Министерство связи и массовых коммуникаций Российской Федерации

Государственное образовательное учреждение
высшего профессионального образования

ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ

ЭЛЕКТРОННАЯ БИБЛИОТЕЧНАЯ СИСТЕМА

Самара

Федеральное агентство связи

Государственное образовательное учреждение высшего профессионального образования

Поволжская государственная академия телекоммуникаций и информатики

Кафедра передачи дискретных сообщений

Задачи и методические указания для практических

занятий по дисциплине

СРЕДСТВА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В СЕТЯХ ПЕРЕДАЧИ ДАННЫХ

для студентов, обучающихся по специальностям 210403,210404,210406

Составители: к.т.н., доцент Крыжановский А.В.

к.т.н., доцент Киреева Н.В.

к.т.н., доцент Пугин В.В.

Редактор: д.т.н., профессор Лихтциндер Б.Я.

Рецензент: д.т.н., профессор Карташевский В.Г

Самара 2008

Задачи и методические указания для практических по дисциплине « Средства обеспечения информационной безопасности в сетях передачи данных» /Сост. к.т.н., доцент А.В.Крыжановский, к.т.н., доцент Н.В.Киреева, к.т.н., доцент В.В.Пугин – Самара, 2008-61 с.,ил.

Приведены краткие теоретические сведения, тексты задач и решения к ним по основным аспектам информационной безопасности: симметричные и асимметричные криптосистемы, политика безопасности, электронная цифровая подпись, распределение ключей в компьютерной сети, протоколы идентификации и аутентификации.

Методические разработки утверждены на заседании кафедры ПДС 12.03.2008 г. протокол № 5.

Редактор – д.т.н., профессор Б.Я.Лихтциндер
Рецензент – д.т.н., профессор В.Г. Карташевский

Содержание

Занятие 1

Традиционные симметричные криптосистемы

1.1 Основные понятия и определения

Большинство средств защиты информации базируется на использовании криптографических шифров и процедур шифрования-расшифрования. В соответствии со стандартом ГОСТ 28147-89 под шифром понимают совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, задаваемых ключом и алгоритмом криптографического преобразования.

Ключ-это конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор только одного варианта из всех возможных для данного алгоритма.

Основной характеристикой шифра является *криптостойкость*, которая определяет его стойкость к раскрытию методами криптоанализа. Обычно эта характеристика определяется интервалом времени, необходимым для раскрытия шифра.

К шифрам, используемым для криптографической защиты информации, предъявляется ряд требований:

- достаточная криптостойкость (надёжность закрытия данных);
- простота процедур шифрования и расшифрования;
- незначительная избыточность информации за счет шифрования;
- нечувствительность к небольшим ошибкам шифрования и др.

В той или иной мере этим требованиям отвечают:

- шифры перестановок;
- шифры замены;
- шифры гаммирования;
- шифры, основанные на аналитических преобразованиях шифруемых данных.

Шифрование перестановкой заключается в том, что символы шифруемого текста переставляются по определенному правилу в пределах некоторого блока

этого текста. При достаточной длине блока, в пределах которого осуществляется перестановка, и сложном неповторяющемся порядке перестановки можно достигнуть приемлемой для простых практических приложений стойкости шифра.

Шифрование заменой (подстановкой) заключается в том, что символы шифруемого текста заменяются символами того же или другого алфавита в соответствии с заранее обусловленной схемой замены.

Шифрование гаммированием заключается в том, что символы шифруемого текста складываются с символами некоторой случайной последовательности, именуемой *гаммой шифра.* Стойкость шифрования определяется, в основном, длиной (периодом) неповторяющейся части гаммы шифра. Поскольку с помощью ЭВМ можно генерировать практически бесконечную гамму шифра, то данный способ является одним из основных для шифрования информации в автоматизированных системах.

Шифрование аналитическим преобразованием заключается в том, что шифруемый текст преобразуется по некоторому аналитическому правилу (формуле).

Например, можно использовать правило умножения вектора на матрицу, причем умножаемая матрица является ключом шифрования (поэтому ее размер и содержание должны храниться в секрете),а символами умножаемого вектора последовательно служат символы шифруемого текста. Другим примером может служить использование так называемых однонаправленных функций для построения криптосистем с открытым ключом.

Процессы шифрования и расшифрования осуществляются в рамках некоторой криптосистемы. Характерной особенностью симметричной криптосистемы является применение одного и того же секретного ключа как при шифровании, так и при расшифровании сообщений.

1.2 Шифры перестановки

При шифровании перестановкой символы шифруемого текста переставляются по определенному правилу в пределах блока этого текста.

1.2.1 Шифрующие таблицы

Правила перестановки букв в сообщении задают шифрующие таблицы. В качестве ключа в шифрующих таблицах используются:

- размер таблицы;
- слово или фраза, задающие перестановку;
- особенности структуры таблицы.

Одним из самых примитивных табличных шифров перестановки является *простая перестановка*, для которой ключом служит размер таблицы.

Задача 1.1 Зашифровать методами простой перестановки сообщение:

ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ

Решение

Сообщение записывается в таблицу поочерёдно по столбцам. Считывание производится по строкам.

Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

Шифртекст записывается группами по пять букв:

ТНПВЕ ГЛЕАР АДОНР ТИЕЬВ ОМОБТ МПЧИР ЫСООЬ

Отправитель и получатель сообщения должны заранее условиться об общем ключе в виде размера таблицы. Объединение букв шифртекста в 5-буквенные группы не входит в ключ шифра и осуществляется для удобства записи несмыслового текста. При расшифровании действия выполняют в обратном порядке.

Несколько большей стойкостью к раскрытию обладает метод шифрования, называемый одиночной перестановкой по ключу. Этот метод отличается от предыдущего тем, что столбцы таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

Задача 1.2 Зашифровать сообщение задачи 1.1. методом одиночной перестановки по ключу. В качестве ключа использовать слово **П Е Л И К А Н.**

Решение

Составим две таблицы, заполненные текстом сообщения и ключевым словом. На рис. 1.1 представлена таблица до перестановки, а на рис. 1.2 – после перестановки.

Ключ→

П	**Е**	**Л**	**И**	**К**	**А**	**Н**
7	2	5	3	4	1	6
Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

Рисунок 1.1 – Таблица до перестановки

А	Е	И	К	Л	Н	П
1	2	3	4	5	6	7
Г	Н	В	Е	П	Л	Т
О	А	А	Д	Р	Н	Е
В	Т	Е	Ь	И	О	Р
П	О	Т	М	Б	Ч	М
О	Р	С	О	Ы	Ь	И

Рисунок 1.2 – Таблица после перестановки

В верхней строке верхней таблицы записан ключ, а номера под буквами ключа определены в соответствии с естественным порядком соответствующих букв ключа в алфавите. Если бы в ключе встретились одинаковые буквы, они бы были пронумерованы слева направо. В нижней таблице столбцы переставлены в соответствии с упорядоченными номерами букв ключа.

При считывании содержимого нижней таблицы по строкам и записи шифртекста группами по пять букв получим шифрованное сообщение:

ГНВЕП ЛТООА ДРНЕР ТЕЬИО РПОТМ БЧМОР СОЫЬИ

Для обеспечения дополнительной скрытности можно повторно зашифровать сообщение, которое уже прошло шифрование. Такой метод шифрования называется *двойной перестановкой*. В случае двойной перестановки столбцов и строк таблицы перестановки определяются отдельно для столбцов и отдельно для строк. Сначала в таблицу записывается текст сообщения, потом поочередно переставляются столбцы, а затем строки. При расшифровании порядок перестановок должен быть обратным.

Задача 1.3 Зашифровать методом *двойной перестановки* сообщение:

П РИ Л Е Т А Ю В О С Ь М О Г О

Для шифрования использовать ключи:

по столбцам- 4 1 3 2, по строкам- 3 1 4 2

Решение

Текст исходного сообщения записывается в таблицу 4×4, т.к. сообщение содержит 16 символов. Затем поочередно переставляются столбцы, а затем строки.

Исходная таблица

	1	2	3	4
1	Т	Ю	А	Е
2	О	О	Г	М
3	Р	Л	И	П
4	О	Ь	С	В

Перестановка столбцов

	4	1	3	2
3	П	Р	И	Л
1	Е	Т	А	Ю
4	В	О	С	Ь
2	М	О	Г	О

Перестановка столбцов

	1	2	3	4
3	Р	Л	И	П
1	Т	Ю	А	Е
4	О	Ь	С	В
2	О	О	Г	М

Если считывать шифртекст из правой таблицы построчно блоками по четыре буквы, то получится следующее:

ТЮАЕ ООГМ РЛИП ОЬСВ

Число вариантов двойной перестановки быстро возрастает при увеличении размера таблицы:

• для таблицы 3×3 - 36 вариантов;

• для таблицы 4×4 - 576 вариантов;

• для таблицы 5×5 - 14400 вариантов.

Однако двойная перестановка не отличается высокой стойкостью и сравнительно просто “взламывается” при любом размере таблицы шифрования.

1.2.2 Шифрование магическими квадратами

Магическими квадратами называют квадратные таблицы с вписанными в их клетки последовательными натуральными числами, начиная от 1, которые дают

в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число.

Шифруемый текст вписывали в магические квадраты в соответствии с нумерацией их клеток. Если затем выписать содержимое такой таблицы по строкам,то получится шифртекст, сформированный благодаря перестановке букв исходного сообщения. Считалось, что созданные с помощью магических квадратов шифртексты охраняет не только ключ, но и магическая сила.

Задача 1.4 Зашифровать сообщение:

П Р И Л Е Т А Ю В О С Ь М О Г О

с помощью магического квадрата. Считать шифртекст построчно блоками по четыре буквы.

Решение

Используем магический квадрат 4×4 и заполним его заданным сообщением. Вначале пронумеруем буквы:

П Р И Л Е Т А Ю В О С Ь М О Г О

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

О	И	Р	М
Е	О	С	Ю
В	Т	А	Ь
Л	Г	О	П

Рисунок 1.3 – Магический квадрат 4х4 и его заполнение сообщением

Шифртекст, получаемый при считывании содержимого правой таблицы по строкам, имеет вид:

Число магических квадратов быстро возрастает с увеличением размера квадрата. Существует только один магический квадрат размером 3x3. Количество магических квадратов 4x4 - 880, а 5x5 - 250000.

1.3 Шифры простой замены

При шифровании заменой (подстановкой) символы шифруемого текста заменяются символами того же или другого алфавита по заранее установленным правилам замены. В шифре простой замены каждый символ исходного текста заменяется символами того же алфавита одинаково на всем протяжении текста. Часто шифры простой замены называют шифрами одноалфавитной подстановки.

λ	ε	υ	ω	γ
ρ	ς	δ	σ	o
μ	η	β	ξ	τ
ψ	π	θ	α	κ
χ	υ		φ	ι

1.3.1 Шифрование на основе квадрата Полибия (полибианского квадрата)

Полибианский квадрат выглядит следующим образом:

Рисунок 1.4 – Полибианский квадрат

Для шифровании в этом полибианском квадрате находили очередную букву открытого текста и записывали в шифртекст букву, расположенную ниже неё в том же столбце. Если буква текста оказывалась в нижней строчке таблицы, то для шифртекста брали самую верхнюю букву из того же столбца.

Задача 1.5 Зашифровать сообщение $\tau\alpha\upsilon\rho o\delta$ с помощью полибианского квадрата.

Решение

Шифртекст имеет вид $\chi\varphi\delta\mu\tau\xi$

1.3.2 Система шифрования Цезаря

Шифр Цезаря является частным случаем шифра простой замены (одноалфавитной подстановки). При шифровании исходного текста каждая буква заменялась на другую букву того же алфавита по следующему правилу. Заменяющая буква определялась путём смещения по алфавиту от исходной буквы на *K* букв. При достижении конца алфавита выполнялся циклический переход к его началу. Цезарь использовал шифр замены при смещении *K=3*. Такой шифр замены можно задать таблицей подстановок, содержащей соответствующие пары букв открытого текста и шифртекста.

A → D	J → M	S → V
B → E	K → N	T → W
C → F	L → O	U → X
D → G	M → P	V → Y
E → H	N → Q	W → Z
F → I	O → R	X → A
G → J	P → S	Y → B
H → K	Q → T	Z → C
I → L	R → U	

Рисунок1.4 - Таблица подстановок Цезаря

Задача 1.6 Зашифровать послание Цезаря:**VENI VIDI VICI.**

Решение

Используя таблицу подстановок (рис.1.4) получаем шифртекст: **YHQL YLGL YLFL**

1.3.3 Система Цезаря с ключевым словом

Система шифрования Цезаря с ключевым словом является одноалфавитной системой подстановок. Особенностью этой системы является использование ключевого слова для смещения и изменения порядка символов в алфавите подстановок.

Задача 1.7 Зашифровать сообщение **SEND MORE MONEY** по системе Цезаря с ключевым словом **DIPLOMAT.**

Решение

Выберем некоторое число k, $0 \le k < 25$. Ключевое слово записывается под буквами алфавита, начиная с буквы, числовой код которой совпадает с выбранным числом k:

0 1 2 3 4 5 10 15 20 25
A B C D E F G H I J K L M NO P QR S T U V W XY Z
D I P L O M A T

Оставшиеся буквы алфавита подстановки записываются после ключевого слова в алфавитном порядке:

5
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
V W X Y Z D I P L O M A T B C E F G H J K N Q R S U

Теперь мы имеем подстановку для каждой буквы произвольного сообщения.

Исходное сообщение **SEND MORE MONEY**

шифруется как **HZBY TCGZ TCBZS**

Разновидностью рассмотренной системы, является система, в которой требование о различии всех букв ключевого слова не является обязательным. В этом случае ключевое слово (или фраза) записывается без повторения одинаковых букв.

Задача 1. 8 Сформировать таблицу подстановок в системе с ключевой фразой

КАК ДЫМ ОТЕЧЕСТВА НАМ СЛАДОК И ПРИЯТЕН

Полагая k = 3 и исключая повторяющиеся буквы в ключевой фразе, получим следующую систему подстановок:

0 3

А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ы Ъ Э Ю Я

Ъ Э Ю К А Д Ы М О Т Е Ч С В Н Л И П Р Я Б Г Ж З Й У Ф Х Ц Ш Щ Ь

Достоинством системы Цезаря с ключевым словом является то, что количество возможных ключевых слов практически неисчерпаемо. Недостатком этой системы является возможность взлома шифртекста на основе анализа частот появления букв.

1.3.4 Шифрующие таблицы Трисемуса

В 1508 г. аббат из Германии Иоганн Трисемус написал печатную работу по криптологии под названием "Полиграфия". В этой книге он впервые систематически описал применение шифрующих таблиц, заполненных алфавитом в случайном порядке. Для получения такого шифра замены обычно использовались таблица для записи букв алфавита и ключевое слово (или фраза). В таблицу сначала вписывалось по строкам ключевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. При шифровании находят в этой таблице очередную букву открытого текста и записывают в шифртекст букву, расположенную ниже неё в том же столбце. Если буква текста оказывается в нижней строке таблицы, тогда для шифртекста берут самая верхнюю букву из того же столбца**.**

Задача 1.9 Зашифровать таблицей Трисемуса сообщение:

ВЫЛЕТАЕМ ПЯТОГО

Решение

Для русского алфавита шифрующая таблица может иметь размер 4×8. Шифрующая таблица выглядит так:

Б	А	Н	Д	Е	Р	О	Л
Ь	В	Г	Ж	З	И	Й	К
М	П	С	Т	У	Ф	Х	Ц
Ч	Ш	Щ	Ы	Ъ	Э	Ю	Я

Рисунок 1.5 - Шифрующая таблица Трисемуса с ключевым словом **БАНДЕРОЛЬ**

Используя эту таблицу в соответствии с вышеизложенной методикой, получаем шифртекст

ПДКЗЫВЗЧШЛЫЙСЙ.

Такие табличные шифры называются монограммными, так как шифрование выполняется по одной букве. Трисемус первым заметил, что шифрующие таблицы позволяют шифровать сразу по две буквы. Такие шифры называются *биграммными.*

1.3.5 Биграммный шифр Плейфейра

Шифр Плейфейра, изобретенный в 1854 г., является наиболее известным биграммным шифром замены. Он применялся Великобританией во время первой мировой войны. Основой шифра Плейфейра является шифрующая таблица со случайно расположенными буквами алфавита исходных сообщений.

Для удобства запоминания шифрующей таблицы отправителем и получателем сообщений можно использовать ключевое слово (или фразу) при заполнении начальных строк таблицы. В целом структура шифрующей таблицы системы Плейфейра полностью аналогична структуре шифрующей таблицы Трисемуса. Поэтому для пояснения процедур шифрования и расшифрования в системе Плейфейра воспользуемся шифрующей таблицей Трисемуса из предыдущей задачи (рис. 1.5).

Процедура шифрования включает следующие шаги.

1. Открытый текст исходного сообщения разбивается на пары букв (биграммы). Текст должен иметь четное количество букв и в нем не должно быть биграмм, содержащих две одинаковые буквы. Если эти требования не выполнены, то текст модифицируется даже из-за незначительных орфографических ошибок.
2. Последовательность биграмм открытого текста преобразуется с помощью шифрующей таблицы в последовательность биграмм шифртекста по следующим правилам:
   2а.Если обе буквы биграммы открытого текста не попадают на одну строку или столбец (как, например, буквы А и Й в табл. на рис.2.6), тогда находят буквы в углах прямоугольника, определяемого данной парой букв. (В нашем примере это – буквы АЙОВ. Пара букв АЙ отображается в пару ОВ. Последовательность букв в биграмме шифртекста должна быть зеркально расположенной по отношению к последовательности букв в биграмме открытого текста.)
   2б.Если обе буквы биграммы открытого текста принадлежат одному столбцу таблицы, то буквами шифртекста считаются буквы, которые лежат под ними. (Например, биграмма НС дает биграмму шифртекста ГЩ.) Если при этом буква открытого текста находится в нижней строке, то для шифртекста берется соответствующая буква из верхней строки того же столбца. (Например, биграмма ВШ дает биграмму шифртекста ПА.)

2в. Если обе буквы биграммы открытого текста принадлежат одной строке таблицы, то буквами шифртекста считаются буквы, которые лежат справа от них. (Например, биграмма НО дает биграмму шифртекста ДЛ.) Если при этом буква открытого текста находится в крайнем правом столбце, то для шифра берут соответствующую букву из левого столбца в той же строке. (Например, биграмма ФЦ дает биграмму шифртекста ХМ.)

Задача 1.10 Зашифровать биграммным шифром Плейфера текст

ВСЕ ТАЙНОЕ СТАНЕТ ЯВНЫМ

Решение

Разобьём этот текст на биграммы:

ВС ЕТ АЙ НО ЕС ТА НЕ ТЯ ВН ЫМ

Данная последовательность биграмм открытого текста преобразуется с помощью шифрующей таблицы (рис. 1.5) в следующую последовательность биграмм шифртекста

ГП ДУ ОВ ДЛ НУ ПД ДР ЦЫ ГА ЧТ

При расшифровании применяется обратный порядок действий.

Шифрование биграммами резко повышает стойкость шифров к вскрытию. Хотя книга И.Трисемуса "Полиграфия" была относительно доступной, описанные в ней идеи получили признание лишь спустя три столетия. По всей вероятности, это было обусловлено плохой осведомленностью криптографов о работах богослова и библиофила Трисемуса в области криптографии.

Занятие 2

Методы шифрования

2.1 Метод перестановок на основе маршрутов Гамильтона

Этот метод реализуется путем выполнения следующих шагов.

Шаг 1. Исходный текст разбивается на блоки. Если длина шифруемого текста не кратна длине блока, то на свободные места последнего блока помещаются служебные символы-заполнители(например,*)

Шаг 2. Символами блока заполняется таблица, в которой для каждого порядкового номера символа в блоке отводится вполне определенное место (рис. 2.1).

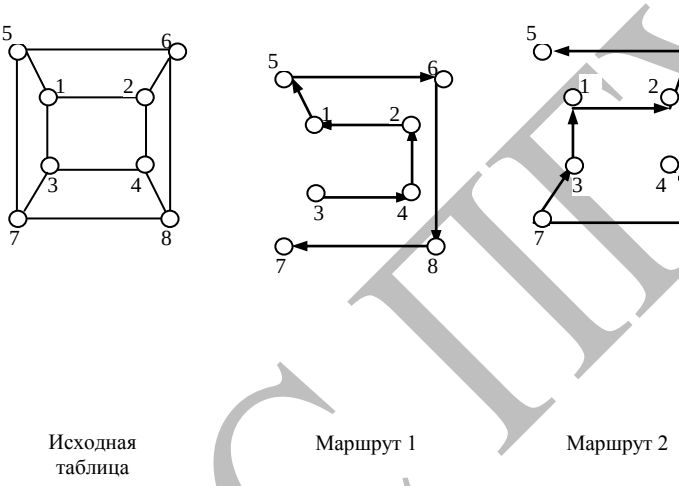


Рисунок 2.1 - Вариант 8-элементной таблицы и маршрутов Гамильтона

Шаг 3. Считывание символов из таблицы осуществляется по одному из маршрутов. Увеличение числа маршрутов повышает криптостойкость шифра. Маршруты выбирают либо последовательно, либо их очерёдность задаётся ключом *K*.

Шаг 4. Зашифрованная последовательность символов разбивается на блоки фиксированной длины *L*. Величина *L* может отличаться от длины блоков, на которые разбивается исходный текст на шаге 1.

Расшифрование производится в обратном порядке.

Задача 2.1 Требуется зашифровать текст T_0 =**<МЕТОДЫ ПЕРЕСТАНОВКИ>**. Ключ и длины зашифрованных блоков равны: *K=<2,1,1>, L=4.* Для шифрования использовать таблицу и два маршрута, представленные на рис.2.1.

Решение

Воспользуемся вышеизложенной методикой построения шифра по шагам.

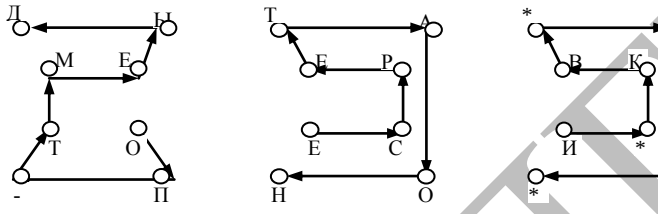
Шаг 1. Исходный текст разбивается на 3 блока:

Блок $Б_1$=**<МЕТОДЫ П>**

Блок $Б_2$=**<ЕРЕС ТАНО>**

$Б_3$=**<ВКИ*****>**

Шаг 2. Заполняется 3 матрицы с маршрутами 2,1,1 (рис.2.2.)



Маршрут 2 Маршрут 1 Маршрут 1

Рисунок 2.2 - Шифрование с помощью маршрутов Гамильтона

Шаг 3. Получение шифртекста путём расстановки символов в соответствии с маршрутами.

T_1 =**<ОП_ТМЕЫДЕСРЕТАОНИ*КВ****>**

Шаг 4. Разбиение на блоки шифртекста

T_1 =**<ОП_Т МЕЫД ЕСРЕ ТАОН И*КВ ****>**

Возможно применение и других маршрутов.

2.2 Аналитические методы шифрования

Среди аналитических методов наибольшее распространение получили методы, основанные на использовании матриц. Зашифрование *К*-го блока исходной информации, представленного в виде вектора $B_к = \|в_{ij}\|$ осуществляется путём перемножения матрицы ключа $A = \|a_{ij}\|$ и вектора B_k. В результате перемножения получается блок шифртекста в виде вектора $C_k = \|c_i\|$, где элементы вектора C_k определяются по формуле:

$$C_i = \sum_{j=1} a_{ij} b_j \,, \qquad (2.1)$$

Расшифрование информации осуществляется путём последовательного перемножения векторов C_k и обратной матрицы A^{-1}.

Задача 2.2 Требуется зашифровать слово T_0=**<ЗАБАВА>** с помощью матрицы-ключа *А.*

$$A = \begin{bmatrix} 1 & 4 & 8 \\ 3 & 7 & 2 \\ 6 & 9 & 5 \end{bmatrix}$$

Решение

1.Определим числовой эквивалент исходного слова как последовательность соответствующих порядковых номеров букв слова T_0:

$$T_Э = <8,1,2,1,3,1>$$

2.Разобьём $T_э$ на два вектора $B_1 = [8,1,2]$ и $B_2 = [1,3,1]$

3. Умножим матрицу A на векторы B_1 и B_2:

$$C_1 = \begin{bmatrix} 1 & 4 & 8 \\ 3 & 7 & 2 \\ 6 & 9 & 5 \end{bmatrix} \begin{bmatrix} 8 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 28 \\ 35 \\ 67 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 1 & 4 & 8 \\ 3 & 7 & 2 \\ 6 & 9 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 21 \\ 26 \\ 38 \end{bmatrix}$$

4. Зашифрованное слово запишем в виде последовательности чисел T_1=<28,35,67,21,26,38>.

Задача 2.3 Расшифровать текст, полученный в задаче 2.2.

Решение

1.Вычисляется определитель $|A| = -115$.

2.Определяется присоединённая матрица A^*, каждый элемент которой является алгебраическим дополнением элемента a_{ij} матрицы *A:*

$$A^* = \begin{bmatrix} 17 & -3 & -15 \\ 52 & -43 & 15 \\ -48 & 22 & -5 \end{bmatrix}$$

3.Получается транспонированная матрица A^T

$$A^T = \begin{bmatrix} 17 & 52 & -48 \\ -3 & -43 & 22 \\ -15 & 15 & -5 \end{bmatrix}$$

4.Вычисляется обратная матрица A^{-1} по формуле:

$$A^{-1} = \frac{A^T}{|A|},$$

В результате вычислений обратная матрица имеет вид:

$$A^{-1} = \begin{bmatrix} -17/115 & -52/115 & 48/115 \\ 3/115 & 43/115 & -22/115 \\ 15/115 & -15/115 & 5/115 \end{bmatrix}$$

5.Определяются векторы B_1 и B_2:

$$B_1 = A^{-1}C_1; \quad B_2 = A^{-1}C_2$$

$$B_1 = \begin{bmatrix} -17/115 & -52/115 & 48/115 \\ 3/115 & 43/115 & -22/115 \\ 15/115 & -15/115 & 5/115 \end{bmatrix} \begin{bmatrix} 28 \\ 35 \\ 67 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \\ 2 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} -17/115 & -52/115 & 48/115 \\ 3/115 & 43/115 & -22/115 \\ 15/115 & -15/115 & 5/115 \end{bmatrix} \begin{bmatrix} 21 \\ 26 \\ 38 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$$

6.Получаем числовой эквивалент расшифрованного слова:

$T_з$=<8,1,2,1,3,1>, который заменяется символами, в результате получается исходное слово

T_o =**<ЗАБАВА>**

Занятие 3

Асимметричная криптосистема RSA.

Расширенный алгоритм Евклида

1. Выбирают два больших простых числа *p*и *q*. Для большей криптостойкости *p* и *q* выбирают равной длины.
2. Вычисляют произведение: $n=pq$
3. Вычисляют $z=(p-1)(q-1)$ и выбирают число e взаимно простое с z, т.е. НОД (e,z)=1.
4. Для вычисления закрытого (секретного) ключа d решается сравнение $ed \equiv 1 \bmod z$ (1)

Решение (1) имеет вид $d = (-1)^{k-1} Q_{k-1}$

Для вычисления ключа d воспользуемся расширенным алгоритмом Евклида. Для этого число $\frac{e}{z}$ обращается в конечную цепную дробь:

$$e = zq_0 + e_1$$
$$z = e_1 q_1 + e_2$$
$$e_1 = e_2 q_2 + e_3$$
$$e_2 = e_3 q_3 + e_4$$
$$\ldots\ldots\ldots\ldots$$
$$e_{k-2} = e_{k-1} q_{k-1} + e_k$$
$$e_{k-1} = e_k q_k + 0$$

Цепная дробь имеет вид: $\frac{e}{z} = [q_0, q_1, q_2, ..., q_k]$, а последовательности $\{P_n\}$ и $\{Q_n\}$ числителей и знаменателей подходящих дробей к цепной дроби определяются рекуррентно:

$P_{-2} = 0, \quad P_{-1} = 1, \quad Q_{-2} = 1, \quad Q_{-1} = 0$.

$P_n = q_n P_{n-1} + P_{n-2}, \ n \geq 0$

$Q_n = q_n Q_{n-1} + Q_{n-2}, \ n \geq 0$

Их вычисления удобно оформить в виде таблицы:

n	-2	-1	0	1	2	……	k-1	k
q_n			q_0	q_1	q_2	……	q_{k-1}	q_k
P_n	0	1	P_0	P_1	P_2	……	P_{k-1}	P_k
Q_n	1	0	Q_0	Q_1	Q_2	……	Q_{k-1}	Q_k

Задача 3.1 Пусть выбраны простые числа $p = 47$ и $q = 71$ и открытый ключ $e = 71$.

Требуется выполнить шифрование и дешифрование в асимметричной криптосистеме RSA сообщения:

688 232 687 966 668 3

Укажите последовательность операций.

Решение

1. $z = (p-1)(q-1) = 46 \cdot 70 = 3220$

2. Найдём секретный ключ d в результате решения сравнения:

$$de \bmod z \equiv 1,$$

$$d \cdot 79 \bmod 3220 \equiv 1.$$

Воспользуемся расширенным алгоритмом Евклида:

79=3220*0+79,

3220=79*40+60,

79=60*1+19,

60=19*3+3,

19=3*6+1,

3=1*3+0.

Результаты вычислений сведём в таблицу:

n	-2	-1	0	1	2	3	4	5
q_n			0	40	1	3	6	3
P_n	0	1						
Q_n	1	0	1	40	41	163	1019	3220

Q_1 Q_2 Q_3 Q_4 Q_5.

$Q_0 = q_0 Q_{-1} + Q_{-2} = 0 \cdot 0 + 1 = 1$

$Q_1 = q_1 Q_0 + Q_{-1} = 40 \cdot 1 + 0 = 40$

$Q_2 = q_2 Q_1 + Q_0 = 1 \cdot 40 + 1 = 41$

$Q_3 = q_3 Q_2 + Q_1 = 3 \cdot 41 + 40 = 163$

$Q_4 = q_4 Q_3 + Q_2 = 6 \cdot 163 + 41 = 1019$

$Q_5 = q_5 Q_4 + Q_3 = 3 \cdot 1019 + 163 = 3220$

$\kappa=5 \quad d = (-1)^{k-1} Q_{k-1} = Q_4 = 1019$

В самом деле $79 \cdot 1019 \equiv 1 \bmod 3220$

$79 \cdot 1019 = 80501$,

$\frac{8051}{3220} = 25{,}0003$

$3220 \cdot 25 = 80500$

Следовательно, d=1019.

3. Разобьём сообщение на блоки m_i, которые должны иметь длину, меньшую, чем $n = pq = 47 \cdot 17 = 3337$. $m_1 = 668$, $m_2 = 232$, $m_3 = 687$, $m_4 = 966$, $m_5 = 668$, $m_6 = 003$

4. Затем шифруем блоки: $C_i = m_i^e \bmod n$

$C_1 = 688^{79} \bmod 3337 = 1570$, и т.д.

Получим криптограмму: C=($C_1, C_2, C_3, C_4, C_5, C_6$) =

=1570 2756 2091 2276 2423 0158

C_1 C_2 C_3 C_4 C_5 C_6.

5. Для дешифрования нужно выполнить возведение в степень, используя ключ дешифрования *d*, т.е. $m_i = C_i^d \bmod n$

$m_1 = (1570)^{1019} \bmod 3337 = 688$ и т.д.

Задача 3.2 Зашифровать и расшифровать сообщение CAB. Для простоты вычислений использовать небольшие числа: *p =3* , *q=11*, открытый ключ *e=7.* Для вычисления секретного ключа *d* воспользоваться расширенным алгоритмом Евклида.

Решение

Действия пользователя B- получателя сообщения.

1. Выбирает *p =3* и *q=11.*

2. Вычисляет модуль *n=p*q=3*11=33.*

3. Вычисляет значение функции Эйлера для *n* = 33:

$$z(33) = (p-1)(q-1) = 2*10 = 20.$$

Выбирает в качестве открытого ключа *e* произвольное число с учетом выполнения условий:

$$1 < e \le 20, \quad \text{НОД}(e, 20) = 1.$$

Пусть *e*= 7.

4. Вычисляет значение секретного ключа *d*, используя расширенный алгоритм Евклида при решении сравнения

$$d \equiv 7^{-1} \pmod{20}.$$

Решение дает *d*= 3.

5. Пересылает пользователю A (отправителю) пару чисел (*n* = 33, *e*= 7).

Действия пользователя A-отправителя сообщения.

6. Представляет шифруемое сообщение как последовательность целых чисел в диапазоне 0 ... 32. Пусть буква A представляется как число 1, буква B – как число 2, буква C – как число 3. Тогда сообщение CAB можно представить как последовательность чисел 312, т.е. $m_1 = 3$, $m_2 = 1$, $m_3 = 2$.

7. Шифрует текст, представленный в виде последовательности чисел m_1, m_2 и m_3, используя ключ e= 7 и n = 33, по формуле

$$m_i^e (\mathrm{mod} n) = m_i^7 (\mathrm{mod} 33)$$

Получает криптограмму:

$$C_1 = 3^7 \text{ (mod 33)} = 2187 \text{ (mod 33)} = 9,$$

$$C_2 = 1^7 \text{ (mod 33)} = 1 \text{ (mod 33)} = 1,$$

$$C_3 = 2^7 \text{ (mod 33)} = 128 \text{ (mod 33)} = 29.$$

Отправляет пользователю В криптограмму

$$C_1, C_2, C_3 = 9, 1, 29.$$

Действия пользователя В.

8. Расшифровывает принятую криптограмму C_1, C_2, C_3, используя секретный ключ d=3, по формуле

$$m_i = C_i^d (\mathrm{mod} n)$$

Получает:

$$m_1 = 9^3 \text{ (mod 33)} = 729 \text{ (mod 33)} = 3,$$

$$m_2 = 1^3 \text{ (mod 33)} = 1 \text{ (mod 33)} = 1,$$

$$m_3 = 29^3 \text{ (mod 33)} = 24389 \text{ (mod 33)} = 2.$$

Таким образом, восстановлено исходное сообщение: С А В

3 1 2

Занятие 4

Политика безопасности

Политика безопасности – это набор правил, которые регулируют управление, защиту и распределение ценной информации.

Политика безопасности включает:

1. Множество возможных операций над объектами.
2. Множество разрешенных операций для каждой пары «субъект-объект».

Существует два типа *политики безопасности:*

1. дискреционная; 2. мандатная.

Дискреционная политика определяет следующие правила.

1. Все субъекты (S) и объекты (O) должны быть идентифицированы.
2. Права доступа субъекта к объекту определяются на основе некоторого внешнего по отношению к системе правила.

Дискреционная политика задается *матрицей доступа:*

S / O	O_1	O_2	…	O_n
S_1	own, r, w			
S_2				
…				
S_m				

Множество прав доступа R=(own, r, w) включает права:

own – владение, r – чтение, w – запись.

Собственник (own) может определять права доступа других субъектов к данному объекту.

Мандатная политика определяет следующие правила.

1. Все субъекты и объекты должны быть идентифицированы.
2. Задан линейно упорядоченный набор меток секретности (решетка секретности).
3. Каждому объекту присваивается метка секретности, определяющая ценность содержащейся в нем информации.
4. Каждому субъекту присваивается метка секретности, определяющая уровень доверия к нему.

В мандатной политике вводится понятие информационного потока.

Информационный поток X $\rightarrow$ Y (X – источник, Y – получатель) разрешен тогда и только тогда, когда c(Y)≥c(X) по решетке секретности.

В системе с двумя доступами r и w мандатная политика определяет следующие правила доступа:

$$X \rightarrow Y \Leftrightarrow c(X) \geq c(Y)$$
$$X \rightarrow Y \Leftrightarrow c(X) \geq c(Y)$$

Задача 4.1 В сети применяется дискреционная политика безопасности и матрица доступа имеет вид:

	O_1	O_2
U_1	own r w	w
U_2		own r w

Обозначения:

U_1 – законный пользователь;

U_2 – злоумышленник;

O_1 – объект, содержащий ценную информацию;

O_2 – объект, содержащий программу «Троянский конь».

Показать, что злоумышленник U_2 может считать ценную информацию объекта O_1.

Решение

1. Злоумышленник U_2 является собственником (own) программы O_2.
2. Злоумышленник U_2 заставляет U_1 каким-либо образом запустить программу «Троянский конь». Для этого O_2 может, например, выглядеть интересной компьютерной игрой.
3. Если пользователь U_1 обратится к O_2, то он запустит скрытую программу T («Троянский конь»). Эта программа обладает правами доступа пользователя O_1, поскольку была им запущена.
4. Программа T, обладая правами доступа пользователя U_1, списывает (w) в объект O_2 информацию, содержащуюся в объекте O_1.
5. Злоумышленник U_2 владеет (own) объектом O_2 и пользуясь своими правами имеет возможность считать из объекта O_2 ценную информацию объекта O_1.

Задача 4.2 Требуется показать, что мандатная политика безопасности устойчива к атаке «Троянский конь».

Решение

Введем обозначения:

U_1 – законный (авторизованный) пользователь;

U_2 – злоумышленник;

O_1 – объект, содержащий ценную информацию;

O_2 – объект, содержащий программу «Троянский конь» T.

1. Предположим, что пользователи U_1 и U_2 находятся на разных уровнях доступа, т.е.

$$c(U_1) > c(U_2) \ , \qquad (4.1)$$

по решетке секретности.

2. Пусть пользователь U_1 размещает в объекте O_1 ценную информацию, т.е. $U_1 \xrightarrow{w} O_1$, при выполнении условия

$$c(O_1) \geq c(U_1) \ , \qquad (4.2)$$

3. «Троянский конь», содержащийся в объекте O_2, может считать информацию из O_1 при выполнении условия:

$$c(O_2) \geq c(O_1) \ , \qquad (4.3)$$

4. Объединяя (4.1), (4.2) и (4.3) получим:

$$\begin{cases} c(U_2) < c(U_1) \leq c(O_1) \\ c(O_2) \geq c(O_1) \end{cases} , \qquad (4.4)$$

Из (4.4) следует, что

$$c(O_2) > c(U_1). \qquad (4.5)$$

Следовательно, пользователь U_2 не имеет доступа к объекту O_2.

Поэтому считывание информации из объекта O_1 и запись ее в объект O_2 для злоумышленника U_2 являются бесполезными.

Расположение меток секретности условно выглядит так:

_____$c(O_2)$

_____$c(O_1)$

_____$c(U_1)$

_____$c(U_2)$

Занятие 5

Алгоритмы электронной цифровой подписи

5.1 Алгоритм цифровой подписи RSA

Сначала необходимо вычислить пару ключей (секретный ключ и открытый ключ). Для этого отправитель (автор) электронных документов выбирает два больших простых числа P и Q, затем находит их произведение

$$N = P * Q \ , \tag{5.1}$$

и значение функции

$$\varphi (N) = (P - 1)(Q - 1) \ , \tag{5.2}$$

Далее отправитель вычисляет число E из условий:

$$E \le \varphi (N), \quad НОД (E, \varphi (N)) = 1 \ , \tag{5.3}$$

и число D из условий:

$$D < N, \quad E * D \equiv 1 \ (\mathrm{mod}\ \varphi (N)) \ , \tag{5.4}$$

Пара чисел (E,N) является открытым ключом. Эту пару чисел автор передает партнерам по переписке для проверки его цифровых подписей. Число D сохраняется автором как секретный ключ для подписывания.

Обобщенная схема формирования и проверки цифровой подписи RSA показана на рис.5.1.

Допустим, что отправитель хочет подписать сообщение M перед его отправкой. Сначала он сжимает сообщение M (блок информации, файл, таблица) с помощью хэш-функции h(·) и получает число m:

$$m = h(M) \ , \tag{5.5}$$

Затем вычисляет цифровую подпись S под электронным документом M, используя хэш-значение m и секретный ключ D:

$$S = m^D \pmod N ,$$

(5.6)

Пара (M,S) передается партнеру-получателю как электрон-ный документ M, подписанный цифровой подписью S, причем подпись S сформирована обладателем секретного ключа D.

После приема пары (M,S) получатель вычисляет хэш-значение сообщения M двумя разными способами. Прежде всего он восстанавливает хэш-значение m′, применяя криптографическое преобразование подписи S с использованием открытого ключа E:

$$m' = S^E \pmod N ,$$

(5.7)

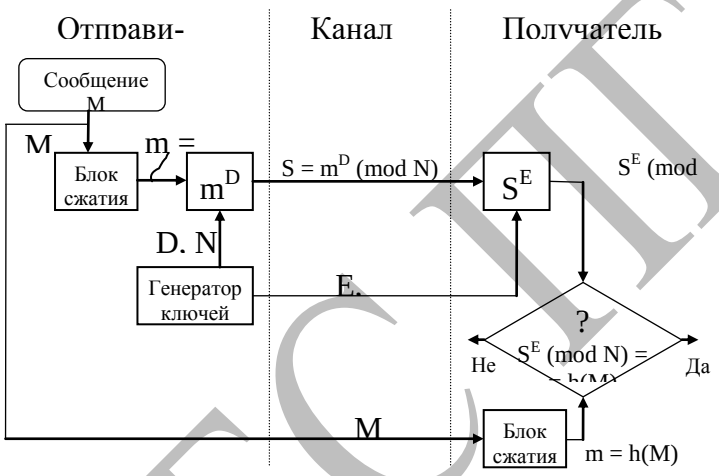


Рисунок 5.1- Обобщенная схема цифровой подписи RSA

Кроме того, он находит результат хэширования принятого сообщения M с помощью такой же хэш-функции h(·):

$$m = h(M),$$

(5.8)

Если соблюдается равенство вычисленных значений, т.е.

$$S^E \pmod N = h(M),$$

(5.9)

то получатель признает пару (M,S) подлинной. Доказано, что только обладатель секретного ключа D может сформировать цифровую подпись S по документу M, а определить секретное число D по открытому числу E не легче, чем разложить модуль N на множители.

Кроме того, можно строго математически доказать, что результат проверки цифровой подписи S будет положительным только в том случае, если при вычислении S был использован секретный ключ D, соответствующий открытому ключу E. Поэтому открытый ключ E иногда называют "идентификатором" подписавшего.

Недостатки алгоритма цифровой подписи RSA.

1. При вычислении модуля N, ключей E и D необходимо проверять большое количество дополнительных условий, что сделать практически трудно. Невыполнение любого из этих условий делает возможным фальсификацию цифровой подписи со стороны того, кто обнаружит такое невыполнение. При подписании важных документов нельзя допускать такую возможность даже теоретически.

2. Для обеспечения криптостойкости цифровой подписи RSA по отношению к попыткам фальсификации на уровне 10^{18} (стандарт США) необходимо использовать при вычислениях N, D и E целые числа не менее 2^{512} (или около 10^{154}) , что требует больших вычислительных затрат, превышающих на 20…30% вычислительные затраты других алгоритмов цифровой подписи при сохранении того же уровня криптостойкости.

3. Цифровая подпись RSA уязвима к так называемой мультипликативной атаке. Иначе говоря, алгоритм цифровой подписи RSA позволяет злоумышленнику без знания секретного ключа D сформировать подписи под теми документами, у которых результат хэширования можно вычислить как произведение результатов хэширования уже подписанных документов.

5.1 Показать, что цифровая подпись RSA уязвима к мультипликативной атаке.

Решение

Допустим, что злоумышленник может сконструировать три сообщения M_1, M_2 и M_3, у которых хэш-значения

$$m_1 = h(M_1), \quad m_2 = h(M_2), \quad m_3 = h(M_3),$$

причем $m_3 = m_1 * m_2 \pmod N$.

Допустим также, что для двух сообщений M_1 и M_2 получены законные подписи

$$S_1 = m_1^D \pmod N \quad \text{и} \quad S_2 = m_2^D \pmod N.$$

Тогда злоумышленник может легко вычислить подпись S_3 для документа M_3, даже не зная секретного ключа D:

$$S_3 = S_1 * S_2 \pmod N.$$

Действительно,

$S_1 * S_2 \pmod N = m_1^D * m_2^D \pmod N = (m_1 m_2)^D \pmod N =$

$= m_3^D \pmod N = S_3$.

Более надежный и удобный для реализации на персональных компьютерах алгоритм цифровой подписи был разработан в 1984 г. американцем арабского происхождения Тахером Эль Гамалем. В 1991 г. НИСТ США обосновал перед комиссией Конгресса США выбор алгоритма цифровой подписи Эль Гамаля в качестве основы для национального стандарта.

5.2 Алгоритм цифровой подписи Эль Гамаля (EGSA)

Название EGSA происходит от слов El Gamal Signature Algorithm (алгоритм цифровой подписи Эль Гамаля). Идея EGSA основана на том, что для обоснования практической невозможности фальсификации цифровой подписи может быть использована более сложная вычислительная задача, чем разложение на множители большого целого числа,– задача дискретного логарифмирования. Кроме того, Эль Гамалю удалось избежать явной слабости алгоритма

цифровой подписи RSA, связанной с возможностью подделки цифровой подписи под некоторыми сообщениями без определения секретного ключа.

Для того чтобы сгенерировать пару ключей (открытый ключ – секретный ключ), сначала выбирают некоторое большое простое целое число P и большое целое число G, причем $G < P$. Отправитель и получатель подписанного документа используют при вычислениях одинаковые большие целые числа P ($\sim 10^{308}$ или $\sim 2^{1024}$) и G ($\sim 10^{154}$ или $\sim 2^{512}$), которые не являются секретными.

Отправитель выбирает случайное целое число X, $1 < X \le (P-1)$, и вычисляет

$$Y = G^X \bmod P \ ,$$

(5.10)

Число Y является открытым ключом, используемым для проверки подписи отправителя. Число Y открыто передается всем потенциальным получателям документов.

Число X является секретным ключом отправителя для подписывания документов и должно храниться в секрете.

Для того чтобы подписать сообщение M, сначала отправитель хэширует его с помощью хэш-функции $h(\cdot)$ в целое число m:

$$m = h(M), \ 1 < m < (P-1),$$

(5.11)

и генерирует случайное целое число K, $1 < K < (P-1)$, такое, что K и $(P-1)$ являются взаимно простыми. Затем отправитель вычисляет целое число a:

$$a = G^K \bmod P \ ,$$

(5.12)

и, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа X целое число b из уравнения

$$m = (X * a + K * b) \ (\mathrm{mod}\ (P-1)),$$

(5.13)

Пара чисел (a,b) образует цифровую подпись S:

$$S = (a,b),$$

(5.14)

проставляемую под документом M.

Тройка чисел (M,a,b) передается получателю, в то время как пара чисел (X,K) держится в секрете.

После приема подписанного сообщения (M,a,b) получатель должен проверить, соответствует ли подпись S = (a,b) сообщению M. Для этого получатель сначала вычисляет по принятому сообщению M число

$$m = h(M),$$

(5.15)

т.е. хэширует принятое сообщение M.

Затем получатель вычисляет значение

$$A = Y^a a^b \pmod P,$$

(5.16)

и признает сообщение M подлинным, если, и только если

$$A = G^m \pmod P,$$

(5.17)

Иначе говоря, получатель проверяет справедливость соотношения

$$Y^a a^b \pmod P = G^m \pmod P ,$$

(5.18)

Можно строго математически доказать, что последнее равенство будет выполняться тогда, и только тогда, когда подпись S=(a,b) под документом M получена с помощью именно того секретного ключа X, из которого был получен открытый ключ Y. Таким образом, можно надежно удостовериться, что отправителем сообщения M был обладатель именно данного секретного ключа X, не раскрывая при этом сам ключ, и что отправитель подписал именно этот конкретный документ M.

Выполнение каждой подписи по методу Эль Гамаля требует нового значения K, причем это значение должно выбираться случайным образом. Если нарушитель раскроет значение K, повторно используемое отправителем, то он сможет раскрыть секретный ключ X отправителя.

Задача 5.2 Сформировать и проверить ЭЦП Эль Гамаля при следующих начальных условиях: P=11, G=2, секретный ключ X=8.

Решение

Вычисляем значение открытого ключа:

$$Y = G^X \bmod P = Y = 2^8 \bmod 11 = 3.$$

Предположим, что исходному сообщению M соответствует хэш-значение m = 5.

Для того,чтобы вычислить цифровую подпись под сообщением M, имеющем хэш-значение m = 5, сначала выберем случайное целое число K = 9. Убедимся, что числа K и (P – 1) являются взаимно простыми. Действительно,

$$\text{НОД}\ (9, 10) = 1.$$

Далее вычисляем элементы a и b подписи:

$$a = G^K \bmod P = 2^9 \bmod 11 = 6,$$

элемент b определяем, используя расширенный алгоритм Евклида:

$$m = (X * a + K * b)\ (\text{mod}\ (P – 1)).$$

При m = 5, a = 6, X = 8, K = 9, P = 11 получаем

$$5 = (6 * 8 + 9 * b)(\text{mod}\ 10)$$

или

$$9 * b \equiv – 43\ (\text{mod}\ 10).$$

Решая сравнение, получаем b = 3. Цифровая подпись представляет собой пару: a = 6, b = 3.

Далее отправитель передает подписанное сообщение. Приняв подписанное сообщение и открытый ключ Y = 3, получатель вычисляет хэш-значение для сообщения M : m = 5, а затем вычисляет два числа:

1) $Y^a a^b\ (\text{mod}\ P) = 3^6 * 6^3\ (\text{mod}\ 11) = 10$;

2) $G^m\ (\text{mod}\ P) = 2^5\ (\text{mod}\ 11) = 10$.

Так как эти два целых числа равны, принятое получателем сообщение признается подлинным.

Следует отметить, что схема Эль Гамаля является характерным примером подхода, который допускает пересылку сообщения M в открытой форме вместе с присоединенным аутентификатором (a,b). В таких случаях процедура установления подлинности принятого сообщения состоит в проверке соответствия аутентификатора сообщению.

Схема цифровой подписи Эль Гамаля имеет ряд преимуществ по сравнению со схемой цифровой подписи RSA.

1. При заданном уровне стойкости алгоритма цифровой подписи целые числа, участвующие в вычислениях, имеют запись на 25% короче, что уменьшает сложность вычислений почти в два раза и позволяет заметно сократить объем используемой памяти.
2. При выборе модуля P достаточно проверить, что это число является простым и что у числа (P –1) имеется большой простой множитель (т.е. всего два достаточно просто проверяемых условия).
3. Процедура формирования подписи по схеме Эль Гамаля не позволяет вычислять цифровые подписи под новыми сообщениями без знания секретного ключа (как в RSA).

Однако алгоритм цифровой подписи Эль Гамаля имеет и некоторые недостатки по сравнению со схемой подписи RSA. В частности, длина цифровой подписи получается в 1,5 раза больше, что, в свою очередь, увеличивает время ее вычисления.

Занятие 6

Распределение ключей в компьютерной сети

При использовании для информационного обмена криптосистемы с симметричным секретным ключом два пользователя, желающие обменяться криптографически защищенной информацией, должны обладать общим секретным ключом. Пользователи должны обменяться общим ключом по каналу связи безопасным образом. Если пользователи меняют ключ достаточно часто, то доставка ключа превращается в серьезную проблему.

Для решения этой проблемы можно применить два способа:

1) использование криптосистемы с открытым ключом для шифрования и передачи секретного ключа симметричной криптосистемы;

2) использование системы открытого распределения ключей Диффи–Хеллмана.

6.1 Алгоритм открытого распределения ключей Диффи – Хеллмана

Алгоритм Диффи–Хеллмана был первым алгоритмом с открытыми ключами (предложен в 1976 г.). Его безопасность обусловлена трудностью вычисления дискретных логарифмов в конечном поле, в отличие от легкости дискретного возведения в степень в том же конечном поле.

Предположим, что два пользователя A и B хотят организовать защищенный коммуникационный канал.

1. Обе стороны заранее уславливаются о модуле N (N должен быть простым числом) и примитивном элементе g, ($1 \le g \le N-1$).

Эти два целых числа N и g могут не храниться в секрете. Как правило, эти значения являются общими для всех пользователей системы.

2. Затем пользователи A и B независимо друг от друга выбирают собственные секретные ключи k_A и k_B (k_A и k_B – случайные большие целые числа, которые хранятся пользователями A и B в секрете).

3. Далее пользователь A вычисляет открытый ключ

$$y_A = g^{k_A} (\mathrm{mod}\ N),$$

а пользователь B – открытый ключ

$$y_B = g^{k_B} (\text{mod } N).$$

4. Затем стороны A и B обмениваются вычисленными значениями открытых ключей y_A и y_B по незащищенному каналу.

5. Далее пользователи A и B вычисляют общий секретный ключ, используя следующие выражения:

пользователь A: $K = (y_B)^{k_A} = (g^{k_B})^{k_A} \ (\text{mod } N)$;

пользователь B: $K' = (y_A)^{k_B} = (g^{k_A})^{k_B} \ (\text{mod } N)$.

При этом $K = K'$, так как $(g^{k_B})^{k_A} = (g^{k_A})^{k_B} \ (\text{mod } N)$.

Схема реализации алгоритма Диффи–Хеллмана показана на рис. 6.1.

Ключ K может использоваться в качестве общего секретного ключа (ключа шифрования ключей) в симметричной криптосистеме. Кроме того, обе стороны A и B могут шифровать сообщения, используя следующее преобразование шифрования (типа RSA): $C = E_K (M) = M^K \ (\text{mod } N)$.

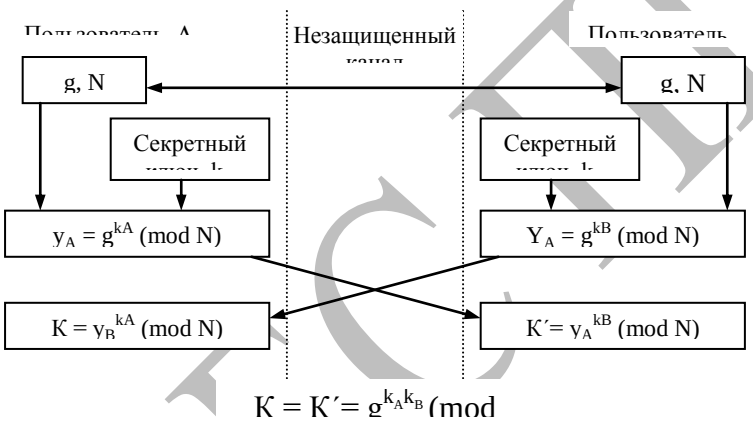


Рисунок 6.1- Схема реализации алгоритма Диффи–Хеллмана

Для выполнения расшифрования получатель сначала находит ключ расшифрования K* с помощью сравнения

$$K * K^* \equiv 1 \ (\text{mod } N - 1), \qquad (6.1)$$

а затем восстанавливает сообщение

$$M = D_K (C) = C^{K^*} (\text{mod } N). \qquad (6.2)$$

Задача 6.1 Реализовать алгоритм открытого распределения ключей Диффи-Хеллмана при следующих начальных условиях: модуль N=47, примитивный

элемент g=23, секретные ключи пользователей А и В: K_A=12, K_B=33 соответственно.

Решение

Для того, чтобы иметь общий секретный ключ К, пользователи А и В сначала вычисляют значения частных открытых ключей:

$$y_A = g^{k_A} \pmod N = 23^{12} \pmod{47} = 27 ,$$

$$y_B = g^{k_B} \pmod N = 23^{33} \pmod{47} = 33$$

После того, как пользователи А и В обменяются своими значениями y_A и y_B, они вычисляют общий секретный ключ

$K = (y_B)^{k_A} \pmod N = (y_A)^{k_B} \pmod N = 33^{12} \pmod{47} = 27^{33} \pmod{47} = = 23^{12*33} \pmod{47} = 25$.

Кроме того, они находят секретный ключ расшифрования, решая следующее сравнение:

$$K * K^* \equiv 1 \pmod{N-1},$$

откуда $K^* = 35$.

Если сообщение М =16, то криптограмма:

$$C = M^K = 16^{25} \pmod{47} = 21.$$

Получатель восстанавливает сообщение :

$$M = C^{K^*} = 21^{35} \pmod{47} = 16.$$

Злоумышленник, перехватив значения N, g, y_A и y_B, тоже хотел бы определить значение ключа К. Очевидный путь для решения этой задачи состоит в вычислении такого значения k_A по N, g, y_A, что $g^{k_A} \bmod N = y_A$ (поскольку в

этом случае, вычислив k_A, можно найти $K=(y_B)^{k_A} \bmod N$). Однако нахождение k_A по N, g и y_A – задача нахождения дискретного логарифма в конечном поле, которая считается неразрешимой.

Выбор значений N и g может иметь существенное влияние на безопасность этой системы. Модуль N должен быть большим и простым числом. Чис-

ло $(N-1)/2$ также должно быть простым числом. Число g желательно выбирать таким, чтобы оно было примитивным элементом множества Z_N.

Алгоритм открытого распределения ключей ДиффиХеллмана позволяет обойтись без защищенного канала для передачи ключей. Однако, работая с этим алгоритмом, необходимо иметь гарантию того, что пользователь A получил открытый ключ именно от пользователя B, и наоборот. Эта проблема решается с помощью электронной подписи, которой подписываются сообщения об открытом ключе.

Метод Диффи–Хеллмана дает возможность шифровать данные при каждом сеансе связи на новых ключах. Это позволяет не хранить секреты на дискетах или других носителях. Не следует забывать, что любое хранение секретов повышает вероятность попадания их в руки конкурентов или противника.

Преимущество метода Диффи–Хеллмана по сравнению с методом RSA заключается в том, что формирование общего секретного ключа происходит в сотни раз быстрее. В системе RSA генерация новых секретных и открытых ключей основана на генерации новых простых чисел, что занимает много времени.

Занятие 7

Протоколы идентификации с нулевой передачей знаний

Широкое распространение интеллектуальных карт (смарт-карт) для разнообразных коммерческих, гражданских и военных применений (кредитные карты, карты социального страхования, карты доступа в охраняемое помещение, компьютерные пароли и ключи, и т.п.) потребовало обеспечения безопасной идентификации таких карт и их владельцев. Во многих приложениях главная проблема заключается в том, чтобы при предъявлении интеллектуальной карты оперативно обнаружить обман и отказать обманщику в допуске, ответе или обслуживании.

Для безопасного использования интеллектуальных карт разработаны протоколы идентификации с нулевой передачей знаний. Секретный ключ владельца карты становится неотъемлемым признаком его личности. Доказательство знания этого секретного ключа с нулевой передачей этого знания служит доказательством подлинности личности владельца карты.

7.1 Упрощенная схема идентификации с нулевой передачей знаний

Схему идентификации с нулевой передачей знаний предложили в 1986 г. У.Фейге, А.Фиат и А.Шамир. Она является наиболее известным доказательством идентичности с нулевой передачей конфиденциальной информации.

Рассмотрим сначала упрощенный вариант схемы идентификации с нулевой передачей знаний для более четкого выявления ее основной концепции. Прежде всего, выбирают случайное значение модуля n, который является произведением двух больших простых чисел. Модуль n должен иметь длину 512…1024 бит. Это значение n может быть представлено группе пользователей, которым придется доказывать свою подлинность. В процессе идентификации участвуют две стороны:

- сторона А, доказывающая свою подлинность,
- сторона В, проверяющая представляемое стороной А доказательство.

Для того чтобы сгенерировать открытый и секретный ключи для стороны A, доверенный арбитр (Центр) выбирает некоторое число V, которое является квадратичным вычетом по модулю n. Иначе говоря, выбирается такое число V, что сравнение

$$x^2 \equiv V \pmod{n}$$

имеет решение и существует целое число

$$V^{-1} \bmod n.$$

Выбранное значение V является *открытым ключом* для A. Затем вычисляют наименьшее значение S, для которого

$$S \equiv \text{sqrt}\,(V^{-1}) \pmod{n}.$$

Это значение S является *секретным ключом* для A.

Теперь можно приступить к выполнению протокола идентификации.

1. Сторона A выбирает некоторое случайное число r, $r < n$. Затем она вычисляет

$$x = r^2 \bmod n$$

и отправляет x стороне B.

2. Сторона B посылает A случайный бит b.

3. Если b=0, тогда A отправляет r стороне B. Если b=1, то A отправляет стороне B

$$y = r * S \bmod n.$$

4. Если 1) b = 0, сторона B проверяет, что

$$1)\ x = r^2 \bmod n,$$

чтобы убедиться, что A знает sqrt (x). Если 2) b=1, сторона B проверяет, что

$$2)\ x = y^2 * V \bmod n,$$

чтобы быть уверенной, что A знает sqrt (V^{-1}).

Эти шаги образуют один цикл протокола, называемый *аккредитацией*. Стороны A и B повторяют этот цикл t раз при разных случайных значениях r и b до тех пор, пока B не убедится, что A знает значение S.

Если сторона A не знает значения S, она может выбрать такое значение r, которое позволит ей обмануть сторону B, если B отправит ей b=0, либо A

может выбрать такое r, которое позволит обмануть B, если B отправит ей b=1. Но этого невозможно сделать в обоих случаях. Вероятность того, что A обманет B в одном цикле, составляет 1/2. Вероятность обмануть B в t циклах равна $(1/2)^t$.

Для того чтобы этот протокол работал, сторона A никогда не должна повторно использовать значение r. Если A поступила бы таким образом, а сторона B отправила бы стороне A на шаге 2 другой случайный бит b, то B имела бы оба ответа A. После этого B может вычислить значение S, и для A все закончено.

7.2.Параллельная схема идентификации с нулевой передачей знаний

Параллельная схема идентификации позволяет увеличить число аккредитаций, выполняемых за один цикл, и тем самым уменьшить длительность процесса идентификации.

Как и в предыдущем случае, сначала генерируется число n как произведение двух больших чисел. Для того, чтобы сгенерировать открытый и секретный ключи для стороны A, сначала выбирают K различных чисел $V_1, V_2, ..., V_K$, где каждое V_i является квадратичным вычетом по модулю n. Иначе говоря, выбирают значение V_i таким, что сравнение

$$x^2 \equiv V_i \bmod n$$

имеет решение и существует $V_i^{-1} \bmod n$. Полученная строка $V_1, V_2, ..., V_K$ является *открытым ключом*.

Затем вычисляют такие наименьшие значения S_i, что

$$S_i = \text{sqrt}\,(V_i^{-1}) \bmod n.$$

Эта строка $S_1, S_2, ..., S_K$ является *секретным ключом* стороны A.

Протокол процесса идентификации имеет следующий вид:

1. Сторона A выбирает некоторое случайное число r, $r<n$. Затем она вычисляет $x=r^2 \bmod n$ и посылает x стороне B.

2. Сторона B отправляет стороне A некоторую случайную двоичную строку из K бит: b1, b2, ..., bK.

3. Сторона A вычисляет

$$y = r * (S_1^{b1} * S_2^{b2} * ... * S_K^{bK}) \bmod n.$$

Перемножаются только те значения S_i, для которых b_i=1. Например, если b_1=1, то сомножитель S_1 входит в произведение, если же b_1=0, то S_1 не входит в произведение, и т.д. Вычисленное значение y отправляется стороне B.

4. Сторона B проверяет, что

$$x = y^2 * (V_1^{b1} * V_2^{b2} * ... * V_K^{bK}) \bmod n.$$

Фактически сторона B перемножает только те значения V_i, для которых b_i=1. Стороны A и B повторяют этот протокол t раз, пока B не убедится, что A знает $S_1, S_2, ..., S_K$.

Вероятность того, что A может обмануть B, равна $(1/2)^{Kt}$. Авторы рекомендуют в качестве контрольного значения брать вероятность обмана B равной $(1/2)^{20}$ при K=5 и t=4.

Задача 7.1 Требуется рассмотреть работу параллельной схемы идентификации с нулевой передачей знаний, если модуль $n=35$ (произведение двух простых чисел 5 и 7).

Решение

Возможные квадратичные вычеты будут следующими:

1: $x^2 \equiv 1 \pmod{35}$ имеет решения: x =1, 6, 29, 34;

4: $x^2 \equiv 4 \pmod{35}$ имеет решения: x = 2, 12, 23, 33;

9: $x^2 \equiv 9 \pmod{35}$ имеет решения: x = 3, 17, 18, 32;

11: $x^2 \equiv 11 \pmod{35}$ имеет решения: x = 9, 16, 19, 26;

14: $x^2 \equiv 14 \pmod{35}$ имеет решения: x = 7, 28;

15: $x^2 \equiv 15 \pmod{35}$ имеет решения: x = 15, 20;

16: $x^2 \equiv 16 \pmod{35}$ имеет решения: x = 4, 11, 24, 31;

21: $x^2 \equiv 21 \pmod{35}$ имеет решения: x =14, 21;

25: $x^2 \equiv 25 \pmod{35}$ имеет решения: x = 5, 30;

29: $x^2 \equiv 29 \pmod{35}$ имеет решения: x = 8, 13, 22, 27;

30: $x^2 \equiv 30 \pmod{35}$ имеет решения: x =10, 25.

Заметим, что 14, 15, 21, 25 и 30 не имеют обратных значений по модулю 35, потому что они не являются взаимно простыми с 35. Следует также отметить, что число квадратичных вычетов по модулю 35, взаимно простых с $n = p * q = 5 * 7 = 35$ (для которых НОД (x, 35) =1), равно

$$(p-1)(q-1)/4 = (5-1)(7-1)/4 = 6.$$

Составим таблицу квадратичных вычетов по модулю 35, обратных к ним значений по модулю 35 и их квадратных корней.

Таблица 7.1

V	V^{-1}	S = sqrt (V^{-1})
1	1	1
4	9	3
9	4	2
11	16	4
16	11	9
29	29	8

Итак, сторона A получает открытый ключ, состоящий из K=4 значений V:

[4, 11, 16, 29].

Соответствующий секретный ключ, состоящий из K=4 значений S:

[3 4 9 8].

Рассмотрим один цикл протокола.

1. Сторона A выбирает некоторое случайное число r=16, вычисляет

$$x = 16^2 \bmod 35 = 11$$

и посылает это значение x стороне B.

2. Сторона B отправляет стороне A некоторую случайную двоичную строку

[1, 1, 0, 1].

3. Сторона A вычисляет значение

$$y = r \cdot (S_1^{b_1} \cdot S_2^{b_2} \cdot \ldots \cdot S_K^{b_K}) \bmod n = 16 \cdot (3^1 \cdot 4^1 \cdot 9^0 \cdot 8^1) \bmod 35 = 31$$

и отправляет это значение *y* стороне B.

4. Сторона B проверяет, что

$$x=y^2\cdot(V_1^{b_1}\cdot V_2^{b_2}\cdot\ldots\cdot V_K^{b_K})\bmod n=31^2\cdot(4^1\cdot11^1\cdot16^0\cdot29^1)\bmod35=11.$$

Стороны A и B повторяют этот протокол *t* раз, каждый раз с разным случайным числом *r*, пока сторона B не будет удовлетворена.

При малых значениях величин, как в данном примере, не достигается настоящей безопасности. Но если n представляет собой число длиной 512 бит и более, сторона B не сможет узнать ничего о секретном ключе стороны A, кроме того факта, что сторона A знает этот ключ.

Приложение

Элементы теории чисел

Модулярная арифметика

Модулярная арифметика часто изучается в школе как "арифметика часов". Если отсчитать 14 часов от 3 часов после полудня, то получится 5 часов утра следующего дня:

$$3 + 14 \equiv 5 \pmod{12}$$

или

$$(3 + 14) \bmod 12 = 5.$$

Это арифметика по модулю 12.

Обычная запись в модулярной арифметике

$$a \equiv b \pmod{n},$$

читается так: "a сравнимо с b по модулю n". Это соотношение справедливо для целых значений a,b и $n \neq 0$, если, и только если

$$a = b + k * n,$$

для некоторого целого k.

Отсюда, в частности, следует

$$n \mid (a - b).$$

Это читается как "n делит (a – b)".

Если $a \equiv b \pmod{n}$,

то b называют *вычетом* числа a по модулю n.

Операцию нахождения вычета числа a по модулю n

$$a \pmod{n},$$

называют приведением числа a по модулю n или *приведением по модулю.*

В нашем примере

$$(3 + 14) \bmod 12 = 17 \bmod 12 = 5,$$

или

$$17 \equiv 5 \pmod{12},$$

число 5 является вычетом числа 17 по модулю 12.

Набор целых чисел от 0 до (n –1) называют *полным набором вычетов по модулю* n. Это означает, что для любого целого a (a > 0) его вычет r по модулю n есть некоторое целое число в интервале от 0 до (n –1), определяемое из соотношения

$$r = a - k * n,$$

где k – целое число.

Например, для n =12 полный набор вычетов:

$$\{0, 1, 2, \ldots, 11\}.$$

Обычно предпочитают использовать вычеты

$$r \in \{0, 1, 2, \ldots, n-1\},$$

но иногда полезны вычеты в диапазоне целых:

$$r \in \left\{-\frac{1}{2}(n-1), \ldots, \frac{1}{2}(n-1)\right\}.$$

Заметим, что

$$-12 \pmod 7 \equiv -5 \pmod 7 \equiv 2 \pmod 7 \equiv 9 \pmod 7 \text{ и т.д.}$$

Модулярная арифметика аналогична во многом обычной арифметике: она коммутативна, ассоциативна и дистрибутивна. Точнее говоря, целые числа по модулю n с использованием операций сложения и умножения образуют коммутативное кольцо при соблюдении законов ассоциативности, коммутативности и дистрибутивности.

Фактически мы можем либо сначала приводить по модулю n, а затем выполнять операции, либо сначала выполнять операции, а затем приводить по модулю n, поскольку приведение по модулю n является *гомоморфным отображением* из кольца целых в кольцо целых по модулю n:

$$(a + b) \bmod n = [a \pmod n + b \pmod n] \bmod n,$$

$$(a - b) \bmod n = [a \pmod n - b \pmod n] \bmod n,$$

$$(a * b) \bmod n = [a \pmod n * b \pmod n] \bmod n,$$

$$[a * (b + c)] \bmod n = \{[a * b \pmod n] + [a * c \pmod n]\} \bmod n.$$

Криптография использует множество вычислений по модулю n, потому что задачи типа вычисления дискретных логарифмов и квадратных корней очень трудны. Кроме того, с вычислениями по модулю удобнее работать, потому что они ограничивают диапазон всех промежуточных величин и результата.

Для модуля n длиной k бит промежуточные результаты любого сложения, вычитания или умножения будут не длиннее
2k бит. Поэтому возведение в степень в модулярной арифметике можно выполнить без генерации очень больших промежуточных результатов.

Вычисление степени числа a по модулю n

$$a^x \bmod n,$$

можно выполнить как ряд умножений и делений. Существуют способы сделать это быстрее. Поскольку эти операции дистрибутивны, быстрее произвести возведение в степень как ряд последовательных умножений, выполняя каждый раз приведение по модулю. Это особенно заметно, если работать с длинными числами (200 бит и более).

Например, если нужно вычислить

$$a^8 \bmod n,$$

не следует применять примитивный подход с выполнением семи перемножений и одного приведения по модулю громадного числа:

$$(a * a * a * a * a * a * a * a) \bmod n.$$

Вместо этого выполняют три малых умножения и три малых приведения по модулю:

$$((a^2 \bmod n)^2 \bmod n)^2 \bmod n.$$

Тем же способом вычисляют

$$a^{16} \bmod n = (((a^2 \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n.$$

Вычисление

$$a^x \bmod n,$$

где x не является степенью 2, лишь немного сложнее. Двоичная запись числа x позволяет представить число x как сумму степеней 2:

$$x = 25_{(10)} \to 1\ \ 1\ \ 0\ \ 0\ \ 1_{(2)}, \quad \text{поэтому} \quad 25 = 2^4 + 2^3 + 2^0.$$

Тогда

$$a^{25} \bmod n = (a * a^{24}) \bmod n = (a * a^8 * a^{16}) \bmod n =$$

$$= a * ((a^2)^2)^2 * (((a^2)^2)^2)^2 \bmod n = ((((a^2 * a)^2)^2)^2 * a) \bmod n.$$

При разумном накоплении промежуточных результатов потребуется только шесть умножений:

$$(((((((a^2 \bmod n) * a) \bmod n)^2 \bmod n)^2 \bmod n)^2 \bmod n) * a) \bmod n.$$

Этот метод уменьшает трудоемкость вычислений до 1,5xk операций в среднем, где k – длина числа в битах.

Поскольку многие алгоритмы шифрования основаны на возведении в степень по модулю n, целесообразно использовать алгоритмы быстрого возведения в степень.

Вычисление обратных величин

В арифметике действительных чисел нетрудно вычислить мультипликативную обратную величину a^{-1} для ненулевого a:

$$a^{-1} = 1/a \quad \text{или} \quad a * a^{-1} = 1.$$

Например, мультипликативная обратная величина от числа 4 равна 1/4, поскольку

$$4 * \frac{1}{4} = 1.$$

В модулярной арифметике вычисление обратной величины является более сложной задачей. Например, решение сравнения

$$4 * x \equiv 1 \pmod 7$$

эквивалентно нахождению таких значений x и k, что

$$4 * x \equiv 7 * k + 1,$$

где x и k – целые числа.

Общая формулировка этой задачи – нахождение такого целого числа x, что

$$a * x \pmod{n} = 1.$$

Можно также записать

$$a^{-1} \equiv x \pmod{n}.$$

Решение этой задачи иногда существует, а иногда его нет. Например, обратная величина для числа 5 по модулю 14 равна 3, поскольку

$$5 * 3 = 15 \equiv 1 \pmod{14}.$$

С другой стороны, число 2 не имеет обратной величины по модулю 14.

Вообще сравнение

$$a^{-1} \equiv x \pmod{n}$$

имеет единственное решение, если a и n – взаимно простые числа.

Если числа a и n не являются взаимно простыми, тогда сравнение

$$a^{-1} \equiv x \pmod{n}$$

не имеет решения.

Сформулируем основные способы нахождения обратных величин. Пусть целое число $a \in \{0, 1, 2, \ldots, n-1\}$.

Если НОД (a, n) =1, то $a * i \pmod{n}$ при $i = 0, 1, 2, \ldots, n-1$ является перестановкой множества $\{0, 1, 2, \ldots, n-1\}$.

Например, если a = 3 и n = 7 (НОД (3,7) =1), то

$$3 * i \pmod{7} \quad \text{при } i = 0, 1, 2, \ldots, 6$$

является последовательностью 0, 3, 6, 2, 5, 1, 4, т.е. перестановкой множества $\{0, 1, 2, \ldots, 6\}$.

Это становится неверным, когда НОД (a, n) ≠ 1. Например, если a = 2 и n = 6, то $2 * i \pmod{6} \equiv 0, 2, 4, 0, 2, 4$ при $i = 0, 1, 2, \ldots, 5$.

Если НОД (a,n) =1, тогда существует обратное число a^{-1}, $0 < a^{-1} < n$, такое, что

$$a * a^{-1} \equiv 1 \pmod{n}.$$

Действительно, a ∗ i (mod n) является перестановкой 0, 1, ..., n –1, поэтому существует i, такое, что

$$a * i \equiv 1 \pmod{n}.$$

Как уже отмечалось, набор целых чисел от 0 до n –1 называют *полным набором вычетов* по модулю n. Это означает, что для любого целого числа a (a > 0) его вычет r = a (mod n) – это некоторое целое число в интервале от 0 до n –1.

Выделим из полного набора вычетов подмножество вычетов, взаимно простых с n. Такое подмножество *называют приведенным набором вычетов.*

Пример. Пусть модуль n =11 – простое число. Полный набор вычетов по модулю 11

$$\{0, 1, 2, \ldots, 10\}.$$

При формировании приведенного набора вычетов из них удаляется только один элемент – 0. Приведенный набор вычетов по модулю 11 имеет 11 – 1 = 10 элементов.

Вообще приведенный набор вычетов по модулю простого числа n имеет n –1 элементов.

Пример. Пусть модуль n =10. Полный набор вычетов по модулю n =10

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

Из них только 1, 3, 7, 9 не имеют общего сомножителя с числом 10. Поэтому приведенный набор вычетов по модулю 10 равен {1, 3, 7, 9}. При формировании этого приведенного набора были исключены элементы:

0 (1 элемент),

кратные 2 (4 элемента),

кратные 5 (1 элемент),

т.е. всего шесть элементов. Вычитая их из 10, получаем 10 – 1 – 4 – 1 = 4, т.е. четыре элемента в приведенном наборе.

Для произведения простых чисел $p * q = n$ приведенный набор вычетов имеет $(p-1)(q-1)$ элементов. При $n=p * q=2 * 5=10$ число элементов в приведенном наборе

$$(p-1)(q-1) = (2-1)(5-1) = 4.$$

Пример. Приведенный набор вычетов по модулю $27=3^3$ имеет 18 элементов:

$$\{1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26\}.$$

Из полного набора вычетов исключены элементы, кратные 3 (всего девять элементов).

Для модуля в виде простой степени n^r приведенный набор вычетов имеет $n^{r-1}(n-1)$ элементов.

При $n = 3, r = 3$ получаем $3^{3-1}(3-1) = 3^2 * 2 = 18$.

Функция Эйлера $\varphi(n)$ характеризует число элементов в приведенном наборе вычетов (табл. П.1).

Таблица П.1

Модуль n	Функция $\varphi(n)$
n – простое	n –1
n^2	n (n –1)
…	…
n^r	$n^{r-1}(n-1)$

$p * q$ (p, q – простые)	$(p-1)(q-1)$
…	…
$\prod_{i=1}^{t} p_i^{e_i}$ (p_i – простые)	$\prod_{i=1}^{t} p_i^{e_i-1}(p_i-1)$

Иначе говоря, функция $\varphi(n)$ – это количество положительных целых, меньших n, которые взаимно просты с n.

Малая теорема Ферма: если n– простое и НОД (a,n)=1, то

$$a^{n-1} \equiv 1 \pmod{n}.$$

Согласно обобщению Эйлером малой теоремы Ферма имеем: если НОД (a,n) =1, то

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$

Если n – простое число, то предыдущий результат, учитывая, что $\varphi(n) = n - 1$, приводится к виду (малой теоремы Ферма)

$$a^{n-1} \equiv 1 \pmod{n}.$$

Основные способы нахождения обратных величин

$$a^{-1} \equiv 1 \pmod{n}.$$

1. Проверить поочередно значения 1, 2, ..., n – 1, пока не будет найден $a^{-1} \equiv 1 \pmod{n}$, такой, что $a * a^{-1} \pmod{n} \equiv 1$.

2. Если известна функция Эйлера $\varphi(n)$, то можно вы-числить

$$a^{-1} \pmod{n} \equiv a^{\varphi(n)-1} \pmod{n},$$

используя алгоритм быстрого возведения в степень.

3. Если функция Эйлера $\varphi(n)$ не известна, можно использовать расширенный алгоритм Евклида.

Проиллюстрируем эти способы на числовых примерах.

1. Поочередная проверка значений 1, 2, ..., n – 1, пока не будет найден $x = a^{-1} \pmod{n}$, такой что $a * x \equiv 1 \pmod{n}$.

Пусть $n = 7$, $a = 5$. Требуется найти $x = a^{-1} \pmod{n}$.

$$a * x \equiv 1 \pmod{n} \quad \text{или} \quad 5 * x \equiv 1 \pmod{7}.$$

$$n - 1 = 7 - 1 = 6.$$

Получаем $x = 5^{-1} \pmod 7 = 3$.

Результаты проверки сведены в табл. П.2.

Таблица П.2

x	5 ∗ x	5 ∗ x (mod 7)
1	5	5
2	10	3
3	15	1
4	20	6
5	25	4
6	30	2

2. Нахождение $a^{-1} \pmod n$, если известна функция Эйлера $\varphi(n)$.

Пусть $n = 7$, $a = 5$. Найти $x = a^{-1} \pmod n = 5^{-1} \pmod 7$. Модуль $n = 7$ – простое число. Поэтому функция Эйлера $\varphi(n) = \varphi(7) = = n - 1 = 6$. Обратная величина от 5 по mod 7

$$a^{-1} \pmod n = a^{\varphi(n)-1} \pmod n =$$

$$= 5^{6-1} \bmod 7 = 5^5 \bmod 7 = (5^2 \bmod 7)(5^3 \bmod 7) \bmod 7 =$$

$$= (25 \bmod 7)(125 \bmod 7) \bmod 7 = (4 * 6) \bmod 7 = 24 \bmod 7 = 3.$$

Итак, $x = 5^{-1} \pmod 7 = 3$.

3. Нахождение обратной величины $a^{-1} \pmod n$ с помощью расширенного алгоритма Евклида.

Алгоритм Евклида можно обобщить способом, который имеет большое практическое значение. При этом способе во время вычисления НОД (a,b) можно попутно вычислить такие целые числа u_1 и u_2, что

$$a * u_1 + b * u_2 = \text{НОД}\ (a,b).$$

Это обобщение (расширение) алгоритма Евклида удобно описать, используя векторные обозначения.

Квадратичные вычеты

Рассмотрим некоторое простое $p > 2$ и число $a < p$. Если число a сравнимо с квадратом некоторого числа x по модулю p, т.е. выполняется сравнение $x^2 \equiv a \pmod p$, тогда a называют *квадратичным вычетом* по модулю p. В противном случае a называют *квадратичным невычетом* по модулю p.

Если a – квадратичный вычет, сравнение $x^2 \equiv a \pmod p$ имеет два решения: $+x$ и $-x$, т.е. a имеет два квадратных корня по модулю p.

Все квадратичные вычеты находят возведением в квадрат элементов $1, 2, 3, ..., (p-1)/2$.

Не все значения $a < p$ являются квадратичными вычетами. Например, при $p = 7$ квадратичные вычеты это 1, 2, 4:

$$1^2 = 1 \equiv \pmod 7,$$

$$2^2 = 4 \equiv 4 \pmod 7,$$

$$3^2 = 9 \equiv 2 \pmod 7,$$

$$4^2 = 16 \equiv 2 \pmod 7,$$

$$5^2 = 25 \equiv 4 \pmod 7,$$

$$6^2 = 36 \equiv 1 \pmod 7.$$

Заметим, что каждый квадратичный вычет появляется в этом списке дважды. Не существует никаких значений x, которые удовлетворяли бы любому из следующих уравнений:

$$x^2 \equiv 3 \pmod 7,$$

$$x^2 \equiv 5 \pmod 7,$$

$$x^2 \equiv 6 \pmod 7.$$

Числа 3, 5 и 6 – квадратичные невычеты по модулю 7. Можно доказать, что существует точно $(p-1)/2$ квадратичных вычетов по модулю p и $(p-1)/2$ квадратичных невычетов по модулю p.

Если a – квадратичный вычет по модулю p, то a имеет точно два квадратных корня: один корень между 0 и $(p-1)/2$, другой корень между $(p-1)/2$ и $(p-1)$.

Один из этих квадратных корней также является квадратичным вычетом по модулю p; он называется *главным квадратным корнем*.

Вычисление квадратных корней при p=7 представлено в табл. П.4.

Таблица П.4

$x^2 \equiv a \pmod 7$	Корни	
	x_1	x_2
$1^2 \equiv 1 \pmod 7$	+1	$-1 = -1 + 7 = 6$
$2^2 \equiv 4 \pmod 7$	+2	$-2 = -2 + 7 = 5$
$3^2 \equiv 2 \pmod 7$	+3	$-3 = -3 + 7 = 4$

Если n – произведение двух простых p и q, т.е. $n = p * q$, то существуют точно

$$(p-1)(q-1)/4$$

квадратичных вычетов по модулю n, взаимно простых с n. Например, по модулю 35 ($p = 5$, $q = 7$, $n = 5 * 7 = 35$) существуют

$$\frac{(5-7)(7-1)}{4} = \frac{4*6}{4} = 6$$

квадратичных вычетов: 1, 4, 9, 11, 16, 29, взаимно простых с 35.