ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Кафедра мультисервисных сетей и информационной безопасности

В.В. Пугин, Е.Ю. Голубничая, С.А. Лабада

КРИПТОАЛГОРИТМЫ

Методические указания по выполнению лабораторной работы

Самара
2018

УДК 681.322.067

П

Рекомендовано к изданию методическим советом ПГУТИ, протокол №60, от 15.05.2018 г.

Рецензент:

доцент кафедры «Системы связи» ФГБОУ ВО ПГУТИ,

к.т.н., доц. Кустова М.Н.

Пугин, В. В., Голубничая, Е. Ю., Лабада, С. А.

П **Криптоалгоритмы**: методические указания по выполнению лабораторной работы / В. В. Пугин, Е. Ю. Голубничая, С. А. Лабада. – Самара: ПГУТИ, 2018. – 29 с.

Методические указания по выполнению лабораторной работы «Криптоалгоритмы» содержат описание основных возможностей и команд программы SPECTralInfo Crypter, которая позволяет имитировать процедуру шифрования и дешифрования различными криптоалгоритмами, а также осуществлять криптоанализ. Методические указания разработаны в соответствии с ФГОС ВО по направлениям подготовки 10.05.02 «Информационная безопасность телекоммуникационных систем», 10.03.01 «Информационная безопасность», и предназначены для проведения лабораторных занятий по дисциплине «Основы информационной безопасности».

1 Криптоалгоритмы

1.1 Основные понятия и классификация криптоалгоритмов

С распространением письменности в человеческом обществе появилась потребность в обмене письмами и сообщениями, что вызвало необходимость сокрытия содержимого письменных сообщений от посторонних. Методы сокрытия содержимого письменных сообщений можно разделить на три группы. К первой группе относятся методы маскировки или стеганографии, которые осуществляют сокрытие самого факта наличия сообщения; вторую группу составляют различные методы тайнописи или криптографии (от греческих слов ktyptos – тайный и grapho – пишу); методы третьей группы ориентированы на создание специальных технических устройств, засекречивания информации.

Первым классом криптосистем, практическое применение которых стало возможно с появлением мощных и компактных вычислительных средств, стали блочные шифры. В 70-е гг. был разработан американский стандарт шифрования DES (принят в 1978 г.). Один из его авторов, Хорст Фейстель (сотрудник IBM), описал модель блочных шифров, на основе которой были построены другие, более стойкие симметричные криптосистемы, в том числе отечественный стандарт шифрования ГОСТ 28147–89. В середине 70-х гг. XX столетия произошел настоящий прорыв в современной криптографии – появление асимметричных криптосистем, которые не требовали передачи секретного ключа между сторонами. Здесь отправной точкой принято считать работу, опубликованную Уитфилдом Диффи и Мартином Хеллманом в 1976 г. под названием «Новые направления в современной криптографии». В ней впервые сформулированы принципы обмена шифрованной информацией без обмена секретным ключом. Независимо к идее асимметричных криптосистем подошел Ральф Меркли. Несколькими годами позже Рон

Ривест, Ади Шамир и Леонард Адлеман открыли систему RSA, первую практическую асимметричную криптосистему, стойкость которой была основана на проблеме факторизации больших простых чисел. Асимметричная криптография открыла сразу несколько новых прикладных направлений, в частности системы электронной цифровой подписи (ЭЦП) и электронных денег. В 1980–90-е гг. появились совершенно новые направления криптографии: вероятностное шифрование, квантовая криптография и другие. Осознание их практической ценности еще впереди. Актуальной остается и задача совершенствования симметричных криптосистем. В этот же период были разработаны нефейстелевские шифры (SAFER, RC6 и др.), а в 2000 г. после открытого международного конкурса был принят новый национальный стандарт шифрования США – AES. Криптография является одним из наиболее мощных средств обеспече- ния конфиденциальности и контроля целостности информации. Во многих отношениях она занимает центральное место среди программно-технических регуляторов безопасности. Например, для портативных компьютеров, физически защитить которые крайне трудно, только криптография позволяет гарантировать конфиденциальность информации даже в случае кражи.

Таким образом, защита данных с помощью шифрования – одно из возможных решений проблемы безопасности. Зашифрованные данные становятся доступными только тем, кто знает, как их расшифровать, и поэтому похищение зашифрованных данных абсолютно бессмысленно для несанкционированных пользователей. Наукой, изучающей математические методы защиты информации путем ее преобразования, является криптология. Криптология разделяется на два направления – криптографию и криптоанализ. Криптография изучает методы преобразования информации, обеспечивающие ее конфиденциальность и аутентичность.

Под конфиденциальностью понимают невозможность получения информации из преобразованного массива без знания дополнительной информации (ключа). Аутентичность информации состоит в подлинности авторства и целостности. Криптоанализ объединяет математические методы нарушения конфиденциальности и аутентичности информации без знания ключей. Существует ряд смежных, но не входящих в криптологию отраслей знания. Так обеспечением скрытности информации в информационных массивах занимается стеганография. Обеспечение целостности информации в условиях случайного воздействия находится в ведении теории помехоустойчивого кодирования.

Современная криптография включает в себя четыре крупных раздела: симметричные криптосистемы, криптосистемы с открытым ключом, системы электронной подписи, управление ключами. Основные направления использования криптографических методов – передача конфиденциальной информации по каналам связи (например, электронная почта), установление подлинности передаваемых сообщений, хранение информации (документов, баз данных) на носителях в зашифрованном виде. В качестве информации, подлежащей шифрованию и расшифрованию, а также электронной подписи будут рассматриваться тексты (сообщения), построенные на некотором алфавите. Под этими терминами понимается следующее.

Алфавит – конечное множество используемых для кодирования информации знаков.

Текст (сообщение) – упорядоченный набор из элементов алфавита.

С теоретической точки зрения не существует четкого различия между кодами и шифрами. Однако в современной практике различие между ними является достаточно четким. Коды оперируют лингвистическими элементами, разделяя шифруемый текст на такие смысловые элементы, как слова и слоги. В шифре всегда различают два

элемента: алгоритм и ключ. Алгоритм позволяет использовать сравнительно короткий ключ для шифрования сколь угодно большого текста. Определим ряд терминов, используемых в криптологии.

Под шифром понимается совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, заданных алгоритмом криптографического преобразования. Криптографическая система, или шифр представляет собой семейство T обратимых преобразований открытого текста в шифрованный. Членам этого семейства можно взаимно однозначно сопоставить число k, называемое ключом. Преобразование T_k определяется соответствующим алгоритмом и значением ключа k.

Ключ – конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор одного варианта из совокупности всевозможных для данного алгоритма. Секретность ключа должна обеспечивать невозможность восстановления исходного текста по шифрованному. Пространство ключей K – это набор возможных значений ключа. Обычно ключ представляет собой последовательный ряд букв алфавита. Следует отличать понятия «ключ» и «пароль». Пароль также является секретной последовательностью букв алфавита, однако используется не для шифрования (как ключ), а для аутентификации субъектов.

Криптосистемы подразделяются на симметричные и асимметричные [или с открытым (публичным) ключом]. В симметричных криптосистемах для зашифрования и для расшифрования используется один и тот же ключ. В системах с открытым ключом используются два ключа открытый (публичный) и закрытый (секретный), которые математически связаны друг с другом. Информация зашифровывается с помощью открытого ключа, который доступен всем желающим, а расшифровывается с помощью закрытого ключа, известного только получателю сообщения.

Термины распределение ключей и управление ключами относятся к процессам системы обработки информации, содержанием которых является выработка и распределение ключей между пользователями.

Зашифрованием данных называется процесс преобразования открытых данных в зашифрованные с помощью шифра, а расшифрованием данных – процесс преобразования закрытых данных в открытые с помощью шифра. Вместо термина «открытые данные» часто употребляются термины «открытый текст» и «исходный текст», а вместо «зашифрованные данные» – «шифрованный текст».

Дешифрованием называется процесс преобразования закрытых данных в открытые при неизвестном ключе и, возможно, неизвестном алгоритме, т.е. методами криптоанализа.

Шифрованием называется процесс зашифрования или расшифрования данных.

Криптостойкостью называется характеристика шифра, определяющая его стойкость к дешифрованию. Обычно эта характеристика определяется периодом времени, необходимым для дешифрования.

1.2 Симметричные криптоалгоритмы

К симметричным криптоалгоритмам относятся скремблеры и блочные шифры.

Скремблерами называются программные или аппаратные реализации алгоритма, позволяющего шифровать побитно непрерывные потоки информации. Сам скремблер представляет собой набор бит, изменяющихся на каждом шаге по определенному алгоритму. После выполнения каждого очередного шага на его выходе появляется шифрующий бит – либо 0, либо

1, который накладывается на текущий бит информационного потока операцией XOR.

Блочные шифры шифруют целые блоки информации (от 4 до 32 байт) как единое целое – это значительно увеличивает стойкость преобразований к атаке полным перебором и позволяет использовать различные математические и алгоритмические преобразования. На сегодняшний день разработано достаточно много стойких блочных шифров. Практически все алгоритмы используют для преобразований определенный набор обратимых математических преобразований. Сеть Фейстеля – метод обратимых преобразований текста, при котором вычисленное от одной из частей текста, накладывается на другие части. Часто сети выполняется таким образом, что для шифрования и дешифрования используется один и тот же алгоритм – различие состоит только в порядке использования материала ключа. Ниже рассмотрим несколько блочных шифров.

Блочный шифр TEA (от англ. Tiny Encryption Algorithm) приведен как пример одного из самых простых в реализации стойких криптоалгоритмов. TEA – блочный алгоритм шифрования типа «Сеть Фейстеля». Алгоритм был разработан на факультете компьютерных наук Кембриджского университета Дэвидом Уилером и Роджером Нидхэмом и впервые представлен в 1994 году на симпозиуме по быстрым алгоритмам шифрования в Левене (Бельгия). Шифр не патентован, широко используется в ряде криптографических приложений и широком спектре аппаратного обеспечения благодаря крайне низким требованиям к памяти и простоте реализации. Алгоритм имеет как программную реализацию на разных языках программирования, так и аппаратную реализацию на интегральных схемах типа FPGA (field-programmable gate array или программируемая пользователем вентильная матрица). Алгоритм шифрования TEA основан на битовых операциях с 64-битным блоком,

имеет 128-битный ключ шифрования. Стандартное количество раундов сети Фейстеля равно 64 (32 цикла), однако, для достижения наилучшей производительности или шифрования, число циклов можно варьировать от 8 (16 раундов) до 64 (128 раундов). Сеть Фейстеля несимметрична из-за использования в качестве операции наложения сложения по модулю 2^{32}. Достоинствами шифра являются его простота в реализации, небольшой размер кода и довольно высокая скорость выполнения, а также возможность оптимизации выполнения на стандартных 32-битных процессорах, так как в качестве основных операций используются операции исключающего «ИЛИ» (XOR), побитового сдвига и сложения по модулю 2^{32}. Поскольку алгоритм не использует таблиц подстановки и раундовая функция довольно проста, алгоритму требуется не менее 16 циклов (32 раундов) для достижения эффективной диффузии, хотя полная диффузия достигается уже за 6 циклов (12 раундов). Алгоритм имеет отличную устойчивость к линейному криптоанализу и довольно хорошую к дифференциальному анализу. Главным недостатком этого алгоритма шифрования является его уязвимость к атакам «на связанных ключах» (от англ. Related-key attack). Из-за простого расписания ключей каждый ключ имеет 3 эквивалентных ключа. Это означает, что эффективная длина ключа составляет всего 126 бит, поэтому данный алгоритм не следует использовать в качестве хеш-функции.

Стандарт AES (англ. Advanced Encryption Standard) является стандартом шифрования США, принятым в 2000-ом году. Он специфицирует алгоритм Rijndael. Этот алгоритм представляет собой симметричный блочный шифр, который работает с блоками данных длиной 128 бит и использует ключи длиной 128, 192 и 256 бит (версии AES-28; AES-192 и AES-256). Сам алгоритм может работать и с другими длинами блоков данных и ключей, но эта возможность в стандарт не вошла. Поддержка AES введена фирмой Intel в семейство процессоров x86

начиная с Intel Core i7-980X Extreme Edition, а затем на процессорах Sandy Bridge.

Алгоритм RC5

RC5 (Ron's Code 5 или Rivest's Cipher 5) – это блочный шифр, разработанный Роном Ривестом из компании RSA Security Inc. с переменным количеством раундов, длиной блока и длиной ключа. Это расширяет сферу использования и упрощает переход на более сильный вариант алгоритма. Существует несколько различных вариантов реализации данного алгоритма, в которых преобразования в «пол-раундах» классического RC5 несколько изменены. В классическом алгоритме используются три примитивных операции и их инверсии:

1) сложение по модулю 2^w;

2) побитовое исключающее ИЛИ (XOR);

3) операции циклического сдвига на переменное число бит.

Шифрование по алгоритму RC5 состоит из двух этапов. Процедура расширения ключа и непосредственно шифрование. Для расшифрования выполняется сначала процедура расширения ключа, а затем операции, обратные процедуре шифрования. Все операции сложения и вычитания выполняются по модулю 2^w. Схема работы алгоритма RC5 представлена на рис. 2.1.

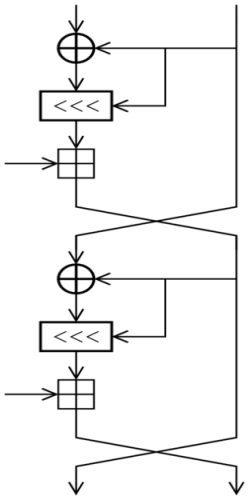

Рис. 2.1. Алгоритм RC5

Поскольку алгоритм RC5 имеет переменные параметры, то для спецификации алгоритма с конкретными параметрами принято обозначение RC5-W/R/b.

W – половина длины блока в битах, возможные значения 16, 32 и 64. Для эффективной реализации величину W рекомендуют брать равным машинному слову. Например, для 32-битных платформ оптимальным будет выбор W=32, что соответствует размеру блока 64 бита.

R – число раундов, возможные значения от 0 до 255. Увеличение числа раундов обеспечивает увеличение уровня безопасности шифра. Так, при R=0 информация шифроваться не будет. Также алгоритм RC5 использует таблицу расширенных ключей размера 2(R+1) слов, которая получается из ключа заданного пользователем.

b – длина ключа в байтах, возможные значения от 0 до 255.

Перед непосредственно шифрованием или расшифрованием данных выполняется процедура расширения ключа. Процедура генерации ключа состоит из четырех этапов: генерация констант, разбиение ключа на слова, построение таблицы расширенных ключей, перемешивание.

Алгоритм RC6

Алгоритм RC6 является продолжением криптоалгоритма RC5, разработанного Рональдом Ривестом (Ron Rivest) – очень известной личностью в мире криптографии. RC5 был незначительно изменен для того, чтобы соответствовать требованиям AES по длине ключа и размеру блока. При этом алгоритм стал еще быстрее, а его ядро, унаследованное от RC5, имеет солидный запас исследований, проведенных задолго до объявления конкурса AES.

Алгоритм является сетью Фейстеля с 4 ветвями смешанного типа: в нем два четных блока используются для одновременного изменения содержимого двух нечетных блоков. Затем производится обычный для сети Фейстеля сдвиг на одно машинное слово, что меняет четные и

нечетные блоки местами. Сам алгоритм предельно прост и изображен на рис. 2.2. Разработчики рекомендуют при шифровании использовать 20 раундов сети, хотя в принципе их количество не регламентируется. При 20 повторах операции шифрования алгоритм имеет самую высокую скорость среди 5 финалистов AES.

Преобразование T(X) очень просто: $T(X)= [(X*(X+1)] \bmod 2^N$. Оно используется в качестве нелинейного преобразования с хорошими показателями перемешивания битового значения входной величины.

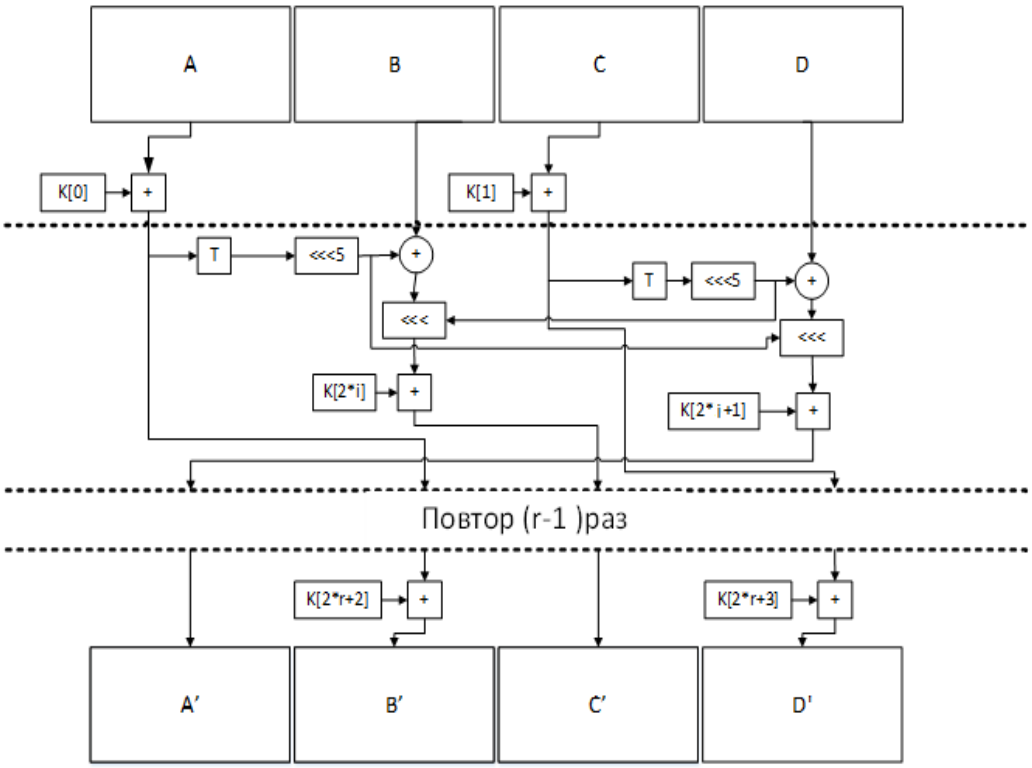

Рис. 2.2. Схема алгоритма RC6

Алгоритм TwoFish

Алгоритм TwoFish разработан компанией Counterpain Security Systems, возглавляемой Брюсом Шнайером (Bruce Schneier). Предыдущая программная разработка этой фирмы, называвшаяся BlowFish, являлась и до сих пор является признанным криптостойким алгоритмом. В алгоритме TwoFish разработчики оставили некоторые удачные решения из

проекта-предшественника, кроме этого произвели тщательные исследования по перемешиванию данных в сети Фейстеля.

Схема работы алгоритма TwoFish представлена на рис. 2.3. Алгоритм представляет собой сеть Фейстеля смешанного типа: первая и вторая ветвь на нечетных раундах производят модификацию третьей и четвертой, на четных раундах ситуация меняется на противоположную. В алгоритме используется криптопреобразование Адамара (Pseudo-Hadamar Transform) – обратимое арифметическое сложение первого потока со вторым, а затем второго с первым.

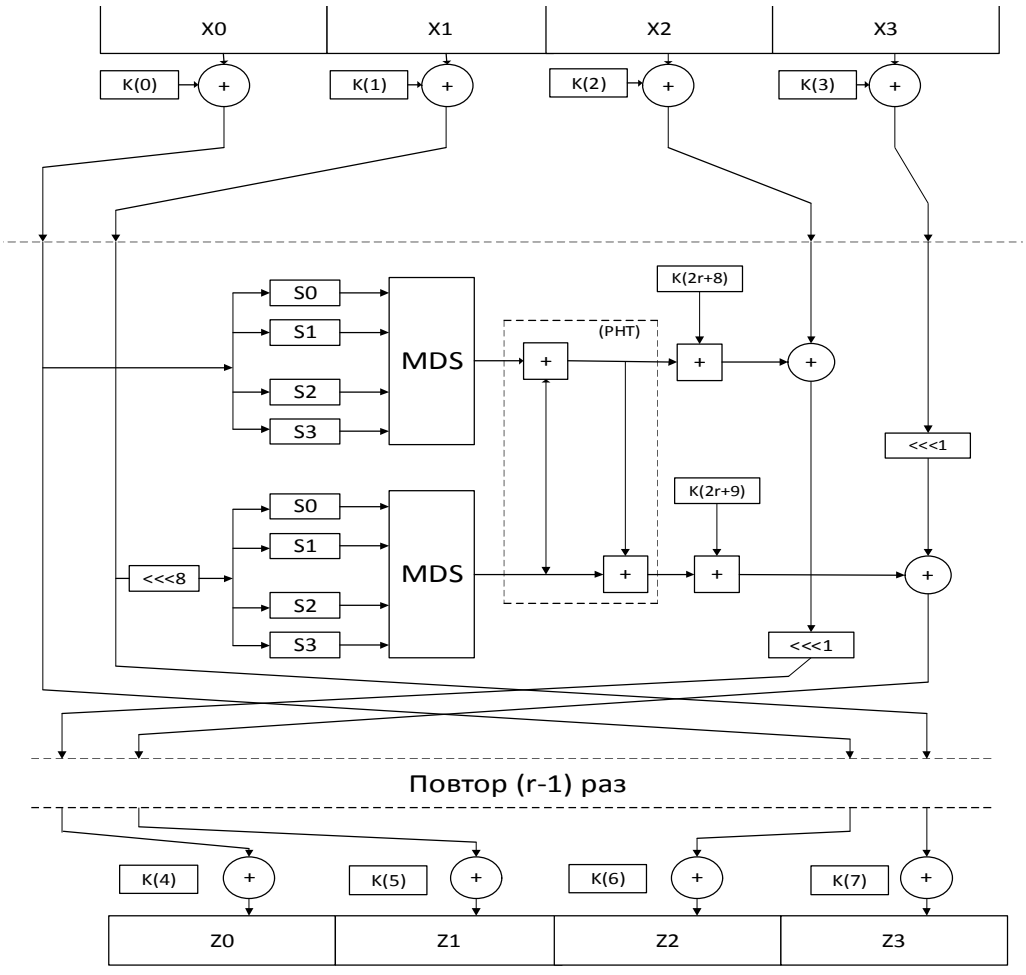

Рис. 2.3. Схема работы алгоритма TwoFish

Единственным нарицанием, поступившим в адрес TwoFish от независимых исследователей, является тот факт, что при расширении материала ключа в алгоритме используется сам же алгоритм. Двойное

применение блочного шифра довольно сильно усложняет его анализ на предмет наличия слабых ключей или недокументированных замаскированных связей между входными и выходными данными.

Алгоритм IDEA

Алгоритм IDEA (от англ. International Data Encryption Algorithm) является блочным шифром. Он оперирует 64-битовыми блоками открытого текста. Несомненным достоинством алгоритма IDEA является то, что его ключ имеет длину 128 бит. Один и тот же алгоритм используется и для шифрования, и для дешифрования. Первая версия алгоритма IDEA была предложена в 1990 г., ее авторы – Х. Лей и Дж. Мэсси. Первоначально алгоритм назывался PES (Proposed Encryption Standard). Улучшенный вариант этого алгоритма, разработанный в 1991 г., получил название IPES (Improved Proposed Encryption Standard). И 1992 г. IPES изменил свое имя на IDEA. Алгоритм IDEA использует при шифровании процессы смешивания и рассеивания, которые легко реализуются аппаратными и программными средствами.

В IDEA используются следующие математические операции:

– поразрядное сложение, но модулю 2 (операция «исключающее ИЛИ»); операция обозначается как (+);

– сложение беззнаковых целых по модулю 2^{16}; операция обозначается как [+];

– умножение беззнаковых целых по модулю (2^{16}+l), причем блок из 16 нулей рассматривается как 2^{16}; операция обозначается как (·).

Все операции выполняются над 16-битовыми субблоками. Эти три операции несовместимы в том смысле, что:

– никакая пара из этих трех операций не удовлетворяет ассоциативному закону;

– никакая пара из этих трех операций не удовлетворяет дистрибутивному закону.

Комбинирование этих трех операций обеспечивает комплексное преобразование входных данных, существенно затрудняя криптоанализ IDEA по сравнению с DES, который базируется исключительно на операции «исключающее ИЛИ». На рис. 2.4 представлена схема работы алгоритма IDEA.

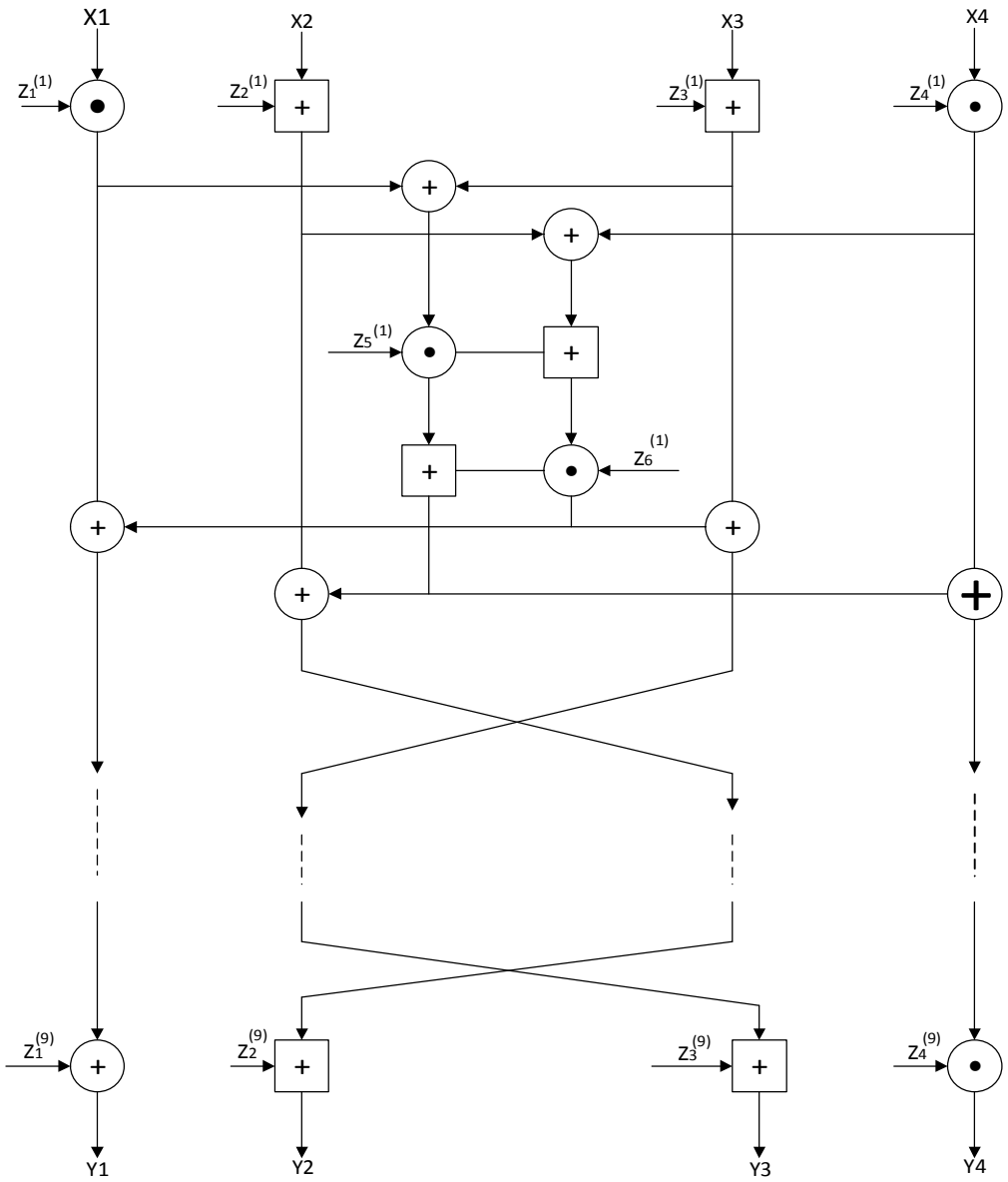

Рис. 2.4. Схема работы алгоритма IDEA (режим шифрования)

64-битовый блок данных делится на четыре 16-битовых субблока. Эти четыре субблока становятся входом в первый цикл алгоритма. Всего выполняется восемь циклов. Между циклами второй и третий субблоки

меняются местами. В каждом цикле выполняется следующая последовательность операций:

1. (·) – умножение субблока X1 и первого подключа;
2. [+] – сложение субблока X2 и второго подключа;
3. [+] – сложение субблока X3 и третьего подключа;
4. (·) – умножение субблока X4 и четвертого подключа;
5. (+) – сложение результатов шагов 1 и 3;
6. (+) – сложение результатов шагов 2 и 4;
7. (·) – умножение результата шага 5 и пятого подключа;
8. [+] – сложение результатов шагов 6 и 7;
9. (·) – умножение результата шага 3 и шестого подключа;
10. [+] – сложение результатов шагов 7 и 9;
11. (+) – сложение результатов шагов 1 и 9;
12. (+) – сложение результатов шагов 3 и 9;
13. (+) – сложение результатов шагов 2 и 10;
14. (+) – сложение результатов шагов 4 и 10.

Выходом цикла являются четыре субблока, которые получаются как результаты выполнения шагов 11, 12, 13 и 14. В завершение цикла второй и третий субблоки меняются местами (за исключением последнего цикла). В результате формируется вход для следующего цикла.

После восьмого цикла осуществляется заключительное преобразование выхода. Полученные четыре субблока Y1...Y4 объединяют в блок шифртекста. Создание подключей Z1...Z6 также относительно несложно. Алгоритм использует всего 52 подключа (по шесть для каждого из восьми циклов и еще четыре для преобразования выхода). Сначала 128-битовый ключ делится на восемь 16-битовых подключей. Это – первые восемь подключей для алгоритма (шесть подключей – для первого цикла и первые два подключа – для второго). Затем 128-битовый ключ циклически сдвигается влево на 25 бит и снова делится на восемь

подключей (четыре подключа – для второго цикла и четыре подключа – для третьего). Ключ снова циклически сдвигается влево на 25 бит для получения следующих восьми подключей и т.д., пока выполнение алгоритма не завершится.

1.3. Асимметричные криптоалгоритмы

Симметричные криптосистемы, несмотря на множество преимуществ, обладают одним серьезным недостатком. Связан он с ситуацией, когда общение между собой производят не три-четыре человека, а сотни и тысячи людей. В этом случае для каждой пары, переписывающейся между собой, необходимо создавать свой секретный симметричный ключ. Кроме того, при нарушении конфиденциальности какой-либо рабочей станции злоумышленник получает доступ ко всем ключам этого пользователя и может отправлять, якобы от его имени, сообщения всем абонентам, с которыми «жертва» вела переписку.

Своеобразным решением этой проблемы явилось появление асимметричной криптографии. Эта область криптографии очень молода по сравнению с другими представителями. Но за время своего существования асимметричная криптография превратилась в одно из основных направлений криптологии, и используется в современном мире также часто, как и симметричные схемы.

Асимметричная криптография изначально задумана как средство передачи сообщений от одного объекта к другому (а не для конфиденциального хранения информации, которое обеспечивают только симметричные алгоритмы). Поэтому дальнейшее объяснение мы будем вести в терминах «отправитель» – лицо, шифрующее, а затем отправляющее информацию по незащищенному каналу и «получатель» – лицо, принимающее и восстанавливающее информацию в ее исходном виде. Основная идея асимметричных криптоалгоритмов состоит в том, что

для шифрования сообщения используется один ключ, а при дешифровании – другой (рис. 2.5). Кроме того, процедура шифрования выбрана так, что она необратима даже по известному ключу шифрования – это второе необходимое условие асимметричной криптографии. То есть, зная ключ шифрования и зашифрованный текст, невозможно восстановить исходное сообщение – прочесть его можно только с помощью второго ключа – ключа дешифрования. А раз так, то ключ шифрования для отправки писем какому-либо лицу можно вообще не скрывать – зная его все равно невозможно прочесть зашифрованное сообщение. Поэтому, ключ шифрования называют в асимметричных системах «открытым ключом», а вот ключ дешифрования получателю сообщений необходимо держать в секрете – он называется «закрытым ключом». Возникает вопрос «Почему, зная открытый ключ, нельзя вычислить закрытый ключ?» – это третье необходимое условие асимметричной криптографии – алгоритмы шифрования и дешифрования создаются так, чтобы, зная открытый ключ, невозможно было вычислить закрытый ключ.

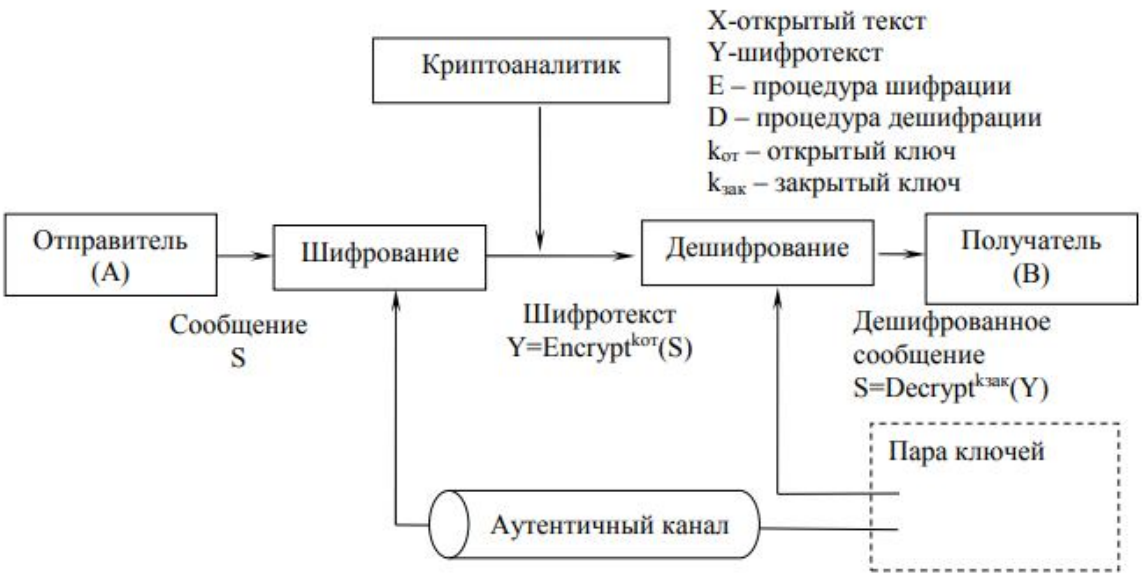

Рис. 2.5. Обобщенная структурная схема асимметричной криптосистемы

Глядя на схему на рис. 2.5 можно сделать вывод, что шифрация информации с использованием закрытого ключа не имеет смысла. Действительно, зачем кодировать информацию, если всякий, кто знает открытый ключ может ее декодировать. Однако, шифрование информации закрытым ключом имеет смысл. Действительно, если мы зашифруем информацию закрытым ключом, а затем сможем дешифровать открытым, то сможем сделать вывод о том, что именно владелец пары ключей и никто другой шифровал информацию (закрытый ключ известен только ему), то есть она становится для нас аутентичной, а в случае использования сертифицированных ключей еще и апеллируемой. Это свойство ключевой пары лежит в основе формирования и верификации электронной цифровой подписи (ЭЦП). ЭЦП – это набор методов, которые позволяют перенести свойства рукописной подписи под документом в область электронного документооборота. Она обеспечивает аутентичность автора сообщения, уникальность подписи, контроль целостности передаваемого сообщения, невозможность переноса подписи под другой документ. Все эти свойства достигаются за счет синтеза асимметричной криптографии и хеширования документов (рис. 2.6).

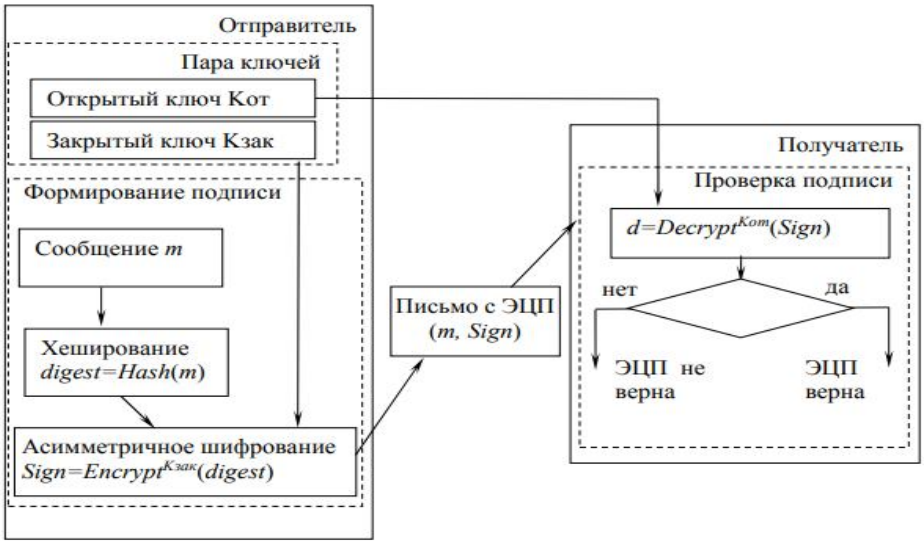

Рис. 2.5. Обобщенная схема процесса формирования/проверки ЭЦП

В настоящее время в банковских компьютерных сетях, особенно для работы с удаленными клиентами (обслуживание кредитных карточек) широко используется алгоритм RSA. Основными математическими результатами, положенными в основу этого алгоритма являются: малая теорема Ферма и функция Эйлера. Открытый текст шифруется блоками, каждый из которых содержит двоичное значение, меньшее некоторого заданного числа *n*. Это значит, что длина блока должна быть меньше или равна $\log_2(n)$. На практике длина блока выбирается равной 2^k битам, где $2^k<n<2^{k+1}$. На рис. 2.6 представлена схема работы алгоритма RSA.

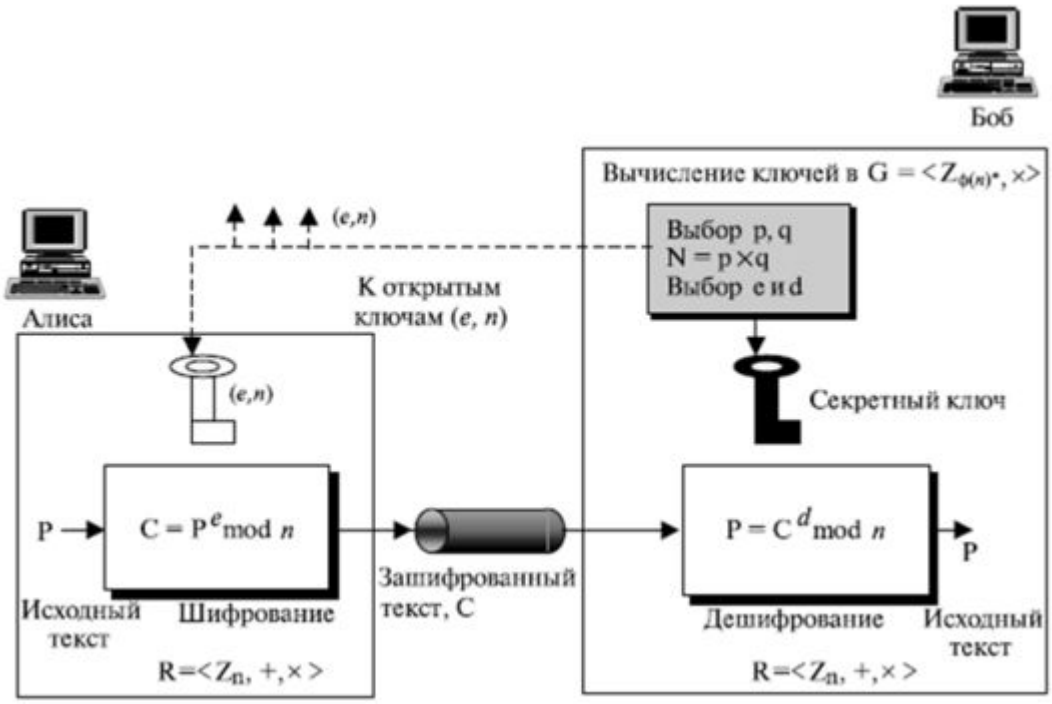

Рис. 2.6. Схема работы алгоритма RSA

2 Описание шифрования и дешифрования в программе SPECTralInfo Crypter

Шифрование.

Если пользователь желает зашифровать какие-либо данные, то он должен выбрать опцию «Зашифровать данные» из стартового окна программы (окна, появляющегося после запуска программы) и нажать кнопку «Вперед». Далее все делается пошагово.

Шаг 1. На этом шаге можно выбрать, что именно вы хотите зашифровать: файл (опция «Файл»), содержимое буфера обмена (опция «Буфер обмена»), или ввести текст для шифрования вручную (опция «Текст»).

Шаг 2. На этом шаге вы можете выбрать алгоритм для шифрования. Алгоритмы различаются своей стойкостью к взлому и скоростью шифрования. Рядом со списком алгоритмов находится диаграмма, иллюстрирующая зависимость этих двух факторов. Рядом с некоторыми алгоритмами даны комментарии об их надежности. Самым надежным на сегодняшний день алгоритмом (из реализованных в программе) является TwoFish.

Шаг 3. На этом шаге программа попросит вас ввести имена исходного файла (который нужно зашифровать) и конечного файла (который будет содержать зашифрованные данные). В целях безопасности Ваших данных программа не будет продолжать работу, если имена исходного и конечного файлов совпадают. Если вы выбрали шифрование текста или буфера обмена, то вместо вопроса об именах файлов появится поле ввода, в которое вы можете ввести текст, а также вставить его из буфера обмена с помощью кнопки «Вставить». Программа интегрируется в Windows, поэтому для шифрования файлов достаточно выбрать пункт «Зашифровать» в его контекстном меню.

Шаг 4. На этом шаге необходимо ввести пароль. Максимальная длина пароля – 85 символов (680 бит). На вскрытие пароля максимальной длины (25685 комбинаций) современными темпами уйдет несколько миллионов лет.

Вы можете вводить пароль «невидимым» (пароль не отображается на экране) текстом (тогда Вам будет предложено ввести подтверждение пароля) или «обычным» (пароль отображается на экране) текстом. Переключатель режимов находится под полями ввода пароля. Рядом с окнами ввода пароля находится индикатор «качества» пароля, чем выше его показания, тем надежней пароль.

Также на этом шаге вы можете включить использование открытого пароля (с помощью флажка «Использовать открытый пароль»). Не забывайте о том, что данные, зашифрованные с использованием открытого пароля, можно расшифровать только с помощью личного пароля. Для управления наборами паролей используйте менеджер паролей.

Завершение шифрования. После ввода пароля некоторое время программа будет обрабатывать данные (появляется заставка). В это время программа не реагирует ни на какие действия пользователя

После завершения шифрования отобразится сообщение о том, что обработка данных завершена; если вы шифровали текстовую информацию, то также отобразится зашифрованный текст, который вы можете скопировать в буфер обмена с помощью кнопки «Копировать».

Дешифрование.

Если пользователь желает расшифровать какие-либо данные, то он должен выбрать опцию «Расшифровать данные» из стартового окна программы (окна, появляющегося после запуска программы) и нажать кнопку «Вперед». Подробное описание шагов приводится ниже.

Шаг 1. На этом шаге можно выбрать, что именно вы хотите расшифровать: файл (опция «Файл»), содержимое буфера обмена (опция

«Буфер обмена»), или ввести текст для расшифровки вручную (опция «Текст»).

Шаг 2. На этом шаге вы можете выбрать алгоритм для расшифровки. Вы должны выбрать тот же самый алгоритм, какой использовали при шифровании. Если Вы забыли, каким алгоритмом Вы пользовались при шифровании, то Вы можете определить его методом перебора всех доступных алгоритмов.

Шаг 3. На этом шаге программа попросит вас ввести имена исходного файла (который нужно расшифровать) и конечного файла (который будет содержать расшифрованные данные). В целях безопасности ваших данных программа не будет продолжать работу, если имена исходного и конечного файлов совпадают. Если вы выбрали расшифровку текста или буфера обмена, то вместо вопроса об именах файлов появится поле ввода, в которое вы можете ввести текст, а также вставить его из буфера обмена с помощью кнопки «Вставить». Программа интегрируется в Windows, поэтому для расшифровки файлов достаточно выбрать пункт «Расшифровать» в его контекстном меню.

Шаг 4. На этом шаге необходимо ввести пароль. Максимальная длина пароля – 85 символов (680 бит). Пароль должен быть тем же, что и при шифровании. Вы можете вводить пароль «невидимым» (пароль не отображается на экране) текстом (тогда Вам будет предложено ввести подтверждение пароля) или «обычным» (пароль отображается на экране) текстом. Переключатель режимов находится под полями ввода пароля. Также на этом шаге вы можете включить использование личного пароля (с помощью флажка «Использовать личный пароль»). Не забывайте о том, что данные, расшифрованные с использованием личного пароля, будут корректны только в том случае, если были зашифрованы с использованием открытого пароля. Для управления наборами паролей используйте менеджер паролей.

После завершения расшифровки отобразится сообщение о том, что обработка данных завершена; если вы расшифровывали текстовую информацию, то, также отобразится расшифрованный текст, который вы можете скопировать в буфер обмена с помощью кнопки «Копировать».

Исходные данные

Таблица 2.1

Исходные данные для выполнения лабораторной работы.

Номер варианта	Тексты	Исследуемый символ	Криптоалгоритм
1	Я человек и ничто человеческое мне не чуждо.	а	TwoFish
	В здоровом теле здоровый дух.		
	Обучая, умчимся.		
	Нет худа без добра.		
	Каждый кузнец своего счастья.		
	Сколько людей, столько и мнений.		
	Никто не становится хорошим случайно.		
	Через сомнение приходим к истине.		
	Все, что благородно, полезно.		
	После дождичка в четверг.		
2	Утро вечера мудренее.	б	RC6
	Опыт лучший учитель.		
	Там беда, где нет согласия.		
	Я выиграл.		
	По секрету.		
	Орел мух не ловит.		
	Я мыслю, значит я существую.		
	Красноречивое молчание.		
	Пока дышу, надеюсь.		
	Незнание не оправдывает.		

3	Деньги не пахнут.	в	CAST 256
	Где леность, там бедность.		
	Если молчишь, соглашаешься.		
	Спеши не спеша.		
	Бумага не краснеет.		
	Жизнью управляет удача, а не мудрость.		
	Природа ничего не делает напрасно.		
	Чужие пороки мы видим, свои не замечаем.		
	Волк не кусает волка.		
	Все прекрасное редко.		
4	Одних судьба ведет, других тащит.	г	IDEA
	Слепой не судит о свете.		
	Одна ласточка весны не делает.		
	Дело прославляет мастера.		
	Всяким вещам есть конец.		
	Закон должен быть кратким.		
	Ни дня без строчки.		
	День учит день.		
	Уходя, уходи.		
	Никто не может знать всего		
5	Каков муж, такова и речь.	д	RC5
	Каков господин, таковы и рабы.		
	Каков царь, таково и племя.		
	Легко истина сама сражается.		
	Платон называл поэтов вождями мудрости.		
	Голос народа – голос истины.		
	Во время мира науки процветают.		
	Медлить опасно.		
	Между сражениями молчат даже музы.		
	Кто любит книги, хорошо учится.		

Задание

1. Шифрование и дешифрование.

Шифрование и дешифрование текстов производится с целью ознакомления с операциями шифрования и дешифрования в программе SPECTralInfo Crypter. Студенты разбиваются на подгруппы (по 2-3 человека). Операции шифрования и дешифрования выполняются поочередно. Результаты шифрования записываются в файл, результаты дешифрования – в табл. 2.2. Задание по шифрованию и дешифрования включает следующие пункты:

1. Зашифруйте текст из табл. 2.1 любым из перечисленных алгоритмов.

2. Полученную криптограмму запишите в Блокноте в файл Shifr.txt, а используемый ключ – в файл Key.txt.

3. Дешифруйте криптограмму из файла Shifr.txt, подбирая алгоритм шифрования и используя ключ из файла Key.txt.

4. Результаты дешифрования занесите в табл. 2.2.

Таблица 2.2

Результаты дешифрования

№ варианта	Исходный текст	Алгоритм	Ключ	Шифртекст
	. . .	. . .	. . .	. . .

2. Криптоанализ

Криптоанализ — это наука о преобразовании шифртекста в открытый текст без знания ключа. В криптоанализе используются следующие атаки:

– полный перебор всех возможных ключей;

– атака на шифртекст (предполагается, что атакующий имеет только шифртекст).

– атака по открытому тексту (предполагает наличие у атакующего возможности зашифровать любой выбранный им текст);

– атака по известному открытому тексту (предполагает знание части открытого текста и соответствующего ему шифртекста);

– атака по времени (атакующий замеряет время, необходимое для операции шифрования или дешифрования).

В задании выполняются элементы криптоанализа, основанные на атаке по открытому тексту. Криптоанализ проводится с целью выявления некоторых закономерностей и включает следующие пункты:

1. Зашифруйте текст, состоящий из исследуемых символов из табл. 2.1, заданным алгоритмом, постоянно добавляя один символ в текст:

1.а

2. аа

3. ааа

…….

16. аааааааааааааааа

2. Анализируя полученные криптограммы, выявите длину повторяющихся блоков криптограмм, а также символы конца неполных блоков.

3. Зашифруйте текст *AAAAAAAA4AAAAAAA*. Измените один символ текста и подсчитайте число изменившихся символов в шифртексте.

4. Выполните пункты 1, 2, 3 для разных ключей*: 000000000, 111111111*, *Ключключ, Xyzxyzxyz.*

5. Результаты криптоанализа занесите в табл. 2.3.

6. Сделайте выводы о свойстве рассеивания алгоритма (рассеивание — это влияние одного знака открытого текста на символы шифртекста)

укажите, какие закономерности при шифровании открытых текстов были выявлены.

Таблица 2.3

Результаты криптоанализа

Текст + символ по варианту	Шифротекст	Ключ
a		000000000
aa		000000000
aaa		000000000
aaaa		000000000
……		……...
aaaaaaaaaaaaaaaa		
baaaaaaaaaaaaaaa		000000000
a		111111111
aa		111111111
aaa		111111111
aaaa		
……		……
aaaaaaaaaaaaaaaa		
baaaaaaaaaaaaaaa		111111111
……		……
baaaaaaaaaaaaaaa		Xyzxyzxyz
……		……
baaaaaaaaaaaaaaa		ключключ

3. *Выводы.* По результатам выполненного задания необходимо сделать выводы, которые могут быть формулированы следующие выводы (пример):

1. Шифртексту <шифртекст> соответствует открытый текст <текст>. Криптоалгоритм оперирует блоками текста, длина блока равна 2 символам. Неполный блок заканчивается символом Q. Изменение одного символа текста приводит к изменению всех символов шифртекста. Плохие ключи (000000000 или 111111111) позволяют выявить дополнительные закономерности (какие?).

Содержание отчета

Отчет по лабораторной работе включает следующие пункты:

l. Цель работы.

2. Схема алгоритма и описание ее работы.

3. Исходные данные.

4. Результаты шифрования и дешифрования.

5. Результаты и выводы по криптоанализу алгоритмов шифрования.

Контрольные вопросы

1. Опишите симметричную и асимметричную криптосистемы.

2. Каков порядок генерации и использования ключей в симметричных и ассиметричных криптоалгоритмах?

3. Назовите известные вам блочные шифры.

4. Назовите блочные шифры с переменной длиной ключа и с постоянной длиной ключа

5. Какие операции используются при шифровании в алгоритме RC5 и в алгоритме RC6?

6. Опишите алгоритм RC6 как сеть Фейстеля.

7. Какие операции используются при шифровании в TwoFish?

8. Опишите алгоритм TwoFish как сеть Фейстеля.

9. Какие операции используются в алгоритме шифрования IDEA?

10. Опишите схему шифрования алгоритма IDEA.

11. Назовите известные вам асимметричные алгоритмы.

12. Каков порядок использования ключей в алгоритме RSA при шифровании, при ЭЦП и при шифровании и ЭЦП?

13. Какие операции используются в криптоалгоритме RSA?

14. Укажите область применения алгоритмов RC5, RC6, TwoFish, IDEA, RSA.

15. Какие атаки используются при криптоанализе?