. е. $i - 1$ раз повторяет операцию дешифрования) – это значение и есть открытый текст M.

Пример 2.11 Пусть $n = 49$; $e = 23$; $C = 29$. Вычисляем:

$$29^{23} \pmod{49} \equiv 8;$$
$$29^{23^2} \pmod{49} \equiv 29^{529} \pmod{49} \equiv 15;$$
$$29^{23^3} \pmod{49} \equiv 29^{12167} \pmod{49} \equiv 29 = C.$$

Следовательно, $M = 15$. Проверим: $C \equiv 15^{23} \pmod{49} \equiv 29$.

Взлом RSA на основе подобранного шифртекста. Имеются методы взлома, предназначенные для вскрытия реализаций алгоритмов RSA. Они вскрывают не сам базовый алгоритм, а использующий его протокол. Важно понимать, что само по себе использование RSA не обеспечивает безопасности. Дело еще и в деталях реализации.

Рассмотрим следующий сценарий.

Сценарий: Еве удалось перехватить сообщение C Алисы, зашифрованное с помощью RSA открытым ключом Алисы. Ева хочет прочитать сообщение M. В математической форме ей нужно определить M, для которого $M \equiv C^d \pmod{n}$. Для вскрытия M она сначала выбирает случайное число x, меньшее n, и находит открытый ключ Алисы e. Затем Ева вычисляет:

$$Y \equiv x^e \pmod{n};$$
$$Z \equiv (Y\,C) \bmod n;$$
$$t \equiv x^{-1} \pmod{n}.$$

Ева просит Алису подписать Z ее закрытым ключом d (Алиса должна подписать сообщение, а не его хэш-значение). Алиса посылает Еве:

$$u \equiv Z^d \pmod{n}.$$

После этого Ева вычисляет:

$$(tu) \bmod n \equiv (x^{-1}\, Z^d) \bmod n \equiv (x^{-1}\, Y^d\, C^d) \bmod n \equiv C^d \pmod{n} \equiv M.$$

И Ева получает M.

Пример 2.12 Предположим, что шифртекст $C \equiv 1537$ зашифровывался с использованием RSA-метода с ключом $(n, e) = (2993, 217)$ (см. пример 2.7).

Криптоаналитик выбирает число $x = 207$ и вычисляет:

$$Y \equiv 207^{217} \pmod{2993} \equiv 1594;$$
$$Z \equiv (1594 \cdot 1537) \bmod 2993 \equiv 1704;$$
$$t \equiv 207^{-1} \pmod{2993} \equiv 882.$$

Затем криптоаналитик просит подписать сообщение Z закрытым ключом d и в результате получает $u \equiv 974$.

Криптоаналитик вычисляет M:

$$M \equiv (882 \cdot 974) \bmod 2993 \equiv 77.$$

На основании рассмотренных атак можно сделать следующие ограничения для алгоритма RSA:

знание одной пары секретного/открытого показателей для данного модуля позволяет криптоаналитику разложить модуль на множители;

знание одной пары секретного/открытого показателей для данного модуля позволяет криптоаналитику вычислить другие пары показателей, не разлагая модуль на множители;

в протоколах сетей связи, применяющих RSA, не должен использоваться общий модуль;

секретные показатели должны быть большими числами;

недостаточно использовать стойкий криптографический алгоритм: должны быть безопасными вся криптосистема и криптографический протокол.

Алгоритм RSA является стандартом де-факто, принятым почти во всем мире. Организация ISO разработала стандарт цифровой подписи, основанный на RSA; RSA служит информационным дополнением стандарта ISO 9796 [16]. Многие компании используют алгоритм PKCS, созданный компанией RSA Data Security, Inc. Алгоритм RSA запатентован в США. Срок действия патента истек 20 сентября 2000 года.

2.8 Алгоритм Рабина

Алгоритм Рабина [17] является модификацией алгоритма RSA. Безопасность алгоритма Рабина опирается на сложность поиска квадратных корней по модулю составного числа.

Выбираются два простых числа – p и q, – сравнимых с 3 mod 4. Эти простые числа являются закрытым ключом, а их произведение $n = p\,q$ – открытым:

$$p \equiv 3 \pmod{4} \equiv -1 \pmod{4};$$
$$q \equiv 3 \pmod{4} \equiv -1 \pmod{4}.$$

Считаем, что e фиксировано и всегда равно 2, тогда криптограмма сообщения M вычисляется как

$$C \equiv M^2 \pmod{n}.$$

Введем вспомогательные величины x и y:

$$x \equiv C^k \pmod{p};$$
$$y \equiv C^l \pmod{q},$$

где $4k = p + 1$, $4l = q + 1$.

Для x^2 и y^2 получаем

$$x^2 \equiv C^{2k} \pmod p \equiv \left[\left(M^2\right)^{\frac{p+1}{4}} \right]^2 \bmod p \equiv M^{p+1} \pmod p \equiv (M^{p-1} M^2) \bmod p \equiv M^2 \pmod p;$$

$$y^2 \equiv C^{2l} \pmod q \equiv \left[\left(M^2\right)^{\frac{q+1}{4}} \right]^2 \bmod q \equiv M^2 \pmod q.$$

Получаем четыре системы уравнений для M_1, M_2, M_3, M_4:

$$\begin{cases} M_1 \equiv x \pmod p; \\ M_1 \equiv y \pmod q; \end{cases} \quad \begin{cases} M_2 \equiv x \pmod p; \\ M_2 \equiv -y \pmod q; \end{cases} \quad \begin{cases} M_3 \equiv -x \pmod p; \\ M_3 \equiv y \pmod q; \end{cases} \quad \begin{cases} M_4 \equiv -x \pmod p; \\ M_4 \equiv -y \pmod q. \end{cases}$$

Одним из этих решений M_1, M_2, M_3 и M_4, является сообщение M. Если сообщение написано словами, выбрать правильное M несложно. С другой стороны, если сообщение является потоком случайных битов (предназначенных для генерации ключей или цифровой подписи), то определить, какое именно M является правильным, задача сложная. Одним из способов решить эту проблему служит добавление к сообщению известного заголовка, выполняемое перед шифрованием.

Пример 2.13 Дано $p = 3$; $q = 11$; $M = 8$.
Проверяем выполнение условий:

$$3 \equiv 3 \pmod 4 \equiv -1 \pmod 4;$$
$$11 \equiv 3 \pmod 4 \equiv -1 \pmod 4.$$

Определяем $n = 3 \cdot 11 = 33$.
Зашифровываем сообщение: $C \equiv 8^2 \pmod{33} \equiv 31$.
Получатель находит $k = (3 + 1) / 4 = 1$; $l = (11 + 1) / 4 = 3$ и вычисляет x и y:

$$x \equiv 31^1 \pmod 3 \equiv 1;$$
$$y \equiv 31^3 \pmod{11} \equiv 3.$$

Составляем четыре системы уравнений и определяем M_1, M_2, M_3 и M_4:

$$\begin{cases} M_1 \equiv 1 \pmod 3; \\ M_1 \equiv 3 \pmod{11}; \end{cases} \quad \begin{cases} M_2 \equiv 1 \pmod 3; \\ M_2 \equiv -3 \pmod{11}; \end{cases} \quad \begin{cases} M_3 \equiv -1 \pmod 3; \\ M_3 \equiv 3 \pmod{11}; \end{cases} \quad \begin{cases} M_4 \equiv -1 \pmod 3; \\ M_4 \equiv -3 \pmod{11}. \end{cases}$$

$$\Downarrow \qquad \Downarrow \qquad \Downarrow \qquad \Downarrow$$

$M_1 = 25$ $\qquad$ $M_2 = 19$ $\qquad$ $M_3 = 14$ $\qquad$ $\boldsymbol{M_4 = 8}$ (!)

Любое из чисел – 25, 19, 14, 8 – могло бы быть исходным сообщением, так как

$$25^2 \equiv 19^2 \equiv 14^2 \equiv 8^2 \equiv 31 \pmod{33}.$$

Пример 2.14 Дано $p = 19$, $q = 11$, $M = 16$.
Проверяем выполнения условий:

$$19 \equiv 3 \pmod 4 \equiv -1 \pmod 4;$$
$$11 \equiv 3 \pmod 4 \equiv -1 \pmod 4.$$

Определяем $n = 19 \cdot 11 = 209$.

Зашифровываем сообщение: $C \equiv 16^2 \pmod{209} \equiv 47$.

Получатель находит $k = (19 + 1) / 4 = 5$; $l = (11 + 1) / 4 = 3$ и вычисляет x и y:

$$x \equiv 47^5 \pmod{19} \equiv 16;$$
$$y \equiv 47^3 \pmod{11} \equiv 5.$$

Составляем четыре системы уравнений и определяем M_1, M_2, M_3 и M_4:

$$\begin{cases} M_1 \equiv 16 (\mathrm{mod} 19); \\ M_1 \equiv 5 (\mathrm{mod} 11); \end{cases} \quad \begin{cases} M_2 \equiv 16 (\mathrm{mod} 19); \\ M_2 \equiv -5 (\mathrm{mod} 11); \end{cases} \quad \begin{cases} M_3 \equiv -16 (\mathrm{mod} 19); \\ M_3 \equiv 5 (\mathrm{mod} 11); \end{cases} \quad \begin{cases} M_4 \equiv -16 (\mathrm{mod} 19); \\ M_4 \equiv -5 (\mathrm{mod} 11). \end{cases}$$

$$\Downarrow \qquad \Downarrow \qquad \Downarrow \qquad \Downarrow$$

$$\boldsymbol{M_1 = 16}\ (!) \qquad M_2 = 149 \qquad M_3 = 60 \qquad M_4 = 193$$

Любое из чисел – 16, 149, 60, 193 – могло бы быть исходным сообщением, так как

$$16^2 \equiv 149^2 \equiv 60^2 \equiv 193^2 \equiv 47 \pmod{209}.$$

Упражнение

Зашифровать сообщение и расшифровать криптограмму с помощью алгоритма Рабина. Согласно варианту выбрать простые числа p и q. В качестве исходного сообщения M взять дату рождения (например, 31 января – исходное сообщение $M = 3101$).

№ варианта	p	q	№ варианта	p	q
1	11, 13, 17	137, 173, 191	16	13, 71, 173	17, 97, 107
2	13, 23, 29	193, 191, 181	17	29, 79, 113	37, 73, 103
3	29, 31, 37	173, 157, 167	18	41, 83, 157	13, 97, 199
4	43, 53, 101	89, 149, 163	19	11, 61, 149	17, 53, 191
5	97, 47, 73	149, 197, 151	20	19, 73, 101	29, 37, 179
6	61, 59, 181	61, 29, 139	21	23, 101, 193	73, 167, 97
7	89, 53, 67	137, 29, 131	22	59, 37, 109	29, 113, 107
8	197, 71, 181	127, 97, 101	23	67, 113, 149	41, 127, 89
9	79, 41, 149	53, 97, 107	24	73, 71, 149	157, 131, 173
10	17, 83, 197	29, 61, 103	25	79, 13, 17	137, 149, 139
11	31, 73, 197	37, 149, 163	26	73, 83, 89	197, 173, 151
12	43, 61, 157	41, 89, 151	27	41, 61, 47	53, 163, 181
13	13, 47, 181	17, 61, 139	28	109, 173, 43	61, 167, 97
14	29, 59, 193	37, 101, 131	29	17, 31, 181	137, 73, 179
15	41, 67, 109	53, 89, 127	30	89, 23, 109	109, 29, 191

2.9 Алгоритм Вильямса

Хью Вильямс (Hugh Williams) [18] улучшил алгоритм Рабина, внеся в него изменения, устраняющие неоднозначность приема.

Выбираются два простых числа p и q так, чтобы $p \equiv 3 \pmod 4$; $q \equiv 3 \pmod 4$ и $n = p\,q$.

Кроме того, используется небольшое целое число s, для которого $J(s, n) = -1$, где $J(s, n)$ – символ Якоби.

Числа s и n передаются отправителю по незащищенному каналу. Секретным ключом является k:

$$k = \frac{1}{2}\left[\frac{1}{4}(p-1)(q-1)+1\right].$$

Для зашифровывания сообщения M вычисляется C_1, такое, что $J(M,n) = (-1)^{C_1}$, и определяются промежуточные сообщения

$$M' \equiv (s^{C_1} M) \bmod n;$$
$$C_2 \equiv M' \pmod 2.$$

Криптограмма вычисляется, как и в алгоритме Рабина:

$$C \equiv (M')^2 \bmod n.$$

Получателю передаются три числа: C, C_1, C_2.

Для расшифровывания C получатель вычисляет M'':

$$\pm M'' \equiv C^k \pmod n;$$
$$M \equiv (s^{-C_1} M'') \bmod n.$$

Правильный знак M'' определяет C_2.

Пример 2.15 Дано $p = 7$; $q = 11$; $M = 8$ и выбираем $s = 2$, для которого $J(2, 77) = -1$.

Зашифровывание сообщения:

$$J(8, 77) = -1, \text{ следовательно } C_1 = 1;$$
$$M' \equiv (2 \cdot 8) \bmod 77 \equiv 16;$$
$$C_2 \equiv 16 \pmod 2 \equiv 0;$$
$$C \equiv 16^2 \pmod{77} \equiv 25.$$

Расшифровывание сообщения:

$$k = \frac{1}{2}\left[\frac{1}{4}(7-1)(11-1)+1\right] = 8;$$
$$M'' \equiv 25^8 \pmod{77} \equiv 16;$$
$$M \equiv (2^{-1} \cdot 16) \bmod 77 \equiv 8.$$

2.10 Алгоритм Эль-Гамаля

Данный алгоритм является альтернативой алгоритму RSA и, при равном значении ключа, обеспечивает ту же криптостойкость. Стойкость алгоритма Эль-Гамаля [19, 20] основана на трудности вычисления дискретных логарифмов.

Участники информационного процесса выбирают простое число p и произвольное целое число q, являющееся первообразным корнем по модулю p (рис. 2.3).

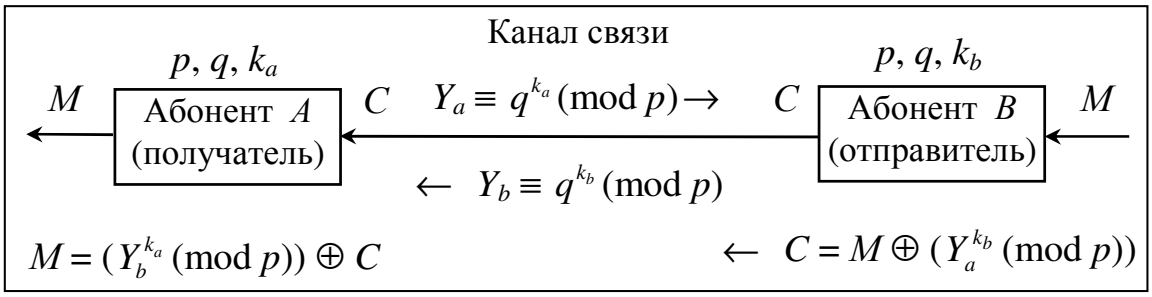

Рисунок 2.3 –Алгоритм Эль-Гамаля

Сторона A генерирует секретный ключ $k_a < p$ и вычисляет открытый ключ

$$Y_a \equiv q^{k_a} (\mathrm{mod}\ p).$$

Сторона B выбирает число $k_b < p$ и с его помощью зашифровывает передаваемое сообщение M следующим образом:

$$Y_b \equiv q^{k_b} (\mathrm{mod}\ p) \text{ и } C = M \oplus (Y_a^{k_b} (\mathrm{mod}\ p)).$$

Величина M представляет собой последовательность двоичных символов, передаваемых в канал связи. Величина $Y_a^{k_b} (\mathrm{mod}\ p)$ перед суммированием преобразуется в последовательность двоичных символов.

Сторона A, получив сообщение в виде Y_b и C, восстанавливает его:

$$M = (Y_b^{k_a} (\mathrm{mod}\ p)) \oplus C.$$

Алгоритм Эль-Гамаля – первый криптографический алгоритм с открытым ключом, используемый для шифрования сообщений и цифровых подписей, применение которого не ограничено патентами США.

Пример 2.16 Пусть $p = 11$; $q = 3$; $k_a = 7$; $k_b = 4$; $M = 6$.

Открытый ключ, посылаемый стороне B, равен

$$Y_a \equiv 3^7 (\mathrm{mod}\ 11) \equiv 2187 (\mathrm{mod}\ 11) \equiv 9.$$

Сообщение, зашифрованное на стороне B, имеет вид

$$Y_b \equiv 3^4 (\mathrm{mod}\ 11) \equiv 81 (\mathrm{mod}\ 11) \equiv 4;$$
$$Y_a^{k_b} \equiv 9^4 (\mathrm{mod}\ 11) \equiv 5;$$
$$C = 110 \oplus 101 = 011.$$

Сторона A, получив зашифрованное сообщение в виде Y_b и C, расшифровывает его:

$$Y_b^{k_a} \equiv 4^7 (\mathrm{mod}\ 11) \equiv 5;$$
$$M = 011 \oplus 101 = 110.$$

Упражнение

Зашифровать сообщение и расшифровать криптограмму с помощью алгоритма Эль-Гамаля. Сообщение $M = 14$.

№ варианта	p	q	k_a	k_b	№ варианта	p	q	k_a	k_b
1	23	11	14	21	16	23	14	17	16
2	29	11	14	21	17	29	14	17	16
3	31	11	14	21	18	31	14	17	16
4	37	11	14	21	19	37	14	17	16
5	43	11	14	21	20	43	14	17	16
6	23	12	15	19	21	23	15	18	13
7	29	12	15	19	22	29	15	18	13
8	31	12	15	19	23	31	15	18	13
9	37	12	15	19	24	37	15	18	13
10	43	12	15	19	25	43	15	18	13
11	23	13	16	18	26	23	16	21	12
12	29	13	16	18	27	29	16	21	12
13	31	13	16	18	28	31	16	21	12
14	37	13	16	18	29	37	16	21	12
15	43	13	16	18	30	43	16	21	12

2.11 Алгоритм Диффи–Хеллмана

Диффи и Хеллман предложили в 1976 году [21] для создания криптографических систем с открытым ключом алгоритм, который, так же как и алгоритм Эль-Гамаля, основан на трудности вычисления дискретного логарифма. Алгоритм Диффи–Хеллмана используется для распределения ключей (генерации секретного ключа), но его нельзя использовать для шифрования сообщения.

В соответствии с этим алгоритмом, участники информационного процесса A и B договариваются о значении большого простого числа p и простом дискретном корне этого числа a (рис. 2.4).

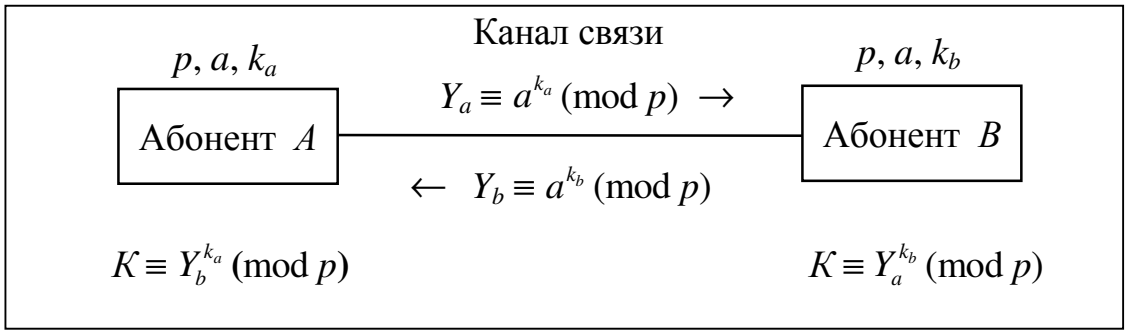

Рисунок 2.4 – Обмен ключами с использованием алгоритма Диффи–Хеллмана

При этом сторона A выбирает случайное число k_a, а сторона B – случайное число k_b таким образом, чтобы выполнялись условия

$$1 < k_a < p - 1 \quad \text{и} \quad 1 < k_b < p - 1.$$

Числа k_a и k_b держатся сторонами A и B, естественно, в секрете.

Сторона A формирует открытый ключ по правилу

$$Y_a \equiv a^{k_a} \pmod p.$$

Аналогично сторона B формирует открытый ключ по правилу

$$Y_b \equiv a^{k_b} \pmod{p}.$$

После обмена несекретными ключами Y_a и Y_b стороны вычисляют значение секретного числа K:

$$K \equiv Y_a^{k_b} \pmod{p} \equiv a^{k_a k_b} \pmod{p};$$
$$K \equiv Y_b^{k_a} \pmod{p} \equiv a^{k_b k_a} \pmod{p}.$$

Полученное число K для злоумышленника является секретным, поскольку решение уравнений Y_a и Y_b для больших чисел не представляется возможным.

Алгоритм Диффи–Хеллмана можно расширить на случай с тремя и более участниками [8].

Пример 2.17 Пусть $p = 13$; $a = 7$; $k_a = 3$; $k_b = 4$.

Открытый ключ, посылаемый стороной A,

$$Y_a \equiv 7^3 \pmod{13} \equiv 343 \pmod{13} \equiv 5.$$

Открытый ключ, посылаемый стороной B,

$$Y_b \equiv 7^4 \pmod{13} \equiv 2401 \pmod{13} \equiv 9.$$

Секретное число, вычисляемое обеими сторонами,

$$K \equiv 5^4 \pmod{13} \equiv 625 \pmod{13} \equiv 1;$$
$$K \equiv 9^3 \pmod{13} \equiv 729 \pmod{13} \equiv 1.$$

Пример 2.18 Пусть $p = 199$; $a = 50$; $k_a = 13$; $k_b = 27$.

Открытый ключ, посылаемый стороной A,

$$Y_a \equiv 50^{13} \pmod{199} \equiv (50^8 \cdot 50^4 \cdot 50) \bmod 199 \equiv (49 \cdot 7 \cdot 50) \bmod 199 \equiv 36.$$

Открытый ключ, посылаемый стороной B,

$$Y_b \equiv 50^{27} \pmod{199} \equiv (50^{16} \cdot 50^8 \cdot 50^3) \bmod 199 \equiv (13 \cdot 49 \cdot 28) \bmod 199 \equiv 125.$$

Секретное число, вычисляемое обеими сторонами,

$$K \equiv 36^{27} \pmod{199} \equiv (36^{16} \cdot 36^8 \cdot 36^3) \bmod 199 \equiv (115 \cdot 151 \cdot 90) \bmod 199 \equiv 103;$$
$$K \equiv 125^{13} \pmod{199} \equiv (125^8 \cdot 125^4 \cdot 125) \bmod 199 \equiv (63 \cdot 62 \cdot 125) \bmod 199 \equiv 103.$$

Патент на алгоритм Диффи–Хеллмана получила группа Public Key Partners (PKP). Срок действия патента истек в 1997 году. Алгоритм Диффи–Хеллмана также работает в коммутативных кольцах. Виктор Миллер (Victor Miller) и Нил Коблиц (Neal Koblitz) расширили этот алгоритм для использования с эллиптическими кривыми. Эль-Гамаль использовал основополагающую идею алгоритма Диффи–Хеллмана для разработки алгоритма шифрования и цифровой подписи.

Упражнение

Задание на генерацию ключа методом Диффи–Хеллмана. Число $p = 101$.

№ варианта	a	k_a	k_b	№ варианта	a	k_a	k_b
1	50	11	12	16	70	11	12
2	50	13	14	17	70	13	14
3	50	15	16	18	70	15	16
4	50	17	18	19	70	17	18
5	50	19	20	20	70	19	20
6	40	11	12	21	55	11	12
7	40	13	14	22	55	13	14
8	40	15	16	23	55	15	16
9	40	17	18	24	55	17	18
10	40	19	20	25	55	19	20
11	60	11	12	26	45	11	12
12	60	13	14	27	45	13	14
13	60	15	16	28	45	15	16
14	60	17	18	29	45	17	18
15	60	19	20	30	45	19	20

2.12 Криптосистемы на эллиптических кривых

2.12.1 Общие понятия

Криптосистемы на *эллиптических кривых* [22] относятся к классу криптосистем с открытым ключом. Их безопасность, как правило, основана на трудности решения задачи дискретного логарифмирования в группе точек эллиптической кривой над конечным полем. Этим и обусловлена их высокая криптостойкость по сравнению с другими алгоритмами. Существуют стойкие криптоалгоритмы на эллиптических кривых, основанные на трудности разложения больших целых чисел, когда эллиптическая кривая задается над конечным кольцом по составному модулю, но они встречаются довольно редко. Однако следует отметить, что криптостойкость является относительным понятием, связанным с понятием наилучшего известного алгоритма взлома системы.

Эллиптические кривые – математический объект, который может быть определен над любым полем. В криптографии обычно используются конечные поля. Для точек на эллиптической кривой вводится операция сложения, которая играет ту же роль, что и операция умножения в криптосистемах RSA и Эль-Гамаля.

Еще одним преимуществом криптосистем на эллиптических кривых является высокая скорость обработки информации. Но и здесь не все так просто. Понятно, что, обладая более высокой криптостойкостью, криптосистемы на эллиптических кривых позволяют использовать ключ меньшей длины. Однако приемлемая для работы в сетях скорость вычислений достигается лишь при использовании специализированных вычислителей (это вполне естественно для криптосистем с открытым ключом) и полей специальных характеристик.

Криптосистемы на эллиптических кривых, как, впрочем, и другие криптосистемы с открытым ключом, нецелесообразно применять для шифрования больших объемов данных. Но зато их можно эффективно использовать для систем цифровой подписи и ключевого обмена. С 1998 года использование эллиптических кривых для решения криптографических задач, таких как цифровая подпись, было закреплено в стандартах США ANSI X9.62 и FIPS 186–2, а в 2001 году аналогичный стандарт – ГОСТ Р34.10–2001 был принят в России. В Украине принят стандарт цифровой подписи, основанный на эллиптических кривых ДСТУ 4145–2002 (см. приложение Е). Заметим, что безопасность таких систем цифровой подписи опирается не только на стойкость алгоритма на эллиптических кривых, но и на стойкость используемой хэш-функции.

Многочисленные исследования показали, что криптосистемы на основе эллиптических кривых превосходят другие системы с открытым ключом по двум важным параметрам: степени защищенности в расчете на каждый бит ключа и быстродействию при программной и аппаратной реализации. Это объясняется тем, что для вычисления обратных функций на эллиптических кривых известны только алгоритмы с экспоненциальным ростом трудоемкости, тогда как для обычных систем предложены субэкспоненциальные методы. В результате тот уровень стойкости, который достигается, скажем, в RSA при использовании 1024-битовых модулей, в системах на эллиптических кривых реализуется при размере модуля 160 бит, что обеспечивает более простую как программную, так и аппаратную реализацию.

Детальное изучение эллиптических кривых требует знаний высшей алгебры, в особенности алгебраической геометрии. Мы, однако, постараемся изложить материал без привлечения сложных алгебраических конструкций и в объеме, достаточном для понимания принципов построения и работы соответствующих криптосистем. Более подробное изложение теории эллиптических кривых и их использования в криптографии может быть найдено в [22, 23, 24].

2.12.2 Группа точек эллиптической кривой

Эллиптические кривые (ECC – Elliptic curve cryptography) – это не эллипсы. Они так называются просто потому, что описываются кубическими уравнениями, подобными тем, которые используются для вычисления кривой эллипса. В общем случае кубические уравнения для эллиптических кривых имеют вид

$$y^2 + axy + by = x^3 + cx^2 + dx + e,$$

где a, b, c, d, и e являются действительными числами, удовлетворяющими некоторым простым условиям. Определение эллиптической кривой включает также некоторый элемент, обозначаемый O и называемый *несобственным элементом* (а также *бесконечным элементом*, или *нулевым элементом*). Такие уравнения называются *кубическими*, или *уравнениями третьего порядка*, поскольку в них наивысший показатель степени равен трем.

Рассмотрим эллиптическую кривую E (рис. 2.5), соответствующую уравнению $y^2 + y = x^3 - x^2$. На этой кривой лежат только четыре точки, координаты которых являются целыми числами. Это точки $A(0, 0)$, $B(1, -1)$, $C(1, 0)$, $D(0, -1)$.

Для определения операции сложения на группе точек эллиптической кривой будем считать, что:

на плоскости существует бесконечно удаленная точка $O \in E$, в которой сходятся все вертикальные прямые;

касательная к кривой пересекает точку касания P два раза.

Теперь можно сформулировать правила сложения точек $P, Q \in E$ (рис. 2.6):

проведем прямую линию через точки P и Q, найдем третью точку S пересечения этой прямой с кривой E;

проведем через точку S вертикальную прямую до пересечения с кривой E в точке T;

искомая сумма $P + Q = T$.

Применив эти правила к группе точек $G = \{A, B, C, D, O\}$, получим (рис. 2.7):

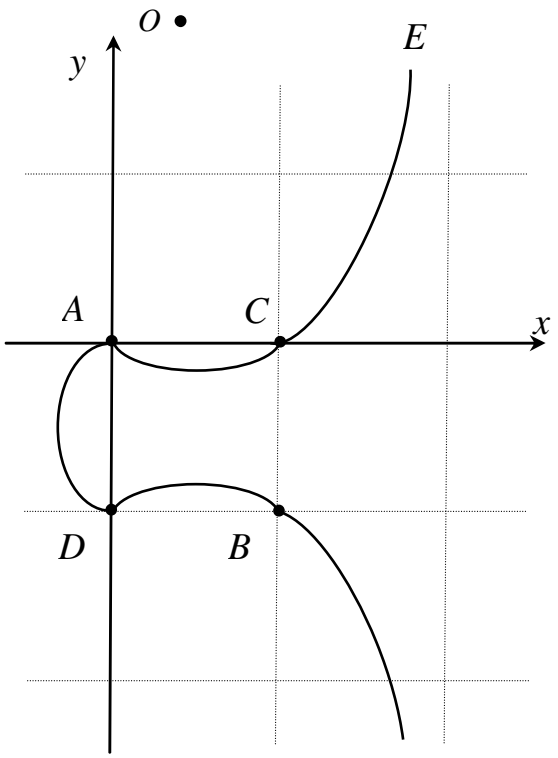

Рисунок 2.5 – Группа из пяти точек эллиптической кривой E; O – бесконечно удаленная точка

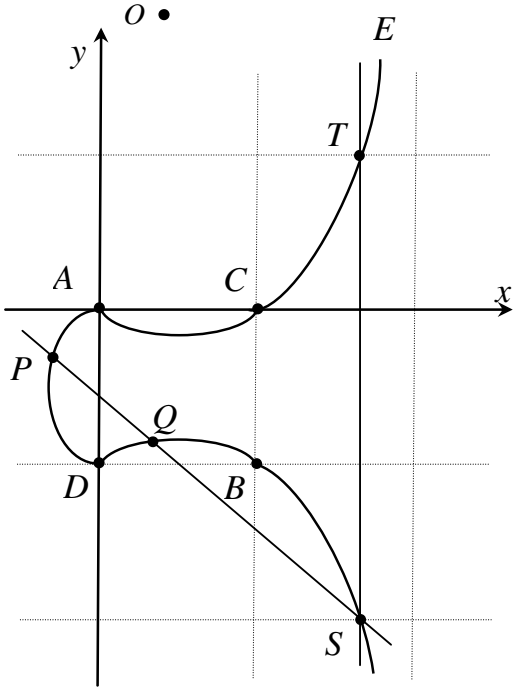

Рисунок 2.6 – Сложение точек на эллиптической кривой $P + Q = T$

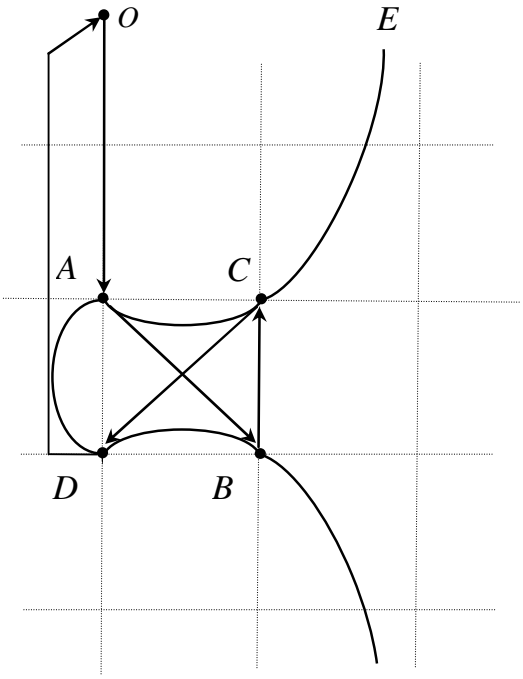

Рисунок 2.7 – Адаптивная абелева группа $\{A, B, C, D, O\}$ на эллиптической кривой E

$$A + A = B;\ \ A + B = C;\ \ A + C = D;\ \ A + D = O,$$

или

$$2A = B;\ \ 3A = C;\ \ 4A = D;\ \ 5A = O;\ \ 6A = A.$$

Для любых точек $P, Q \in G$ справедливо отношение $P + Q = Q + P$. Для любой точки $P \in G$ справедливо $P + O = P$; иначе говоря, точка O – аддитивный единичный элемент группы G.

2.12.3 Эллиптическая кривая над полем *GF*(*p*)

В реальных криптосистемах используется уравнение $y^2 \equiv x^3 + ax + b \pmod p$, где $a, b \in GF(p)$, $4a^3 + 27b^2 \pmod p \neq 0$, $p > 3$ – простое. Группа $E(GF(p))$ состоит изо всех точек (x, y); $x, y \in GF(p)$, удовлетворяющих уравнению, и бесконечно удаленной точки O.

Множество $E_p(a, b)$ состоит изо всех точек (x, y), $x \geq 0$, $p > y$, удовлетворяющих уравнению $y^2 \equiv x^3 + ax + b \pmod p$, и точки в бесконечности O. Количество точек в $E_p(a, b)$ будем обозначать $\#E_p(a, b)$. Эта величина имеет большое значение для криптографических приложений эллиптических кривых.

Определенная над точками из $E(GF(p))$ операция сложения алгебраически может быть описана следующим образом.

1 $P + O = O + P = P$.

2 Если $P = (x, y)$, то $P + (x, -y) = O$. Точка $(x, -y)$ является отрицательным значением точки P и обозначается $-P$. Заметим, что $(x, -y)$ лежит на эллиптической кривой и принадлежит $E_p(a, b)$. Например, в случае $E_{23}(1, 1)$ для $P = (13, 7)$ имеем $-P = (13, -7)$. Но $-7 \bmod 23 \equiv 16$, таким образом, $-P = (13, 16)$.

3 Если $P = (x_1, y_1)$ и $Q = (x_2, y_2)$, то $P + Q = (x_3, y_3)$ определяется в соответствии с правилами

$$x_3 \equiv \lambda^2 - x_1 - x_2 \pmod p;$$
$$y_3 \equiv \lambda\,(x_1 - x_3) - y_1 \pmod p,$$

где

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{если} \quad P \neq Q; \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{если} \quad P = Q. \end{cases}$$

Число λ – угловой коэффициент секущей, проведенной через точки $P = (x_1, y_1)$ и $Q = (x_2, y_2)$. При $P = Q$ секущая превращается в касательную, чем и объясняется наличие двух формул для вычисления λ.

Пример 2.19 Рассмотрим кривую

$$E_7(2, 6):\ y^2 \equiv x^3 + 2x + 6 \pmod 7.$$

Проверим условие:

$$(4 \cdot 2^3 + 27 \cdot 6^2) \bmod 7 \equiv 3 \neq 0\ .$$

Итак, данная кривая несингулярна. Найдем какую-нибудь (случайную) точку в $E_7(2, 6)$. Пусть $x = 5$, тогда

$$y^2 \equiv 5^3 + 2 \cdot 5 + 6 \pmod 7 \equiv (125 + 10 + 6) \bmod 7 \equiv 1 \bmod 7$$

и $y \equiv 1 \pmod 7$ или $y \equiv -1 \equiv 6 \pmod 7$. Найдены сразу две точки: (5, 1) и (5, 6).

Найдем еще пару точек путем вычисления композиции. Вначале найдем 2(5, 1).

$$\lambda = \frac{3 \cdot 5^2 + 2}{2 \cdot 1} = \frac{0}{2} \equiv 0 \pmod 7;$$
$$x_3 = 0 - 2 \cdot 5 \equiv 4 \pmod 7;$$
$$y_3 = 0\,(5 - 4) - 1 \equiv 6 \pmod 7.$$

Получили $2(5, 1) = (4, 6)$ (можно убедиться, что полученная точка лежит на кривой, подставив ее координаты в уравнение). Найдем еще одну точку $3(5, 1) = (5, 1) + (4, 6)$.

$$\lambda = \frac{6 - 1}{4 - 5} = -\frac{5}{1} \equiv 2 \pmod 7;$$
$$x_3 = 2^2 - 5 - 4 \equiv 2 \pmod 7;$$
$$y_3 = 2\,(5 - 2) - 1 \equiv 5 \pmod 7.$$

Получили $3(5, 1) = (2, 5)$.

Итак, найдены четыре точки. Для криптографического использования кривой важно знать, сколько всего точек в множестве $E_7(2, 6)$.

Пример 2.20 Пусть $p = 23$.

Рассмотрим эллиптическую кривую E: $y^2 = x^3 + x + 1$. $E_{23}(1, 1)$ состоит из точки O, а также из следующих точек: (0, 1); (0, 22); (1, 7); (1, 16); (3, 10); (3, 13); (4, 0); (5, 4); (5, 19); (6, 4); (6, 19); (7, 11); (7, 12); (9, 7); (9, 16); (11, 3); (11, 20); (12, 4); (12, 19); (13, 7); (13, 16); (17, 3); (17, 20); (18, 3); (18, 20); (19, 5); (19, 18).

Пусть $P = (3, 10)$ и $Q = (9, 7)$. Найдем $P + Q$ и $2P$. Пусть $P + Q = (x_3, y_3)$, тогда

$$\lambda = \frac{7 - 10}{9 - 3} = -\frac{1}{2} \equiv 11 \pmod{23};$$

$$x_3 = 121 - 3 - 9 = 109 = -6 \equiv 17 \pmod{23};$$
$$y_3 = 11(3 + 6) - 10 = 89 \equiv 20 \pmod{23}.$$

Таким образом, $P + Q = (17, 20)$. Найдем $2P = P + P = (x_3, y_3)$, тогда

$$\lambda = \frac{3 \cdot 9 + 1}{20} = \frac{1}{4} \equiv 6 \pmod{23};$$
$$x_3 = 36 - 6 = 30 \equiv 7 \pmod{23};$$

$$y_3 = 6(3 - 7) - 10 = -34 \equiv 12 \pmod{23}.$$

Таким образом, $2P = (7, 12)$.

Пример 2.21 Пусть $p = 5$, $a = b = 1$, $4a^3 + 27b^2 = 4 + 4 \equiv 8 \pmod 5 \neq 0$.

Рассмотрим эллиптическую кривую $y^2 = x^3 + ax + b$. $E_5(1, 1)$ состоит из следующих точек:

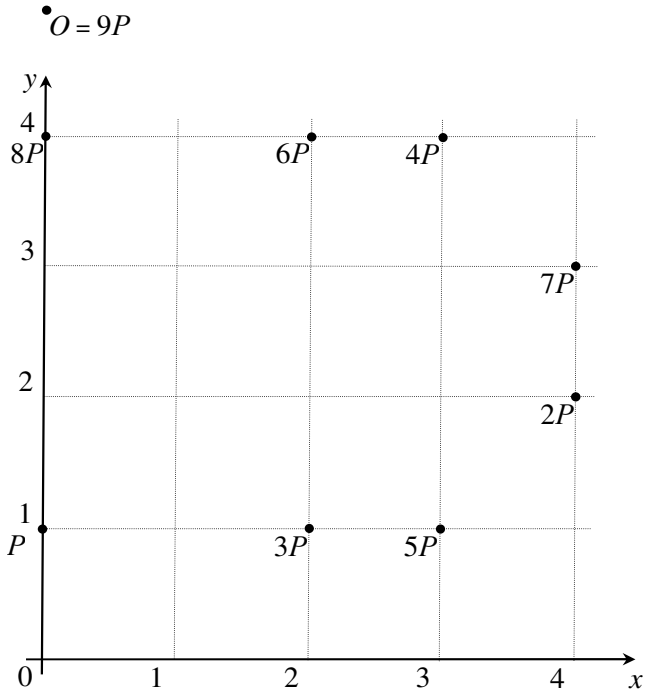

Рисунок 2.8 – Аддитивная группа $E(Z_5)$

$P = (0, 1)$, $2P = (4, 2)$, $3P = (2, 1)$, $4P = (3, 4)$, $5P = (3, 1)$, $6P = (2, 4)$, $7P = (4, 3)$, $8P = (0, 4)$, $9P = O$, при этом $10P = P$ (рис. 2.8)

2.12.4 Выбор параметров кривой

Рассмотрим основные рекомендации по выбору параметров эллиптической кривой, предназначенной для решения криптографических задач, а именно задач по выбору коэффициентов a, b и модуля p. Фактически критерием выбора служит невозможность осуществления определенного рода атак, предложенных для некоторых классов кривых. Излагаемые ниже рекомендации следуют стратегии выбора случайной кривой. Эта стратегия считается наиболее надежной с точки зрения обеспечения стойкости результирующей криптосистемы. Альтернативный подход, не рассматриваемый здесь, состоит в систематическом конструировании кривой с заданными свойствами, что обычно оказывается более эффективным с вычислительной точки зрения. Для реализации этого подхода предложены специальные методы, но получаемые кривые фактически выбираются из относительно небольшого класса и вызывают подозрения на наличие некоторых специфических свойств, которые могут позволить со временем изобрести алгоритмы для их взлома.

Опишем по шагам процесс формирования случайной кривой.

1 Выбираем случайно простое число p. Битовая длина числа p $t = \lfloor \log p \rfloor + 1$ должна быть такой, чтобы сделать невозможным применение общих методов нахождения логарифмов на кривой, имеющих трудоемкость $T(2^{t/2})$. Величина $t = 128$ бит (четыре машинных слова на 32-битовых компью-

терах) сегодня недостаточна, так как имеются сообщения о взломе соответствующих кривых. Другое рассуждение основано на том, что шифр на эллиптической кривой должен быть не менее стойким, чем блоковый шифр AES (Advanced Encryption Standard). Считается, что стойкость AES обеспечивается полной длиной его ключа, которая составляет 128, 196 или 256 бит. Так как стойкость шифра на эллиптической кривой определяется величиной $t/2$, длина модулей эллиптических кривых должна составлять соответственно 256, 392 и 512 бит.

2 Выбираем случайные числа a, и b такие, что $a, b \pmod p \neq 0$ и $4a^3 + 27b^2 \pmod p \neq 0$. Обратим внимание на то, что при вычислении композиции точек параметр b нигде не фигурирует. Поэтому для повышения эффективности счета иногда рекомендуют случайно выбирать только b, а a принимать равным небольшому целому числу. Так, стандарт США FIPS 186–2 предполагает использование кривых с параметром $a = -3$, что несколько упрощает вычисления.

3 Определяем число точек на кривой $n = \#E_p(a, b)$ (это самый трудоемкий этап описываемого процесса). Важно, чтобы n имело большой простой делитель q, а лучше всего само было простым числом, $n = q$. Если n разлагается на малые множители, то в $E_p(a, b)$ существует много малых подмножеств со своими генераторами и алгоритм Полига–Хеллмана [25] быстро вычисляет логарифм на кривой через логарифмы в этих малых подмножествах. Если поиск кривой с $n = q$ занимает слишком много времени, то можно допустить $n = h\,q$, где h – небольшое число. Еще раз следует подчеркнуть, что стойкость криптосистемы на эллиптической кривой определяется не модулем p, а числом элементов q в подмножестве точек кривой. Но если множитель h – небольшое число, то q является величиной того же порядка, что и p. Если n не соответствует предъявляемым требованиям, то необходимо вернуться к шагу 2.

4 Проверяем, выполняются ли неравенства $(p^k - 1) \bmod q \neq 0$ для всех k, $0 < k < 32$. Если нет, то возвращаемся к шагу 2. Эта проверка предотвращает возможность MOV-атаки (названной по фамилиям ее авторов Menezes, Okamoto, Vanstone), а также исключает из рассмотрения так называемые суперсингулярные кривые и кривые с $\#E_p(a, b) = p - 1$ [26, 27]. Метод MOV и указанные особые типы кривых позволяют свести задачу вычисления логарифма на кривой к более простым задачам.

5 Проверяем, выполняется ли неравенство $q \neq p$. Если нет, то возвращаемся к шагу 2. Дело в том, что для кривых с $q = p$, которые называются аномальными, существуют эффективные методы вычисления логарифмов [26, 27].

6 На данном шаге подходящая для криптографических приложений кривая получена. Мы имеем параметры p, a, b, количество точек n и размер подмножества точек q. Обычно еще требуется найти точку G – генератор этого подмножества. Если $q = n$, то любая точка (кроме O) является генератором. Если $q < n$, то выбираем случайные точки G', пока не получим $G = [n/q]G' \neq O$. Чтобы получить случайную точку на кривой, берем случайное число $x < p$, вычисляем $e \equiv (x^3 + ax + b) \bmod p$ и пытаемся извлечь квадратный корень $y = \sqrt{e} \bmod p$. Если корень существует, то получаем точку (x, y), в противном

случае пробуем другое число x. Алгоритмы вычисления квадратных корней по модулю простого числа могут быть найдены в [25].

Задача, которую решает криптоаналитик при использовании криптосистемы на базе эллиптических уравнений, называется *задачей дискретного логарифмирования на эллиптической кривой* и формулируется следующим образом. Даны точки P и Q на эллиптической кривой порядка n, где n – число точек на кривой. Необходимо найти единственную точку x, такую, что $P = xQ$.

Рассмотрим использование эллиптических кривых в криптографии.

2.12.5 Обмен ключами по схеме Диффи–Хеллмана

Обмен ключами с использованием эллиптических кривых может быть выполнен следующим образом. Сначала выбирается простое число p и параметры a и b для эллиптической кривой. Это задает эллиптическую группу точек $E_p(a, b)$. Затем в $E_p(a, b)$ выбирается *генерирующая точка* $G = (x, y)$. При выборе G важно, чтобы наименьшее значение n, при котором $nG = O$, оказалось очень большим простым числом. Параметры $E_p(a, b)$ и G криптосистемы являются параметрами, известными всем участникам. Обмен ключами между участники информационного процесса A и B можно провести по следующей схеме.

1 Сторона A выбирает целое число k_a, меньшее n. Это число будет личным ключом участника A. Затем сторона A генерирует открытый ключ $Y_a = k_a G$. Открытый ключ представляет собой некоторую точку из $E_p(a, b)$.

2 Точно так же, как и в п. 1, сторона B выбирает личный ключ k_b и вычисляет открытый ключ $Y_b = k_b G$.

3 Участник A генерирует секретный ключ $K = k_a Y_b$, а участник B генерирует секретный ключ $K = k_b Y_a$.

Две формулы в п. 3 дают один и тот же результат, поскольку

$$k_a Y_b = k_a (k_b G) = k_b (k_a G) = k_b Y_a.$$

Пример 2.22 Пусть $p = 211$; $G = (2, 2)$; $E_p(0, –4)$, что соответствует кривой $y^2 = x^3 – 4$. Можно подсчитать, что $241G = O$. Личным ключом участника A является $k_a = 121$, поэтому открытым ключом стороны A будет

$$Y_a = 121(2, 2) = (115, 48).$$

Личным ключом участника B является $k_b = 203$, поэтому открытым ключом стороны B будет

$$Y_b = 203(2, 2) = (130, 203).$$

Общим секретным ключом является

$$K = 121(130, 203) = 203(115, 48) = (161, 69).$$

Следует обратить внимание на то, что общий секретный ключ представляет собой пару чисел. Если этот ключ предполагается использовать в качестве сеансового ключа для традиционного шифрования, то из этой пары чисел необходимо генерировать одно подходящее значение. Можно, например, использовать просто координату x или некоторую простую функцию от x.

В литературе можно найти анализ нескольких подходов к зашифровыванию / расшифровыванию, предполагающих использование эллиптических кривых. Рассмотрим наиболее простой из этих подходов. Первой задачей в рассматриваемой системе является зашифровывание открытого текста сообщения M, которое будет пересылаться в виде значения $x - y$ для точки P_M. Здесь точка P_M будет представлять зашифрованный текст и впоследствии будет расшифровываться.

Сторона A выбирает личный ключ k_a и генерирует открытый ключ Y_a. Чтобы зашифровать и послать сообщение P_M пользователю B, пользователь A выбирает случайное положительное целое число r и вычисляет зашифрованный текст C_M, состоящий из пары точек $C_M = (rGP_M + rY_b)$.

Следует заметить, что сторона A использует открытый ключ Y_b стороны B. Чтобы расшифровать этот шифртекст, сторона B умножает первую точку в паре на секретный ключ B и вычитает результат из второй точки:

$$P_M + rY_b - k_b\,(rG) = P_M + r(k_bG) - k_b\,(rG) = P_M.$$

Пользователь A замаскировал сообщение P_M с помощью добавления к нему rY_b. Никто, кроме этого пользователя, не знает значения r, поэтому, хотя Y_b и является открытым ключом, никто не сможет убрать маску rY_b. Однако пользователь A включает в сообщение и "подсказку", которой будет достаточно, чтобы убрать маску тому, кто имеет личный ключ k_b. Криптоаналитику для восстановления сообщения придется вычислить r по данным G и rG, что представляется трудной задачей.

Пример 2.23 Рассмотрим случай: $p = 751$; $G = (0, 376)$; $E_p(-1, 188)$, что соответствует кривой $y^2 = x^3 - x + 188$.

Предположим, что сторона A собирается отправить стороне B сообщение, которое кодируется эллиптической точкой $P_M = (562, 201)$. Для этого участник A выбирает случайное число $r = 386$ и находит открытый ключ участника B – $Y_b = (201, 5)$. Вычисляет $386\,(0, 376) = (676, 558)$ и $(562, 201) + 386\,(201, 5) = (385, 328)$. Таким образом, участник A должен послать сообщения $\{(676, 558), (385, 328)\}$.

Безопасность, обеспечиваемая криптографическим подходом на основе эллиптических кривых, зависит от того, насколько трудной для решения оказывается задача определения r по данным rP и P. Эту задачу обычно называют *проблемой логарифмирования на эллиптической кривой*. Наиболее быстрым из известных на сегодня методов логарифмирования на эллиптический кривой является ρ-метод Полларда (Pollard) [28].

2.12.6 Протокол Месси–Омуры

Протокол Месси–Омуры позволяет передать сообщение от абонента A абоненту B по открытому каналу связи без предварительной передачи какой бы то ни было ключевой информации (рис. 2.9).

Рисунок 2.9 – Протокол Месси–Омуры

Пусть $E_p(a, b)$ – эллиптическая кривая, известная участникам информационного процесса. Абонент A выбирает ключ зашифровывания число k_a, взаимно простое с n, при котором $nG = O$, и вычисляет обратное число q_a:

$$q_a \equiv \left(k_a^{-1}\right) \bmod n.$$

Аналогично, абонент B выбирает число k_b, взаимно простое с n, и вычисляет обратное число q_b, т. е. создает свой ключ:

$$q_b \equiv \left(k_b^{-1}\right) \bmod n.$$

Абонент A помещает сообщение M в некоторую точку P_M высокого порядка эллиптической кривой и, умножив ее на свой секретный ключ k_a, получает точку

$$Y_a = k_a P_M.$$

Эту точку Y_a абонент A посылает абоненту B. Абонент B вычисляет

$$Y_b = k_b Y_a,$$

и посылает результат Y_b абоненту A, который снимает свой “замок”, вычисляя

$$C = q_a Y_b$$

и возвращает полученную точку C абоненту B.

Абонент B расшифровывает сообщение P_M, т. е. умножает полученную от абонента A точку C на свой секретный ключ:

$$P_M = q_b C.$$

Действительно, с учетом коммутативности и ассоциативности операции группы

$$P_M = q_b C = (q_a q_b) Y_b = (q_a q_b k_b) Y_a = (q_a q_b k_b k_a) P_M = (k_a q_a) (k_b q_b) P_M = P_M.$$

Сообщение M, вложенное в точку P_M, может быть использовано в качестве ключа симметричной криптосистемы. Заметим, что в данном случае не требуется опубликования никакой иной информации о параметрах протокола, кроме самой эллиптической кривой. Платой за это является необходимость трехкратной передачи по открытым каналам.

Пример 2.24 Пусть $E_p(0, -4)$; $G = (2, 2)$; $241G = O$, $n = 211$, что соответствует кривой $y^2 = x^3 - 4$. Предположим, что пользователь A собирается отправить пользователю B сообщение, которое кодируется эллиптической точкой $P_M = (208, 179)$. Для этого пользователь A выбирает число $k_a = 5$ и находит

$$q_a \equiv \left(5^{\varphi(241)-1}\right) \bmod 241 \equiv 5^{239} \left(\bmod 241\right) \equiv 193.$$

Аналогично, сторона B выбирает число $k_b = 7$ и вычисляет

$$q_b \equiv \left(7^{\varphi(241)-1}\right) \bmod 241 \equiv 7^{239} \left(\bmod 241\right) \equiv 69.$$

Согласно протоколу определяем:

$$Y_a = 5(208, 179) = (150, 85);$$
$$Y_b = 7(150, 85) = (156, 201);$$
$$C = 193(156, 201) = (120, 180);$$
$$P_M = 69(120, 180) = (208, 179).$$

Таким образом, абонент B получил передаваемое сообщение: $P_M = (208, 179)$.

2.12.7 Шифр Эль-Гамаля на эллиптической кривой

Для пользователей выбираются общая эллиптическая кривая $E_p(a, b)$ и точка G на ней такие, что G, $2G$, $3G$, ..., qG – различные точки и $qG = O$ для некоторого простого числа q.

Каждый пользователь сети выбирает число k, $0 < k < q$, которое хранит как свой секретный ключ, и вычисляет точку на кривой $Y = kG$, которая будет его открытым ключом. Параметры кривой и список открытых ключей передаются всем пользователям сети.

Допустим, пользователь A хочет передать сообщение пользователю B. Будем считать, что сообщение представлено в виде числа $M < p$.

Пользователь A выполняет следующие действия:

1 выбирает случайное число r, $0 < r < q$;

2 вычисляет $R = rG$, $P = rY_b = (x, y)$;

3 зашифровывает $C \equiv (M\,x) \bmod p$;

4 посылает пользователю B шифртекст (R, C).

Пользователь B, после получения (R, C) выполняет следующие действия:

1 вычисляет $Q = k_bR = (x, y)$;

2 расшифровывает $M \equiv (C\,x^{-1}) \bmod p$.

Предоставим обоснование протокола. Для этого достаточно показать, что

$$k_bR = k_b\,(rG) = r\,(k_bG) = rY_b,$$

т. е. $Q = P$.

Координата x точки Q остается секретной для криптоаналитика, так как он не знает числа r. Криптоаналитик может попытаться вычислить r из точки P, но для этого ему нужно решить проблему дискретного логарифмирования на кривой, что считается сложной задачей.

Наиболее вероятным вариантом использования рассмотренного алгоритма будет передача в качестве числа M секретного ключа для блочного или по-

точного шифра. В этом случае разумно выбирать параметры кривой так, чтобы $\log q$ примерно вдвое превышал длину ключа шифра.

Пример 2.25 Пусть $p = 211$; $G = (2, 2)$; $E_p(0, -4)$, что соответствует кривой $y^2 = x^3 - 4$. Можно подсчитать, что $241G = O$, тогда выбираем $r = 43$. Личным ключом пользователя B является $k_b = 91$, поэтому открытым ключом B будет

$$Y_b = 91(2, 2) = (206, 121).$$

Пользователь A хочет передать сообщение $M = 25$ пользователю B. Пользователь A вычисляет:

1 $R = 43(2, 2) = (124, 119)$;
2 $P = 43(206, 121) = (142, 15)$;
3 $C \equiv (25 \cdot 142) \bmod 211 \equiv 174$;
4 посылает пользователю B шифртекст $\{(124, 119), 174)\}$.

Пользователь B вычисляет:

1 $Q = 91(124, 119) = (142, 15)$;
2 $M \equiv (174 \cdot 142^{-1}) \bmod 211 \equiv (174 \cdot 159) \bmod 211 \equiv 25$.

Упражнения

1 Для эллиптической кривой с параметрами $p = 7$; $a = 2$; $b = 6$ вычислить следующие композиции точек: $2(2, 2)$, $2(4, 6)$, $(1, 3) + (1, 4)$, $(3, 5) + (5, 1)$.

2 Зашифровать и расшифровать с использованием алгоритма Эль-Гамаля на эллиптической кривой:

а) $p = 211$; $G = (2, 2)$; $E_p(0, -4)$; $r = 2$; $k_b = 2$; $M = 2$.
б) $p = 751$; $G = (0, 376)$; $E_p(-1, 188)$; $r = 2$; $k_b = 2$; $M = 2$.

3 ХЭШ-ФУНКЦИИ

3.1 Однонаправленные хэш-функции

Во всем многообразии проблем обеспечения информационной безопасности, решаемых при помощи криптографических методов и средств, задача обеспечения целостности и достоверности передаваемой информации представляется на сегодняшний день одной из самых острых. С учетом современных требований к информационно-телекоммуникационным системам эта задача все чаще и чаще превращается в серьезную проблему. Особенно актуальна она в финансовой сфере, поскольку для надежного функционирования платежной системы необходимым условием является сохранение всеми документами целостности и достоверности.

Неотъемлемой частью электронно-цифровой подписи является использование хэш-функций. *Хэш-функцией* (англ. *hash* – измельчать, перемешивать) называется преобразование h, превращающее информационную последовательность M произвольной длины в последовательность фиксированной длины $h(M)$, называемую *хэш-кодом*. Кроме того, хэш-функции широко применяются и для решения ряда других вопросов, связанных с обеспечением защиты потоков данных, например для хэширования паролей пользователей с целью дальнейшего их шифрования и хранения в базе данных. Данный метод применяется в ОС Windows NT (используется хэш-функция MD4 совместно с DES). Функция хэширования может служить в качестве криптографической контрольной суммы – кодом обнаружения изменений (MDC – Manipulation Detection Code) или проверки целостности сообщения (MIC – Message Integrity Check).

Одной из самых важных характеристик хэш-функций, обусловивших их широкое внедрение в практику, оказалась способность получать из открытого текста большой длины (например в хэш-функции SHA максимальная длина открытого текста ограничена 2^{64} битами) хэш-кода гораздо меньшей длины (в российском стандарте ГОСТ Р 34.11–94 длина хэш-кода составляет 256 бит, западные хэш-функции в основном имеют хэш-код длиной 160…180 бит), что в некоторых случаях позволяет достаточно эффективно сократить сетевой трафик. Применение хэш-функций дает возможность устранять избыточность открытого текста, что при дальнейшем криптографическом преобразовании хэш-кода открытого текста положительно сказывается на криптографических свойствах зашифрованного сообщения.

К функции $h(M)$ предъявляются следующие требования:

результат работы хэш-функций должен зависеть от всех двоичных символов исходного сообщения, а также от их взаимного расположения;

хэш-функция должна быть стойкой в смысле обращения;

хэш-функция должна быть стойкой в смысле обнаружения коллизий.

Область использования хэш-функций:

защита паролей при их передаче и хранении;

формирование контрольных кодов MDC;

получение сжатого образа сообщения перед формированием электронной подписи;

оперативный контроль хода программ.

Существует три метода построения хэш-функций:

на основе какой-либо трудновычисляемой математической задачи;

на основе алгоритмов блочного шифрования;

разработанные с нуля.

Каждый из указанных методов имеет свои достоинства и недостатки, однако наиболее распространенными на сегодняшний день оказались последние два. Это связано с тем, что при построении хэш-функций с нуля появляется возможность учитывать такое их свойство, как эффективная программная реализация. Широкое применение хэш-функций, построенных на основе алгоритмов блочного шифрования, является результатом тщательной проработки вопроса стойкости многих из существующих алгоритмов.

Наиболее известные алгоритмы получения хэш-образов сообщений – MD5, SHA, RIPE-MD, ГОСТ Р 34.11–94, TIGER, HAVAL.

MD5 – представитель семейства алгоритмов вычисления хэш-функций MD (Message Digest Algorithm), предложенного Р. Ривестом [29]; разработан в 1991 г.; преобразует информационную последовательность произвольной длины в хэш-образ разрядностью 128 бит.

RIPE-MD – разработан в рамках европейского проекта RIPE (Race Integrity Primitives Evaluation) Европейского сообщества; является модификацией алгоритма MD4; преобразует информационную последовательность произвольной длины в хэш-образ разрядностью 128 (RIPE-MD-128) или 160 бит (RIPE-MD-160) [30].

TIGER – разработан Р. Андерсоном и Э. Бихэмом; предназначен для реализации на 64-разрядных компьютерах; преобразует информационную последовательность произвольной длины в хэш-образ разрядностью 192 бита.

HAVAL – однонаправленная хэш-функция переменной длины. Функция HAVAL является модификацией функции MD5. Алгоритм HAVAL обрабатывает сообщение блоками размером в 1024 разряда, что в два раза больше, чем в алгоритме MD5. В HAVAL используется восемь 32-разрядных переменных сцепления, т. е. в два раза больше, чем в алгоритме MD5, и переменное число раундов обработки – от трех до пяти (на каждом раунде исполняется 16 шагов). Функция HAVAL может выдавать хэш-значения размером в 128, 160, 192, 224 или 256 разрядов [30, 36].

Рассмотрим два примера практической реализации хэш-функций: SHA, построенной с нуля, и ГОСТ Р 34.11–94 – на основе блочного алгоритма шифрования ГОСТ 28147–89.

3.2 Алгоритм стойкого хэширования SHA

Алгоритм Secure Hash Algorithm (SHA – алгоритм стойкого хэширования) является частью стандарта SHS (Secure Hash Standard), принятого в 1993 году Национальным институтом стандартов и технологий США (NIST), Агентством национальной безопасности (АНБ) США [31].

Рассмотрим версию алгоритма SHA-1, в котором осуществляется преобразование информационной последовательности произвольной длины в хэш-образ разрядностью 160 бит, называемый *сверткой сообщения* (Message Digest).

Работа алгоритма начинается с того, что входная последовательность делится на блоки по 512 бит. Перед тем как разбить ее, необходимо, чтобы длины полученных блоков в битовом выражении были равны 512 битам. Для этого к данной последовательности приписываются единица и необходимое количество нулей, чтобы ее длина стала на 64 бита меньше числа, кратного 512. Затем к последовательности приписывается 64-битовое представление длины входной последовательности. Пусть после дополнения получена информационная последовательность

$$M = m_1, m_2, \ldots, m_i, \ldots, m_n;\ i = \overline{1, n};\ |m_i| = 512.$$

Далее инициализируются пять 32-разрядных переменных:

$$A = 67452301\text{h};\ B = \text{EFCDAB89h};\ C = 98\text{BADCFEh};$$
$$D = 10325476\text{h};\ E = \text{C3D2E1F0h},$$

при этом стартовый вектор хэширования (синхропосылка) есть результат конкатенации этих переменных, т. е.

$$\text{SHA}_0 = (A, B, C, D, E).$$

На вход i-го цикла преобразования SHA_i поступает i-тый блок информационной последовательности и результат работы предыдущего цикла SHA_{i-1}, т. е.

$$\text{SHA}_i = h(m_i, \text{SHA}_{i-1}).$$

Основной цикл, совершаемый над одним 512-битовым блоком, состоит из четырех раундов, каждый из которых включает по 20 операций. Каждая операция представляет собой набор нелинейных функций от трех переменных (B, C и D) и операций циклического сдвига и суммирования. Эти функции имеют следующий вид:

1-й раунд	$f_1(B, C, D) = (B \wedge C) \vee (\overline{B} \wedge D)$	при $0 \le t \le 19$;
2-й раунд	$f_2(B, C, D) = B \oplus C \oplus D$	при $20 \le t \le 39$;
3-й раунд	$f_3(B, C, D) = (B \wedge C) \vee (B \wedge D) \vee (C \vee D)$	при $40 \le t \le 59$;
4-й раунд	$f_4(B, C, D) = B \oplus C \oplus D$	при $60 \le t \le 79$.

Как можно заметить, используются только три функции. Для $0 \le t \le 19$ функция является условным выражением: если B, то C, иначе D. Для $20 \le t \le 39$ и $60 \le t \le 79$ функция дает бит четности. Для $40 \le t \le 59$ функция является истинной, когда истинны не менее двух ее аргументов.

Для каждого раунда определяется одна константа:

$K_1 = 5\text{A827999h} = 2^{1/2}/4$	при $0 \le t \le 19$;
$K_2 = 6\text{ED9EBA1h} = 3^{1/2}/4$	при $20 \le t \le 39$;
$K_3 = 8\text{F1BBCDCh} = 5^{1/2}/4$	при $40 \le t \le 59$;
$K_4 = \text{CA62ClD6h} = 10^{1/2}/4$	при $60 \le t \le 79$,

где t – номер операции ($0 \le t \le 79$).

Логика цикла показана на рис. 3.1.

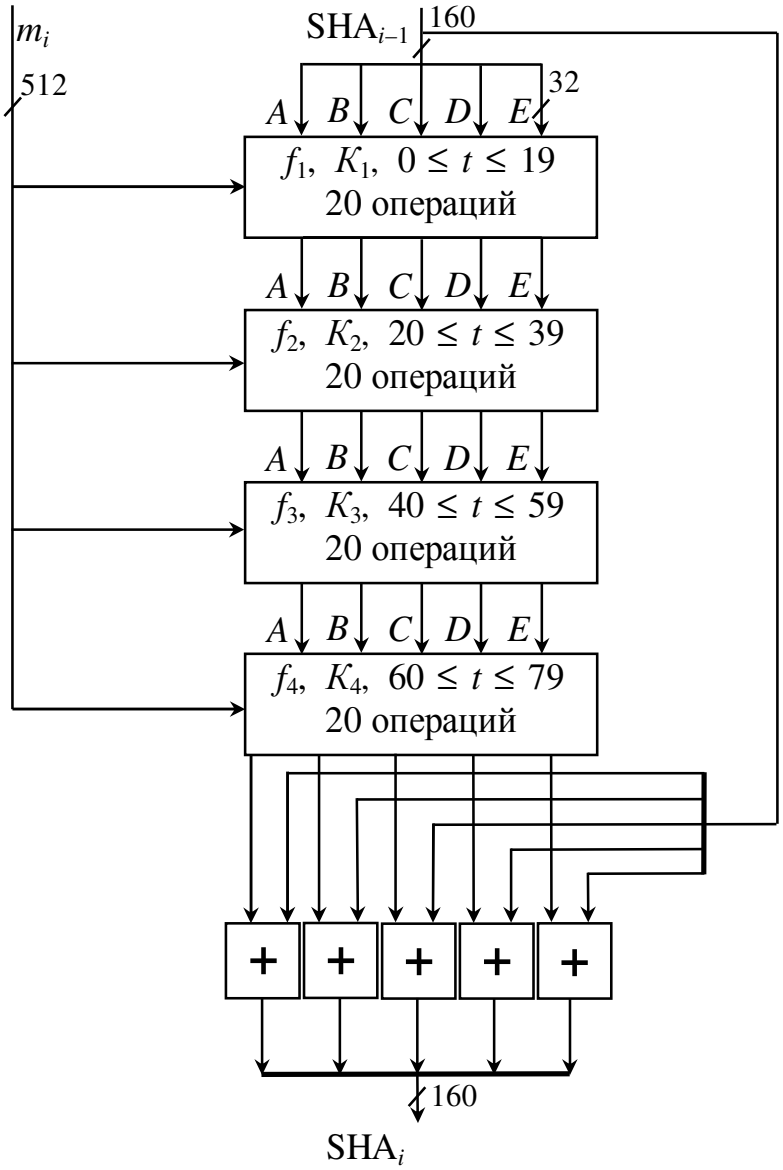

Рисунок 3.1 – Обработка одного 512-битового блока в SHA-1

Все четыре раунда имеют подобную структуру, но в каждом применяется своя логическая функция. В каждом раунде на вход подается текущий 512-битовый блок m_i и 160-битовое значение буфера *ABCDE*. Кроме того, на каждом шаге используется добавляемая к текущему значению константа K_t. Выходное значение четвертого раунда (восьмидесятый шаг) добавляется ко входному значению первого раунда, в результате чего получается SHA_i. Сложение выполняется по модулю 2^{32}. После обработки всех 512-битовых блоков на выходе получаем 160-битовый профиль сообщения.

Рассмотрим логику любой из 80-ти операций обработки одного 512-битового блока. Каждая операция имеет вид (рис. 3.2)

$$A, B, C, D, E \leftarrow (E + f_t(B, C, D) + \mathrm{Rol}^5(A) + W_t + K_t), A, \mathrm{Rol}^{30}(B), C, D;$$

где $\mathrm{Rol}^5(A)$ – циклический сдвиг влево 32-битового аргумента A на 5 бит; W_t – 32-битовое слово, извлеченное из текущего 512-битового блока ввода;

$\mathrm{Rol}^{30}(B)$ – циклический сдвиг влево 32-битового аргумента B на 30 бит; + – сложение по модулю 2^{32}.

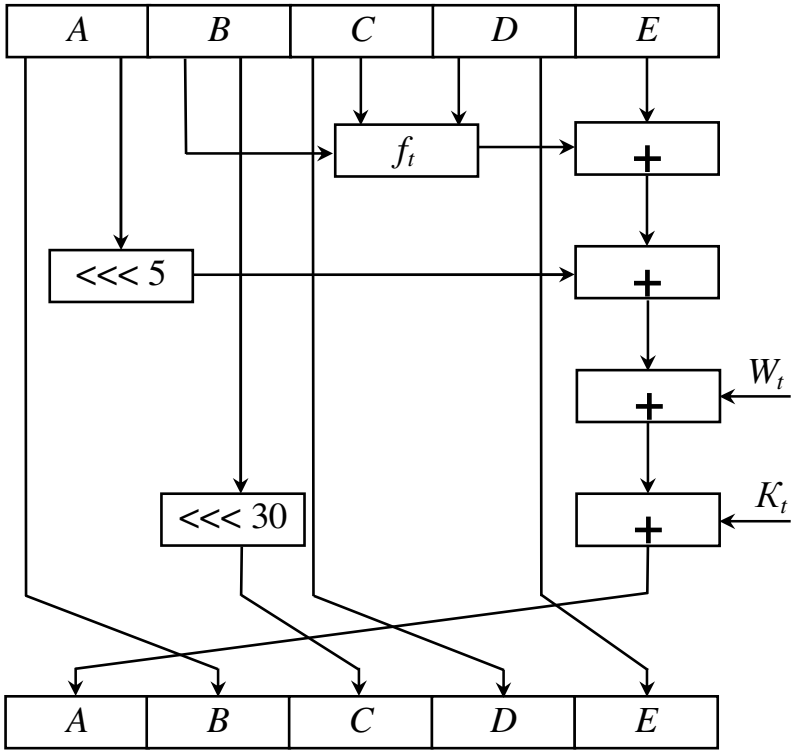

Рисунок 3.2 – Схема одной операции SHA-1

Каждая из функций f_t получает на вход три 32-битовых слова и выдает на выходе одно 32-битовое слово. Значения всех функций представлены в табл. 3.1.

Таблица 3.1 – Значение логических функций SHA-1

B	C	D	$f_{0\ldots19}$	$f_{20\ldots39}$	$f_{40\ldots59}$	$f_{60\ldots79}$
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	0	0	1	0
1	1	0	1	0	1	0
1	1	1	1	1	1	1

Осталось указать, как из 512-битового блока сообщения m_i извлекаются 32-битовые значения слов W_t. Соответствующая схема показана на рис. 3.3. Первые 16 значений W_t являются непосредственно 16-ю словами текущего блока. Остальные значения определяются формулой

$$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1 \quad \text{для} \quad 16 \le t \le 79.$$

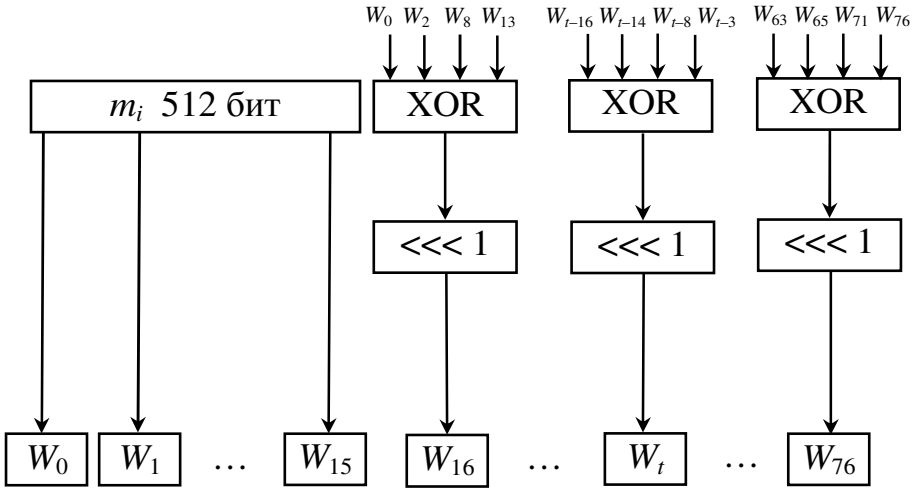

Рисунок 3.3 – Создание последовательности из 80-ти слов для обработки одного блока SHA-1

Итак, на первых 16-ти шагах обработки значение W_t равно соответствующему слову в блоке сообщения. Для остальных 64-х шагов значение W_t является результатом циклического сдвига влево на один бит результата связывания операцией XOR четырех из предшествующих значений W_t. В SHA-1 16 слов блока расширяются до 80-ти слов для использования с функцией сжатия. Это порождает немалую избыточность и взаимосвязь блоков сжимаемого сообщения, однако усложняет задачу нахождения блоков сообщения, порождающих одинаковый вывод функции сжатия.

Алгоритм SHA напоминает алгоритм MD4, но отличается наличием расширяющего преобразования, дополнительным раундом обработки, улучшенным лавинным эффектом и вычислением 160-разрядного хэш-значения. В настоящее время существуют версии алгоритма SHA-256 и SHA-512.

3.3 Функция хэширования ГОСТ Р 34.11–94

Идея использовать алгоритм блочного шифрования для построения надежных схем хэширования выглядит естественной. Однако при таком подходе возникают две проблемы. Во-первых, размер блока большинства блочных шифров недостаточен для того, чтобы хэш-функция была устойчива против метода на основе парадокса “дня рождения” (приложение Г). Во-вторых, предлагаемый метод требует задания некоторого ключа, на котором происходит шифрование. В дальнейшем этот ключ необходимо держать в секрете, ибо злоумышленник, зная его и хэш-значение, может выполнить процедуру в обратном направлении. Следующим шагом в развитии идеи использовать блочный шифр для хэширования является подход, при котором очередной блок текста подается в качестве ключа, а хэш-значение предыдущего шага – в качестве входного блока. Выход алгоритма блочного шифрования является текущим хэш-значением. Существует масса модификаций этого метода, в том числе хэш-функции, выход которых в два раза длиннее блока. В ряде модификаций промежуточное хэш-значение суммируется покоординатно по модулю два с бло-

ком текста. В этом случае подразумевается, что размер ключа и блока у шифра совпадают. В литературе встречаются 12 различных схем хэширования для случая, когда размер ключа и блока у шифра совпадают [8]:

$$H_i = E_{H_{i-1}}(M_i) \oplus M_i;$$
$$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1};$$
$$H_i = E_{H_{i-1}}(M_i) \oplus M_i \oplus H_{i-1};$$
$$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i;$$
$$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1};$$
$$H_i = E_{M_i}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1};$$
$$H_i = E_{M_i}(H_{i-1}) \oplus M_i \oplus H_{i-1};$$
$$H_i = E_{M_i}(H_{i-1} \oplus M_i) \oplus H_{i-1};$$
$$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus M_i;$$
$$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus H_{i-1};$$
$$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus H_{i-1};$$
$$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus M_i,$$

где $E_k(M)$ обозначает результат применения алгоритма блочного шифрования с ключом k к блоку. Во всех подобных схемах полагают $H_0 = I_{\text{Н}}$, где $I_{\text{н}}$ – начальное значение.

Стойкость подобных схем зависит от криптографических и иных свойств алгоритмов блочного шифрования, лежащих в их основе. В частности, даже если алгоритм шифрования является стойким, некоторые из предложенных схем могут быть подвержены коллизиям. К подобным эффектам могут приводить такие свойства алгоритма шифрования, как комплементарность (шифрование инвертированного открытого текста на инвертированном ключе приводит к инвертированному шифртексту), наличие слабых и полуслабых ключей и т. п.

В основе функции хэширования ГОСТ Р 34.11–94 [32, 44] лежит алгоритм блочного шифрования ГОСТ 28147–89 [33]. Функция преобразует информационную последовательность произвольной длины в хэш-образ разрядностью 256 бит.

Пусть M – входная информационная последовательность длиной $|m|$. Последовательность разбивается на 256-разрядные блоки. Последний неполный блок дополняется до требуемого размера. Добавляются два 256-разрядных блока, содержащих код длины последовательности (L) и контрольную сумму (I). Каждый блок m_i полученной расширенной последовательности $Ext(m)$ рассматривается как результат конкатенации четырех 64-разрядных двоичных наборов:

$$m_i = (A_i, B_i, C_i, D_i).$$

Тогда процесс вычисления хэш-образа $h(M)$ может быть описан следующим образом:

$$\text{GOST}_i = h(m_i, \text{GOST}_{i-1}),$$

где GOST_i – результат i-го цикла преобразования; GOST_0 – 256-разрядный стартовый вектор хэширования, на выбор которого ограничений не накладывается.

Процедура вычисления функции h состоит из последовательности шагов:

шаг 1:

- инициализируются переменные L (текущее значение длины обработанной части входной последовательности) и I (значение контрольной суммы);
- если длина входной необработанной последовательности больше 256 ($|M| > 256$), алгоритм переходит к шагу 3, в противном случае производятся следующие действия;

шаг 2:

- $L \equiv (L + |M|) \bmod 256$;
- $I \equiv (I + Ext(M)) \bmod 256$; $Ext(M) = (0^{256-|M|}, M)$, где $0^{256-|M|}$ – последовательность битовых нулей длиной $256 - |M|$. Таким образом, на этом этапе вычисляется текущее значение контрольной суммы;
- $\text{GOST} = h(Ext(M), \text{GOST})$ – вычисляется значение функции хэширования h от аргументов, представляющих собой хэшируемый блок и начальный вектор хэширования GOST;
- $\text{GOST} = h(L, \text{GOST})$;
- $\text{GOST} = h(I, \text{GOST})$;
- хэш-образ сообщения $h(M) = \text{GOST}$;

шаг 3:

- представим информационную последовательность M в виде $M = (M_L, M_R)$, $|M_L| = 256$;
- $\text{GOST} = h(M_L, \text{GOST})$;
- $L \equiv (L + 256) \bmod 256$;
- $\text{I} \equiv (I + M_L) \bmod 256$;
- $M = M_L$.

Алгоритм вычисления GOST_i включает в себя три шага:

генерация четырех 256-разрядных ключей k_{A_i}, k_{B_i}, k_{C_i}, k_{D_i} для зашифровывания в режиме простой замены 64-разрядных частей i-го блока;

зашифровывание частей блока m_i с использованием алгоритма ГОСТ 28147–89;

перемешивание результата зашифровывания.

Генерация секретных ключей. При генерации секретных ключей используются данные h и M (входная последовательность, представленная в двоичном виде) и инициализируются следующие константы

$$C_2 = C_4 = 0^{256}$$

и

$$C_3 = 1^8 0^8 1^{16} 0^{24} 1^{16} 0^8 (0^8 1^8)^2 1^8 0^8 (0^8 1^8)^4 (1^8 0^8)^4,$$

где a^k – двоичная последовательность из k бинарных знаков a – {0, 1}. Заданы два преобразования для 256-разрядных блоков

$$X = (A, B, C, D) = (b_{32}, \ldots, b_2, b_1),$$

где b_i – байты блока;

$$A(X) = (C \oplus D, A, B, C);$$

$$P(X) = (b_{\varphi(32)}, \ldots, b_{\varphi(2)}, b_{\varphi(1)}),$$

где $\varphi(i + 1 + 4(k - 1)) = 8i + k$,
$0 \le i \le 3,\ 1 \le k \le 8$.

Функция P представляет собой перестановку над подблоками длиной 8 бит исходной двоичной последовательности длиной 256 бит. Функция $A(X)$ преобразует эту последовательность путем разделения на четыре подблока по 64 бита каждый. Алгоритм генерации секретных ключей показан на рис. 3.4.

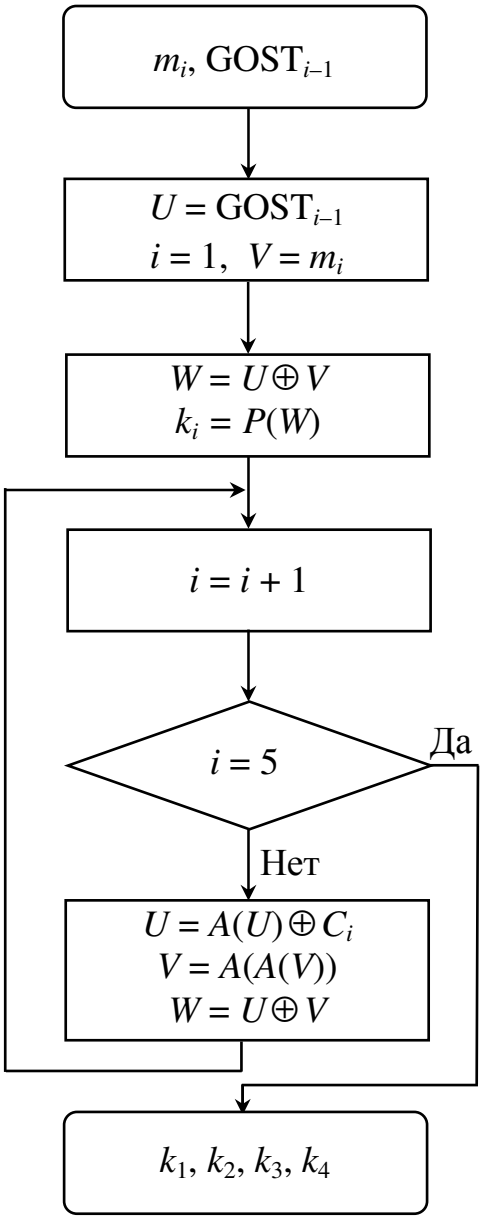

Рисунок 3.4 – Алгоритм генерации секретных ключей ГОСТ Р 34.11–94

Шифрующее преобразование. На следующем шаге части блока подвергаются зашифровыванию на полученных ключах:

$$E(m_i) = E(E_{k_{A_i}}(A_i), E_{k_{B_i}}(B_i), E_{k_{C_i}}(C_i), E_{k_{D_i}}(D_i)).$$

Перемешивающая функция. Результат цикла преобразования формируется после шага перемешивания. Задано преобразование 256-разрядного блока

$$X = (w_{16}, \ldots, w_2, w_1),$$

где w_i – 16-разрядные слова блока:

$$T(X) = (w_1 \oplus w_2 \oplus w_3 \oplus w_4 \oplus w_{13} \oplus w_{16}, w_{16}, \ldots, w_3, w_2).$$

Результат цикла преобразования:

$$\text{GOST}_i = h(m_i, \text{GOST}_{i-1}) = T^{61}(\text{GOST}_{i-1} \oplus T(m_i \oplus T^{12}(E(m_i)))),$$

где степень функции T обозначает, сколько раз она применяется к битовой последовательности.

Криптографическая стойкость данной хэш-функции основана на стойкости применяемого в ней алгоритма блочного шифрования, используемого в режиме простой замены.

При практическом использовании хэш-функции должны выполняться следующие требования:

алгоритм должен обладать высокой скоростью обработки информации (это особенно актуально для банковских приложений, где требуется особая оперативность обработки информации);

хэш-функция должна быть стойкой против атаки методом “грубой силы”;

программная реализация хэш-функции должна быть оптимизирована под использование на современной аппаратно-программной базе.

Этим требованиям должен удовлетворять как сам алгоритм выработки хэш-значения, так и хэширующая функция.

В современных условиях алгоритмическое повышение скорости выработки хэш-значения может быть достигнуто за счет применения простого преобразования, которое переводит одно сообщение в другое посредством элементарной операции, например удаления произвольного блока сообщения. Подобными преобразованиями можно также описать зависимость между двумя практически не отличающимися друг от друга сообщениями. Данный тип сообщения очень часто встречается в банковском деле, например с целью заполнения бланков платежных поручений. Отсюда следует, что для увеличения скорости обработки необходимо, чтобы алгоритм выработки хэш-значения включал в себя также алгоритм вычисления хэш-значения одного сообщения из хэш-значения другого сообщения, которое получается из начального с помощью элементарного преобразования.

3.4 Стойкость хэш-функций

С точки зрения криптографической стойкости, важным свойством хэш-функций является отсутствие коллизий, т. е. невозможность найти такие значения $x \neq y$, чтобы $h(x) = h(y)$. В криптографических приложениях важным понятием является *криптографически стойкая хэш-функция*, для которой не существует эффективного алгоритма нахождения значений $x \neq y$, где выполнялось бы условие $h(x) = h(y)$ (функция, стойкая в сильном смысле), или не существует эффективного алгоритма нахождения коллизии при заданном x такого $y \neq x$, что $h(x) = h(y)$. Р. Андерсон показал [34], что отсутствие коллизий не позволяет судить о практической стойкости хэш-функций. Иначе говоря, данное требование носит формальный характер. Практически значимым является отсутствие у хэш-функций корреляции. Свободной от корреляции называется хэш-функция, у которой невозможно найти пары таких значений $x \neq y$, что вес Хэмминга двоичного вектора $h(x)$ xor $h(y)$ будет меньше веса Хэмминга применительно к двоичному вектору $h(M)$ для некоторого малого M. Свобода от корреляции, с точки зрения криптографической стойкости, является гораздо более сильным свойством хэш-функций, чем свобода от коллизий.

4 ЭЛЕКТРОННО-ЦИФРОВАЯ ПОДПИСЬ

4.1 Общие положения

Наиболее важной областью применения криптографии с открытым ключом являются цифровые подписи. На протяжении многих веков при ведении деловой переписки, заключении контрактов и оформлении любых других важных бумаг подпись ответственного лица или исполнителя была непременным условием признания его статуса или неоспоримым свидетельством его важности. Подобный акт преследовал две цели:

гарантирование истинности письма путем сличения подписи с имеющимся образцом;

гарантирование авторства документа (с юридической точки зрения).

Выполнение данных требований основывается на следующих свойствах подписи:

подпись аутентична, т. е. с ее помощью получателю документа можно доказать, что она принадлежит подписывающему (на практике это определяется графологической экспертизой);

подпись неподделываема, т. е. служит доказательством, что только тот человек, чей автограф стоит на документе, мог подписать данный документ и никто другой не смог бы этого сделать;

подпись непереносима, т. е. является частью документа и поэтому перенести ее на другой документ невозможно;

документ с подписью является неизменяемым, т. е. после подписания его невозможно изменить, оставив данный факт незамеченным;

подпись неоспорима, т. е. человек, подписавший документ, в случае признания экспертизой, что именно он засвидетельствовал данный документ, не может оспорить факт подписания;

любое лицо, имеющее образец подписи, может удостовериться в том, что данный документ подписан владельцем подписи.

С переходом к безбумажным способам передачи и хранения данных, а также с развитием систем электронного перевода денежных средств, в основе которых – электронный аналог бумажного платежного поручения, проблема виртуального подтверждения аутентичности документа приобрела особую остроту. Развитие любых подобных систем теперь немыслимо без существования электронных подписей под электронными документами. Однако применение и широкое распространение *электронно-цифровых подписей* (ЭЦП) повлекло целый ряд правовых проблем. Так, ЭЦП может применяться на основе договоренностей внутри какой-либо группы пользователей системы передачи данных и, в соответствии с договоренностью внутри данной группы, должна иметь юридическую силу. Но будет ли электронная подпись иметь доказательную силу в суде, например при оспаривании факта передачи платежного поручения? Да, так как в 2003 году в Украине приняты законы: “Про електронні документи

та електронний документообіг", "Про електронний цифровий підпис" (приложения Б, В).

Хотя ЭЦП сохранила практически все основные свойства обычной подписи, все-таки некоторые особенности реализации электронного автографа делают ее отдельным классом подписей. Поэтому юридические, правовые и методологические аспекты применения ЭЦП должны учитывать ее специфику.

Существует несколько методов построения схем ЭЦП, а именно:

1 Шифрование *электронного документа* (ЭД) на основе симметричных алгоритмов. Данная схема предусматривает наличие в системе третьего лица (арбитра), пользующегося доверием участников обмена подписанными подобным образом электронными документами. Взаимодействие пользователей данной системой производится по следующему алгоритму:

участник A зашифровывает сообщение на своем секретном ключе k_A, знание которого разделено с арбитром, затем шифрованное сообщение передается арбитру с указанием адресата данного сообщения (информация, идентифицирующая адресата, передается также в зашифрованном виде);

арбитр расшифровывает полученное сообщение на ключе k_A, производит необходимые проверки и затем зашифровывает на секретном ключе участника B (k_B). Далее зашифрованное сообщение посылается участнику B вместе с информацией, что оно пришло от участника A;

участник B расшифровывает данное сообщение и убеждается в том, что отправителем является участник A.

Авторизацией документа в данной схеме будет считаться сам факт зашифровывания ЭД секретным ключом и передача зашифрованного ЭД арбитру. Основным преимуществом этой схемы является наличие третьей стороны, исключающее какие-либо спорные вопросы между участниками информационного обмена, т. е. в данном случае не требуется дополнительной системы арбитража ЭЦП. Недостатком схемы является наличие третьей стороны и использование симметричных алгоритмов шифрования.

2 Шифрование ЭД с использованием асимметричных алгоритмов шифрования. Фактом подписания документа в данной схеме является зашифровывание документа на секретном ключе его отправителя. Эта схема тоже используется довольно редко вследствие того, что длина ЭД может оказаться критичной. Применение асимметричных алгоритмов для зашифровывания сообщений большой длины неэффективно с точки зрения скоростных характеристик. В этом случае не требуется наличия третьей стороны, хотя она может выступать в роли сертификационного органа открытых ключей пользователей.

3 Развитием предыдущей идеи стала наиболее распространенная схема ЭЦП, а именно: зашифровывание окончательного результата обработки ЭД хэш-функцией при помощи асимметричного алгоритма. Структурная схема такого варианта построения ЭЦП представлена на рис. 4.1.

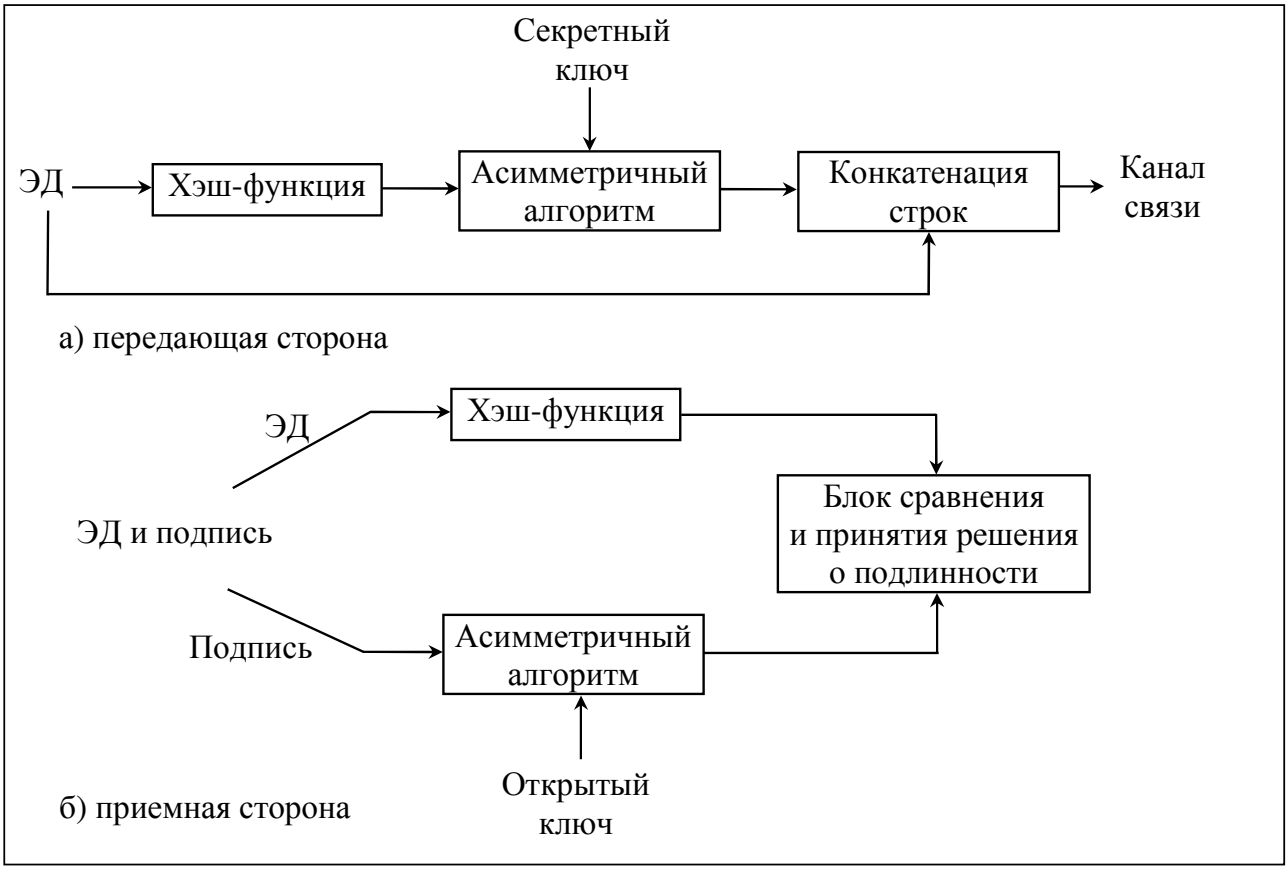

Рисунок 4.1 – Структурная схема построения ЭЦП

Генерация подписи происходит следующим образом:

1 Участник *A* вычисляет хэш-код от ЭД. Полученный хэш-код проходит процедуру преобразования с использованием своего секретного ключа, после чего полученное значение (которое и является ЭЦП) вместе с ЭД отправляется участнику *B*.

2 Участник *B* должен получить ЭД с ЭЦП и сертифицированный открытый ключ участника *A*, а затем произвести расшифровывание на нем ЭЦП, сам ЭД подвергается операции хэширования, после чего результаты сравниваются и, если они совпадают, ЭЦП признается истинной, в противном случае – ложной.

Стойкость данного типа ЭЦП основана на стойкости асимметричных алгоритмов шифрования и применяемых хэш-функций.

Кроме рассмотренных существуют “экзотические” варианты построения схем ЭЦП (групповая подпись, неоспариваемая подпись, доверенная подпись и т. п.). Появление этих разновидностей обусловлено многообразием задач, решаемых с помощью электронных технологий передачи и обработки ЭД.

В общем случае подписанный ЭД выглядит как пара, состоящая из бинарных строк (M, S), где M представляет собой ЭД, а S – решение уравнения $E_k(S) = M$, где E_k является функцией с секретом.

В связи с вышеизложенным определением ЭЦП, можно выделить следующие ее свойства:

является неподделываемой, поскольку решить уравнение $E_k(S) = M$ может только обладатель секрета k;

однозначно идентифицирует автора, т. е. человека, подписавшего данный документ;

верификация подписи производится на основе знания функции E_k;

является непереносимой на другой ЭД; исключение составляет случай, когда для используемой хэш-функции обнаружены коллизии;

ЭД с ЭЦП может передаваться по открытым каналам, поскольку любое изменение ЭД приведет к тому, что процедура проверки ЭЦП выявит данный факт.

4.2 Алгоритм цифровой подписи RSA

Технология применения электронной цифровой подписи предполагает наличие сети абонентов, посылающих друг другу электронные документы. В этой ситуации для формирования электронной подписи каждого абонента используют отдельную пару ключей – K_1 и K_2. Секретный ключ K_2 известен только пользователю, а его идентификационный номер ID и ключ K_1 помещают в общедоступный для других абонентов сети каталог. Это позволяет любому абоненту сети проверять истинность цифровой подписи документов, получаемых от ее владельца. Значение идентификационного номера используется в некоторых алгоритмах формирования сигнатуры.

Наиболее распространенной системой формирования электронной подписи является система, в основе которой лежит алгоритм RSA. Обобщенная схема формирования и проверки цифровой подписи RSA [35] показана на рис. 4.2.

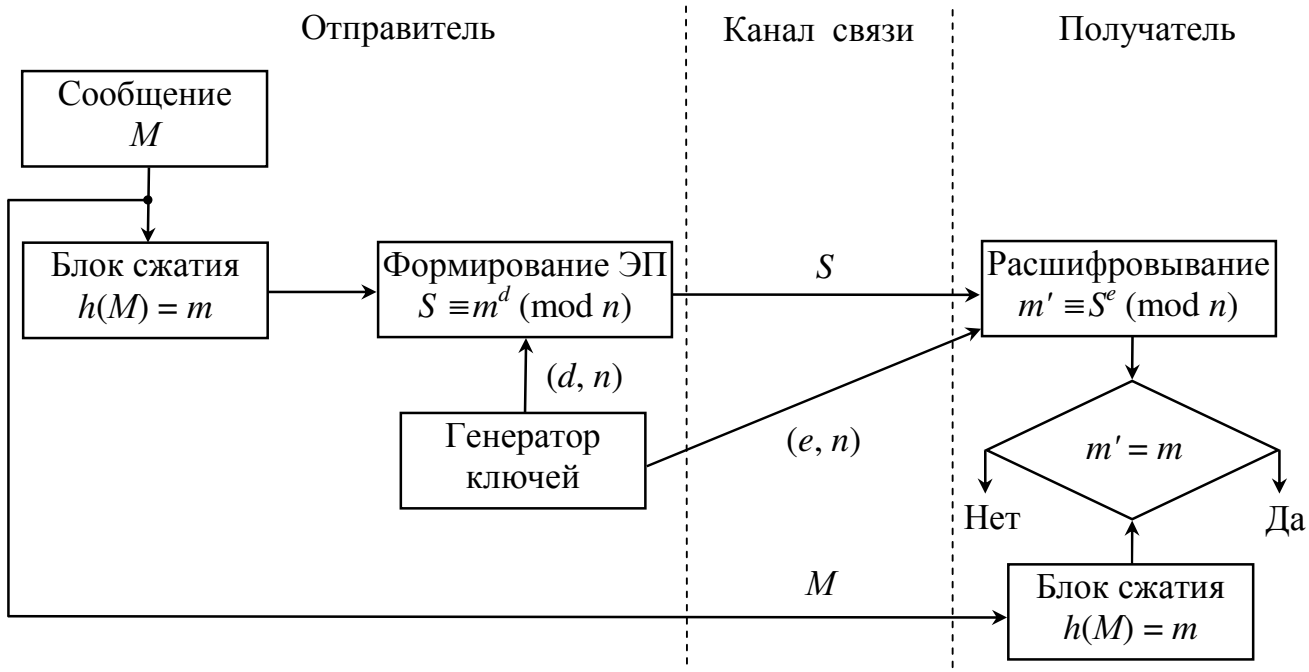

Рисунок 4.2 – Обобщенная схема цифровой подписи RSA

Выбираются большие простые числа p и q, вычисляется число $n = p\,q$, функция Эйлера $\varphi(n) = (p-1)(q-1)$ и число $e < \varphi(n)$, взаимно простое с $\varphi(n)$ (открытый ключ $K_1 = e$). Наконец, вычисляется число d (секретный ключ

$K_2 = d$), взаимно обратное с e. В открытый каталог помещают ключ (n, e), а ключ d хранится у автора документа.

Предположим, что отправитель хочет подписать сообщение M перед его отправкой. При этом предполагается, что сам текст документа шифровать не требуется. Сначала сообщение M сжимают с помощью хэш-функции h в целое число m:

$$h(M) = m.$$

Затем отправитель зашифровывает m известным только ему значением ключа d:

$$S \equiv m^d \pmod{n}.$$

Пара чисел (M, S) передается адресату как электронный документ M, подписанный электронной подписью S.

Адресат, получив подписанный документ (M, S), вычисляет значение m двумя разными способами. Во-первых, он восстанавливает хэш-значение m', применяя криптографическое преобразование подписи S с использованием открытого ключа e:

$$m' \equiv S^e \pmod{n}.$$

Во-вторых, получатель находит результат хэширования принятого сообщения M с помощью такой же хэш-функции h:

$$h(M) = m.$$

Если оба значения совпадают $m' = m$, т. е. соблюдается равенство

$$S^e \pmod{n} = h(M),$$

то получатель признает пару (M, S) подлинным значением документа.

В качестве хэш-функции в схеме подписи RSA используют функции семейства MD. Открытый ключ e называют *идентификатором* подписавшего.

Пример 4.1 Пусть $p = 5$ и $q = 11$; тогда

$$n = 11 \cdot 5 = 55, \quad \varphi(n) = (11–1)(5–1) = 40.$$

Пусть заданы ключи $e = 3$ и $d = 27$. В открытый каталог помещают значения (55, 3), а ключ d хранится у автора документа.

Отправитель хочет подписать сообщение M, для которого значение хэш-функции равно 13:

$$h(M) = 13.$$

В этом случае отправитель вычисляет

$$S \equiv 13^{27} \pmod{55} \equiv 7$$

и формирует подписанное сообщение

$$(M, 7).$$

Получатель, получив подписанное сообщение, вычисляет значение хэш-функции

$$h(M) = 13$$

и

$$m' \equiv 7^3 \pmod{55} \equiv 13.$$

Значения m' и хэш-функции $h(M)$ совпали, значит подпись верна.

Недостатки алгоритма цифровой подписи RSA:

При вычислении модуля n, ключей e и d для системы цифровой подписи RSA необходимо проверять большое количество дополнительных условий. Невыполнение любого из этих условий делает возможным фальсификацию цифровой подписи со стороны того, кто обнаружит такое невыполнение. При подписании важных документов нельзя допускать такой возможности даже теоретически.

Цифровая подпись RSA уязвима к так называемой мультипликативной атаке. Иначе говоря, алгоритм цифровой подписи RSA позволяет криптоаналитику без знания секретного ключа d сформировать подписи под теми документами, у которых результат хэширования можно вычислить как произведение результатов хэширования уже подписанных документов.

Пример 4.2 Допустим, что криптоаналитик может сконструировать три сообщения – M_1, M_2 и M_3, имеющих хэш-значения

$$m_1 = h(M_1); \; m_2 = h(M_2); \; m_3 = h(M_3),$$

причем

$$m_3 \equiv (m_1 \, m_2) \bmod n.$$

Допустим также, что для двух сообщений – M_1 и M_2 – получены законные подписи

$$S_1 \equiv m_1^{\,d} \pmod{n}$$

и

$$S_2 \equiv m_2^{\,d} \pmod{n}.$$

Тогда криптоаналитик может легко вычислить подпись S_3 для документа M_3, даже не зная секретного ключа d:

$$S_3 \equiv (S_1 \, S_2) \bmod n.$$

Действительно,

$$(S_1 \, S_2) \bmod n \equiv (m_1^{\,d} \, m_2^{\,d}) \bmod n \equiv (m_1 m_2)^d \bmod n \equiv m_3^{\,d} \pmod{n} \equiv S_3.$$

Атака на подпись RSA в схеме с нотариусом. Имеется электронный нотариус, подписывающий проходящие через него документы (однонаправленные хэш-функции не используются, нотариус шифрует своим закрытым ключом все сообщения). M – некоторый открытый текст, который нотариус не желает подписывать. Злоумышленник знает открытый ключ (e, n) нотариуса и хочет подписать документ M.

Злоумышленник выбирает некое случайное число x, взаимно простое с n, и вычисляет

$$y \equiv x^e \pmod{n},$$

затем получает значение $M' = (y\ M) \bmod n$ и передает его на подпись нотариусу. Нотариус подписывает (ведь это уже не текст M)

$$M'^{d} \pmod{n} \equiv S.$$

Злоумышленник получает

$$S \equiv M'^{d} \pmod{n} \equiv (y\ M)^d \pmod{n} = ((x^e)^d\ M^d) \bmod n = (x\ M^d) \bmod n,$$

следовательно, решает равенство

$$M^d \equiv (S\ x^{-1}) \bmod n,$$

значение которого и является подписью текста M.

Пример 4.3 Пусть $n = 2993$; $e = 217$; $M = 77$ – некоторый открытый текст, который нотариус не желает подписывать. Злоумышленник выбирает случайное число $x = 23$ и вычисляет

$$y \equiv 23^{217} \pmod{2993} \equiv 2505;$$

$$M' = (2505 \cdot 77) \bmod 2993 \equiv 1333.$$

Значение $M' = 1333$ передает на подпись нотариусу, и в результате получает $S = 2793$. Злоумышленник решает равенство

$$M^d = S \equiv (2793 \cdot 23^{-1}) \bmod 2993$$

и находит $S = 1683$, которое и является подписью $M = 77$.

4.3 Электронная подпись на базе шифра Эль-Гамаля

Пусть отправитель собирается подписать документ M. Отправитель выбирает большое простое число p и число g. Эти числа передаются или хранятся в открытом виде и могут быть общими для целой группы пользователей. Отправитель выбирает случайное число k – секретный ключ, $1 < k < p - 1$, и вычисляет

$$Y \equiv g^k \pmod{p}.$$

Число Y публикует в качестве открытого ключа. Опишем последовательность действий для построения подписи. Вначале вычисляется значение хэш-функции $h(M) = m$ и выбирается случайно число x такое, что $x < p - 1$ и взаимно простое с $p - 1$, и вычисляются числа

$$r \equiv g^x \pmod{p};$$
$$u \equiv (m - k\ r) \bmod (p - 1);$$
$$s \equiv (x^{-1}\ u) \bmod (p - 1).$$

Формируется подписанное сообщение (M, r, s).

Получатель прежде всего вычисляет значение хэш-функции $h(M) = m$ и затем проверяет подпись, используя равенство

$$Y^r r^s \equiv g^m \pmod{p}.$$

Если равенство выполняется, то подпись верна.

Пример 4.4 Пусть $p = 23$; $g = 5$; $k = 7$; $h(M) = 3$; $x = 5$.

Отправитель вычисляет открытый ключ

$$Y \equiv 5^7 \pmod{23} \equiv 17.$$

Переходит к вычислению подписи:

$$r \equiv 5^5 \pmod{23} \equiv 20;$$
$$u \equiv (3 - 7 \cdot 20) \bmod (23 - 1) \equiv 17;$$
$$s \equiv (5^{-1} \cdot 17) \bmod (23 - 1) \equiv 21.$$

Формируется подписанное сообщение в виде (M, 20, 21). Подписанное сообщение передается получателю.

Получатель проверяет подлинность подписи. Вначале он вычисляет значение хэш-функции $h(M) = 3$, а затем вычисляет

$$(17^{20} \cdot 20^{21}) \pmod{23} \equiv 10;$$
$$5^3 \pmod{23} \equiv 10.$$

Получатель делает вывод, что подпись верна.

Упражнения

Во всех задачах будем предполагать, что $h(M) = m$ для всех значений m.

1 Построить подпись RSA для сообщения m при следующих параметрах пользователя:

а) $p = 5$, $q = 11$, $d = 27$, $m = 7$;
б) $p = 5$, $q = 13$, $d = 29$, $m = 10$;
в) $p = 7$, $q = 11$, $d = 43$, $m = 5$;
г) $p = 7$, $q = 13$, $d = 29$, $m = 15$.

2 Для указанных открытых ключей пользователя RSA проверить подлинность подписанных сообщений:

а) $n = 55$, $e = 3$, (7, 28), (22, 15), (16, 36);
б) $n = 65$, $e = 5$, (6, 42), (10, 30), (6, 41);
в) $n = 77$, $e = 7$, (13, 41), (11, 28), (5, 26);
г) $n = 91$, $e = 5$, (15, 71), (11, 46), (16, 74).

3 Абоненты некой сети применяют подпись Эль-Гамаля с общими параметрами $p = 23$, $g = 5$. Для указанных секретных параметров абонентов найти открытый ключ Y и построить подпись для сообщения m:

а) $k = 1$, $x = 3$, $m = 15$;
б) $k = 10$, $x = 15$, $m = 5$;
в) $k = 3$, $x = 13$, $m = 8$;
г) $k = 18$, $x = 7$, $m = 5$.
д) $k = 9$, $x = 19$, $m = 15$.

4 Для указанных открытых ключей Y пользователей системы Эль-Гамаля с общими параметрами $p = 23$, $g = 5$ проверить подлинность подписанных сообщений:

а) $Y = 1$, (15; 20, 3), (15; 10, 5), (15; 19, 3);
б) $Y = 10$, (5; 19, 17), (7; 17, 8), (6; 17, 8);
в) $Y = 3$, (3; 17, 12), (2; 17, 12), (8; 21, 11);
г) $Y = 18$, (5; 17, 1), (5; 11, 3), (5; 17, 10);
д) $Y = 11$, (15; 7, 1), (10; 15, 3), (15; 7, 16).

4.4 Стандарт цифровой подписи DSS

4.4.1 Алгоритм цифровой подписи DSA

Национальный институт стандартов и технологии США опубликовал федеральный стандарт обработки информации FIPS PUB 186, известный также как DSS (Digital Signature Standard – стандарт цифровой подписи) [37]. Стандарт DSS основан на алгоритме хэширования SHA и представляет новую технологию использования цифровой подписи – алгоритм DSA (Digital Signature Algorithm – алгоритм цифровой подписи). Алгоритм DSA является ”классическим” примером схемы ЭЦП на основе использования хэш-функции и асимметричного алгоритма шифрования. Стандарт DSS был предложен в 1991 г., а его исправленная версия – в 1993 г. в ответ на возникшие сомнения в безопасности соответствующей схемы. В 1996 году в него были внесены незначительные изменения.

В данном стандарте подпись представляет собой два больших целых числа, полученных в соответствии с процедурами и параметрами, определенными в DSS. Стойкость системы в целом основана на сложности нахождения дискретных логарифмов в конечных полях.

В стандарте DSS используется алгоритм, призванный обеспечить только функцию цифровой подписи. В отличие от RSA, данный алгоритм не может служить для шифрования или обмена ключами.

Алгоритм цифровой подписи DSA создан с учетом трудностей вычисления дискретных логарифмов и опирается на схемы, предложенные Эль-Гамалем и Шнорром (Schnorr).

В алгоритме используются следующие три параметра, которые являются открытыми и предполагаются известными группе пользователей. Выбирается 160-битовое простое число q. Затем выбирается простое число p длиной между битами 512…1024 и q простой делитель числа $(p – 1)$. Наконец, выбирается число g вида $h^{(p-1)/q} \pmod p$, где h является целым числом между 1 и $(p – 1)$, с тем ограничением, что g должен быть больше 1. Алгоритм схематически представлен в табл. 4.1.

Таблица 4.1 – Алгоритм цифровой подписи DSA

Глобальные компоненты открытого ключа: p – простое число, $2^{L-1} < p < 2^L$, где $512 \leqq L \leqq 1024$ и L является кратным 64, т. е. длиной между битами 512 … 1024 с шагом 64 бита; q – простой делитель $(p-1)$, где $2^{159} < q < 2^{160}$, т. е. длиной 160 битов; $g \equiv h^{(p-1)/q} \pmod p$, где h является любым целым числом таким, что $1 < h < (p-1)$ и $h^{(p-1)/q} \pmod p > 1$.
Личный ключ пользователя k – случайное или псевдослучайное число, $0 < k < q$.
Открытый ключ пользователя $Y \equiv g^k \pmod p$.
Секретный номер сообщения пользователя x – случайное или псевдослучайное число, $0 < x < q$.
Создание подписи: $r \equiv (g^x \bmod p) \bmod q$; $s \equiv [x^{-1}(h(M) + k\,r)] \bmod q$. Подпись (r, s).
Проверка подписи (верификация): $w \equiv (s'^{-1}) \bmod q$; $u_1 \equiv [h(M')w] \bmod q$; $u_2 \equiv (r'\,w) \bmod q$; $v \equiv [(g^{u_1} Y^{u_2}) \bmod p] \bmod q$. *Проверка*: $v = r'$.

Параметры p, q и g публикуются для всех пользователей системы ЭД с ЭЦП. Имея эти числа, каждый пользователь выбирает личный ключ и генерирует открытый ключ. Личный ключ k должен быть числом от 1 до $(q-1)$ и должен выбираться случайным или псевдослучайным образом (держится в секрете). Открытый ключ вычисляется на основе личного ключа по формуле $Y \equiv g^k \pmod p$. Вычислить Y по имеющемуся значению k относительно просто. Однако при имеющемся значении открытого ключа задача определения значения k по значению Y оказывается трудной, поскольку для этого требуется вычислить дискретный логарифм Y по основанию g и по модулю p.

Создание подписи (рис. 4.3). Для генерации ЭЦП пользователь вычисляет две величины – r и s, являющиеся функциями компонентов открытого ключа (p, q, g), личного ключа пользователя k, хэш-кода сообщения $h(M)$ (хэш-код M вычисляется с использованием алгоритма SHA-1) и некоторого целого числа x, которое должно выбираться случайным или псевдослучайным образом и быть уникальным для каждого выполнения подписи.

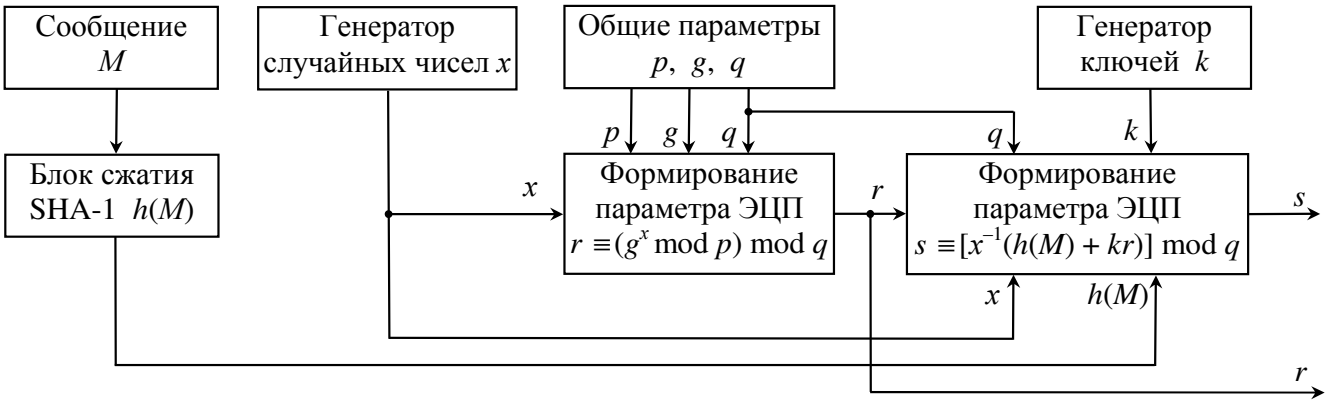

Рисунок 4.3 – Создание подписи DSA

Значения r и s являются ЭЦП сообщения M и передаются вместе с ним по открытым каналам связи.

Проверка подписи (рис. 4.4). Пусть принято сообщение M' и его значения r' и s'.

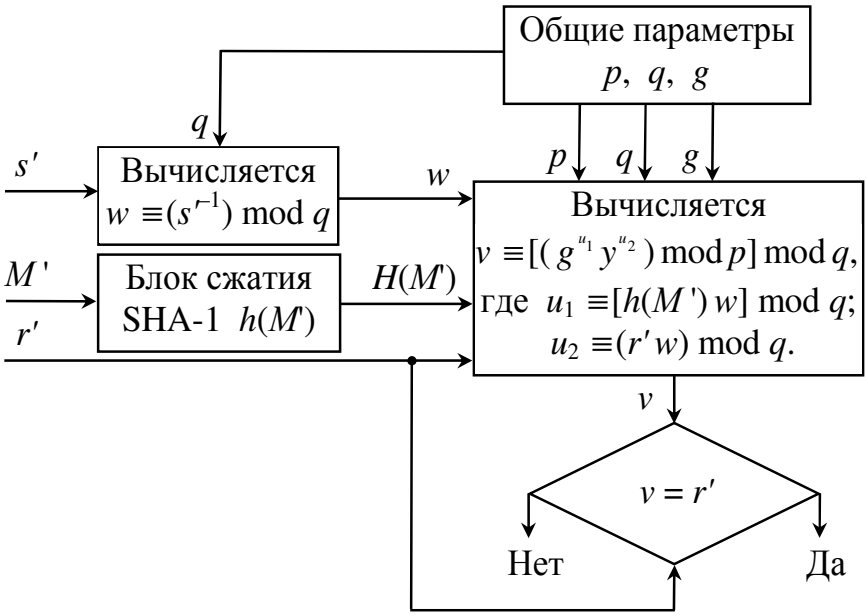

Рисунок 4.4 – Проверка подписи DSA

Получатель проверяет выполнение условий $0 < r' < q$ и $0 < s' < q$; если хотя бы одно из них нарушено, подпись отвергается. Затем вычисляется величина v, являющаяся функцией компонентов открытого ключа, открытого ключа отправителя и хэш-кода поступившего сообщения. Если эта величина соответствует компоненту r' подписи, то подпись подтверждается. Обратите внимание на то, что проверка в конце осуществляется со значением r', которое не зависит от сообщения. Обоснование равенства $v = r$ в алгоритме цифровой подписи DSA представлено в приложении Д.

Пример 4.5 Пусть $p = 211$; $q = 7$; $g = 144$; $k = 23$; $x = 3$; $h(M) = 15$. Открытый ключ пользователя

$$Y \equiv 144^{23} \pmod{211} \equiv 58.$$

Создание подписи:

$$r \equiv (144^3 \bmod 211) \bmod 7 \equiv 123 \pmod 7 \equiv 4;$$
$$s \equiv [3^{-1}(13 + 23 \cdot 4)] \bmod 7 \equiv (3^{-1} \cdot 105) \bmod 7 \equiv 3.$$

Подпись (4, 3). Проверка подписи:

$$w \equiv (3^{-1}) \bmod 7 \equiv 5;$$
$$u_1 \equiv (15 \cdot 5) \bmod 7 \equiv 5;$$
$$u_2 \equiv (4 \cdot 5) \bmod 7 \equiv 6;$$
$$v \equiv [(144^5 \cdot 58^6) \bmod 211] \bmod 7 \equiv 123 \pmod 7 \equiv 4.$$
$$v = r = 4 \text{ – подпись верна.}$$

4.4.2 Стойкость DSA

Криптографическая стойкость схемы DSA против атак методом “грубой силы” в первую очередь зависит от размера параметров p и q (в данном случае 512 и 160 бит). Соответственно криптостойкость против атаки методом “грубой силы” на параметр p будет 2^{160}. А успешная атака на параметр q возможна только в том случае, если злоумышленник сможет вычислять дискретные логарифмы в полях Галуа $GF(2^{512})$ с количеством предварительных вычислений пропорционально

$$L(p) = \mathrm{e}^{\sqrt{\ln p \ln\ln p}}.$$

Одной из теоретически возможных атак на схему DSA является компрометация параметра x. Для каждой подписи требуется новое значение x, которое должно быть выбрано случайным образом. Если злоумышленник найдет значение x, употреблявшееся при подписании сообщения (такое возможно, если будут обнаружены некоторые слабости в процедуре генерации x), секретный ключ k может быть воспроизведен. Другой возможный вариант – две подписи были сгенерированы на одном значении x. В этом случае злоумышленник тоже в состоянии восстановить k. Следовательно, одним из факторов, повышающих безопасность использования схем ЭЦП, является наличие “хорошего” генератора случайных чисел.

4.5 Стандарт электронной подписи ГОСТ Р 34.10–94

В ГОСТе Р 34.10–94 [42] цифровая подпись представляет собой два больших целых числа. Общедоступные параметры схемы ЭЦП p, q и g должны удовлетворять следующим условиям:

p – простое число, $2^{509} < p < 2^{512}$ или $2^{1020} < p < 2^{1024}$;
q – простой делитель $(p - 1)$ и $2^{254} < q < 2^{256}$;
g: $g^q \pmod p \equiv 1$ и $1 < g < p - 1$.

Секретный ключ пользователя k выбирается случайным образом и должен удовлетворять неравенству $0 < k < q$. Открытый ключ пользователя вычисляется в соответствии с равенством $Y \equiv g^k \pmod p$.

Генерация ЭЦП. Процедура генерации подписи сообщения M состоит из следующих шагов:

1 Вычисляется хэш-код сообщения M: $h(M) = m$ (хэш-функция, используемая в данном стандарте в соответствии с ГОСТ Р 34.11–94), если $h(M) (\mathrm{mod}\ p) \equiv 0$, то $h(M)$ присваивается значение $0\ldots0_{255}1$.

2 Случайным образом выбирается значение x (аналогично DSA), удовлетворяющее условию $0 < x < q$.

3 Вычисляется значение $r \equiv (g^x \bmod p) \bmod q$; если $r = 0$, следует вернуться к предыдущему этапу и выработать другое значение x.

4 Вычисляется значение $s \equiv (k\,r + x\,h(M)) \,(\mathrm{mod}\ q)$; если $s = 0$, то необходимо выработать другое значение x. В противном случае подписью сообщения M являются числа r и s.

Проверка ЭЦП. Процедура проверки ЭЦП состоит из последовательности действий:

1 Проверяется выполнение условий $0 < s < q$ и $0 < r < q$, и, если хотя бы одно из них не выполняется, подпись отвергается.

2 Вычисляется хэш-код принятого сообщения $h(M) = m$; если $h(M) (\mathrm{mod}\ p) \equiv 0$, то битовое представление $h(M)$ равно $0\ldots0_{255}1$.

3 Вычисляются значения: $z_1 \equiv (s\ h(M)^{-1}) \bmod q$; $z_2 \equiv (-r\ h(M)^{-1}) \bmod q$.

4 Вычисляется значение $u \equiv [\,g^{z_1} Y^{z_2} \,(\mathrm{mod}\ p)] \,(\mathrm{mod}\ q)$.

5 Проверяется равенство $r = u$; если оно выполняется, подпись принимается, т. е. в этом случае считается, что сообщение подписано данным отправителем и в процессе передачи не была нарушена его целостность.

З 2001 года вместо ГОСТ Р 34.10–94 применяется новый российский стандарт ГОСТ Р 34.10-2001, описывающий алгоритмы формирования и проверки электронной цифровой подписи (полное название: “ГОСТ Р 34.10–2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи”) [43].

Пример 4.6 Пусть $p = 67$; $q = 11$; $g = 25$; $k = 6$; $x = 8$; $h(M) = 3$. Открытый ключ пользователя

$$Y \equiv 25^6 \,(\mathrm{mod}\ 67) \equiv 62.$$

Создание подписи для сообщения M:

$$r \equiv (25^8 \bmod 67) \bmod 11 \equiv 24 \bmod 11 \equiv 2;$$
$$s \equiv (6 \cdot 2 + 8 \cdot 3) \bmod 11 \equiv 36 \bmod 11 \equiv 3.$$

Получаем подписанное сообщение (M, 2, 3).

Теперь выполним проверку подписи. Если сообщение не изменено, то $h(M) = 3$. Вычислим

$$z_1 \equiv (3 \cdot 3^{-1}) \bmod 11 \equiv 12 \,(\mathrm{mod}\ 11) \equiv 1;$$
$$z_2 \equiv (-2 \cdot 3^{-1}) \bmod 11 \equiv 3;$$
$$v \equiv [(25^1 \cdot 63^3) \bmod 67] \bmod 11 \equiv 24 \,(\mathrm{mod}\ 11) \equiv 2.$$

Поскольку $v = r = 2$, значит подпись верна.

Упражнения

Во всех задачах будем предполагать, что $h(M) = m$ для всех значений m.

1 Группа пользователей ГОСТа Р 34.10–94 имеет в своем распоряжении общие параметры $q = 11$; $p = 67$; $g = 25$. Вычислить открытый ключ Y и построить подпись для сообщения m при следующих секретных параметрах:

а) $k = 3$, $x = 1$, $m = 10$;
б) $k = 8$, $x = 3$, $m = 1$;
в) $k = 5$, $x = 9$, $m = 5$;
г) $k = 2$, $x = 7$, $m = 6$.

2 Для указанных открытых ключей Y пользователей ГОСТа Р 34.10–94 с общими параметрами $q = 11$, $p = 67$, $g = 25$ проверить подлинность подписанных сообщений:

а) $Y = 14$, (10; 4, 5), (10; 7, 5), (10; 3, 8);
б) $Y = 24$, (1; 3, 5), (1; 4, 3), (1; 4, 5);
в) $Y = 40$, (7; 7, 4), (7; 9, 2), (5; 9, 2);
г) $Y = 22$, (6; 9, 5), (8; 8, 3), (7; 4, 1);
д) $Y = 64$, (10; 7, 3), (7; 7, 10), (8; 7, 5).

4.6 Алгоритм электронной подписи ECDSA

Алгоритм ECDSA (Elliptic Curve Digest Signature Algorithm) принят в качестве стандартов ANSI X9.62 и IEEE P1363 [38, 39, 40]. ECDSA алгоритм с открытым ключом для создания цифровой подписи, аналогичный по своему строению DSA, но определённый в отличие от него не над полем целых чисел, а в группе точек эллиптической кривой.

Алгоритм генерации ключей:

1 Выбираем эллиптическую кривую E, определенную на Z_p. Число точек в $E(Z_p)$ должно делиться на большое целое n.

2 Выбираем точку $P \in E(Z_p)$ порядка n.

3 Выбираем случайное число $d \in [1, n-1]$.

4 Вычисляем $Q = d\mathrm{P}$.

5 Секретным ключом объявляем d, открытым – (E, P, n, Q).

Алгоритм формирования подписи под сообщением M:

1 Выбираем случайное число $k \in [1, n-1]$.

2 Вычисляем $kP = (x_1, y_1)$ и $r \equiv x_1 \pmod n$. Если $r \neq 0$, переходим к шагу 3, в противном случае – возвращаемся к шагу 1.

3 Вычисляем $k^{-1} \pmod n$.

4 Вычисляем $s \equiv [k^{-1}(h(M) + dr)] \bmod n$. Если $s \neq 0$, переходим к шагу 5, в противном случае – возвращаемся к шагу 1.

5 Подписью под сообщением M является пара целых чисел (r, s).

Примечания:

1 В качестве хэш-функции $h(M)$ на шаге 4 в стандартах ANSI X9.62 и IEEE P1363 используется SHA-1.

2 При $r = 0$ результат вычисления s не зависит от секретного ключа d.

3 При $s = 0$ необходимого для проверки подписи числа $s^{-1} \pmod n$ не существует.

Алгоритм проверки подписи (r, s) сообщения M с помощью открытого ключа (E, P, n, Q):

1 Если r и s – целые числа, принадлежащие интервалу $[1, n-1]$, переходим к шагу 2, в противном случае – результат проверки отрицательный (подпись отвергается).

2 Вычисляем $w \equiv s^{-1} \pmod n$ и $h(M)$.

3 Вычисляем $u_1 \equiv [h(M)w] \bmod n$ и $u_2 \equiv (r\,w) \bmod n$.

4 Вычисляем $u_1P + u_2Q = (x_0, y_0)$ и $v \equiv x_0 \pmod n$.

5 Подпись верна в том случае, если $v = r$.

4.7 Классификация атак на схемы электронной подписи

Стойкость схемы электронной подписи зависит от стойкости используемых криптоалгоритмов и хэш-функций и определяется относительно пары угроза–атака.

Приведем классификацию атак на схемы электронной подписи:

атака на основе известного открытого ключа (key-only attack) – самая слабая из атак, практически всегда доступная противнику;

атака на основе известных подписанных сообщений (known-message attack) – в распоряжении злоумышленника имеется некоторое (полиномиальное от k) число пар (M, S), где M – некоторое сообщение, а S – допустимая подпись для него, при этом злоумышленник не может влиять на выбор M;

простая атака с выбором подписанных сообщений (generic chosen-message attack) – злоумышленник имеет возможность выбрать некоторое количество подписанных сообщений, при этом открытый ключ он получает после данного выбора;

направленная атака с выбором сообщений (directed chosen-message attack) – выбирая подписанные сообщения, злоумышленник знает открытый ключ;

адаптивная атака с выбором сообщений (adaptive chosen-message attack) – злоумышленник знает открытый ключ; выбор каждого следующего подписанного сообщения он может делать на основе знания допустимой подписи предыдущего выбранного сообщения.

Каждая атака направлена на достижение определенной цели. Можно выделить следующие виды *угроз для схем электронной подписи* (в порядке возрастания силы):

экзистенциальная подделка (existential forgery) – создание злоумышленником подписи для какого-нибудь, возможно бессмысленного, сообщения m', отличного от перехваченного;

селективная подделка (selective forgery) – создание подписи для заранее выбранного сообщения;

универсальная подделка (universal forgery) – нахождение эффективного алгоритма формирования подписи, функционально эквивалентного *S*;

полное раскрытие (total break) – вычисление секретного ключа, возможно отличного от k^{secret}, соответствующего открытому ключу k^{public}, что дает возможность формировать подписи для любых сообщений.

Наиболее надежными являются схемы, стойкие против самой слабой из угроз на основе самой сильной из атак, т. е. против экзистенциальной подделки на основе атаки с выбором подписанных сообщений. Справедливо следующее утверждение: *схемы электронной подписи, стойкие против экзистенциальной подделки на основе атаки с выбором подписанных сообщений, существуют тогда и только тогда, когда существуют односторонние функции* [1, 28].

4.8 Особые схемы электронной подписи

В некоторых ситуациях могут потребоваться схемы электронной подписи, отличных от рассмотренных классических схем. Известны следующие специальные схемы электронной подписи:

схема подписи "вслепую", когда абонент *A* подписывает документ, не зная его содержимого (предложена Дэвидом Чаумом);

схема групповой подписи, которая позволяет верификатору убедиться в принадлежности полученного сообщения некоторой группе претендентов, но кто именно из членов группы подписал документ, верификатор определить не в состоянии;

схема разделяемой подписи, которая формируется только при участи определенного количества участников протокола, иначе говоря, данная схема является объединением классической схемы подписи и *схемы разделения секрета*;

схема конфиденциальной (неотвергаемой) подписи, которая не может быть проверена без участия сформировавшего ее участника протокола;

схема неоспаримой подписи, в которой подделка подписи может быть доказана.

4.9 Электронные деньги

Рассмотренные в предыдущих главах криптографические методы часто используются в качестве инструментов для решения других практически важных задач. Современная криптография позволяет решать проблемы, которые ранее считались в принципе неразрешимыми. Причем в настоящее время многие такие возможности криптографии используются в реальных компьютерных системах. Это и заключение коммерческих сделок в режиме удаленного взаимодействии участников, и осуществление денежных расчетов по сети, и проведение выборов по компьютерным сетям и многое другое. Обратим внимание на то, что криптографические алгоритмы не просто предоставляют новые возможности пользователю (например, не нужно ходить в банк, можно произвести все необходимые операции со своего домашнего компьютера). Важно то, что эти

алгоритмы способны обеспечивать надежность значительно более высокую, чем традиционные механизмы. Например, если бумажную банкноту можно подделать, и случаи подделок весьма многочисленны, то электронную банкноту, созданную при помощи криптографических методов, подделать практически невозможно.

Во многих странах люди оплачивают покупки при помощи электронных карточек, заказывают авиабилеты через Интернет, покупают самые разнообразные товары в Интернет-магазинах. Сведения о покупках накапливаются в магазинах и банках. Поэтому появилась новая проблема, иногда называемая “проблема Большого Брата”.

Суть проблемы состоит в том, что исчезает анонимность процесса покупки, т. е. информация о покупках любого человека может стать известной третьим лицам и использоваться против него. Например, сведения о покупке билета на поезд или самолет могут представлять интерес для преступников и т. д., и т. п. Поэтому возникла идея разработать такие схемы электронных платежей, которые сохраняли бы анонимность покупателя в той же степени, что и при расчете наличными деньгами. Такие протоколы называются *электронными*, или *цифровыми деньгами* (digital cache), что подчеркивает их основное свойство – обеспечивать ту же степень анонимности, что и обычные деньги. Некоторые схемы уже используются в реальной жизни. Описываемая ниже схема была предложена Д. Чаумом (David Chaum)[11].

Рассмотрим две плохие схемы, а затем хорошую, чтобы было легче попять суть метода.

Вначале дадим более точную постановку задачи. Имеются три участника: банк, покупатель и магазин. Покупатель и магазин имеют соответствующие счета в банке, и покупатель хочет купить товар в магазине. Покупка осуществляется в виде трехступенчатого процесса:

1) покупатель снимает нужную сумму со своего счета в банке;

2) покупатель “пересылает” деньги в магазин;

3) магазин сообщает об этом в банк, соответствующая сумма денег зачисляется на счет магазина, а покупатель забирает товар (или последний ему доставляется).

Наша цель – выбрать такую схему, чтобы она была надежна; чтобы банк не знал, кто купил товар, т. е. была сохранена анонимность обычных денег.

Опишем *первую плохую схему* (она базируется на RSA). Банк имеет следующую информацию: секретные числа p, q, d и открытые e, n.

---

[11] В 1982 году **Дэвид Чаум** основал Международную Ассоциацию Криптографических Исследований (IACR), которая в настоящее время организовывает академические конференции по исследованиям криптографии. Он внес значительный вклад в продвижение электронных денег частично в роли основателя DigiCash и электронной платежной системы eCash. Вклады Чаума в криптографию включают изобретение двух сетей анонимности. Он также изобрел нескольких важных цифровых подписей, различные методы для анонимного мандата, первые методы для анонимных цифровых сделок и цифровые деньги. Чаум является создателем системы шифрования электронного голосования.

Допустим, покупатель решил израсходовать некоторую заранее оговоренную с банком сумму (например 100 грн.). (Сначала рассмотрим случай, когда может использоваться “банкнота” только одного номинала скажем, 100 грн.) Покупатель высылает в банк число x, которое будет номером банкноты (обычно требуется, чтобы генерировалось случайное число в промежутке $(2, n-1)$). Банк вычисляет число

$$s \equiv x^d \pmod{n}$$

и формирует банкноту (x, s), которую возвращает покупателю, предварительно уменьшив его счет на 100 грн. Параметр s в банкноте – это подпись банка. Никто не может подделать подпись, так как число d секретно.

Покупатель предъявляет банкноту (x, s) в магазине, чтобы купить товар. Магазин отправляет эту банкноту в банк для проверки. Прежде всего банк проверяет правильность подписи (эту проверку мог бы сделать и магазин, используя открытые ключи банка). Но, кроме этого, банк хранит все номера возвратившихся к нему банкнот и проверяет, нет ли числа x в этом списке. Если x есть в списке, то платеж не принимается (кто-то пытается использовать банкноту повторно), и банк сообщает об этом магазину. Если же все проверки прошли успешно, то банк добавляет 100 грн. на счет магазина, а магазин отпускает товар покупателю.

Недостаток этой схемы – отсутствует анонимность. Банк, а также все, кто имеет доступ к открытым линиям связи, могут запомнить, какому покупателю соответствует число x, и тем самым выяснить, кто купил товар.

Рассмотрим *вторую плохую схему*, которая уже обеспечивает анонимность. Эта схема базируется на так называемой “слепой” подписи.

Снова покупатель хочет купить товар. Он генерирует число x, которое теперь не будет посылаться в банк. Затем он генерирует случайное число r, взаимно простое с n, и вычисляет число

$$g \equiv (x\, r^e) \bmod n.$$

Число g покупатель отправляет в банк. Банк вычисляет число

$$s' \equiv g^d \pmod{n}$$

и отправляет s' обратно покупателю (не забыв при этом снять 100 грн. с его счета).

Покупатель вычисляет

$$s \equiv (s'\, r^{-1}) \bmod n,$$

т. е. получена подпись банка к x, но самого числа x ни банк, ни кто-либо другой не видел. Вычисление s' называется “слепой подписью”, так как реальное сообщение x подписывающий не видит и узнать не может. Таким образом, покупатель имеет число x, которое никому не известно и никогда не передавалось по каналам связи. Покупатель формирует банкноту (x, s) и действует так же, как в первой плохой схеме. Но теперь никто не знает, кому принадлежит эта банкнота, т. е. она стала анонимной, как обычная бумажная банкнота.

Действия магазина и банка после предъявления покупателем банкноты (x, s) ничем не отличаются от действий, описанных в первой схеме.

Почему же данная схема плоха? Она имеет недостаток: можно сфабриковать фальшивую банкноту, если известны хотя бы две настоящие. Делается это так. Путь злоумышленник (будь то покупатель или магазин) имеет две настоящие банкноты – (x_1, s_1) и (x_2, s_2). Тогда он легко сможет изготовить фальшивую банкноту (x_3, s_3), вычислив числа

$$x_3 \equiv (x_1\, x_2) \bmod n;$$
$$s_3 \equiv (s_1\, s_2) \bmod n.$$

Действительно,

$$x_3^d \equiv (x_1\, x_2)^d \bmod n \equiv (\, x_1^d\, x_2^d\,) \bmod n \equiv (s_1\, s_2) \bmod n \equiv s_3,$$

т. е. s_3 является правильной подписью для x_3, и у банка нет никаких оснований, чтобы не принять эту фальшивую банкноту (он просто не сможет отличить ее от подлинной). Это так называемое мультипликативное свойство системы RSA.

Опишем, наконец, *хорошую схему*, в которой устранены все недостатки первых двух. В одном варианте такой схемы используется некоторая односторонняя функция $F(x)$. Функция F не секретна и известна всем (покупателю, банку и магазину).

Банкнота теперь определяется как пара чисел (x, s_F), где

$$s_F \equiv [F(x)]^d \bmod n,$$

т. е. подписывается не x, а значение $F(x)$.

Покупатель генерирует x (никому его не показывая), вычисляет $F(x)$, подписывает в банке при помощи "слепой" подписи число $F(x)$ и формирует банкноту (x, s_F). Эта банкнота обладает всеми хорошими свойствами, как и во второй схеме, в то же время подделать такую банкноту невозможно, так как невозможно вычислить обратную функцию. Для проверки подписи (т. е. подлинности банкноты) нужно вычислить $F(x)$ и убедиться, что

$$s_F^e \,(\bmod n) \equiv F(x).$$

Заметим, что при выборе односторонней функции нужно проявлять осторожность. Например, функция $F(x) \equiv a^2 \bmod n$ не годится для рассматриваемого протокола. На практике в качестве $F(x)$ обычно используются криптографические хэш-функции, описанные в главе 3. Все остальные действия магазина и банка остаются такими же, как и в ранее описанных схемах.

Есть еще один, более простой, способ борьбы с мультипликативным свойством системы RSA – внесение избыточности в сообщение. Допустим, что длина модуля n – 1024 бита. Такой же может быть и длина числа x. Будем записывать (случайно выбираемый) номер банкноты только в младшие 512 бит x, а в старшие 512 бит x запишем некоторое фиксированное число. Это фиксированное число может нести полезную информацию, такую как номинал банкноты и наименование банка. Теперь банк при предъявлении ему банкноты будет обяза-

тельно проверять наличие фиксированного заголовка в параметре x и отвергать банкноту в случае его отсутствия. Вероятность того, что при перемножении двух чисел по модулю n результат совпадет с ними в 512-ти битах, пренебрежимо мала. Поэтому получить фальшивую банкноту по формуле не удастся.

Пример 4.7 Пусть в качестве секретных параметров банка выбраны числа $p = 17$; $q = 7$; $d = 77$. Соответствующие им открытые параметры $n = 119$; $e = 5$. Для исключения возможности подделки банкнот их допустимыми номерами считаются только числа, состоящие из двух одинаковых десятичных цифр, например 11, 77, 99. Когда покупатель хочет получить банкноту, он вначале случайным образом выбирает ее номер из числа допустимых. Предположим, покупатель выбрал $x = 33$. Затем находит случайное число $r = 67$, взаимно простое с n (НОД (67, 119) = 1). Далее покупатель вычисляет

$$g \equiv (33 \cdot 67^5) \bmod 119 \equiv (33 \cdot 16) \bmod 119 \equiv 52.$$

Именно число $g = 52$ покупатель посылает в банк.

Банк списывает со счета покупателя 100 грн. и отправляет ему число

$$s' \equiv (52^{77}) \bmod 119 \equiv 103.$$

Покупатель вычисляет

$$s \equiv (103 \cdot 67^{-1}) \bmod 119 \equiv (103 \cdot 16) \bmod 119 \equiv 101$$

и получает платежеспособную банкноту

$$(x, s) = (33, 101).$$

Эту банкноту он приносит (или посылает) в магазин, чтобы купить товар.

Магазин предъявляет банкноту в банк. Банк делает следующие проверки:

1) номер банкноты $x = 33$ состоит из двух одинаковых десятичных цифр (т. е. содержит требуемую избыточность);

2) ранее банкнота с таким номером не предъявлялась;

3) подпись банка верна, т. е. $33^5 \pmod{119} \equiv 101$.

Так как все проверки прошли успешно, банк зачисляет 100 грн. (это фиксированный номинал банкноты) на счет магазина, о чем ему и сообщает. Магазин отпускает товар покупателю.

В завершение разберем еще две проблемы, возникающие в связи с рассмотренной схемой электронных денег.

Первая проблема. В представленной схеме независимо действующие покупатели или даже один покупатель, который не помнит номеров ранее использованных им банкнот, могут случайно сгенерировать две или более банкноты с одинаковыми номерами. По условиям протокола банк примет к оплате только одну из таких банкнот (ту, которая будет предъявлена первой). Однако примем во внимание размеры чисел, используемых в протоколе. Если номер банкноты – чис-

ло длиной 512 бит и покупатели генерируют его действительно случайным образом, то вероятность получения когда либо двух одинаковых номеров пренебрежимо мала.

Вторая проблема состоит в том, что в рассмотренной схеме используются только банкноты одного фиксированного номинала, что, конечно же, неудобно для покупателя. Решение проблемы использования банкнот разного номинала возможно следующим образом. Банк заводит несколько пар (e_i, d_i) и объявляет, что e_1 соответствует, например, 500 грн., e_2 – 200 грн. и т. д. Когда покупатель запрашивает "слепую" подпись в банке, он дополнительно сообщает, какого номинала банкноту он хочет получить. Банк снимает с его счета сумму, равную указанному номиналу, и формирует подпись, используя соответствующее секретное число d_i. Когда впоследствии банк получает подписанную банкноту, он использует для проверки подписи по очереди числа e_1, e_2 и т. д. Если подпись оказалась верна для какого-то e_i, то принимается банкнота i-того номинала. В случае, когда параметр x банкноты содержит фиксированный заголовок с указанием ее номинала, задача проверки подписи облегчается – банк сразу использует нужный ключ e_i.

Упражнение

В системе электронных денег выбраны секретные параметры банка $q = 7$; $p = 17$; $d = 25$ и соответствующие им открытые параметры $n = 119$; $e = 5$. Сформировать электронные банкноты со следующими номерами:

а) $x = 11$, $r = 5$;
б) $x = 99$, $r = 6$;
в) $x = 55$, $r = 10$;
г) $x = 44$, $r = 15$;
д) $x = 77$, $r = 30$.

ТЕСТЫ ДЛЯ РЕКТОРСКОЙ ПРОВЕРКИ

1 Из каких частей состоит автоматизированная система?

- сети связи, аппаратно-программных средств, персонала; информационных ресурсов;
- информационной технологии, внешней среды; вычислительной системы; персонала и информации;
- информационных ресурсов; персонала; операционной системы; внешней среды; распределенной вычислительной системы.

2 Какие свойства информации подлежат защите в автоматизированных системах и системах телекоммуникации?

- конфиденциальность, целостность, доступность, защищенность;
- конфиденциальность, целостность, доступность;
- целостность, доступность, секретность.

3 Что определяет понятие “документ” согласно нормативной базе Украины?

- информация, которая зафиксирована на любом материальном носителе;
- информация, которая зафиксирована на любом материальном носителе в определенном законом порядке;
- информация, которая зафиксирована на любом материальном носителе и зарегистрирована в государственном органе.

4 В каких трех аспектах должна решаться проблема защиты информации в телекоммуникационных системах?

- усовершенствование соответствующей нормативной базы, организационных мер и программно-аппаратных средств;
- усовершенствованием соответствующей нормативной базы, разработка современных защищенных информационных технологий и усовершенствование распределенных вычисленных сетей.

5 Что называют угрозами информационным объектам?

- потенциально возможные события, которые приводят к нарушениям политики безопасности;
- потенциально возможные события, которые приводят к потерям конфиденциальности, целостности и доступности информации;
- потенциально возможные события, которые приводят к потерям электронных документов.

6 Что называют атаками на информационные объекты?

- попытка реализации угрозы;
- подготовка, реализация несанкционированного доступа к объекту, который защищается, и устранение следов нападения;
- факт несанкционированного доступа к объекту, который защищается.

7 Атака – это попытка реализации угрозы?

- да;
- нет.

8 Угрозы, которые имеют субъективную природу, могут быть только случайными?

- да;
- нет.

9 Конфиденциальность, целостность и доступность – свойства информации?

- да;
- нет.

10 Целостность – это отсутствие умышленного искажения информации?

- да;
- нет.

11 Автоматизированная система это:

- система, которая осуществляет автоматизированную обработку данных и в состав которой входят технические средства их обработки (средства вычислительной техники и связи), а также методы и процедуры, программное обеспечение;
- система, с помощью которой обеспечивается связь между пользователями.

12 Какие алгоритмы называют симметричными?

- блочные, поточные, с открытым ключом;
- алгоритмы, в которых операции шифрования выполняются одним ключом.

13 Какие криптографические преобразования используются для создания симметричных криптографических алгоритмов?

- гаммирование, перестановки и замены;
- функциональные преобразования, перестановки и замены;
- параметрические преобразования, перестановки и замены, гаммирование.

14 Какие криптографические преобразования используются в стандарте DES?

- обратные табличные перестановки, табличные замены, перестановки с расширением блоков, перестановки с сжатием блоков, циклические перестановки;
- табличные перестановки, табличные замены, функциональные преобразования.

15 Какие криптографические преобразования используются в стандарте ГОСТ 28147–89?

- обратные табличные перестановки, табличные замены, перестановки с расширением блоков, перестановки со сжатием блоков, циклические перестановки;
- табличные перестановки, табличные замены и функциональные преобразования.

16 К поточным алгоритмам шифрования относится:
- алгоритм, который использует разные ключи зашифровывания и расшифровывания;
- алгоритм, в котором каждый символ открытого текста зашифровывается независимо от других и расшифровывается точно так же.

17 ГОСТ 28147–89 работает в следующих рабочих режимах:
- гаммирование с обратной связью по выходу;
- сцепление блоков шифра;
- простая замена.

18 Циклический регистр сдвига в алгоритме ГОСТ 28147–89 выполняется:
- на 11 разрядов вправо;
- на 11 разрядов влево;
- на 11 разрядов вправо, потом влево, в зависимости от номера цикла.

19 Число циклов шифрования в алгоритме DES:
- 16;
- 32;
- 64;
- 128;
- 256.

20 Размер блока данных в алгоритме ГОСТ 28147–89:
- 16;
- 32;
- 64;
- 128;
- 256.

21 Размер ключа в алгоритме ГОСТ 28147–89:
- 16;
- 32;
- 64;
- 128;
- 256.

22 Размер ключа в алгоритме DES:
- 16;
- 32;
- 48;
- 59;
- 64.

23 Количество циклов шифрования в алгоритме ГОСТ 28147–89:
- 16;
- 32;
- 64;
- 128;
- 256.

24 Какие параметры выбираются в алгоритме RSA:
- два больших простых числа, ключ зашифровывания;
- функция Эйлера, ключ расшифровывания;
- ключ зашифровывания и расшифровывания.

25 Для чего предназначен алгоритм Диффи–Хеллмана:
- для шифрования сообщений?
- для генерации секретного ключа?

26 Алгоритм Эль-Гамаля можно использовать:
- для шифрования сообщений и генерации секретного ключа?
- для генерации секретного ключа?
- для шифрования сообщений и создания цифровой подписи?

27 Хэш-функция применяется в криптографии:
- для шифрования сообщений?
- для генерации секретного ключа?
- в протоколах аутентификации, цифровой подписи?

28 Зашифровать сообщение M с помощью алгоритма RSA, если известно: $n = 15$; ключ зашифровывания $e = 3$; $M = 3$.
- 16;
- 9;
- 12;
- 25.

29 Расшифровать криптограмму C с помощью алгоритма RSA, если известно: $n = 21$; ключ расшифровывания $d = 3$; $C = 5$.
- 20;
- 17;
- 15;
- 7.

30 Найти наибольший общий делитель для чисел 85, 34.
- 1;
- 5;
- 7;
- 17.

31 Вычислить $4^8 \pmod{14}$.
- 1;
- 2;
- 4^4;
- 14.

32 Найти x, если $3^x \equiv 5 \pmod{7}$.
- 3;
- 5;
- 7;
- 6.

33 Решить сравнение $6x \equiv 2 \pmod{11}$.

- 3;
- 4;
- 6;
- 11.

34 Вычислить функцию Эйлера $\varphi(49)$.

- 26;
- 36;
- 48;
- 42.

35 Зашифровать сообщение M с помощью алгоритма Рабина, если известно: $p = 3$; $q = 7$; $M = 9$.

- 27;
- 60;
- 18;
- 21.

36 Зашифровать сообщение M с помощью алгоритма Эль-Гамаля, если известно: $p = 5$; открытый ключ стороны $A - Y_a = 3$; секретный ключ стороны $B - k_b = 2$; $M = 6$.

- 1;
- 2;
- 3;
- 4.

37 Определить ключ методом Диффи–Хеллмана, если известно: $p = 5$; открытый ключ стороны $A - Y_a = 6$; секретный ключ стороны $B - k_b = 3$.

- 1;
- 2;
- 3.

ЛИТЕРАТУРА

1 **Введение** в криптографию / Под общ. ред. В.В. Ященко. – СПб.: Питер, 2001. – 288 с.: ил.

2 **Барабаш А.В., Шанкин Г.П.** Криптография: Науч.-поп. изд. Серия "Аспекты защиты". – М.: Солон-Р, 2002. – 511 с.

3 **Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф.** Защита информации в компьютерных системах и сетях. – М.: Радио и связь, 2001. – 376 с.

4 **Анин Б.Ю.** Защита компьютерной информации. – СПб.: БХВ, 2000. – 384 с.

5 **Саломаа А.** Классическая криптография: Пер. с англ. – М.: Мир, 1996. – 304 с.

6 **Петров А.А.** Компьютерная безопасность. Криптографические методы защиты. – М.: ДМК, 2000. – 448 с.

7 **Жельников В.** Криптография от папируса до компьютера. – М.: ABF, 1997. – 336 с.

8 **Шнайдер Б.** Прикладная криптография. – М.: Триумф, 2003. – 816 с.

9 **Андерсон, Джеймс А.** Дискретная математика и комбинаторика: Пер. с англ. – М.: Изд. дом "Вильямс", 2003. – 960 с.: ил.

10 **Hellman M.E.** The mathematics of public-key cryptography. Scientific American, 1979.

11 **Shamir A.** A Polynomial Time Algorithm for Breaking the Basic Merkle Hellman Cryptosystem // IEEE Transactions on Information Theory. – 1984, Sep. – V. IT-30, n. 5. – Pp. 1699-1704.

12 **Rives R.L., Shamir A. and Adleman L.M.** A Method tor Obtaining Digital Signatures and Public-Key Cryptosystems // Communications of the ACM. – 1978, Feb. – V. 21, n. 2. – Pp.120-126.

13 **Rives R.L., Shamir A. and Adleman L.M.** On Digital Signatures and Public Key Cryptosystems // MIT Laboratory for Computer Science: Technical Report. MIT/LCSATR-212, 1979, Jan.

14 **Simmons G.J.** A Wea' Privacy Protocol Using the RSA Cryptosystem // Cryptologia. – 1983, Apr. – V. 7, n. 2. – Pp. 180-182.

15 **DeLaurentis I.M.** A Further Weakness in the Common Modulus Protocol for the RSA Cryptosystem // Cryptologia. – 1984. – V. 8, n. 3. – Pp. 253-259.

16 **ISO/IEC 9796.** Information Technology Security Techniques. Digital Signature Scheme Giving Message Recovery. International Organization for Standardization. 1991.

17 **Rabin M.O.** Digital Signatures and Public-Key Functions as Intractable as Factorization // MIT Laboratory for Computer Science: Technical Report. MIT/LCS/TR – 1979, Jan., p. 212.

18 **Williams H.C.** A Modification of the RSA Public-Key Encryption Procedure // IEEE Transactions on Information Theory. – 1980, Nov. – V. IT-26, n. 6. – Pp. 726-729.

19 **ElGamal T.** A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Advances in Cryptology: Proceedings of CRYPTO 84, Springer Verlag, 1985. – Pp. 1-18.

20 **ElGamal T.** On Computing Logarithms Over Finite Fields. Advances in Cryptology: Proceedings of CRYPTO 85, Springer Verlag, 1986. – Pp. 396-402.

21 **Diffie W. and Hellman M.E.** New Directions in Cryptography // IEEE Transactions on Information Theory. – 1976, Nov. – V. IT-22, n. 6. – Pp. 44-54.

22 **Болотов А.А., Гашков С.Б., Фролов А.Б.** Алгоритмические основы эллиптической криптографии. – М.: Мэй, 2000. – 100 с.

23 **Болотов А.А., Гашков С.Б., Фролов А.Б.** Элементарное введение в эллиптическую криптографию: Алгебраические и алгоритмические основы. – М.: КомКнига, 2006. – 328 с.

24 **Болотов А.А., Гашков С.Б., Фролов А.Б.** Элементарное введение в эллиптическую криптографию: Протоколы криптографии на эллиптических кривых. – М.: КомКнига, 2006. – 280 с.

25 **Menezes A., van Oorschot P., Vanstone S.** Handbook of Applied Cryptography. – CRC Press, 1996. – 661 p.

26 **Blake I., Seroussi G., Smart N.** Elliptic Curves in Cryptography. – Cambridge University Press, 1999. – 204 p.

27 **Menezes A.** Elliptic Curve Public Key Cryptosystems. – Kluwer Academic Publishers, 1993.

28 **Анохин М. И., Варновский Н. П., Сидельников В. М.** Криптография в банковском деле. – М.: МИФИ, 1997.

29 **Rivest R.** The MD5 Message-Digest Algorithm. RFC 1321, MIT and RSA Data Security, Inc. 1992, Apr.

30 **Research** and Development in Advanced Communication Technologies in Europe. RIPE Integrity Primitives: Final Report of RACE Integrity Primitives Evaluation (R1040). RACE. 1992, June.
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.8213

31 **Federal** Information Processing Standards Publication 180-1 SECURE HASH STANDARD. 1995, Apr.
http://www.netnsk.ru/publica/security/sec_05.htm

32 **Информационная технология.** Криптографическая защита информации. Функция хэширования. ГОСТ Р 34.11–94. (утв. Постановлением Госстандарта РФ от 23.05.1994 № 154).

33 **ГОСТ 28147–89.** Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.
http://protect.gost.ru/document.aspx?control=7&id=139177

34 **Anderson R.** Security Engineering: A Guide to Building Dependable Distributed Systems, 2001, John Wiley & Sons, 640 pages.

35 **PKCS #1** v 2.1: RSA Cryptography Standard RSA Laboratories. 2002, June. http://www.rsa.com

36 **Zheng Y., Pieprzyk J. and Seberry J.** HAVAL A One-Way Hashing Algorithm with Variable Length of Output: Springer Verlag, 1993. – Pp. 83-104.

37 **National** Institute of Standards and Technology. NIST FIPS PUB 186, Digital Signature Standard, U.S. Department of Commerce, 1994, May.

38 **ANSI X9.62–1999.** Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999.

39 **ANSI X9.63–1999.** Public Key Cryptography For The Financial Services Industry: Elliptic Curve Key Agreement and Transport Protocols, 1999.

40 **IEEE Std 1363–2000.** IEEE Standard Specifications for Public-Key Cryptography, 2000.

41 **Goldwasser S., Bellare M.** Lecture notes on cryptography. 2008, July. http://www-cse.ucsd.edu/users/mihir/crypto-lecnotes.html

42 **ГОСТ Р 34.10–94**. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма. http://www.securitylab.ru/informer/240665.php

43 **ГОСТ Р 34.10–2001.** Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи http://protect.gost.ru/v.aspx?control=7&id=131131

44 **ГОСТ Р 34.11–94.** Информационная технология. Криптографическая защита информации. Функция хэширования. http://protect.gost.ru/document.aspx?control=7&id=134550

45 **ГОСТ 34.311–95.** Информационная технология. Криптографическая защита информации. Функция хэширования. http://protect.gost.ru/document.aspx?control=7&id=132760

Приложение А

Таблица простых чисел, не превышающих 2200

2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	61	67	71	73	79	83	89	97
101	103	107	109	113	127	131	137	139	149	151	157	163	167	173	179	181	191	193	197	199				
211	223	227	229	233	239	241	251	257	263	269	271	277	281	283	293									
307	311	313	317	331	337	347	349	353	359	367	373	379	383	389	397									
401	409	419	421	431	433	439	443	449	457	461	463	467	479	487	491	499								
503	509	521	523	541	547	557	563	569	571	577	587	593	599											
601	607	613	617	619	631	641	643	647	653	659	661	673	677	683	691									
701	709	719	727	733	739	743	751	757	761	769	773	787	797											
809	811	821	823	827	829	839	853	857	859	863	877	881	883	887										
907	911	919	929	937	941	947	953	967	971	977	983	991	997											
1009	1013	1019	1021	1031	1033	1039	1049	1051	1061	1063	1069	1087	1091	1093	1097									
1103	1109	1117	1123	1129	1151	1153	1163	1171	1181	1187	1193													
1201	1213	1217	1223	1229	1231	1237	1249	1259	1277	1279	1283	1289	1291	1297										
1301	1303	1307	1319	1321	1327	1361	1367	1373	1381	1399														
1409	1423	1427	1429	1433	1439	1447	1451	1453	1459	1471	1481	1483	1487	1489	1493	1499								
1511	1523	1531	1543	1549	1553	1559	1567	1571	1579	1583	1597													
1601	1607	1609	1613	1619	1621	1627	1637	1657	1663	1667	1669	1693	1697	1699										
1709	1721	1723	1733	1741	1747	1753	1759	1777	1783	1787	1789													
1801	1811	1823	1831	1847	1861	1867	1871	1873	1877	1879	1889													
1901	1907	1913	1931	1933	1949	1951	1973	1979	1987	1993	1997	1999												
2003	2011	2017	2027	2029	2039	2053	2063	2069	2081	2083	2087	2089	2099											
2111	2113	2129	2131	2137	2141	2143	2153	2161	2179															

Приложение Б

Закон Украины

“Об электронных документах и электронном документообороте”

ЗАКОН УКРАЇНИ

Про електронні документи та електронний документообіг

Цей Закон встановлює основні організаційно-правові засади електронного документообігу та використання електронних документів.

Розділ І ЗАГАЛЬНІ ПОЛОЖЕННЯ

Стаття 1. Визначення термінів

У цьому Законі терміни вживаються в такому значенні:

адресат – фізична або юридична особа, якій адресується електронний документ;

дані – інформація, яка подана у формі, придатній для її оброблення електронними засобами;

посередник – фізична або юридична особа, яка в установленому законодавством порядку здійснює приймання, передавання (доставку), зберігання, перевірку цілісності електронних документів для задоволення власних потреб або надає відповідні послуги за дорученням інших суб’єктів електронного документообігу;

обов’язковий реквізит електронного документа – обов’язкові дані в електронному документі, без яких він не може бути підставою для його обліку і не матиме юридичної сили;

автор електронного документа – фізична або юридична особа, яка створила електронний документ;

суб’єкти електронного документообігу – автор, підписував, адресат та посередник, які набувають передбачених законом або договором прав і обов’язків у процесі електронного документообігу.

Стаття 2. Сфера дії Закону

Дія цього Закону поширюється на відносини, що виникають у процесі створення, відправлення, передавання, одержання, зберігання, оброблення, використання та знищення електронних документів.

Стаття 3. Законодавство про електронні документи та електронний документообіг

Відносини, пов’язані з електронним документообігом та використанням електронних документів, регулюються Конституцією України, Цивільним кодексом України, законами України “Про інформацію”, “Про захист інформації в автоматизованих системах”, “Про державну таємницю”, “Про зв’язок”, “Про обов’язковий примірник документів”, “Про Національний архівний фонд та архівні установи”, цим Законом, а також іншими нормативно-правовими актами.

Якщо міжнародним договором України, згода на обов'язковість якого надана Верховною Радою України, встановлено інші правила, ніж ті, що передбачені цим Законом, застосовуються правила міжнародного договору.

Стаття 4. Державне регулювання електронного документообігу

Кабінет Міністрів України та інші органи виконавчої влади в межах повноважень, визначених законом, реалізують державну політику електронного документообігу. Державне регулювання у сфері електронного документообігу спрямовано на:

реалізацію єдиної державної політики електронного документообігу;

забезпечення прав і законних інтересів суб'єктів електронного документообігу;

нормативно-правове забезпечення технології оброблення, створення, передавання, одержання, зберігання, використання та знищення електронних документів.

Розділ II ЕЛЕКТРОННИЙ ДОКУМЕНТ

Стаття 5. Електронний документ

Електронний документ – документ, інформація в якому зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити документа.

Склад та порядок розміщення обов'язкових реквізитів електронних документів визначається законодавством.

Електронний документ може бути створений, переданий, збережений і перетворений електронними засобами у візуальну форму.

Візуальною формою подання електронного документа є відображення даних, які він містить, електронними засобами або на папері у формі, придатній для приймання його змісту людиною.

Стаття 6. Електронний підпис

Електронний підпис є обов'язковим реквізитом електронного документа, який використовується для ідентифікації автора та/або підписувача електронного документа іншими суб'єктами електронного документообігу.

Накладанням електронного підпису завершується створення електронного документа.

Відносини, пов'язані з використанням електронних цифрових підписів, регулюються законом.

Використання інших видів електронних підписів в електронному документообігу здійснюється суб'єктами електронного документообігу на договірних засадах.

Стаття 7. Оригінал електронного документа

Оригіналом електронного документа вважається електронний примірник документа з обов'язковими реквізитами, у тому числі з електронним цифровим підписом автора.

У разі надсилання електронного документа кільком адресатам або його зберігання на кількох електронних носіях інформації кожний з електронних примірників вважається оригіналом електронного документа.

Якщо автором створюються ідентичні за документарною інформацією та реквізитами електронний документ та документ на папері, кожен з документів є оригіналом і має однакову юридичну силу.

Оригінал електронного документа повинен давати змогу довести його цілісність та справжність у порядку, визначеному законодавством; у визначених законодавством випадках може бути пред'явлений у візуальній формі відображення, в тому числі у паперовій копії.

Електронна копія електронного документа засвідчується у порядку, встановленому законом.

Копією документа на папері для електронного документа є візуальне подання електронного документа на папері, яке засвідчене в порядку, встановленому законодавством.

Стаття 8. Правовий статус електронного документа та його копії

Юридична сила електронного документа не може бути заперечена виключно через те, що він має електронну форму.

Допустимість електронного документа як доказу не може заперечуватися виключно на підставі того, що він має електронну форму.

Електронний документ не може бути застосовано як оригінал:

1. свідоцтва про право на спадщину;
2. документа, який відповідно до законодавства може бути створений лише в одному оригінальному примірнику, крім випадків існування централізованого сховища оригіналів електронних документів;
3. в інших випадках, передбачених законом.

Нотаріальне посвідчення цивільно-правової угоди, укладеної шляхом створення електронного документа (електронних документів), здійснюється у порядку, встановленому законом.

Розділ ІІІ ЗАСАДИ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

Стаття 9. Електронний документообіг

Електронний документообіг (обіг електронних документів) – сукупність процесів створення, оброблення, відправлення, передавання, одержання, зберігання, використання та знищення електронних документів, які виконуються із застосуванням перевірки цілісності та у разі необхідності з підтвердженням факту одержання таких документів.

Порядок електронного документообігу визначається державними органами, органами місцевого самоврядування, підприємствами, установами та організаціями всіх форм власності згідно з законодавством.

Стаття 10. Відправлення та передавання електронних документів

Відправлення та передавання електронних документів здійснюються автором або посередником в електронній формі за допомогою засобів інформаційних, телекомунікаційних, інформаційно-телекомунікаційних систем або шляхом відправлення електронних носіїв, на яких записано цей документ.

Якщо автор і адресат у письмовій формі попередньо не домовилися про інше, датою і часом відправлення електронного документа вважаються дата і час, коли відправлення електронного документа не може бути скасовано особою, яка його відправила. У разі відправлення електронного документа шляхом пересилання його на електронному носії, на якому записано цей документ, датою і часом відправлення вважаються дата і час здавання його для пересилання.

Вимоги підтвердження факту одержання документа, встановлені законодавством у випадках відправлення документів рекомендованим листом або передавання їх під розписку, не поширюються на електронні документи. У таких випадках підтвердження факту одержання електронних документів здійснюється згідно з вимогами цього Закону.

Стаття 11. Одержання електронних документів

Електронний документ вважається одержаним адресатом з часу надходження авторові повідомлення в електронній формі від адресата про одержання цього електронного документа автора, якщо інше не передбачено законодавством або попередньою домовленістю між суб'єктами електронного документообігу.

Якщо попередньою домовленістю між суб'єктами електронного документообігу не визначено порядок підтвердження факту одержання електронного документа, таке підтвердження може бути здійснено в будь-якому порядку автоматизованим чи іншим способом в електронній формі або у формі документа на папері. Зазначене підтвердження повинно містити дані про факт і час одержання електронного документа та про відправника цього підтвердження.

У разі ненадходження до автора підтвердження про факт одержання цього електронного документа вважається, що електронний документ не одержано адресатом.

Якщо автор і адресат у письмовій формі попередньо не домовилися про інше, електронний документ вважається відправленим автором та одержаним адресатом за їх місцезнаходженням (для фізичних осіб – місцем проживання), у тому числі якщо інформаційна, телекомунікаційна, інформаційно-телекомунікаційна система, за допомогою якої одержано документ, знаходиться в іншому місці. Місцезнаходження (місце проживання) сторін визначається відповідно до законодавства.

Стаття 12. Перевірка цілісності електронного документа

Перевірка цілісності електронного документа проводиться шляхом перевірки електронного цифрового підпису.

Стаття 13. Зберігання електронних документів та архіви електронних документів

Суб'єкти електронного документообігу повинні зберігати електронні документи на електронних носіях інформації у формі, що дає змогу перевірити їх цілісність на цих носіях.

Строк зберігання електронних документів на електронних носіях інформації повинен бути не меншим від строку, встановленого законодавством для відповідних документів на папері.

У разі неможливості зберігання електронних документів на електронних носіях інформації протягом строку, встановленого законодавством для відповідних документів на папері, суб'єкти електронного документообігу повинні вживати заходів щодо дублювання документів на кількох електронних носіях інформації та здійснювати їх періодичне копіювання відповідно до порядку обліку та копіювання документів, встановленого законодавством. Якщо неможливо виконати зазначені вимоги, електронні документи повинні зберігатися у вигляді копії документа на папері (у разі відсутності оригіналу цього документа на папері). При копіюванні електронного документа з електронного носія інформації обов'язково здійснюється перевірка цілісності даних на цьому носії.

При зберіганні електронних документів обов'язкове додержання таких вимог:

1. інформація, що міститься в електронних документах, повинна бути доступною для її подальшого використання;
2. має бути забезпечена можливість відновлення електронного документа у тому форматі, в якому він був створений, відправлений або одержаний;
3. у разі наявності повинна зберігатися інформація, яка дає змогу встановити походження та призначення електронного документа, а також дату і час його відправлення чи одержання.

Суб'єкти електронного документообігу можуть забезпечувати додержання вимог щодо збереження електронних документів шляхом використання послуг посередника, у тому числі архівної установи, якщо така установа додержується вимог цієї статті. Створення архівів електронних документів, подання електронних документів до архівних установ України та їх зберігання в цих установах здійснюється у порядку, визначеному законодавством.

Розділ IV ОРГАНІЗАЦІЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

Стаття 14. Організація електронного документообігу

Електронний документообіг здійснюється відповідно до законодавства України або на підставі договорів, що визначають взаємовідносини суб'єктів електронного документообігу.

Використання електронного документа у цивільних відносинах здійснюється згідно із загальними вимогами вчинення правочинів, встановлених цивільним законодавством.

Стаття 15. Обіг електронних документів, що містять інформацію з обмеженим доступом

Суб'єкти електронного документообігу, які здійснюють його на договірних засадах, самостійно визначають режим доступу до електронних документів, що містять конфіденційну інформацію, та встановлюють для них систему (способи) захисту.

В інформаційних, телекомунікаційних, інформаційно-телекомунікаційних системах, які забезпечують обмін електронними документами, що містять інформацію, яка є власністю держави, або інформацію з обмеженим доступом, повинен забезпечуватися захист цієї інформації відповідно до законодавства.

Стаття 16. Права та обов'язки суб'єктів електронного документообігу

Суб'єкти електронного документообігу користуються правами та мають обов'язки, які встановлено для них законодавством.

Якщо в процесі організації електронного документообігу виникає необхідність у визначенні додаткових прав та обов'язків суб'єктів електронного документообігу, що не визначені законодавством, такі права та обов'язки можуть встановлюватися цими суб'єктами на договірних засадах.

Стаття 17. Вирішення спорів між суб'єктами електронного документообігу

Вирішення спорів між суб'єктами електронного документообігу здійснюється в порядку, встановленому законом.

Стаття 18. Відповідальність за порушення законодавства про електронні документи та електронний документообіг

Особи, винні в порушенні законодавства про електронні документи та електронний документообіг, несуть відповідальність згідно з законами України.

Розділ V ПРИКІНЦЕВІ ПОЛОЖЕННЯ

1. Цей Закон набирає чинності через шість місяців з дня його опублікування.
2. Кабінету Міністрів України протягом шести місяців з дня набрання чинності цим Законом:

підготувати та подати на розгляд Верховної Ради України відповідні пропозиції про внесення змін до законодавчих актів України;

забезпечити прийняття нормативно-правових актів, передбачених цим Законом;

забезпечити перегляд і скасування міністерствами, іншими центральними органами виконавчої влади України їх нормативно-правових актів, що суперечать цьому Закону;

разом з Національним банком України розробити та внести на розгляд Верховної Ради України програму заходів щодо впровадження електронних документів, електронного документообігу та електронного цифрового підпису, стимулювання підприємств, установ і організацій, які впроваджують електронний документообіг.

м. Київ
22 травня 2003 року
№ 851–IV

Приложение В

Закон Украины “Об электронной цифровой подписи”

ЗАКОН УКРАЇНИ

Про електронний цифровий підпис

Цей Закон визначає правовий статус електронного цифрового підпису та регулює відносини, що виникають при використанні електронного цифрового підпису.

Дія цього Закону не поширюється на відносини, що виникають під час використання інших видів електронного підпису, в тому числі переведеного у цифрову форму зображення власноручного підпису.

Якщо міжнародним договором, згода на обов’язковість якого надана Верховною Радою України, встановлено інші правила, ніж ті, що передбачені цим Законом, застосовуються правила міжнародного договору.

Стаття 1. Визначення термінів

У цьому Законі терміни вживаються у такому значенні:

електронний підпис – дані в електронній формі, які додаються до інших електронних даних або логічно з ними пов’язані та призначені для ідентифікації підписувача цих даних;

електронний цифровий підпис – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача. Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа;

засіб електронного цифрового підпису – програмний засіб, програмно-апаратний або апаратний пристрій, призначені для генерації ключів, накладення та/або перевірки електронного цифрового підпису;

особистий ключ – параметр криптографічного алгоритму формування електронного цифрового підпису, доступний тільки підписувачу;

відкритий ключ – параметр криптографічного алгоритму перевірки електронного цифрового підпису, доступний суб’єктам відносин у сфері використання електронного цифрового підпису;

засвідчення чинності відкритого ключа – процедура формування сертифіката відкритого ключа;

сертифікат відкритого ключа (далі – сертифікат ключа) – документ, виданий центром сертифікації ключів, який засвідчує чинність і належність відкритого ключа підписувачу. Сертифікати ключів можуть розповсюджуватися в електронній формі або у формі документа на папері та використовуватися для ідентифікації особи підписувача;

посилений сертифікат відкритого ключа (далі – посилений сертифікат ключа) – сертифікат ключа, який відповідає вимогам цього Закону, виданий ак-

редитованим центром сертифікації ключів, засвідчувальним центром, центральним засвідчувальним органом;

акредитація – процедура документального засвідчення компетентності центра сертифікації ключів здійснювати діяльність, пов'язану з обслуговуванням посилених сертифікатів ключів;

компрометація особистого ключа – будь-яка подія та/або дія, що призвела або може призвести до несанкціонованого використання особистого ключа;

блокування сертифіката ключа – тимчасове зупинення чинності сертифіката ключа;

підписувач – особа, яка на законних підставах володіє особистим ключем та від свого імені або за дорученням особи, яку вона представляє, накладає електронний цифровий підпис під час створення електронного документа;

послуги електронного цифрового підпису – надання у користування засобів електронного цифрового підпису, допомога при генерації відкритих та особистих ключів, обслуговування сертифікатів ключів (формування, розповсюдження, скасування, зберігання, блокування та поновлення), надання інформації щодо чинних, скасованих і блокованих сертифікатів ключів, послуги фіксування часу, консультації та інші послуги, визначені цим Законом;

надійний засіб електронного цифрового підпису – засіб електронного цифрового підпису, що має сертифікат відповідності або позитивний експертний висновок за результатами державної експертизи у сфері криптографічного захисту інформації. Підтвердження відповідності та проведення державної експертизи цих засобів здійснюється у порядку, визначеному законодавством.

Стаття 2. Суб'єкти правових відносин у сфері послуг електронного цифрового підпису

Суб'єктами правових відносин у сфері послуг електронного цифрового підпису є:

підписувач;

користувач;

центр сертифікації ключів;

акредитований центр сертифікації ключів;

центральний засвідчувальний орган;

засвідчувальний центр органу виконавчої влади або іншого державного органу (далі – засвідчувальний центр);

контролюючий орган.

Стаття 3. Правовий статус електронного цифрового підпису

Електронний цифровий підпис за правовим статусом прирівнюється до власноручного підпису (печатки) у разі, якщо:

електронний цифровий підпис підтверджено з використанням посиленого сертифіката ключа за допомогою надійних засобів цифрового підпису;

під час перевірки використовувався посилений сертифікат ключа, чинний на момент накладення електронного цифрового підпису;

особистий ключ підписувача відповідає відкритому ключу, зазначеному у сертифікаті.

Електронний підпис не може бути визнаний недійсним лише через те, що він має електронну форму або не ґрунтується на посиленому сертифікаті ключа.

Стаття 4. Призначення електронного цифрового підпису

Електронний цифровий підпис призначений для забезпечення діяльності фізичних та юридичних осіб, яка здійснюється з використанням електронних документів.

Електронний цифровий підпис використовується фізичними та юридичними особами – суб'єктами електронного документообігу для ідентифікації підписувача та підтвердження цілісності даних в електронній формі.

Використання електронного цифрового підпису не змінює порядку підписання договорів та інших документів, встановленого законом для вчинення правочинів у письмовій формі.

Нотаріальні дії із засвідчення справжності електронного цифрового підпису на електронних документах вчиняються відповідно до порядку, встановленого законом.

Стаття 5. Особливості застосування електронного цифрового підпису

Органи державної влади, органи місцевого самоврядування, підприємства, установи та організації державної форми власності для засвідчення чинності відкритого ключа використовують лише посилений сертифікат ключа.

Інші юридичні та фізичні особи можуть на договірних засадах засвідчувати чинність відкритого ключа сертифікатом ключа, сформованим центром сертифікації ключів, а також використовувати електронний цифровий підпис без сертифіката ключа.

Розподіл ризиків збитків, що можуть бути заподіяні підписувачам, користувачам та третім особам, які користуються електронними цифровими підписами без сертифіката ключа, визначається суб'єктами правових відносин у сфері послуг електронного цифрового підпису на договірних засадах.

Захист прав споживачів послуг електронного цифрового підпису, а також механізм реалізації захисту цих прав регулюються цим Законом та Законом України "Про захист прав споживачів".

У випадках, коли відповідно до законодавства необхідне засвідчення дійсності підпису на документах та відповідності копій документів оригіналам печаткою, на електронний документ накладається ще один електронний цифровий підпис юридичної особи, спеціально призначений для таких цілей.

Порядок застосування електронного цифрового підпису органами державної влади, органами місцевого самоврядування, підприємствами, установами та організаціями державної форми власності визначається Кабінетом Міністрів України.

Порядок застосування цифрового підпису в банківській діяльності визначається Національним банком України.

Стаття 6. Вимоги до сертифіката ключа

Сертифікат ключа містить такі обов'язкові дані:

найменування та реквізити центру сертифікації ключів (центрального засвідчувального органу, засвідчувального центру);

зазначення, що сертифікат виданий в Україні;

унікальний реєстраційний номер сертифіката ключа;

основні дані (реквізити) підписувача – власника особистого ключа;

дату і час початку та закінчення строку чинності сертифіката;

відкритий ключ;

найменування криптографічного алгоритму, що використовується власником особистого ключа;

інформацію про обмеження використання підпису.

Посилений сертифікат ключа, крім обов'язкових даних, які містяться в сертифікаті ключа, повинен мати ознаку посиленого сертифіката ключа.

Інші дані можуть вноситися у посилений сертифікат ключа на вимогу його власника.

Стаття 7. Права та обов'язки підписувача

Підписувач має право:

вимагати скасування, блокування або поновлення свого сертифіката ключа;

оскаржити дії чи бездіяльність центру сертифікації ключів у судовому порядку.

Підписувач зобов'язаний:

зберігати особистий ключ у таємниці;

надавати центру сертифікації ключів дані згідно з вимогами статті 6 цього Закону для засвідчення чинності відкритого ключа;

своєчасно надавати центру сертифікації ключів інформацію про зміну даних, відображених у сертифікаті ключа.

Стаття 8. Центр сертифікації ключів

Центром сертифікації ключів може бути юридична особа незалежно від форми власності або фізична особа, яка є суб'єктом підприємницької діяльності, що надає послуги електронного цифрового підпису та засвідчила свій відкритий ключ у центральному засвідчувальному органі або засвідчувальному центрі з дотриманням вимог статті 6 цього Закону.

Обслуговування фізичних та юридичних осіб здійснюється центром сертифікації ключів на договірних засадах.

Центр сертифікації ключів має право:

надавати послуги електронного цифрового підпису та обслуговувати сертифікати ключів;

отримувати та перевіряти інформацію, необхідну для реєстрації підписувача і формування сертифіката ключа безпосередньо у юридичної або фізичної особи чи у її уповноваженого представника.

Центр сертифікації ключів зобов'язаний:

забезпечувати захист інформації в автоматизованих системах відповідно до законодавства;

забезпечувати захист персональних даних, отриманих від підписувача, згідно з законодавством;

встановлювати під час формування сертифіката ключа належність відкритого ключа та відповідного особистого ключа підписувачу;

своєчасно скасовувати, блокувати та поновлювати сертифікати ключів у випадках, передбачених цим Законом;

своєчасно попереджувати підписувача та додавати в сертифікат відкритого ключа підписувача інформацію про обмеження використання електронного цифрового підпису, які встановлюються для забезпечення можливості відшкодування збитків сторін у разі заподіяння шкоди з боку центру сертифікації ключів;

перевіряти законність звернень про скасування, блокування та поновлення сертифікатів ключів та зберігати документи, на підставі яких були скасовані, блоковані та поновлені сертифікати ключів;

цілодобово приймати заяви про скасування, блокування та поновлення сертифікатів ключів;

вести електронний перелік чинних, скасованих і блокованих сертифікатів ключів;

забезпечувати цілодобово доступ користувачів до сертифікатів ключів та відповідних електронних переліків сертифікатів через загальнодоступні телекомунікаційні канали;

забезпечувати зберігання сформованих сертифікатів ключів протягом строку, передбаченого законодавством для зберігання відповідних документів на папері;

надавати консультації з питань, пов'язаних з електронним цифровим підписом.

Зберігання особистих ключів підписувачів та ознайомлення з ними в центрі сертифікації ключів забороняються.

Стаття 9. Акредитований центр сертифікації ключів

Центр сертифікації ключів, акредитований в установленому порядку, є акредитованим центром сертифікації ключів.

Акредитований центр сертифікації ключів має право:

надавати послуги електронного цифрового підпису та обслуговувати виключно посилені сертифікати ключів;

отримувати та перевіряти інформацію, необхідну для реєстрації підписувача і формування посиленого сертифіката ключа, безпосередньо у юридичної або фізичної особи чи її представника.

Акредитований центр сертифікації ключів має виконувати усі зобов'язання та вимоги, встановлені законодавством для центру сертифікації ключів, та додатково зобов'язаний використовувати для надання послуг електронного цифрового підпису надійні засоби електронного цифрового підпису.

Порядок акредитації та вимоги, яким повинен відповідати акредитований центр сертифікації ключів, встановлюються Кабінетом Міністрів України.

Стаття 10. Засвідчувальний центр

Кабінет Міністрів України за необхідності визначає засвідчувальний центр центрального органу виконавчої влади для забезпечення реєстрації, засвідчення чинності відкритих ключів та акредитації групи центрів сертифікації ключів, які надають послуги електронного цифрового підпису цьому органу і підпорядкованим йому підприємствам, установам та організаціям.

Інші державні органи за необхідності, за погодженням з Кабінетом Міністрів України, визначають свої засвідчувальні центри, призначені для виконання функцій, зазначених у частині першій цієї статті.

Засвідчувальний центр по відношенню до групи центрів сертифікації ключів, зазначених у частині першій цієї статті, має ті ж функції і повноваження, що й центральний засвідчувальний орган стосовно центрів сертифікації ключів.

Засвідчувальний центр відповідає вимогам, встановленим законодавством для акредитованого центру сертифікації ключів.

Засвідчувальний центр реєструється, засвідчує свій відкритий ключ і акредитується у центральному засвідчувальному органі.

Положення про засвідчувальний центр центрального органу виконавчої влади затверджується Кабінетом Міністрів України.

Стаття 11. Центральний засвідчувальний орган

Центральний засвідчувальний орган визначається Кабінетом Міністрів України.

Центральний засвідчувальний орган:

формує і видає посилені сертифікати ключів засвідчувальним центрам та центрам сертифікації ключів з дотриманням вимог статті 6 цього Закону;

блокує, скасовує та поновлює посилені сертифікати ключів засвідчувальних центрів та центрів сертифікації ключів у випадках, передбачених цим Законом;

веде електронні реєстри чинних, блокованих та скасованих посилених сертифікатів ключів засвідчувальних центрів та центрів сертифікації ключів;

веде акредитацію центрів сертифікації ключів, отримує та перевіряє інформацію, необхідну для їх акредитації;

забезпечує цілодобово доступ засвідчувальних центрів та центрів сертифікації ключів до посилених сертифікатів ключів та відповідних електронних реєстрів через загальнодоступні телекомунікаційні канали;

зберігає посилені сертифікати ключів засвідчувальних центрів та центрів сертифікації ключів;

надає засвідчувальним центрам та центрам сертифікації ключів консультації з питань, пов’язаних з використанням електронного цифрового підпису.

Центральний засвідчувальний орган відповідає вимогам, встановленим законодавством для акредитованого центру сертифікації ключів.

Положення про центральний засвідчувальний орган затверджується Кабінетом Міністрів України.

Стаття 12. Контролюючий орган

Функції контролюючого органу здійснює спеціально уповноважений центральний орган виконавчої влади у сфері криптографічного захисту інформації.

Контролюючий орган перевіряє дотримання вимог цього Закону центральним засвідчувальним органом, засвідчувальними центрами та центрами сертифікації ключів.

У разі невиконання або неналежного виконання обов'язків та виявлення порушень вимог, встановлених законодавством для центру сертифікації ключів, засвідчувального центру, контролюючий орган дає розпорядження центральному засвідчувальному органу про негайне вжиття заходів, передбачених законом.

Стаття 13. Скасування, блокування та поновлення посиленого сертифіката ключа

Акредитований центр сертифікації ключів негайно скасовує сформований ним посилений сертифікат ключа у разі:

закінчення строку чинності сертифіката ключа;

подання заяви власника ключа або його уповноваженого представника;

припинення діяльності юридичної особи – власника ключа;

смерті фізичної особи – власника ключа або оголошення його померлим за рішенням суду;

визнання власника ключа недієздатним за рішенням суду;

надання власником ключа недостовірних даних;

компрометації особистого ключа.

Центральний засвідчувальний орган негайно скасовує посилений сертифікат ключа центру сертифікації ключів, засвідчувального центру у разі:

припинення діяльності з надання послуг електронного цифрового підпису;

компрометації особистого ключа.

Центральний засвідчувальний орган, засвідчувальний центр, акредитований центр сертифікації ключів негайно блокують посилений сертифікат ключа:

у разі подання заяви власника ключа або його уповноваженого представника;

за рішенням суду, що набрало законної сили;

у разі компрометації особистого ключа.

Скасування і блокування посиленого сертифіката ключа набирає чинності з моменту внесення до реєстру чинних, скасованих і блокованих посилених сертифікатів із зазначенням дати та часу здійснення цієї операції.

Центральний засвідчувальний орган, засвідчувальний центр, акредитований центр сертифікації ключів негайно повідомляють про скасування або блокування посиленого сертифіката ключа його власника.

Блокований посилений сертифікат ключа поновлюється:

у разі подання заяви власника ключа або його уповноваженого представника;

за рішенням суду, що набрало законної сили;

у разі встановлення недостовірності даних про компрометацію особистого ключа.

Стаття 14. Припинення діяльності центру сертифікації ключів

Центр сертифікації ключів припиняє свою діяльність відповідно до законодавства.

Про рішення щодо припинення діяльності центр сертифікації ключів повідомляє підписувачів за три місяці, якщо інші строки не визначено законодавством. Підписувачі мають право обирати за власним бажанням будь-який центр сертифікації ключів для подальшого обслуговування, якщо інше не передбачено законодавством. Після повідомлення про припинення діяльності центр сертифікації ключів не має права видавати нові сертифікати ключів. Усі сертифікати ключів, що були видані центром сертифікації ключів, після припинення його діяльності скасовуються.

Центр сертифікації ключів, що повідомив про припинення своєї діяльності, зобов'язаний забезпечити захист прав споживачів шляхом повернення грошей за послуги, що не можуть надаватися в подальшому, якщо вони були попередньо оплачені.

Акредитований центр сертифікації ключів додатково повідомляє про рішення щодо припинення діяльності центральний засвідчувальний орган або відповідний засвідчувальний центр.

Акредитований центр сертифікації ключів протягом доби, визначеної як дата припинення його діяльності, передає посилені сертифікати ключів, відповідні реєстри посилених сертифікатів ключів та документовану інформацію, яка підлягає обов'язковій передачі, відповідному засвідчувальному центру або центральному засвідчувальному органу.

Порядок передачі акредитованим центром сертифікації ключів посилених сертифікатів ключів, відповідних реєстрів посилених сертифікатів ключів та документованої інформації, яка підлягає обов'язковій передачі, встановлюється Кабінетом Міністрів України.

Стаття 15. Відповідальність за порушення законодавства про електронний цифровий підпис

Особи, винні у порушенні законодавства про електронний цифровий підпис, несуть відповідальність згідно з законом.

Стаття 16. Розв'язання спорів

Спори, що виникають у сфері надання послуг електронного цифрового підпису, розв'язуються в порядку, встановленому законом.

Стаття 17. Визнання іноземних сертифікатів ключів

Іноземні сертифікати ключів, засвідчені відповідно до законодавства тих держав, де вони видані, визнаються в Україні чинними у порядку, встановленому законом.

Стаття 18. Прикінцеві положення

1. Цей Закон набирає чинності з 1 січня 2004 року.
2. До приведення законів України та інших нормативно-правових актів у відповідність із цим Законом вони застосовуються у частині, що не суперечить цьому Закону.
3. Пункт 14 статті 9 Закону України “Про ліцензування певних видів господарської діяльності” (Відомості Верховної Ради України, 2000 р., № 36, ст. 299) після слів “надання послуг в галузі криптографічного захисту інформації” доповнити словами “(крім послуг електронного цифрового підпису)”.
4. Кабінету Міністрів України протягом шести місяців з дня набрання чинності цим Законом:

підготувати та внести до Верховної Ради України пропозиції про внесення змін до законів України, що випливають із цього Закону;

забезпечити приведення своїх нормативно-правових актів, а також нормативно-правових актів міністерств та інших центральних органів виконавчої влади у відповідність з цим Законом;

визначити центральний засвідчувальний орган;

забезпечити прийняття нормативно-правових актів, передбачених цим Законом.

5. Національному банку України протягом шести місяців з дня набрання чинності цим Законом привести свої нормативно-правові акти у відповідність з цим Законом.
6. Кабінету Міністрів України разом з Національним банком України, іншими органами державної влади протягом шести місяців з дня набрання чинності цим Законом розробити та внести на розгляд Верховної Ради України програму заходів щодо впровадження електронного документа, електронного документообігу та електронного цифрового підпису.

м. Київ
22 травня 2003 року
№ 852–IV

Приложение Г

Математическое обоснование атак, в основе которых лежит парадокс о днях рождения

Парадокс задачи о днях рождения часто используют в элементарных курсах по теории вероятностей, чтобы продемонстрировать, что результаты теории вероятностей иногда противоречат интуитивным представлениям. Проблема может быть сформулирована так: чему равно минимальное значение k, при котором вероятность того, что по крайней мере у двоих из группы k человек дни рождения совпадают, оказывается равной 0,5? Игнорируем 29 февраля и предположим, что каждый день рождения одинаково вероятен. Чтобы ответить на поставленный вопрос, определим $P(n, k)$ (имеет место по крайней мере одно совпадение среди k элементов, где каждый элемент принимает одно из n одинаково вероятных значений от 1 до n).

Таким образом, требуется найти наименьшее значение k, при котором $P(365, k) \geq 0{,}5$. Сначала определим вероятность того, что совпадений не обнаружится. Обозначим ее как $Q(365, k)$. Для $k > 365$ невозможно, чтобы все значения оказались различными. Поэтому можно предположить, что $k \leq 365$. Рассмотрим число различных способов N, позволяющих получить k значений без повторений. Для первого элемента мы имеем на выбор любое из 365-ти значений, для второго элемента – любое из 364-х оставшихся и т. д. Поэтому для числа подходящих способов получаем:

$$N = 365 \cdot 364 \ldots (365 - k + 1) = \frac{365!}{(365-k)!}.$$

Если исключить условие отсутствия совпадений, то каждый элемент может принимать любое из 365-ти возможных значений, что в сумме дает 365^k вариантов. Поэтому вероятность отсутствия совпадений равна отношению числа вариантов без совпадений к общему числу вариантов:

$$Q(365, k) = \frac{365!/(365-k)!}{365^k} = \frac{365!}{365^k (365-k)!};$$

$$P(365, k) = 1 - Q(365, k) = 1 - \frac{365!}{365^k (365-k)!}.$$

Данная функция показана на рис. Г.1.

Вероятности могут показаться удивительно большими, если вы не сталкивались с подобной проблемой раньше. Многие думают, что для получения вероятности хотя бы одного совпадения, превышающей 0,5, в группе должно быть около 100 человек, а на самом деле достаточно всего 23-х, поскольку $P(365, 23) = 0{,}5073$. Для $k = 100$ вероятность хотя бы одного совпадения равна 0,9999997.

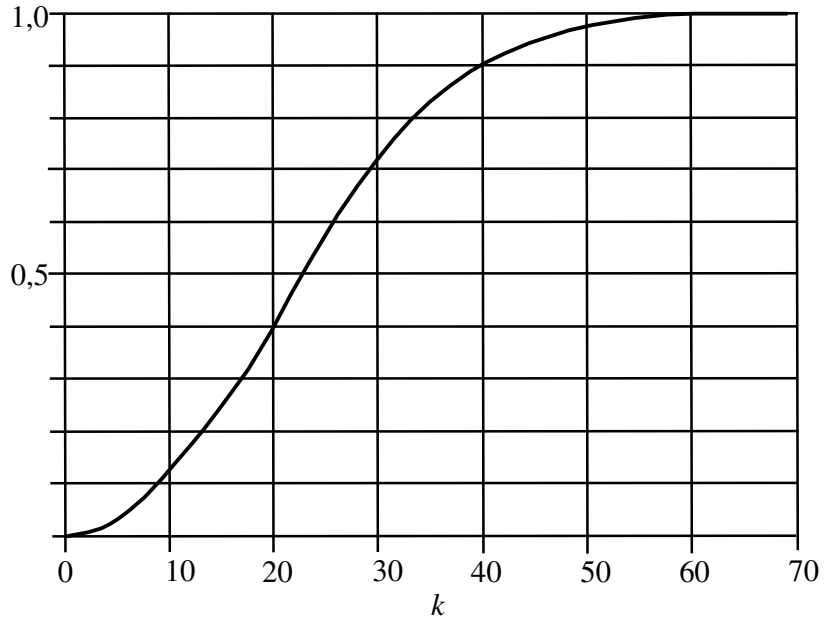

Рисунок Г.1 – Парадокс задачи о днях рождения

Данный результат кажется удивительным, возможно, потому, что для каждого отдельного человека в группе вероятность того, что с его днем рождения совпадет день рождения кого-то другого в группе, достаточно мала. Но мы рассматриваем вероятность того, что для какой-нибудь пары людей в группе дни рождения совпадут. В группе из 23-х человек имеется $\frac{23(23-1)}{2} = 253$ различные пары, поэтому и вероятности такие высокие.

Одно полезное неравенство. Перед тем как приступить к рассмотрению обобщения парадокса задачи о днях рождения, докажем одно полезное неравенство:

$$(1 - x) \le e^{-x} \quad \text{для всех } x \ge 0.$$

Это неравенство иллюстрируется графиком на рис. Г.2.

Чтобы убедиться в истинности этого неравенства, следует обратить внимание на то, что нижняя линия является прямой, касательной к графику функции e^{-x} в точке $x = 0$. Наклон этой прямой в точности равен производной e^{-x} в точке $x = 0$:

$$f(x) = e^{-x};$$
$$f'(x) = \frac{d}{dx} e^{-x} = -e^{-x};$$
$$f'(0) = -1.$$

Касательной к e^{-x} в точке $x = 0$ является прямая вида $ax + b$, для которой $a = -1$ и которая принимает значение $e^0 = 1$ в точке $x = 0$. Такой функцией является $(1 - x)$. Заметим также, что для малых x имеется $(1 - x) \approx e^{-x}$.

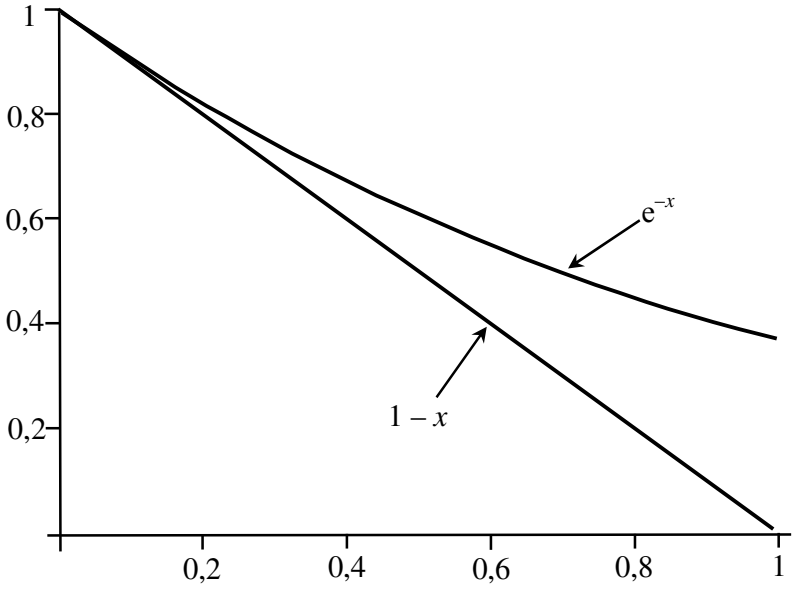

Рисунок Г.2 – Неравенство $(1 - x) \le e^{-x}$

Общий случай. Задачу о днях рождения можно обобщить: если дана целочисленная случайная величина с равномерным распределением значений от 1 до n и имеется выборка, состоящая из k значений этой случайной величины ($k \le n$), то какова вероятность $P(n, k)$ того, что среди значений в выборке по крайней мере два совпадают? Задача, к которой сводится парадокс о днях рождения, является частным случаем только что сформулированной проблемы (при $n = 365$). Используя аргументацию, приведенную выше, получаем обобщение равенства

$$P(n, k) = 1 - \frac{n!}{(n-k)!n^k}.$$

Можно представить его в виде

$$P(n, k) = 1 - \frac{n(n-1)\ldots(n-k+1)}{n^k} = 1 - \left[\frac{n-1}{n} \cdot \frac{n-2}{n} \ldots \frac{n-k+1}{n}\right] =$$

$$= 1 - \left[\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\ldots\left(1-\frac{k-1}{n}\right)\right];$$

$$P(n, k) > 1 - \left[\left(e^{-1/n}\right)\left(e^{-2/n}\right)\ldots\left(e^{-(k-1)/n}\right)\right] > 1 - e^{-[(1/n)+(2/n)+\ldots+((k-1)/n)]} > 1 - e^{-(k(k-1))/2n}.$$

Теперь зададимся вопросом: при каком значении k мы получим неравенство $P(n, k) > 0{,}5$? Для этого требуется, чтобы

$$1/2 = 1 - e^{-(k(k-1))/2n};$$

$$2 = e^{(k(k-1))/2n};$$

$$\ln(2) = \frac{k(k-1)}{2n}.$$

Для больших k значение $k\,(k-1)$ можно заменить на k^2, и тогда получим

$$k = \sqrt{2\ln(2)n} = 1{,}18\sqrt{n} \approx \sqrt{n}\,.$$

Для проверки предположим, что $n = 365$, тогда $k = 1{,}18\ \sqrt{365} = 22{,}54$, что очень близко к правильному значению 23.

Теперь можно сформулировать суть атаки, в основе которой лежит парадокс задачи о днях рождения. Предположим, что имеется функция H, допускающая 2^m вариантов вывода (т. е. m-битовый вывод). Если на вход H подать k вариантов случайного ввода, то каким должно быть значение k, чтобы можно было ожидать на выходе хотя бы одно совпадение [т. е. $H(y) = H(x)$ для некоторых введенных (x, y)]? Получаем:

$$k = \sqrt{2\ln(2)2^m} = 1{,}18\ \sqrt{2^m} \approx \sqrt{2^m}\,.$$

Приложение Д

Обоснование алгоритма цифровой подписи

Целью данного приложения является доказательство того, что в процессе верификации подписи будет получено равенство $v = r$, если подпись верна.

Лемма 1 Для любого целого числа t:

$$\text{если } g \equiv h^{(p-1)/q} \pmod{p}, \text{ то } g^t \pmod{p} \equiv g^{t \bmod q} \pmod{p}.$$

Доказательство. По теореме Ферма, поскольку h является взаимно простым по отношению к p, мы имеем $h^{p-1} \pmod{p} \equiv 1$. Следовательно, для любого неотрицательного целого числа n

$$g^{nq} \pmod{p} \equiv h^{(p-1)/q} \pmod{p} \equiv h^{((p-1)/q)nq} \pmod{p} \equiv h^{(p-1)n} \pmod{p} \equiv$$
$$\equiv (h^{(p-1)} \bmod p)^n \bmod p.$$

Поэтому для неотрицательных целых чисел n и z мы получаем

$$(g^{nq+z}) \bmod p \equiv (g^{nq} g^z) \bmod p \equiv [(g^{nq} \bmod p)(g^z \bmod p)] \bmod p \equiv g^z \pmod{p}.$$

Любое целое неотрицательное число t может быть представлено единственным способом в форме $t = nq + z$, где n и z являются целыми неотрицательными числами и $0 < z < q$. Поэтому $z \equiv t \pmod{q}$, откуда и вытекает необходимый результат.

Лемма 2 $(y^{(rw) \bmod q}) \bmod p \equiv (g^{(krw) \bmod q}) \bmod p$.

Доказательство. По определению, $y \equiv g^k \pmod{p}$. Тогда

$$(y^{(rw) \bmod q}) \bmod p \equiv (g^k \bmod p)^{(rw) \bmod q} \bmod p \equiv (g^{k((rw) \bmod q)}) \bmod p \equiv$$
$$\equiv (g^{(k((rw) \bmod q)) \bmod q}) \bmod p \equiv (g^{(krw) \bmod q}) \bmod p.$$

Лемма 3 $((H(M) + kr)w) \bmod q \equiv x$.

Доказательство. По определению, $s \equiv [x^{-1}(H(M) + kr)] \bmod q$. Также, ввиду того, что q является простым, любое неотрицательное целое число меньше q имеет мультипликативное обратное. Поэтому $(xx^{-1}) \bmod q \equiv 1$. Далее имеем

$$(xs) \bmod q \equiv (x(x^{-1}((H(M) + kr)) \bmod q)) \bmod q \equiv$$
$$\equiv (x(x^{-1}(H(M) + kr))) \bmod q \equiv (((xx^{-1}) \bmod q)((H(M) + kr) \bmod q)) \bmod q \equiv$$
$$\equiv (H(M) + kr) \bmod q.$$

По определению, $w \equiv s^{-1} \bmod q$, поэтому $(ws) \bmod q \equiv 1$. Следовательно,

$$((H(M) + kr)w) \bmod q \equiv (((H(M) + kr)w) \bmod q)(w \bmod q)) \bmod q \equiv$$
$$\equiv (((xs) \bmod q)(w \bmod q)) \bmod q \equiv (wxs) \bmod q \equiv$$
$$\equiv ((x \bmod q)((ws) \bmod q)) \bmod q \equiv x \bmod q.$$

Поскольку $0 < x < q$, получаем $x \bmod q \equiv x$. Что и требовалось доказать.

Теорема. В обозначениях табл. 4.1 имеет место равенство $v = r$.

Доказательство.

$$v \equiv ((g^{u_1} Y^{u_2}) \bmod p) \bmod q \equiv ((g^{(H(M)w) \bmod q} y^{(rw) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv ((g^{(H(M)w) \bmod q} g^{krw) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv ((g^{(H(M)w) \bmod q + (krw) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv ((g^{(H(M)w + krw) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv ((g^{(H(M) + kr)w) \bmod q}) \bmod p) \bmod q \equiv$$
$$\equiv (g^x \bmod p) \bmod q \equiv r.$$

Что и требовалось доказать.

Приложение Е

Примеры вычисления цифровой подписи согласно ДСТУ 4145–2002

Ниже приводятся выдержки из национального стандарта Украины “Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих”. Стандарт устанавливает механизм создания цифровой подписи, которая основывается на свойствах групп точек эллиптических кривых над полями $GF(2^m)$.

1 ОБЧИСЛЕННЯ ЗАГАЛЬНИХ ПАРАМЕТРІВ ЦИФРОВОГО ПІДПИСУ

Загальні параметри цифрового підпису можуть бути однаковими задля довільної кількості користувачів цифрового підпису. Цей розділ встановлює правила обчислення загальних параметрів цифрового підпису.

1.1 Вибір основного поля

Якщо використовують оптимальний нормальний базис, то основне поле треба обирати з-посеред полів $GF(2^m)$, степені яких наведено в табл. Е.1.

Таблиця Е.1 – Припустимі основні поля з оптимальним нормальним базисом

Степінь поля m	173	179	191	233	239	251	281
Степінь поля m	293	359	419	431	443	491	509

Для реалізації цього стандарту можна використовувати еліптичні криві, наведені в табл. Е.2.

Таблиця Е.2 – Рекомендовані еліптичні криві в поліноміальному базисі

№ пп.	m	Еліптична крива
1	163	A = 1 B = 5FF6108462A2DC8210AB403925E638A19C1455D21 n = 400000000000000000002BEC12BE2262D39BCF14D
2	167	A = 1 B = 6EE3CEEB230811759F20518A0930F1A4315A827DAC n = 3FFFFFFFFFFFFFFFFFFFFB12EBCC7D7F29FF7701F
3	173	A = 0 B = 108576C80499DB2FC16EDDF6853BBB278F6B6FB437D9 n = 8000000000000000000000189B4E67606E3825BB2831
4	179	A = 1 B = 4A6E0856526436F2F88DD07A341E32D04184572BFB710 n = 3FFFFFFFFFFFFFFFFFFFFFB981960435FE5AB64236EF
5	191	A = 1 B = 7BC86E2102902EC4D5890E8B6B4981FF27E0482750FEFC03 n = 40000000000000000000000069A779CAC1DABC6788F7474F
6	233	A = 1 B = 06973B15095675534C7CF7E64A21BD54EF5DD3B8A0326AA936ECE454D2C n = 1000000000000000000000000000013E974E72F8A6922031D2603CFE0D7

Дозволено задавати основне поле поліноміальним чи то оптимальним нормальним базисом. Якщо використовують поліноміальний базис, то основне поле

треба обирати з-посеред полів $GF(2^m)$, степені яких наведено в табл. Е.3. Поліноміальний базис задають примітивними тричленами чи примітивними п'ятичленами. Використовування примітивних многочленів, наведених у табл. Е.3, не є обов'язковим.

Таблиця Е.3 – Припустимі основні поля з поліноміальним базисом і рекомендовані примітивні многочлени

№ пп.	Степінь поля m	Примітивний многочлен	№ пп.	Степінь поля m	Примітивний многочлен
1	163	$x^{163}+x^7+x^6+x^3+1$	31	337	$x^{337}+x^{10}+x^6+x+1$
2	167	$x^{167}+x^6+1$	32	347	$x^{347}+x^{17}+x^6+x+1$
3	173	$x^{173}+x^{10}+x^2+x+1$	33	349	$x^{349}+x^6+x^5+x^2+1$
4	179	$x^{179}+x^4+x^2+x+1$	34	353	$x^{353}+x^{26}+x^7+x^3+1$
5	181	$x^{181}+x^7+x^6+x+1$	35	359	$x^{359}+x^{18}+x^4+x^2+1$
6	191	$x^{191}+x^9+1$	36	367	$x^{367}+x^{21}+1$
7	193	$x^{193}+x^{15}+1$	37	373	$x^{373}+x^9+x^6+x+1$
8	197	$x^{197}+x^{21}+x^2+x+1$	38	379	$x^{379}+x^{17}+x^6+x+1$
9	199	$x^{199}+x^{11}+x^2+x+1$	39	383	$x^{383}+x^9+x^5+x+1$
10	211	$x^{211}+x^{12}+x^6+x+1$	40	389	$x^{389}+x^{17}+x^{10}+x+1$
11	223	$x^{223}+x^{12}+x^2+x+1$	41	397	$x^{397}+x^{22}+x^3+x+1$
12	227	$x^{227}+x^{21}+x^2+x+1$	42	401	$x^{401}+x^{29}+x^4+x+1$
13	229	$x^{229}+x^{21}+x^2+x+1$	43	409	$x^{409}+x^{15}+x^6+x+1$
14	233	$x^{233}+x^9+x^4+x+1$	44	419	$x^{419}+x^{21}+x^{14}+x+1$
15	239	$x^{239}+x^{15}+x^2+x+1$	45	421	$x^{421}+x^7+x^4+x+1$
16	241	$x^{241}+x^{15}+x^4+x+1$	46	431	$x^{431}+x^6+x^3+x+1$
17	251	$x^{251}+x^{14}+x^4+x+1$	47	433	$x^{433}+x^{15}+x^5+x+1$
18	257	$x^{257}+x^{12}+1$	48	439	$x^{439}+x^8++x^3+x^2+1$
19	263	$x^{263}+x^{27}+x^2+x+1$	49	443	$x^{443}+x^{28}+x^3+x+1$
20	269	$x^{269}+x^7+x^6+x+1$	50	449	$x^{449}+x^{25}+x^5+x^3+1$
21	271	$x^{271}+x^{16}+x^3+x+1$	51	457	$x^{457}+x^{16}+1$
22	277	$x^{277}+x^{23}+x^3+x^2+1$	52	461	$x^{461}+x^{23}+x^4+x+1$
23	281	$x^{281}+x^9+x^4+x+1$	53	463	$x^{463}+x^{24}+x^3+x+1$
24	283	$x^{283}+x^{26}+x^9+x+1$	54	467	$x^{467}+x^{28}+x^3+x+1$
25	293	$x^{293}+x^{11}+x^6+x+1$	55	479	$x^{479}+x^{25}+x^6+x+1$
26	307	$x^{307}+x^8+x^4+x^2+1$	56	487	$x^{487}+x^{15}+x^2+x+1$
27	311	$x^{311}+x^{29}+x^4+x+1$	57	491	$x^{491}+x^{17}+x^6+x^2+1$
28	313	$x^{313}+x^7+x^3+x+1$	58	499	$x^{499}+x^{29}+x^6+x^2+1$
29	317	$x^{317}+x^9+x^5+x^2+1$	59	503	$x^{503}+x^3+1$
30	331	$x^{331}+x^{12}+x^5+x^2+1$	60	509	$x^{509}+x^{23}+x^3+x^2+1$

1.2 Перетворювання даних

1.2.1 Перетворювання елемента основного поля на ціле число

У цьому підрозділі встановлено алгоритм перетворювання елемента основного поля $x \in GF(2^m)$ на ціле число a.

Вхідні дані алгоритму: елемент основного поля $x \in GF(2^m)$, $x = (x_{m-1}, ..., x_0)$, і порядок базової точки еліптичної кривої n.

Результат виконання алгоритму – ціле число $a = (a_{L-1}, ..., a_0)$, що задовольняє умову $L = L(a) < L(n)$.

Алгоритм перетворювання елемента основного поля на ціле число:

1 Якщо елемент x основного поля дорівнює 0, то $a \leftarrow 0$ $L = L(a) \leftarrow 1$, кінець алгоритму.

2 Обчислюють ціле число $k = L(n) - 1$.

3 Приймають $a_i = x_i$ для $i = 0, ..., k - 1$ і знаходять j, що дорівнює найбільшому індексові i, за якого $a_i = 1$. Якщо такого індексу нема, то приймають $a \leftarrow 0$ і закінчують виконування алгоритму.

4 Двійковий рядок $(a_j, ..., a_0)$ довжини $L = L(a) = j + 1$ зображує ціле число a, яке є результат виконання алгоритму.

1.2.2 Перетворювання геш-коду на елемент основного поля

У цьому підрозділі встановлено алгоритм перетворювання результату обчислення функції гешування $(h_{L_H-1}, ..., h_0)$ на елемент основного поля $x \in GF(2^m)$, $x = (x_{m-1}, ..., x_0)$.

Вхідні дані алгоритму: геш-код $(h_{L_H-1}, ..., h_0)$.

Результат виконання алгоритму – елемент основного поля $x \in GF(2^m)$, $x = (x_{m-1},..., x_0)$.

Алгоритм перетворювання результату обчислення функції гешування на елемент основного поля:

1 Обчислюють ціле число $k = \min(m, L_H)$.

2 Приймають $x_i = h_i$ для $i = 0, ..., k - 1$.

3 Якщо $k < m$, то приймають $x_i = 0$ для $i = 0, ..., m - 1$.

4 Двійковий рядок $(x_{m-1}, ..., x_0)$ зображує елемент x основного поля, який є результат виконання алгоритму.

1.2.3 Перетворювання пари цілих чисел на цифровий підпис

У цьому підрозділі встановлено алгоритм перетворювання пари цілих чисел (r, s), які задовольняють умови $0 < r < n$, $0 < s < n$, на цифровий підпис $D = (D_{L_D-1}, ..., D_0)$.

Вхідні дані алгоритму: пара цілих чисел (r, s) у двійковому зображенні: $r = (r_{L(r)-1}, ..., r_0)$, $s = (s_{L(s)-1}, ..., s_0)$, $0 < r < n$, $0 < s < n$, довжина цифрового підпису L_D: $L_D \geq 2L(n)$, L_D є кратне до 16.

Результат виконання алгоритму – цифровий підпис $D = (D_{L_D-1}, ..., D_0)$ довжини L_D.

Алгоритм перетворювання пари цілих чисел на цифровий підпис:

1 Приймають $l = L_D/2$.

2 Утворюють двійковий рядок R за правилом:

2.1 приймають $R_i = r_i$ для $i = 0, ..., L(r) - 1$;

2.2 приймають $R_i = 0$ для $i = L(r), ..., l - 1$.

3 Утворюють двійковий рядок S за правилом:

3.1 приймають $S_i = s_i$ для $i = 0, ..., L(s) - 1$.

3.2 приймають $S_i = 0$ для $i = L(s), ..., l - 1$.

4 Рядок D є конкатенація двох рядків $S \| R$.

5 Двійковий рядок $D = (D_{L_D-1}, ..., D_0)$ довжини L_D є результат виконання алгоритму.

1.2.4 Перетворювання двійкового рядка на пару цілих чисел

Цей підрозділ встановлює алгоритм перетворювання двійкового рядка D парної довжини L_D на пару цілих чисел $r = (r_{L(r)-1}, ..., r_0)$, $s = (s_{L(s)-1}, ..., s_0)$.

Вхідні дані алгоритму: двійковий рядок $D = (D_{L_D-1}, ..., D_0)$ парної довжини L_D.

Результат виконання алгоритму – пара цілих чисел $r = (r_{L(r)-1}, ..., r_0)$ та $s = (s_{L(s)-1}, ..., s_0)$.

Алгоритм перетворювання двійкового рядка на пару цілих чисел:

1 Обчислюють ціле число $l = L_D/2$.

2 Приймають $r_i = D_i$ для $i = 0, ..., l - 1$.

3 Визначають у j як найбільше i, $i = 0, ..., l - 1$, для якого $r_i = 1$.

4 Якщо такого індексу нема, то $r \leftarrow 0$, $j = 0$ і переходять до кроку 6.

5 Двійковий рядок $(r_{L(r)-1}, ..., r_0)$, $L(r) = j + 1$, зображає ціле число r.

6 Приймають $s_i = D_{i+l}$ для $i = 0, ..., l - 1$.

7 Визначають індекс j як найбільше i, $i = 0, ..., l - 1$, для якого $s_i = 1$.

8 Якщо такого індексу нема, то $s \leftarrow 0$, $j = 0$ і переходять до кроку 10.

9 Двійковий рядок $(s_{L(s)-1}, ..., s_0)$, $L(s) = j + 1$, зображає ціле число s.

10 Пара цілих чисел r та s є результат виконання алгоритму.

2 ПРИКЛАДИ ОБЧИСЛЕНЬ ЦИФРОВОГО ПІДПИСУ

2.1 Обчислення й перевіряння цифрового підпису в поліноміальному базисі

У цьому розділі наведено приклади обчислення й перевіряння цифрового підпису з використанням поліноміального та оптимального нормального базисів. У прикладах обчислень двійкові рядки наведено у вигляді рядків шістнадцяткових цифр: двійковий рядок у разі потреби доповнюють ліворуч у такий спосіб, щоби довжина рядка стала кратною до чотирьох, потім рядок поділяють на групи по чотири двійкових розряди, а кожну таку групу замінюють на шістнадцяткову цифру, що відповідає цій групі двійкових символів.

Вибір загальних параметрів. За основне поле використовують скінченне поле $GF(2^{163})$. Елементи цього поля зображають у поліноміальному базисі, що відповідає примітивному многочленові $x^{163} + x^7 + x^6 + x^3 + 1$ (див. табл. Е.3).

Використовується еліптична крива над полем $GF(2^{163})$:

$$y^2 + xy = x^3 + x^2 + 5FF6108462A2DC8210AB403925E638A19C1455D21.$$

Порядок цієї еліптичної кривої ділиться на просте число

$$n = 400000000000000000002BEC12BE2262D39BCF14D,$$

яке є порядком базової точки.

Обчислення базової точки еліптичної кривої здійснюють наступним чином обчислюємо випадкову точку еліптичної кривої

$$P = (x_P, y_P) = (72D867F93A93AC27DF9FF01AFFE74885C8C540420,$$
$$0224A9C3947852B97C5599D5F4AB81122ADC3FD9B).$$

Оскільки $nP = O$, то точка P – шукана базова точка еліптичної кривої.

За особистий ключ цифрового підпису візьмемо ціле число

$$d = 183F60FDF7951FF47D67193F8D073790C1C9B5A3E.$$

Обчислимо відкритий ключ цифрового підпису, що відповідає обраному особистому ключу:

$$Q = -dP = (x_Q, y_Q) = (057DE7FDE023FF929CB6AC785CE4B79CF64ABDC2DA,$$
$$3E85444324BCF06AD85ABF6AD7B5F34770532B9AA).$$

Нехай використовується довжина цифрового підпису $L_D = 512$.

Припустімо, що функцію гешування обрано згідно з ГОСТ 34.311–95 [45] ($iH = 1$) і її використовують за промовчанням. У цьому разі $L_H = 256$. За промовчанням приймемо також, що iH не передається.

Обчислення цифрового підпису. Обчислимо геш-функцію за повідомленням T. Нехай результат гешування буде

$$H(T) = 09C9C44277910C9AAEE486883A2EB95B7180166DDF73532EEB76EDAEF52247FF.$$

Перетворимо результат обчислення функції гешування $H(T)$ на елемент основного поля згідно з 1.2.2. Перетворення цього рядка на елемент основного поля полягає у виділенні з цього рядка $\min(m, L_H) = 163$ молодших розрядів. У результаті перетворення отримаємо елемент основного поля

$$h = 03A2EB95B7180166DDF73532EEB76EDAEF52247FF.$$

Нехай ціле число

$$e = 1025E40BD97DB012B7A1D79DE8E12932D247F61C6.$$

Обчислимо точку eP:

$$eP = (x_{eP}, y_{eP}) = (42A7D756D70E1C9BA62D2CB43707C35204EF3C67C,$$
$$5310AE5E560464A95DC80286F17EB762EC544B15B).$$

Тоді F_e дорівнює координаті x_{eP} цієї точки:

$$F_e = 42A7D756D70E1C9BA62D2CB43707C35204EF3C67C.$$

Обчислимо добуток елементів основного поля

$$y = hF_e = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.$$

Перетворимо елемент основного поля y на ціле число r згідно з 1.2.1:

r = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Обчислимо ціле число

$s = e + dr \bmod n$ = 2100D86957331832B8E8C230F5BD6A332B3615ACA.

Перетворимо пару цілих чисел (r, s) на цифровий підпис D згідно з 1.2.3:

D = 0000000000000000000000002100D86957331832B8E8C230F5BD6A332B3615ACA
0000000000000000000000000274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Підписане повідомлення довжини $L = L_T + L_D$ має вигляд $T \| D$.

Перевіряння цифрового підпису. Перевіримо цифровий підпис, обчислений вище. Ппи перевірянні цифрового підпису використовують ті самі загальні параметри, обчислений вище відкритий ключ та геш-функцію за промовчанням (iH = 1, L_H = 256, iH не передається).

Перевіряємо цифровий підпис:

D = 0000000000000000000000002100D86957331832B8E8C230F5BD6A332B3615ACA
0000000000000000000000000274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Перевіримо довжину цифрового підпису: L_D = 512, тобто чи є це число кратне до 16 і більше за подвоєну довжину двійкового зображення порядку базової точки n.

Обчислюємо $L_T = L - L_D$. Вважаємо, що підписаний текст прийнято без спотворень, тому $L_T > 0$.

Обчислюємо $H(T)$. Підписаний текст прийнято без спотворень, тому результат обчислення функції гешування є, як і за обчислення цифрового підпису,

$H(T)$ = 09C9C44277910C9AAEE486883A2EB95B7180166DDF73532EEB76EDAEF52247F
F.

Перетворюємо результат обчислення функції гешування на елемент основного поля згідно з 1.2.2:

h = 03A2EB95B7180166DDF73532EEB76EDAEF52247FF.

Перетворюємо цифровий підпис на пару цілих чисел (r, s) згідно з 1.2.4:

r = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7;
s = 2100D86957331832B8E8C230F5BD6A332B3615ACA.

Переконуємося, що $0 < r < n$ та $0 < s < n$.

Обчислюємо точку еліптичної кривої

$R = sP + rQ = (x_R, y_R)$ = (42A7D756D70E1C9BA62D2CB43707C35204EF3C67C,
5310AE5E560464A95DC80286F17EB762EC544B15B).

Обчислюємо елемент основного поля

$y = hx_R$ = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Перетворюємо елемент основного поля y на ціле число $\tilde{r}$ згідно з 1.2.1:

$\tilde{r}$ = 274EA2C0CAA014A0D80A424F59ADE7A93068D08A7.

Оскільки $r = \tilde{r}$, то підпис є дійсний.

Учебное издание

Онацкий Алексей Витальевич
Йона Лариса Григорьевна

АСИММЕТРИЧНЫЕ МЕТОДЫ ШИФРОВАНИЯ

Учебное пособие

Редактор *И. В. Ращупкина*
Компьютерная верстка *Ж. А. Гардыман*