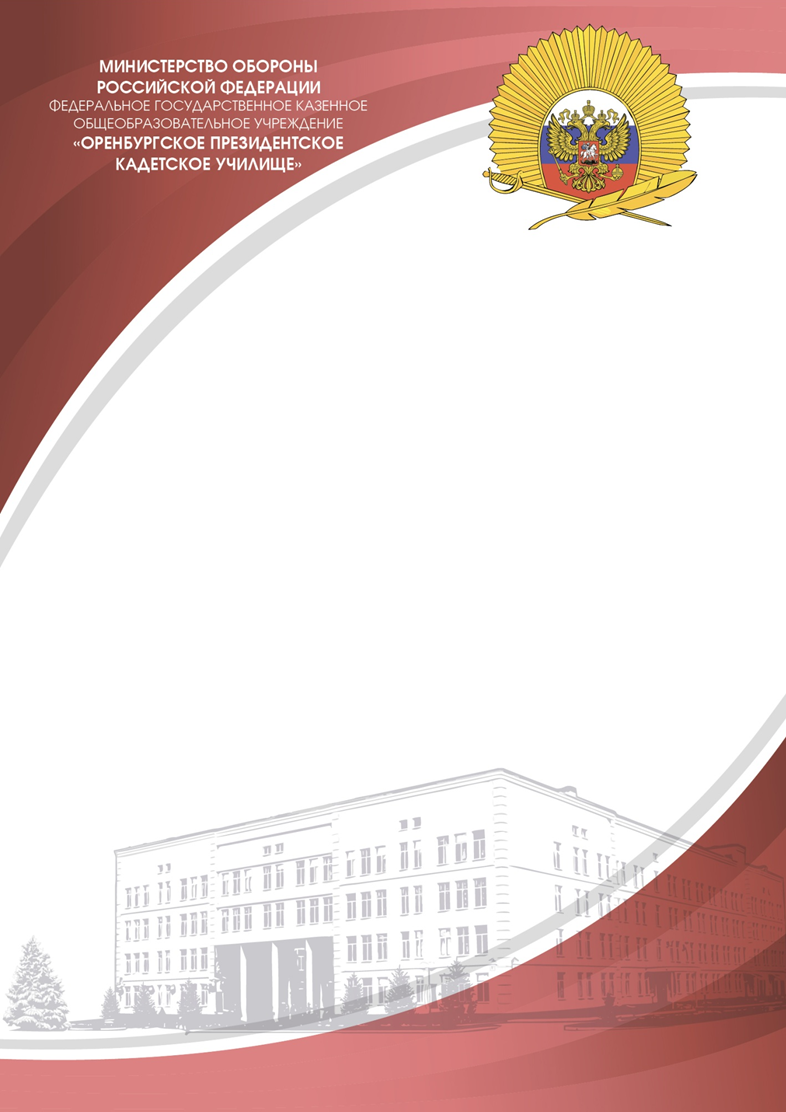
****

**Элективный курс по**

**КРИПТОГРАФИИ**

***(9 классы)***

методическая разработка

Оренбург, 2016

УДК 372.8 – преподавание отдельных дисциплин

**Рецензент:**

Дженжер В.О., кандидат физико-математических наук, доцент, заведующий кафедры информатики и методики преподавания информатики ФГБОУ ВПО "Оренбургский государственный педагогический университет".

**Составитель:**

**Евлампьев А.В. Элективный курс по Криптографии (9 классы): методическая разработка. – Оренбург: Оренбургское ПКУ, 2016. –67 с.**

В разработке представлены методические рекомендации и учебная программа по проведению элективного курса по Криптографии (отбор содержания, формы занятий, контроль знаний и др.), который направлен на оказание помощи учащимся в выборе «своего» профиля, для обучения в старших классах, в направлении учеников в необходимую на сегодняшний день сферу деятельности, в сферу цифровых технологий и защиты информации.

Рассмотрено на заседании предметно-методической кафедры информатики и ИКТ Оренбургского ПКУ (протокол № 1 от 18.01.2016 г.). Рекомендовано к использованию в образовательном процессе училища.

© ФГКОУ «Оренбургское президентское кадетское училище», 2016

© Евлампьев А.В., 2016

Содержание

[Введение 4](#_Toc438752334)

[1 Концепция профильного образования 5](#_Toc438752335)

[2 Структура элективного курса 9](#_Toc438752336)

[2.1 Пояснительная записка 9](#_Toc438752337)

[2.2 Учебно-тематический план элективного курса «Криптография» 10](#_Toc438752338)

[2.3 Содержание элективного курса 11](#_Toc438752339)

[***Занятие №1.*** *История криптологии. Первые послания. Простейшие алгоритмы шифрования* 11](#_Toc438752340)

[***Занятие №2.*** *Программирование простейших алгоритмов* 16](#_Toc438752341)

[***Занятие №3.*** *Классификация алгоритмов шифрования. Характеристики алгоритмов* 17](#_Toc438752342)

[***Занятие №4.*** *Симметричные криптосистемы* 26](#_Toc438752343)

[***Занятие №5.*** *Шифрование посредством алгоритма с закрытым ключом* 28](#_Toc438752344)

[***Занятие №6.*** *Асимметричные криптосистемы* 29](#_Toc438752345)

[***Занятие №7.*** *Шифрование посредством алгоритма с открытым ключом* 31](#_Toc438752346)

[***Занятие №8.*** *Комбинированные алгоритмы* 33](#_Toc438752347)

[***Занятие №9.*** *Электронная цифровая подпись* 35](#_Toc438752348)

[***Занятие №10.*** *Официальные стандарты AES и ГОСТ* 38](#_Toc438752349)

[***Занятие №11.*** *Элементы криптоанализа* 42](#_Toc438752350)

[***Занятие №12.*** *Перспективные направления криптографии* 52](#_Toc438752351)

[Заключение 55](#_Toc438752352)

[Список использованных источников 56](#_Toc438752353)

[Приложения 58](#_Toc438752354)

# Введение

*Чтоб мысль врага узнать, сердца вскрывают, а письма и подавно.*

В. Шекспир, Король Лир.

О важности сохранения информации в тайне знали уже в древние времена, когда с появлением письменности появилась и опасность прочтения её нежелательными лицами. На сегодняшний день количество передаваемой информации возросло в миллионы и миллиарды раз, поэтому актуальность её защиты от несанкционированного доступа возрастает с каждым новым битом информации.

В соответствии с Концепцией профильного образования, введённой в нашей стране с 2002 года, «профильное обучение - средство дифференциации и индивидуализации обучения». Профильное обучение позволяет более полно учитывать личностные интересы, склонности и способности учащихся и создаёт условия для получения образования в зависимости от их профессиональной направленности и желаний.

Известно, что неотъемлемой частью профильного обучения является организация и проведение элективных курсов по предметам.

Поэтому разработка новых учебных пособий, методических рекомендаций, учебных программ является важной задачей педагогической науки. Элективные курсы – это обязательные для посещения курсы по выбору учащихся, которые представляют широкие возможности для реализации принципов дифференцированного обучения, так как позволяют учитывать интересы учащихся, которые хотели бы получить углублённые знания по интересующему их направлению. Разработка программ таких курсов является важной и новой задачей современного школьного образования.

В связи с профилизацией современных старших классов, **актуальной** остаётся проблема оказания помощи учащимся в выборе «своего» профиля, для обучения в старших классах, в направлении учеников в необходимую на сегодняшний день сферу деятельности, в сферу цифровых технологий и защиты информации.

# 1 Концепция профильного образования

В соответствии с Концепцией профильного образования, «профильное обучение - средство дифференциации и индивидуализации обучения, позволяющее за счёт изменений в структуре, содержании и организации образовательного процесса более полно учитывать интересы, склонности и способности учащихся, создавать условия для обучения старшеклассников в соответствии с их профессиональными интересами и намерениями в отношении продолжения образования».

Основными **задачами профильного обучения**являются:

создание условий для учёта и развития учебно-познавательных и профессиональных интересов, способностей и потребностей учащихся в процессе их общеобразовательной подготовки;

воспитание в учащихся любви к труду, обеспечение условий для их жизненного и профессионального самоопределения, формирование готовности к сознательному выбору и овладению будущей профессией;

формирование социальной, коммуникативной, информационной, технической, технологической компетенций учащихся на допрофильном уровне, направленность обучения на выбор будущей профессии;

обеспечение перспективной связи между общим средним и будущим профессиональным образованием [11].

Основными **принципами профильного обучения и предпрофильной подготовки** учащихся считаем следующие.

Принцип фуркации. Его сущность состоит в том, чтобы образовательная система выполняла свои основные социальные функции - селекционную, тестирующую, распределительную в соответствии с современным ей социальным заказом. Поэтому во время общеобразовательной подготовки учащихся происходят процессы распределения учащихся по уровню образовательной подготовки, по интересам, потребностям, способностям и наклонностям. Этот процесс происходит в соответствии с возрастными особенностями школьников и детерминирован потребностями в специалистах различного профиля в региональном и общегосударственном контексте.

Принцип вариативности и альтернативности. Согласно этому принципу создаётся современное содержание образования в предпрофильных и профильных классах, построенное на постоянном разрешении диалектического противоречия между обязательными и вариативными курсами (в количественном и качественном отношении). Кроме этого, принципу вариативности и альтернативности соответствует построение учебных программ, технологий обучения и учебно-методического обеспечения учебного процесса в предпрофильных и профильных классах современной школы.

Принцип последовательности и непрерывности состоит в необходимости последовательной дифференциации учащихся соответственно их интересам, способностям, наклонностям в различных сферах человеческого знания, а также в последовательной и непрерывной взаимосвязи между предпрофильной подготовкой и профильным обучением, профильным и профессиональным обучением.

Принцип гибкости реализуется в создании таких форм организации профильного обучения, которые наиболее бы соответствовали запросам как индивидуума, так и общества; к таким гибким формам относится и модель организации профильного обучения, основанная на сетевой кооперации. Кроме этого принцип гибкости в организации профильного обучения обеспечивает возможность изменения профиля в соответствии с изменениями в интересах и наклонностях учащегося в ходе предпрофильной и профильной подготовки [18].

Согласуясь с нормативно-правовыми основами организации профильного обучения в отечественной системе образования, мы трактуем *основную категорию нашего исследования* - **профильное обучение** - как систему организации среднего образования, при которой в старших классах обучение проходит по разным программам (профилям) с преобладанием тех или иных предметов.

Можно выделить несколько вариантов или моделей организации профильного обучения, представленных в «Концепции профильного образования».

**1) Модель внутришкольной профилизации**

Отдельная школа может быть однопрофильной (реализовывать только один из избранных ею профилей), или организовать на старшей ступени несколько профилей, т.е. быть многопрофильной. Возможен вариант, когда школа в целом не ориентирована на конкретные (один или несколько) фиксированные профили, но за счёт значительного увеличения числа элективных курсов представляет школьникам - в том числе, в форме многообразных учебных межклассных групп - в полной мере осуществлять свои индивидуальные профильные образовательные программы, включая в них те или иные профильные и элективные курсы.

**2) Модель сетевой организации**

В подобной модели профильное обучение учащихся конкретной школы осуществляется за счёт целенаправленного и организованного привлечения образовательных ресурсов иных образовательных учреждений. Оно может строиться в *двух основных вариантах*.

Первый связан с объединением нескольких школ вокруг наиболее сильной школы, обладающей достаточным материальным и кадровым потенциалом, которая для группы школ выполняет роль «ресурсного центра». В этом случае каждая из школ данной группы обеспечивает в полном объёме базовые общеобразовательные курсы и ту часть профильного обучения (профильные и элективные курсы), которую она способна реализовать в рамках своих возможностей. Остальную профильную подготовку берет на себя «ресурсный центр».

Второй вариант основан на кооперации школы с иными образовательными учреждениями и образовательными ресурсами - учреждений дополнительного, высшего, среднего и начального профессионального образования. В этом случае учащимся предоставляется право выбора получения профильного образования либо в собственной школе, либо в кооперированных с ней образовательных структурах дистанционные курсы, заочные школы, Малые Академии при вузах, учреждения системы НПО/СПО и др.).

Поиск новых решений проблемы педагогического сопровождения профессионального самоопределения учащейся молодёжи в условиях профильного обучения в системе сетевой кооперации предполагает:

усиление интеграции образовательных и предметных областей с внеучебной практикой социально-профессионального самоопределения школьников;

ориентацию на консолидацию ресурсов и усилий с другими учебными заведениями (межшкольными учебными комбинатами, училищами, колледжами, лицеями, вузами);

обеспечение профильной подготовки школьников на основе вариативности, с учётом заявленных ими индивидуальных маршрутов, соответствующих интересам, склонностям, способностям и запросам рынка труда, а также обязательной предпрофильной подготовки учащихся, включающей овладения школьниками способностью получения представлений об образе «Я», мире профессий, рынке труда; приобретение практического опыта для обоснованного выбора профиля обучения;

оказание психолого-педагогической помощи учителю в переориентации его с деятельности назидателя, ментора на деятельность фасилитатора, направленную на эмпатию, обеспечение успеха школьника в выборе профиля обучения и дальнейшей профессиональной деятельности.

Анализ сущности профильного образования, исходя из предыдущего опыта профилизации и разработанных нормативно-правовых документов позволяет определить основные факторы реализации требований к организации профильного обучения в старшей школе, а именно:

Интенсивное, подробное информирование на предпрофильной стадии обо всех возможных путях продолжения образования.

Разработка учебных планов старшей школы на основе потребностей учащихся, при которой не школа предлагает готовый профильный пакет (предметов), а учащийся создаёт индивидуальный учебный план.

Организация элективных групп на предпрофильной стадии и профильных классов на старшей ступени на основе использования вариативного компонента базисного учебного плана.

Укрупнение школы «7 - 9» и её отделение от школы «1 - 6» в связи со спецификой задач предпрофильной подготовки.

Система формирования самоопределения учащихся основной школы и вовлечение родителей в определение профиля их обучения на третьей ступени.

Гуманистическая парадигма в преподавании как профильных, так и непрофильных предметов. Согласно гуманистической парадигме в центре обучения не предмет, а учащийся. В преподавании первичны интересы и возможности ученика, а не логика предмета и соответствующей науки.

«Плавающий» стандарт преподавания профильных и непрофильных предметов - не строго заданный базовый или профильный уровень, а уровень не ниже базового в соответствии с зоной ближайшего развития учащихся. На уровне образовательного учреждения - построение программы профильных и непрофильных предметов на основании зоны ближайшего развития учащихся.

Переход от «вычитательной» оценки (учитывающей ошибки и незнание учащихся в виде понижения оценки) к **«**накопительной**»** (количественно и качественно учитывающей достижения учащегося).

Повышение квалификации педагогов, включённое в деятельность педагогических коллективов образовательных учреждений по профессионализации образовательного пространства, направленное на: переход на гуманистическую парадигму содержания образования, формирование деятельной гуманистической позиции; освоение современных педагогических технологий, необходимых для реализации профильного обучения.

# 2 Структура элективного курса

## 2.1 Пояснительная записка

Программа элективного курса «Криптография» предназначена для предпрофильной подготовки учащихся 9 класса с ориентацией на информационно-технологический профиль обучения. Разрабатываемый элективный курс является предметным прикладным. В ходе изучения школьной дисциплины «Информатика и ИКТ» очень поверхностно изучаются основы криптологии и криптографии в частности, поэтому целями элективного курса являются:

1. Знакомство учащихся с важнейшими путями и методами применения знаний по предмету на практике, развитие интереса учащихся к современным информационным технологиям, программированию, шифрованию;

2. Оказание помощи в выборе профиля дальнейшего образования.

Данный образовательный курс расширяет и углубляет базовый компонент образования по дисциплине «Информатика и ИКТ», обеспечивает интеграцию с математикой с элементами теории чисел.

Он позволит полнее учесть интересы и профессиональные намерения старшеклассников, и, следовательно, сделать обучение более интересным для учащихся, получить более высокие результаты при сдаче государственной итоговой аттестации.

**Задачи** курса:

- способствовать самоопределению ученика и/или выбору дальнейшей профессиональной деятельности;

- создавать положительную мотивацию обучения на планируемом профиле;

- познакомить учащихся с ведущими для данного профиля видами деятельности;

- активизировать познавательную деятельность школьников;

- повысить информационную и коммуникативную компетентность учащихся;

- развивать умения самостоятельно оценивать и принимать решения;

- расширить кругозор учащихся.

## 2.2 Учебно-тематический план элективного курса «Криптография»

Элективный курс «Криптография» рассчитан на 34 часа, по одному занятию в неделю в течение всего учебного года. Занятия и выполняемые задания постепенно усложняются к концу курса. Расчёт часов и тематика занятий представлены в таблице 2.1.

Таблица 2.1

Учебно-тематический план

| № п/п | Наименование разделов и тем | Кол-во часов | Вид занятия | Формы текущего контроля успеваемости и промежуточной аттестации |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
|  | История криптологии. Первые послания. Простейшие алгоритмы шифрования | 1 | ТЗ | Шифровка и расшифровка посланий Цезаря в тетради (Excel) |
|  | Программирование простейших алгоритмов | 4 | ПЗ | Разработка 2-х программ для применения алгоритма Цезаря, квадрата Полибия |
|  | Классификация алгоритмов шифрования. Характеристики алгоритмов | 2 | ТЗ |  |
|  | Симметричные криптосистемы | 2 | ТЗ | Шифрование в Excel, элементы криптоанализа – вычисление ключа |
| 1. П | Шифрование посредством алгоритма с закрытым ключом | 6 | ПЗ | Модернизация программы шифра Цезаря с разными ключами. Создание сети для передачи шифровок |
|  | Асимметричные криптосистемы | 2 | ТЗ |  |
|  | Шифрование посредством алгоритма с открытым ключом | 8 | ПЗ | Разработка программы для шифрования текста алгоритмом с открытым ключом |
|  | Комбинированные алгоритмы | 1 | ТЗ |  |
|  | Электронная цифровая подпись | 1 | ТЗ |  |
|  | Официальные стандарты AES и ГОСТ | 2 | ТЗ |  |
|  | Элементы криптоанализа | 3 | ПЗ | Разработка программ для выяснения ключа шифра Цезаря |
|  | Перспективные направления криптографии | 2 | ТЗ |  |
|  | **Итого:** | **34** |  |  |

## 2.3 Содержание элективного курса

***Занятие №1.*** *История криптологии. Первые послания. Простейшие алгоритмы шифрования*

***Цели занятия:***

* познакомить учащихся с историей криптографии;
* изучить шифр Цезаря и квадрата Полибия;

***Задачи занятия:***

* объяснить актуальность защиты информации;
* рассказать об этапах развития криптографии и особенностях каждого этапа;
* рассмотреть принципы простейших алгоритмов перестановки и моноалфавитной подстановки.

***Учебные вопросы, рассматриваемые на занятии:***

1. Информационная безопасность. Актуальность защиты информации.

2. Развитие криптографии от папируса до компьютера.

***Материал для проведения занятия***

*1.* *Информационная безопасность. Актуальность защиты информации.*

Последнее время сообщения об атаках на информацию, о хакерах и компьютерных взломах наполнили все средства массовой информации. Что же такое «атака на информацию»? Дать определение этому действию на самом деле очень сложно, поскольку информация, особенно в электронном виде, представлена сотнями различных видов. Информацией можно считать и отдельный файл, и базу данных, и одну запись в ней, и целиком программный комплекс. И все эти объекты могут подвергнуться и подвергаются атакам со стороны некоторой социальной группы лиц.

При хранении, поддержании и предоставлении доступа к любому информационному объекту его владелец, либо уполномоченное им лицо, накладывает явно либо самоочевидно набор правил по работе с ней. Умышленное их нарушение классифицируется как атака на информацию.

С массовым внедрением компьютеров во все сферы деятельности человека объем информации, хранимой в электронном виде, вырос в тысячи раз. И теперь скопировать за полминуты и унести дискету с файлом, содержащим план выпуска продукции, намного проще, чем копировать или переписывать кипу бумаг. А с появлением компьютерных сетей даже отсутствие физического доступа к компьютеру перестало быть гарантией сохранности информации [14].

Понятие «Безопасность» охватывает широкий круг интересов, как отдельных лиц, так и целых государств. В наше мобильное время цифровых технологий, когда интернет проник во все сферы деятельности человека: начиная от пересылки SMS и заканчивая базами данных огромных предприятий и организаций, системами управления автоматизированными процессами запусками ракет и т.д., видное место отводится проблеме информационной безопасности, обеспечению защиты конфиденциальной информации.

Полугодовой отчёт Cisco по информационной безопасности за 2014 год, обнародованный в США на конференции Black Hat, содержит анализ «слабых звеньев» информационных систем. Такие «слабые звенья» (ими могут быть устаревшее ПО, плохо написанный код, ошибки пользователей или оставленные без внимания информационные активы) облегчают злоумышленникам использование уязвимостей, например, с помощью запросов DNS, наборов эксплойтов, усиленных атак, компрометации POS-систем, вредоносной рекламы, программ-вымогателей, протоколов шифрования, социальной инженерии и спама, адаптированного к реальным событиям.

В отчёте показано, что организации, уделяющие внимание только наиболее популярным уязвимостям в ущерб высококритичным, привычным и скрытым угрозам, подвергают себя большему риску. Злоумышленники, раз за разом атакующие оставленные без внимания устаревшие приложения и инфраструктуры с известными уязвимостями, могут так и не быть обнаружены, поскольку специалисты в области ИБ заняты более заметными уязвимостями, такими как Heartbleed.

**По словам Джона Н. Стюарта (John N. Stewart), старшего вице-президента и директора компании Cisco по информационной безопасности,** «многие компании в своём развитии полагаются на Интернет. Чтобы преуспеть в этой стремительно меняющейся среде, исполнительное руководство должно учитывать и брать, в организационном смысле, под контроль сопутствующие киберриски. Анализ и понимание уязвимостей системы информационной безопасности в значительной мере основаны на способности отдельных организаций и всей отрасли добиться осведомлённости о киберрисках на самых высоких уровнях, в том числе в советах директоров, и сделать кибербезопасность не просто технологией, а частью деятельности организации. Сегодня, чтобы уметь противостоять атаке в течение всего её жизненного цикла, организации должны использовать решения для информационной безопасности, способные действовать везде, где угрозы могут обнаружиться» [12].

О важности сохранения информации в тайне знали уже в древние времена, когда с появлением письменности появилась и опасность прочтения её нежелательными лицами. На сегодняшний день количество передаваемой информации возросло в миллионы и миллиарды раз, поэтому а**ктуальность** её защиты от несанкционированного доступа возрастает с каждым новым битом информации.

Существовали три основных способа защиты информации. Один из них предполагал защиту её чисто силовыми методами: охрана документа - носителя информации - физическими лицами, передача его специальным курьером и т.д. Второй способ получил название «стеганография» латино-греческое сочетание слов, означающих в совокупности «тайнопись». Он заключался в сокрытии самого факта наличия информации. Третий способ защиты информации заключался в преобразовании смыслового текста в некий набор хаотических знаков (или букв алфавита). Получатель данного донесения имел возможность преобразовать его в то же самое осмысленное сообщение, если обладал ключом к его построению. Этот способ защиты информации называется криптографическим. Криптография – (от др.-греч. κρυπτός — скрытый и γράφω — пишу) - «скрыто пишу». По утверждению ряда специалистов криптография по возрасту - ровесник египетских пирамид. В документах древних цивилизаций - Индии, Египта, Месопотамии - есть сведения о системах и способах составления шифрованных писем [2].

Исторически стеганография появилось первым, но затем во многом было вытеснено криптографией. Тайнопись осуществляется самыми различными способами. Общей чертой этих способов является то, что скры­ваемое сообщение встраивается в некоторый безобидный, не привлекающий внимание объект. Затем этот объект открыто транспортируется адресату. При криптографии наличие шифрованного сообщения само по себе привлекает внимание противников, при стеганографии же наличие скрытой связи остаётся незаметным [7]. ­

*2. Развитие криптографии от папируса до компьютера.*

Историю криптографии условно можно разделить на 4 этапа:

1. Наивная криптография.

2. Формальная криптография.

3. Научная криптография.

4. Компьютерная криптография.

Для **наивной криптографии** (до XV века) характерно использование любых (обычно примитивных) способов запуты­вания противника относительно содержания шифруемых тек­стов. На начальном этапе для защиты информации использова­лись методы *кодирования* и *стеганографии,* которые родствен­ны, но не тождественны криптографии.

Большинство из используемых шифров сводились к *пере­становке* или *моноалфавитной подстановке.* Одним из первых зафиксированных примеров является шифр Цезаря, состоящий в замене каждой буквы исходного текста на другую, отстоящую от неё в алфавите на определённое число позиций. Пример данного шифра приведён в таблице 2.2. В данном случае ключ равен 3, то есть каждая буква заменяется на другую, сдвинутую на 3 позиции.

Таблица 2.2

Пример шифра Цезаря с ключом равным 3

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| А | Б | В | Г | Д | Е | Ё | … | Э | Ю | Я |
| Г | Д | Е | Ё | Ж | З | И | … | А | Б | В |

Другой шифр, полибианский квадрат, авторство которого приписывает­ся греческому писателю Полибию, является общей моноалфа­витной подстановкой, которая проводится с помощью случайно заполненной алфавитом квадратной таблицей (для греческого алфавита размер составляет 5×5). Каждая буква исходного тек­ста заменяется на букву, стоящую в квадрате снизу от неё. Для русского алфавита квадрат Полибия представлен в таблице 2.3.

Этап **формальной криптографии** (кон. XV века (IX века на Ближнем Востоке (Ал-Кинди)) - нач. XX века) связан с появлением формализованных и относительно стойких к ручному криптоанализу шифров. В европейских странах это произошло в эпоху Возрождения, когда развитие науки и торговли вызвало спрос на надёжные способы защиты информации. Важная роль на этом этапе принадлежит Леону Батисте Альберти, итальянскому архитектору, который одним из первых предложил *многоалфавитную подстановку.* Данный шифр, получивший имя дипломата XVI века Блеза Вижинера, состоял в последовательном «сложении» букв исходного текста с ключом (процедуру можно облегчить с помощью специальной таблицы). Его работа «Трактат о шифре» (1466) считается пер­вой научной работой по криптологии.

Таблица 2.3

Пример полибианского квадрата для русского алфавита

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ё | Й | Ц | У | К | Е |
| Н | Г | Ш | Щ | З | Х |
| Ъ | Ф | Ы | В | А | П |
| Р | О | Л | Д | Ж | Э |
| Я | Ч | С | М | И | Т |
| Ь | Б | Ю |  | , | . |
| - | : | ; | < | > | = |

Одной из первых печатных работ, в которой обобщены и сформулированы известные на тот момент алгоритмы шифро­вания является труд «Полиграфия» (1508 г.) немецкого аббата Иоганна Трисемуса. Ему принадлежат два небольших, но важ­ных открытия: способ заполнения полибианского квадрата (первые позиции заполняются с помощью легко запоминаемого ключевого слова, остальные - оставшимися буквами алфавита) и шифрование пар букв (биграмм).

В XIX веке голландец Керкхофф сформулировал главное требование к криптографическим системам, которое остаётся актуальным и поныне: секретность шифров должна быть осно­вана на секретности ключа, но не алгоритма.

Наконец, последним словом в донаучной криптографии, ко­торое обеспечили ещё более высокую криптостойкость, а так­же позволило автоматизировать (в смысле механизировать) процесс шифрования стали *роторные криптосистемы.*

Одной из первых подобных систем стала изобретённая в 1790 году Томасом Джефферсоном, будущим президентом США механическая машина. Многоалфавитная подстановка с помощью роторной машины реализуется вариацией взаимного положения вращающихся роторов, каждый из которых осуще­ствляет «прошитую» в нем подстановку.

Практическое распространение роторные машины получили только в начале XX века. Одной из первых практически исполь­зуемых машин, стала немецкая Enigma, разработанная в 1917 году Эдвардом Хеберном и усовершенствованная Артуром Кирхом. Роторные машины активно использовались во время второй мировой войны. Помимо немецкой машины Enigma использовались также устройства Sigaba (США), Турех (Велико­британия), Red, Orange и Purple[[1]](#footnote-1) (Япония). Роторные системы - вершина формальной криптографии так как относительно про­сто реализовывали очень стойкие шифры. Успешные криптоатаки на роторные системы стали возможны только с появлени­ем ЭВМ в начале 40-х годов.

Главная отличительная черта **научной криптографии** (30-е - 60-е годы XX века) - появление криптосистем со строгим ма­тематическим обоснованием криптостойкости. К началу 30-х годов окончательно сформировались разделы математики, яв­ляющиеся научной основой криптологии: теория вероятностей и математическая статистика, общая алгебра, теория чисел, на­чали активно развиваться теория алгоритмов, теория информа­ции, кибернетика. Своеобразным водоразделом стала работа Клода Шеннона «Теория связи в секретных системах» (1949), где сформулированы теоретические принципы криптографиче­ской защиты информации. Шеннон ввёл понятия «рассеивание» и «перемешивание», обосновал возможность создания сколь угодно стойких криптосистем.

В 60-х годах ведущие криптографические школы подошли к созданию *блочных шифров,* ещё более стойких по сравнению с роторными криптосистемами, однако допускающие практиче­скую реализацию только в виде цифровых электронных уст­ройств.

**Компьютерная криптография** (с 70-х годов XX века) обя­зана своим появлением вычислительным средствам с произво­дительностью, достаточной для реализации криптосистем, обес­печивающих при большой скорости шифрования на несколько порядков более высокую криптостойкость, чем «ручные» и «механические» шифры.

Первым классом криптосистем, практическое применение которых стало возможно с появлением мощных и компактных вычислительных средств, стали блочные шифры. В 70-е годы был разработан *американский стандарт шифрования DES* (принят в 1978 году). Один из его авторов, Хорст Фейстел (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.*

С появлением DES обогатился и криптоанализ, для атак на американский алгоритм был создано несколько новых видов криптоанализа (линейный, дифференциальный и т.д.), практи­ческая реализация которых опять же была возможна только с появлением мощных вычислительных систем.

В середине 70-х годов произошёл 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 году под названием «Новые направления в современной криптографии». В ней впервые сформулированы принципы об­мена шифрованной информацией без обмена секретным клю­чом. Асимметричная криптография открыла сразу несколько новых прикладных направлений, в частности системы *элек­тронной цифровой подписи (ЭЦП)* и *электронных денег.*

В 80-90-е годы 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 80-90-х годах были разработаны нефейстеловские (симметричный блочный криптоалгоритм на основе подстановочно-перестановочной сети) шифры (SAFER, RC6 и др.), а в 2000 году по­сле открытого международного конкурса был принят новый национальный стандарт шифрования США - AES. [3]

***Занятие №2.*** *Программирование простейших алгоритмов*

***Цели занятия:***

* создание программы для шифрования посланий с использованием алгоритма шифра Цезаря;
* создание программы для шифрования посланий с использованием алгоритма квадрата Полибия;

***Задачи занятия:***

* разобрать принцип шифра Цезаря;
* составить алгоритм работы программы «Шифр Цезаря»;
* создать программу на основе алгоритма;
* разобрать принцип работы квадрата Полибия;
* составить алгоритм и на его основании разработать программу шифрования на основе квадрата Полибия.

***Учебные вопросы, рассматриваемые на занятии:***

1. Шифр Цезаря.

2. Квадрат Полибия.

***Материал для проведения занятия***

*Задания и упражнения для выполнения:*

1. Разработать программу в PascalABC:

а. для шифрования посланий посредством шифра Цезаря;

б. разработать программу в PascalABC для шифровки-дешифровки шифра Цезаря.

Пример текста и шифротекста с кодом 3:

VENI VIDI VICI → YHQL YLGL YLFL

2. Разработать программу в PascalABC для шифрования текста с использованием русскоязычного квадрата Полибия см. таблицу 2.4.

Примеры универсальной программы (шифр Цезаря), реализованной различными способами показаны в Приложении 1.

На данном занятии можно программировать задачи с фиксированным значением ключа (в оригинале шифра Цезаря он равен 3). В пятом занятии проводим шифровании с любым ключом.

В связи с тем, что при работе на клавиатуре каждому вводимому символу согласно таблицы ASCII ставиться уникальный десятичный код, а буквы всех алфавитов (русского и английского) идут последовательно друг за другом, то это можно использовать в программировании. Текст данной программы представлен в Приложении 1.

Таблица 2.4

Квадрат Полибия для русского алфавита

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | А | Б | В | Г | Д |
| B | Е | Ж | З | И | К |
| C | Л | М | Н | О | П |
| D | Р | С | Т | У | Ф |
| E | Х | Ц | Ч | Ш | Щ |
| F | Ы | Ь | Э | Ю | Я |

Пример программы, реализующей шифрование с использованием квадрата Полибия показан в Приложении 1. На выходе мы получаем пары латинских букв, которые показывают место исходной буквы в нашем квадрате Полибия.

Дополнительные задания:

1. Усовершенствовать программу с учётом строчных и прописных букв.

2. Модернизировать программный код с учётом возможных пробелов в тексте.

***Занятие №3.*** *Классификация алгоритмов шифрования. Характеристики алгоритмов*

***Цели занятия:***

* выучить общую классификацию алгоритмов шифрования;
* познакомить учащихся с характеристиками криптосистем;

***Задачи занятия:***

* довести до учеников классификацию алгоритмов шифрования по разным основаниям (признакам);
* познакомить учеников с основными и наиболее важными характеристиками и свойствами алгоритмов шифрования;
* ознакомить с элементами анализа криптосистем.

***Учебные вопросы, рассматриваемые на занятии:***

1. Классификация алгоритмов шифрования.

2. Характеристики алгоритмов шифрования.

3. Анализ характеристик алгоритмов.

***Материал для проведения занятия***

*1. Классификация алгоритмов шифрования*

Существует несколько классификаций. Рассмотрим некоторые из них.

**1. По области применения** шифров различают криптосистемы ограниченного и общего использования.

Стойкость **криптосистемы ограниченного использования** основывается на сохранении в секрете алгоритма криптографического преобразования в силу его уязвимости, малого количества ключей или отсутствия таковых (секретные кодовые системы).

Стойкость **криптосистемы общего использования** основывается на секретности ключа и сложности его подбора потенциальным противником.

**2. По особенностям алгоритма шифрования** криптосистемы общего использования можно разделить на следующие виды см. **Ошибка! Источник ссылки не найден.**.

В **одноключевых** системах для шифрования и дешифрования используется один и тот же ключ.

В шифрах **перестановки** все буквы открытого текста остаются в зашифрованном сообщении, но меняют свои позиции. В шифрах **замены** наоборот, позиции букв в шифровке остаются теми же, что и у открытого текста, но символы открытого текста заменяются символами другого алфавита.

В **аддитивных** шифрах буквы алфавита заменяются числами, к которым затем добавляются числа секретной случайной (псевдослучайной) числовой последовательности (гаммы). Состав гаммы меняется в зависимости от используемого ключа. Обычно для шифрования используется логическая операция «Исключающее ИЛИ» (XOR). При дешифровании та же гамма накладывается на зашифрованные данные. Гаммирование широко используется в военных криптографических системах.

В **двухключевых** системах для шифрования и дешифрования используется два совершено разных ключа. При использовании **детерминированного** алгоритма шифрование и расшифрование посредством соответствующей пары ключей возможно только единственным способом. **Вероятностный** алгоритм при шифровании одного и того же исходного сообщения с одним и тем же ключом может давать разные шифротексты, которые при расшифровке дают один и тот же результат.

**Квантовая криптография** вносит в процесс шифрования естественную неопределённость квантового мира. Процесс отправки и приёма информации выполняется посредством объектов квантовой механики, например, при помощи электронов в электрическом токе, или фотонов в линиях волоконно-оптической связи. Самым ценным свойством этого вида шифрования является то, что при посылке сообщения отправляющая и принимающая сторона с достаточно большой вероятностью (99.99…%) могут установить факт перехвата зашифрованного сообщения.

**Комбинированные (составные)**методы предполагают использование для шифрования сообщения сразу нескольких методов (например, сначала замена символов, а затем их перестановка). Наиболее распространённый – это объединение симметричного и ассиметричного алгоритма: данные шифруются при помощи симметричного алгоритма (он более быстрый), но секретное слово, передаваемое по закрытому каналу связи, шифруется посредством алгоритма с открытым ключом.

Все шифры по алгоритму преобразования также делят на потоковые и блочные. В **потоковых** шифрах преобразование выполняется отдельно над каждым символом исходного сообщения. Для **блочных** шифров информация разбивается на блоки фиксированной длины, каждый из которых шифруется и расшифровывается отдельно.

Кроме приведённой на **Ошибка! Источник ссылки не найден.** классификации, шифры в зависимости от особенностей алгоритма делят также на потоковые и блочные. В **потоковых шифрах** одна и та же процедура преобразования выполняется над каждым символом независимо от других. В **блочных шифрах** исходное сообщение разбивается на блоки фиксированной длины, каждый из которых шифруется и расшифровывается отдельно.

*2. Характеристики криптоалгоритмов*

Одним из ключевых понятий в криптографии является **стойкость шифра (криптостойкость)** - способность шифра противостоять всевозможным атакам на него. Под **атакой на шифр** понимают попытку вскрытия этого шифра [4].

Среди наиболее важных **показателей криптостойкости**: количество всех возможных ключей шифра и среднее время, необходимое для криптоанализа (вскрытия).

Понятие стойкости шифра является центральным для криптографии. Хотя качественно понять его довольно легко, но получение строгих доказуемых оценок стойкости для каждого конкретного шифра – проблема нерешённая. Поэтому стойкость конкретного шифра оценивается только путём всевозможных попыток его вскрытия и зависит от квалификации криптоаналитиков, атакующих шифр. Такую процедуру иногда называют **проверкой стойкости** [4].

В зависимости от стойкости шифры делятся на три группы:

- совершенные шифры – шифры, заведомо неподдающиеся вскрытию (при правильном использовании);

- шифры, допускающие неоднозначное вскрытие (при попытке вскрытия противником конкретной шифрограммы он может получить несколько правдоподобных вариантов исходного сообщения);

- шифры, допускающие однозначное вскрытие (основная масса шифров) [9].

Нет невзламываемых шифров. Все системы шифрования просто делают взламывание шифровок или заведомо дороже содержащейся в сообщении информации, или затягивают расшифровывание на неприемлемо большой срок по времени. При разработке шифра устанавливают приемлемые цену или время взламывания и дальше уже не обращают внимания на очень богатых или терпеливых взломщиков. Необходимую сложность ключа в классических криптографических системах вычислить нетрудно, если знать технические возможности источника угрозы и плату за ошибку в оценке надёжности шифра. Число необходимых ключей можно записать в виде формулы:

***N=[время жизни шифра]/[скорость подбора ключей]/[шанс взлома]***

Время жизни шифра вряд ли целесообразно задавать больше 25 лет. В Британии секретнейшие правительственные решения по истечении этого срока публикуют для историков. А вот скорость подбора ключей всегда вызывает споры. Например, публикация американского стандарта шифрования DES вызвала у ряда криптографов призывы его бойкотировать, так как они утверждали, что за 20 миллионов долларов можно создать ЭВМ, перебирающую миллион ключей в секунду и раскалывающую шифр скорее чем за сутки. Правда, представители национального бюро стандартов возражали, что на создание такой сверхбыстродействующей ЭВМ уйдёт 5 лет, а это время - приемлемый срок жизни стандарта и шифров. Последняя величина в формуле, выражающая риск от взлома шифра за указанный срок, весьма индивидуальна и обычно имеет величину от одной тысячной до одной миллионной в зависимости от области его применения.

Криптографы считают, что программа шифрования общедоступна и секрета не представляет, значит, сложность вскрытия шифра зависит лишь от количества секретных ключей. У такого определения сложности раскалывания шифра есть уязвимое место. Согласно известной теореме Гёделя проблема криптографической атаки на шифр не имеет универсального решения. А это означает: всегда есть надежда, что кому-нибудь придёт в голову остроумная идея, позволяющая неожиданно просто прочесть нераскрываемые до этого шифры. Следовательно, несмотря на прогресс в создании устойчивых шифров, криптографы буквально ходят по краю пропасти, зажмурив глаза, а криптоаналитики пытаются их туда столкнуть. Нужно сказать, что практика криптологии и не отрицает этого. Действительно, ряд теоретических открытий дал пути для чтения некоторых типов шифров, раньше казавшихся нераскрываемыми. Тем не менее, больше похожа на реальную ситуацию другая схема, считающая, что теоретики если и смогут основательно упростить подбор ключа, то все равно останется так много переборов, что их выполнение за ограниченное время на современной технике будет невозможно. Иными словами, раскрытие очень длинного ключа методом полного перебора в реальных условиях невозможно, однако не исключена возможность создания принципиально нового метода результативной атаки на шифр.

Теперь подведём итоги. Известны два основных типа шифров, комбинации которых образуют классические криптографические системы. Главная идея, положенная в основу их конструирования, состоит в комбинации функций, преобразующих исходные сообщения в текст шифровки, то есть превращающих эти исходные сообщения с помощью секретных ключей в нечитаемый вид. Но непосредственное применение функций сразу ко всему сообщению реализуется очень редко. Все практически применяемые криптографические методы связаны с разбиением сообщения на большое число частей фиксированного размера, каждая из которых шифруется отдельно, если не независимо. Это существенно упрощает задачу шифрования, так как сообщения обычно имеют различную длину. Поэтому можно выделить четыре основных метода шифрования: потоковый, блочный, с открытым ключом и свёрточный. Их различия заключаются в следующем:

* При потоковом шифровании каждый знак текста шифровки является функцией значения и положения соответствующего знака открытого текста. Знаками бывают биты, байты и редко единицы текста покрупнее. Потоковое шифрование представляет собой шифровку замены знаков.
* При блочном шифровании исходный текст сначала разбивается на равные по длине блоки бит. К блокам применяется зависящая от ключа функция шифрования для преобразования их в блоки шифровки такой же длины. Обычно блоки шифруются взбиванием.
* Основное отличие систем с открытым ключом состоит в том, что в ней знание ключа шифрования недостаточно для расшифровывания и наоборот. Системы с открытым ключом всегда блочные.
* Если в потоковом и блочном шифре функция шифрования зависит только от ключа, то в шифрах со свёрткой она зависит как от ключа, так и от одного или более предшествующих символов или блоков текста шифровки.

Чтобы классифицировать (см. Таблицу 2.5) эти методы шифрования данных необходимо определить некоторое число характерных признаков, которые можно использовать для установления различия между шифрами. Принимается также, что каждый блок или знак сообщения шифруется отдельно в определённом порядке. При этом условии выделяются четыре характерных признака классификации криптографических систем.

1. Нужно определить, над какими объектами исходного текста производятся криптографические преобразования: над отдельными битами, байтами или блоками. Например, в шифре Энигмы операции производятся побайтно, в то время как DES оперирует одновременно со множеством бит, называемым блоком. Свойство системы использовать блоки бит назовём блочностью.

2. Функции шифрования могут зависеть от знаков сообщения. В системах, обладающих свойством зависимости функции шифрования от знаков сообщения, будет наблюдаться размножение ошибок: если при передаче исказится хоть один бит шифровки, то после расшифровывания текст будет содержать некоторое количество ошибок. Зависимость функции шифрования от знаков сообщения назовём свёрточностью.

Таблица 2.5

Свойства систем шифрования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Типы систем | Блочность | Свёрточность | Транзитивность | Симметричность |
| Потоковые | - | - | - | + |
| Блочные | + | - | + | - |
| С открытым ключом | + | - | + | + |
| Свёрточные | + | + | + | + |

3. Шифрование отдельных знаков сообщения может зависеть от их положения в тексте. В приведённом только что примере свёрточного шифрования, различные знаки сообщения шифруются с учётом их положения. Поэтому потеря при передаче любой части текста шифровки приведёт к неправильному расшифрованию всех последующих частей сообщения. Независимость знаков шифра от их положения в исходном тексте назовём транзитивностью.

4. Системы шифрования могут быть как симметричные, с одним ключом, так и асимметричными, с разными ключами для шифрования и расшифровывания. Основное различие между ними состоит в том, что в асимметричной системе знание одного ключа недостаточно для раскрытия соответствующего ему второго ключа.

*3. Анализ характеристик алгоритмов*

Для реализации потокового шифрования необходимо сконструировать генератор ключевой последовательности, выход которого используется для познакового шифрования открытого текста. К преимуществам потоковых шифров относятся отсутствие размножения ошибок, простая реализация и высокая скорость шифрования. Основным же недостатком является необходимость передачи информации синхронизации перед заголовком сообщения, которая должна быть принята до расшифровывания любого сообщения. Это связано с тем, что если два различных сообщения шифруются на одном и том же ключе, то для расшифровывания этих сообщений должна использоваться одна и та же ключевая последовательность. Это может создать угрозу криптографической стойкости системы, и поэтому всегда необходимо применять дополнительный, случайно выбираемый ключ, который передается в начале сообщения и используется для модификации основного ключа шифрования.

Потоковые шифры широко используются в военных и других системах, близких к ним по назначению, для шифрования данных и преобразованных в цифровую форму речевых сигналов. До недавнего времени эти применения преобладали для данного метода шифрования. Это объясняется, в частности, относительной простотой конструирования и реализации генераторов хороших шифрующих последовательностей. Но главным фактором, конечно, было отсутствие размножения ошибок в потоковом шифре. Так как для передачи данных и речевых сообщений в стратегической связи используются каналы недостаточно высокого качества, то любая криптографическая система, увеличивающая и без того нередкие ошибки, неприменима. Кроме указанных приложений потоковые шифры используются и в системах шифрования каналов передачи данных, где их непрерывное действие препятствует эффективному криптоанализу. Единственным стандартным методом потокового шифрования является вариант, употреблённый в стандарте шифрования данных DES.

Главное свойство блочного шифрования состоит в том, что каждый бит блока текста шифровки является функцией почти всех бит соответствующего блока открытого текста, и никакие два блока открытого текста не могут быть представлены одним и тем же блоком текста шифровки. Основное преимущество простого блочного шифрования состоит в том, что в хорошо сконструированной системе небольшие изменения открытого текста или ключа вызывают большие и непредсказуемые изменения в тексте шифра. Однако употребление простейших блочных шифров связано с серьёзными недостатками. Первый из них состоит в том, что если ко всем блокам применить один и тот же ключ, то даже при сравнительно большой длине блока, как в DES, возможен криптоанализ «со словарём». Ясно, что блок небольшого размера, как в стандарте DES, может даже повториться в сообщении вследствие большой избыточности открытого текста. Это может привести к тому, что идентичные блоки открытого текста в сообщении будут представлены идентичными блоками текста шифровки, что даёт криптоаналитику шанс в угадывании текста и атаке на ключ. Другой потенциальный недостаток этого шифра связан с размножением ошибок внутри блока. Результатом изменения одного бита в принятом блоке шифровки будет неправильное расшифровывание всего блока. Вследствие отмеченных недостатков простейшие блочные шифры не употребляются для шифрования длинных сообщений. Но в финансовых учреждениях, где сообщения часто состоят всего из одного или двух блоков, блочные шифры типа DES широко применяются в простейшем варианте. Поскольку использование этого варианта связано с необходимостью частой смены ключа шифрования, то вероятность шифрования двух идентичных блоков открытого текста на одном и том же ключе очень мала.

Возможно образование смешанных систем потокового и блочного шифрования c использованием лучших свойств каждого из этих шифров. В таких системах потоковое шифрование комбинируется с перестановками бит в блоках. Открытый текст сначала шифруется, как при обычном потоковом шифровании, затем полученный текст шифровки разбивается на блоки фиксированного размера и в каждом блоке дополнительно производится перестановка под управлением ключа. В результате получается шифр, не размножающий ошибки, но обладающий дополнительным свойством, которого нет у примитивного потокового шифра замены на основе операции XOR, состоящим в том, что перехватчик не знает, какому биту открытого текста соответствует бит текста шифровки. Благодаря этому зашифрованное сообщение становится гораздо более сложным для раскрытия.

Применение криптографических систем блочного шифрования в сочетании со свёрткой даёт ряд важных преимуществ. Первое и самое значительное, это возможность использования их для обнаружения манипуляций с сообщениями, производимых активными перехватчиками. При этом используется факт размножения ошибок, а также способность этих систем генерировать код аутентификации сообщений. Другое преимущество состоит в том, что такие шифры, в отличие от блочных, могут не требовать начальной синхронизации. Это значит, что если какая-то часть сообщения была пропущена при приёме, то оставшаяся часть его все равно может быть расшифрована. Таким системам свойственны и определённые недостатки. Основной из них - это размножение ошибок. Один ошибочный бит при передаче вызовет блок ошибок в расшифрованном тексте. При этом требование увеличения зависимости одного знака текста шифровки от как можно более длинного его предыдущего участка, повышает криптографическую стойкость, но противоречит требованию надёжности, связанному с размножением ошибок. Другой недостаток состоит в том, что разработка и реализация систем шифрования с обратной связью оказываются более трудными, чем систем потокового шифрования. Впрочем, размножение ошибок может быть и положительным свойством, если важно, чтобы данные принимались точно и даже одна ошибка в них недопустима. Так как сплошной поток ошибок может быть обнаружен несравненно легче, чем одиночный сбой, то в данном случае размножение ошибок уже не является недостатком шифра.

Первый вывод, который можно сделать из проведённого анализа, состоит в том, что в большинстве практических криптографических систем применяются алгоритмы или потокового шифрования, или шифрования со свёрткой. Большинство криптографических систем потокового шифрования использует оригинальные алгоритмы для коммерческого сектора или секретные правительственные алгоритмы. Такое положение, очевидно, сохранится ещё в ближайшие годы. Возможно также, что большинство систем шифрования со свёрткой будет основано на применении алгоритмов блочного шифрования в специальном варианте, в частности, наиболее известного алгоритма блочного шифрования DES. О других методах шифрования можно сказать, что несмотря на быстрый рост публикаций по криптографическим системам с открытым ключом, они ещё не полностью выдержали испытание временем, хотя обойтись без них сейчас уже невозможно. Поэтому для потенциальных пользователей криптографии приходится выбирать между системами потокового шифрования и системами шифрования со свёрткой, основанными на применении алгоритмов блочного шифрования типа DES. Однако существуют определённые области применения, например, финансовые операции, где вполне возможно использование методов простейшего блочного шифрования. Выбор криптографического алгоритма в значительной мере зависит от его назначения. Некоторые данные, которыми можно руководствоваться при выборе типа криптографических систем, приведены в таблице 2.6.

Таблица 2.6

Характеристики алгоритмов шифрования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Типы систем | Размножение ошибок | Аутентификация | Алгоритм | Скорость |
| Потоковые | нет | нет | нет типовых | большая |
| Блочные | есть | нет | DES и аналоги | большая |
| С открытым ключом | есть | есть | RSA, ЭльГамаля | очень низкая |
| Сверточные | есть | есть | делаются на основе блочных | большая |

К характеристикам, влияющим на выбор типа криптографической системы, в первую очередь относится требуемая скорость шифрования. Например, при очень низкой скорости вполне возможно применение криптографической системы RSA. Системы блочного шифрования в программном варианте тоже относятся к низкоскоростным, но в аппаратном варианте, например, по алгоритму DES, могут работать почти с любыми скоростями. Если требуются очень высокие скорости, то лучшей может быть система потокового шифрования с высоким быстродействием как в программном, так и в аппаратном вариантах. Если же требуется аутентификация, то должны применяться системы со свёрткой текста шифровки или открытым ключом. И, наконец, если имеющийся канал связи относится к каналам, подверженным ошибкам, которые не должны размножаться криптографической системой, то выбору системы потокового шифрования нет альтернативы [8].

*Нераскрываемые шифры*

Почти все используемые на практике шифры характеризуются как условно надёжные, поскольку они могут быть раскрыты в принципе при наличии неограниченных вычислительных возможностей. Абсолютно надёжные шифры нельзя разрушить даже при наличии неограниченных вы­числительных возможностей. Доказательство существования и единствен­ности абсолютно надёжного шифра получил К. Шеннон с помощью разра­ботанного им теоретико-информационного метода исследования шифров. Таким образом, единственный абсолютно надёжный шифр, который ис­пользуется на практике, это так называемый *одноразовый блокнот,* в ос­нове которого лежит та же идея, что и шифре Цезаря. Рассмотрим его основную идею.

В русском варианте число символов *расширенного алфавита,* т.е. совокупности букв, а также знаков препинания и пробела между словами, равно 44. Занумеровав все символы расширенного алфавита числами от 0 до 43, можно рассматривать любой передаваемый текст, как последова­тельность **{аn}** чисел множества **А={0,1,2,...,43}.** Имея случайную после­довательность **{сn}** из чисел множества **А** той же длины что и передавае­мый текст (ключ), складываем по модулю 44 число ***ап*** передаваемого текс­та с соответствующим числом ***сп*** ключа

**an+cn≡bn(mod 44), 0 ≤ bn** *≤* **43**,

получим последовательность **{bn}** знаков шифрованного текста.

Чтобы получить передаваемый текст, можно воспользоваться тем же ключом:

**ап≡bп-с(mod *44), 0 ≤*****ап** ≤ ***43***.

У двух абонентов, находящихся в секретной переписке, имеются два одинаковых блокнота, составленных из отрывных страниц, на каждой из которых напечатана таблица со случайными числами или буквами, т.е. случайная последовательность чисел из множества **А.** Таблица имеет только две копии: одна используется отправителем, другая - получателем. Отправитель свой текст шифрует указанным выше способом при помощи первой страницы блокнота. Зашифровав сообщение, он уничтожает ис­пользованную страницу и отправляет его второму абоненту. Получатель шифрованного текста расшифровывает его и также уничтожает использо­ванный лист блокнота. Нетрудно видеть, что одноразовый шифр нераск­рываем в принципе, так как символ в тексте может быть заменён любым другим символом и этот выбор совершенно случаен [5].

***Занятие №4.*** *Симметричные криптосистемы*

***Цели занятия:***

* изучить принципы симметричного шифрования.

***Задачи занятия:***

* понять принцип симметричного шифрования;
* попробовать передачу текстов посредством симметричного шифрования.

***Учебные вопросы, рассматриваемые на занятии:***

1. Схема и принципы симметричного шифрования.

2. Пример шифра с симметричным шифрованием.

***Материал для проведения занятия***

*1. Схемы и принципы симметричного шифрования*

В общедоступной литературе математические задачи криптографии на современном уровне впервые были рассмотрены К. Шенноном в работе [16]. В этой работе К. Шеннон с помощью предложенного им теоретико-информационного подхода решил некоторые из важнейших проблем теоретико-информационной криптографии. В частности, им показано, что абсолютной надёжностью могут обладать только те шифры, у которых объем ключа не меньше объема шифруемой информации, а также приведены примеры таких шифров. Там же были предложены и принципы построения криптографически надёжных преобразований с помощью композиции некоторых разнородных отображений и т. п.

В указанной работе Шеннона были сформулированы и доказаны математическими средствами необходимые и достаточные условия недешифруемости системы шифра. Было установлено, что единственным недешифруемым шифром является, так называемая, лента одноразового использования *(One-time Pad),* когда открытый текст шифруется с помощью случайного ключа такой же длины. Это обстоятельство делает абсолютно стойкий шифр очень дорогим в эксплуатации.

Прежде всего, Шеннон сделал вывод, что во всех, даже очень сложных шифрах, в качестве типичных компонентов можно выделить шифры замены и перестановки.

Абсолютно стойкие шифры применяются в сетях связи с небольшим объёмом передаваемой информации, которые используют, как правило, для передачи особо важной государственной информации.

Это объясняется тем, что каждый передаваемый текст должен иметь свой собственный, единственный и неповторимый ключ. Следовательно, перед использованием этого шифра все абоненты должны быть обеспечены достаточным запасом случайных ключей и должна быть исключена возможность их повторного применения. Выполнение этих требований необычайно трудно и дорого.

При реализации с помощью вычислительной технике, как говорилось ранее, алгоритмы шифрования получили ещё одну классификацию - их подразделяют на блочные и поточные. Для алгоритмов, используемых в компьютерных модемах, типичный размер блока составляет 64 бита – достаточно большое значения, чтобы помешать анализу, и достаточно небольшое и удобное для работы.

Блочный шифр ***Ak*** представляет собой автомат, входами и выходами которого являются последовательности ***X*** и ***ψ = Ak (X)*** длины ***n.*** Входная последовательность ***X*** разбивается на блоки длины ***n*** и каждый блок шифруется независимо один от другого одним ключом ***K*.**

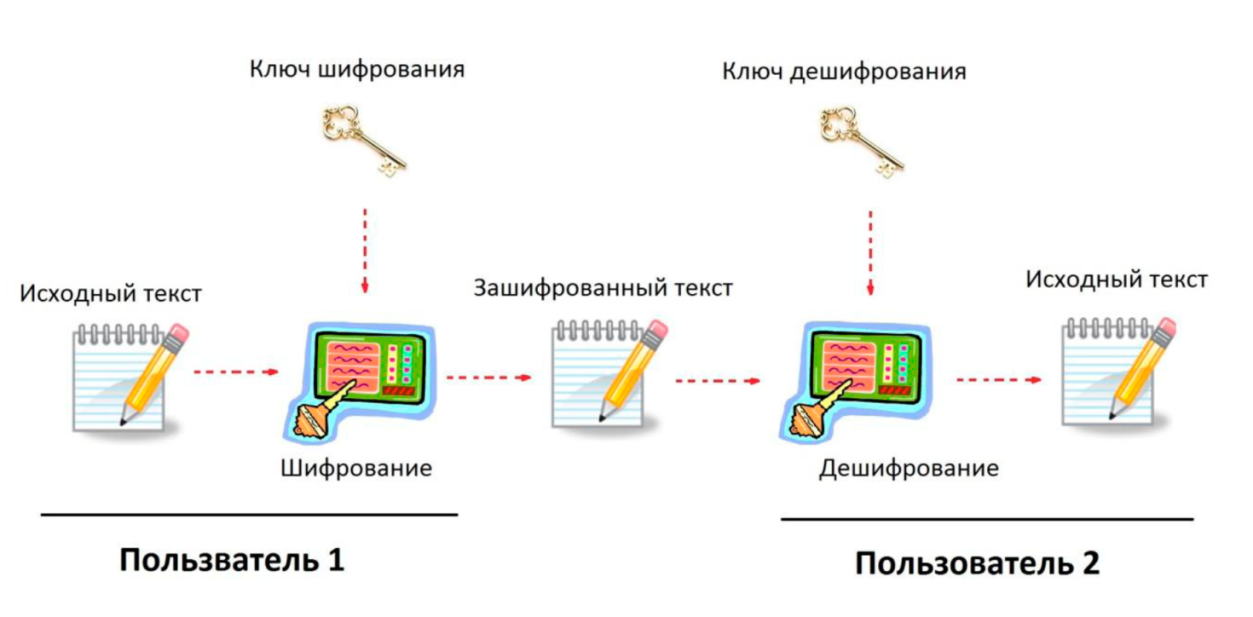


Рис. 2. Общая схема шифрования данных

Поточный шифр представляет собой автономный автомат, который вырабатывает псевдослучайную двоичную последовательность ***γ*** = (***γ0***, … ***γn …***)*.* В качестве шифрованной информации выступает последовательность ***ξ = ξ0****, …****ξn****…).* Обычно в качестве функции наложения ***ϕ***используется функция сложения в каком-либо конечном поле или кольце, в частности, в двоичном случае ***ξ = β ⊕ γ***. В последнем случае обратное преобразование (дешифрование) осуществляется по формуле ***β = ξ ⊕ γ***, что позволяет на обоих концах канала связи иметь шифраторы с одинаковыми ключами [6].

Поточные шифры, как правило, более производительны, чем блочные и используются для шифрования речи, сетевого трафика и иных данных с заранее неизвестной длиной. При достаточно частой смене ключа для выработки гаммы - последовательности поточные шифры обеспечивают достаточную стойкость.

*2. Пример шифра с секретным ключом*

Для примера рассмотрим шифр Гая Юлия Цезаря адаптировав его к русскому алфавиту. Шифр Цезаря является классическим шифром замены.

В этом шифре каждая буква сообщения заменяется на другую, номер которой в алфавите на три больше. Например, А заменяется на Г, Б на Д и т.д. Три последние буквы русского алфавита — Э, Ю, Я — шифруются бук­вами А, Б, В соответственно. Например, слово ПЕРЕМЕНА после применения к нему шифра Цезаря превращается в ТИУИПИРГ (ес­ли исключить букву Ё и считать, что в алфавите 32 буквы).

Последующие римские цезари модифицировали шифр, исполь­зуя смещение в алфавите на четыре, пять и более букв. Мы можем описать их шифр в общем виде, если пронумеруем (закодируем) бук­вы русского алфавита числами от 0 до 31 (исключив букву Ё). Тогда

***с= (т + к)* mod 32, (2.1)**

где ***т***и***с****—* номера букв соответственно сообщения и шифротекста, а ***к*** *—* некоторое целое число, называемое ключом шифра (в рассмот­ренном выше шифре Цезаря ***к = 3***). (Здесь и в дальнейшем ***a* *mod b***обозначает остаток от деления целого числа ***а***на целое число ***b***, причем остаток берется из множества ***{0, 1 ,..., b —* *1}***. Например, ***13 mod 5 = 3***.)

Чтобы дешифровать зашифрованный текст, нужно применить «обратный» алгоритм

***т* = *(с — к*) *mod 32 (2.2)***

Можно представить себе ситуацию, когда источник и получа­тель сообщений договорились использовать шифр (2.1), но для того, чтобы усложнить задачу противника, решили иногда менять ключ шифра. Для этого Пользователь А каким-либо образом генерирует число ***А\****, передаёт его Пользователю В по закрытому каналу связи, и после этого они об­мениваются сообщениями, зашифрованными с помощью этого клю­ча *к.* Замену ключа можно проводить, например, перед каждым сеансом связи или после передачи фиксированного числа букв (ска­жем, каждую десятку символов шифровать со своим ***к****)* и т.п. В та­ком случае говорят, что ключ порождается источником ключа. Схе­ма рассмотренной криптосистемы с секретным ключом приведена на рис. 2.1.

*Задачи и упражнения для выполнения:*

1. Определить ключи шифра Цезаря для русского алфавита, если известен оригинал и его шифрованное послание:

а. АПЕЛЬСИН – САЦЬНВЩЮ

б. АБРИКОС – ЫЬЛГЕЙМ

2. Расшифровать послание, если известно, что оно зашифровано шифром Цезаря с неизвестным ключом ***k, 0 < k < 32***:

а. ФХНЗКЧ,

б. ЦЩЕБФ.

***Занятие №5.*** *Шифрование посредством алгоритма с закрытым ключом*

***Цели занятия:***

* закрепить знания по принципам шифрования с закрытым ключом.

***Задачи занятия:***

* составить программы для шифрования и дешифрования с использованием алгоритма шифра Цезаря с любым ключом;
* провести тренировку и одновременно проверку составленных программ.

***Учебные вопросы, рассматриваемые на занятии:***

1. Разработка программы, реализующей цифр Цезаря с любым ключом, используя остаток от деления.

2. Проверка работы системы передачи данных через шифрограммы.

***Материал для проведения занятия***

*Задания и упражнения для выполнения:*

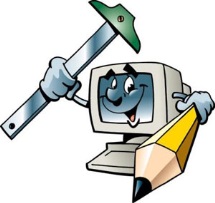
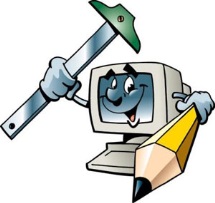
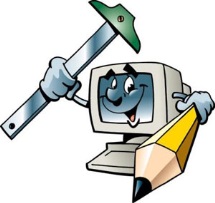
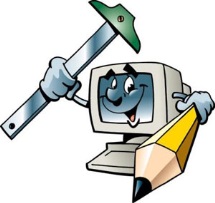
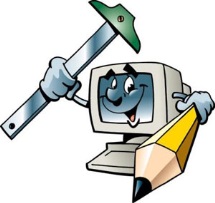
1. Разработать программу в PascalABC для шифрования и дешифрования методом Цезаря с любым ключом используя остаток от деления (mod). Пример программы показан в Приложении 2.

2. Руководитель занятия, имитируя закрытый канал связи, раздает учащимся ключи шифрования и дешифрования, таким образом, чтобы у двух соседних компьютеров ключи шифрования и дешифрования были одинаковы. Первому ученику выдаётся текст для шифрования. И, согласно схемы, указанной на рис. 2.2, сообщение передаётся по кругу через всех учеников.

По завершении круга можно сделать вывод о правильности схемы шифрования и дешифрования всего группы. Можно провести вариации задачи: все участники шифруют с разными ключами текст (шифрование в несколько проходов одним алгоритмом), а последний зная алгоритм пытается его расшифровать.

В идеале для организации и проведения занятия компьютеры должны быть завязаны в сеть, с общим доступом к разным папкам пары компьютеров, чтобы исключить участие бумаги в проведении занятий. Все программы шифрования необходимо сохранять, они понадобятся для дальнейших занятий.

**Преподователь**



Ключ 1

Ключ 2

Ключ 3

Ключ 4

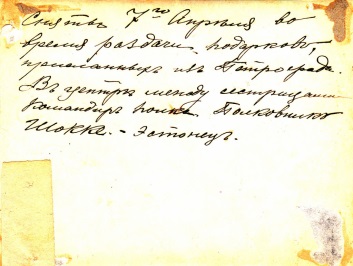
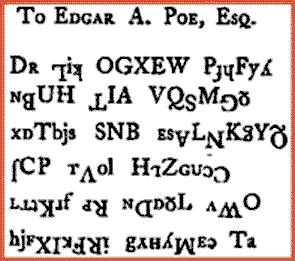


Рис. 2. Схема передачи шифрограммы

***Занятие №6.*** *Асимметричные криптосистемы*

***Цели занятия:***

* понять принцип асимметричных криптосистем.

***Задачи занятия:***

* рассмотреть принцип асимметричных криптосистем;
* на примере разобрать порядок работы алгоритма с открытым ключом.

***Учебные вопросы, рассматриваемые на занятии:***

1. Теория криптосистем с открытым ключом.

2. Асимметричная криптосистема в числах.

***Материал для проведения занятия***

*1. Теория криптосистем с открытым ключом.*

Пусть абоненты ***А***и ***В***условились организовать секретную пере­писку между собой с открытым ключом. Тогда каждый из них, независи­мо от другого, выбирает два достаточно больших простых числа (***p1, p2*** и ***q1, q2***), находит их произведение, функцию Эйлера от этого произведения и выбирает случайное число (***a*** и ***b***), меньшее этого вычисленного значения функции Эйлера и взаимно простое с ним. Итак:

***A: p1, p2, rA = p1p2, ϕ (rA), (a, ϕ (rA)) = 1, 0 < a < ϕ (rA),***

***B: q1, q2, rB = q1q2, ϕ (rB), (b, ϕ (rB)) = 1, 0 < b < ϕ (rB),***

Затем печатается телефонная книга, доступная всем желающим, которая имеет вид, представленный в таблице 2.7:

Таблица 2.7

Телефонная книга

|  |  |  |
| --- | --- | --- |
| ***А*** | ***rA*** | ***a*** |
| ***В*** | ***rB*** | ***b*** |

***rA*** – произведение двух простых чисел, известных только ***А***,

***a***– открытый ключ, доступный любому, кто хочет передать секретное сообщение ***А,***

***rB*** – произведение двух простых чисел, извест­ных только ***B***,

***b*** – открытый ключ, доступный любому, кто хочет передать секретное сообщение ***В.***

Каждый из абонентов находит свой секретный ключ из сравнений:

***a×α ≡ 1 (mod ϕ (rA)), 0 < α < ϕ (rA), b×β ≡ 1 (mod ϕ (rB)), 0 < β < ϕ (rB),***

Получаем таблицу 2.8

Таблица 2.8

Телефонная книга с закрытыми ключами

|  |  |  |
| --- | --- | --- |
| Абонент | Открытые ключи | Закрытые ключи |
| ***А*** | ***rA, a*** | ***α*** |
| ***В*** | ***rB, b*** | ***β*** |

Пусть абонент ***А***решает послать сообщение ***m***абоненту **В**.

***А: m → В***и пусть ***0 < m < rB***, иначе текст делят на куски длины ***rB***.

Сначала ***А***зашифровывает сообщение открытым ключом абонента ***В***, который есть в телефонной книге, и находит.

***m1 ≡ mb (mod rB), 0 < m1 < rB***, и отправляет абоненту ***В****.* Абонент ***В****,* расшифровывает это сообщение своим секретным ключом.

***m2 ≡ m1β (mod rB), 0 < m2 < rB,***

и получает ***m2 = m.***

В самом деле.

***m2 ≡ m1β ≡ (mβ)b ≡ mbβ (mod rB).***

Но ***b×β ≡ 1 (mod ϕ(rB))***, следовательно ***m2 ≡ m (mod rB)***. Но так как ***0 < m < rB, 0 < m2 < rB*** то ***m2 = m***.

*2. Асимметричная криптосистема в числах*

*Пример*. Пусть абоненты ***A*** и ***B*** решили установить между собой скрытую связь с открытым ключом.

Абонент ***A*** выбрал простые числа ***р1=7643*** и ***р2=8753***, их произведе­ние ***rа = 66899179***, функцию Эйлера ***ϕ(rA) = р1р2(1-1/р1)(1-1/р2) = 66882784***. Затем он выбирает случайное число ***a = 9467*** (открытый ключ) и находит секретный ключ из решения сравнения. ***a×α ≡ 1 (mod ϕ(rA)) = 9467×α ≡ 1 (mod 66882784)***, ***0 < α < ϕ(rA)***, т.е. ***α = 30993427***.

Абонент *B* выбрал простые числа ***q1 = 7481*** и ***q2 = 9539***, их произведе­ние ***rB = 71361259***, функцию Эйлера ***ϕ(rB)*** = ***rB (1-1/q1)(1-1/q2)=71344240***. Затем он выбирает случайное число ***b = 74671*** (открытый ключ) и находит секретный ключ из решения сравнения. ***b×β ≡ 1 (mod ϕ(rB)) = 74671×β ≡ 1 (mod 71344240)***, ***0 < β < ϕ(rB))***, т.е. ***β = 33289711***.

Общие данные можно занести в таблицу 2.9:

Таблица 2.9

|  |  |  |
| --- | --- | --- |
| Абонент | Открытые ключи | Закрытые ключи |
| ***А*** | ***66899179, 9467*** | ***30993427*** |
| ***В*** | ***71361259, 74671*** | ***33289711*** |

Абонент ***A*** решает послать сверхсекретное сообщение абоненту ***B*** ***m = 95637***. Тогда он шифрует сообщение открытым ключом абонента ***B***. ***m1 = mb(mod rB) = 9563774671 (mod 71361259) = 25963634***.

Абонент ***B***, получив это сообщение, расшифровывает его своим секретным ключом.

***m2 = m1β (mod rB) = 2596363433289711 (mod 71361259) = 95637***.

***Занятие №7.*** *Шифрование посредством алгоритма с открытым ключом*

***Цели занятия:***

* разработка программы шифрования посредством алгоритма с открытым ключом;

***Задачи занятия:***

* закрепить знания принципа работы систем с открытым ключом;
* создание программ, помогающих решить задачу шифрования с открытым ключом;

***Учебные вопросы, рассматриваемые на занятии:***

1. Решение вспомогательных задач.

2. Разработка программы шифрования.

***Материал для проведения занятия***

*1. Решение вспомогательных задач.*

Для написания программы шифрования с открытым ключом, согласно алгоритма, указанного в занятии №6, разобьём задачу на подзадачи:

1. Определение простых чисел:

**Functiom Flag(n: longint): Boolean;**

**var**

i, k: longint;

**begin**

k := Round (sqrt(n)); {максимально возможный делитель}

**if** n = 2 **then** flag := true

**else if not** odd (n) **then** flag := false

**else begin**

flag := true;

**for** i := 2 **to** k **do**

**if** n **mod** i = 0 **then** flag := false

**end**;

**if** flag **then** writeln('Простое число') **else** writeln('Составное число');

readln;

**end**.

2. Определение функции Эйлера.

**function** f(n : **longint**) : **longint**;

**var**

        res, en, i : **longint**;

**begin**

        res := n; n := round(sqrt(n)) + 1;

**for** i := 2 **to** en **do**

**if** (n **mod** i = 0) **then** **begin** *//если i - делитель*

**while** (n **mod** i = 0) **do** *//пока делится, делим*

                    n := n **div** i;

                dec(res, res **div** i);

**end**;

**if** (n > 1) **then** dec(res, res **div** i);

        f := res;

**end**;

3. Определение являются ли два числа взаимно простыми

**Function** Nod(x, y: Longint): longint; {Функция вычисления НОД, чтобы числа были взаимно простыми}

**Begin**

**Repeat**

**If** x > y **Then** x:=x **Mod** y **Else** y := y **Mod** x;

**Until** (x=0) **Or** (y=0);{до тех пор, пока одно из чисел не станет равно нулю}

Nod := x+y;

**End**;

4. Нахождение остатка от деления произведения двух чисел

**Function** Ostatok (x, y, t: longint): longint; {Вычисление остатка от деления произведения x\*y на t}

**var**

i, k: longint;

**begin**

k := 1;

**for** i := 1 **to** y **do**

k := (k\*x) **mod** t;

Ostatok := k;

**end**;

*2. Разработка программы шифрования.*

Полностью листинг программы можно посмотреть в Приложении 3.

***Занятие №8.*** *Комбинированные алгоритмы*

***Цели занятия:***

* познакомить учащихся с различными способами комбинирования алгоритмов шифрования.

***Задачи занятия:***

* рассмотреть способы комбинирования криптосистем с использование плюсов каждой из них.

***Учебные вопросы, рассматриваемые на занятии:***

1. Комбинированные алгоритмы

***Материал для проведения занятия***

*1. Комбинированные алгоритмы*

Симметричное и асимметричное шифрование остаются актуальными и криптографически гарантированными методами защиты информации. При передаче и хранении ценной или секретной информации её защищают с помощью криптосистем смешанного типа. С помощью двухключевых асимметричных алгоритмов решается задача распределения ключей для симметричных шифров.

Алгоритмы с симметричными ключами имеют очень высокую производительность. Криптография с симметричными ключами очень стойкая, что делает практически невозможным процесс дешифрования без знания ключа. При прочих равных условиях стойкость определяется длиной ключа. Так как для шифрования и дешифрования используется один и тот же ключ, при использовании таких алгоритмов требуются очень надёжные механизмы для распределения ключей.

Асимметричное шифрование сложнее. В схеме кодирования с открытым ключом невозможно вычислить процедуру дешифрования, зная процедуру шифрования. Более точно, время работы алгоритма, вычисляющего процедуру расшифровки, настолько велико, что его нельзя выполнить на любых современных компьютерах, равно как и на любых компьютерах будущего.

Следует отметить, что алгоритм асимметричного шифрования сильно проигрывает симметричным с точки зрения времени шифрования и расшифровки данных, поэтому многие современные системы шифрования используют комбинацию асимметричной и традиционной симметричной систем шифрования. Шифрование с открытым ключом используется для передачи симметричного ключа, который служит непосредственно для шифрования информации.

Всем системам открытого шифрования присущи следующие основные недостатки: ключ должен передаваться по секретному каналу; к службе генерации ключей предъявляются повышенные требования, обусловленные тем, что для ***n*** абонентов при схеме взаимодействия «каждый с каждым» требуется ***n·(n-1)/2*** ключей, то есть зависимость числа ключей от числа абонентов является квадратичной. Основным недостатком симметричного шифрования является то, что секретный ключ должен быть известен и отправителю, и получателю.

Шифрование с несимметричным ключом является более криптостойким, но более медленным и дорогим по сравнению с шифрованием с симметричным ключом. Двухключевые шифры редко применяются для непосредственного шифрования. Они в основном используются при работе с электронными документами.

Схема работы комбинированной системы приведена на рис. 2.3.



Рис. 2. Схема шифрования с симметричным и ассиметричным алгоритмом

На первом этапе пользователь А шифрует исходный файл с помощью симметричного алгоритма. На втором этапе он получает из открытых источников публичный (открытый) ключ, принадлежащий пользователю В, и с помощью этого ключа зашифровывает симметричный ключ, который использовался на первом этапе. На третьем этапе зашифрованный файл и зашифрованный симметричный ключ передаются пользователю В посредством интернета. После того как пользователь В на четвёртом этапе получил передаваемые ему данные, он на пятом этапе расшифровывает симметричный ключ с помощью своего частного (закрытого) ключа. На последнем, шестом этапе пользователь В, применяя расшифрованный на предыдущем этапе симметричный ключ, расшифровывает посланный ему файл.

Когда кто-то получает от вас сообщение, зашифрованное вашим частным ключом, он уверен в аутентичности послания, шифрование эквивалентно поставленной подписи (электронная цифровая подписи).

***Занятие №9.*** *Электронная цифровая подпись*

***Цели занятия:***

* изучить принцип цифровой подписи;
* понять схему реализации цифровой подписи.

***Задачи занятия:***

* познакомиться с основными свойствами подписи;
* рассмотреть 2 варианта реализации ЭЦП;
* рассмотреть возможные минусы ЭЦП и способы устранения;

***Учебные вопросы, рассматриваемые на занятии:***

1. Положительные свойства цифровой подписи и реализация ЦП через посредника.

2. Реализация электронной цифровой подписи через асимметричные криптосистему.

***Материал для проведения занятия***

*1. Положительные свойства цифровой подписи и реализация ЦП через посредника*

Рукописные подписи издавна используются как доказательство авторства документа или, по крайней мере, согласия с ним. Какими же свойствами обладает подпись?

1. Достоверность. Она убеждает получателя документа в том, что подписавший сознательно подписал документ.

2. Неподдельность. Она доказывает, что именно подписавший, и никто иной, сознательно подписал документ.

3. Однократность применения. Она не может быть использована повторно. Она является частью документа, жулик не сможет перенести подпись на другой документ.

4. Неизменность. Подписанный документ нельзя изменить. После того, как документ подписан, его невозможно изменить.

5. Необратимость. От неё невозможно отречься. Подпись и документ материальны. Подписавший не сможет впоследствии утверждать, что он не подписывал документ.

В действительности, ни одно из этих утверждений не является полностью справедливым. Подписи можно подделать, документы, в некоторых случаях, могут быть изменены после подписания. Однако, мы миримся с этими проблемами из-за того, что мошенничество затруднительно и может быть обнаружено.

Хотелось бы реализовать что-нибудь подобное и на компьютерах, но есть ряд проблем. Во-первых, компьютерные файлы скопировать не просто, а очень просто. Даже если подпись человека трудно подделать (например, графическое изображение рукописной подписи), можно легко вырезать правильную подпись из одного документа и вставить в другой. Простое наличие такой подписи ничего не означает. Во-вторых, компьютерные файлы очень легко можно изменить после того, как они подписаны, не оставляя ни малейшего следа изменения.

*Подпись документа с помощью симметричных криптосистем и посредника*

Пользователь А хочет подписать цифровое сообщение и отправить его Пользователю В. Он может это сделать с помощью посредника и симметричной криптосистемы.

Посредник – третье «лицо», обладающее властью, которому доверяют. Он может связываться со всеми пользователями, желающими подписывать цифровые документы. Он выдаёт секретный ключи пользователям. Эти ключи определяются задолго до начала действия протокола и могут быть использованы многократно для многих подписей.

Порядок передачи сообщения выглядит следующим образом:

* Пользователь А шифрует своё сообщение Пользователю В своим ключом и посылает его Посреднику;
* Посредник, зная все ключи*,* расшифровывает сообщение, добавляет к расшифрованному сообщению утверждение, что он получил это сообщение от Пользователя А, и шифрует это новое сообщение ключом, который знает только Пользователь В*;*
* Посредник посылает новое сообщение Пользователю В;
* Пользователь В расшифровывает сообщение своим ключом.Он может прочитать и сообщение, и подтверждение Посредника, что сообщение отправлено именно Пользователем А.

Откуда Посредник узнает, что сообщение пришло именно от Пользователя А, а не от какого-то самозванца? Он делает этот вывод из шифрования сообщения.

Также ли это хорошо, как подпись на бумаге? Посмотрим на требуемые свойства:

1. Эта подпись достоверна. В нашем случае Посредник – залуживает доверия, и Он знает, что сообщение получено от Пользователя А. Подтверждение Посредника служит доказательством для Пользователя В.

2. Эта подпись неподдельна. Только Пользователь А (и Посредник, но ему все верят) знает ключ которым шифрует сообщение Пользователь А*,* поэтому только Пользователь А мог прислать Посреднику сообщение, зашифрованное этим ключом.Если кто-нибудь попытается выдать се­бя за Пользователя А, Посредником это сразу будет обнаружено, и он не заверит подлинность сообщения от Пользователя А.

3. Эту подпись нельзя использовать повторно. Если Пользователь В попытается взять подтверждение Посредника и присоединить его к другому сообщению, Пользователь А начнёт возмущаться. Посредник попросит Пользователя В предъявить его сообщение и шифрованное сообщение Пользователя А. Но так как Пользователь В не обладает ключом Пользователя А, то подлог сообщений будет сразу обнаружен.

4. Подписанный документ нельзя изменить. Если Пользователь попытается, получив документ, изменить его, Посредник обнаружит мошенничество сравнением исходящего сообщения от Пользователя А и нового измененного сообщения от Пользователя В.

5. От подписи невозможно отказаться. Если впоследствии Пользователь А заявит, что он никогда не посылал сообщение, подтверждение Посредника докажет обратное. (Слова и действия Посредника – всегда истина).

Эти протоколы работают, но они требуют от Посредника немалых затрат времени. Он должен целыми днями расшифровывать и шифровать сообщения, посредничая между каждой парой людей, которые хотят обмениваться подписанными документами. Он должен хранить сообщения в базе данных (хотя этого можно избежать, посылая получателю копию шифрованного сообщения отправителя). Он будет узким местом любой системы связи, даже если он - просто бесчувственная компьютерная программа.

Такого посредника, которому будут доверять все корреспонденты, тяжело найти и тяжело сохранить. Посредник должен быть непогрешим и абсолютно безопасен, иначе ему нельзя будет доверять. Такая схема теоретически может работать, но она недостаточно хороша для практического применения.

*Подпись документа с помощью криптографии с открытыми ключами*

Существуют алгоритмы с открытыми ключами, которые можно использовать для цифровых подписей. В некоторых алгоритмах - примером является RSA - для шифрования может быть использован или открытый, или закрытый ключ. Зашифруйте документ своим закрытым ключом, и вы получите надёжную цифровую подпись. В других случаях - примером является DSA - для цифровых подписей исполь­зуется отдельный алгоритм, который невозможно использовать для шифрования. Эта идея впервые была изо­бретена Диффи и Хеллманом. Основной протокол прост и представлен на **Ошибка! Источник ссылки не найден.**:

Этот протокол гораздо лучше предыдущего. Посредник становиться не нужным. Такая подпись соответствует всем требованиям:

1. Эта подпись достоверна. Когда Пользователь В расшифровывает сообщение с помощью открытого ключа Пользователя А, он знает что она подписала это сообщение.

2. Эта подпись неподдельна. Только Пользователь А знает свой закрытый ключ.

3. Эту подпись нельзя использовать повторно. Подпись является функцией документа и не может быть перенесена на другой документ.

4. Подписанный документ нельзя изменить. После любого изменения документа подпись не сможет больше подтверждаться открытым ключом Пользователя А.

5. От подписи невозможно отказаться. Пользователю В не требуется помощь Пользователя А при проверке её подписи.

*Подпись документа и метки времени*

На самом деле, при определённых условиях Пользователь В сможет смошенничать. Он может повторно использовать документ и подпись совместно. Это не имеет значения, если подписанным документом был контракт или договор, а если цифровая подпись была поставлена под чеком? Можно сохранить копию электронного чека и получать по нему деньги ежедневно!

Поэтому в цифровые подписи часто включают метки времени. Дата и время подписания документа добав­ляются к документу и подписываются вместе со всем содержанием сообщения. Банк сохраняет эту метку времени в базе данных. Теперь, если мошенник попытается получить наличные по чеку во второй раз, банк проверит метку времени по своей базе данных. Так как банк уже оплатил этот чек с той же меткой времени, то повторных выплат не будет.

***Занятие №10.*** *Официальные стандарты AES и ГОСТ*

***Цели занятия:***

* познакомить со стандартом шифрования AES, принятым в США;
* познакомить с ГОСТ 28147-89 – стандартом шифрования, принятым в РФ.

***Задачи занятия:***

* рассмотреть общие принципы шифрования алгоритма AES;
* рассмотреть схему и принцип применения алгоритма шифрования ГОСТ 28147-89;
* обосновать криптостойкость российского алгоритма.

***Учебные вопросы, рассматриваемые на занятии:***

1. Стандарт шифрования США – AES.

2. ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.

3. Криптостойкость алгоритма ГОСТ 28147-89.

***Материал для проведения занятия***

*1. Стандарт шифрования США – AES.*

Используя суперкомпьютер стоимостью 250 тыс. долл., сотрудники RSA Laboratory «взломали» утвержденный правительством США алгоритм шифрования данных (DES) менее чем за три дня. (Предыдущий рекорд по скорости взлома был установлен с помощью огромной сети, состоящей из десятков тысяч компьютеров, и составил 39 дней). На специально организованной по этому случаю пресс-конференции ученые с беспокойством говорили о том, что злоумышленники вряд ли упустят случай воспользоваться подобной уязвимостью.

Эксперимент проходил в рамках исследования DES Challenge II, проводимого RSA Laboratory под руководством общественной организации Electronic Frontier Foundation (EFF), которая занимается проблемами информационной безопасности и личной тайны в Internet.

Суперкомпьютер, построенный в RSA Laboratory для расшифровки данных, закодированных методом DES по 56-разрядному ключу, получил название EFF DES Cracker.

Как утверждали правительственные чиновники и некоторые специалисты, для взлома кода DES требуется суперкомпьютер стоимостью в несколько миллионов долларов.

«Правительству пора признать ненадежность DES и поддержать создание более мощного стандарта шифрования», - сказал президент EFF Барри Штайнхардт [1].

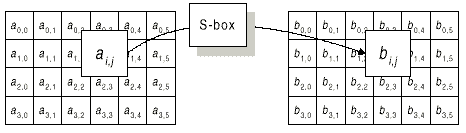
26 мая 2002 года алгоритм AES был объявлен правительством США новым стандартом шифрования в результате пятилетнего конкурса, в котором принимали участие изначально 15 алгоритмов кандидатов из различных стран мира.

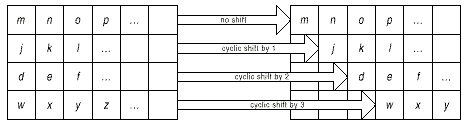
Данный алгоритм разработан двумя специалистами по криптографии из Бельгии Йона Дамена и Винсента Рэмена. Он является нетрадиционным блочным шифром, поскольку не использует сеть Фейштеля для криптопреобразований. Алгоритм представляет каждый блок кодируемых данных в виде двумерного массива байт размером 4х4, 4х6 или 4х8 в зависимости от установленной длины блока. Далее на соответствующих этапах преобразования производятся либо над независимыми столбцами, либо над независимыми строками, либо вообще над отдельными байтами в таблице.

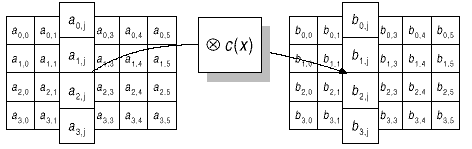
Все преобразования в шифре имеют строгое математическое обоснование. Сама структура и последовательность операций позволяют выполнять данный алгоритм эффективно как на 8-битных так и на 32-битных процессорах. В структуре алгоритма заложена возможность параллельного исполнения некоторых операций, что на многопроцессорных рабочих станциях может еще поднять скорость шифрования в 4 раза.

Алгоритм состоит из некоторого количества раундов (от 10 до 14 – это зависит от размера блока и длины ключа), в которых последовательно выполняются следующие операции:

1. ByteSub – табличная подстановка 8х8 бит

  
2. ShiftRow – сдвиг строк в двумерном массиве на различные смещения

  
3. MixColumn – математическое преобразование, перемешивающее данные внутри столбца



4. AddRoundKey – добавление материала ключа операцией XOR

  
В последнем раунде операция перемешивания столбцов отсутствует, что делает всю последовательность операций симметричной.

Алгоритм демонстрирует превосходную производительность на всех рассматриваемых в состязании платформах. Для шифра характерны быстрое разворачивание ключа и низкие требования к памяти, так что он также хорошо работает и в аппаратной реализации, и в ограниченных по памяти условиях. Простая конструкция схемы и консервативный выбор операций должны облегчить дальнейший криптоанализ шифра. Кроме того, специалистами отдельно отмечается, что избранные конструкторами операции относительно просто защитить от известных опасных атак на физическую реализацию криптоалгоритма. Ещё одна важная положительная характеристика (хотя и не рассматривавшаяся при выборе финалистов) - в шифре имеется существенный потенциал к распараллеливанию, то есть к получению выгод в производительности благодаря применению компьютерных процессоров, позволяющих одновременно выполнять множество инструкций.

*2. ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования*

**ГОСТ 28147-89** — советский и российский стандарт симметричного шифрования, введённый в 1990 году, также является стандартом СНГ.

Схема алгоритма представлена на рис. 2.4. Из рисунка видно, что схема алгоритма довольно проста, что упрощает его программную или аппаратную реализацию.

Алгоритм шифрует информацию блоками по 64 бита, которые разбиваются на два субблока (N1 и N2) по 32 бита в каждом. Субблок N1 определенным образом обрабатывается, после чего его значение складывается со значением субблокаN2 *(*сложение выполняется по модулю 2), затем суб­блоки меняются местами. Такое преобразование выполняется определенное количество раундов: 16 или 32 в зависимости от режима работы алгоритма (описаны далее). Вкаждом раунде выполняются следующие операции:

1. Наложение ключа. Содержимое субблока N1 складывается по модулю 232 с частью ключа *Кх.*

Ключ шифрования алгоритма ГОСТ 28147-89 имеет размерность 256 би­тов, а *Кх*— это его 32-битная часть, т. е. 256-битный ключ шифрования представляется в виде конкатенации 32-битных подключей (рис. 2.5):

*К1, К2, К3, К4, К5, К6, К7.*

В процессе шифрования используется один из этих подключей — в зави­симости от номера раунда и режима работы алгоритма.

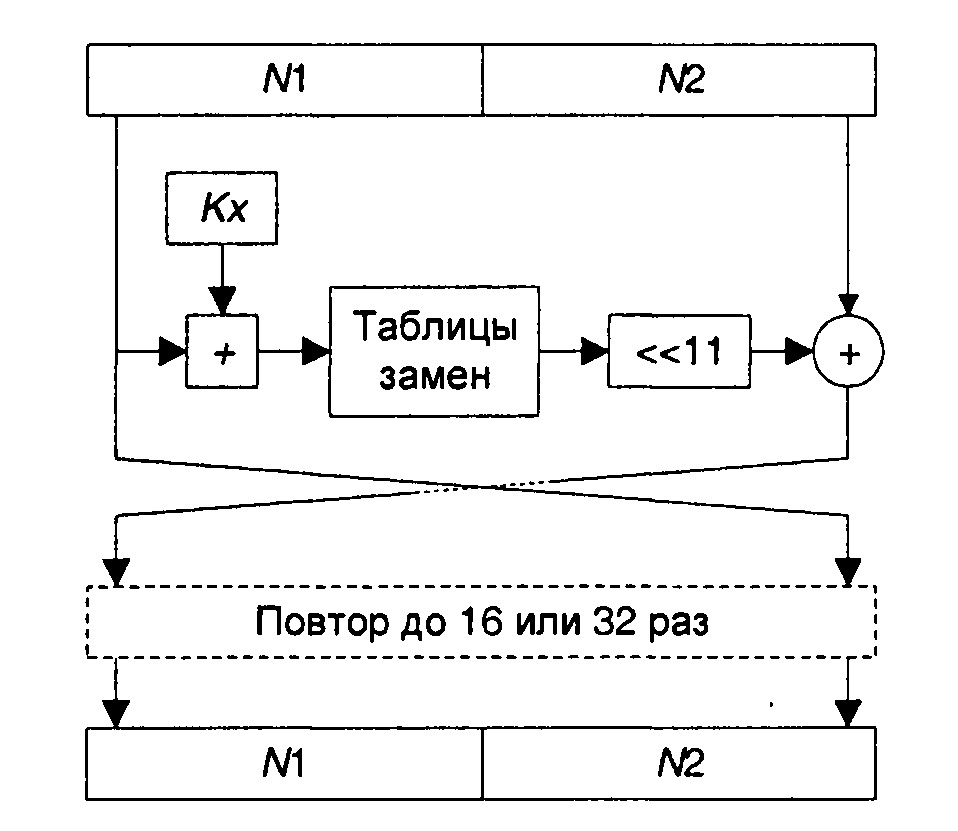


Рис. 2. Схема алгоритма ГОСТ 28147-89

2.Табличная замена. После наложения ключа субблок N1 разбивает­ся на 8 частей по 4 бита, значение каждой из которых по отдельности заменяется в соответствии с таблицей замены для данной части суб­блока. Табличные замены (Substitution box, S-box) часто используются в современных алгоритмах шифрования, поэтому стоит рассмотреть их подробнее.

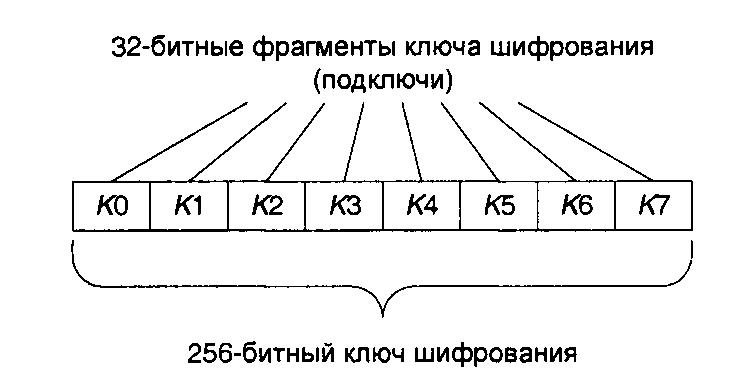


Рис. 2. Ключ шифрования алгоритма ГОСТ

Табличная замена используется таким образом: на вход подаётся блок данных определённой размерности (в этом случае — 4-битный), числовое представление которого определяет номер выходного значения. Напри­мер, имеем S-box следующего вида:

4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1.

Пусть на вход пришёл 4-битный блок «0100», т. е. значение 4. Согласно таблице, выходное значение будет равно 15, т.е. «1111» (0 заменяется на 4, 1 — на 11, значение 2 не изменяется и т. д.).

Как видно, схема алгоритма весьма проста, что означает, что наибольшая нагрузка по шифрованию данных ложится на таблицы замен. К сожале­нию, алгоритм обладает тем свойством, что существуют «слабые» табли­цы замен, при использовании которых алгоритм может быть раскрыт криптоаналитическими методами. К числу слабых относится, например, таблица, в которой выход равен входу:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

3.Побитовый циклический сдвиг влево на 11 битов.

*3. Криптостойкость алгоритма ГОСТ 28147-89*

В 1994 г. описание алгоритма ГОСТ 28147-89 было переведено на англий­ский язык и опубликовано; именно после этого стали появляться ре­зультаты его анализа, выполненного зарубежными специалистами; однако в течение значительного времени не было найдено каких-либо атак, прибли­жающихся к практически осуществимым.

Высокая стойкость алгоритма ГОСТ 28147-89 дости­гается за счет следующих факторов:

большой длины ключа — 256 битов; вместе с секретной синхропосылкой эффективная длина ключа увеличивается до 320 битов;

32 раундов преобразований; уже после 8 раундов достигается полный эф­фект рассеивания входных данных: изменение одного бита блока откры­того текста повлияет на все биты блока шифртекста, и наоборот, т. е. су­ществует многократный запас стойкости [10].

***Занятие №11.*** *Элементы криптоанализа*

***Цели занятия:***

* познакомить учащихся с элементами криптоанализа;
* изучить типы криптоаналитического вскрытия.

***Задачи занятия:***

* довести теорию криптоанализа;
* проанализировать основные типы атак на шифрованные послания;
* рассмотреть различные способы криптоаналитического вскрытия.

***Учебные вопросы, рассматриваемые на занятии:***

1. Теория криптоанализа.

2. Пример криптоанализа алгоритма Цезаря.

3. Другие способы возможного криптоанализа.

***Материал для проведения занятия***

*1. Теория криптоанализа*

Смысл криптографии - в сохранении открытого текста (или ключа, или и того, и другого) в тайне от злоумышленников (также называемых взломщиками, соперниками, врагами, перехватчиками). Предполагается, что злоумышленники полностью контролируют линии связи между отправителем и получателем.

Криптоанализ - это наука получения открытого текста, не имея ключа . Успешно проведенный криптоанализ может раскрыть открытый текст или ключ. Он также может обнаружить слабые места в криптосистемах, что в конце концов приведет к предыдущему результату. (Раскрытие ключа не криптологическими способами называется компрометацией.)

Попытка криптоанализа называется вскрытием. Основное предположение криптоанализа, впервые сфор­мулированное в девятнадцатом веке Датчманом А. Керкхофсом, состоит в том, что безопасность полностью определяется ключом. Керкхофс предполагает, что у криптоаналитика есть пол­ное описание алгоритма и его реализации. (Конечно же, у ЦРУ не в обычае сообщать Моссад о своих криптографических алгоритмах, но Моссад возможно все равно добудет их.) Хотя в реальном мире криптоаналитики не всегда обладают подробной информацией, такое предположение является хорошей рабочей гипотезой. Если противник не сможет взломать алгоритм, даже зная, как он работает, то тем более враг не сможет вскрыть алгоритм без этого знания.

С формальной точки зрения, укрепление любого звена кроме самого слабого — пустая трата времени. На практике же все обстоит далеко не так про­сто. Злоумышленник может не знать, какое звено является самым слабым, и нападет на более сильное звено. Самое слабое звено может различаться для разных типов злоумышленников. Прочность любого звена зависит от навыков злоумышленника и имеющихся у него средств. Поэтому самое слабое звено во многом определяется той или иной ситуацией. Необходимо укрепить любое звено, которое в определенной ситуации может оказаться самым слабым. [15]

Существует четыре основных типа криптоаналитического вскрытия. Для каждого из них, конечно, предпо­лагается, что криптоаналитик обладает всей полнотой знания об используемом алгоритме шифрования:

1. Вскрытие с использованием только шифротекста. У криптоаналитика есть шифротексты нескольких сообщений, зашифрованных одним и тем же алгоритмом шифрования. Задача криптоаналитика состоит в раскрытии открытого текста как можно большего числа сообщений или, что лучше, получении ключа (ключей), использованного для шифрования сообщений, для дешифрирования других сообщений, зашифрованных теми же ключами.

2. Вскрытие с использованием открытого текста. У криптоаналитика есть доступ не только к шифротекстам нескольких сообщений, но и к открытому тексту этих сообщений. Его задача состоит в полу­чении ключа (или ключей), использованного для шифрования сообщений, для дешифрирования других сообщений, зашифрованных тем же ключом (ключами).

3. Вскрытие с использованием выбранного открытого текста. У криптоаналитика не только есть доступ к шифротекстам и открытым текстам нескольких сообщений, но и возможность выбирать о т-крытый текст для шифрования. Это предоставляет больше вариантов, чем вскрытие с использованием открытого текста, так как криптоаналитик может выбирать шифруемые блоки открытого текста, что может дать больше информации о ключе. Его задача состоит в получении ключа (или ключей), использованного для шифрования сообщений, или алгоритма, позволяющего дешифрировать новые с о-общения, зашифрованные тем же ключом (или ключами) .

4. Адаптивное вскрытие с использованием открытого текста Это частный случай вскрытия с ис­пользованием выбранного открытого текста. Криптоаналитик не только может выбирать шифруемый текст, но также может строить свой последующий выбор на базе полученных результатов шифрования. При вскрытии с использованием выбранного открытого текста криптоаналитик мог вы­брать для шифрования только один большой блок открытого текста, при адаптивном вскрытии с и с-пользованием выбранного открытого текста он может выбрать меньший блок открытого текста, затем выбрать следующий блок, используя результаты первого выбора и так далее.

Существует по крайней мере ещё три типа криптоаналитической вскрытия.

5. Вскрытие с использованием выбранного шифротекста Криптоаналитик может выбрать различ­ные шифротексты для дешифрирования и имеет доступ к дешифрированным открытым текстам. На­пример, у криптоаналитика есть доступ к «черному ящику», который выполняет автоматическое д е-шифрирование. Его задача состоит в получении ключа.

Такой тип вскрытия обычно применим к алгоритмам с открытым ключом. Вскрытие с использование выбранного шифротекста иногда также эффективно против симметричных алгоритмов. (Иногда вскрытие с использованием выбранного открытого текста и вскрытие с использованием выбранного шифротекста вместе называют вскрытием с использованием выбранного текста.)

6. Вскрытие с использованием выбранного ключа. Такой тип вскрытия означает не то, что криптоаналитик может выбирать ключ, а что у него есть некоторая информация о связи между различными ключами.

7. Бандитский криптоанализ. Криптоаналитик угрожает, шантажирует или пытает кого-нибудь, пока не получит ключ. Взяточничество иногда называется вскрытием с покупкой ключа. Это очень мощные способы вскрытия, часто являющиеся наилучшим путем взломать алгоритм.

Вскрытия с известным открытым текстом и с использованием выбранного открытого текста встречаются чаще, чем можно подумать. Не является невозможным для криптоаналитика добыть открытый текст шифрованного сообщения или подкупить кого-нибудь, кто зашифрует выбранное сообщение. Может и не потребо­ваться никого подкупать - передав письмо послу, вы, возможно, обнаружите, что письмо будет зашифровано и отправлено в его страну для изучения. Многие сообщения имеют стандартные начало и окончание, что может быть известно криптоаналитику. Особенно уязвим шифрованный исходный код из-за частого использования ключевых слов: #define, struct, else, return. Те же проблемы и у шифрованного исполнимого кода: функции, циклические структуры и так далее. Вскрытия с известным открытым текстом (и вскрытия с выбранным шифротекстом) успешно использовались в борьбе с немцами и японцами в ходе Второй мировой войны.

И не забывайте о предположении Керкхофса: если мощь вашей новой криптосистемы опирается на то, что взломщик не знает, как работает алгоритм, вы пропали. Если вы считаете, что хранение принципа работы ал­горитма в секрете лучше защитит вашу криптосистему, чем предложение академическому сообществу проанализировать алгоритм, вы ошибаетесь. А если вы думаете, что кто-то не сможет дезассемблировать ваш исходный код и восстановить ваш алгоритм, вы наивны. В 1994 году такое произошло с алгоритмом RC4. Нашими лучшими алгоритмами являются те, которые были разработаны открыто, годами взламывались лучшими криптографами мира и все еще несокрушимы. (Агентство Национальной Безопасности хранит свои алгоритмы в секрете, но у них работают лучшие криптографы мира, а у вас - нет. Кроме того, они обсуждают свои алгоритмы друг с другом, полагаясь на способность товарища обнаружить все слабости в своей работе.)

У криптоаналитиков не всегда есть доступ к алгоритмам (например, вскрытие в ходе Второй мировой войны Соединенными Штатами японского дипломатического кода PURPLE), но часто они его получают. Если алгоритм используется в коммерческой программе безопасности, то это просто вопрос времени и денег, удастся ли дизассемблировать программу и раскрыть алгоритм. Если же алгоритм используется в военной системе связи, то это просто вопрос времени и денег купить (или украсть) аппаратуру и реконструировать алгоритм.

Те, кто стремится получить нераскрываемый шифр, считая этот шифр таковым только потому, что они сами не смогли его взломать, либо гении, либо дураки. К несчастью, последних в мире достаточно много. Остере­гайтесь людей, расхваливающих надежность своих алгоритмов, но отказывающихся их опубликовать. Доверять таким алгоритмам нельзя.

Хорошие криптографы опираются на мнение других, отделяя хорошие алгоритмы от плохих [17].

*2. Пример криптоанализа алгоритма Цезаря*

Обратимся теперь к анализу действий противника, пытающего­ся расшифровать сообщение и узнать секретный ключ, иными слова­ми, вскрыть, или взломать шифр. Каждая попытка вскрытия шифра называется атакой на шифр (или на криптосистему). В криптогра­фии принято считать, что противник может знать использованный алгоритм шифрования, характер передаваемых сообщений и пере­хваченный шифротекст, но не знает секретный ключ. Это называется «правилом Керкгоффса», в честь учёного, впервые сформу­лировавшего основные требования к шифрам. Иногда это правило кажется «перестраховкой», но такая «перестра­ховка» отнюдь не лишняя, если, скажем, передается распоряжение о переводе миллиона долларов с одного счета на другой.

В нашем примере: Пользователь С знает, что шифр был построен в соответ­ствии с (2.1), что исходное сообщение было на русском языке и что был передан шифротекст ТИУИПИРГ, но ключ С не известен.

Наиболее очевидная попытка расшифровки — последователь­ный перебор всех возможных ключей (это так называемый метод «грубой силы» (brute-force attack)). Итак, Пользователь С перебирает последова­тельно все возможные ключи *к =*1, 2, ..., подставляя их в алгоритм дешифрования и оценивая получающиеся результаты. Попробуем и мы использовать этот метод. Результаты дешифрования по (2.2) при различных ключах и шифротексте ТИУИПИРГ сведены в таблице 2.10. В большинстве случаев нам достаточно было расшифровать две-три буквы, чтобы отвергнуть соответствующий ключ (из-за отсутствия слова в русском языке, начинающегося с такого фрагмента).

Таблица 2.10

Расшифровка слова ТИУИПИРГ путем перебора ключей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *к* | *т* | *к* | *т* | *к* | *т* | *к* | *т* |
| 1 | СЗТ | 9 | ЙЯ | 17 | БЧ | 25 | ЩП |
| 2 | РЖС | 10 | ИЮЙ | 18 | АЦБ | 26 | ШОЩ |
| **3** | **ПЕРЕМЕНА** | 11 | ЗЭИ | 19 | ЯХА | 27 | ЧН |
| 4 | ОДП | 12 | ЖЬ | 20 | ЮФ | 28 | ЦМ |
| 5 | НГ | 13 | ЕЫ | 21 | ЭУ | 29 | ХЛЦ |
| 6 | MB | 14 | ДЪ | 22 | Ь | 30 | ФК |
| 7 | ЛБМ | 15 | ГЩ | 23 | Ы | 31 | УЙ |
| 8 | КАЛАЗ | 16 | ВШГ | 24 | Ъ | 32 | ТИУИПИРГ |

Из таблицы 2.10 мы видим, что был использован ключ ***к* = *3*** и зашифровано сообщение ПЕРЕМЕНА. Причем для того, чтобы про­ верить остальные возможные значения ключа, нам не требовалось дешифровать все восемь букв, а в большинстве случаев после ана­лиза двух-трех букв ключ отвергался (только при ***к* *= 8*** надо было дешифровать пять букв, зато при *к =* 22,23,24 хватало и одной, так как в русском языке нет слов, начинающихся с Ь, Ъ, Ы).

Из этого примера мы видим, что рассмотренный шифр совер­шенно нестоек, для его вскрытия достаточно проанализировать нес­колько первых букв сообщения и после этого ключ ***к***однозначно определяется (и, следовательно, однозначно дешифруется все сооб­щение).

В чем же причины нестойкости рассмотренного шифра и как можно было бы увеличить его стойкость? Рассмотрим еще один при­мер. Пользователь А спрятал важные документы в ячейке камеры хранения, снабженной пяти декадным кодовым замком. Теперь он захотела бы сообщить Пользователю В комбинацию цифр, открывающую ячейку. Он реши­ла использовать аналог шифра Цезаря, адаптированный к алфавиту, состоящему из десятичных цифр:

***с= (т + к)* mod 10, (2.3)**

Допустим, А послала В шифротекст 26047. Пользователь С пытается рас­шифровать его, последовательно перебирая все возможные ключи. Результаты попыток сведены в таблицу 2.11.

Таблица 2.11

Расшифровка сообщения 26047 путём перебора ключей

|  |  |  |  |
| --- | --- | --- | --- |
| ***к*** | ***m*** | ***к*** | ***m*** |
| 1 | 15936 | 6 | 60481 |
| 2 | 04825 | 7 | 59370 |
| 3 | 93714 | 8 | 48269 |
| 4 | 82603 | 9 | 37158 |
| 5 | 71592 | 0 | 26047 |

Мы видим, что все полученные варианты равнозначны и пользователь С не может понять, какая именно комбинация истинна. Анализируя шиф­ротекст, он не может найти значения секретного ключа. Конечно, до перехвата сообщения у С было 105 возможных значений кодовой комбинации, а после — только 10. Однако важно отметить то, что в данном случае всего 10 значений ключа. Поэтому при таком ключе (однадесятичная цифра) А и В не могли рассчитывать на большую секретность.

В первом примере сообщение — текст на русском языке, поэто­му оно подчиняется многочисленным правилам, различные буквы и их сочетания имеют различные вероятности и, в частности, мно­гие наборы букв вообще запрещены. (Это свойство называется из­быточностью текста). Поэтому-то и удалось легко подобрать ключ и дешифровать сообщение, т.е. избыточность позволила «взломать» шифр. В противоположность этому, во втором примере все комби­нации цифр допустимы. «Язык» кодового замка не содержит избы­точности. Поэтому даже простой шифр, применённый к сообщени­ям этого языка, становится невскрываемым.

Описанная в приведённых примерах атака называется атакой по *шифротексту.* Но часто на шифр может быть проведена атака по *известному тексту.* Это происходит, если Пользователь С получает в своё распоряжение какие-либо открытые тексты, соответствующие ран­нее переданным зашифрованным. Сопоставляя пары «текст-шифротекст», Пользователь С пытается узнать секретный ключ, чтобы с его помощью дешифровать все последующие сообщения от А к В.

Можно представить себе и более «серьёзную» атаку — атаку по *выбранному тексту,* когда противник пользуется не только предо­ставленными ему парами «текст-шифротекст», но может и сам фор­мировать нужные ему тексты и шифровать их с помощью того клю­ча, который он хочет узнать. Например, во время Второй мировой войны американцы, подкупив охрану, выкрали шифровальную ма­шину в японском посольстве на два дня и имели возможность по­давать ей на вход различные тексты и получать соответствующие шифровки. (Они не могли взломать машину с целью непосредствен­ного определения заложенного в неё секретного ключа, так как это было бы замечено и повлекло бы за собой смену всех ключей.)

Может показаться, что атаки по известному и выбранному тек­сту надуманы и далеко не всегда возможны. Отчасти это так. Но раз­работчики современных криптосистем стремятся сделать их неуяз­вимыми даже и по отношению к атакам по выбранному тексту и на этом пути достигнуты значительные успехи. Иногда считается, что более надёжно использовать шифр, противостоящий атаке по выбранному тексту, чем организационно обеспечивать неосуществи­мость такой атаки, хотя наиболее осторожные пользователи делают и то, и другое [13].

*3. Различные способы криптоанализа*

Оценки времени и стоимости вскрытия грубой силой

Вспомните, что вскрытие грубой силой обычно является вскрытием с использованием известного открытого текста, для этого нужно немного шифротекста и соответствующего открытого текста. Если вы предполагаете, что наиболее эффективным способа взлома алгоритма является вскрытие грубой силой - большое допущение - то ключ должен быть достаточно длинным, чтобы сделать вскрытие невозможным. Насколько длинным?

Скорость вскрытия грубой силой определяется двумя параметрами: количеством проверяемых ключей и скоростью проверки одного ключа. Большинство симметричных алгоритмов в качестве ключа могут использовать в качестве ключа любую битовую последовательность фиксированной длины. Длина ключа DES составля­ет 56 бит, всего может быть 256 возможных ключей. Длина ключей для ряда алгоритмов, обсуждаемых в этой книге, равны 64 битам, всего может быть 264 возможных ключей. Другие алгоритмы используют 128-битовые ключи.

Скорость, с которой может быть проверен каждый ключ, имеет менее важное значение. Для проводимого анализа я предполагаю, что скорость проверки ключа для каждого алгоритма примерно одинакова. В действительности скорость проверки одного алгоритма может быть в два, три или даже десять раз выше, чем другого. Но так как для тех длин ключей, для которых мы проводим поиск, время поиска в миллионы раз больше, чем время проверки одного ключа, небольшие отличия в скорости проверки не имеют значения.

В криптологической среде большинство споров по поводу вскрытия грубой силой сконцентрированы вокруг алгоритма DES. В 1977 году Уитфилд Диффи и Мартин Хеллман сформулировали условия существования специализированной машины по взлому DES. Эта машина состоит из миллионов микросхем, каждая из которых проверяет миллион ключей в секунду. Такая машина за два часа сможет проверить 256 за 20 часов. При вскрытии алгоритма с 64-битовым ключом проверка всех 264 потребует 214 дней.

Задача вскрытия грубой силой как будто специально придумана для параллельных процессоров. Каждый процессор проверяет подмножество пространства ключей. Процессорам не нужно обмениваться между собой информацией, единственным используемым сообщением будет сообщение, сигнализирующее об успехе. Не тре­буется и доступ к одному участку памяти. Сконструировать машину с миллионом процессоров, каждый из которых работает независимо от других, нетрудно.

Сконструировать машину для взлома грубой силой решил Майкл Винер. (Он сконструировал машину для DES, но анализ может быть выполнен почти для всех алгоритмов.) Он разработал специализиро­ванные микросхемы, платы и стойки, оценил затраты и сделал вывод, что за миллион долларов можно построить машину, которая сможет взломать 56-битный ключ DES key в среднем за 3.5 часа (и наверняка за 7 часов). Соотношение стоимость/скорость является линейным. Вспомните о законе Мура: мощь вычислительных средств удваивается приблизительно каждые 18 месяцев. Это означает, что затраты будут уменьшаться на порядок каждые пять лет, и то, что в 1995 году стоит миллион долларов, в 2000 году будет стоить около 100000 долларов. Ещё более упростить процесс вычислений могла бы конвейеризация.

Для 56-битовых ключей эти суммы оказываются вполне по карману большинству крупных корпораций и многим криминальным организациям. Военные бюджеты большинства промышленно развитых стран могут позволить взламывать и 64-битные ключи. Вскрытие 80-битного ключа все ещё за пределами возможного, но если текущая тенденция сохранится, то через каких-нибудь тридцать лет все может измениться.

Конечно, нелепо прогнозировать компьютерную мощь на 35 лет вперёд (см. Таблицу 2.12). Технологические прорывы, попу­лярные в научной фантастике, могут сделать эти прогнозы смешными. С другой стороны, неизвестные в на­стоящее время физические ограничения могут сделать эти прогнозы нереально оптимистичными. В криптогра­фии умнее быть пессимистом. Применение в алгоритме 80-битного ключа кажется недостаточно дальновидным. Используйте ключ, длина которого, по меньшей мере, 112 бит.

Таблица 2.12

Оценки среднего времени для аппаратного вскрытия грубой силой в 1995 году

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Стоимость, $ | Длина ключей в битах | | | | | |
| 40 | 56 | 64 | 80 | 112 | 128 |
| 100 000 | 2 сек | 35 ч | 1 год | 70000 лет | 1014 лет | 1019 лет |
| 1 000 000 | 0,2 сек | 3,5 ч | 37 дн | 7000 лет | 1013 лет | 1018 лет |
| 10 000 000 | 0,02 сек | 21 мин | 4 дн | 700 лет | 1012 лет | 1017 лет |
| 100 000 000 | 2 млсек | 2 мин | 9 ч | 70 лет | 1011 лет | 1016 лет |
| 1 миллиард | 0,2 млсек | 13 сек | 1 ч | 7 лет | 1010 лет | 1015 лет |
| 10 миллиардов | 0,02 млсек | 1 сек | 5,4 мин | 245 дн | 109 лет | 1014 лет |
| 100 миллиардов | 2 мксек | 0,1 сек | 32 сек | 24 дн | 108 лет | 1013 лет |
| 1 триллион | 0,2 мксек | 0,01 сек | 3 сек | 2,4 дн | 107 лет | 1012 лет |
| 10 триллионов | 0,02 мксек | 1 млсек | 0,3 сек | 6 ч | 106 лет | 1011 лет |

Если взломщик очень сильно хочет взломать ключ, все, что ему нужно, это потратить деньги. Следовательно, стоит попытаться определить минимальную «цену» ключа: в пределах какой стоимости сведе­ний можно пользоваться одним ключом прежде, чем его вскрытие станет экономически выгодным? Крайний случай: если шифрованное сообщение стоит $1.39, то нет финансового смысла устанавливать аппаратуру стои­мостью 10 миллионов долларов для взлома этого ключа. С другой стороны, если стоимость открытого текста – 100 миллионов долларов, то дешифрирование этого одиночного сообщения вполне окупит стоимость аппаратуры взлома. Кроме того, стоимость многих сообщений со временем очень быстро падает.

*Программное вскрытие*

Без специализированной аппаратуры и огромных параллельных машин вскрытие грубой силой намного сложнее. Программное вскрытие в тысячи раз медленнее, чем аппаратное.

Реальная угроза программного вскрытия грубой силой страшна не своей неизбежностью, а тем, что такое вскрытие «свободно». Ничего не стоит загрузить простаивающий микрокомпьютер проверкой возможных ключей. Если правильный ключ будет найден - замечательно, если нет - ничего не потеряно. Ничего не стоит ис­пользовать для этого целую сеть микрокомпьютеров. В недавних экспериментах с DES 40 рабочих станций в течение одного дня сумели проверить 234 ключей. При этой скорости для проверки всех ключей потребуется четыре миллиона дней, но если попытки вскрытия будут предприняты достаточным количеством людей, то кому-нибудь где-нибудь повезёт.

Основной угрозой программного вскрытия является слепое везение. Представьте себе университетскую сеть из 512 объединённых в сеть рабочих станций. Для некоторых университетских городков это сеть весьма среднего размера. Такие сети могут даже расползтись по всему миру, координируя свою деятельность по электронной почте. Пусть каждая рабочая станция способна работать (с алгоритмом) со скоростью 15000 шифрований в секунду... С учётом накладных расходов на проверку и смену ключей уменьшим скорость до 8192 проверок в секунду на машину. Чтобы, используя описанную систему, исчер­пать пространство (56-битовых) ключей потребуется 545 лет (в предположении, что сеть тратит на эту задачу 24 часа в сутки). Заметим, однако, что с помощью таких вычислений сторонники нашего студента получают один шанс из 200000 рас­крыть ключ в течение одного дня. За долгий уикенд их шансы возрастают до одного из шестидесяти шести тысяч. Чем быст­рее их аппаратура, или чем больше задействовано машин, тем лучше становятся их шансы. Вероятность заработать на жизнь, выигрывая на скачках, невысока, но разве не эти выигрыши заполняют собой пресс-релизы. К примеру, это гораздо большая вероятность, чем возможность выигрыша в правительственных лотереях. «Один на миллион?» «Один раз за тысячу лет?» Больше невозможно с полной ответственностью делать такие заявления. Является ли приемлемым этот продолжающийся риск?

Использование алгоритма с 64-битовым ключом вместо 56-битового ключа делает это вскрытие в 256 раз сложнее. А 40-битовый ключ делает картину просто безрадостной. Сеть из 400 компьютеров с производитель­ностью 32000 шифрований (на сегодняшний день персональные компьютеры в миллион раз производительнее) в секунду может за день выполнить вскрытие грубым взломом 40-битового ключа. Производительность современных суперкомпьютеров по сравнению с 1997 годом увеличилась в 1000 раз, а персональных в сотни тысяч раз.

128-битовый ключ вроде и должен быть надёжным, но современные алгоритмы шифрования AES используют ключи 128/192/256 бит что делает нелепым даже мысль о скрытии грубым взломом. По оценке Gartner к концу 2014 года в мире будет использоваться более 2 миллиардов компьютеров. Даже если все эти компьютеры будут брошены на вскрытие грубой силой, и каждый из них будет выполнять миллиард шифрований в секунду, время раскрытия ключа все равно будет в сотни тысяч раз больше времени существования вселенной.

*Нейронные сети*

Нейронные сети не слишком пригодны для криптоанализа, в первую очередь из-за формы пространства решений. Лучше всего нейронные сети работают с проблемами, имеющими непрерывное множество решений, одни из которых лучше других. Это позволяет нейронным сетям обучаться, предлагая все лучшее и лучшие решения. Отсутствие непрерывности в алгоритме почти не оставляет места обучению: вы либо раскроете ключ, либо нет. (По крайней мере, это верно при использовании любого хорошего алгоритма.) Нейронные сети хоро­шо работают в структурированных средах, где обучение возможно, но не в высокоэнтропийном, кажущемся случайным мире криптографии.

*Вирусы*

Самая большая трудность в получении миллионов компьютеров для вскрытия грубым взломом - это убедить миллионы компьютерных владельцев принять участие во вскрытии. Вы могли бы вежливо попросить, но это требует много времени, и они могут сказать нет. Вы могли бы пробовать силой ворваться в их компьютеры, но это потребует еще больше времени и может закончиться вашим арестом. Вы могли бы также использовать компьютерный вирус, чтобы распространить программу взлома среди как можно большего количества компьюте­ров.

Взломщик пишет и выпускает на волю компьютерный вирус. Этот вирус не переформатирует жёсткий диск, не удаляет файлы, но во время простоя компьютера он работает на криптоаналитической проблемой грубого взлома. Различные исследования показали, что компьютер простаивает от 70 до 90 процентов времени, так что у вируса не будет проблем со временем для решения этой задачи. Если он нетребователен и в других отношениях, то его работа даже не будет заметна.

В конце концов, одна из машин наткнётся на правильный ключ. В этот момент имеются два варианта про­должения. Во-первых, вирус мог бы породить другой вирус. Он не делал бы ничего, кроме самовоспроизведения и удаления всех найденных копий вскрывающего вируса, но содержал бы информацию о правильном ключе. Этот новый вирус просто распространялся бы среди компьютеров, пока не добрался бы до компьютера человека, который написал первоначальный вирус.

Другим, трусливым подходом был бы вывод на экран следующего сообщения:

В этом компьютере есть серьёзная ошибка. Пожалуйста, позвоните 1-8001234567 и продиктуйте оператору следующее 64-битовое число:

xxxx xxxx xxxx xxxx

Первому, кто сообщит об этой ошибке будет выплачено вознаграждение 100 долларов.

Насколько эффективно такое вскрытие? Пусть типичный заражённый компьютер проверяет тысячу ключей в секунду. Эта скорость намного меньше потенциальных возможностей компьютера, ведь мы полагаем, что он иногда будет делать и другие вещи. Предположим также, что типичный вирус инфицирует 10 миллионов машин. Этот вирус может вскрыть 56-битовый ключ за 83 дня, а 64 битовый - за 58 лет. Вам возможно при­шлось бы подкупить разработчиков антивирусного программного обеспечения, но это уже ваши проблемы. Лю­бое увеличение скорости компьютеров или распространения вируса, конечно, сделало бы это нападение более эффективным.

***Занятие №12.*** *Перспективные направления криптографии*

***Цели занятия:***

* познакомить с перспективными направлениями в криптографии.

***Задачи занятия:***

* довести тенденции в области программного обеспечения;
* познакомить с элементами квантовой криптографии.

***Учебные вопросы, рассматриваемые на занятии:***

1. Развитие облачных технологий.

2. Квантовая криптография.

***Материал для проведения занятия***

*1. Развитие облачных технологий*

Облачные технологии продолжают трансформировать отрасль за отраслью, появляясь там, где их появление, казалось бы, наименее логично. Процесс в значительной степени напоминает рождение и триумфальное шествие компьютеров по разнообразному ландшафту человеческой деятельности. Сегодня уже мало кто задумывается о том, как компьютеры изменили изготовление газет и журналов, производство, сельское хозяйство, а особенно бизнес во всех его проявлениях. Теперь же точно так же меняют всё вокруг облака, и некоторые области уже по второму кругу. Например, бухгалтерию.

В 1994 году Главное управление безопасности ФАПСИ разработало первый стандарт электронной подписи в России, но тогда в стране было ещё весьма смутное время, поэтому по-настоящему об электронной подписи заговорили лишь 8 лет спустя, в 2002-м, когда был утверждён новый стандарт криптозащиты ЭП, фактически уравнивающий российское понятие «электронная подпись» и международное — «цифровая подпись». Так что история этой технологии в нашей стране хоть и насчитывает уже двадцать лет, но фактически применяется не больше десяти.

И бóльшую часть этого десятилетия технология работала следующим образом. На компьютеры организации (как правило, лишь в бухгалтерии), устанавливалось специальное ПО для работы с ЭП, а на USB-носителе содержались персонализированные ключи, хранившиеся в единственном экземпляре. Надо сказать, что безопасность в этом случае обеспечивалась практически полная. Не завладев той самой «флешкой» с ключами — токеном, — невозможно было подписать документы от имени организации. Но ведь были и минусы! Токен можно украсть, потерять, уничтожить физически — и тогда придётся вновь проходить процедуру авторизации в удостоверяющем центре. А если нужно подписать срочные документы? Словом, облачные технологии уже стояли у порога, чтобы навсегда изменить очередную отрасль, и сегодня сектор электронного документооборота может стать локомотивом их развития.

Как привычная технология ЭП выглядит в облаке? Удостоверяющий центр создаёт вашу электронную подпись и располагает её в собственном облаке. Никаких токенов в этом случае не нужно: авторизация происходит по СМС, через привязанный мобильный телефон. Сама подпись располагается в облаке, поэтому подписывать счета и прочие документы можно с любого устройства с доступом в интернет: с офисного компьютера, с личного ноутбука, с планшета или даже смартфона. У такого подхода есть очевидные плюсы:

1. Её стоимость ниже. Приобретение облачной электронной подписи требует меньше затрат, чем покупка в обычном режиме. Это связано с тем, что для работы с этой подписью не нужно приобретать носитель и средство криптографической защиты информации (далее — СКЗИ). В случае облачной электронной подписи СКЗИ находится только на сервере, где хранится закрытый ключ. Всё это оформляется соответствующими соглашениями и доверенностями.

2. Мобильность. Сейчас интернет есть практически везде, а это значит, что и подписывать документы облачной электронной подписью можно с любого планшета, смартфона, устройства, поддерживающего выход в интернет. Ни бумага, ни электронная подпись на носителе не дают такой возможности. СКЗИ для мобильных устройств сейчас, конечно, разрабатываются, но вообще без СКЗИ на вашем устройстве, согласитесь, работать проще. Кроме того, закрытый ключ облачной ЭП не придётся устанавливать вам лично или платить сотруднику УЦ, который все настроит. Не нужно будет обучать пользователей работе со СКЗИ и сертификатами ЭП.

Но, обладая массой положительных качеств, облачная подпись имеет и негативные моменты. Несмотря на то, что через популярные бухгалтерские сервисы уже выдали более 100 000 облачных ЭП за 2013 год, широкое применение подписей все ещё под вопросом.

Одним из главных минусов — скорее из понятийной области. Сущность электронной подписи подразумевает замену собственноручной: то есть вы лично, собственными руками подписываете документ с помощью конфиденциальной части ключа. Она должна быть у вас и только у вас. В облачном же варианте закрытый ключ оказывается не в ваших руках — а где-то там на сервере УЦ. То есть фактически вы ставите подпись не собственными руками, а через посредника. Конечно, все это будет оформлено документально, а сами серверы надёжно защищены, но не во всех организациях служба безопасности одобрит такое. Если вам важно, чтобы документы подписывали сами владельцы закрытых ключей, то облачная электронная подпись вам не подойдёт.

В целом же перспективы облачной ЭП и электронного документооборота в нашей стране обнадёживают. Госдума уже утвердила план развития электронного правительства до 2018 года, который включает и ряд мер по содействию бизнесу. Например, «снижение среднего числа обращений представителей бизнес-сообщества в орган госвласти для получения одной госуслуги». И пусть тезис звучит не очень внушительно, поскольку количество обращений планируют сократить лишь до двух, это уже определённый прогресс, приводящий нас к европейскому сценарию. То есть такой ситуации когда открыть бизнес, заплатить налоги и подписать любые документы можно будет в интернете, а зачастую и со смартфона [18].

*2. Квантовая криптография*

В перспективах возможно разработать методы и их программно-аппаратную реализацию по увеличению длины псевдослучайной последовательности, а также осуществить передачу ключей на основе квантовой криптографии. Идея вычислительной системы основана на вероятностной логике и работает с квантовыми битами - кубитами, которые могут находится в трёх состояниях - двух фиксированных и в состояние суперпозиции. Квантовые компьютеры должны обладать высокой вычислительной мощностью, но ограничение квантовой механики не позволяли создать рабочий модели. В конце 2012 года учёным удалось создать из двух кубитов первый прототип квантового логического инвертора - элемент, осуществляющий операцию «контролируемое НЕ». Конечно, одного логического элемента недостаточно для создания полноценной вычислительной системы, но последние исследования в области квантовой физики открывают пути к дальнейшим развитиям в данной области.

# Заключение

Основной военной специальностью, напрямую связанной с предметом Информатика, на которую могут поступить кадеты Оренбургского президентского кадетского училища, является "Информационная безопасность". При поступлении в высшее военное учебное заведение на эту специальность необходимо сдавать ЕГЭ по информатике.

Поэтому основной целью элективного курса является знакомство учащихся с важнейшими путями и методами применения знаний, полученных при изучении предмета "Информатика и ИКТ" на практике, развитие интереса учащихся к современным информационным технологиям, программированию, шифрованию, что в дальнейшем окажет помощь в выборе профиля дальнейшего обучения.

В ходе элективного курса кадеты более подробно познакомятся с программированием и применением, полученных знаний для сохранения информации в безопасности, изучат способы шифрования и дешифрования данных, познакомятся с методами криптоанализа.

В связи с профилизацией современных старших классов, актуальнойостаётся проблема оказания помощи учащимся в выборе «своего» профиля, для обучения в старших классах, в направлении учеников в необходимую на сегодняшний день сферу деятельности, в сферу цифровых технологий и защиты информации. Именно эту проблему и решает элективный курс.

# Список использованных источников

1. Алгоритм DES взломан за три дня [Электронный ресурс] // Международный компьютерный ежедневник [сайт] <http://www.osp.ru/cw/1998/28-29/30844/> (дата обращения: 20.10.2014).
2. Александр Бабаш «Тайна головы раба» [Электронный ресурс] // Санкт-Петербургский государственный политехнический университет [сайт] <http://www.ssl.stu.neva.ru/psw/crypto/babash_article.html> (дата обращения: 9.09.2014).
3. Баричев С.Г, Серов Р.Е. Основы современной криптографии: Учебное пособие. – М.: Горячая линия – Телеком, 2002.
4. Введение в криптографию / Под. ред. В.В. Ященко. – СПб.: Питер, 2001. – 288 с.
5. Гатчин Ю.А., Коробейников А.Г. Основы криптографических алгоритмов. Учебное пособие. СПб: ГИТМО (ТУ), 2002. 29 с.
6. Гончаров Н.О. Симметричное шифрование (гаммирование) [Электронный ресурс] // Наука и образование. Электронное научно-техническое издание [сайт] <http://technomag.edu.ru/doc/187185.html> (Дата обращения 21.09.2014).
7. Грибунин В.С., Оков И.Н., Туринцев И.В. Цифровая стенография. – Москва: Солон-пресс, 2009. – 272 с.
8. Жельников В. Криптография от папируса до компьютера. – М.: Изд‑во »ABF», 1996. – 335 с.
9. Криптография: Методические указания / В.В. Анисимов. – Хабаровск: Изд-во ДВГУПС, 2004. – 32 с.: ил.
10. Панасенко С.П. Алгоритмы шифрования. Специальный справочник. – СПб.: БХВ-Петербург, 2009. – 576 с.: ил.
11. Письмо Минобразования РФ от 13.11.2003 №14-51-277/13 Об элективных курсах в профильном обучении.
12. Полугодовой отчёт Cisco по информационной безопасности проливает свет на «слабые звенья» ИС [Электронный ресурс] // Компания Cisco [сайт] <http://www.cisco.com/web/RU/news/releases/txt/2014/08/080714a.html> (дата обращения: 23.09.2014)
13. Рябко Б.Я., Фионов А.Н. Криптографические методы защиты информации: учебное пособие для ВУЗов. – М.: Горячая линия – Телеком, 2005. – 229 с.: ил. – ISBN 5-93517-265-8.
14. Современная ситуация в области информационной безопасности [Электронный ресурс] // CITForum.ru on-line библиотека [сайт] <http://citforum.ru/internet/infsecure/its2000_03.shtml> (дата обращения: 14.09.2014)
15. Фергюсон Нильс, Шнайер Брюс. Практическая криптография: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 424 с.: ил. – Парал. Тит. англ.
16. Шеннон К.Э. Работы по теории информации и кибернетике, М.: ИЛ, 1963. 832 с.
17. Шнайер Брюс. Прикладная криптография.  (Applied Cryptography) 2-ое издание. – М.: Изд-во «Триумф», 2012. – 816 с.
18. Электронная подпись и бизнес: применят ли в Росии облачную ЭП? [Электронный ресурс] // ООО «Компьютерра-онлайн» [сайт] <http://www.computerra.ru/102651/elektronnaya-podpis-i-biznes-primenyayut-li-v-rossii-oblachnuyu-etsp/> (дата обращения: 23.09.2014).

# Приложения

Приложение 1

Реализация шифра Цезаря различными способами

Пример 1

**Program Caesar;**

**var**

lalp: integer;

**const**

alp = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя' +

'АБВГДЕЁЖЗИЙКЛМНПОРСТУФХЦЧШЩЪЫЬЭЮЯ' + 'abcdefghijklmnopqrstuvwxyz' +

'ABCDEFGHIJKLMNOPQRSTUVWXYZ' + '0123456789';

**function** Caesar\_cipher(str: string; way: integer; key: integer): string;

**var**

s: string;

i, t: integer;

**begin**

**if** way > 0 **then**

key := -key;

key := key **mod** lalp;

**if** key = 0 **then**

**begin**

writeln('Error: wrong key (key not equals zero and not divisible by ',

lalp, ')');

Caesar\_cipher := str;

**exit**;

**end**;

s := str;

**for** i := 1 **to** length(str) **do**

**begin**

t := pos(str[i], alp);

**if** t <> 0 **then**

s[i] := alp[(t - 1 + key + lalp) **mod** lalp + 1];

**end**;

Caesar\_cipher := s;

**end**;

**begin**

lalp := length(alp);

writeln(Caesar\_cipher('Veni Vidi Vici', 0, 3));

// 0 - шифрование; не 0 - дешифрование

// 3 - сдвиг

readln;

**end**.

Пример 2

**Uses** CRT;

**Var**

n:integer;

**function** code(toCode:string):string;

**var**

x,y: integer;

**begin**

writeln('shag coda');

readln(n);

**for** x := 1 **to** length(toCode) **do**

**begin**

y :=(Ord(toCode[x])+n);

toCode[x]:= Chr(y);

**end**;

code := toCode;

**end**;

**function** decode(toCode:string):string;

**var**

x, y: integer;

**begin**

**for** x := 1 **to** length(toCode) **do**

**begin**

y := (Ord(toCode[x])-n);

toCode[x] := Chr(y);

**end**;

decode:=toCode;

**end**;

**var**

s: string[200];

**begin**

writeln('text');

readln(s);

s := (code(s));

writeln(s);

writeln(decode(s));

readln

**end**.

Программа реализующая алгоритм квадрата Полибия

**const**

TPolibius: **array**['A' .. 'F', 'A' .. 'E'] **of** char = (

('а', 'б', 'в', 'г', 'д'),

('е', 'ж', 'з', 'и', 'к'),

('л', 'м', 'н', 'о', 'п'),

('р', 'с', 'т', 'у', 'ф'),

('х', 'ц', 'ч', 'ш', 'щ'),

('ы', 'ь', 'э', 'ю', 'я')

);

**function** PolibiusEncipher(toCode: string): string;

**var**

i: integer;

ix, jx: char;

s: string;

**begin**

s := '';

**for** i := 1 **to** length(toCode) **do**

**begin**

**for** ix := 'A' **to** 'F' **do**

**for** jx := 'A' **to** 'E' **do**

**if** TPolibius[ix, jx] = toCode[ i ] **then**

**begin**

s := s + ix + jx; **break**;

**end**;

**end**;

PolibiusEncipher := s

**end**;

**function** PolibiusDecipher(toDecode: string): string;

**var**

i: integer;

s: string;

**begin**

s := '';

i := 1;

**while** i <= length(toDecode) **do**

**begin**

s := s + TPolibius[toDecode[ i ], toDecode[succ(i)]];

inc(i, 2);

**end**;

PolibiusDecipher := s

**end**;

**var**

s: string;

**begin**

s := PolibiusEncipher('квадрат');

writeln(s);

writeln('s = ', PolibiusDecipher(s));

**end**.

Приложение 2

Программа с шифрования с закрытым ключом

**Program** Cript\_Caesar;

**Uses** CRT;

**Var**

a, b, c: string;

i, j, k, d: integer;

**Begin**

ClrScr;

writeln ('Введите текст для шифрования');

readln (a);

writeln ('Введите ключ');

readln (d);

c: = '';

b: = 'абвгдежзийклмнопрстуфхцчшщъыьэюя';

**for** i: = 1 **to** length(a) **do**

**for** j: = 1 **to** length(b) **do**

**begin**

k: = (j+d) **mod** 32;

**if** a[i] = b[j] **then** c: = c+b[k];

**end**;

writeln (c);

readkey;

**End**.

Приложение 3

Текст программы шифрования с открытым ключом

**Function** Flag(n: longint): Boolean; {Функция проверки простоты числа}

**var**

i, k: longint;

**begin**

k := Round (sqrt(n)); {максимально возможный делитель}

**if** n = 2 **then** flag := true

**else if not** odd (n) **then** flag := false

**else begin**

flag := true;

**for** i := 2 **to** k **do**

**if** n **mod** i = 0 **then** flag := false

**end**;

**end**;

**Function** f(n : longint): longint; {функция вычисления фунции Эйлера}

**var**

res, i: longint;

**begin**

res := n;

**for** i := 2 **to** n **do**

**if** (n **mod** i = 0) **then** {если i - делитель}

**begin**

**while** (n **mod** i = 0) **do** {пока делится, делим}

n := n **div** i;

dec(res, res **div** i);

**end**;

**if** (n > 1) **then** dec(res, res **div** i);

f := res;

**end**;

**Function** VVod(Flag1:boolean): longint; {функция повторения ввода, если число не простое}

**var**

z: longint;

**begin**

**While Not** Flag1 **do**

**begin**

write('Это составное число,');

write ('повторите ввод: ');

readln (z);

Flag1 := Flag (z);

**end**;

Vvod := z;

**end**;

**Function** Nod(x, y: Longint): longint; {Функция вычиления НОД, чтобы числа были взаимно простымы}

**Begin**

**Repeat**

**If** x > y **Then** x:=x **Mod** y **Else** y := y **Mod** x;

**Until** (x=0) **Or** (y=0);{до тех пор, пока одно из чисел не станет равно нулю}

Nod := x+y;

**End**;

**Function** Alfa (x,y: longint):longint; {Функция вычисления секретного кода}

**var**

i: longint;

**Begin**

i:=0;

**repeat**

i:=i+1

**until** (x\*i+1) **mod** y = 0;

Alfa := round((x\*i+1)/y);

**End**;

**Function** Ostatok (x, y, t: longint): longint; {Вычисление остатка от деления произведения x\*y на t}

**var**

i, k: longint;

**begin**

k := 1;

**for** i := 1 **to** y **do**

k := (k\*x) **mod** t;

Ostatok := k;

**end**;

**Var**

p1, p2, q1, q2, Ra, Rb, a, b, Fa, Fb, Aa, Bb : longint;

N, m, m1, m2: longint;

**Begin**

write ('Введите p1 - '); {ВВод данных для абонента А}

readln (p1);

**If Not** Flag(p1) **Then** p1 := Vvod(Flag(p1));

write ('Введите p2 - ');

readln (p2);

**If Not** Flag(p2) **Then** p2 := Vvod(Flag(p2)); {p1 := 13; p2 := 17;}

Ra := p1\*p2;

writeln ('Ra = p1 \* p2 = ', Ra);

Fa := f(Ra);

Write ('Введите открытый ключ а, взаимно простой с с функцией Эйлера Fa = ', Fa, '; - ');

Readln (a);

N := Nod (Fa, a);

**While** N > 1 **do**

**Begin**

write('Это не взаимно простые числа,');

write ('повторите ввод: ');

readln (a);

N := Nod(Fa, a)

**End**;

Aa := Alfa (Fa,a);

writeln ('Сектретный ключ - ', Aa);

write ('Введите q1 - '); {ВВод данных для абонента А}

readln (q1);

**If Not** Flag(q1) **Then** q1 := Vvod(Flag(q1));

write ('Введите q2 - ');

readln (q2);

**If Not** Flag(q2) **Then** q2 := Vvod(Flag(q2)); {q1 := 11; q2 := 29;}

Rb := q1\*q2;

writeln ('Rb = q1 \* q2 = ', Rb);

Fb := f(Rb);

Write ('Введите открытый ключ b, взаимно простой с функцией Эйлера Fb = ', Fb, '; - ');

Readln (b);

N := Nod (Fb, b);

**While** N > 1 **do**

**Begin**

write('Это не взаимно простые числа,');

write ('повторите ввод: ');

readln (b);

N := Nod(Fb, b)

**End**;

Bb := Alfa (Fb,b);

writeln ('Сектретный ключ - ', Bb);

Write ('Введите послание абонента А для В m = ');

Readln (m);

m1 := Ostatok (m, b, Rb);

Writeln ('Шифровка от абонента А к абоненту В зашифрованная открытым ключом абонента А m1 = ', m1);

m2 := Ostatok (m1, Bb, Rb);

Writeln ('Расшифрованный абонентом В текст при помощи секретного ключа будет m2 = ', m2);

**End**.

Приложение 4

Глоссарий

**AES** (Advanced Encryption Standard), также известный как **Rijndael** - симметричный блочный алгоритм шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительсвом США с 2002 года. По состоянию на 2009 год AES является одним из самых распространённых алгоритмов симметричного шифрования. Поддержка AES (и только его) введена фирмой Intel в семейство процессоров x86 начиная с Intel Core i7-980X Extreme Edition, а затем на процессорах Sandy Bridge.

**DES** (Data Encryption Standard) — симметричный алгоритм шифрования, разработанный фирмой Ай-Би-Эм и утверждённый правительством США в 1977 году (до 2002 года, потом его заменил AES) как официальный стандарт (FIPS 46-3). DES имеет блоки по 64 бита и 16-цикловую структуру сети Фейстеля для шифрования использует ключ с длиной 56 бит. Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP-1) преобразований.

**DoS** (от англ. *Denial of Service* — отказ от обслуживания) — хакерская атака на вычислительную систему (обычно совершенная хакерами) с целью довести её до отказа, то есть создание таких условий, при которых легальные пользователи системы не могут получить доступ к предоставляемым системным ресурсам (серверам), либо этот доступ затруднён. Отказ «вражеской» системы может быть и шагом к овладению системой (если в нештатной ситуации ПО выдаёт какую-либо критическую информацию — например, версию, часть программного кода и т. д.). Но чаще это мера экономического давления: простой службы, приносящей доход, счета от провайдера и меры по уходу от атаки ощутимо бьют «цель» по карману. В настоящее время DoS и DDoS-атаки наиболее популярны, так как позволяют довести до отказа практически любую систему, не оставляя юридически значимых улик.

**Аутентичность**информа­ции состоит в *подлинности авторства* и *целостности.*

**Биграмм** – шифрование пар букв.

**Безопасность информации** - состояние информации, информационных ресурсов и информационных систем, при котором с требуемой вероятностью обеспечивается защита информации (данных) от утечки, хищения, утраты, несанкционированного уничтожения, искажения, модификации (подделки), копирования, блокирования информации и т.п. [14].

**Злоумышленник** - лицо, осуществляющее осознанные действия по нарушению информационной безопасности объекта защиты.

**Информация** - сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления [18].

Под **информационной безопасностью** Российской Федерации понимается состояние защищенности ее национальных интересов в информационной сфере, определяющихся совокупностью сбалансированных интересов личности, общества и государства [9].

**Кибербезопасность** — это набор средств, стратегии, принципы обеспечения безопасности, гарантии безопасности, руководящие принципы, подходы к управлению рисками, действия, профессиональная подготовка, практический опыт, страхование и технологии, которые могут быть использованы для защиты киберсреды, ресурсов организации и пользователя. Ресурсы организации и пользователя включают подсоединенные компьютерные устройства, персонал, инфраструктуру, приложения, услуги, системы электросвязи и всю совокупность переданной и/или сохраненной информации в киберсреде. Кибербезопасность состоит в попытке достижения и сохранения свойств безопасности у ресурсов организации или пользователя, направленных против соответствующих угроз безопасности в киберсреде. Общие задачи обеспечения безопасности включают следующее: доступность; целостность, которая может включать аутентичность и неотказуемость; конфиденциальность.

Под **конфиденциальностью**понимают невозможность полу­чения информации из преобразованного массива без знания до­полнительной информации (ключа).

**Криптоанализ** (греч. κρυπτός — скрытый и ανάλυση — разложение, расчленение) – наука, занимающаяся вопросами оценки сильных и слабых сторон методов шифрования, а также разработкой методов, позволяющих взламывать криптосистемы, другими словами объединяет математические методы наруше­ния конфиденциальности и аутентичности информации без зна­ния ключей.

**Криптография** (греч. κρυπτός — скрытый и γράφω — пишу, рисую) – наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним) и аутентичности (целостности и подлинности авторства, а также невозможности отказа от авторства) информации.

**Криптология** *-* наука, изучающая математические методы защиты инфор­мации путем ее преобразования. Криптология разделяется на два направления - *криптографию* и *криптоана­лиз.*

**Кубит** – (q-бит, кьюбит, кубит; от *quantum bit*) — квантовый разряд или наименьший элемент для хранения информации в квантовом компьютере.

**Полибианский квадрат** – греческий писатель Полибий: случайно заполненный алфавитом квадратной таблицы каждая буква исходного текста заменяется на букву, стоящую в квадрате снизу от неё (пример моноалфавитной подстановки).

**При́нцип Керкго́ффса** — правило разработки криптографических систем, согласно которому противник знает об используемой системе шифрования всё, кроме применяемых ключей. В формулировке Шеннона «Враг знает систему».

**Роторные криптосистемы** – многоалфавитная подстановка с помощью роторной машины, реализуемая вариацией взаимного положения вращающихся роторов, каждый из которых осуществляет прошитую в нём подстановку (первая машина изобретена Томасом Джефферсоном, Enigma – немецкая машина).

**Сложение по модулю** – остаток от деления суммы на число.

**Стеганография** – наука о скрытой передачи информации.

**Теорема Эйлера:** ***aϕ(m) ≡ 1 (mod m)***

**Функция Эйлера** ***ϕ(n)*** — мультипликативная арифметическая функция, равная количеству натуральных чисел, меньших ***n*** и взаимно простых с ним. При этом полагают, что число 1 взаимно просто со всеми натуральными числами, и ***ϕ(1)=1.*** Функция Эйлера ***ϕ(n)*** показывает, сколько натуральных чисел из отрезка ***[1, n-1]***  имеют c ***n*** только один общий делитель — единицу. Функция Эйлера определена на множестве натуральных чисел, и значения её лежат во множестве натуральных чисел. Например, для числа 24 существует 8 меньших него и взаимно простых с ним чисел (1, 5, 7, 11, 13, 17, 19, 23), поэтому ***ϕ(24)=8***.

**Шифр Блеза Вижинера** – последовательное «сложение» букв исходного алфавита с ключом (многоалфавитная подстановка).

**Эксплойт**, **эксплоит**, **сплоит** (англ. *exploit*, эксплуатировать) – компьютерная программа, фрагмент программного кода или последовательность команд, использующие уязвимости в программном обеспечении и применяемые для проведения атаки на вычислительную систему. Целью атаки может быть, как захват контроля над системой (повышение привилегий), так и нарушение её функционирования (DoS-атака).

**Электро́нная по́дпись** (ЭП), **Электро́нная цифровая по́дпись** (ЭЦП) – реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий установить отсутствие искажения информации в электронном документе с момента формирования подписи и проверить принадлежность подписи владельцу сертификата ключа подписи.

1. Названия даны американскими криптоаналитиками [↑](#footnote-ref-1)