1. ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ И ЗАЩИТЫ ИНФОРМАЦИИ

1.1. Информация и информационная безопасность

Информация (лат. informatio — разъяснение, изложение), первоначально — сведения, передаваемые людьми устным, письменным или другим способом с помощью условных сигналов, технических средств и т.д. С середины 20-го века **информация** является общенаучным понятием, включающим в себя:

- сведения, передаваемые между людьми, человеком и автоматом, автоматом и автоматом;

- сигналы в животном и растительном мире;

- признаки, передаваемые от клетки к клетке, от организма к организму;

- и т.д.

Другими словами, информация носит фундаментальный и универсальный характер, являясь многозначным понятием. Эту мысль можно подкрепить словами Н. Винера (отца кибернетики): «Информация есть информация, а не материя и не энергия».

Согласно традиционной философской точке зрения, информация существует независимо от человека и является свойством материи. В рамках рассматриваемой дисциплины, под **информацией** (в узком смысле) мы будем понимать сведения, являющиеся объектом сбора, хранения, обработки, непосредственного использования и передачи в информационных системах[1].

Опираясь на это определение информации, рассмотрим понятия информационной безопасности и защиты информации.

В Доктрине информационной безопасности Российской Федерации под термином **информационная безопасность** понимается состояние защищенности национальных интересов в информационной сфере, определяемых совокупностью сбалансированных интересов личности, общества и государства.

В более узком смысле, под **информационной безопасностью** мы будем понимать состояние защищенности информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера (информационных угроз, угроз информационной безопасности), которые могут нанести неприемлемый ущерб субъектам информационных отношений [18].

Защита информации – комплекс правовых, организационных и технических мероприятий и действий по предотвращению угроз информационной безопасности и устранению их последствий в процессе сбора, хранения, обработки и передачи информации в информационных системах.

Важно отметить, что информационная безопасность – это одна из характеристик информационной системы, т.е. информационная система на определенный момент времени обладает определенным состоянием (уровнем) защищенности, а защита информации – это процесс, который должен выполняться непрерывно на всем протяжении жизненного цикла информационной системы[2].

Рассмотрим более подробно составляющие этих определений.

Под **субъектами информационных отношений** понимаются как владельцы, так и пользователи информации и поддерживающей инфраструктуры [18].

К **поддерживающей инфраструктуре** относятся не только компьютеры, но и помещения, системы электро-, водо- и теплоснабжения, кондиционеры, средства коммуникаций и, конечно, обслуживающий персонал.

Ущерб может быть **приемлемым** или **неприемлемым**. Очевидно, застраховаться от всех видов ущерба невозможно, тем более невозможно сделать это экономически целесообразным способом, когда стоимость защитных средств и мероприятий не превышает

размер ожидаемого ущерба. Значит, с чем-то приходится мириться и защищаться следует только от того, с чем смириться никак нельзя. Иногда таким недопустимым ущербом является нанесение вреда здоровью людей или состоянию окружающей среды, но чаще порог неприемлемости имеет материальное (денежное) выражение, а целью защиты информации становится уменьшение размеров ущерба до допустимых значений [18].

Информационная угроза – потенциальная возможность неправомерного или случайного воздействия на объект защиты, приводящая к потере, искажению или разглашению информации.

Таким образом, концепция информационной безопасности, в общем случае, должна отвечать на три вопроса:

- Что защищать?

- От чего (кого) защищать?

- Как защищать?

---

[1]**Информационная система** (автоматизированная информационная система) — это совокупность технических (аппаратных) и программных средств, а также работающих с ними пользователей (персонала), обеспечивающая информационную технологию выполнения установленных функций.

[2]**Жизненный цикл информационной системы** – непрерывный процесс, начинающийся с момента принятия решения о создании информационной системы и заканчивающийся в момент полного изъятия ее из эксплуатации.

1.2. Основные составляющие информационной безопасности

Спектр интересов субъектов, связанных с использованием информационных систем, можно разделить на следующие составляющие: обеспечение **доступности**, **целостности** и **конфиденциальности** информационных ресурсов и поддерживающей инфраструктуры.

Иногда в число основных составляющих информационной безопасности включают защиту от несанкционированного доступа (НСД) к информации, под которым понимают доступ к информации, нарушающий правила разграничения доступа с использование штатных средств[3]. В то же время обеспечение конфиденциальности как раз и подразумевает защиту от НСД.

Дадим определения основных составляющих информационной безопасности [18].

Доступность информации – свойство системы обеспечивать своевременный беспрепятственный доступ правомочных (авторизованных) субъектов к интересующей их информации или осуществлять своевременный информационный обмен между ними. Информационные системы создаются (приобретаются) для получения определенных информационных услуг. Если по тем или иным причинам предоставить эти услуги пользователям становится невозможно, это, очевидно, наносит ущерб всем субъектам информационных отношений. Особенно ярко ведущая роль доступности проявляется в разного рода системах управления – производством, транспортом и т.п. Внешне менее драматичные, но также весьма неприятные последствия – и материальные, и моральные – может иметь длительная недоступность информационных услуг, которыми пользуется большое количество людей (продажа железнодорожных и авиабилетов, банковские услуги и т.п.).

Целостность информации – свойство информации, характеризующее ее устойчивость к случайному или преднамеренному разрушению или несанкционированному изменению. Целостность можно подразделить на статическую (понимаемую как неизменность информационных объектов) и динамическую (относящуюся к корректному выполнению сложных действий (транзакций[4])). Средства контроля динамической целостности применяются, в частности, при анализе потока финансовых сообщений с целью выявления кражи, переупорядочения или дублирования отдельных сообщений. Целостность оказывается важнейшим аспектом информационной безопасности в тех случаях, когда информация служит «руководством к действию». Рецептура лекарств, предписанные медицинские процедуры, набор и характеристики комплектующих изделий, ход технологического процесса – все это примеры информации, нарушение целостности которой может оказаться в буквальном смысле смертельным.

Конфиденциальность информации – свойство информации быть известной и доступной только правомочным субъектам системы (пользователям, программам, процессам). Конфиденциальность – самый проработанный у нас в стране аспект информационной безопасности. К сожалению, практическая реализация мер по обеспечению конфиденциальности современных информационных систем наталкивается в России на серьезные трудности. Во-первых, сведения о технических каналах утечки информации являются закрытыми, так что большинство пользователей лишено возможности составить представление о потенциальных рисках. Во-вторых, на пути пользовательской криптографии как основного средства обеспечения конфиденциальности стоят многочисленные законодательные препоны и технические проблемы.

Если вернуться к анализу интересов различных категорий субъектов информационных отношений, то почти для всех, кто реально использует ИС, на первом месте стоит доступность. Практически не уступает ей по важности целостность – какой смысл в информационной услуге, если она содержит искаженные сведения? Наконец, конфиденциальная информация есть как у организаций, так и отдельных пользователей.

Из всего выше приведенного следует два следствия.

1. Трактовка проблем, связанных с информационной безопасностью, для разных категорий субъектов может существенно различаться. Для иллюстрации достаточно сопоставить режимные государственные организации и учебные заведения. В первом случае «пусть лучше все сломается, чем враг узнает хотя бы один секрет», во втором – «да нет у нас никаких секретов, лишь бы все работало».

2. Информационная безопасность не сводится исключительно к защите от НСД к информации, это принципиально более широкое понятие. Субъект информационных отношений может пострадать (понести убытки и/или получить моральный ущерб) не только от НСД, но и от поломки системы, вызвавшей перерыв в работе.

---

[3]**Штатные средства** - совокупность программного и аппаратного обеспечения рассматриваемой информационной системы.

[4]**Транзакция** - одно действие или их последовательность, выполняемых одним или несколькими пользователями (прикладными программами) с целью осуществления доступа или изменения информации, воспринимаемых как единое целое и переводящих ее из одного непротиворечивого (согласованного) состояния в другое непротиворечивое состояние.

1.3. Объекты защиты

Основными **объектами защиты** при обеспечении информационной безопасности являются:

- все виды информационных ресурсов. **Информационные ресурсы (документированная информация)** - информация, зафиксированная на материальном носителе с реквизитами, позволяющими ее идентифицировать;

- права граждан, юридических лиц и государства на получение, распространение и использование информации;

- система формирования, распространения и использования информации (информационные системы и технологии, библиотеки, архивы, персонал, нормативные документы и т.д.);

- система формирования общественного сознания (СМИ, социальные институты и т.д.).

1.4. Категории и носители информации

Неотъемлемой частью любой информационной системы является информация. По характеру ограничений (реализации) конституционных прав и свобод в информационной сфере выделяют четыре основных **вида правовой** (регламентированной законами)

информации:

- информация с ограниченным доступом;

- информация без права ограничения;

- иная общедоступная информация (например, за деньги);

- информация, запрещенная к распространению.

Информация с ограниченным доступом делится на государственную тайну и конфиденциальную.

К **государственной тайне** относятся защищаемые государством сведения в области его военной, внешнеполитической, экономической, разведывательной, контрразведывательной и оперативно-розыскной деятельности, распространение которых может нанести ущерб безопасности РФ. Владельцем государственной тайны является само государство. Требования по защите этой информации и контроль за их соблюдением регламентируются законом РФ «О государственной тайне» [4]. В нем законодательно установлен Перечень сведений, сопоставляющих государственную тайну, и круг сведений, не подлежащих к отнесению к ней. Предусмотрена судебная защита прав граждан в связи с необоснованным засекречиванием. Определены органы защиты государственной тайны:

- межведомственная комиссия по защите государственной тайны;

- федеральные органы исполнительной власти, уполномоченные в области:

  - обеспечения безопасности - Федеральная служба по техническому и экспортному контролю (ФСТЭК);

  - обороны – Министерство обороны;

  - внешней разведки – Федеральная служба безопасности (ФСБ обеспечивает, в т.ч. криптографическую защиту);

  - противодействия техническим разведкам и технической защиты информации – ФСТЭК;

- другие органы.

Конфиденциальная информация – документированная информация, правовой режим которой установлен специальными нормами действующего законодательства в области государственной, коммерческой, промышленной и другой общественной деятельности. Этой информацией владеют различные учреждения, организации и отдельные индивидуумы. В Указе Президента РФ «Перечень сведений конфиденциального характера» [5] конфиденциальная информация разбита на семь видов:

- персональные данные;

- тайна следствия и судопроизводства;

- **служебная тайна** - служебная информация ограниченного распространения о госорганах или подведомственных им организациях, а также информация, получаемая из внешних источников работниками госорганов при исполнении обязанностей;

- **профессиональная тайна** - информация, полученная гражданами (физическими лицами) при исполнении ими профессиональных обязанностей или организациями при осуществлении ими определенных видов деятельности [2] (врачебная, нотариальная, адвокатская тайна, тайна переписки, телефонных переговоров, почтовых отправлений, телеграфных или иных сообщений и т.д.);

- **коммерческая тайна** - научно-техническая, технологическая, производственная, финансово-экономическая или иная информация, в том числе составляющая секреты производства (ноу-хау), которая имеет действительную или потенциальную коммерческую ценность в силу неизвестности её третьим лицам;

- сведения о сущности изобретения, полезной модели или промышленного образца по официальной публикации информации о них;

- сведения, содержащиеся в личных делах осужденных, а также сведения о принудительном исполнении судебных актов.

Под **персональными данными** понимается любая информация, относящаяся прямо или косвенно к определенному или определяемому физическому лицу (субъекту персональных данных). Несмотря на то, что это информация ограниченного доступа, она является полностью открытой для субъекта персональных данных. Только сам субъект решает вопрос о передаче, обработке и использовании своих персональных данных, а также определяет круг субъектов, которым эти данные могут быть сообщены. Некоторая часть персональных данных может не иметь режима защиты, являясь общеизвестными (например, фамилия, имя и отчество). В Законе РФ «О персональных данных» [6] выделены следующие **права субъектов персональных данных** (кроме некоторых категорий граждан: владеющих государственной тайной, осужденных и т.д.):

- информационное самоопределение;

- доступ к своим персональным данным;

- внесение изменений в свои персональные данные;

- блокирование персональных данных;

- обжалование неправомерных действий в отношении персональных данных;

- возмещение ущерба.

В статье 24 Конституции РФ [1] предусмотрена защита некоторой части персональных данных - «1. Сбор, хранение, использование и распространение информации о частной жизни лица без его согласия не допускаются».

Государственные органы и организации, органы местного самоуправления имеют право на работу с персональными данными в пределах своей компетенции, установленной действующим законодательством, или на основании лицензии. В последнем случае с ними могут работать также негосударственные юридические и физические лица.

В статье 7 Закона РФ «О государственной тайне» [4] определен перечень сведений, не подлежащих отнесению к государственной тайне и засекречиванию (**информация без права ограничения**):

- о чрезвычайных происшествиях и катастрофах, угрожающих безопасности и здоровью граждан, и их последствиях, а также о стихийных бедствиях, их официальных прогнозах и последствиях;

- о состоянии экологии, здравоохранения, санитарии, демографии, образования, культуры, сельского хозяйства, а также о состоянии преступности;

- о привилегиях, компенсациях и социальных гарантиях, предоставляемых государством гражданам, должностным лицам, предприятиям, учреждениям и организациям;

- о фактах нарушения прав и свобод человека и гражданина;

- о размерах золотого запаса и государственных валютных резервах Российской Федерации;

- о состоянии здоровья высших должностных лиц Российской Федерации;

- о фактах нарушения законности органами государственной власти и их должностными лицами.

Должностные лица, принявшие решения о засекречивании перечисленных сведений либо о включении их в этих целях в носители сведений, составляющих государственную тайну, несут **уголовную**, **административную** или **дисциплинарную** ответственность в зависимости от причиненного обществу, государству и гражданам материального и морального ущерба. Граждане вправе обжаловать такие решения в суд.

В ряде статей Конституции РФ [1] также прописан беспрепятственный доступ граждан и их объединений к общественно значимой информации:

- статья 24 - «2. Органы государственной власти и органы местного самоуправления, их должностные лица обязаны обеспечить каждому возможность ознакомления с документами и материалами, непосредственно затрагивающими его права и свободы, если иное не предусмотрено законом»;

- статья 42 - «Каждый имеет право на благоприятную окружающую среду, достоверную информацию о ее состоянии и на

возмещение ущерба, причиненного его здоровью или имуществу экологическим правонарушением».

Информация, запрещенная к распространению, определена в многочисленных нормативных документах. В частности:

- статья 29 Конституции РФ - «2. Не допускаются пропаганда или агитация, возбуждающие социальную, расовую, национальную или религиозную ненависть и вражду. Запрещается пропаганда социального, расового, национального, религиозного или языкового превосходства»;

- Кодекс Российской Федерации об административных правонарушениях:

  - статья 5.61 «Оскорбление»;

  - статья 5.62 «Дискриминация»;

  - статья 6.13 «Пропаганда наркотических средств, психотропных веществ или их прекурсоров, растений, содержащих наркотические средства или психотропные вещества либо их прекурсоры, и их частей, содержащих наркотические средства или психотропные вещества либо их прекурсоры, новых потенциально опасных психоактивных веществ;

  - статья 6.17 «Нарушение законодательства Российской Федерации о защите детей от информации, причиняющей вред их здоровью и (или) развитию»;

  - статья 6.21 «Пропаганда нетрадиционных сексуальных отношений среди несовершеннолетних»;

  - статья 6.26 «Организация публичного исполнения произведения литературы, искусства или народного творчества, содержащего нецензурную брань, посредством проведения театрально-зрелищного, культурно-просветительного или зрелищно-развлекательного мероприятия»;

  - статья 14.48 «Представление недостоверных результатов исследований (испытаний)»;

  - статья 17.9 «Заведомо ложные показание свидетеля, пояснение специалиста, заключение эксперта или заведомо неправильный перевод»;

  - статья 20.3 «Пропаганда либо публичное демонстрирование нацистской атрибутики или символики, либо атрибутики или символики экстремистских организаций, либо иных атрибутики или символики, пропаганда либо публичное демонстрирование которых запрещены федеральными законами»;

  - статья 20.29 «Производство и распространение экстремистских материалов»;

  - и другие;

- Уголовный кодекс Российской Федерации:

  - статья 110 «Доведение до самоубийства» - «д) в публичном выступлении, публично демонстрирующемся произведении, средствах массовой информации или информационно-телекоммуникационных сетях (включая сеть "Интернет")»;

  - статья 119 «Угроза убийством или причинением тяжкого вреда здоровью»;

  - статья 128.1 «Клевета»;

  - статья 142 «Фальсификация избирательных документов, документов референдума»;

  - статья 155 «Разглашение тайны усыновления (удочерения)»;

  - статья 172.1 «Фальсификация финансовых документов учета и отчетности финансовой организации»;

  - статья 205.2 «Публичные призывы к осуществлению террористической деятельности, публичное оправдание терроризма или пропаганда терроризма»;

  - статья 207 «Заведомо ложное сообщение об акте терроризма»;

- статья 217.2 «Заведомо ложное заключение экспертизы промышленной безопасности»;

- статья 242 «Незаконные изготовление и оборот порнографических материалов или предметов»;

- статья 280 «Публичные призывы к осуществлению экстремистской деятельности»;

- статья 303 «Фальсификация доказательств и результатов оперативно-розыскной деятельности»;

- статья 306 «Заведомо ложный донос»;

- статья 307 «Заведомо ложные показание, заключение эксперта, специалиста или неправильный перевод»;

- статья 319 «Оскорбление представителя власти»;

- статья 354 «Публичные призывы к развязыванию агрессивной войны»;

- статья 354.1 «Реабилитация нацизма»;

- и другие.

Основными **носителями информации** являются:

- открытая печать (газеты, журналы, отчеты, реклама и т.д.);

- люди;

- средства связи (радио, телевидение, телефон, пейджер и т.д.);

- документы (официальные, деловые, личные и т.д.);

- электронные, магнитные и другие носители, пригодные для автоматической обработки данных.

1.5. Средства защиты информации

Принято различать следующие средства защиты:

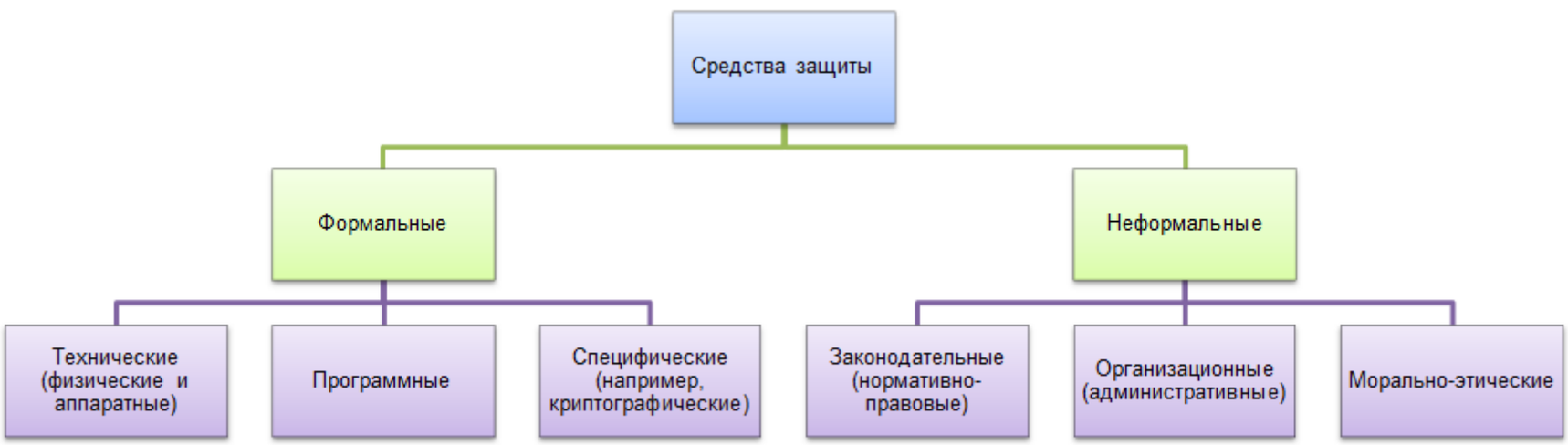


Рис.1.1. Классификация средств защиты

I. Формальные средства защиты – выполняют защитные функции строго по заранее предусмотренной процедуре без участия

человека.

Физические средства - механические, электрические, электромеханические, электронные, электронно-механические и тому подобные устройства и системы, которые функционируют автономно от информационных систем, создавая различного рода препятствия на пути дестабилизирующих факторов (замок на двери, жалюзи, забор, экраны).

Аппаратные средства - механические, электрические, электромеханические, электронные, электронно-механические, оптические, лазерные, радиолокационные и тому подобные устройства, встраиваемые в информационных системах или сопрягаемые с ней специально для решения задач защиты информации.

Программные средства - пакеты программ, отдельные программы или их части, используемые для решения задач защиты информации. Программные средства не требуют специальной аппаратуры, однако они ведут к снижению производительности информационных систем, требуют выделения под их нужды определенного объема ресурсов и т.п.

К **специфическим средствам** защиты информации относятся криптографические методы. В информационных системах криптографические средства защиты информации могут использоваться как для защиты обрабатываемой информации в компонентах системы, так и для защиты информации, передаваемой по каналам связи. Само преобразование информации может осуществляться аппаратными или программными средствами, с помощью механических устройств, вручную и т.д.

II. Неформальные средства защиты – регламентируют деятельность человека.

Законодательные средства – законы и другие нормативно-правовые акты, с помощью которых регламентируются правила использования, обработки и передачи информации ограниченного доступа и устанавливаются меры ответственности за нарушение этих правил. Распространяются на всех субъектов информационных отношений. В настоящее время отношения в сфере информационной безопасности регулируются более чем 80 законами и нормативными документами, иногда достаточно противоречивыми.

Организационные средства - организационно-технические и организационно-правовые мероприятия, осуществляемые в течение всего жизненного цикла защищаемой информационной системы (строительство помещений, проектирование информационных систем, монтаж и наладка оборудования, испытания и эксплуатация информационных систем). Другими словами – это средства уровня организации, регламентирующие перечень лиц, оборудования, материалов и т.д., имеющих отношение к информационным системам, а также режимов их работы и использования. К организационным мерам также относят сертификацию информационных систем или их элементов, аттестацию объектов и субъектов на выполнение требований обеспечения безопасности и т.д.

Морально-этические средства - сложившиеся в обществе или в данном коллективе моральные нормы или этические правила, соблюдение которых способствует защите информации, а нарушение приравнивается к несоблюдению правил поведения в обществе или коллективе, ведет к потере престижа и авторитета. Наиболее показательные пример – кодекс профессионального поведения членов Ассоциации пользователей ЭВМ США.

1.6. Способы передачи конфиденциальной информации на расстоянии

Способов передачи конфиденциальной информации на расстоянии существует множество, среди которых можно выделить три основных направления [9].

1. Создать абсолютно надежный, недоступный для других канал связи между абонентами.

2. Использовать общедоступный канал связи, но скрыть сам факт передачи информации.

3. Использовать общедоступный канал связи, но передавать по нему нужную информацию в таком преобразованном виде, чтобы восстановить ее мог только адресат.

Проанализируем эти возможности.

1. С древних времен практиковалась охрана документа (носителя информации) физическими лицами, передача его специальным курьером (человеком (дипломатом) или животным (голубиная почта)) и т.д. Но, документ можно выкрасть, курьера можно перехватить, подкупить, в конце концов, убить. В настоящий момент для реализации данного механизма защиты используются современные телекоммуникационные каналы связи. Однако следует заметить, что данный подход требует значительных

капитальных вложений. При современном уровне развития науки и техники сделать такой канал связи между удаленными абонентами для многократной передачи больших объемов информации практически нереально.

2. Разработкой средств и методов скрытия факта передачи сообщения занимается **стеганография**. Первые следы стеганографических методов теряются в глубокой древности. Так, в трудах древнегреческого историка Геродота встречается описание двух методов сокрытия информации: на обритую голову раба записывалось необходимое сообщение, а когда его волосы отрастали, он отправлялся к адресату, который вновь брил его голову и считывал доставленное сообщение. Второй способ заключался в следующем: сообщение наносилось на деревянную дощечку, а потом она покрывалась воском, и, тем самым, не вызывала никаких подозрений. Потом воск соскабливался, и сообщение становилось видимым. В настоящий момент стеганографические методы в совокупности с криптографическими нашли широкое применение в целях сокрытия и передачи конфиденциальной информации.

3. Разработкой методов преобразования информации с целью ее защиты от несанкционированного прочтения занимается **криптография**.

Вопросы для самопроверки

1. Дайте определение понятиям: «информация», «информационная безопасность», «защита информации», «информационная угроза».

2. Дайте характеристику основным составляющим информационной безопасности.

3. Перечислите основные объекты защиты.

4. Дайте характеристику понятиям «государственная тайна», «конфиденциальная информация» и «персональные данные».

5. Дайте характеристику средствам защиты информации.

6. Перечислите способы тайной передачи информации на расстоянии.

2. ИСТОРИЯ КРИПТОГРАФИИ

2.1. Введение

В истории развития криптографии можно выделить три этапа [11]:

- наивная криптография;

- формальная криптография;

- математическая криптография.

2.2. Наивная криптография

О важности сохранения информации в тайне знали уже в древние времена, когда с появлением письменности появилась и опасность прочтения ее нежелательными лицами. Более того, первоначально письменность сама по себе была криптографической системой, так как в древних обществах ею владели только избранные. С широким распространением письменности криптография стала формироваться как самостоятельная наука. В документах древних цивилизаций - Индии, Египта, Месопотамии - есть сведения о системах и способах составления шифрованных писем.[9].

Для наивной криптографии (до начала XVI в.) характерно использование любых, обычно примитивных, способов запутывания противника относительно содержания передаваемых сообщений. На начальном этапе для защиты информации использовались методы кодирования и стеганографии, которые родственны, но не тождественны криптографии. Шифровальные системы сводились к использованию перестановки или замены букв на различные символы (другие буквы, знаки, рисунки, числа и т.п.). Одни и те же способы шифрования использовались повторно, ключи были короткими, использовались примитивные способы преобразования исходной информации в зашифрованное сообщение. Это позволяло, однажды установив алгоритм шифрования, быстро расшифровывать сообщения.

[43] Точное время возникновения этих способов обмена тайной информацией теряется в глубине веков, и установить его невозможно. Историки полагают, что первые протокриптографические приемы появились в Древнем Египте около 4 тыс. лет назад. Писцы, составлявшие жизнеописания правителей, стремились придать стандартным иероглифам необычный вид на монументах и гробницах, чтобы сообщить надписям менее обыденный и более почтительный стиль. Жрецы пользовались этим же приемом при переписывании религиозных текстов, чтобы те выглядели для мирян загадочнее и внушительнее. Такие «переводы» становились все менее понятными простому люду, который в результате оказывался во все большей зависимости от жрецов.

По мере развития египетской цивилизации ширилось использование иероглифов. С увеличением количества надписей, высекавшихся на стенах храмов, люди теряли к ним интерес. Египтологи считают, что писцы тогда стали еще больше видоизменять некоторые знаки в стремлении пробудить любопытство и привлечь внимание населения. Эти модификации никоим образом не были кодами или шифрами, но они заключали в себе два основных принципа криптологии, а именно: изменение письма и сокрытие его смысла. Бесспорных доказательств, указывающих на широкое использование модификаций иероглифов для сокрытия дипломатических, торговых или военных планов в Древнем Египте, нет.

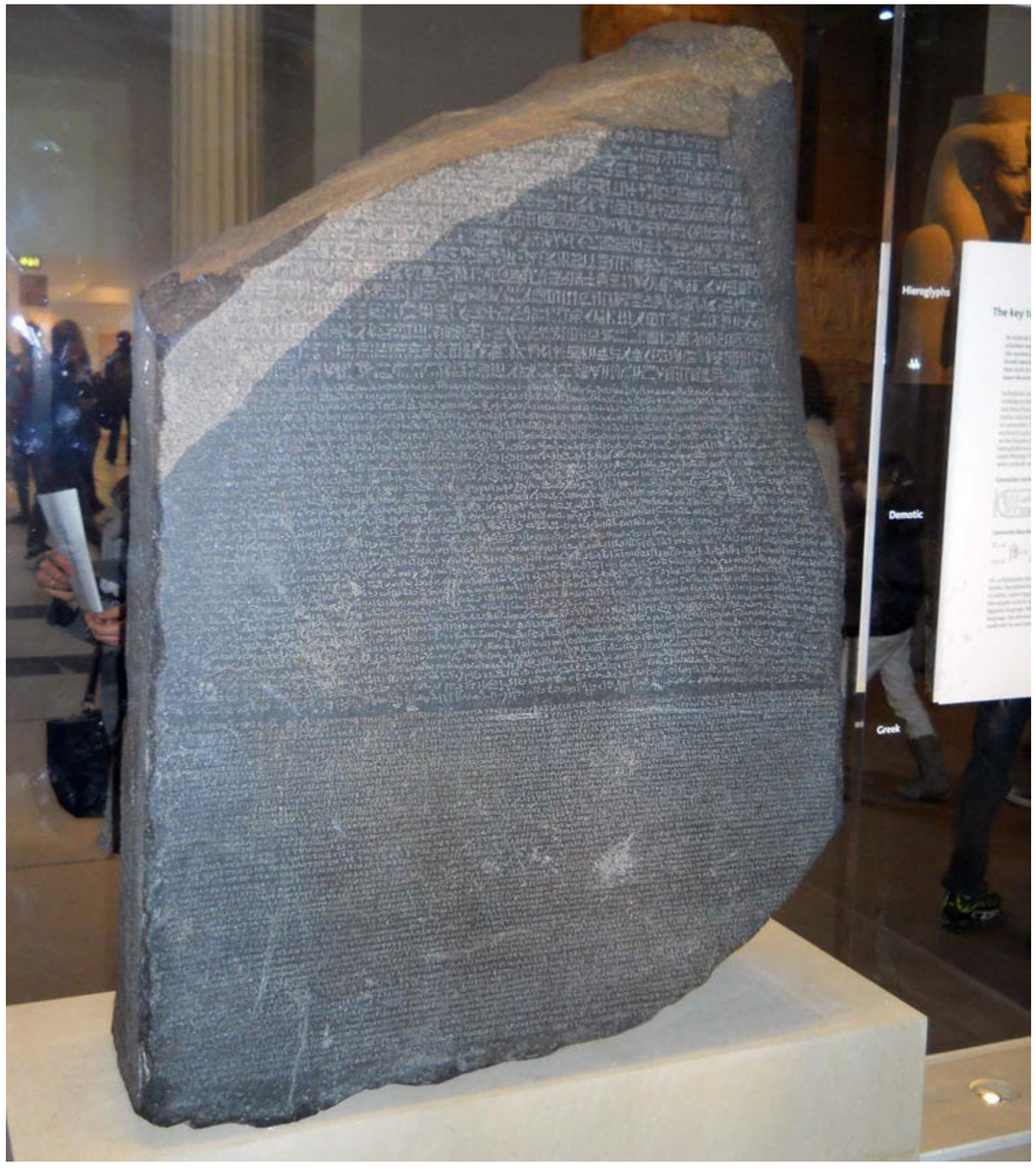


Рис.2.1. Розеттский камень[1]

Более явные криптологические примеры дошли до нас от цивилизаций Месопотамии — от вавилонян, ассирийцев, халдеев, использовавших особую систему письма — клинопись. В 1500 г. до н. э. на глиняной табличке был записан тщательно охраняемый рецепт глазури для гончарных изделий. Знаки, определяющие необходимые ингредиенты, были намеренно перемешаны. Таким образом, мы имеем право утверждать, что эта табличка является самой ранней известной секретной записью.

Примерно с 500 г. до н. э. в Индии также широко применялись секретные записи, в частности в донесениях шпионов и текстах, предположительно использовавшихся Буддой. Методы засекречивания включали в себя фонетическую замену, когда согласные и гласные менялись местами, использование перевернутых букв и запись текста под случайными углами. Различные индийские трактаты, ярким примером которых является «Артхашастра» (около 321—300 гг. до н. э.), показывают, что индийцы были хорошо знакомы со способами сокрытия информации. Согласно «Камасутре», классическому произведению об эротике и других видах наслаждений, написанному Ватсьяяной около IV в. до н.э., владение приемами составления секретного письма (наряду с музыкой, кулинарией и шахматами) является одним из 64 искусств, которыми должны владеть женщины.

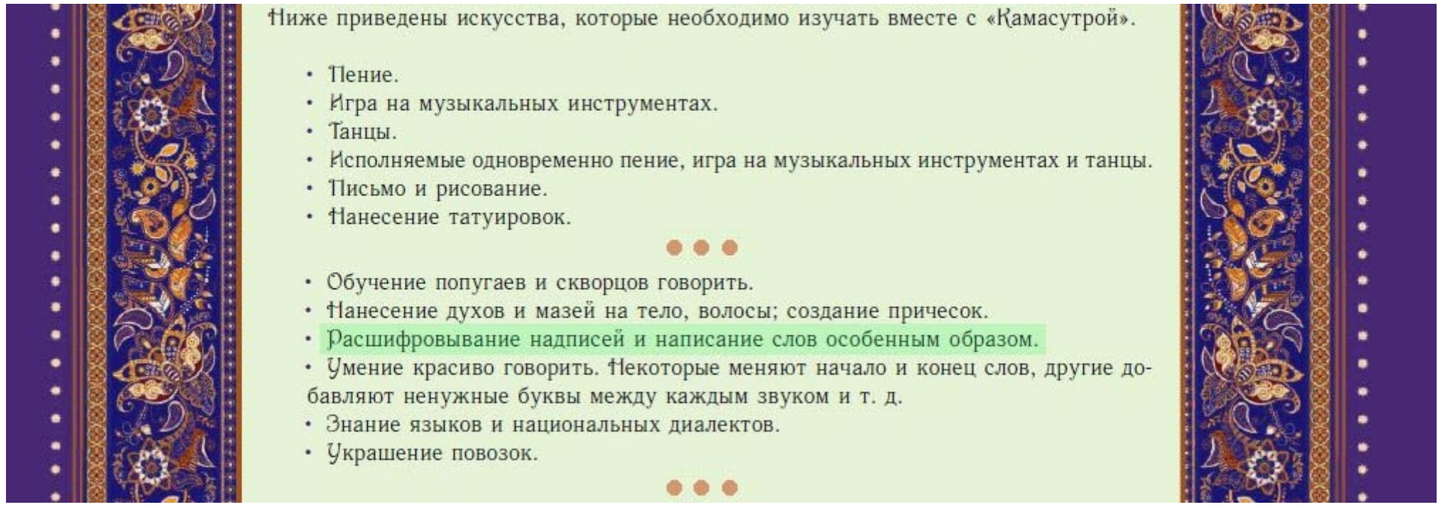


Рис.2.2. Фрагмент «Камасутры»

Одним из первых, документально зафиксированных примеров является **шифр Цезаря**, состоящий в замене каждой буквы исходного текста на другую, отстоящую от нее в алфавите на определенное число позиций. Другой шифр, **полибианский квадрат**, авторство которого приписывается греческому писателю Полибию, является шифром простой однозначной замены. В квадрат выписывались буквы алфавита (для греческого алфавита размер составлял 5x5). Каждая буква исходного текста заменялась на пару цифр – номер строки и столбца на пересечении которых стояла шифруемая буква.

С VIII века н. э. развитие криптографии происходит в основном в арабских странах. Считается, что арабский филолог Халиль аль-Фарахиди первым обратил внимание на возможность использования стандартных фраз открытого текста для дешифрования. Он предположил, что первыми словами в письме на греческом языке византийскому императору будут «Во имя Аллаха», что позволило ему прочитать оставшуюся часть сообщения. Позже он написал книгу с описанием данного метода — «Китаб аль-Муамма» («Книга тайного языка») [17]. В 855 г. выходит «Книга о большом стремлении человека разгадать загадки древней письменности» арабского учёного Абу Бакр Ахмед ибн Али Ибн Вахшия ан-Набати, одна из первых книг о криптографии с описаниями нескольких шифров, в том числе с применением нескольких алфавитов. Также к IX веку относится первое известное упоминание о частотном криптоанализе — в книге Аль-Кинди «Манускрипт о дешифровке криптографических сообщений». В 1412 г. выходит 14-томная энциклопедия Шихаба ал-Калкашанди «Субх ал-Ааша», один из разделов которой «Относительно сокрытия в буквах тайных сообщений» содержал описание семи шифров замены и перестановки, частотного криптоанализа, а также таблицы частоты появления букв в арабском языке на основе текста Корана. В словарь криптологии арабы внесли такие понятия как алгоритм и шифр.

Рис.2.3. Фрагмент первой страницы «Манускрипта о дешифровке криптографических сообщений» Аль-Кинди

В древние времена широкое применение нашли различные простейшие криптографические устройства.

Греческим поэтом Архилохом, жившим в VII веке до н.э., упоминается устройство под названием **сцитала** (греч. σκυτάλη - жезл). Достоверно известно, что сцитала использовалась в войне Спарты против Афин в конце V века до н. э. Оно представляет собой цилиндр (иногда жезл командующего) и узкую полоску пергамента, обматывавшуюся вокруг него по спирали, на которой в свою очередь писалось сообщение [17].

Рис.2.4. Сцитала

Шифруемый текст писался на пергаментной ленте по длине палочки, после того как длина палочки оказывалась исчерпанной, она поворачивалась и текст писался далее, пока либо не заканчивался текст, либо не исписывалась вся пергаментная лента. В последнем случае использовался очередной кусок пергаментной ленты. Для расшифровки адресат использовал палочку такого же диаметра, на которую он наматывал пергамент, чтобы прочитать сообщение. Античные греки и спартанцы в частности, использовали этот шифр для связи во время военных кампаний. Однако такой шифр может быть легко взломан. Например, метод взлома сциталы был предложен ещё Аристотелем. Он состоит в том, что не зная точного диаметра палочки, можно использовать конус, имеющий переменный диаметр и перемещать пергамент с сообщением по его длине до тех пор, пока текст не начнёт читаться - таким образом дешифруется диаметр сциталы.

Другим широко известным криптографическим устройством защиты информации был **«диск Энея»** - инструмент для защиты информации, придуманный Энеем Тактиком в IV веке до н.э. Устройство представляло собой диск диаметром 13-15 см и толщиной 1-2 см с проделанными в нём отверстиями, количество которых равнялось числу букв в алфавите. Каждому отверстию ставилась в соответствие конкретная буква. В центре диска находилась катушка с намотанной на неё ниткой [17].

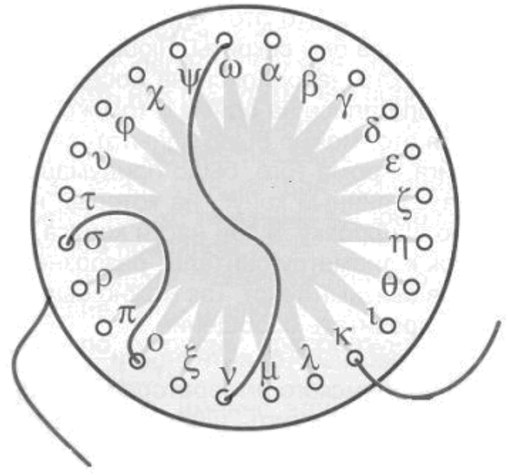


Рис.2.5. Диск Энея

Механизм шифрования был очень прост. Для того, чтобы зашифровать послание, необходимо было поочерёдно протягивать свободный конец нити через отверстия обозначающие буквы исходного не зашифрованного сообщения. В итоге, сам диск, с продетой в его отверстия ниткой, и являлся зашифрованным посланием. Получатель сообщения последовательно вытягивал нить из каждого отверстия, тем самым получал последовательность букв. Но эта последовательность являлась обратной по отношению к исходному сообщению, то есть он читал сообщение наоборот. Зашифрованное сообщение было доступно к прочтению любому, кто смог завладеть диском. Так как сообщение предавали обычные гонцы, а не воины, Эней предусмотрел возможность быстрого уничтожения передаваемой информации. Для этого было достаточно вытянуть всю нить за один из её концов, либо сломать диск, просто наступив на него. На самом деле «диск Энея» нельзя назвать настоящим криптографическим инструментом, поскольку прочитать сообщение мог любой желающий. Но это устройство стало прародителем первого по истине криптографического инструмента, изобретение которого также принадлежит Энею.

Прибор получил название **«Линейка Энея»**. Она представляла собой устройство, имеющее отверстия, количество которых равнялось количеству букв алфавита. Каждое отверстие обозначалось своей буквой; буквы по отверстиям располагались в произвольном порядке. К линейке была прикреплена катушка с намотанной на неё ниткой. Рядом с катушкой имелась прорезь. При шифровании нить протягивалась через прорезь, а затем через отверстие, соответствующее первой букве шифруемого текста, при этом на нити завязывался узелок в месте прохождения её через отверстие; затем нить возвращалась в прорезь и аналогично зашифровывалась вторая буква текста и т. д. После окончания шифрования нить извлекалась и передавалась получателю сообщения. Получатель, имея идентичную линейку, протягивал нить через прорезь до отверстий, определяемых узлами, и восстанавливал исходный текст по буквам отверстий. Такой шифр является одним из примеров шифра замены: когда буквы заменяются на расстояния между узелками с учетом прохождения через прорезь. Ключом шифра являлся порядок расположения букв по отверстиям в линейке. Посторонний, получивший нить (даже имея линейку, но без нанесенных на ней букв), не сможет прочитать передаваемое сообщение.

[1]**Розеттский камень** - плита из гранодиорита, найденная в 1799 г. в Египте возле небольшого города Розетта (теперь Рашид), недалеко от Александрии, с выбитыми на ней тремя идентичными по смыслу текстами, в том числе двумя на древнеегипетском языке и одним на древнегреческом языке. Древнегреческий был хорошо известен лингвистам, и сопоставление трёх текстов послужило отправной точкой для расшифровки египетских иероглифов.

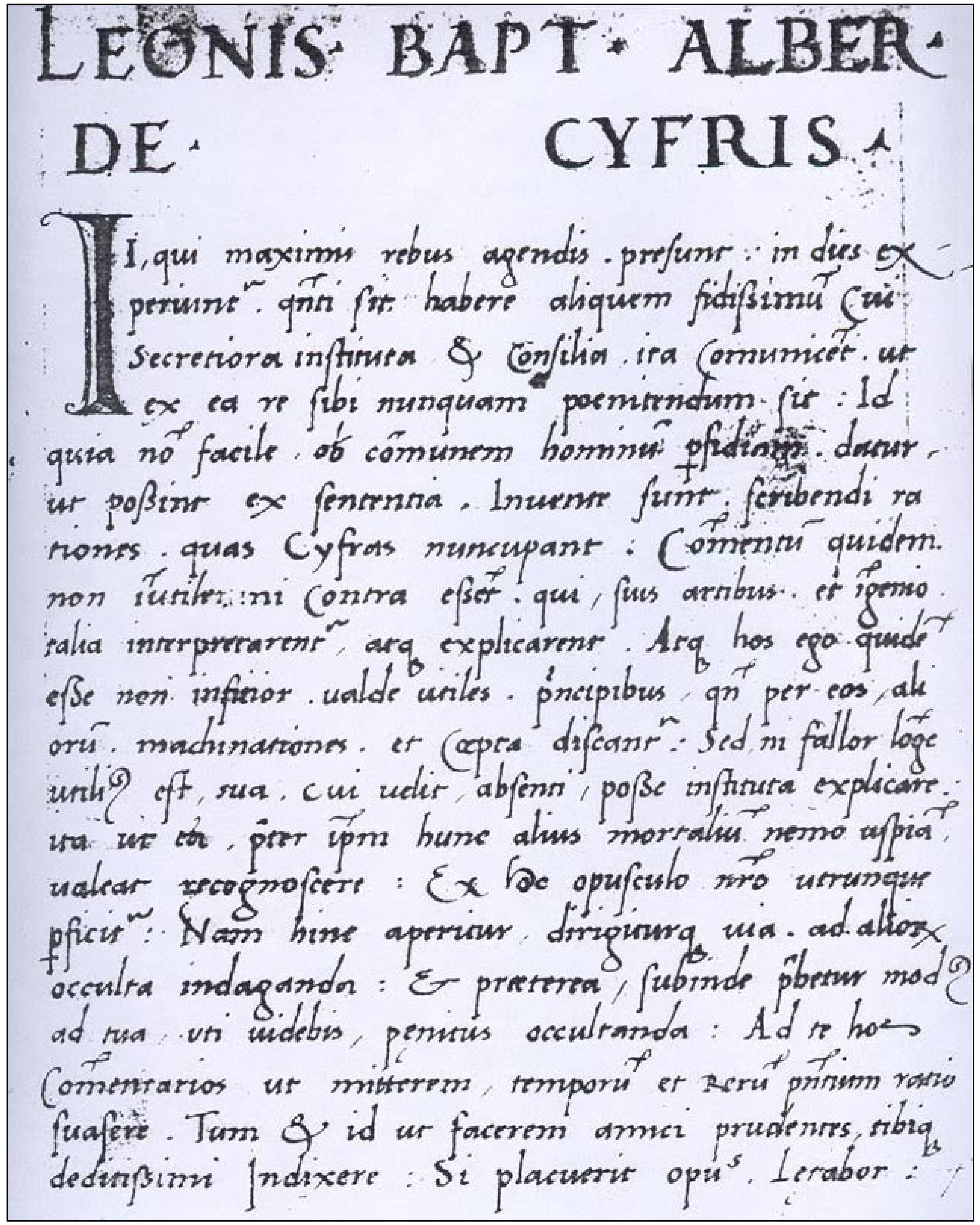
2.3. Формальная криптография

Этап формальной криптографии (конец XV – начало XX вв.) 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 надежные способы защиты информации.

К концу XIV в. между итальянскими городами-государствами в переписке уже применялись **«номенклаторы»**[2] (лат. nomen — «имя» и calator — «раб», «слуга»). Они состояли из кодовых обозначений для слогов, слов и имен, а также алфавитов шифрозамен. Вплоть до XIX в. номенклаторы оставались самой широко используемой системой сокрытия содержания сообщений [43].

Симеоне де Крема (Simeone de Crema) был первым (1401 г.), кто использовал таблицы омофонов для сокрытия частоты появления гласных в тексте при помощи более чем одной шифрозамены (**шифры многозначной замены**).

Отцом западной криптографии называют учёного эпохи Возрождения Леона Баттисту Альберти. Изучив методы вскрытия использовавшихся в Европе моноалфавитных шифров (**шифров однозначной замены**), он попытался создать шифр, который был бы устойчив к частотному криптоанализу. Его «Трактат о шифрах» был представлен в папскую канцелярию в 1466 г. и считается первой научной работой по криптографии. Он предложил вместо единственного секретного алфавита, как в моноалфавитных шифрах, использовать два или более, переключаясь между ними по какому-либо правилу (полиалфавитные шифры).



LEONIS BAPT. ALBER.
DE CYFRIS.

Ii, qui maximis rebus agendis presunt: in dies ex-
periuntur quanti sit habere aliquem fidissimum cui
secretiora instituta & consilia ita comunicet. ut
ex ea re sibi nunquam poenitendum sit: Id
quia non facile ob comunem hominum pfidiam datur,
ut possint ex sententia, Inventae sunt scribendi ra-
tiones quas Cyfras nuncupant: Comentum quidem
non inutile ... contra esse: qui suis artibus et ingenio
talia interpretarent, atq explicarent. Atq hos ego quide
esse non inficior valde utiles principibus qui per eos ali-
orum machinationes et coepta discant. Sed ni fallor loge
utilius est, sua, cui velit, absenti posse instituta explicare.
ita ut ea, preter ipsum hunc alius mortalium nemo usquam
valeat recognoscere: Ex hoc opusculo nostro utrunque
pficitur: Nam hinc aperitur dirigiturq via ad aliorum
occulta indaganda: & preterea subinde pbetur modus
ad tua, uti videbis, penitus occultanda: Ad te hos
Comentarios ut mitterem, temporum et rerum presentium ratio
suasere. Tum & id ut facerem amici prudentes, tibiq
deditissimi Indixere: Si placuerit opus, Letabor:

Рис.2.6. Фрагмент первой страницы «Трактата о шифрах» Леона Баттисто Альберти

Однако флорентийский учёный так и не смог оформить своё открытие в полную работающую систему, что было сделано уже его последователями (Блез Вижинер).

Другой работой, в которой обобщены и сформулированы известные на тот момент алгоритмы шифрования, является труд «Полиграфия» (1518 г.) немецкого аббата Иоганна Трисемуса (Тритемия). Он же первым заметил, что шифровать можно и по две буквы за раз - биграммами (хотя первый **биграммный шифр** Playfair был предложен лишь в XIX веке).

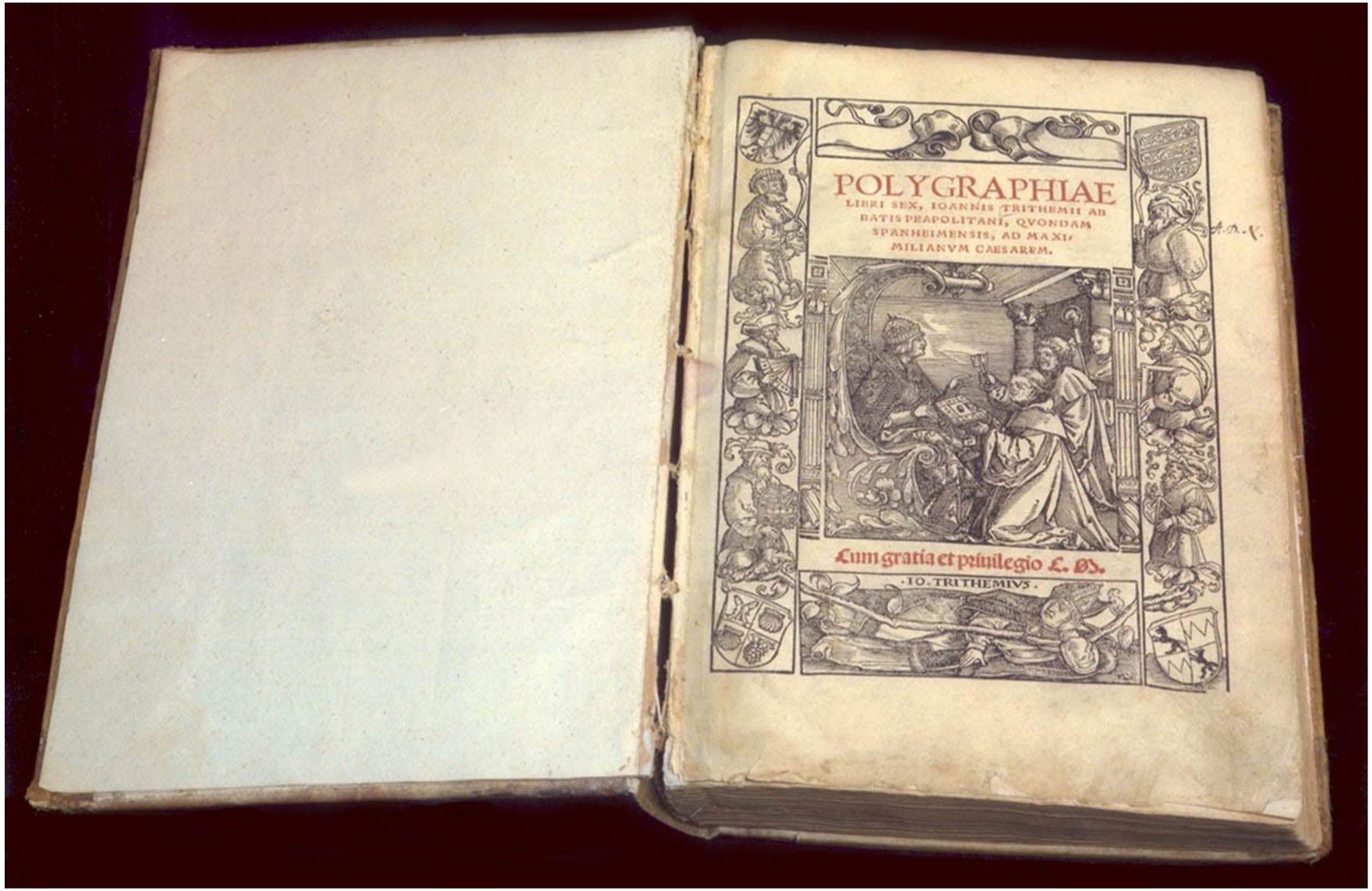


Рис.2.7. Первый печатный труд по криптографии - «Полиграфия» Иоганна Трисемуса [Национальный музей криптографии, США]

В 1550 г. итальянский математик Джероламо Кардано, состоящий на службе у папы римского, предложил новую технику шифрования - решётку Кардано. Этот способ сочетал в себе как стеганографию (искусство скрытого письма), так и криптографию. Затруднение составляло даже понять, что сообщение содержит зашифрованный текст, а расшифровать его, не имея ключа (решётки) в то время было практически невозможно. Решётку Кардано считают первым **транспозиционным шифром**, или, как ещё называют, геометрическим шифром, основанным на положении букв в шифротексте.

Значительный толчок криптографии дало изобретение телеграфа. Сама передача данных перестала быть секретной, и сообщение, в теории, мог перехватить кто угодно. Интерес к криптографии возрос, в том числе, и среди простого населения, в результате чего многие попытались создать индивидуальные системы шифрования. Преимущество телеграфа было явным и на поле боя, где командующий должен был отдавать немедленные приказы на поле сражения, а также получать информацию с мест событий. Это послужило толчком к развитию полевых шифров.

В 1863 г. Фридрих Касиски (англ. Friedrich Kasiski) опубликовал метод, впоследствии названный его именем, позволявшим быстро и эффективно вскрывать практически любые шифры того времени, в т.ч. полиалфавитные. Метод состоял из двух частей - определение периода шифра и дешифровка текста с использованием частотного криптоанализа.

Die Geheimschriften

und

die Dechiffrir-Kunst.

Mit besonderer Berücksichtigung

der deutschen und der französischen Sprache

von

F. W. Kasiski,

Major z. D.

Berlin, 1863.

Druck und Verlag von E. S. Mittler und Sohn.

(Zimmerstraße 84. 85.)

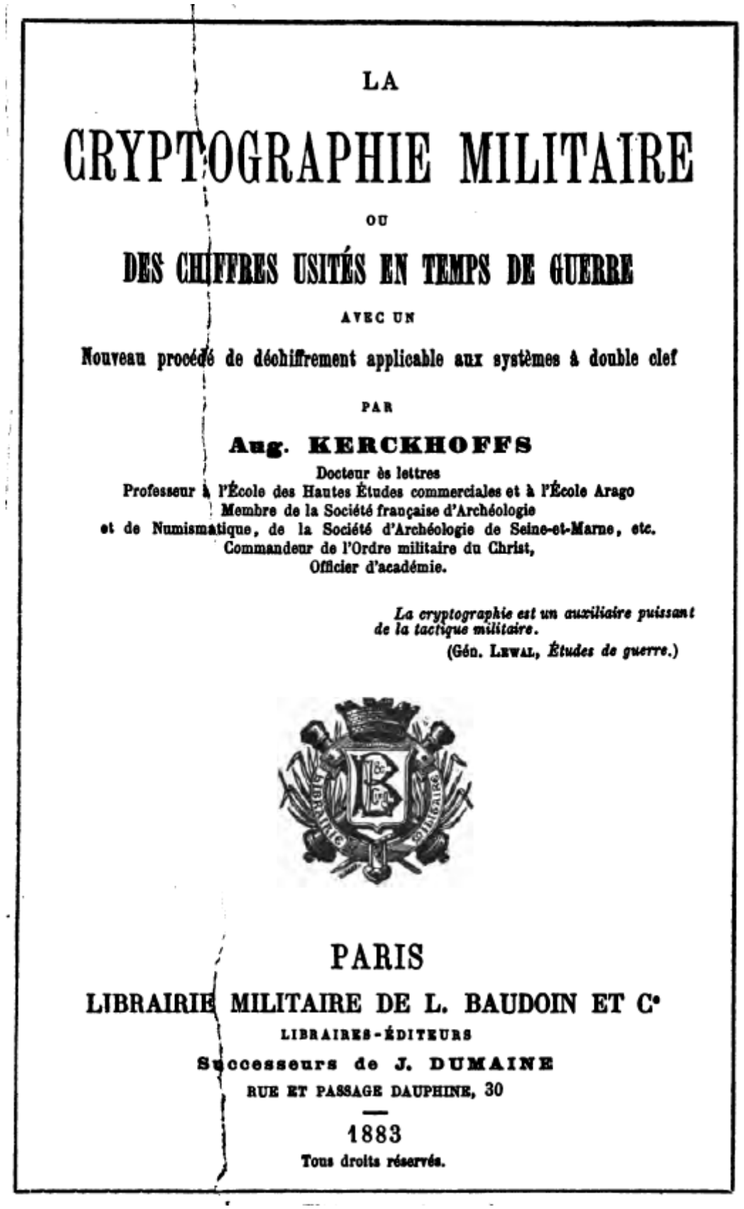
Рис.2.8. Титульный лист «Тайнописи и искусства дешифрования» Фридриха Касиски

В 1883 г. голландец Огюст Керкгоффс[3] опубликовал труд под названием «Военная криптография» (фр. «La Cryptographie Militaire»). В нём он описал шесть требований, которым должна удовлетворять защищённая система. Хотя к некоторым из них стоит относиться с подозрением, стоит отметить труд за саму попытку:

1. шифр должен быть физически, если не математически, невскрываемым;
2. система не должна требовать секретности, на случай, если она попадёт в руки врага;
3. ключ должен быть простым, храниться в памяти без записи на бумаге, а также легко изменяемым по желанию корреспондентов;
4. зашифрованный текст должен (без проблем) передаваться по телеграфу;
5. аппарат для шифрования должен быть легко переносимым, работа с ним не должна требовать помощи нескольких лиц;

6. аппарат для шифрования должен быть относительно прост в использовании, не требовать значительных умственных усилий или соблюдения большого количества правил.

Им же был сформулирован известный **«принцип Керкгоффса»** - правило разработки криптографических систем, согласно которому в засекреченном виде держится только определённый набор параметров алгоритма, называемый ключом, а сам алгоритм шифрования должен быть открытым. Другими словами, при оценке надёжности шифрования необходимо предполагать, что противник знает об используемой системе шифрования всё, кроме применяемых ключей.



LA

CRYPTOGRAPHIE MILITAIRE

OU

DES CHIFFRES USITÉS EN TEMPS DE GUERRE

AVEC UN

Nouveau procédé de déchiffrement applicable aux systèmes à double clef

PAR

Aug. KERCKHOFFS

Docteur ès lettres
Professeur à l'École des Hautes Études commerciales et à l'École Arago
Membre de la Société française d'Archéologie
et de Numismatique, de la Société d'Archéologie de Seine-et-Marne, etc.
Commandeur de l'Ordre militaire du Christ,
Officier d'académie.

La cryptographie est un auxiliaire puissant de la tactique militaire.
(Gén. LEWAL, *Études de guerre.*)

PARIS

LIBRAIRIE MILITAIRE DE L. BAUDOIN ET Cᵉ

LIBRAIRES-ÉDITEURS

Successeurs de J. DUMAINE

RUE ET PASSAGE DAUPHINE, 30

1883

Tous droits réservés.

Рис.2.9. Титульный лист брошюры «Военная криптография» Огюста Керкгоффса

В 1920 г. вышла монография американского криптографа российского происхождения Уильяма Ф. Фридмана «Индекс совпадения и его применение в криптографии» (англ. «Index of Coincidence and Its Applications in Cryptography»). Работа вышла в открытой печати, несмотря на то, что была выполнена в рамках военного заказа. Двумя годами позже Фридман ввёл в научный обиход термины **криптология** и **криптоанализ** [13].

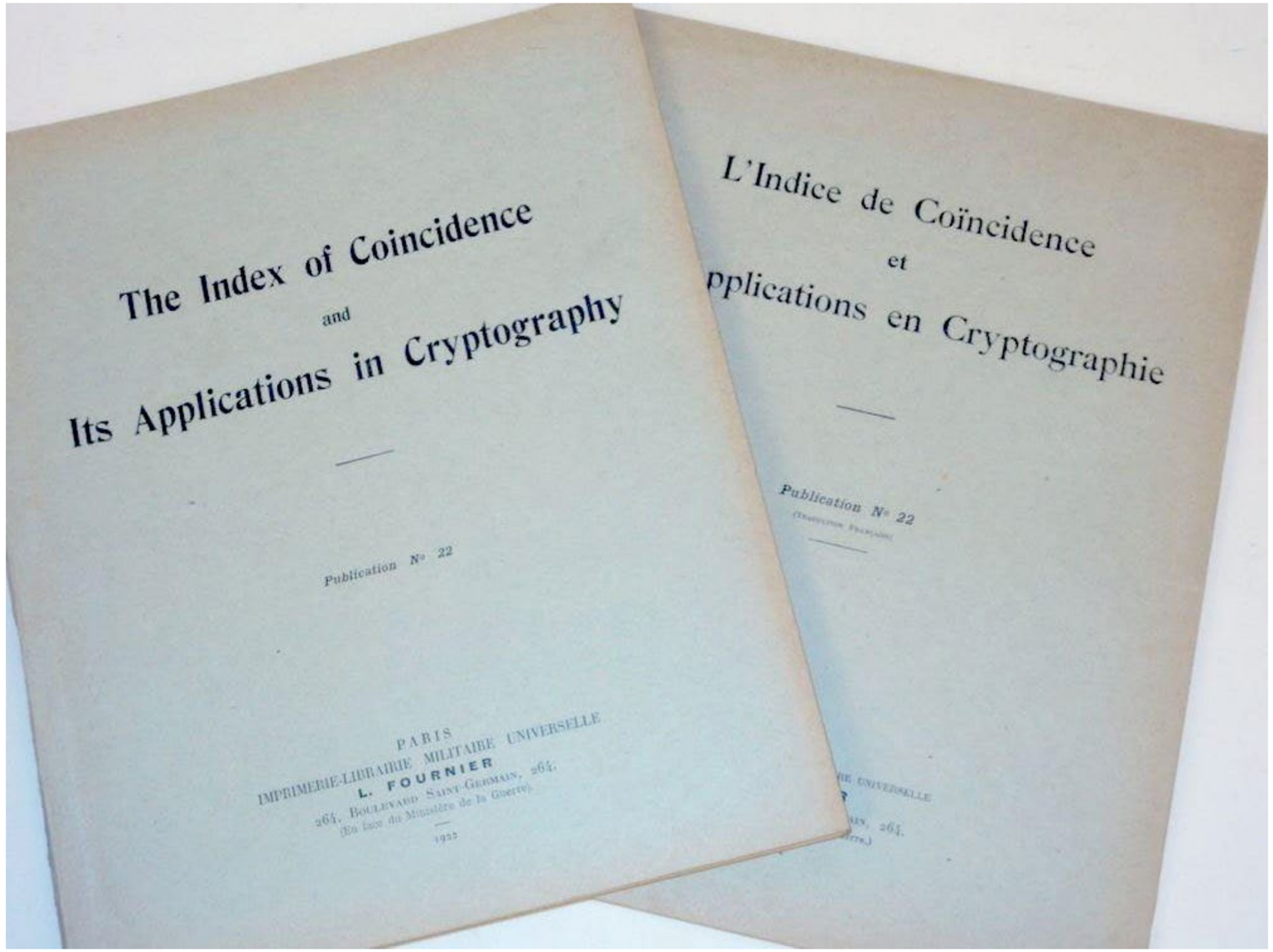


Рис.2.10. «Индекс совпадения и его применение в криптографии» Уильяма Фридмана

Криптография оказала влияние и на **литературу**. Упоминания о криптографии встречаются ещё во времена Гомера и Геродота, хотя они описывали искусство шифрования в контексте различных исторических событий. Первым вымышленным упоминанием о криптографии можно считать роман «Гаргантюа и Пантагрюэль» французского писателя XVI века Франсуа Рабле, в одной из глав которого описываются попытки чтения зашифрованных сообщений. Упоминание встречается и в «Генрихе V» Шекспира. Впервые как центральный элемент художественного произведения криптография используется в рассказе «Золотой жук» Эдгара Аллана По 1843 г. В нём писатель не только показывает способ раскрытия шифра, но и результат, к которому может привести подобная деятельность - нахождение спрятанного сокровища. Одним из лучших описаний применения криптографии является рассказ 1903 г. Артура Конан Дойля «Пляшущие человечки». В рассказе великий сыщик Шерлок Холмс сталкивается с разновидностью шифра, который не только прячет смысл написанного, но, используя символы, похожие на детские картинки, скрывает сам факт передачи секретного сообщения. В рассказе герой успешно применяет частотный анализ, а также предположения о структуре и содержании открытых сообщений для разгадывания шифра.

Перед началом Второй мировой войны ведущие мировые державы имели электромеханические шифрующие устройства, результат работы которых считался невскрываемым. Эти устройства делились на два типа - роторные машины и машины на цевочных дисках. К первому типу относят **«Энигму»**, использовавшуюся войсками Германии и её союзников, второго типа - американская **М-209**. В СССР производились оба типа машин.

Энигма 1
(Германия)

М-209
(США)

Коралл
(СССР)

Рис.2.11. Шифровальные устройства [en.wikipedia.org, www.cryptomuseum.com]

Успешные криптоатаки на подобного рода криптосистемы стали возможны только с появлением ЭВМ.

---

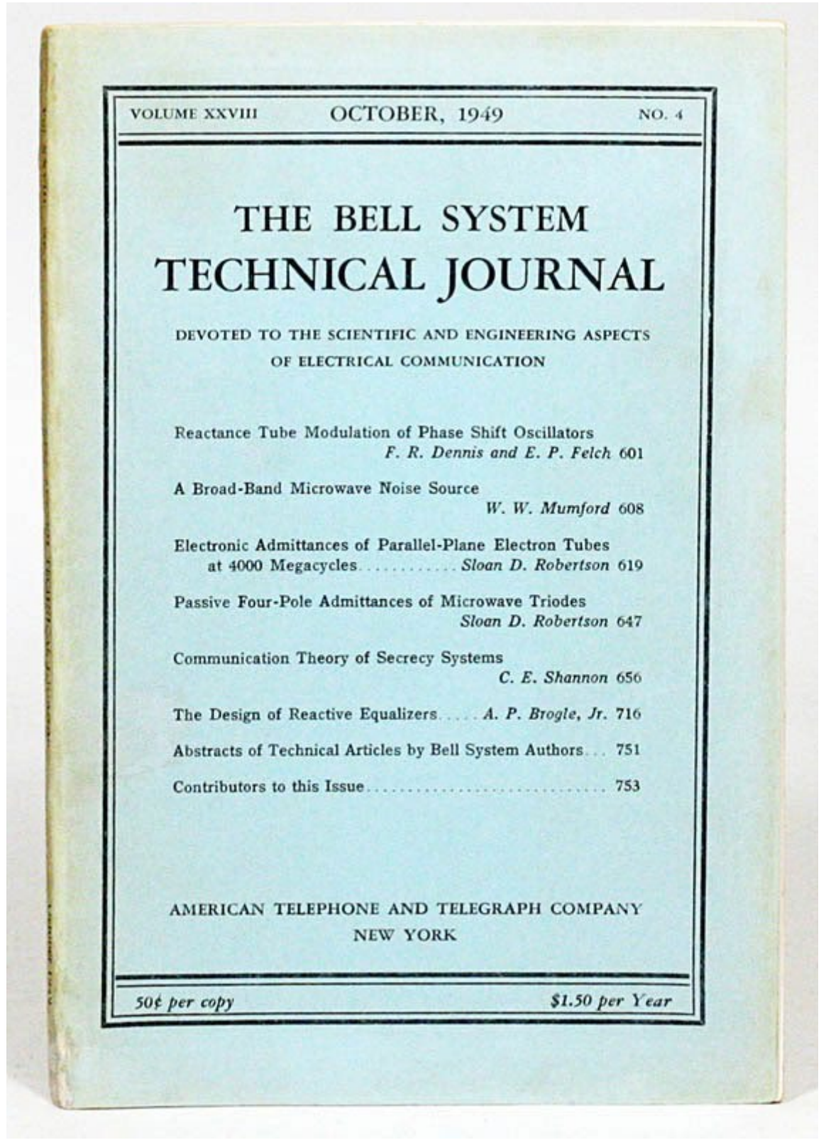
[2]Первоначально **номенклатором** назывался раб, который обязан был знать и называть своему господину имена граждан города и всех рабов в доме, а также провозглашать названия подаваемых кушаний.

[3]Огюст Керкгоффс (Auguste Kerckhoffs, 1835 - 1903 гг.) - голландский лингвист и криптограф, профессор Парижской высшей школы коммерции во второй половине XIX века. В русских источниках встречаются разные переводы фамилии - Керкхофф, Кирхгоф, Керкгоффс, Керхофс, Керкхоффс. Полное имя, полученное при крещении, - Жан Вильгельм Губерт Виктор Франсуа Александр Огюст Керкгоффс фон Ниувенгоф [17, 43].

2.4. Математическая криптография

После Первой мировой войны правительства стран засекретили все работы в области криптографии. К началу 1930-х годов окончательно сформировались разделы математики, являющиеся основой для будущей науки: общая алгебра, теория чисел, теория вероятностей и математическая статистика. К концу 1940-х годов построены первые программируемые счётные машины, заложены основы теории алгоритмов, кибернетики. Тем не менее, в период после Первой мировой войны и до конца 1940-х годов в открытой печати было опубликовано совсем немного работ и монографий, но и те отражали далеко не самое актуальное состояние дел. Наибольший прогресс в криптографии достигается в военных ведомствах.

Ключевой вехой в развитии криптографии является **фундаментальный труд Клода Шеннона** «Теория связи в секретных системах» (англ. Communication Theory of Secrecy Systems) - секретный доклад, представленный автором в 1945 г., и опубликованный им в «Bell System Technical Journal» в 1949 г. В этой работе, по мнению многих современных криптографов, был впервые показан подход к криптографии в целом как к математической науке.



VOLUME XXVIII OCTOBER, 1949 NO. 4

THE BELL SYSTEM
TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS
OF ELECTRICAL COMMUNICATION

AMERICAN TELEPHONE AND TELEGRAPH COMPANY
NEW YORK

50¢ per copy $1.50 per Year

Рис.2.12. Сборник со статьей «Теория связи в секретных системах» Клода Шеннона

В 1960-х годах начали появляться различные блочные шифры, которые обладали большей криптостойкостью по сравнению с результатом работы роторных машин. Однако они предполагали обязательное использование цифровых электронных устройств - ручные или полумеханические способы шифрования уже не использовались.

В 1967 г. выходит книга Дэвида Кана «Взломщики кодов». Хотя книга не содержала сколько-нибудь новых открытий, она подробно описывала имеющиеся на тот момент результаты в области криптографии, большой исторический материал, включая успешные случаи использования криптоанализа, а также некоторые сведения, которые правительство США полагало всё ещё секретными. Но главное - книга имела заметный коммерческий успех и познакомила с криптографией десятки тысяч людей. С этого момента начали понемногу появляться работы и в открытой печати.

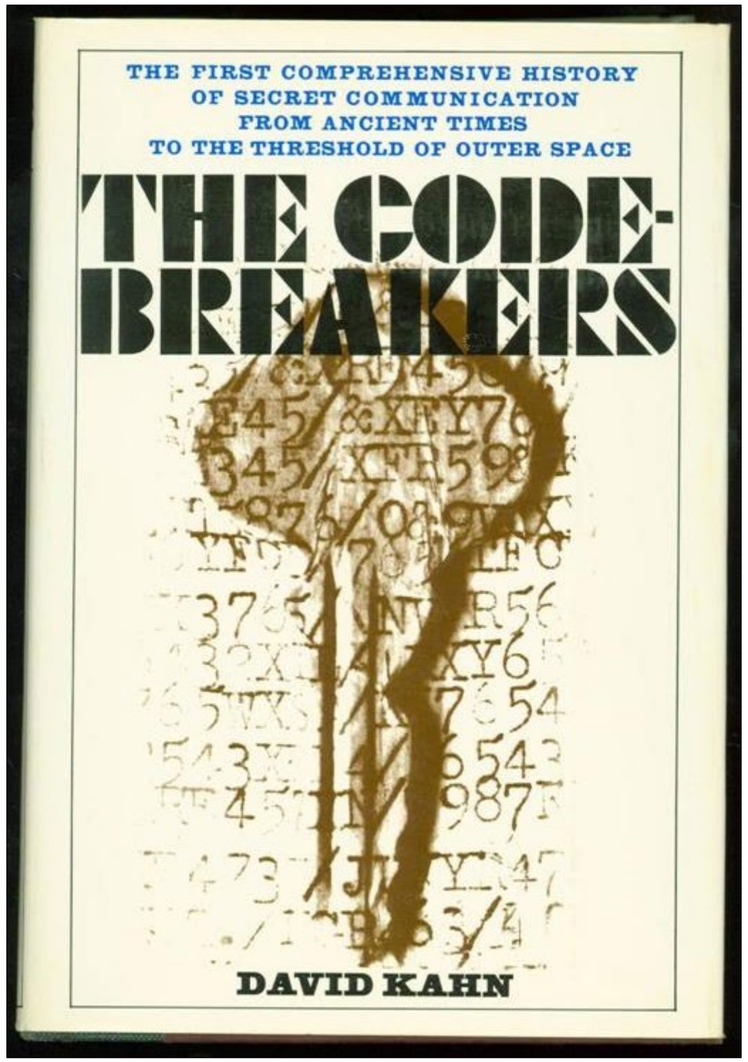


Рис.2.13. Обложка первого издания «Взломщиков кодов» Дэвида Кана

Примерно в это же время Хорст Фейстель переходит из Военно-воздушных сил США на работу в лабораторию корпорации IBM. Там он занимается разработкой новых методов в криптографии и разрабатывает **ячейку Фейстеля**, являющуюся основой многих современных шифров, в том числе шифра Lucifer, ставшего прообразом шифра DES – бывшего стандарта шифрования США, первого в мире открытого государственного стандарта на шифрование данных. На основе ячейки Фейстеля были созданы и другие блочные шифры, в том числе TEA (1994 г.), Twofish (1998 г.), IDEA (2000 г.), а также бывший (ГОСТ 28147-89) и действующий (ГОСТ 34.12-2015) российские стандарты шифрования.

В 1976 г. публикуется работа Уитфилда Диффи и Мартина Хеллмана «Новые направления в криптографии» (англ. «New Directions in Cryptography»). Данная работа открыла новую область в криптографии, теперь известную как **криптография с открытым ключом**. Также в работе содержалось описание алгоритма Диффи - Хеллмана - Меркла, позволявшего сторонам сгенерировать общий секретный ключ, используя открытый канал связи.

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

***Abstract*—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.**

I. INTRODUCTION

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The development of computer controlled communication networks promises effortless and inexpensive contact

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channels without compromising the security of the system. In a *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enciphered in such a way that only the intended receiver is able to decipher it. As such, a public key cryptosystem is a

Рис.2.14. Начало статьи «Новые направления в криптографии» Уитфилда Диффи и Мартина Хеллмана

Хотя работа Диффи-Хеллмана создала большой теоретический задел для открытой криптографии, первой реальной криптосистемой с открытым ключом считают алгоритм RSA (названный по имени авторов - Рон Ривест (R. Rivest), Ади Шамир (A. Shamir) и Леонард Адлеман (L. Adleman)). Опубликованная в августе 1977 г., работа позволила сторонам обмениваться секретной информацией, не имея заранее выбранного секретного ключа. Стоит отметить, что и алгоритм Диффи - Хеллмана - Меркла, и RSA были впервые открыты в английских спецслужбах, но не были ни опубликованы, ни запатентованы из-за секретности.

В России для шифрования с открытым ключом стандарт отсутствует, однако для электронной цифровой подписи (органически связанной с шифрованием с открытым ключом) принят ГОСТ Р 34.10-2001 (ГОСТ Р 34.10-2012), использующий криптографию на эллиптических кривых.

Создание асимметричных криптосистем подтолкнуло математиков и криптоаналитиков к изучению способов факторизации, дискретного логарифмирования, операций над эллиптическими кривыми в конечном поле и т.д.

Относительно новым методом является **вероятностное шифрование**. Вероятностное шифрование предложили Шафи Гольдвассер (Goldwasser) и Сильвио Микэли (Micali). Шифрование было названо «вероятностным» в связи с тем, что один и тот же исходный текст при шифровании с использованием одного и того же ключа может преобразовываться в совершенно различные шифротексты. При использовании криптосистем с открытым ключом существует возможность подбора открытого текста сопоставлением перехваченного шифротекста с результатом шифрования. Вероятностное шифрование позволяет на порядки увеличить сложность такого вида атаки.

Чарльз Беннет (Charles Bennet) и Жиль Брассард (Gilles Brassard), опираясь на работу Стивена Уиснера (Stephen Wiesner), разработали теорию **квантовой криптографии**, которая базируется скорее на квантовой физике, нежели на математике. Процесс отправки и приёма информации выполняется посредством объектов квантовой механики (например, при помощи электронов в электрическом токе, или фотонов в линиях волоконно-оптической связи). Основанная на принципах квантовой механики, эта система, в отличие от обычной криптографии, теоретически позволяет гарантированно защитить информацию от злоумышленника, даже если тот обладает самой современной технологией и неограниченными вычислительными мощностями. На данный момент, разрабатываются только прототипы квантовых криптосистем.

В то же время эффекты квантовой физики, возможно, смогут использоваться и для криптоанализа. Если будут построены квантовые компьютеры, то это поставит под вопрос существование современной криптографии.

Применение криптографии в решении вопросов аутентификации, целостности данных, передачи конфиденциальной информации по каналам связи и т.п. стало неотъемлемым атрибутом информационных систем. В современном мире криптография находит множество различных применений - она используется в сотовой связи, платном цифровом телевидении, при подключении к Wi-Fi, для защиты билетов от подделок на транспорте, в банковских операциях, в системах электронных платежей и т.д.

Вопросы для самопроверки

1. Назовите основные этапы развития криптографии.

3. ОСНОВНЫЕ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ. КЛАССИФИКАЦИЯ ШИФРОВ

3.1. Основные термины и определения

Одно из основных понятий криптографии - шифр (араб. صِفْر (sifr) - «ноль», «ничто», «пустота», откуда фр. chiffre - «цифра»; арабы первыми стали заменять буквы на цифры с целью защиты исходного текста).

Под **шифром** понимается совокупность методов и способов обратимого преобразования информации с целью ее защиты от несанкционированного доступа (обеспечения конфиденциальности информации).

Составными элементами шифра являются:

- алфавиты для записи исходных сообщений (защищаемой информации, открытого текста) и шифрованных сообщений (шифртекстов, шифрограмм, криптограмм);

- алгоритмы криптографического преобразования (зашифрования и дешифрования);

- множество ключей.

Азбука или **алфавит** (греч. ἀλφάβητος) - форма письменности, основанная на стандартном наборе знаков, один или набор которых соответствуют фонемам[1] языка. В общем случае **алфавит для записи исходных сообщений** и **алфавит для записи шифрованных сообщений** могут отличаться. Например, исходные сообщения записываются с помощью букв, а шифрограммы с помощью цифр или графических обозначений.

Алгоритм криптографического преобразования — набор правил (инструкций), определяющих содержание и порядок операций по шифрованию и дешифрованию информации.

Шифрование (зашифрование) — процесс применения шифра к защищаемой информации, т.е. преобразование исходного сообщения в зашифрованное.

Дешифрование (расшифрование) — процесс, обратный шифрованию, т. е. преобразование шифрованного сообщения в исходное.

Алгоритмы шифрования и зашифрования, как правило, отличаются друг от друга, но могут и совпадать. В частности, в некоторых комбинированных блочных шифрах (DES, ГОСТ 28147-89) алгоритмы совпадают, но отличаются порядком использования ключа (ключевых элементов). Алгоритм шифрования может включать в себя предварительное кодирование, а алгоритм расшифрования - обратное перекодирование. Например, в аддитивных шифрах перед шифрованием буквы (символы) исходного сообщения заменяются на числа, а результат расшифрования в виде чисел преобразуется в буквы (символы). Аналогичная ситуация имеет место и в некоторых шифрах замены (шифр Хилла), комбинированных блочных шифрах, асимметричных шифрах.

Ключ – переменный параметр шифра, обеспечивающий выбор одного преобразования из совокупности всевозможных для данного алгоритма и сообщения. В общем случае, **ключ** – это минимально необходимая информация (за исключением сообщения, алфавитов и алгоритма), необходимая для шифрования и дешифрования сообщений.

Используя понятие ключа, процессы шифрования и дешифрования можно описать в виде соотношений:

$$f(P, k_1) = C, \qquad (3.1)$$

$$g(C, k_2) = P, \qquad (3.2)$$

где P (англ. public - открытый) - открытое сообщение;
C (англ. cipher - шифрованный) - шифрованное сообщение;
f - алгоритм шифрования;
g - алгоритм расшифрования;
k_1 – ключ зашифрования, известный отправителю;
k_2 – ключ расшифрования, известный адресату.

В настоящее время разработкой методов шифрования и дешифрования информации (в т.ч. и без знания ключа) занимается **криптология** (греч. κρυπτός — тайный, λόγος — слово, знание). Криптология разделяется на два направления — криптографию и криптоанализ. Цели этих двух правлений криптологии прямо противоположны.

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.

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.

Следует различать криптографические методы сокрытия информации (шифрование) и кодирование. **Кодирование** – представление информации в альтернативном виде. Шифрование является частным случаем кодирования и предназначено для обеспечения конфиденциальности информации. С точки зрения решения этой задачи, кодирование – это безключевое шифрование, в котором преобразование построено на алгоритме кодирования или кодовой таблице. Различают:

- **общедоступные кодовые системы**, предназначенные для преобразования информации в форму, удобную для передачи, хранения или автоматической обработки (азбука Морзе, Unicode, штрихкод);

- **секретные кодовые системы**, предназначенные, как и криптографические системы, для обеспечения конфиденциальности (номенклаторы, «Код Госдепартамента», «Американский код для окопов», «Речные коды : Потомак»).

Параллельно с этим развивается **теория кодирования** - научное направление, которое разрабатывает и изучает методы защиты информации от случайных искажений в каналах связи. Таким образом, в настоящее время, термины кодирование и шифрование применяются для обозначения самостоятельных научных направлений и употреблять их как синонимы некорректно.

Следует отметить, что исторически в криптографии закрепились некоторые военные слова (противник, атака на шифр и др.) Они наиболее точно отражают смысл соответствующих некоторых криптографических понятий [9].

Криптография занимается методами преобразования информации, которые бы не позволили противнику извлечь ее из перехватываемых сообщений. При этом по каналу связи передается не сама защищаемая информация, а результат ее преобразования, и для противника возникает сложная задача вскрытия шифра. **Вскрытие (взламывание) шифра** - процесс получения защищаемой информации из шифрованного сообщения без знания примененного ключа. Способность шифра противостоять всевозможным атакам на него называют **криптографической стойкостью (криптостойкостью) шифра**. Под **атакой на шифр** понимают попытку вскрытия этого шифра.

Понятие стойкости шифра является центральным для криптографии. Хотя качественно понять его довольно легко, но получение строгих доказуемых оценок стойкости для каждого конкретного шифра - проблема нерешенная. Поэтому стойкость конкретного шифра оценивается только путем всевозможных попыток его вскрытия и зависит от квалификации **криптоаналитиков**, атакующих шифр. Такую процедуру иногда называют **проверкой стойкости**. Она делается из предположения, что противник знает сам алгоритм преобразования, но не знает ключа (см. Огюст Керкгоффс, п.2). Противник также может знать некоторые характеристики открытых текстов, например, общую тематику сообщений, их стиль, некоторые стандарты, форматы и т.д.

Однако помимо перехвата и вскрытия шифра противник может пытаться получить защищаемую информацию многими другими способами. Наиболее известным из таких способов является агентурный, когда противник каким-либо путем склоняет к сотрудничеству одного из законных пользователей и с помощью этого агента получает доступ к защищаемой информации. В такой ситуации криптография бессильна.

Противник может попытаться не получить, а уничтожить или модифицировать защищаемую информацию в процессе ее передачи. Это - совсем другой тип угроз для информации, отличный от перехвата и вскрытия шифра. Для защиты от таких угроз разрабатываются свои специфические методы.

Следовательно, на пути от одного законного пользователя к другому информация должна защищаться различными способами, противостоящими различным угрозам. В данной ситуации, противник будет стремиться найти самое слабое звено, чтобы с наименьшими затратами добраться до информации. А значит, и законные пользователи должны учитывать это обстоятельство в своей стратегии защиты: бессмысленно делать какое-то звено очень прочным, если есть заведомо более слабые звенья.

Что касается применения конкретных криптографических методов, то следует отметить, что не существует единого шифра, подходящего для всех случаев. **Выбор способа шифрования зависит**:

- от вида защищаемой информации (документальная, телефонная, телевизионная, компьютерная и т.д.);

- от объема и требуемой скорости передачи шифрованной информации;

- от ценности защищаемой информации (некоторые тайны [например, государственные, военные и др.] должны сохраняться десятилетиями, а некоторые [например, биржевые] - уже через несколько часов можно разгласить);

- от возможностей владельцев секретной информации, а также от возможностей противника (одно дело - противостоять одиночке, а другое дело - мощной государственной

структуре).

Процесс криптографического закрытия данных может осуществляться как программно, аппаратно, так и иными способами. Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д. Программная реализация более практична, допускает известную гибкость в использовании.

---

[1] **Фонема** (др.-греч. φώνημα - «звук») - минимальная смыслоразличимая единица языка.

3.2. Основные требования, предъявляемые к криптосистемам

Для современных криптографических систем можно сформулировать следующие требования:

- сложность и трудоёмкость процедур шифрования и дешифрования должны определяться в зависимости от требуемого уровня защиты информации (необходимо обеспечить надежную защиту информации);

- временные и стоимостные затраты на защиту информации должны быть приемлемыми при заданном уровне ее секретности (затраты на защиту не должны быть чрезмерными);

- процедуры шифрования и дешифрования не должны зависеть от длины сообщения;

- количество всех возможных ключей шифра должно быть таковым, чтобы их полный перебор с помощью современных информационных технологий (в т.ч. и распределенных вычислений) был невозможен за приемлемое для противника время;

- любой ключ из множества возможных должен обеспечивать надежную защиту информации;

- незначительное изменение ключа должно приводить к существенному изменению вида зашифрованного сообщения;

- избыточность сообщений, вносимая в процессе шифрования, должна быть как можно меньшей (хорошим считается результат, когда длина шифрограммы не превышает длину исходного текста);

- зашифрованное сообщение должно поддаваться чтению только при наличии ключа.

3.3. Классификация шифров

Существует несколько классификаций шифров. Рассмотрим некоторые из них.

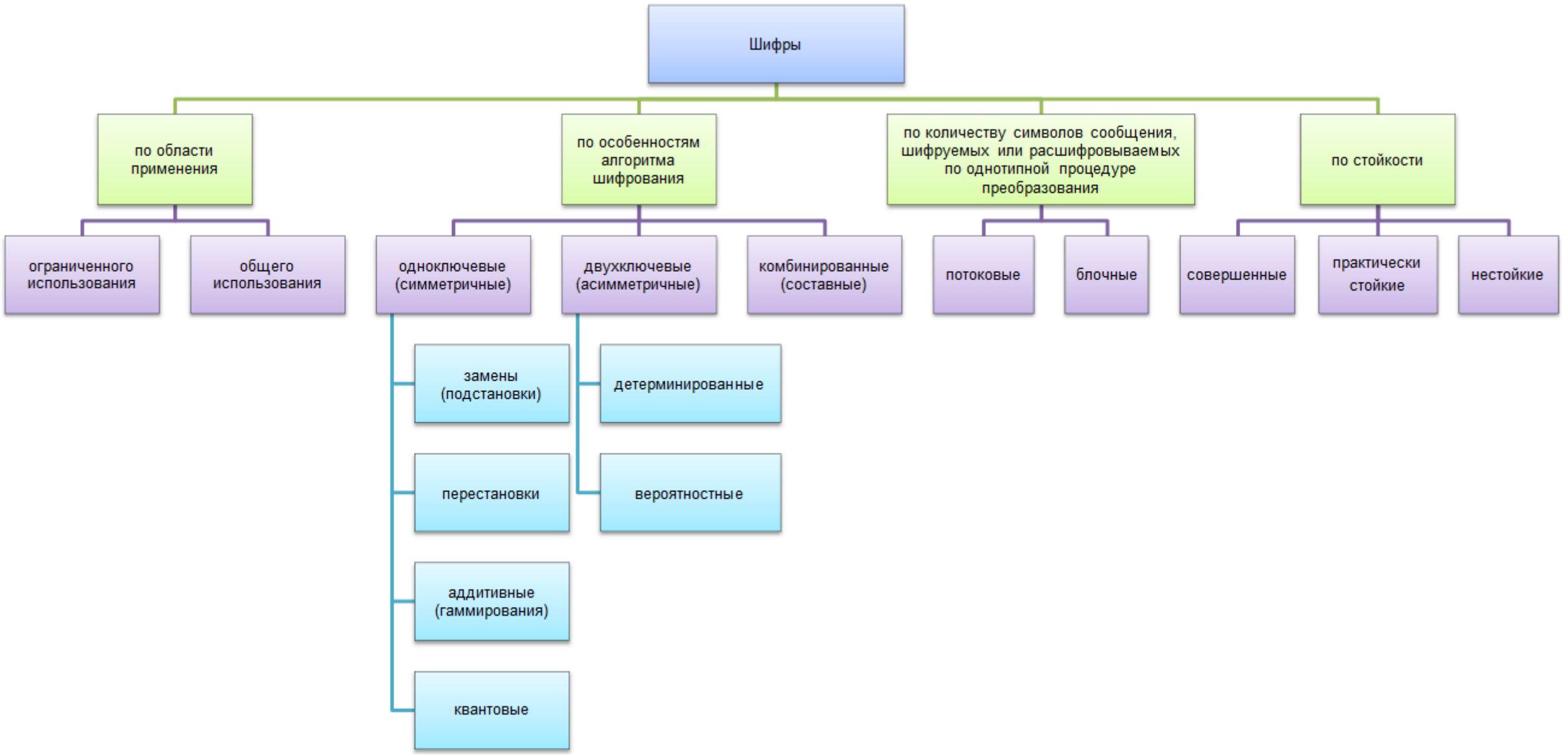


Рис.3.1. Классификация шифров

I. По области применения различают шифры ограниченного и общего использования.

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

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

II. По особенностям алгоритма шифрования шифры общего использования можно разделить на следующие виды.

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

В шифрах перестановки все буквы открытого текста остаются в шифрограмме, но меняют свои позиции. В шифрах замены наоборот, позиции букв в шифрограмме 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), после чего берется остаток от деления по модулю (операция mod). Если исходное сообщение и гамма представляются в битовом виде, то при шифровании и расшифровании применяется логическая операция «Исключающее ИЛИ» (XOR, сложение по модулю 2).

Квантовая криптография вносит в процесс шифрования естественную неопределенность 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 могут установить факт перехвата противником зашифрованного сообщения.

В **двухключевых** системах для шифрования и дешифрования используется два совершено разных ключа. В детерминированных шифрах при шифровании одного и того же сообщения одним и тем же ключом всегда будет получаться один и тот же шифртекст. В вероятностных шифрах в процедуре шифрования используется дополнительная случайная величина (число) - в результате при шифровании одного и того же исходного сообщения одним и тем же ключом могут получиться разные шифртексты, которые при расшифровке дадут один и тот же результат (исходное сообщение).

Комбинированные (составные) методы предполагают использование для шифрования сообщения сразу нескольких методов (например, сначала замена символов, а затем их перестановка).

III. По количеству символов сообщения (или его кодовой замены), шифруемых или расшифровываемых по однотипной процедуре преобразования различают:

- **потоковые шифры** – процедура преобразование применяется к отдельному символу сообщения;

- **блочные шифры** – процедура преобразование применяется к набору (блоку) символов сообщения.

Отличить потоковый шифр от блочного можно по следующему признаку - если в результате разбиения исходного сообщения на отдельные элементарные символы и применения к ним однотипной процедуры преобразования получаемая шифрограмма эквивалентна той, которая получается при применении преобразования «как будто ко всему исходному сообщению», то шифр потоковый, иначе блочный.

IV. По стойкости шифры делятся на три группы:

- **совершенные (абсолютно стойкие, теоретически стойкие)** – шифры, заведомо неподдающиеся вскрытию (при правильном использовании). Дешифрование секретного сообщения приводит к нескольким осмысленным равновероятным открытым сообщениям;

- **практически (вычислительно, достаточно) стойкие** – шифры, вскрытие которых за приемлемое время невозможно на современном или перспективном уровне вычислительной техники. Практическая стойкость таких систем базируется на теории сложности и оценивается исключительно на какой-то определенный момент времени с двух позиций:

  - вычислительная сложность полного перебора;

  - известные на данный момент слабости (уязвимости) и их влияние на вычислительную сложность;

- **нестойкие** шифры.

Вопросы для самопроверки

1. Дайте определение понятиям «шифр», «ключ», «дешифрование».
2. Перечислите основные требования, предъявляемые к криптосистемам.
3. Дайте классификацию криптосистем по алгоритму шифрования.
4. Дайте классификацию криптосистем по стойкости шифра.

4. ШИФРЫ ЗАМЕНЫ

4.1. Основы шифрования

Сущность шифрования методом замены заключается в следующем [9]. Пусть шифруются сообщения на русском языке и замене подлежит каждая буква этих сообщений. Тогда, букве **А** исходного алфавита сопоставляется некоторое множество символов (шифрозамен) $\mathbf{M_А}$**, Б –** $\mathbf{M_Б}$**, …, Я –** $\mathbf{M_Я}$. Шифрозамены выбираются таким образом, чтобы любые два множества ($\mathbf{M_I}$ и $\mathbf{M_J}$, $\mathbf{i \neq j}$) не содержали одинаковых элементов ($\mathbf{M_I \cap M_J = Ø}$).

Таблица, приведенная на рис.4.1, является ключом шифра замены. Зная ее, можно осуществить как шифрование, так и расшифрование.

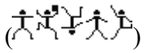
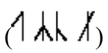
А	Б	...	Я
$M_А$	$M_Б$	...	$M_Я$

Рис.4.1. Таблица шифрозамен

При шифровании каждая буква **А** открытого сообщения заменяется любым символом из множества $\mathbf{M_А}$. Если в сообщении содержится несколько букв **А**, то каждая из них заменяется на любой символ из $\mathbf{M_А}$. За счет этого с помощью одного ключа можно получить различные варианты шифрограммы для одного и того же открытого сообщения. Так как множества $\mathbf{M_А, M_Б, ..., M_Я}$ попарно не пересекаются, то по каждому символу шифрограммы можно однозначно определить, какому множеству он принадлежит, и, следовательно, какую букву открытого сообщения он заменяет. Поэтому расшифрование возможно и открытое сообщение определяется единственным образом.

Приведенное выше описание сущности шифров замены относится ко всем их разновидностям за исключением полиалфавитных шифров, в которых для зашифрования разных символов исходного алфавита могут использоваться одинаковые шифрозамены (т.е. $\mathbf{M_I \cap M_J \neq Ø}$, $\mathbf{i \neq j}$).

Метод замены часто реализуется многими пользователями при работе на компьютере. Если по забывчивости не переключить на клавиатуре набор символов с латиницы на кириллицу, то вместо букв русского алфавита при вводе текста будут печататься буквы латинского алфавита («шифрозамены»).

Для записи исходных и зашифрованных сообщений используются строго определенные алфавиты. Алфавиты для записи исходных и зашифрованных сообщений могут отличаться. Символы обоих алфавитов могут быть представлены буквами, их сочетаниями, числами, рисунками, звуками, жестами и т.п. В качестве примера можно привести пляшущих человечков из рассказа А. Конан Дойла () и рукопись руницического письма () из романа Ж. Верна «Путешествие к центру Земли».

Шифры замены можно разделить на следующие **подклассы** (разновидности).

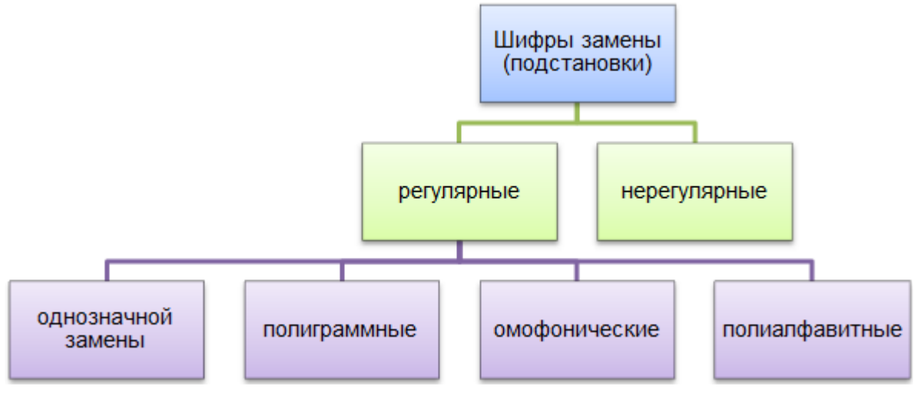


Рис.4.2. Классификация шифров замены

I. Регулярные шифры. Шифрозамены состоят из одинакового количества символов или отделяются друг от друга разделителем (пробелом, точкой, тире и т.п.).

1. Шифры однозначной замены (моноалфавитные, простые подстановочные). Количество шифрозамен для каждого символа исходного алфавита равно 1 ($|\mathbf{M_i}| = 1$ для одного символа).

2. Полиграммные шифры. Аналогичны шифрам однозначной замены за исключением того, что шифрозамене соответствует сразу блок символов исходного сообщения ($|\mathbf{M_i}| \geq 1$ для блока символов).

3. Омофонические шифры (однозвучные, многозначной замены). Количество шифрозамен для отдельных символов исходного алфавита больше 1 ($|\mathbf{M_i}| \geq 1$ для одного символа).

4. Полиалфавитные шифры (многоалфавитные). Состоят из нескольких шифров однозначной замены. Выбор варианта алфавита для зашифрования одного символа зависит от особенностей метода шифрования ($|\mathbf{M_i}| > 1$ для одного символа);

II. Нерегулярные шифры. Шифрозамены состоят из разного количество символов, записываемых без разделителей.

4.2. Шифры однозначной замены

Максимальное количество ключей для любого шифра этого вида не превышает **n!**, где **n** – количество символов в алфавите. С увеличением числа **n** значение **n!** растет очень быстро (1! = 1, 5! = 120, 10! = 3628800, 15! = 1307674368000). При больших **n** для приближенного вычисления **n!** можно воспользоваться формулой Стирлинга

$$n! \approx \sqrt{2\pi n} * \left(\frac{n}{e}\right)^n. \qquad (4.1)$$

Шифр Цезаря. Согласно описаниям историка Светония в книге «Жизнь двенадцати цезарей» данный шифр использовался Гаем Юлием Цезарем для секретной переписке со своими генералами (I век до н.э.) [44]. Применительно к русскому языку суть его состоит в следующем. Выписывается исходный алфавит (**А, Б, ..., Я**), затем под ним выписывается тот же алфавит, но с циклическим сдвигом на 3 буквы влево.

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

Рис.4.3. Таблица шифрозамен для шифра Цезаря

При зашифровке буква **А** заменяется буквой **Г**, **Б** - на **Д** и т. д. Так, например, исходное сообщение «АБРАМОВ» после шифрования будет выглядеть «ГДУГПСЕ». Получатель сообщения «ГДУГПСЕ» ищет эти буквы в нижней строке и по буквам над ними восстанавливает исходное сообщение «АБРАМОВ».

Ключом в шифре Цезаря является величина сдвига нижней строки алфавита. Количество ключей для всех модификаций данного шифра применительно к алфавиту русского языка равно 33. Существуют различные модификации шифра Цезаря, в частности атбаш и лозунговый шифр.

Атбаш. В Ветхом Завете существует несколько фрагментов из священных текстов, которые зашифрованы с помощью шифра замены, называемого атбаш. Этот шифр состоит в замене каждой буквы другой буквой, которая находится в алфавите на таком же расстоянии от конца алфавита, как оригинальная буква - от начала. Например, в русском алфавите буква **А** заменяется на **Я**, буква **Б** - на **Ю** и т.д. В оригинальном Ветхом Завете использовались буквы еврейского алфавита. Так, в книге пророка Иеремии (25:26) слово «Бабель» (Вавилон) зашифровано как «Шешах» [44].

Лозунговый шифр. Для данного шифра построение таблицы шифрозамен основано на лозунге (ключе) – легко запоминаемом слове. Вторая строка таблицы шифрозамен заполняется сначала словом-лозунгом (причем повторяющиеся буквы отбрасываются), а затем остальными буквами, не вошедшими в слово-лозунг, в алфавитном порядке. Например, если выбрано слово-лозунг «ДЯДИНА», то таблица имеет следующий вид.

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

Рис.4.4. Таблица шифрозамен для лозунгового шифра

При шифровании исходного сообщения «АБРАМОВ» по приведенному выше ключу шифрограмма будет выглядеть «ДЯПДКМИ».

В качестве лозунга рекомендуется выбирать фразу, в которой содержаться конечные буквы алфавита. В общем случае, количество вариантов нижней строки (применительно к русскому языку) составляет 33! ($\geq 10^{35}$).

Полибианский квадрат. Шифр изобретен греческим государственным деятелем, полководцем и историком Полибием (203-120 гг. до н.э.). Применительно к русскому алфавиту и индийским (арабским) цифрам суть шифрования

заключалась в следующем. В квадрат 6х6 выписываются буквы (необязательно в алфавитном порядке).

	1	2	3	4	5	6
1	А	Б	В	Г	Д	Е
2	Ё	Ж	З	И	Й	К
3	Л	М	Н	О	П	Р
4	С	Т	У	Ф	Х	Ц
5	Ч	Ш	Щ	Ъ	Ы	Ь
6	Э	Ю	Я	-	-	-

Рис.4.5. Таблица шифрозамен для полибианского квадрата

Шифруемая буква заменяется на координаты квадрата (строка-столбец), в котором она записана. Например, если исходное сообщение «АБРАМОВ», то шифрограмма – «11 12 36 11 32 34 13». В Древней Греции сообщения передавались с помощью оптического телеграфа (с помощью факелов). Для каждой буквы сообщения вначале поднималось количество факелов, соответствующее номеру строки буквы, а затем номеру столбца.

Тюремный шифр [43]. Эта звуковая разновидность полибианского квадрата была разработана заключенными. Система состояла из нескольких ударов, обозначающих строки и столбцы в таблице с буквами алфавита. Один удар, а потом еще два соответствовали строке 1 и столбцу 2, т.е. букве **Б**. Пауза служила разделителем между строками и столбцами. Таким образом, зашифровать исходное сообщение «АБРАМОВ» можно следующим образом.

А	тук ___ тук
Б	тук ___ тук, тук
Р	тук, тук, тук ___ тук, тук, тук, тук, тук, тук
А	тук ___ тук
М	тук, тук, тук ___ тук, тук
О	тук, тук, тук ___ тук, тук, тук, тук
В	тук ___ тук, тук, тук

Рис.4.6. Пример использования тюремного шифра

Шифрующая система Трисемуса (Тритемия). В 1508 г. аббат из Германии Иоганн Трисемус написал печатную работу по криптологии под названием «Полиграфия». В этой книге он впервые систематически описал применение шифрующих таблиц, заполненных алфавитом в случайном порядке. Для получения такого шифра замены обычно использовались таблица для записи букв алфавита и ключевое слово (или фраза). В таблицу сначала вписывалось по строкам ключевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. На рис.4.7 изображена таблица с ключевым словом «ДЯДИНА».

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

Рис.4.7. Таблица шифрозамен для шифра Трисемуса

Каждая буква открытого сообщения заменяется буквой, расположенной под ней в том же столбце. Если буква находится в последней строке таблицы, то для ее шифрования берут самую верхнюю букву столбца. Например, исходное сообщение «АБРАМОВ», зашифрованное – «ЖЗЦЖУФЙ».

Шифр масонов [43]. В XVIII в. масоны создали шифр, чтобы скрыть от общественности свои коммерческие сделки. Как поведали те, кто прежде состоял в рядах этого общества, масоны пользовались способом засекречивания, весьма похожим на шифр розенкрейцеров. В «решетке» и в углах находятся точки, которыми заменяются буквы:

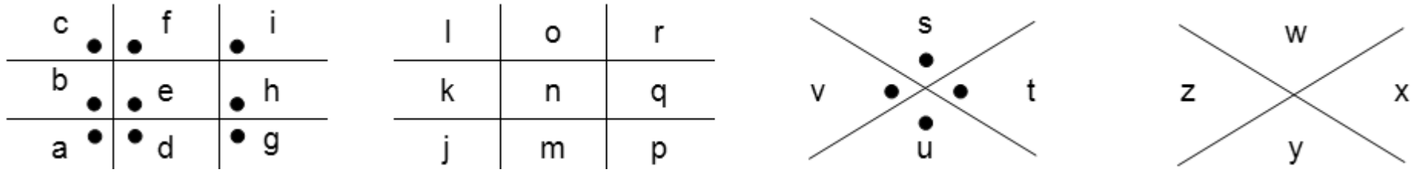


Рис.4.8. Шифрозамены шифра масонов

Так как клятвы хранить тайну нарушались не раз, большинство Великих лож масонов в США больше не пользуются письменными шифрами, предпочитая передавать устные инструкции во время закрытых ритуалов.

С помощью шифра масонов можно легко расшифровать следующую фразу.

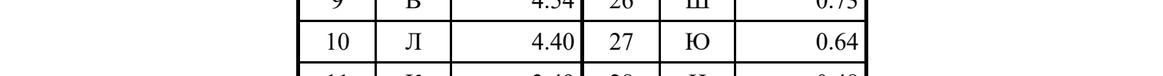


Рис.4.9. Пример использования шифра масонов

Это первый уровень, на котором находятся все впервые вступившие в общество члены: Blue Lodge (рус. «Голубая (Синяя) ложа»).

Одним из существенных **недостатков шифров однозначной замены** является их легкая вскрываемость. При вскрытии шифрограмм используются различные приемы, которые даже при отсутствии мощных вычислительных средств позволяют добиться положительного результата. Один из таких приемов базируется на том, что в шифрограммах остается информация о частоте встречаемости букв исходного текста. Если в открытом сообщении часто встречается какая-либо буква, то в шифрованном сообщении также часто будет встречаться соответствующий ей символ. Еще в 1412 г. Шихаба ал-Калкашанди в своем труде «Субх ал-Ааша» привел таблицу частоты появления арабских букв в тексте на основе анализа текста Корана. Для разных языков мира существуют подобные таблицы. Так, например, для букв русского алфавита по данным "Национального корпуса русского языка" такая таблица выглядит следующим образом [17].

Таблица 4.1. Частота появления букв русского языка в текстах

№ п/п	Буква	Частота, %	№ п/п	Буква	Частота, %
1	О	10.97	18	Ь	1.74
2	Е	8.45	19	Г	1.70
3	А	8.01	20	З	1.65
4	И	7.35	21	Б	1.59
5	Н	6.70	22	Ч	1.44
6	Т	6.26	23	Й	1.21
7	С	5.47	24	Х	0.97
8	Р	4.73	25	Ж	0.94
9	В	4.54	26	Ш	0.73
10	Л	4.40	27	Ю	0.64
11	К	3.49	28	Ц	0.48
12	М	3.21	29	Щ	0.36
13	Д	2.98	30	Э	0.32
14	П	2.81	31	Ф	0.26
15	У	2.62	32	Ъ	0.04
16	Я	2.01	33	Ё	0.04
17	Ы	1.90			

Существуют подобные таблицы для пар букв (биграмм). Например, часто встречаемыми биграммами являются «то», «но», «ст», «по», «ен» и т.д. Другой прием вскрытия шифрограмм основан на исключении возможных сочетаний букв. Например, в текстах (если они написаны без орфографических ошибок) нельзя встретить сочетания «чя», «щы», «ъ» и т.п.

Для усложнения задачи вскрытия шифров однозначной замены еще в древности перед шифрованием из исходных сообщений исключали пробелы и/или гласные буквы. Другим способом, затрудняющим вскрытие, является шифрование **биграммами** (парами букв).

4.3. Полиграммные шифры

Полиграммные шифры замены - это шифры, в которых одна шифрозамена соответствует сразу нескольким символам исходного текста.

Биграммный шифр Порты [43]. Шифр Порты, представленный им в виде таблицы, является первым известным биграммным шифром. Размер его таблицы составлял 20 х 20 ячеек; наверху горизонтально и слева вертикально записывался стандартный алфавит (в нем не было букв J, K, U, W, X и Z). В ячейках таблицы могли быть записаны любые числа, буквы или символы - сам Джованни Порта пользовался символами - при условии, что содержимое ни одной из ячеек не повторялось. Применительно к русскому языку таблица шифрозамен может выглядеть следующим образом.

	А	Б	В	Г	Д	Е (Ё)	Ж	З	И (Й)	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
А	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015	016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031
Б	032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047	048	049	050	051	052	053	054	055	056	057	058	059	060	061	062
В	063	064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079	080	081	082	083	084	085	086	087	088	089	090	091	092	093
Г	094	095	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124
Д	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155
Е (Ё)	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186
Ж	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217
З	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248
И (Й)	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279
К	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310
Л	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341
М	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372
Н	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403
О	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434
П	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465
Р	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496
С	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527
Т	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558
У	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589
Ф	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620
Х	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651
Ц	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682
Ч	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713
Ш	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744
Щ	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775
Ъ	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806

Ы	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837
Ь	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868
Э	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899
Ю	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930
Я	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961

Рис.4.10. Таблица шифрозамен для шифра Порты

Шифрование выполняется парами букв исходного сообщения. Первая буква пары указывает на строку шифрозамены, вторая - на столбец. В случае нечетного количества букв в исходном сообщении к нему добавляется вспомогательный символ («пустой знак»). Например, исходное сообщение «АБ РА МО В», зашифрованное – «002 466 355 093». В качестве вспомогательного символа использована буква «Я».

Шифр Playfair (англ. «Честная игра»). В начале 1850-х гг. Чарлз Уитстон придумал так называемый «прямоугольный шифр». Леон Плейфер, близкий друг Уитстона, рассказал об этом шифре во время официального обеда в 1854 г. министру внутренних дел лорду Пальмерстону и принцу Альберту. А поскольку Плейфер был хорошо известен в военных и дипломатических кругах, то за творением Уитстона навечно закрепилось название «шифр Плейфера».

Данный шифр стал первым буквенным биграммным шифром (в биграммной таблице Порты использовались символы, а не буквы). Он был предназначен для обеспечения секретности телеграфной связи и применялся британскими войсками в Англо-бурской и Первой мировой войнах. Им пользовалась также австралийская служба береговой охраны островов во время Второй мировой войны.

Шифр предусматривает шифрование пар символов (биграмм). Таким образом, этот шифр более устойчив к взлому по сравнению с шифром простой замены, так как затрудняется частотный анализ. Он может быть проведен, но не для 26 возможных символов (латинский алфавит), а для 26 х 26 = 676 возможных биграмм. Анализ частоты биграмм возможен, но является значительно более трудным и требует намного большего объема зашифрованного текста.

Для шифрования сообщения необходимо разбить его на биграммы (группы из двух символов), при этом, если в биграмме встретятся два одинаковых символа, то между ними добавляется заранее оговоренный вспомогательный символ (в оригинале – **X**, для русского алфавита - **Я**). Например, «зашифрованное сообщение» становится «за ши фр ов ан но ес о**Я** об ще ни е**Я**». Для формирования ключевой таблицы выбирается лозунг и далее она заполняется по правилам шифрующей системы Трисемуса. Например, для лозунга «ДЯДИНА» ключевая таблица выглядит следующим образом.

Д	Я	И	Н	А	Б
В	Г	Е	Ё	Ж	З
Й	К	Л	М	О	П
Р	С	Т	У	Ф	Х
Ц	Ч	Ш	Щ	Ъ	Ы
Ь	Э	Ю	-	1	2

Рис.4.11. Ключевая таблица для шифра Playfair

Затем, руководствуясь следующими правилами, выполняется зашифровывание пар символов исходного текста:

1. Если символы биграммы исходного текста встречаются в одной строке, то эти символы замещаются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется на первый символ этой же строки.

2. Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящимися непосредственно под ними. Если символ является нижним в столбце, то он заменяется на первый символ этого же столбца.

3. Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются на символы, находящиеся в тех же строках, но соответствующие другим углам прямоугольника.

Пример шифрования.

- биграмма «за» формирует прямоугольник – заменяется на «жб»;

- биграмма «ши» находятся в одном столбце – заменяется на «юе»;

- биграмма «фр» находятся в одной строке – заменяется на «хс»;

- биграмма «ов» формирует прямоугольник – заменяется на «йж»;

- биграмма «ан» находятся в одной строке – заменяется на «ба»;

- биграмма «но» формирует прямоугольник – заменяется на «ам»;

- биграмма «ес» формирует прямоугольник – заменяется на «гт»;

- биграмма «оя» формирует прямоугольник – заменяется на «ка»;

- биграмма «об» формирует прямоугольник – заменяется на «па»;

- биграмма «ще» формирует прямоугольник – заменяется на «шё»;

- биграмма «ни» формирует прямоугольник – заменяется на «ан»;

- биграмма «ея» формирует прямоугольник – заменяется на «ги».

Шифрограмма – «жб юе хс йж ба ам гт ка па шё ан ги».

Для расшифровки необходимо использовать инверсию этих правил, откидывая символы **Я** (или **X**), если они не несут смысла в исходном сообщении.

Шифр Хилла [17, 43]. Первый практически реализуемый способ шифрования с использованием алгебры был придуман в 1929 г. математиком Лестером Хиллом - профессором из Хантер-колледжа в Нью-Йорке, статья которого «Cryptography in an Algebraic Alphabet» была опубликована в журнале «The American Mathematical Monthly».

Каждой букве алфавита сопоставляется число. Для русского алфавита можно использовать простейшую схему: А = 0, Б = 1, ..., Я = 32. Для зашифрования блок исходного сообщения из **n** букв рассматривается как n-мерный вектор чисел и умножается на матрицу размером **n** x **n** по модулю 33. Данная матрица, совместно с кодовой таблицей сопоставления букв алфавита с числами, является ключом зашифрования. Для расшифрования применяется обратная матрица[1] по модулю.

Например, для триграммных замен могут использоваться следующие матрицы зашифрования / расшифрования.

6	27	1		2	26	17
13	16	32		26	20	4
28	17	15		13	30	21
матрица зашифрования				матрица расшифрования		

Рис.4.12. Матрицы зашифрования / расшифрования

Исходное сообщение «АБРАМОВ», дополненное двумя вспомогательными буквами «яя» (для кратности трем), после сопоставления букв с числами будет выглядеть следующим образом «0 1 17 0 13 15 2 32 32». После перемножения троек чисел на матрицу зашифрования шифрограмма примет следующий вид «11 32 8 3 28 17 17 11 24» (или в буквенном эквиваленте «КЯЗ ГЪР РКЧ»).

АБР - 0 1 17

(6 * 0 + 27 * 1 + 1 * 17) mod 33 = 11 (К)

(13 * 0 + 16 * 1 + 32 * 17) mod 33 = 32 (Я)

(18 * 0 + 17 * 1 + 15 * 17) mod 33 = 8 (З)

АМО - 0 13 15

(6 * 0 + 27 * 13 + 1 * 15) mod 33 = 3 (Г)

(13 * 0 + 16 * 13 + 32 * 15) mod 33 = 28 (Ъ)

(28 * 0 + 17 * 13 + 15 * 15) mod 33 = 17 (Р)

Вяя - 2 32 32

(6 * 2 + 27 * 32 + 1 * 32) mod 33 = 17 (Р)

(13 * 2 + 16 * 32 + 32 * 32) mod 33 = 11 (К)

(28 * 2 + 17 * 32 + 15 * 32) mod 33 = 24 (Ч)

Для расшифрования тройки чисел шифрограммы необходимо умножить на матрицу расшифрования.

КЯЗ - 11 32 8

(2 * 11 + 26 * 32 + 17 * 8) mod 33 = 0 (А)
(26 * 11 + 20 * 32 + 4 * 8) mod 33 = 1 (Б)
(13 * 11 + 30 * 32 + 21 * 8) mod 33 = 17 (Р)

ГЪР - 3 28 17

(2 * 3 + 26 * 28 + 17 * 17) mod 33 = 0 (А)
(26 * 3 + 20 * 28 + 4 * 17) mod 33 = 13 (М)
(13 * 3 + 30 * 28 + 21 * 17) mod 33 = 15 (О)

РКЧ - 17 11 24

(2 * 17 + 26 * 11 + 17 * 24) mod 33 = 2 (В)
(26 * 17 + 20 * 11 + 4 * 24) mod 33 = 32 (я)
(13 * 17 + 30 * 11 + 21 * 24) mod 33 = 32 (я)

В результате будет получен набор чисел «0 1 17 0 13 15 2 32 32», соответствующий исходному сообщению со вспомогательными символами «АБРАМОВяя».

---

[1]**Обратная матрица** - матрица A^{-1}, при умножении на которую, исходная матрица А дает в результате единичную матрицу Е.

4.4. Омофонические шифры

Другое направление повышения стойкости шифров замены состоит в том, чтобы каждое множество шифрообозначений $\mathbf{M_i}$ содержало более одного элемента. При использовании такого шифра одну и ту же букву (если она встречается несколько раз в сообщении) заменяют на разные шифрозамены из $\mathbf{M_i}$. Это позволяет скрыть истинную частоту встречаемости букв открытого сообщения.

Система омофонов. В 1401 г. Симеоне де Крема стал использовать таблицы омофонов для сокрытия частоты появления гласных букв в тексте при помощи более чем одной шифрозамены. Такие шифры позже стали называться **шифрами многозначной замены** или **омофонами**[2]. Они получили развитие в XV веке. В книге «Трактат о шифрах» Леона Баттисты Альберти (итальянский ученый, архитектор, теоретик искусства, секретарь папы Климентия XII), опубликованной в 1466 г., приводится описание шифра замены, в котором каждой букве ставится в соответствие несколько эквивалентов, число которых пропорционально частоте встречаемости буквы в открытом тексте. Так, если ориентироваться на табл.4.1, то число шифрозамен для буквы **О** должно составлять 110, для буквы **Е** – 85 и т.д. При этом каждая шифрозамена должна состоять из 3 цифр и их общее количество равно 1000. На рис.4.12 представлен фрагмент таблицы шифрозамен.

№ п/п	А	Б	В	…	М	…	О	…	Р	…	Я
1	311	128	175	…	037	…	248	…	064	…	266
2	357	950	194	…	149	…	267	…	189	…	333
…	…	…	…	…	…	…	…	…	…	…	…
16	495	990	199	…	349	…	303	…	374	…	749
…	…		…	…	…	…	…	…	…	…	…
20	519		427	…	760	…	306	…	469	…	845
…	…		…	…	…	…	…	…	…		
32	637		524	…	777	…	432	…	554		
…	…		…				…	…	…		
45	678		644				824	…	721		
…	…						…	…	…		
47	776						828	…	954		
…	…						…				

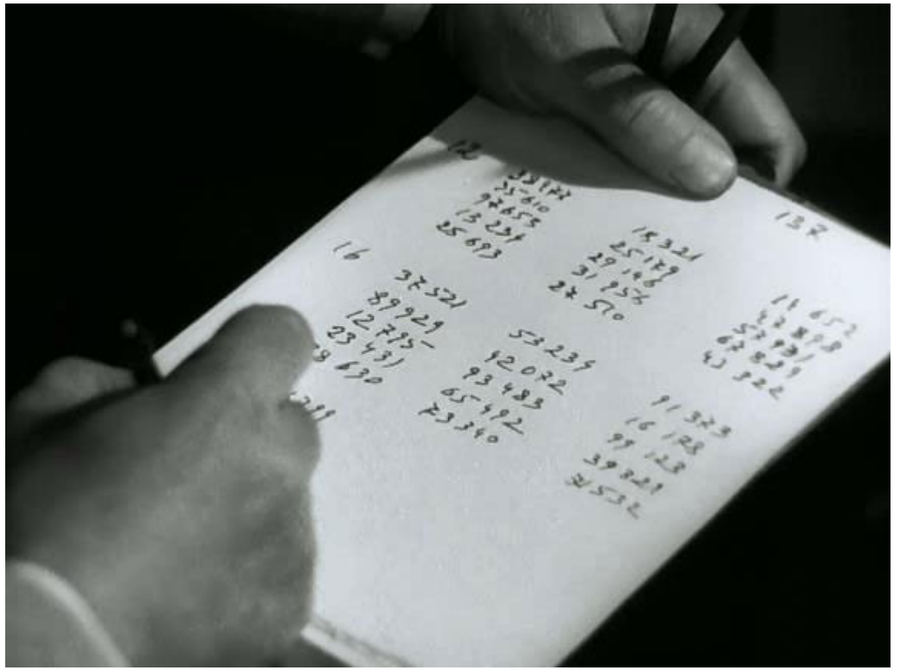
80	901		886
…			…
110			903

Рис.4.13. Фрагмент таблицы шифрозамен для системы омофонов

При шифровании символ исходного сообщения заменяется на любую шифрозамену из своего столбца. Если символ встречается повторно, то, как правило, используют разные шифрозамены. Например, исходное сообщение «АБРАМОВ» после шифрования может выглядеть «357 990 374 678 037 828 175».

Книжный шифр. [14] Заметным вкладом греческого ученого Энея Тактика в криптографию является предложенный им так называемый книжный шифр, описанный в сочинении «Об обороне укреплённых мест». Эней предложил прокалывать малозаметные дырки в книге или в другом документе над буквами секретного сообщения. Интересно отметить, что в Первой мировой войне германские шпионы использовали аналогичный шифр, заменив дырки на точки, наносимые симпатическими чернилами[3] на буквы газетного текста. Описанные способы передачи секретных сообщений (с помощью точек) относятся к стеганографическим методам сокрытия информации.

После Первой мировой войны книжный шифр приобрел иной вид. Шифрозамена для каждой буквы определялась набором цифр, которые указывали на номер страницы, строки и позиции в строке.



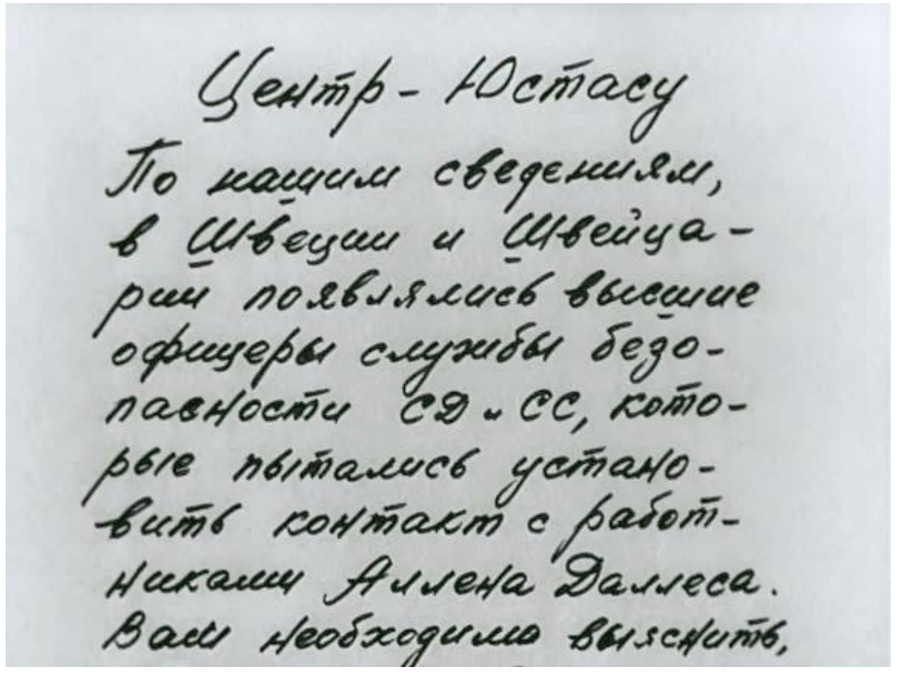


Рис.4.14. Пример использования книжного шифра
(кадры из советского сериала «Семнадцать мгновений весны»)

Количество книг, изданных за всю историю человечества, является величиной ограниченной (по крайней мере, явно меньше, чем 15!). Однако отсутствие полной электронной базы по изданиям делает процедуру вскрытия шифрограмм почти не выполнимой.

Вариантные шифры [43]. Вариантные шифры напоминают полибианский квадрат, но для каждой строки и столбца используется по два буквенных идентификатора. В квадрат (прямоугольник) шифрозамен вначале записывается ключевое слово без повторяющихся букв, а затем дополняется не вошедшими в него буквами по порядку следования в алфавите. Каждой строке и столбцу квадрата ставится в соответствие по две буквы алфавита. Буквы для идентификации строк и столбцов не должны повторяться.

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

Рис.4.15. Пример таблицы шифрозамен вариантного шифра с ключевым словом «ДЯДИНА»

Комбинации букв-идентификаторов строки и столбца дают по восемь шифрозамен для каждой буквы исходного текста. Например, для буквы **Д** возможны шифрозамены: **ФЙ**, **ЙФ**, **ФЦ**, **ЦФ**, **ЫЙ**, **ЙЫ**, **ЫЦ** и **ЦЫ**. Для таблицы шифрозамен, приведенной на рис. 4.15, исходное сообщение «АБРАМОВ» может быть зашифровано как «ЫЗ ЫХ ОЦ ЗФ ГР РЩ АЙ».

---

[2]**Омофоны** (греч. homos - одинаковый и phone - звук) - слова, которые звучат одинаково, но пишутся по-разному и имеют разное значение.

[3]**Симпатические (невидимые) чернила** — чернила, записи которыми являются изначально невидимыми и становятся видимыми только при определенных условиях (нагрев, освещение, химический проявитель и т. д.).

4.5. Полиалфавитные шифры

Напомним, что полиалфавитные шифры состоят из нескольких шифров однозначной замены и отличаются друг от друга способом выбор варианта алфавита для зашифрования одного символа.

Диск Альберти. В «Трактате о шифрах» Альберти приводит первое точное описание многоалфавитного шифра на основе шифровального диска.

Рис.4.16. Реплика диска Альберти, используемого Конфедерацией во время Гражданской войны в Америке [www.cryptomuseum.com]

Он состоял из двух дисков – внешнего неподвижного и внутреннего подвижного дисков, на которые были нанесены буквы алфавита. Процесс шифрования заключался в нахождении буквы открытого текста на внешнем диске и замене ее на букву с внутреннего диска, стоящую под ней. После этого внутренний диск сдвигался на одну позицию и шифрование второй буквы производилось уже по новому шифралфавиту. Ключом данного шифра являлся порядок расположения букв на дисках и начальное положение внутреннего диска относительно внешнего.

Таблица Трисемуса. Одним из шифров, придуманных немецким аббатом Трисемусом, стал многоалфавитный шифр, основанный на так называемой «таблице Трисемуса» - таблице со стороной равной **n**, где **n** – количество символов в алфавите. В первой строке матрицы записываются буквы в порядке их очередности в алфавите, во второй – та же последовательность букв, но с циклическим сдвигом на одну позицию влево, в третьей – с циклическим сдвигом на две позиции влево и т.д.

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

Рис.4.17. Таблица Трисемуса

Первая строка является одновременно и алфавитом для букв открытого текста. Первая буква текста шифруется по первой строке, вторая буква по второй и так далее. После использования последней строки вновь возвращаются к первой. Так сообщение «АБРАМОВ» приобретет вид «АВТГРУЗ».

Система шифрования Виженера. В 1586 г. французский дипломат Блез Виженер представил перед комиссией Генриха III описание простого, но довольно стойкого шифра, в основе которого лежит таблица Трисемуса.

Перед шифрованием выбирается ключ из символов алфавита. Сама процедура шифрования заключается в следующем. По i-ому символу открытого сообщения в первой строке определяется столбец, а по i-ому символу ключа в крайнем левом столбце – строка. На пересечении строки и столбца будет находиться i-ый символ, помещаемый в шифрограмму. Если длина ключа меньше сообщения, то он используется повторно. Например, исходное сообщение «АБРАМОВ», ключ – «ДЯДИНА», шифрограмма – «ДАФИЪОЁ».

Справедливости ради, следует отметить, что авторство данного шифра принадлежит итальянцу Джованни Батиста Беллазо, который описал его в 1553 г. История «проигнорировала важный факт и назвала шифр именем Виженера, несмотря на то, что он ничего не сделал для его создания» [13]. Беллазо предложил называть секретное слово или фразу **паролем** (ит. password; фр. parole - слово).

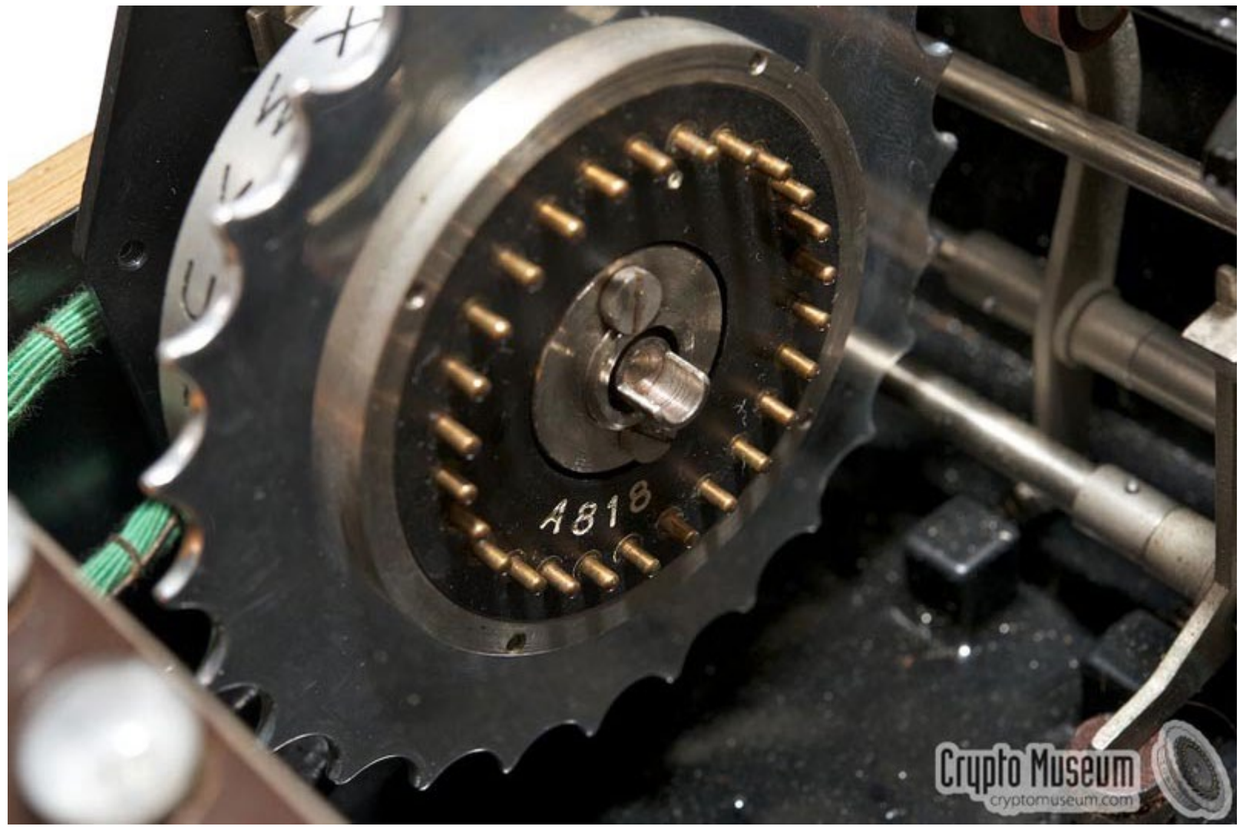
В 1863 г. Фридрих Касиски опубликовал алгоритм атаки на этот шифр, хотя известны случаи его взлома шифра некоторыми опытными криптоаналитиками и ранее. В частности, в 1854 г. шифр был взломан изобретателем первой аналитической вычислительной машины Чарльзом Бэббиджем, хотя этот факт стал известен только в XX в., когда группа ученых разбирала вычисления и личные заметки Бэббиджа [44]. Несмотря на это шифр Виженера имел репутацию исключительно стойкого к «ручному» взлому еще долгое время. Так, известный писатель и математик Чарльз Лютвидж Доджсон (Льюис Кэрролл) в своей статье «Алфавитный шифр», опубликованной в детском журнале в 1868 г., назвал шифр Виженера невзламываемым. В 1917 г. научно-популярный журнал «Scientific American» также отозвался о шифре Виженера, как о неподдающемся взлому [13].

Роторные машины. Идеи Альберти и Беллазо использовались при создании электромеханических роторных машин первой половины XX века. Некоторые из них использовались в разных странах вплоть до 1980-х годов. В большинстве из них использовались роторы (механические колеса), взаимное расположение которых определяло текущий алфавит шифрозамен, используемый для выполнения подстановки. Наиболее известной из роторных машин является немецкая машина времен Второй мировой войны «Энигма» [8].

Рис.4.18. Энигма [www.cryptomuseum.com]

Выходные штыри одного ротора соединены с входными штырями следующего ротора и при нажатии символа исходного сообщения на клавиатуре замыкали электрическую цепь, в результате чего загоралась лампочка с символом шифрозамены.

а) четыре последовательно соединённых ротора



б) штыри ротора

Рис.4.19. Роторная система Энигмы [www.cryptomuseum.com]

Шифрующее действие «Энигмы» показано для двух последовательно нажатых клавиш - ток течёт через роторы, «отражается» от рефлектора, затем снова через роторы.

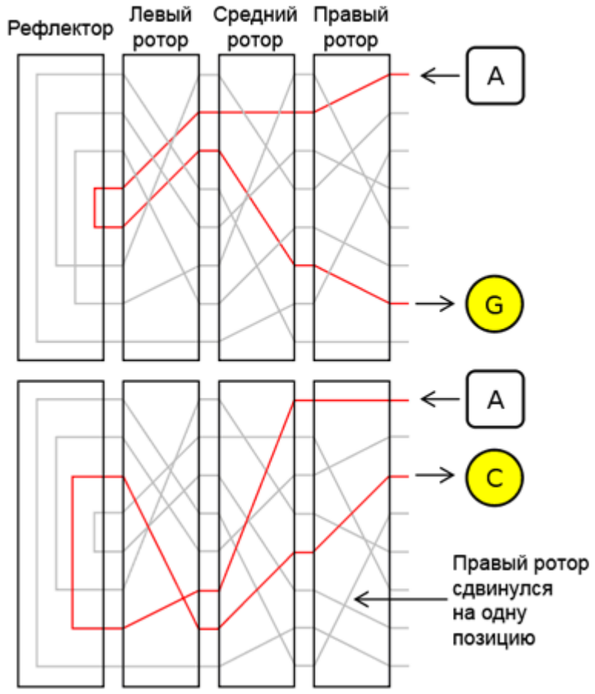


Рис.4.20. Схема шифрования

Примечание. Серыми линиями показаны другие возможные электрические цепи внутри каждого ротора. Буква **A** шифруется по-разному при последовательных нажатиях одной клавиши, сначала в **G**, затем в **C**. Сигнал идет по другому маршруту за счёт поворота одного из роторов после нажатия предыдущей буквы исходного сообщения.

Шифры Тени [43]. Главными развлечениями для американцев тридцатых годов XX века были бульварное чтиво и радио. Для раскрутки своих книжек издательство Street & Smith проспонсировало радиопередачу, ведущим в которой был Тень (англ. Shadow), загадочный рассказчик со зловещим голосом, который в начале каждого выпуска заявлял: «Кто знает, что за зло прячется в сердцах людей? Тень знает!». Успех радиопередачи подтолкнул издательство к решению начать выпускать серию книг, в которой главным героем был бы Тень. Свои услуги предложил Уолтер Гибсон, большой любитель фокусов и головоломок. Под псевдонимом Максвелл Грант он принялся писать роман за романом, да с такой скоростью, что за свою жизнь написал почти 300 книжек о грозе тех, кто нечист помыслами. В новелле «Цепочка смерти» супергерой воспользовался так называемым кодом направления, хотя на самом деле он действует скорее как шифр, чем как код.

												
a	b	c	d	e	f	g	h	i	j	k	l	m
												
n	o	p	q	r	s	t	u	v	w	x	y	z
												
пробел		1	2	3	4							

Рис.4.21. Таблица шифрозамен и управляющих символов

Управляющие символы в последней строке таблицы служат для изменения кода (выбора шифралфавита) для зашифрования/дешифрования. Линии внутри каждого кружка фактически являются стрелками, подсказывающими адресату, как держать лист бумаги. Символ 1 означает, что лист надо держать как обычно: верх и низ расположены на своих местах, а сообщение читается слева направо. Символ 2 требует поворота на 90° вправо, а символ 3 указывает, что лист бумаги следует перевернуть вверх ногами. Символ 4 обозначает поворот на 90° влево. Эти дополнительные символы могут появляться перед любой строчкой текста, а также в ее середине.

Из нижеприведенного примера можно узнать настоящие имя и фамилию супергероя.

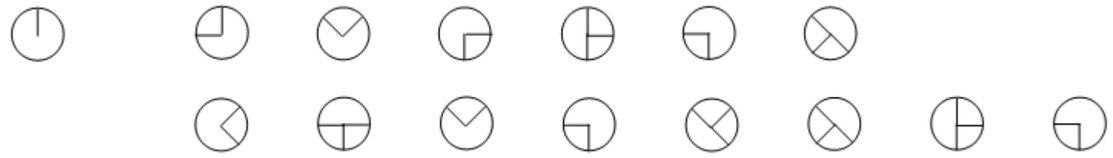


Рис.4.22. Настоящие имя и фамилия Тени

Согласно первому управляющему символу, лист следует держать обычным образом, не поворачивая, и после замены буквы образуют «Lamont Cranston» (Ламонт Крэнстон).

4.6. Нерегулярные шифры

Еще одним направлением повышения стойкости шифров замены заключается в использовании нерегулярных шифров. В приведенных выше шифрах (**регулярных**) шифрозамены состоят из строго определенного количества символов (букв, цифр, графических элементов и т.д.) или в шифрограмме они отделяются друг от друга специальными символами (пробелом, точкой, запятой и т.д.). В нерегулярных шифрах шифрозамены состоят из разного количества символов и записываются в шифрограмме в подряд (без выделения друг от друга), что значительно затрудняет криптоанализ.

Совмещенный шифр (совмещенная таблица) [43]. Данный шифр применялся еще семейством Ардженти - криптологами, разрабатывавшими шифры для Папы Римского в XVI в. В XX столетии этим способом пользовались коммунисты в ходе гражданской войны в Испании. В начале войны противники фашизма в Испании контролировали большинство крупных городов и защищали свою связь, включая радиопередачи, с помощью различных методов шифрования, в том числе совмещенных шифров.

Вариант коммунистов получил название «совмещенный» из-за необычного использования одно- и двухцифровых шифрозамен, благодаря чему сообщение приобретало дополнительную защиту от потенциального дешифровальщика. Некоторые буквы зашифровывались одной цифрой, другие же - парой цифр. При этом криптоаналитик противника совершенно не представлял, где в перехваченных сообщениях находятся одноцифровые, а где двухцифровые шифрозамены.

Таблица шифрозамен состоит из 10 столбцов с нумерацией 0, 9, 8, 7, 6, 5, 4, 3, 2 и 1. В начальную строку вписывается ключевое слово без повторяющихся букв. В последующие строки вписываются по десять не вошедших в него букв по порядку следования в алфавите. Строки, за исключением начальной, нумеруются по порядку, начиная с 1.

	0	9	8	7	6	5	4	3	2	1
	Д	Я	И	Н	А					
1	Б	В	Г	Е	Ё	Ж	З	Й	К	Л
2	М	О	П	Р	С	Т	У	Ф	Х	Ц
3	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	-	-

Рис.4.23. Пример таблицы шифрозамен совмещенного шифра с ключевым словом «ДЯДИНА»

При шифровании буквы исходного сообщения, входящие в ключевое слово, заменяются на одну цифру (номер столбца), остальные – двумя (номера строки и столбца). Например, для приведенной выше таблицы шифрозамен исходное сообщение «АБРАМОВ» будет зашифровано как «610276202919».

При получении шифрограммы адресат знает, что когда появляются цифры 1, 2 или 3, с ними обязательно связана еще одна цифра, поскольку они представляют собой цифровую пару. Так что 35 - это, несомненно, пара, а 53 - нет, ведь в таблице нет строки с номером 5. Перехват такого сообщения третьей стороной даст ей всего лишь ряд цифр, потому что криптоаналитик противника не имеет ни малейшего представления, какие цифры одиночные, а какие входят в состав пар.

Вопросы для самопроверки

1. В чем заключается основная идея криптографических преобразований шифров замены?

2. Перечислите основные разновидности шифров замены.

3. Дайте характеристику разновидностям шифров замены.

4. Назовите основной недостаток шифра однозначной замены.

5. ШИФРЫ ПЕРЕСТАНОВКИ

5.1. Основы шифрования

Перестановка представляет собой способ шифрования, при котором для получения шифрограммы буквы исходного сообщения меняют местами. Типичным примером перестановки являются анаграммы, ставшие популярными в XVII в. **Анаграмма** (греч. ανα - «снова» и γράμμα - «запись») - литературный приём, состоящий в перестановке букв или звуков определённого слова (или словосочетания), что в результате даёт другое слово или словосочетание [17]. Например: апельсин - спаниель, полковник - клоповник, горилка - рогалик, лепесток - телескоп.

Прародителем анаграммы однако считают поэта и грамматика Ликофрона, который жил в Древней Греции в III веке до н. э. Как сообщал византийский автор Иоанн Цец, из имени царя Птоломея он составил первую из известных нам анаграмм: Ptolemaios - Apo Melitos, что в переводе означает «из мёда», а из имени царицы Арсинои: Arsinoe - Ion Eras («фиалка Геры»). В XVII—XIX вв. среди естествоиспытателей было принято зашифровывать свои открытия в виде анаграмм, что служило двум нуждам: скрыть гипотезу до её окончательной проверки и утвердить авторство на открытие, когда оно будет подтверждено. Так, в 1610 г. Галилео Галилей для закрепления авторства на открытие спутников Сатурна зашифровал латинскую фразу «Altissimun planetam tergeminum observavi» («Высочайшую планету тройную наблюдал») следующим образом: «Smaismrmilmepoetaleumibunenugttauiras» (буквы «v» и «u» в латинских текстах часто считались взаимозаменяемыми)[1].

Доподлинно не известно, когда появился шифр перестановки, но вполне возможно, что писцы в древности переставляли буквы в имени своего царя ради того, чтобы скрыть его подлинное имя или в ритуальных целях [43].

Все шифры перестановки делятся на два **подкласса**:

- шифры одинарной (простой) перестановки. При шифровании символы перемещаются с исходных позиций в новые один раз;

- шифры множественной (сложной) перестановки. При шифровании символы перемещаются с исходных позиций в новые несколько раз.

---

[1]В те времена самой дальней из известных («высочайшей») планет был Сатурн. В действительности, в силу несовершенства телескопа, используемого Галилеем, он наблюдал не спутники, а кольца Сатурна, которые выступали по краям планеты. Эти выступы («ушки») и были приняты им за спутники. Свою гипотезу о них в виде анаграммы Галилей отослал Иоганну Кеплеру, который, потратив множество сил, перевел ее как «Salve umbistineum geminatum Martia proles» (лат., «Возрадуйтесь, два протуберанца - сыны Марса»). Кеплер не обратил внимание на то, что в его переводе была лишняя буква, т.к. по его соображениям у Марса должны были быть два спутника, неоткрытые на тот момент. Спутники Марса (Фобос и Деймос) были открыты американским астрономом Асафом Холлом позже - в 1877 г. [54].

5.2. Шифры одинарной перестановки

В общем случае для данного класса шифров при шифровании и дешифровании используется таблица перестановок.

1	2	3	…	n
I_1	I_2	I_3	…	I_n

Рис.5.1. Таблица перестановок

В первой строке данной таблицы указывается позиция символа в исходном сообщении, а во второй – его позиция в шифрограмме. Таким образом, максимальное количество ключей для шифров перестановки равно **n!**, где **n** – длина сообщения.

Шифр простой одинарной перестановки. Для шифрования и дешифрования используется таблица перестановок, аналогичная показанной на рис.5.2.

1	2	3	4	5	6	7
2	4	1	7	6	5	3

Рис.5.2. Таблица перестановок

Например, если для шифрования исходного сообщения «АБРАМОВ» использовать таблицу, представленную на рис.5.2, то шифрограммой будет «РАВБОМА». Для использования на практике такой шифр не удобен, так как при больших значениях **n** приходится работать с длинными таблицами и для сообщений разной длины необходимо иметь свою таблицу перестановок.

Шифр блочной одинарной перестановки. При использовании этого шифра задается таблица перестановки блока символов, которая последовательно применяется до тех пор, пока исходное сообщение не закончится. Если исходное сообщение не кратно размеру блока, тогда оно при шифровании дополняется произвольными символами.

1	2	3
2	3	1

Рис.5.3. Таблица перестановок

Для примера выберем размер блока, равный 3, и примем таблицу перестановок, показанную на рис.5.3. Дополним исходное сообщение «АБРАМОВ» буквами **Ь** и **Э**, чтобы его длина была кратна 3. В результате шифрования получим «РАБОАМЭВЬ».

Количество ключей для данного шифра при фиксированном размере блока равно **m!**, где **m** – размер блока.

Шифры маршрутной перестановки. Широкое распространение получили шифры перестановки, использующие некоторую геометрическую фигуру (плоскую или объемную). Преобразования состоят в том, что в фигуру исходный текст вписывается по ходу одного маршрута, а выписывается по другому. Один из таких шифров – шифр «Сцитала» - упоминался ранее. Некоторые из них приводятся ниже.

Шифр табличной маршрутной перестановки. Наибольшее распространение получили шифры маршрутной перестановки, основанные на таблицах. При шифровании в такую таблицу вписывают исходное сообщение по определенному маршруту, а выписывают (получают шифрограмму) - по другому. Для данного шифра маршруты вписывания и выписывания, а также размеры таблицы являются ключом.

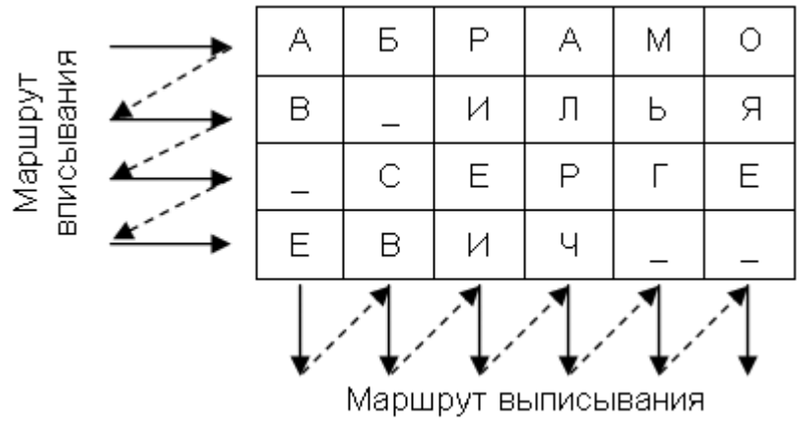


Рис.5.4. Пример использования шифра маршрутной перестановки

Например, исходное сообщения «АБРАМОВ ИЛЬЯ СЕРГЕЕВИЧ» вписывается в прямоугольную таблицу размерами 4х6, маршрут вписывания – слева-направо сверху-вниз, маршрут выписывания – сверху-вниз слева-направо. Шифрограмма в этом случае выглядит «АВ_ЕБ_СВРИЕИАЛР ЧМЬГ_ОЯЕ_».

Шифр вертикальной перестановки. Является разновидностью предыдущего шифра. К особенностям шифра можно отнести следующие:

- количество столбцов в таблице фиксируется и определяется длиной ключа;

- маршрут вписывания - строго слева-направо сверху-вниз;

- шифрограмма выписывается по столбцам в соответствии с их нумерацией (ключом).

Ключ	Д	Я	Д	И	Н	А
	2	6	3	4	5	1
Текст	А	Б	Р	А	М	О
	В	_	И	Л	Ь	Я
	_	С	Е	Р	Г	Е
	Е	В	И	Ч	_	_

Рис.5.5. Пример использования шифра вертикальной перестановки

В качестве ключа можно использовать слово или фразу. Тогда порядок выписывания столбцов соответствует алфавитному порядку букв в ключе. Например, если ключевым словом будет «ДЯДИНА», то присутствующая в нем буква **А** получает номер 1, **Д** – 2 и т.д. Если какая-то буква входит в слово несколько раз, то ее появления нумеруются последовательно слева направо. В примере первая буква **Д** получает номер 2, вторая **Д** – 3.

При шифровании сообщения «АБРАМОВ ИЛЬЯ СЕРГЕЕВИЧ» результат будет «ОЯЕ_АВ_ЕРИЕИАЛРЧМЬГ_Б_СВ».

Шифр «Перекресток» [43]. Для перемешивания букв могут использоваться фигуры специального вида. Один из таких способов носит название «перекресток». В приведенном ниже примере рисуют крестообразные фигуры в количестве, достаточном, чтобы разместить в них все буквы сообщения. Открытый текст записывают вокруг этих фигур заранее оговоренным способом - в нашем случае по часовой стрелке. Таким образом, сообщение «АБРАМОВ ИЛЬЯ СЕРГЕЕВИЧ» может выглядеть следующим образом:

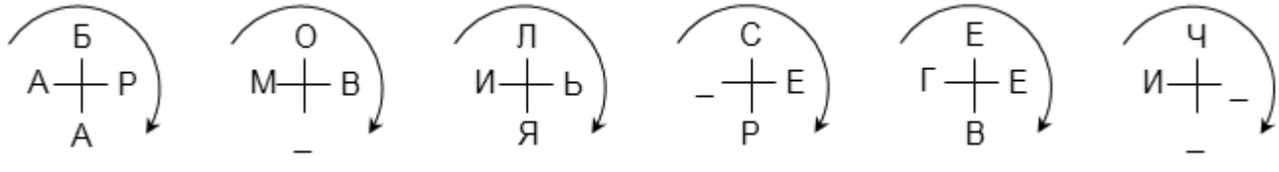


Рис.5.6. Пример размещения открытого текста в шифре «Перекресток»

Буквы берутся построчно. Вначале берется оговоренное количество букв (N) из первой строки, затем удвоенное количество букв (2N) из второй и снова N букв из третьей строки. Например, при N = 3 шифрограмма будет выглядеть «БОЛАРМВИЬА_ЯСЕЧ_ЕГЕИ_РВ_».

Шифры с использованием треугольников и трапеций [43]. Помочь выполнить перестановки могут как треугольники, так и трапеции. Открытый текст вписывается в эти фигуры в соответствии с количеством слов и формой выбранной фигуры, которая может быть растянута или сжата, чтобы в ней поместилось сообщение. Для первой фигуры, треугольника, открытый текст записывается построчно от вершины до основания.

				А						
			Б		Р					
		А		М		О				
	В		_		И		Л			
Ь		Я		_		С		Е		
Р		Г		Е		Е		В		И
Д	Я	Д	И	Н	А	Д	Я	Д	И	Н
2	10	3	6	8	1	4	11	5	7	9

Рис.5.7. Пример использования шифра перестановки при вписывании в треугольник

Ниже записывается ключевое слово. Поскольку основание широкое, ключевое слово повторяется. Буквы строки с ключевым словом нумеруются последовательно согласно их алфавитному порядку. Зашифрованное сообщение выписывается по столбцам согласно выполненной нумерации. Таким образом, для открытого текста «АБРАМОВ ИЛЬЯ СЕРГЕЕВИ» и ключевого слова «ДЯДИНА» шифрограмма будет выглядеть «АМ_РВГРИЕЛВАЯЕБ_ЕИЬРС».

Шифр «Поворотная решетка» [14, 17, 43]. В 1550 г. итальянский математик Джероламо Кардано[2], состоящий на службе у папы Римского, в книге «О тонкостях» предложил новую технику шифрования - **решётку Кардано**.

Изначально решетка Кардано представляла собой трафарет с прорезанными в нем отверстиями. В этих отверстиях на листе бумаги, который клали под решетку, записывались буквы, слоги и слова сообщения. Далее трафарет снимался, и свободное пространство заполнялось более или менее осмысленным текстом для маскировки секретного послания. Такой метод сокрытия информации относится к стеганографии.

Позднее был предложен шифр «поворотная решетка» или, как его еще называют, «решетка для вьющихся растений», поскольку она напоминала отверстия в деревянных решетках садовых строений. Этот шифр считают первым **транспозиционным** (геометрическим) шифром.

Несмотря на то, что между изначальным предложением Кардано и шифром «поворотная решетка» большая разница, методы сокрытия информации, основанные на использовании трафаретов, принято называть «решетками Кардано».

Для шифрования и дешифрования с помощью данного шифра изготовляется прямоугольный трафарет с четным количеством строк и столбцов. В трафарете вырезаются клетки таким образом, чтобы при наложении его на таблицу того же размера четырьмя возможными способами, его вырезы полностью покрывали все ячейки таблицы ровно по одному разу.

При шифровании трафарет накладывается на таблицу. В видимые ячейки таблицы выписываются буквы исходного текста слева-направо сверху-вниз. Далее трафарет поворачивается и вписывается следующая часть букв. Эта операция повторяется еще два раза. Шифрограмму выписывают из итоговой таблицы по определенному

маршруту.

Таким образом, ключом при шифровании является трафарет, порядок его поворотов и маршрут выписывания.

Пример шифрования сообщения «АБРАМОВ+ДЯДИНА» показан на рис.5.8. Результат шифрования – «АДВ_МНРДБЯ+_ОААИ».

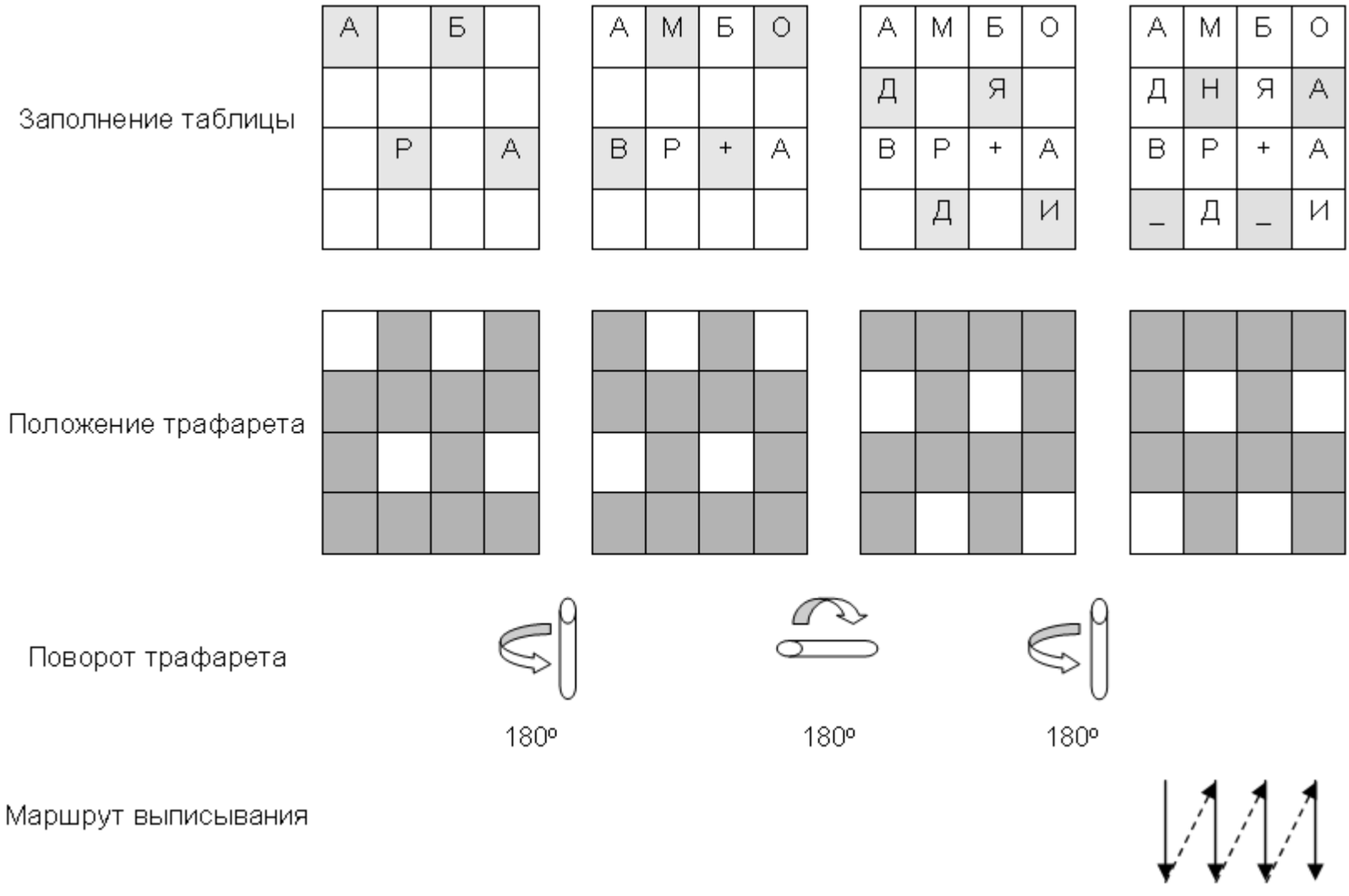


Рис.5.8. Пример использования шифра «поворотная решетка»

Данный метод шифрования применялся нидерландскими правителями для секретных посланий в 1740-х гг. Он также использовался в армии кайзера Вильгельма в Первую мировую войну. Для шифрования немцы использовали решетки разных размеров, которым французские криптоаналитики дали собственные кодовые имена: Анна (25 букв), Берта (36 букв), Дора (64 буквы) и Эмиль (81 буква). Однако использовались решетки очень недолго (всего четыре месяца) к огромному разочарованию французов, которые только-только начали подбирать к ним ключи.

Магические квадраты. Магическими квадратами называются квадратные таблицы со вписанными в их клетки последовательными натуральными числами начиная с 1, которые в сумме по каждому столбцу, каждой строке и каждой диагонали дают одно и то же число.

Впервые эти квадраты появились в Китае, где им и была приписана некоторая «магическая сила». По преданию, описанному в одной из пяти канонических книг Древнего Китая - Шу-Цзин (Книге записанных преданий), в 2200 году до н.э. из реки Ло вышла огромная черепаха (по другой версии - дракон), символ вечности. На ее панцире были видны пятна, образовывавшие удивительный рисунок.

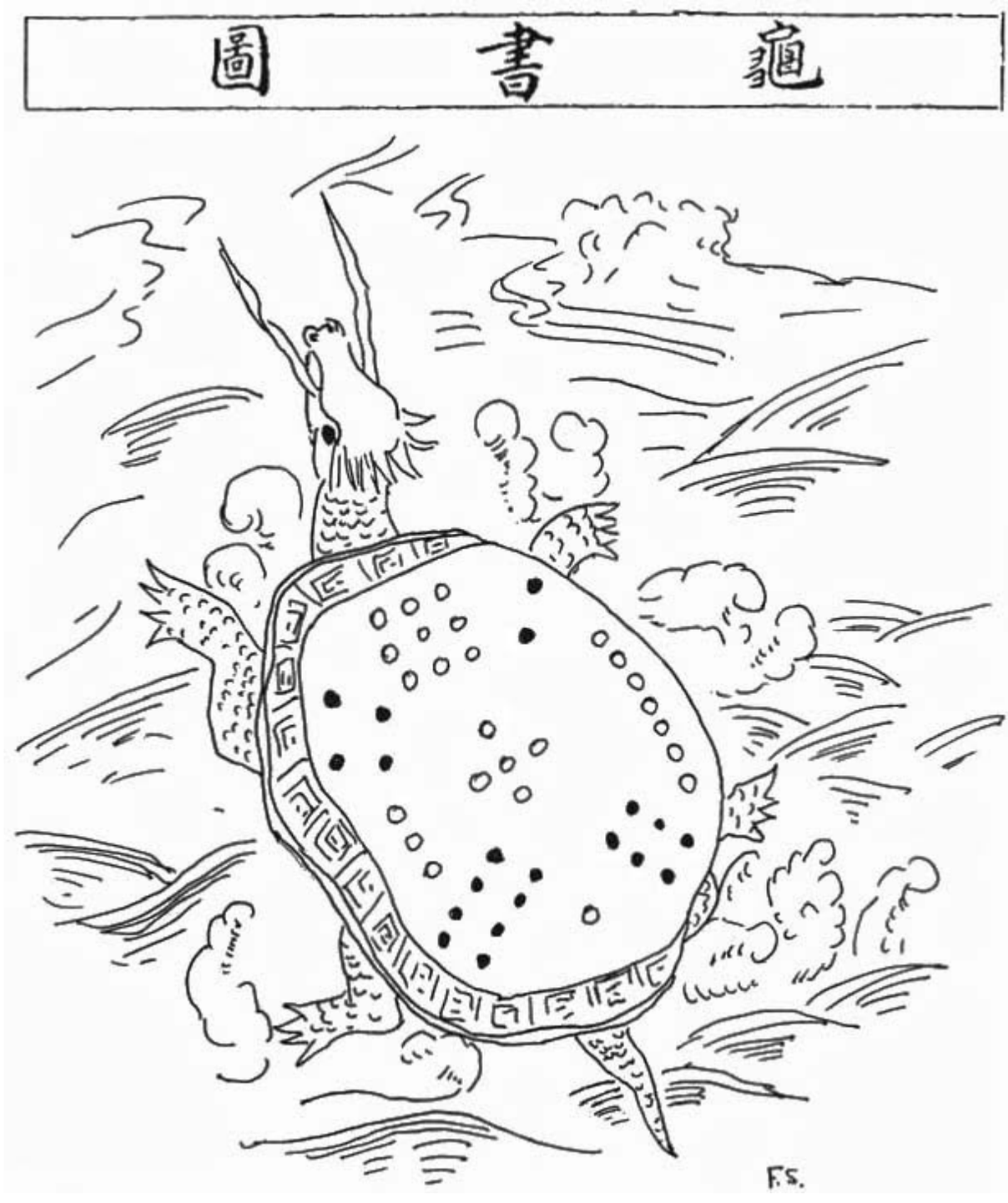


Рис.5.9. Магический квадрат Ло Шу

Когда черепаха вышла из воды, высыхали лужи после недавнего ливня. Великий Юй взял эту черепаху и рассмотрел странный узор на ее панцире. Этот узор вдохновил его на создание трактата под названием «Хун Фань» («Великий план»), в котором говорилось о физике, астрологии, предсказаниях, морали, политике и религии [53].

Магические квадраты широко применялись для передачи секретной информации. При шифровании исходное сообщение вписывалось в квадрат по приведенной в них нумерации, после чего шифрограмма выписывалась по строкам. Количество возможных магических квадратов (ключей) быстро возрастает с увеличением их размера. Так, существует лишь один магический квадрат размером 3x3, если не принимать во внимание его повороты. Магических квадратов 4x4 насчитывается уже 880, а число магических квадратов размером 5x5 около 250000. Поэтому магические квадраты больших размеров могли быть хорошей основой для надежной системы шифрования того времени, потому что ручной перебор всех вариантов ключа для этого шифра был немыслим [17].

Рассмотрим квадрат размером 4x4. В него вписываются числа от 1 до 16. Его магия состоит в том, что сумма чисел по строкам, столбцам и полным диагоналям равняется одному и тому же числу - 34.

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

Рис.5.10. Магический квадрат 4х4

Шифрование по магическому квадрату производилось следующим образом. Например, требуется зашифровать фразу: «АБРАМОВДЯДИНА...». Буквы этой фразы вписываются последовательно в квадрат согласно записанным в них числам: позиция буквы в предложении соответствует порядковому числу. В пустые клетки ставится точка или любая буква.

16 .	3 Р	2 Б	13 А
5 М	10 Д	11 И	8 Д
9 Я	6 О	7 В	12 Н
4 А	15 .	14 .	1 А

Рис.5.11. Пример шифрования с помощью магического квадрата

После этого шифрованный текст записывается в строку (считывание производится слева-направо сверху-вниз, построчно) – «.РБАМДИДЯОВНА..А».

---

[2]Джелорамо Кардано (1501 – 1576 гг.) - итальянский математик, инженер, философ, медик и астролог. В его честь названы открытые Сципионом дель Ферро формулы решения кубического уравнения (Кардано первым их опубликовал) и карданный вал (известного ещё Леонардо да Винчи). Написал около 240 книг (131 из них была опубликована, 111 остались в виде рукописей) [43].

5.3. Шифры множественной перестановки

В данном подклассе шифров используется идея повторного шифрования уже зашифрованного сообщения или многократной перестановки символов исходного сообщения перед попаданием в итоговую шифрограмму.

Шифр двойной перестановки. В таблицу по определенному маршруту записывается текст сообщения, затем переставляются столбцы, а потом переставляются строки. Шифрограмма выписывается по определенному маршруту.

Пример шифрования сообщения «АБРАМОВ+ДЯДИНА» показан на следующем рисунке. Результат шифрования – «ОАБЯ+_АИВ_РДМНАД».

Рис.5.12. Пример использования шифра двойной перестановки

Ключом к шифру являются размеры таблицы, маршруты вписывания и выписывания, а также порядки перестановки столбцов и строк. Если маршруты являются фиксированными величинами, то количество ключей равно **n!*m!**, **n** и **m** – количество столбцов и строк в таблице.

Несмотря на многоступенчатую процедуру шифрования, включая двойную перестановку, данный шифр может быть эквивалентно заменен шифром простой одинарной перестановки. На следующем рисунке приведена таблица эквивалентных одинарных перестановок для примера шифрования, приведенного на рис. 5.12.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
А	Б	Р	А	М	О	В	+	Д	Я	Д	И	Н	А	_	_
15	3	11	7	13	1	9	5	16	4	12	8	14	2	10	6
О	А	Б	Я	+	_	А	И	В	_	Р	Д	М	Н	А	Д

Рис.5.13. Таблица эквивалентных одинарных перестановок

Аналогичную замену на шифр блочной одинарной перестановки можно выполнить и для других шифров: табличная маршрутная перестановка, «поворотная решетка», «магический квадрат» и др.

Вопросы для самопроверки

1. В чем заключается основная идея криптографических преобразований шифров перестановки?

2. Перечислите основные разновидности шифров перестановки.

3. Дайте характеристику разновидностям шифров перестановки.

6. ШИФРЫ ГАММИРОВАНИЯ

Шифры гаммирования (аддитивные шифры) являются самыми эффективными с точки зрения стойкости и скорости преобразований (процедур зашифрования и дешифрования). По стойкости данные шифры относятся к классу совершенных. Для зашифрования и дешифрования используются элементарные арифметические операции – открытое/зашифрованное сообщение и гамма, представленные в числовом виде, складываются друг с другом по модулю (**mod**). Напомним, что результатом сложения двух целых чисел по модулю является остаток от деления (например, 5+10 mod 4 = 15 mod 4 = 3).

В литературе шифры этого класса часто называют потоковыми, хотя к потоковым относятся и другие разновидности шифров. В шифрах гаммирования может использоваться сложение по модулю N (общий случай) и по модулю 2 (частный случай, ориентированный на программно-аппаратную реализацию).

Сложение по модулю N. В 1888 г. француз маркиз де Виари в одной из своих научных статей, посвященных криптографии, доказал, что при замене букв исходного сообщения и ключа на числа справедливы формулы:

$$C_i = (P_i + K_i) \bmod N, \qquad (6.1)$$

$$P_i = (C_i + N - K_i) \bmod N, \qquad (6.2)$$

где P_i, C_i - i-ый символ открытого и шифрованного сообщения;
N - количество символов в алфавите;
K_i - i-ый символ гаммы (ключа).

Данные формулы позволяют выполнить зашифрование / расшифрование по Виженеру при замене букв алфавита числами согласно следующей таблице (применительно к русскому алфавиту):

Таблица 6.1. Таблица кодирования символов

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

Например, для шифрования используется русский алфавит (N = 33), открытое сообщение – «АБРАМОВ», гамма – «ЖУРИХИН». При замене символов на числа буква **А** будет представлена как 0, **Б** – 1, …, **Я** – 32. Результат шифрования показан в следующей таблице.

Таблица 6.2. Пример аддитивного шифрования по модулю N = 33

Символ								
	открытого сообщения, P_i	А	Б	Р	А	М	О	В
		0	1	17	0	13	15	2
	гаммы, K_i	Ж	У	Р	И	Х	И	Н
		7	20	17	9	22	9	14
	шифрограммы, C_i	Ж	Ф	Б	И	В	Ч	П
		7	21	1	9	2	24	16

Сложение по модулю 2 (шифр Вернама). Значительный успех в криптографии связан с именем американца Гильберто Вернама [15]. В 1917 г. он, будучи сотрудником телеграфной компании AT&T, совместно с Мейджором Джозефом Моборном предложил идею автоматического шифрования телеграфных сообщений. Речь шла о своеобразном наложении гаммы на знаки алфавита, представленные в соответствии с телетайпным кодом Бодо пятизначными «импульсными комбинациями». Например, буква **А** представлялась комбинацией («– – – + +»), а комбинация («+ + – + +») означала перехода от букв к цифрам. На бумажной ленте, используемой при работе телетайпа, знаку «+» соответствовало наличие отверстия, а знаку «–» - его отсутствие. При считывании с ленты металлические щупы проходили через отверстия, замыкали электрическую цепь и, тем самым, посылали в линию импульс тока.

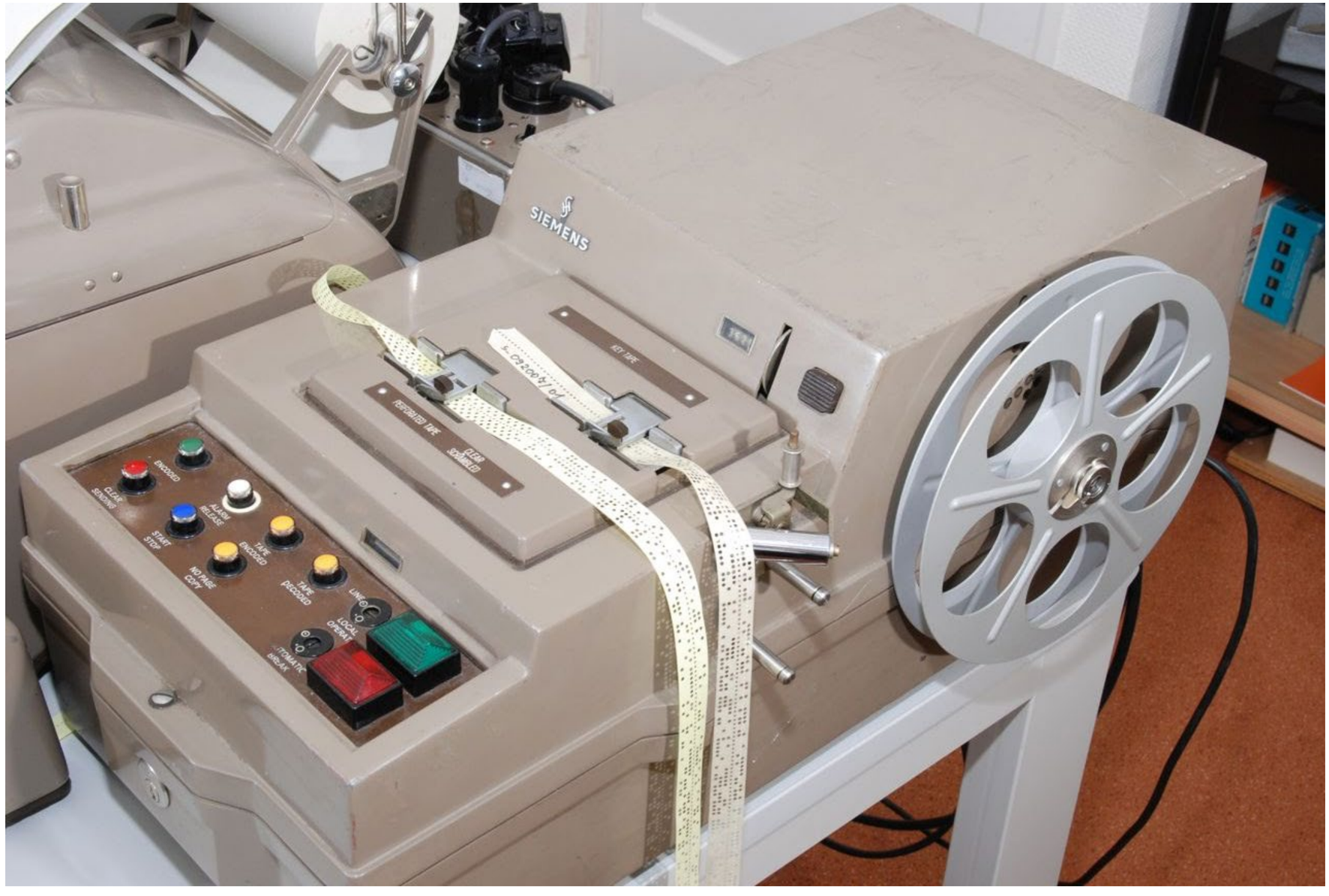


Рис.6.1. Шифровальная машина Siemens M-190 [www.cryptomuseum.com]

Вернам предложил электромеханически покоординатно складывать «импульсы» знаков открытого текста с «импульсами» гаммы, предварительно нанесенными на ленту. Сложение проводилось «по модулю 2» (⊕, для булевых величин аналог этой операции – XOR, «Исключающее ИЛИ»). Имеется в виду, что если «+» отождествить с 1, а «–» с 0, то сложение определяется двоичной арифметикой:

⊕	0	1
0	0	1

1	1	0

Таким образом, при данном способе шифрования символы текста и гаммы представляются в двоичном виде, а затем каждая пара двоичных разрядов складывается по модулю 2. Процедуры шифрования и дешифрования выполняются по следующим формулам:

$$C_i = P_i \oplus K_i, \qquad (6.3)$$

$$P_i = C_i \oplus K_i. \qquad (6.4)$$

Вернам сконструировал и устройство для такого сложения. Замечательно то, что процесс шифрования оказался полностью автоматизированным. Кроме того, оказались слитыми воедино процессы зашифрования / расшифрования и передачи по каналу связи.

В 1918 г. два комплекта соответствующей аппаратуры были изготовлены и испытаны. Испытания дали положительные результаты. Единственное неудовлетворение специалистов - криптографов было связано с гаммой. Дело в том, что первоначально гамма была нанесена на ленту, склеенную в кольцо. Несмотря на то, что знаки гаммы на ленте выбирались случайно, при зашифровании длинных сообщений гамма регулярно повторялась. Этот недостаток так же отчетливо осознавался, как и для шифра Виженера. Уже тогда хорошо понимали, что повторное использование гаммы недопустимо даже в пределах одного сообщения. Попытки удлинить гамму приводили к неудобствам в работе с большим кольцом. Тогда был предложен вариант с двумя лентами, одна из которых шифровала другую, в результате чего получалась гамма, имеющая длину периода, равную произведению длин исходных периодов.

Шифры гаммирования стали использоваться немцами в своих дипломатических представительствах в начале 1920-х гг., англичанами и американцами – во время Второй мировой войны. Разведчики-нелегалы ряда государств использовали **шифрблокноты**[1]. Шифр Вернама применялся на правительственной «горячей линии» между Вашингтоном и Москвой, где ключевые материалы представляли собой перфорированные бумажные ленты, производившиеся в двух экземплярах [23].

Рис.6.2. Одноразовый шифровальный блокнот (СССР, ГДР, 1960-е гг.) [www.cryptomuseum.com]

Перед иллюстрацией использования шифра приведем таблицу кодов символов Windows 1251 и их двоичное представление.

Таблица 6.3. Коды символов Windows 1251 и их двоичное представление

Буква	Dec-код	Bin-код	Буква	Dec-код	Bin-код	Буква	Dec-код	Bin-код
А	192	1100 0000	Л	203	1100 1011	Ц	214	1101 0110

Б	193	1100 0001	М	204	1100 1100	Ч	215	1101 0111
В	194	1100 0010	Н	205	1100 1101	Ш	216	1101 1000
Г	195	1100 0011	О	206	1100 1110	Щ	217	1101 1001
Д	196	1100 0100	П	207	1100 1111	Ъ	218	1101 1010
Е	197	1100 0101	Р	208	1101 0000	Ы	219	1101 1011
Ж	198	1100 0110	С	209	1101 0001	Ь	220	1101 1100
З	199	1100 0111	Т	210	1101 0010	Э	221	1101 1101
И	200	1100 1000	У	211	1101 0011	Ю	222	1101 1110
Й	201	1100 1001	Ф	212	1101 0100	Я	223	1101 1111
К	202	1100 1010	Х	213	1101 0101			

Примечание. Dec-код – десятичный код символа, Bin-код – двоичный код символа.

Пример шифрования сообщения «ВОВА» с помощью ключа «ЮЛЯ» показан в следующей таблице. Так как длина ключа меньше длины открытого сообщения, то для генерации гаммы он циклически повторяется.

Таблица 6.4. Пример аддитивного шифрования по модулю 2

Открытое сообщение, P_i	Буква	В	О	В	А
	Dec-код	194	206	194	192
	Bin-код	1100 0010	1100 1110	1100 0010	1100 0000
Гамма, K_i	Буква	Ю	Л	Я	Ю
	Dec-код	222	203	223	222
	Bin-код	1101 1110	1100 1011	1101 1111	1101 1110
Шифрограмма, C_i	Dec-код	28	5	29	30
	Bin-код	0001 1100	0000 0101	0001 1101	0001 1110

Шифрование по модулю 2 обладает замечательным свойством - вместо истинной гаммы противнику можно сообщить ложную гамму, которая при наложении на шифрограмму даст осмысленное выражение.

Таблица 6.5. Пример использования ложной гаммы

Шифрограмма, C_i	Dec-код	28	5	29	30
	Bin-код	0001 1100	0000 0101	0001 1101	0001 1110
Ложная гамма, K'_i	Буква	Ю	Е	М	Б
	Dec-код	222	197	204	193
	Bin-код	1101 1110	1100 0101	1100 1100	1100 0001
Ложное открытое сообщение, P'_i	Dec-код	194	192	209	223
	Bin-код	1100 0010	1100 0000	1101 0001	1101 1111
	Буква	В	А	С	Я

Пример, представленный в таблице, иллюстрирует совершенность шифров гаммирования. Для перехваченной шифрограммы противник может подобрать большое количество гамм, дающих при расшифровании осмысленные сообщения и не имеющих ничего общего с истинным открытым сообщением. Более того, под любое ложное открытое сообщение найдется ложная гамма.

Стойкость аддитивных шифров определяется, главным образом, качеством гаммы, которое зависит от длины периода и случайности распределения по периоду [8].

Длиной периода гаммы называется минимальное количество символов, после которого последовательность начинает повторяться. **Случайность распределения символов по периоду** означает отсутствие закономерностей между появлением различных

символов в пределах периода.

Для обеспечения абсолютной стойкости необходимо, чтобы последовательность символов в пределах периода гаммы обладала следующими свойствами:

- была случайной (должна отсутствовать закономерность в появлении символов гаммы);

- символы алфавита гаммы были распределены нормально (равновероятно);

- совпадала по размеру или была больше исходного открытого текста;

- применялась только один раз.

Первые два свойства (требования) проверяются с помощью различных тестов. Наиболее популярный и авторитетный из них «Набор статистических тестов для генераторов случайных и псевдослучайных чисел для криптографических приложений» (англ. NIST SP 800-22 «A statistical test suite for random and pseudorandom number generators for cryptographic applications»), включающий в себя 15 тестов.

---

[1]**Классический одноразовый шифровальный блокнот** - большой неповторяющийся случайный набор символов ключа, написанный на листах бумаги, склеенных в блокнот. Шифровальщик при личной встрече снабжался блокнотом, каждая страница которого содержала ключ. Такой же блокнот имелся и у принимающей стороны. Использованные страницы после однократного использования уничтожались.

6.2. Генерация случайных последовательностей

Несмотря на все достоинства шифров гаммирования, одной из ключевых проблем их применения на практике является получение качественных гамм. Для процедур зашифрования/дешифрования можно использовать истинно случайные или псевдослучайные гаммы (последовательности). **Псевдослучайная последовательность** – последовательность чисел, которая была вычислена по определённой процедуре, но имеет все свойства случайной последовательности чисел в рамках решаемой задачи. Отличие истинно случайных последовательностей от псевдослучайных заключается в невозможности предсказания (расчета, определения) символов в ней. Таким образом, любой алгоритмически устроенный программно-аппаратный комплекс не может генерировать истинно случайные последовательности, т.к. он работает по строго определенным правилам, а значит результат (гамма) предсказуем. Как сказал Джон фон Нейман, «всякий, кто питает слабость к арифметическим методам получения случайных чисел, грешен вне всяких сомнений» [63].

Истинно случайные гаммы могут быть получены путем оцифровки случайных физических или антропогенных процессов.

Псевдослучайные гаммы получают путем применения рекуррентных формул[2] или полноценных алгоритмов. При этом отсутствие истинной случайности не мешает получать криптографически стойкие последовательности, в т.ч. и с бесконечным периодом.

---

[2]**Рекуррентная формула** - формула вида $a_n = f(n, a_{n-1}, a_{n-2}, \ldots, a_{n-p})$, определяющая каждый член последовательности a_n, как функцию от p предыдущих членов и возможно номера члена последовательности n.

6.2.1. Генерация истинно случайных последовательностей

Как было отмечено выше, генерация истинно случайных последовательностей возможна путем оцифровки случайных физических или антропогенных процессов.

Среди **физических процессов**, которые рассматривались в качестве источника случайных последовательностей, можно выделить следующие [17, 64]:

- **дробовый шум** – беспорядочные изменения (флуктуации) принимаемого или передаваемого сигнала относительно его среднего значения, вызванные дискретностью потоков фотонов и электронов в оптических и радиоэлектронных устройствах;

- **тепловой шум** – равновесный шум, обусловленный тепловым движением носителей заряда в проводнике, в результате чего на концах проводника возникает флуктуирующая разность потенциалов. В микропроцессорах Intel с архитектурой Ivy Bridge для аппаратной генерации случайных последовательностей используется тепловой шум в кристаллах кремния [65]. Для усиления энтропии[3] поток битов проходит дополнительную фильтрацию и шифрование AES в режимах CBC и CRT;

Рис.6.3. Микропроцессор Core i7-3770 с архитектурой Ivy Bridge и инструкцией генерации случайных чисел RdRand

- движение жидкостей в лавовых лампах. Компания CloudFlare, через сеть которой проходит до 10% траффика Интернета, в качестве одного из источников случайных последовательностей использует стену с сотней лавовых ламп;

Рис.6.4. «Стена энтропии» в компании CloudFlare

- движение ленточек в потоке воздуха вентилятора;

Рис.6.5. Хаотичное движение ленточек в потоке воздуха

- **радиоактивный распад** – спонтанное изменение состава или внутреннего строения нестабильных атомных ядер путём испускания элементарных частиц, гамма-квантов и/или ядерных фрагментов. Компания CloudFlare в качестве одного из источников случайных последовательностей использует радиоактивный элемент;

Рис.6.6. Счетчик Гейгера

- **электромагнитное излучение** – электромагнитные волны, возникающие при возмущении магнитного или электрического поля. Генераторы случайных последовательностей на базе квантовых явлений предлагает большое количество компаний (ID Quantique, QuintessenceLabs, ComScire и др.);

Рис.6.7. Квантовый генератор случайных чисел [www.idquantique.com/random-number-generation/overview]

- **лавинный пробой в полупроводниках** – электрический пробой p-n-перехода, вызванный лавинным размножением носителей заряда под действием сильного электрического поля;

Рис.6.8. Устройство ChaosKey [altusmetrum.org/ChaosKey]

- нестабильная частота свободно работающего осциллятора. Компания RAND использовала это явление для публикации в 1955 г. книги «Миллион случайных цифр со стандартным отклонением 100 000», а компания AT&T в 1986 г. представила коммерческую микросхему-генератор [66].

К **антропогенным процессам**, используемых в качестве источника случайных последовательностей, можно отнести следующие:

- время между нажатиями клавиш;

- движения компьютерной мыши;

- изменение скорости вращения жесткого диска, вызванного турбулентностью воздуха [8];

- время между принятыми пакетами в сети (например, пинг сайта Google).

Для большинства рассмотренных выше генераторов на базе физических и антропогенных процессов характерны определенные **недостатки**:

- необходимость передачи гамм всем участникам обмена данными. Так как гамма генерируется в одном месте и она случайна, то принимающая данные сторона для расшифровки предварительно должна получить эту гамму (проблема обмена ключами);

- медленная скорость генерации числовых последовательностей;

- антропогенные процессы могут иметь скрытые зависимости – каждый пользователь обладает своим «подчерком» работы с компьютером.

В заключение обзора генераторов истинно случайных последовательностей рассматривается оригинальный метод, предложенный в 2013 г. учеными Корнельского университета [42]. Кратко принцип нового метода шифрования заключается в том, что отправитель и получатель (Алиса и Боб) во время встречи формируют общий ключ, облучая кусочки стекла изображением-паттерном с ID_i (внешне он напоминает QR-код). В результате отражения и преломления, характер которого индивидуален для каждого куска стекла, у Алисы и Боба получаются собственные случайные изображения, которые затем оцифровываются (получаются ключи $keyA_i$ и $keyB_i$). Из этих изображений и составляется общий ключ ($keyAB_i = keyA_i \oplus keyB_i$).

Схема генерации одного общего ключа показана на следующем рисунке.

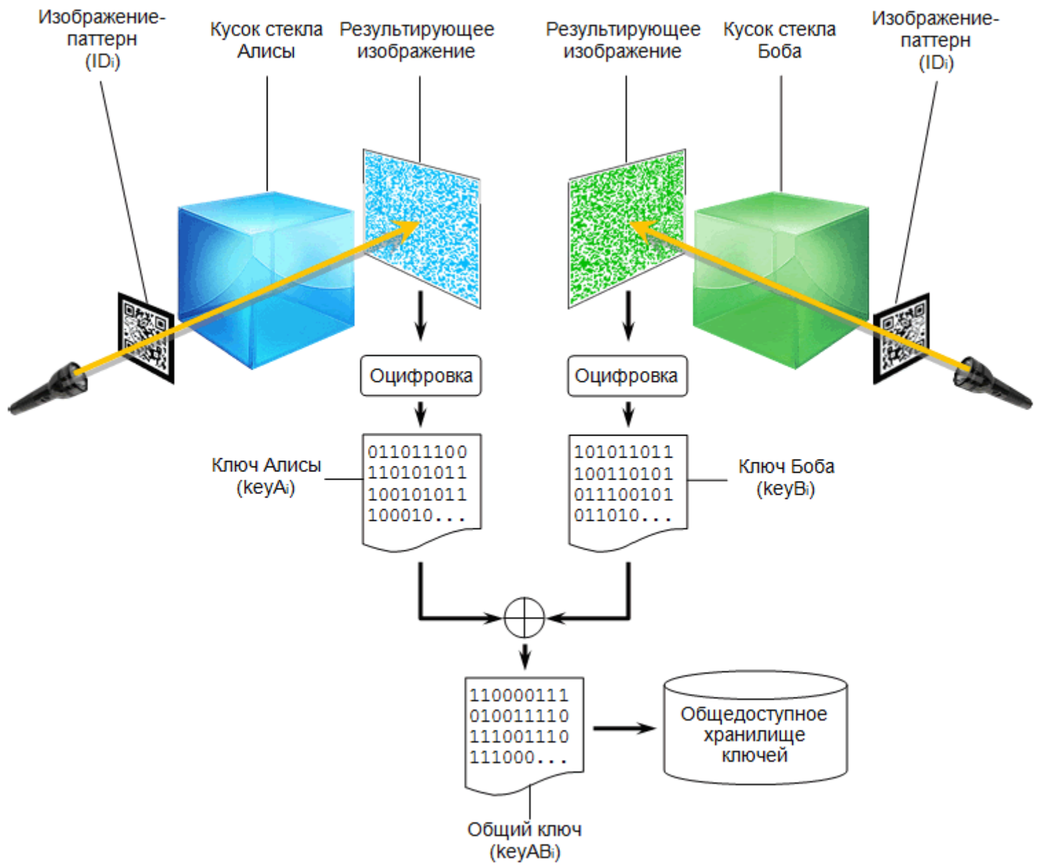


Рис.6.9. Схема генерации одного общего ключа

Как и в классической системе Вернама, каждый случайный паттерн (ключи $keyA_i$, $keyB_i$ и $keyAB_i$) можно использовать лишь однажды. В связи с этим Алиса и Боб должны сформировать достаточное количество ключей, облучая свои куски стекла разными паттернами. После генерации общие ключи $keyAB_i$ помещаются в общедоступное хранилище.

Процедура обмена шифрограммами выглядит следующим образом.

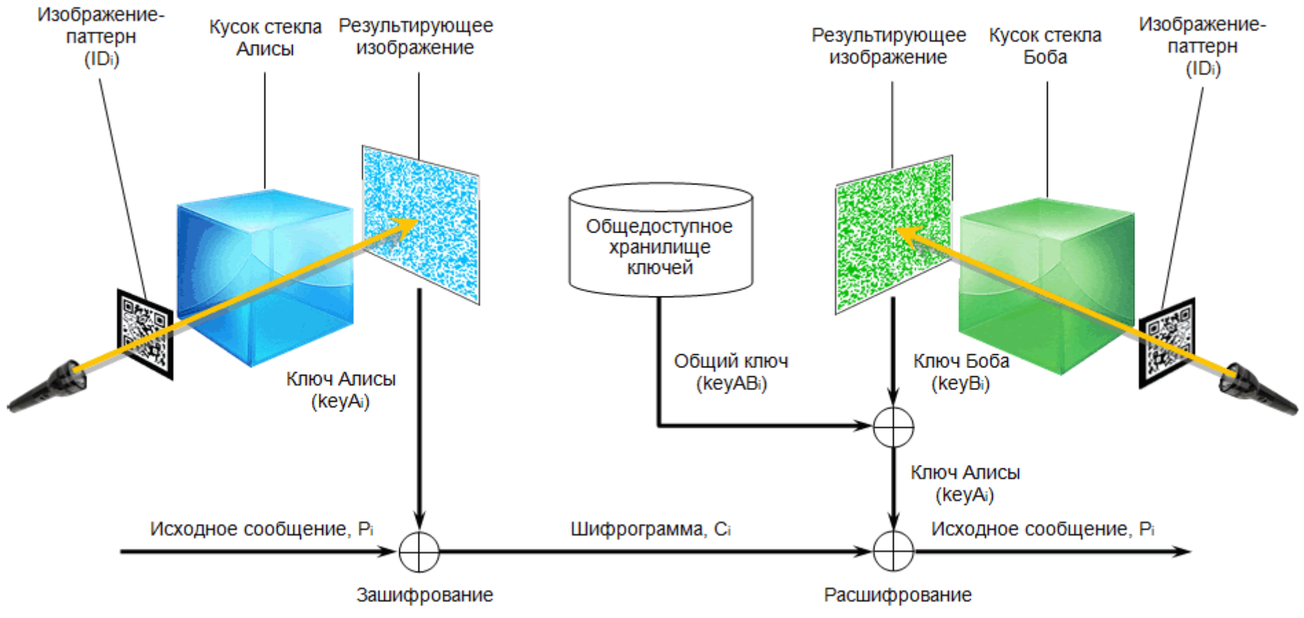


Рис.6.10. Схема зашифрования и дешифрования сообщения

1) Алиса выбирает паттерн с ID_i, облучает свой кусочек стекла и оцифровывает полученное изображение, получая ключ $keyA_i$.

2) Исходное сообщение P_i Алиса складывает по модулю 2 с ключом $keyA_i$ ($C_i = P_i \oplus keyA_i$) и отсылает шифрограмму C_i с идентификатором паттерна ID_i Бобу.

3) Боб по полученному идентификатору ID_i из общедоступного хранилища считывает общий ключ $keyAB_i$ и, облучая паттерн с ID_i и свой кусочек стекла, генерирует собственный ключ $keyB_i$.

4) Складывая по модулю 2 ключи $keyAB_i$ и $keyB_i$, Боб получает ключ $keyA_i$ ($keyA_i = keyAB_i \oplus keyB_i$).

5) Для прочтения исходного сообщения P_i Боб складывает по модулю 2 шифрограмму C_i и ключ $keyA_i$ ($P_i = C_i \oplus keyA_i$).

По утверждению авторов схемы, взломать шифр Вернама можно только украв сами кусочки полупрозрачного стекла. Однако даже в этом случае у Алисы и Боба будет около 24 часов на определение факта кражи, поскольку злоумышленнику потребуется определить структуру стеклянного образца. Другие возможные проблемы при использовании данной схемы:

- потеря кусочка стекла;

- появление сколов и царапин на стекле;

- необходимость точного взаимного расположения и ориентации источника света, изображения-паттерна, кусочка стекла и полотна с результирующим изображением при оцифровке гаммы.

---

[3]**Информационная энтропия** – мера неопределённости или непредсказуемости информации, неопределённость появления какого-либо символа первичного алфавита.

6.2.2. Генерация псевдослучайных последовательностей

Генераторы псевдослучайных последовательностей получили наибольшее распространение. По схеме использования они делятся на синхронные и самосинхронизирующиеся [8, 23].

Схема шифрования с использованием **синхронных генераторов** представлена на следующем рисунке.

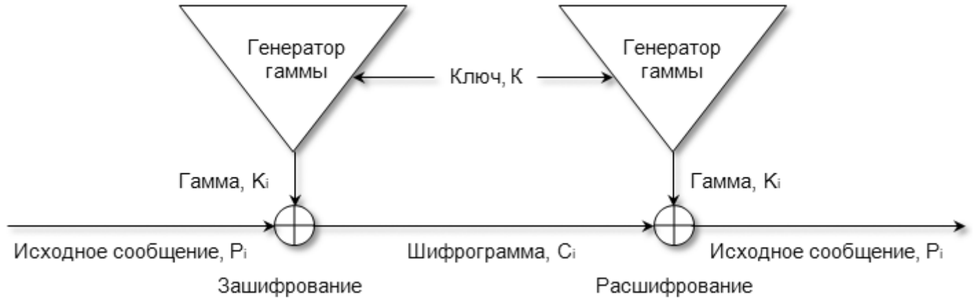


Рис.6.11. Схема шифрования с использованием синхронных генераторов

При этом генератор гаммы, как правило, состоит из трех основных блоков.

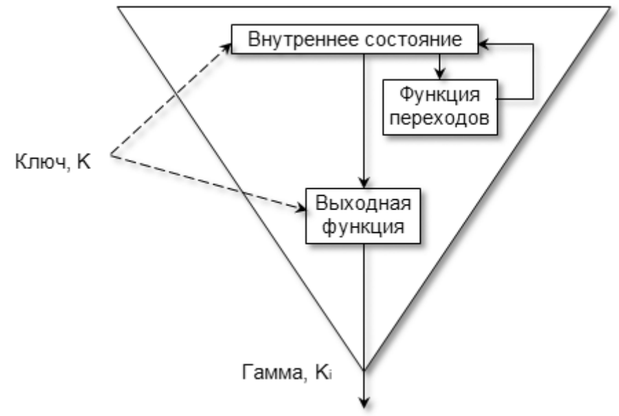


Рис.6.12. Устройство генератора гаммы с внутренней обратной связью

Внутреннее состояние описывает текущее состояние генератора гаммы. **Начальное внутреннее состояние**, как правило, определяется ключом **K**. Два генератора, с одинаковым ключом и одинаковым внутренним состоянием, создают одинаковые гаммы. **Функция переходов** считывает текущее внутреннее состояние и генерирует новое внутреннее состояние. **Выходная функция** считывает внутреннее состояние и генерирует бит (биты) гаммы $\mathbf{K_i}$.

В другой разновидности, так называемых **генераторах типа счетчик**, отсутствует блок с функцией переходов. В отличие от генераторов с обратной связью, они позволяют вычислить i-й бит гаммы, не вычисляя всех предыдущих битов. Для этого генератор устанавливается в i-е внутреннее состояние, после чего вычисляется соответствующий ему i-й бит гаммы. Это свойство полезно использовать для обеспечения произвольного доступа к файлам данных, что позволяет расшифровать отдельный фрагмент данных, не расшифровывая файл полностью.

В синхронном генераторе гамма генерируется независимо от потока сообщения. На шифрующей стороне генератор гаммы последовательно выдает биты гаммы $\mathbf{K_i}$. На расшифровывающей стороне другой генератор гаммы один за другим выдает идентичные биты гаммы. Эта схема работает нормально, если оба генератора синхронизированы.

Недостаток синхронного генератора. Если один из генераторов пропускает один из циклов или бит шифрограммы теряется при передаче, то все символы шифрограммы, следующие за ошибкой, расшифровываются некорректно. В этом случае отправитель и получатель должны синхронизировать генераторы и заново передать некорректно расшифрованную часть сообщения.

Достоинство синхронного генератора. Отсутствие распространения ошибок. Если во время передачи бит $\mathbf{C_i}$ изменит свое значение, что намного вероятнее его потери, то некорректно расшифровывается только один измененный бит.

В **самосинхронизирующемся генераторе** каждый бит гаммы представляет собой функцию фиксированного числа предыдущих битов шифрограммы. Используемые при таком шифровании генераторы гаммы называют **генераторами с обратной связью по шифрограмме (шифртексту)**.

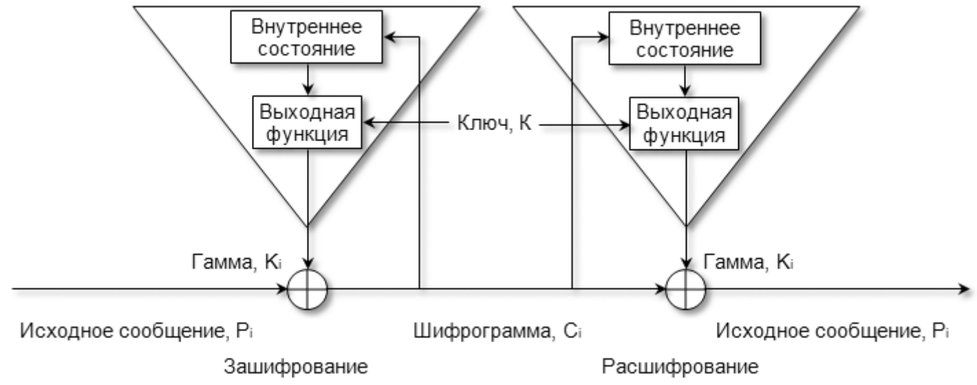


Рис.6.13. Схема шифрования с использованием самосинхронизирующихся генераторов гаммы

Внутреннее состояние зависит от **n** предыдущих битов шифрограммы. Каждое сообщение начинается случайным заголовком (**вектор инициализации, синхропосылка**) длиной **n** бит, после прохождения которого оба генератора гаммы синхронизируются.

Недостатки самосинхронизирующегося генератора.

1. Распространение ошибки. Для каждого бита шифрограммы, искаженного при передаче, расшифровывающий генератор выдает **n** некорректных битов гаммы. Следовательно, измененный бит влияет на внутреннее состояние, каждой ошибке шифрограммы будет соответствовать **n** ошибок открытого текста.

2. При потере бита $\mathbf{C_i}$ необходимо заново передать часть сообщения, но в отличие от синхронных генераторов, синхронизация намного проще.

Для генерации псевдослучайных последовательностей используют рекуррентные формулы или полноценные алгоритмы. В первом случае члены числовой последовательности не только рассчитываются рекуррентно, но и впоследствии становятся частью гаммы. Во втором случае для генерации гаммы используются более сложные правила, в т.ч. для повышения энтропии гаммы могут применять хеш-функции и шифрование.

В качестве иллюстрации ниже приведены некоторые из них.

А) Рекуррентные формулы:

- линейный конгруэнтный генератор;

- инверсный конгруэнтный генератор;

- аддитивный генератор;

- регистр сдвига с линейной обратной связью.

А.1) Линейный конгруэнтный генератор – генератор псевдослучайных чисел, в котором новый член последовательности рассчитывается на базе предыдущего через линейную зависимость. В общем случае линейный конгруэнтный генератор задается выражением:

$$X_{i+1} = (a\,X_i + b) \bmod m, \qquad (6.5)$$

где a, b и m – некоторые коэффициенты.

Агоритм RANDU - один из вариантов линейного конгруэнтного генератора псевдослучайных чисел, десятилетиями использовавшийся на мейнфреймах. Он определяется рекуррентным соотношением:

$$X_{i+1} = (65539\,X_i) \bmod 2^{31}, \qquad (6.6)$$

где X_0 - нечётное.

Пример псевдослучайной последовательности, порождаемой алгоритмом RANDU при начальном значении $X_0 = 1$:

65 539
393 225
1 769 499
7 077 969
26 542 323
...
388 843 697
238 606 867
79 531 577
477 211 307
1 (повтор для элемента № 536 870 913)

.

К сожалению, линейные конгруэнтные генераторы нельзя использовать в криптографии, т.к. они предсказуемы. Впервые линейные конгруэнтные генераторы были взломаны Джимом Ридсом, а затем Джоан Бояр [8]. Ей удалось также вскрыть **квадратичные генераторы**

$$X_{i+1} = (a X_i^2 + b X_i + c) \bmod m \qquad (6.7)$$

и **кубические генераторы**

$$X_{i+1} = (a X_i^3 + b X_i^2 + c X_i + d) \bmod m. \qquad (6.8)$$

A.2) Инверсный конгруэнтный генератор (генератор Эйхенауэра – Лена) – генератор псевдослучайных чисел, в котором новый член последовательности рассчитывается как обратное число к предыдущему по модулю. Общая формула генератора выглядит следующим образом:

$$X_{i+1} = (a X_i^{-1} + b) \bmod m, \qquad (6.9)$$

где a, b и m – некоторые коэффициенты, $0 \le a < m$, $0 \le b < m$.

Как и в случае с линейным конгруэнтным генератором получаемая последовательность периодична с максимальным периодом m. Данные генераторы считаются более стойкими, но требует значительных вычислений.

A.3) Аддитивный генератор (запаздывающий генератор Фибоначчи) – генератор псевдослучайных чисел, в котором новый член последовательности зависит более чем от одного из предшествующих. Одна из ранних реализаций генератора последовательности Фибонначи с запаздыванием была предложена 1958 г. Дж. Ж. Митчелом и Д. Ф. Муром:

$$X_{i+1} = (X_{i-a} + X_{i-b}) \bmod m, \qquad (6.10)$$

где a = 24, b = 55 – запаздывание, $b > a \ge 1$;
X_0 .. X_{54} – произвольные целые числа;
m – чётное число.

Более сложные реализации аддитивных генераторов применяются в алгоритмах Fish, Pike, Mush и др.

A.4) Регистр сдвига с линейной обратной связью (РСЛОС) – упорядоченный набор битов, у которого значение входного (вдвигаемого слева, старшего) бита равно линейной булевой функции от значений остальных битов регистра до сдвига. Теорию последовательности регистров сдвига разработал в 1965 г. главный криптограф норвежского правительства Эрнст Селмер [8].

Длиной регистра называется количества битов в нем. В качестве булевой функции для РСЛОС чаще всего используют сложение по модулю 2. Такие РСЛОС обычно записывают в виде многочлена (полинома) или упорядоченной последовательности. Запись $x^8 + x^4 + x^3 + x^2 + 1$ или (8, 4, 3, 2, 0) означает, что длина регистра 8 битов, а входной бит рассчитывается по формуле $b_7 = b_4 \oplus b_3 \oplus b_2 \oplus b_0$ (для битов нумерация начинается справа и с нулевой позиции). Выходной (выдвигаемый справа, младший) бит будет являться частью генерируемой гаммы. Расчет требуемой гаммы осуществляется в цикле, на каждой итерации которого выполняются следующие операции.

1. Добавление выходного бита b_0 к гамме.

2. Расчет входного бита b_{n-1} по формуле.

3. Сдвиг битов регистра вправо на одну позицию.

4. Занесение рассчитанного входного бита b_{n-1} в позицию n-1.

В следующей таблице приведен пример генерации гаммы для регистра, инициализированного значением 10001100_2.

Таблица 6.6. Пример генерации гаммы для РСЛОС $x^8 + x^4 + x^3 + x^2 + 1$

№ п/п	Исходное состояние регистра	Бит гаммы, b_0	Входной бит, $b_7 = b_4 \oplus b_3 \oplus b_2 \oplus b_0$	Сдвиг регистра вправо на одну позицию	Занесение входного бита b_7 в регистр
0	1 0 0 0 1 1 0 0	0	$0 \oplus 1 \oplus 1 \oplus 0 = 0$	1 0 0 0 1 1 0	0 1 0 0 0 1 1 0
1	0 1 0 0 0 1 1 0	0	$0 \oplus 0 \oplus 1 \oplus 0 = 1$	0 1 0 0 0 1 1	1 0 1 0 0 0 1 1
2	1 0 1 0 0 0 1 1	1	$0 \oplus 0 \oplus 0 \oplus 1 = 1$	1 0 1 0 0 0 1	1 1 0 1 0 0 0 1
3	1 1 0 1 0 0 0 1	1	$1 \oplus 0 \oplus 0 \oplus 1 = 0$	1 1 0 1 0 0 0	0 1 1 0 1 0 0 0
4	0 1 1 0 1 0 0 0	0	$0 \oplus 1 \oplus 0 \oplus 0 = 1$	0 1 1 0 1 0 0	1 0 1 1 0 1 0 0
...	...	...	...	...	...
Гамма		00110...			

Для регистра длиной n максимальное количество состояний (период гаммы) составляет 2^n–1 (Это число равно 2^n–1, а не 2^n, поскольку заполнение РСЛОС нулями влечет вывод регистром бесконечной последовательности нулей, что совершенно бесполезно). Для получения такого максимального периода многочлен должен быть примитивным по модулю 2. **Примитивный многочлен степени n** – неприводимый многочлен, который является делителем $x^{2^n-1}+1$, но не является делителем x^d+1 для всех d, являющихся делителями 2^n–1. Чем больше битов регистра используются для расчета входного бита, тем лучше (повышается стойкость). Многочлены с большим количеством членов называют **плотными**, с малым – **разреженными**.

РСЛОС обладают высокой скоростью генерации чисел, хорошими статистическими свойствами, а также возможностью простой реализации на аппаратном уровне. Исследователями предложено несколько десятков различных вариантов реализации генераторов на базе РСЛОС, в т.ч. с применением комбинации нескольких РСЛОС одновременно и разных функций расчета входного или выходного бита (регистры сдвига с нелинейной обратной связью, обобщенной обратной связью, обратной связью по переносу и т.д.).

Б) Алгоритмы:

- алгоритм Блюм - Блюм - Шуба;

- RC4;

- Trivium;

- генератор на базе иррациональных чисел.

Б.1) Алгоритм Блюм - Блюм - Шуба (англ. Algorithm Blum - Blum - Shub, BBS) предложен в 1986 г. Ленор Блюм, Мануэлем Блюмом и Майклом Шубом.

Рекуррентная формула BBS выглядит следующим образом:

$$X_{i+1} = X_i^2 \bmod m, \qquad (6.11)$$

где $m = p * q$ – является произведением двух больших простых p и q, сравнимых с 3 по модулю 4.

На каждом шаге алгоритма гамма получаются из X_i путём взятия либо паритетного бита, либо одного или более младших битов X_i. Начальное значение X_0 должно быть **взаимно простым**[4] с m – в противном случае у генерируемой последовательности быстро появляется период.

Пример с использованием двух малых простых чисел.

p = 7 (7 mod 4 = 3) и q = 19 (19 mod 4 = 3).
m = 7 * 19 = 133.
X_0 = 53.

Таблица 6.7. Пример генерации гаммы по алгоритму BBS

№ п/п	X		Четный паритетный бит	Младший бит	2 младших бита
	Dec-код	Bin-код			
0	53	110101	0	1	01
1	16	10000	1	0	00
2	123	1111011	0	1	11
3	100	1100100	1	0	00
4	25	11001	1	1	01
…	…	…	…	…	…
Гамма			01011…	10101…	0100110001…

Как было отмечено выше, X_0 должно быть взаимно простым с m. Если для генерации последовательности с параметрами p = 7, q = 19 и m = 133 выбрать X_0 = 57 = 3 * 19, то она будет состоять из одних X_i = 57 (57^2 mod 133 = 57). Таким образом период последовательности будет равным 1, что не лучшим образом сказывается на качестве гаммы. Слишком малый период может возникать и в случае, когда X_0 и m взаимно просты. Например, если при тех же параметрах выбрать X_0 = 130, то последовательность будет выглядеть: 130, 9, 81, 44, 74, 23, 130, ...

Особенностью этого алгоритма является то, что для получения X_n необязательно вычислять все n - 1 предыдущих чисел, если известно начальное состояние генератора X_0 и числа p и q, то n-ное значение может быть вычислено по формуле:

$$X_n = X_0^{2^n \bmod ((p-1)(q-1))} \bmod m. \qquad (6.12)$$

Б.2) RC4 (от англ. Rivest cipher или Ron’s code) был создан сотрудником компании «RSA Security» Рональдом Ривестом в 1987 г. В течение семи лет шифр являлся коммерческой тайной, и точное описание алгоритма предоставлялось только после подписания соглашения о неразглашении, но в сентябре 1994 г. его описание было анонимно отправлено в список рассылки «Cypherpunks» [17].

Этап 1. Инициализация S-блока.

Перед генерацией гаммы выполняется инициализация **S-блока** длиной $\mathbf{L_s}$. Обычно $\mathbf{L_s}$ выбирается кратно 8 битов (2^8 = 256, 2^{16} = 65536 и т.п.). Для инициализации используется ключ **K** длиной $\mathbf{L_K}$ от 40 до 2048 битов по следующему алгоритму.

```
1. Цикл A. Для i := 0 до Ls - 1
      1.1. S[i] := i
2. j := 0
3. Цикл B. Для i := 0 до Ls - 1
      3.1. j := (j + S[i] + K[i mod LK]) mod Ls
      3.2. Поменять местами S[i] и S[j]
```

Этап 2. Генерация гаммы.

Непосредственная генерация гаммы выполняется циклически блоками по $\mathbf{L_s}$ битов до достижения необходимой длины псевдослучайной последовательности $\mathbf{L_g}$. Алгоритм генерации следующий.

```
1. i := 0
2. j := 0
3. L := 0
4. Цикл. Пока L < Lg
```

4.1. $i := (i + 1) \bmod L_S$

4.2. $j := (j + S[i]) \bmod L_S$

4.3. Поменять местами S[i] и S[j]

4.4. $t := (S[i] + S[j]) \bmod L_S$

4.5. Блок гаммы := S[t]

4.6. $L := L + L_S$

RC4 получил широкое распространение в криптосистемах и протоколах, в частности [17]:

- WEP (англ. Wired Equivalent Privacy) — алгоритм для обеспечения безопасности сетей Wi-Fi;
- WPA (англ. Wi-Fi Protected Access) — обновленный алгоритм сертификации устройств сетей Wi-Fi;
- BitTorrent protocol encryption — протоколы пиринговых файлообменных сетей;
- SSL (англ. Secure Sockets Layer) — криптографический протокол передачи данных в сети;
- Kerberos — сервер аутентификации Kerberos;
- PDF (англ. Portable Document Format) — межплатформенный формат электронных документов, разработанный фирмой Adobe Systems;
- Skype — программное обеспечение IP-телефонии;
- и др.

Обнаруженные уязвимости в стандартной реализации RC4 привели к отказу от его использования в некоторых криптосистемах и протоколах, а также появлению различных его модификаций: RC4A, RC4+, VMPC, Spritz.

Б.3) Trivium был представлен в 2008 г. как часть европейского проекта eSTREAM по профилю 2 (аппаратно ориентированные шифры) и в настоящий момент имеет статус международного стандарта «ISO/IEC 29192-3:2012. Информационные технологии - Методы безопасности - Легкая криптография - Часть 3: Поточные шифры» (англ. «ISO/IEC 29192-3:2012. Information technology - Security techniques - Lightweight cryptography - Part 3: Stream ciphers»). Авторами генератора (шифра) являются Кристоф Де Канньер и Барт Пренел [17].

По аналогии с RC4 процедура выработки гаммы состоит из двух этапов: инициализации S-блока и непосредственно генерации гаммы.

Этап 1. Инициализация S-блока.

Длина **S-блока** составляет 288 битов. При этом он условно делится на 3 части длинами 93, 84 и 111 битов. В первую часть $\mathbf{S_1}$ заносится 80-битовый ключ **K** с добавлением 13 нулевых битов, во вторую часть $\mathbf{S_2}$ заносится 80-битовый вектор инициализации **IV** с добавлением 4 нулевых битов и в последнюю часть $\mathbf{S_3}$ заносятся нулевые биты за исключением трех последних единичных битов. После первоначальной инициализации в цикле осуществляется связывание битов из разных частей со сдвигами битов вправо в каждой из них. Весь алгоритм инициализации выглядит следующим образом.

1. $S_1 = (s_1, s_2, \ldots, s_{93}) := (K_1, K_2, \ldots, K_{80}, 0, \ldots, 0)$

2. $S_2 = (s_{94}, s_{95}, \ldots, s_{177}) := (IV_1, IV_2, \ldots, IV_{80}, 0, \ldots, 0)$

3. $S_3 = (s_{178}, s_{179}, \ldots, s_{288}) := (0, \ldots, 0, 1, 1, 1)$

4. Цикл. Для i := 1 до 288

4.1. $t_1 := s_{66} \oplus (s_{91} \wedge s_{92}) \oplus s_{93} \oplus s_{171}$

4.2. $t_2 := s_{162} \oplus (s_{175} \wedge s_{176}) \oplus s_{177} \oplus s_{264}$

4.3. $t_3 := s_{243} \oplus (s_{286} \wedge s_{287}) \oplus s_{288} \oplus s_{69}$

4.4. Сдвиг первой части на один бит вправо $S_1 := (\_, s_1, s_2, \ldots, s_{92})$

4.5. Занесение t_3 в первую позицию первой части S_1 := (t_3, s_1, s_2, ..., s_{92})

4.6. Сдвиг второй части на один бит вправо S_2 := (_, s_{94}, s_{95}, ..., s_{176})

4.7. Занесение t_1 в первую позицию второй части S_2 := (t_1, s_{94}, s_{95}, ..., s_{176})

4.8. Сдвиг третьей части на один бит вправо S_3 := (_, s_{178}, s_{179}, ..., s_{287})

4.9. Занесение t_2 в первую позицию третьей части S_3 := (t_2, s_{178}, s_{179}, ..., s_{287})

Этап 2. Генерация гаммы.

Для генерации гаммы длиной $\mathbf{L_g}$ в каждой итерации цикла на основе 15 битов **S-блока** вычисляется один бит гаммы и выполняются операции, идентичные рассмотренным в инициализации **S-блока**. Алгоритм генерации следующий.

1. L := 0
2. Цикл. Пока L < L_g

2.1. Бит гаммы := $s_{66} \oplus s_{93} \oplus s_{162} \oplus s_{177} \oplus s_{243} \oplus s_{288}$

2.2. t_1 := $s_{66} \oplus (s_{91} \wedge s_{92}) \oplus s_{93} \oplus s_{171}$

2.3. t_2 := $s_{162} \oplus (s_{175} \wedge s_{176}) \oplus s_{177} \oplus s_{264}$

2.4. t_3 := $s_{243} \oplus (s_{286} \wedge s_{287}) \oplus s_{288} \oplus s_{69}$

2.5. Сдвиг первой части на один бит вправо S_1 := (_, s_1, s_2, ..., s_{92})

2.6. Занесение t_3 в первую позицию первой части S_1 := (t_3, s_1, s_2, ..., s_{92})

2.7. Сдвиг второй части на один бит вправо S_2 := (_, s_{94}, s_{95}, ..., s_{176})

2.8. Занесение t_1 в первую позицию второй части S_2 := (t_1, s_{94}, s_{95}, ..., s_{176})

2.9. Сдвиг третьей части на один бит вправо S_3 := (_, s_{178}, s_{179}, ..., s_{287})

2.10. Занесение t_2 в первую позицию третьей части S_3 := (t_2, s_{178}, s_{179}, ..., s_{287})

2.11. L := L + 1

Существуют упрощенные модификации генератора гаммы Univium, Bivium, Trivium-toy и Bivium-toy.

Б.4) Иррациональное число – вещественное число, которое не является рациональным, т.е. не может быть представлено в виде обыкновенной дроби $\pm\frac{m}{n}$, где m, n – натуральные числа. К наиболее известным иррациональным числам относятся $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$, ln2, e и π [17].

Иррациональное число может быть представлено в виде бесконечной непериодической десятичной дроби. Другими словами в дробной части такого числа бесконечное количество цифр и в их записи отсутствует период. Это говорит о том, что любая конечная случайная числовая последовательность рано или поздно обязательно будет являться частью иррационального числа. Данные обстоятельства могут сделать иррациональные числа ценным источником псевдослучайных последовательностей. Для генерации гаммы можно будет указать номер позиции в дробной части, с которой следует начать выбирать последовательность цифр для гаммы. Очевидно, что данный номер должен быть очень большим, чтобы потенциальный противник не смог за приемлемое время его определить методом перебора.

К **проблемам** использования иррациональных чисел в качестве псевдослучайных числовых последовательностей относятся:

- отсутствие доказательства нормальности чисел;

- относительно сложный алгоритм расчета, подразумевающий, как правило, использование рекуррентных процедур;

- наличие в своем составе «плохих» последовательностей (например, последовательностей требуемой для гаммы длины, но состоящих из одних нулей или имеющих легко определяемый период).

В качестве примера рассмотрим число **π**.

Равновероятность появления цифр в записи (≈ нормальность) числа π не доказана, хотя анализ первых 200 млрд. десятичных цифр говорит в пользу этого.

Таблица 6.8. Количество появлений десятичных цифр в первых 200 млрд. знаках числа π

Цифра	Количество появлений
0	20 000 030 841
1	19 999 914 711
2	20 000 136 978
3	20 000 069 393
4	19 999 921 691
5	19 999 917 053
6	19 999 881 515
7	19 999 967 594
8	20 000 291 044
9	19 999 869 180

Для расчета цифр числа π можно использовать приближенные формулы, ряды, пределы, интегралы и т.п. В первом случае получается неточное значение и, как правило, с периодом («теряется» иррациональность), в других требуется сложная процедура расчета с получением всех предшествующих цифр, начиная с первой.

Важным достижением в области расчетов числа π стала формула Бэйли – Боруэйна – Плаффа, открытая в 1997 г. Саймоном Плаффом и названная в честь авторов статьи, в которой она впервые была опубликована [67].

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k}\left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right). \qquad (6.13)$$

Эта формула примечательна тем, что она позволяет извлечь любую конкретную шестнадцатеричную или двоичную цифру числа π без вычисления предыдущих. К сожалению, расчеты по этой формуле достаточно медлительны, но это уже значительный шаг вперед для использования иррациональных чисел в практической криптографии.

---

[4] **Взаимно простые числа** – числа, не имеющие общих делителей, кроме 1, т.е. наибольший общий делитель которых равен 1.

6.3. Отличие ключа от гаммы

Очень часто понятия «ключ» и «гамма» отождествляют, но между ними есть принципиальные отличия. Напомним, **ключ** – минимально необходимая информация (за исключением сообщения, алфавитов и алгоритма), необходимая для шифрования и дешифрования сообщений. **Гаммой** является вся числовая последовательность, используемая для шифрования или дешифрования сообщения с длиной не менее этого сообщения.

Для иллюстрации отличий приведена следующая таблица.

Таблица 6.9. Отличия ключа от гаммы

Метод генерации гаммы	Ключ	Гамма
На базе случайных физических или антропогенных процессов	Ключ соответствует гамме	
На базе слова или фразы (см. табл. 6.2 и 6.4)	Слово или фраза	Циклически повторяющееся слово или фраза

Линейный или инверсный конгруэнтный генератор	Начальное число и коэффициенты формул	Генерируемая числовая последовательность
Регистр сдвига с линейной обратной связью	Вид полинома и исходное состояние регистра	Генерируемая числовая последовательность
RC4	Ключ K	Числовая последовательность из частей S-блока
На базе числа π	Номер начальной позиции в дробной части, с которой выбираются цифры числа π	Последовательность цифр числа π, начиная с заданной позиции

6.4. Стандарты и спецификации США

Развитая система стандартов и спецификаций по генерации и тестированию псевдослучайных последовательностей существует в США. ANSI (Американский национальный институт стандартов – англ. American national standards institute) и NIST (Американский национальный институт стандартизации – англ. National Institute of Standards and Technology) разработали следующие документы:

- стандарты ANSI серии X9.82:
  - ANSI X9.82 «Random Number Generation. Part 1: Overview and Basic Principles» (рус. «Генерация случайных чисел. Часть 1: Обзор и основные принципы»);
  - ANSI X9.82 «Financial Services - Random Number Generation. Part 2: Entropy Sources» (рус. «Финансовые услуги – Генерация случайных чисел. Часть 2: Источники энтропии»);
  - ANSI X9.82 «Random Number Generation. Part 3: Deterministic Random Bit Generators» (рус. «Генерация случайных чисел. Часть 3: Детерминированные генераторы случайных битов»);
  - ANSI X9.82 «Random Number Generation. Part 4: Random Bit Generator Constructions» (рус. «Генерация случайных чисел. Часть 4: Конструкции генератора случайных битов»);
- специальные публикации NIST серии 800:
  - NIST SP 800-22 Rev.1a «A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications» (рус. «Набор статистических тестов для генераторов случайных и псевдослучайных чисел для криптографического приложений»);
  - NIST SP 800-90A «Recommendation for Random Number Generation Using Deterministic Random Bit Generators» (рус. «Рекомендации для генерации случайных чисел с использованием детерминированных генераторов случайных битов»).

Вопросы для самопроверки

1. В чем заключается основная идея криптографических преобразований аддитивных шифров?
2. Назовите основные характеристики гаммы.
3. При каких условиях применения гаммы аддитивный шифр можно считать совершенным.
4. Опишите схемы шифрования с использованием синхронных и самосинхронизирующихся потоковых шифров.

7. КВАНТОВОЕ ШИФРОВАНИЕ

7.1. Основы квантовой физики

Квант (от лат. quantum - «сколько») - неделимая порция какой-либо величины в физике. В основе понятия лежит представление о том, что некоторые физические величины могут принимать только определенные (дискретные) значения, величина которых кратна некоторому минимальному значению (постоянной Планка). **Постоянная Планка** (квант действия) - коэффициент, связывающий величину энергии электромагнитного излучения с его частотой (или некоторую другую пару физических величин)

$$h = \frac{E}{v} = 6.62606957(29) * 10^{-34}, \text{ Дж*с} \qquad (7.1)$$

где E - энергия электромагнитного излучения, Дж;
v - частота электромагнитного излучения, 1/с.

На текущий момент большинство ученых считает, что на макроскопическом уровне (звезд, людей, молекул) действуют законы классической физики, а на микроскопическом (элементарных частиц – кварков, фотонов, электронов) – иные, в т.ч. и квантовой. В рамках квантовой физики любая элементарная частица рассматривается как квант возбуждения поля, характерной для этой частицы. Квантовые поля могут взаимодействовать между собой, в результате чего кванты (частицы) могут превращаться друг в друга.

Одной из элементарных частиц является фотон. **Фотон** (от др.-греч. φῶς - «свет») - квант электромагнитного излучения (электромагнитная волна) с нулевыми массой покоя и зарядом. Согласно современным представлениям фотону свойственен **корпускулярно-волновой дуализм**. С одной стороны, фотон демонстрирует свойства волны в явлениях дифракции[1] и интерференции[2] в том случае, если размеры препятствий сравнимы с длиной волны фотона. С другой стороны, процессы взаимодействия фотонов с веществом (излучение и поглощение) можно успешно истолковать только на основе представлений о нем, как о дискретной частице.

Излучение светящегося тела можно представить как потоки фотонов, направленных от него во все стороны.

Рис.7.1. Излучение светящегося тела

Эти потоки распространяются, как правило, вдоль прямолинейного луча, за исключением прохода малых препятствий (см. дифракцию) или вблизи массивных тел (планет, звезд). Т.о. поток фотонов можно представить как набор электромагнитных волн, каждая из которых колеблется (поляризована) в одной

плоскости.

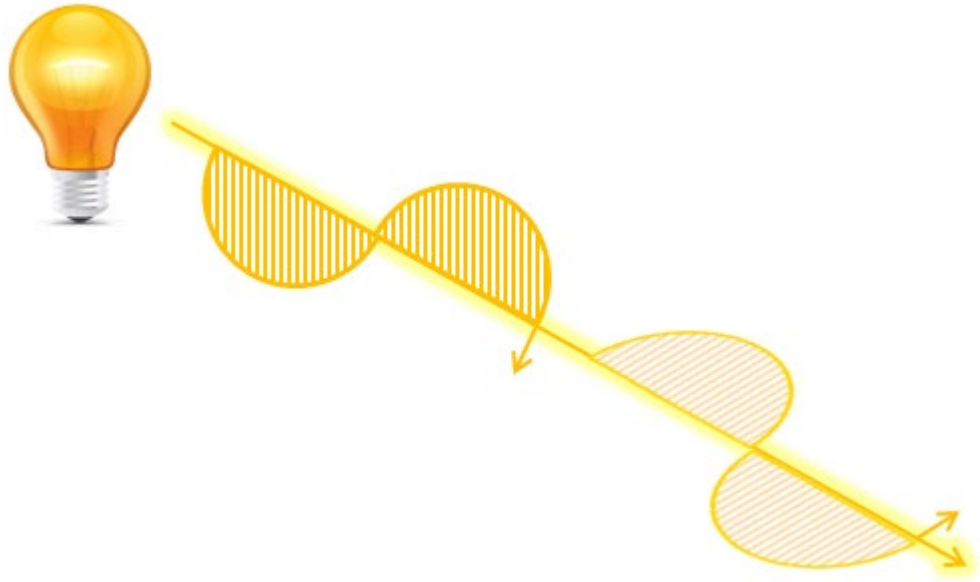


Рис.7.2. Поток из двух фотонов (электромагнитных волн)

При пропускании оптического излучения через **линейный поляризатор** (например, призму Глана) на выходе можно получить поток фотонов с требуемой поляризацией. Образно этот процесс можно представить, как прохождение фотонов через фильтр с плоским отверстием, который проходит только часть фотонов – те, плоскость поляризации которых совпадает с ориентацией отверстия фильтра.

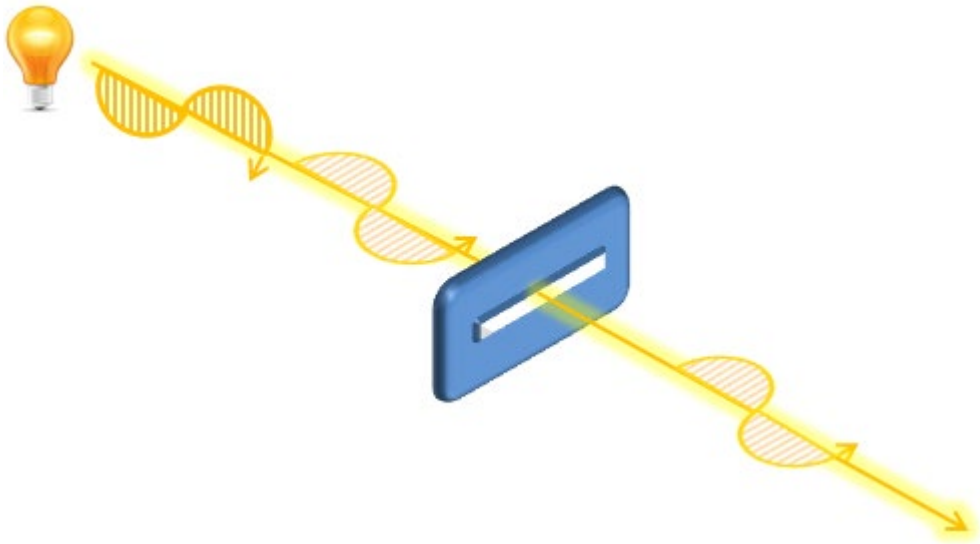


Рис.7.3. Прохождение потока фотонов сквозь фильтр

Если сгенерировать поток фотонов с требуемой поляризацией можно, то точно определить поляризацию произвольно перехваченных фотонов современные измерительные устройства не позволяют. Это обстоятельство и позволило использовать квантовую механику для шифрования информации.

Для кодирования информации, представленной в битовом виде, достаточно двух линейных поляризаторов с разными плоскостями поляризации. Фотоны, ориентированные в одной плоскости будут соответствовать «1», в другой – «0». На этом принципе основана работа передающего устройства (генератора фотонов).

Принимающее устройство может быть построено на базе линейного поляризатора или **поляризационной разделительной призмы**.

В первом случае, если принятый фотон проходит сквозь поляризатор и регистрируется стоящим за ним фотодетектором, то он идентифицируется как «1», в противном случае – как «0».

Во втором случае, при прохождении фотона через призму он перенаправляется на один из двух фотодетекторов, соответствующих двум взаимно перпендикулярным плоскостям поляризации (**базису**). Перенаправление выполняется на тот фотодетектор, чья плоскость поляризации наиболее близка к плоскости поляризации принятого фотона. Если передающая и принимающая стороны используют одинаковый базис поляризации (например, горизонтально-вертикальный), то принимающая сторона будет однозначно идентифицировать поступающие фотоны (например, с горизонтальной поляризацией как «1», с вертикальной - как «0»). Если принимающая сторона будет использовать другой базис поляризации, то вероятность правильной идентификации снижается. В частности, если у принимающей стороны базис будет повернут на 45^{o} относительно передающей, то вероятность правильной идентификации будет примерно 50%.

---

[1]**Дифракция** (лат. diffractus - буквально разломанный, переломанный) – отклонение света (световых волн, фотонов) от прямолинейного распространения при прохождении сквозь малые отверстия или огибании малых препятствий.

[2]**Интерференция света** – перераспределение интенсивности света в результате наложения (суперпозиции) нескольких световых волн.

7.2. Основы квантового шифрования

7.2.1. Шифрования с использованием линейных поляризаторов на принимающей стороне

Для отправки и приема фотонов стороны могут использовать четыре фильтра (поляризатора).

а) горизонтальный | б) вертикальный | в) слева-направо снизу-вверх | г) слева-направо сверху-вниз

Рис.7.4. Фильтры для отправки и приема фотонов

Если передающая и принимающая стороны для одного фотона (бита информации) будут использовать одинаковые фильтры, то он будет регистрироваться фотодетектором на принимающей стороне и идентифицироваться как «1», в противном случае – как «0».

Для обмена секретной информацией сторонам достаточно использовать два фильтра. При этом они заранее оговаривают порядок их использования на принимающей стороне, но он должен быть случайным для потенциального противника. Этот порядок будет являться ключом шифрования/дешифрования. Отправитель, зная последовательность применения фильтров получателем (ключ), для отправки сообщения должен применить соответствующий фильтр, чтобы получатель правильно идентифицировал принятый фотон.

В следующей таблице приведен пример отправки сообщения «00110101_2» при использовании получателем горизонтального и вертикального фильтров.

Таблица 7.1. Пример отправки и приема сообщения

Посылаемое сообщение, бит	0	0	1	1	0	1	0	1
Фильтры, применяемые получателем (ключ)								
Фильтры, выбираемые отправителем								
Принятое сообщение, бит	0	0	1	1	0	1	0	1

Т.к. противник не знает порядок использования фильтров, то он не может правильно интерпретировать перехваченные фотоны.

В целях обеспечения стойкости данной схемы шифрования ключ, как и в шифрах гаммирования, должен быть истинно случайным и одноразовым. Для генерации таких ключей можно использовать генераторы гамм. Например, если в гамме очередной бит равен «1», то получатель должен использовать горизонтальный фильтр, иначе – вертикальный. Для генерации ключа (гаммы) можно использовать квантовые протоколы обмена ключами (например, протокол BB84).

Для повышения стойкости шифра рекомендуется менять пары фильтров на стороне получателя. Это позволит также определить факт прослушивания канала связи. При перехвате фотона противник вынужден создать его копию (клон) и послать ее законному получателю. Если противник будет использовать не ту пару фильтров или разделительную призму, то поляризацию части фотонов он не сможет правильно определить, а значит и создать требуемые копии. Для проверки перехвата сообщения отправитель и получатель могут вычислить хеш-образы (контрольные суммы), соответственно, посланного и принятого сообщений. Сравнив хеш-образы, которые можно передавать по отрытому каналу не защищая, они могут сделать вывод о прослушивания канала связи или о корректности передачи (целостности) данных.

7.2.2. Шифрования с использованием поляризационной разделительной призмы на принимающей стороне

Для передачи секретной информации передающая и принимающая стороны используют два базиса поляризации: прямолинейный и диагональный. При этом порядок чередования базисов для отдельных фотонов (битов информации), как и в предыдущей схеме, должен быть известен сторонам и случайным для потенциального противника. Этот порядок представляет собой ключ шифрования/дешифрования, для формирования которого можно использовать генераторы гамм или протокол BB84. При этом, если в гамме очередной бит равен «1», то стороны должны использовать прямолинейный базис, иначе – диагональный.

Для отправки фотонов передающая сторона применяет четыре фильтра

Таблица 7.2. Фильтры для отправки фотонов

Базис	Тип фильтра	Значение бита
прямолинейный 	горизонтальный 	0
	вертикальный	1

		
диагональный 	слева-направо снизу-вверх 	0
	слева-направо сверху-вниз 	1

Тип фильтра для передачи отдельного фотона выбирается в зависимости от определенного сторонами порядка использования базисов. Аналогичным образом поступает и принимающая сторона – применяет базис поляризации в зависимости от ключа.

Таблица 7.3. Пример отправки и приема сообщения

Посылаемое сообщение, бит	0	0	1	1	0	1	0	1
Базисы, применяемые получателем (ключ)								
Фильтры, выбираемые отправителем								
Принятое сообщение, бит	0	0	1	1	0	1	0	1

Противник, перехватив фотон, не может со стопроцентной уверенностью определить его поляризацию: действительно он угадал применяемый базис (≈ идентифицировал отправленный бит) или это был базис, расположенный под углом 45^{o} к применяемому? В последнем случае вероятность правильной идентификации бита равна 50%.

Это обстоятельство позволяет также определить факт прослушивания канала связи. При перехвате фотона противник вынужден создать его копию (клон) и послать ее законному получателю. Определив значение бита для перехваченного фотона, у него будет два равновероятных варианта поляризации для копии, соответствующие двум разных базисам. Для проверки перехвата сообщения передающая и принимающая стороны могут вычислить хеш-образы (контрольные суммы), соответственно, посланного и принятого сообщений и сравнить их.

Вопросы для самопроверки

1. Дайте определение понятиям «квант» и «фотон».

2. В чем заключается природа корпускулярно-волнового дуализма фотона?

3. В чем суть шифрования с помощью фотонов?

8. КОМБИНИРОВАННЫЕ ШИФРЫ

8.1. Шифры ADFGX и ADFGVX

Комбинированные (составные) шифры предполагают использование для шифрования сообщения сразу нескольких методов (например, сначала замена символов, а затем их перестановка).

Два самых известных полевых шифра в истории криптографии - ADFGX и ADFGVX. Шифр ADFGX впервые был использован во время решающих этапов Первой мировой войны, когда в марте 1918 г. кайзеровские генералы начали крупное наступление. Шифр ADFG(V)X был разработан полковником Фрицем Небелем, офицером связи, служившим в штабе германской армии [43]. Оба шифра предполагают вначале применение к буквам исходного сообщения замену, после чего для получения окончательной шифрограммы выполняется перестановка.

Таблица шифрозамен ADFGX представляет собой матрицу 5 x 5, а для ADFGVX – 6 x 6. Строки и столбцы обозначаются буквами, входящими в название шифра. Пример таблицы шифрозамен для шифра ADFGVX применительно к русскому алфавиту показан на следующем рисунке.

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

Рис.8.1. Пример таблицы шифрозамен для шифра ADFGVX

Шифрозамена для буквы исходного текста состоит из букв, обозначающих строку и столбец, на пересечении которых она находится (см. шифр «Полибианский квадрат»). Например, для сообщения «АБРАМОВ» набор шифрозамен будет «FD AX GA FD FG FV DA».

На втором этапе для выполнения перестановки полученный набор шифрозамен вписывается построчно сверху-вниз в таблицу, количество столбцов в которой строго определено (ADFGX) или соответствует количеству букв в ключевом слове (ADFGVX). Нумерация столбов либо оговаривается сторонами (ADFGX) либо соответствует положению букв ключевого слова в алфавите, как в шифре вертикальной перестановки (ADFGVX). Например, для полученного выше набора шифрозамен перестановочная таблица с ключевым словом «ДЯДИНА» показана на следующем рисунке.

Д	Я	Д	И	Н	А
2	6	3	4	5	1
F	D	A	X	G	A
F	D	F	G	F	V
D	A				

Рис.8.2. Перестановочная таблица шифра ADFGVX с ключевым словом «ДЯДИНА»

На третьем этапе буквы выписываются из столбцов в соответствии с их нумерацией, при этом считывание происходит по столбцам,

а буквы объединяются в пятибуквенные группы. Таким образом, окончательная шифрограмма для рассматриваемого примера будет выглядеть «AVFFD AFXGG FDDA».

8.2. Основы блочного комбинированного шифрования

Среди комбинированных методов шифрования наиболее распространенными являются методы блочного шифрования. **Блочное шифрование** предполагает разбиение исходного открытого текста на равные блоки, к которым применяется однотипная процедура шифрования. В настоящее время блочные шифры широко используются на практике. Российский и бывший американский стандарты шифрования (DES) относятся именно к этому классу шифров. В основе шифров лежат так называемые «**сети Фейстеля**» [17]. В 1971 г. Хорст Фейстель (Horst Feistel) запатентовал два устройства с различными алгоритмами шифрования, названные затем общим именем «Люцифер» (Lucifer). Одно из устройств использовало конструкцию, впоследствии названную «сетью Фейстеля» («Feistel cipher», «Feistel network»). Работа над созданием новых криптосистем велась им в стенах IBM вместе с Доном Копперсмитом (Don Coppersmith). Проект «Люцифер» был скорее экспериментальным, но стал базисом для алгоритма Data Encryption Standard (DES). В 1977 г. DES стал стандартом в США на шифрование данных вплоть до 2001 г. и до последнего времени широко использовался в криптографических системах.

Сеть Фейстеля состоит из нескольких **ячеек**.

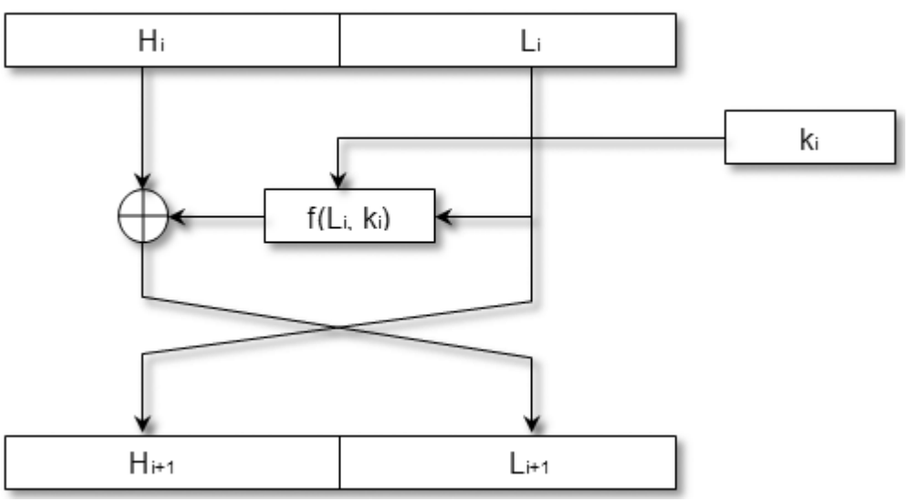


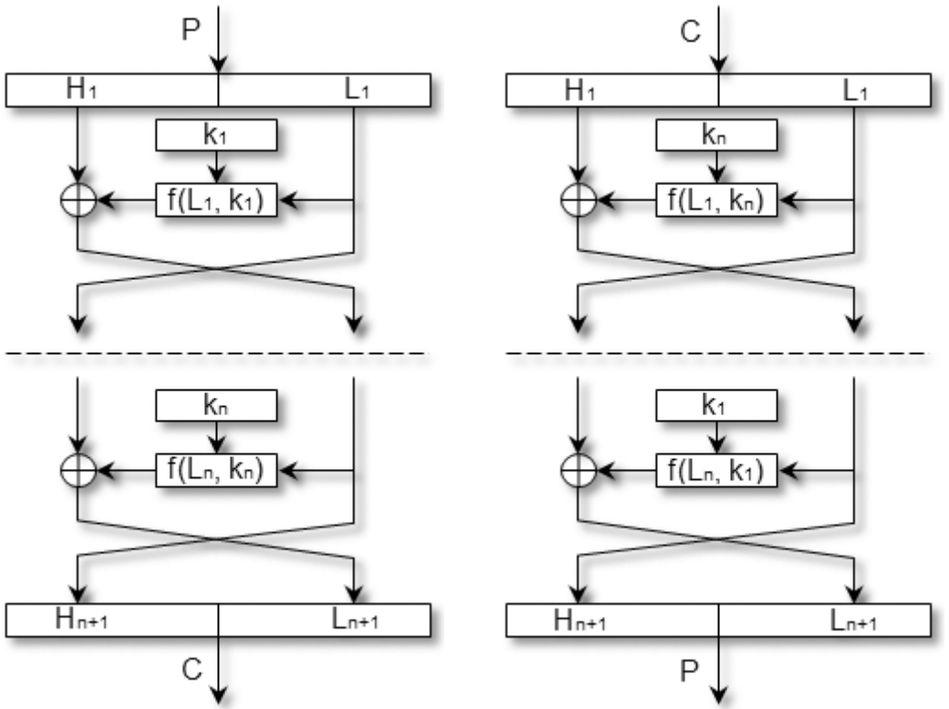
Рис.8.3. Ячейка Фейстеля

На определенном раунде шифрования шифруемый блок разбивается на две равные половинки - $\mathbf{H_i}$ (высший, high) и $\mathbf{L_i}$ (низший, low). К правой половинке применяется функция шифрования **f** с использованием ключевого элемента $\mathbf{k_i}$ (части ключа или модификации части ключа). После этого выполняется сложение по модулю два левой половинки и результата модификации правой половинки. В результате шифруемым блоком для следующего раунда будет объединение половинок, полученных по формулам

$$H_{i+1} = L_i, \qquad (8.1)$$

$$L_{i+1} = H_i \oplus f(L_i, k_i). \qquad (8.2)$$

В общем виде, центральная часть блочных шифров, основанных на сетях Фейстеля, выглядит следующим образом.



Зашифрование Расшифрование

Рис.8.4. Сеть Фейстеля

Как видно, процедуры зашифрования и расшифрования полностью идентичны, за исключением порядка использования ключевых элементов $\mathbf{k_i}$. При этом функция шифрования **f** может быть сколь угодно сложной и не обязательно обратимой. Обратимость гарантирует сама конструкция сети.

Шифрование при помощи данной конструкции легко реализуется как на программном уровне, так и на аппаратном, что обеспечивает широкие возможности применения. На основе сетей Фейстеля разработано большое количество шифров, среди которых: DES, ГОСТ 28147-89, Blowfish, CAST, FEAL, IDEA, Khufu, Twofish и многие другие.

8.3. DES

DES (Data Encryption Standard, стандарт шифрования данных) - федеральный стандарт шифрования США в 1977-2001 гг. **для использования во всех несекретных правительственных каналах связи** (FIPS PUB 46 «Data Encryption Standard») [8, 16]. Несмотря на то, что в настоящий момент федеральным стандартом шифрования США является Rijndael (AES), рассмотрение DES позволяет понять основные принципы блочного шифрования.

В алгоритме, лежащем в основе DES, используются методы замены, перестановки и гаммирования (сложение по модулю 2).

Открытое сообщение разбивается на блоки длиной 64 бита. Если длина сообщения не кратна 64, оно дополняется справа недостающим количеством битов.

Данные шифруются ключом длиной 56 битов. На самом деле ключ имеет размер 64 бита, однако реально для выработки ключевых элементов используются только 56 из них. Самые младшие биты каждого байта ключа (8-ой, 16-ый, …, 64-ый) не попадают в ключевые элементы и служат исключительно для контроля четности. Требуется, чтобы сумма битов каждого байта ключа, включая контрольный, была четной (четный паритетный бит).

Для решения разнообразных криптографических задач, разработаны **четыре рабочих режима**, реализующих DES:

- электронная кодовая книга ECB (Electronic Code Book);

- сцепление блоков шифра CBC (Cipher Block Chaining);

- обратная связь по шифртексту CFB (Cipher Feed Back);

- обратная связь по выходу OFB (Output Feed Back).

Режим ECB (электронная кодовая книга - Electronic Code Book).

Открытое сообщение разбивают на 64-битовые блоки. Каждый из них шифруют независимо с использованием одного и того же ключа шифрования.

Общая схема шифрования блока изображена на рис.8.5 [24].

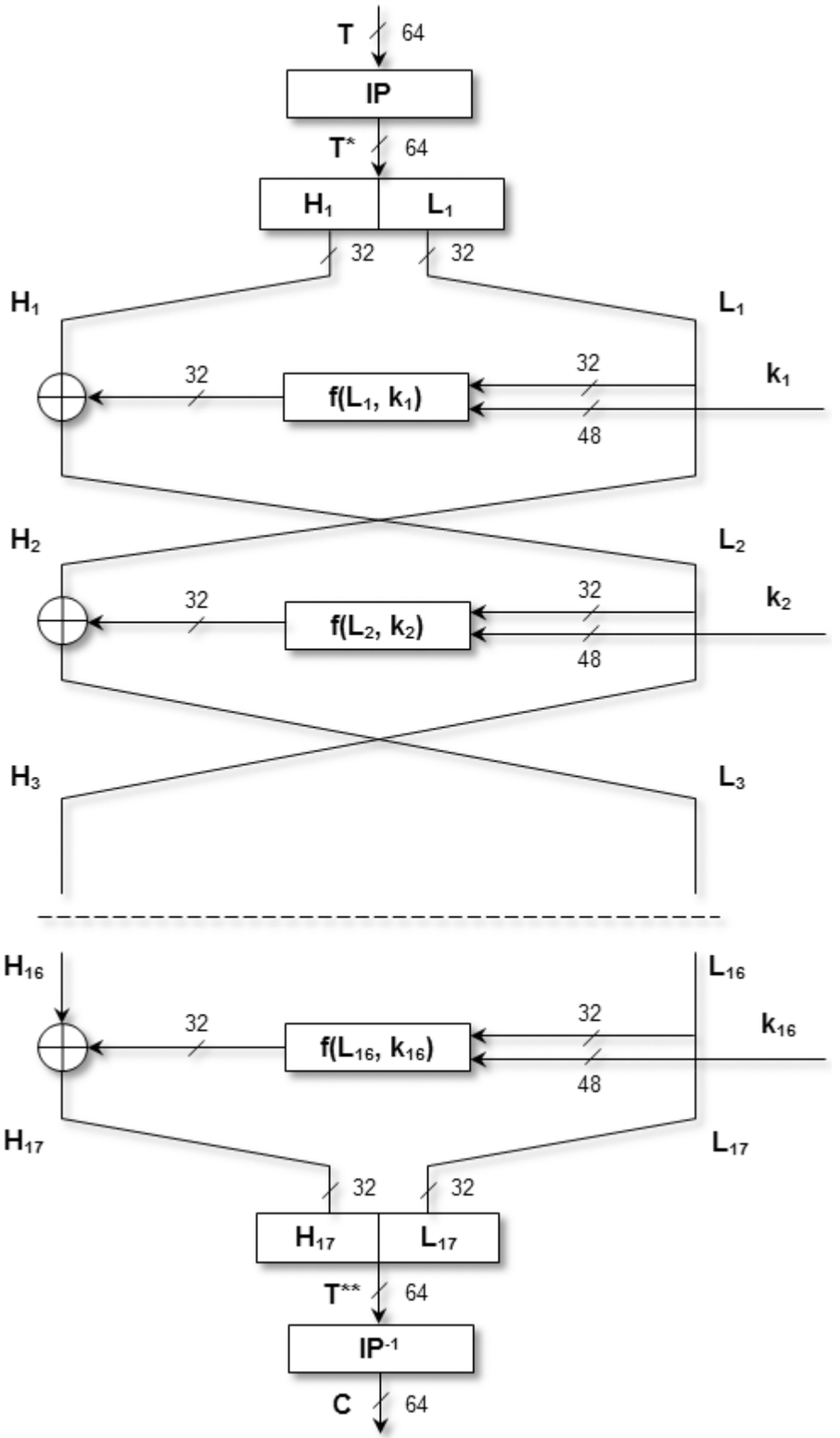


Рис.8.5. Схема шифрования блока

1. Шифрование 64-битового блока данных **T** начинается с начальной перестановки битов **IP**.

Таблица 8.1. Начальная перестановка IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4

62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

В таблице указывается новое положение соответствующего бита. Таким образом, при выполнении начальной перестановки 58-ый бит станет 1-ым, 50-ый – 2-ым, 42-ой – 3-им и т.д.

2. Результат перестановки **T*** разделяется на две 32-битовые части $\mathbf{H_1}$ и $\mathbf{L_1}$, с которыми выполняются 16 раундов преобразования.

3. В каждом раунде **i** старшая половина $\mathbf{H_i}$ блока модифицируется путем побитового прибавления к ней по модулю 2 ($\oplus$) результата вычисления функции шифрования **f**, зависящей от младшей половины блока $\mathbf{L_i}$ и 48-битового ключевого элемента $\mathbf{k_i}$. Ключевой элемент $\mathbf{k_i}$ вырабатывается из ключа шифрования. Между раундами старшая и младшая половины блока меняются местами. В последнем раунде происходит то же самое, за исключением обмена значениями половинок блока.

4. Полублоки $\mathbf{H_{17}}$ и $\mathbf{L_{17}}$ объединяются в полный блок **T****, в котором выполняется конечная битовая перестановка $\mathbf{IP^{-1}}$, обратная начальной.

Таблица 8.2. Конечная перестановка IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Результат последней операции и является выходным значением цикла шифрования – зашифрованным блоком **C**.

Все перестановки в таблицах **IP** и $\mathbf{IP^{-1}}$ подобраны разработчиками таким образом, чтобы максимально затруднить процесс расшифровки путём подбора ключа.

Схема функции шифрования **f** приведена на рис.8.6.

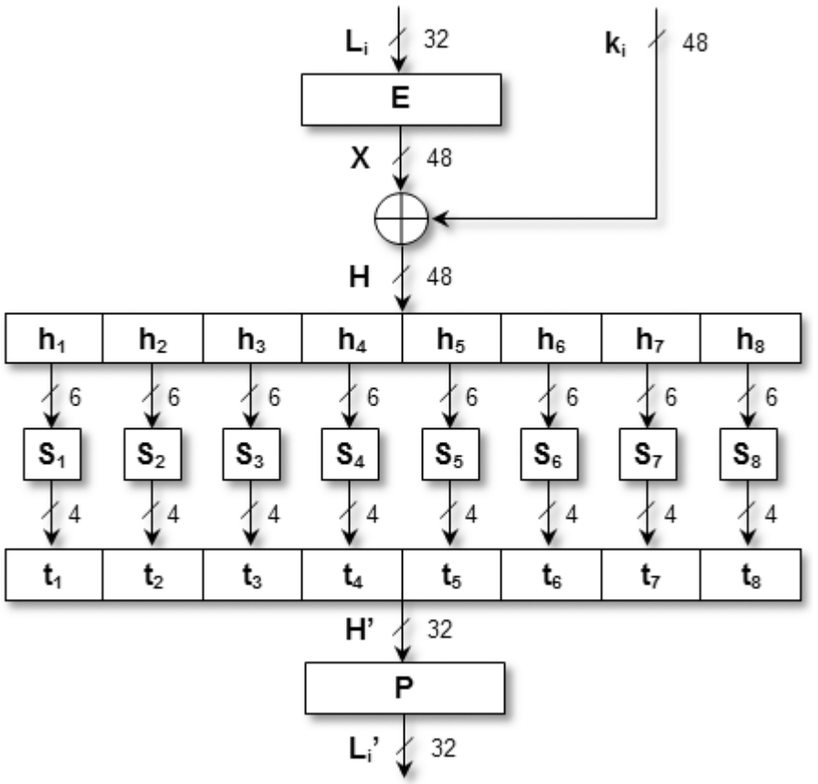


Рис.8.6. Схема функции шифрования

1. На вход поступает 32-битовая половина шифруемого блока $\mathbf{L_i}$ и 48-битовый ключевой элемент $\mathbf{k_i}$.

2. $\mathbf{L_i}$ разбивается на 8 тетрад по 4 бита. Каждая тетрада по циклическому закону дополняется крайними битами из соседних тетрад до 6-битного слова (функция расширения **E**). Цикличность означает, что первый бит $\mathbf{L_i}$ добавляется последним в последнее слово, а последний бит $\mathbf{L_i}$ добавляется первым в первое слово. Далее выполняется объединение тетрад в 48-битный блок **X**. Например, $\mathbf{L_i}$ = 0111 0110 1… …0 1101_2, тогда **X** = 101110 101101 … 011010_2.

3. **X** побитово суммируется по модулю 2 (⊕) с ключевым элементом $\mathbf{k_i}$.

4. 48-битовый блок данных **H** разделяется на восемь 6-битовых элементов, обозначенных $\mathbf{h_1, h_2, …, h_8}$.

5. Каждое из значений $\mathbf{h_j}$ преобразуется в новое 4-битовое значение $\mathbf{t_j}$ с помощью соответствующего узла замены $\mathbf{S_j}$.

Таблица 8.3. Значения t_j

Узел замены	Номер строки	Номер столбца															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15

	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Если на вход $\mathbf{S_j}$ поступает блок $\mathbf{h_j=b_1b_2b_3b_4b_5b_6}$, то двухбитовое число $\mathbf{b_1b_6}$ указывает номер строки матрицы, а четырёхбитовое число $\mathbf{b_2b_3b_4b_5}$ - номер столбца в таблице узлов замен. В результате применения узла замены $\mathbf{S_j}$ к блоку $\mathbf{h_j}$ получается число (от 0 до 15), которое преобразуется в $\mathbf{t_j}$. Например, в узел замены S_3 поступает $h_3=101011_2$. Тогда, номер строки равен 3_{10} ($b_1b_6=11_2$), номер столбца – 5_{10} ($b_2b_3b_4b_5=0101_2$), $t_3=9_{10}$ (1001_2).

6. Полученные восемь элементов $\mathbf{t_j}$ вновь объединяются в 32-битовый блок **H'**.

7. В **H'** выполняется перестановка битов **P**.

Таблица 8.4. Перестановка P

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Результат последней операции и является выходным значением функции шифрования $\mathbf{L_i'}$.

Ключевые элементы вырабатываются из ключа с использованием сдвигов и битовых выборок-перестановок. Таким образом, ключевые элементы состоят исключительно из битов исходного ключа, «перетасованных» в различном порядке. Схема выработки ключевых элементов показана на следующем рисунке.

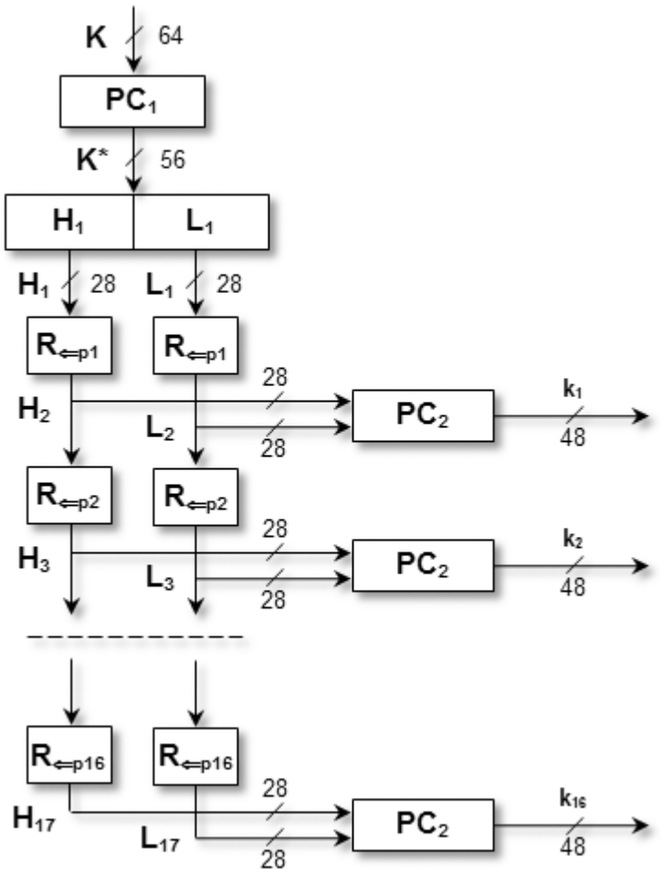


Рис.8.7. Схема выработки ключевых элементов

1. Выработка ключевых элементов из ключа **K** начинается со входной выборки-перестановки битов $\mathbf{PC_1}$, которая отбирает 56 из 64 битов ключа и располагает их в другом порядке.

Таблица 8.5. Перестановка PC_1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

2. Результат выборки-перестановки **K*** разделяется на две 28-битовые части: старшую $\mathbf{H_1}$ и младшую $\mathbf{L_1}$.

3. 16 раз выполняется процедура.

3а. В зависимости от номера итерации обе части циклически сдвигаются на 1 или 2 бита влево.

Таблица 8.6. Циклический сдвиг

Номер итерации	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сдвиг (бит)	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

3б. После сдвига части объединяются и из них с помощью выборки-перестановки $\mathbf{PC_2}$ отбираются 48 битов, которые и формируют очередной ключевой элемент.

Таблица 8.7. Перестановка PC_2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Алгоритмы шифрования и расшифрования DES-ECB в общем виде выражаются следующими схемами

$$C = DES(T) = IP(T) \rightarrow H_1 \oplus f(L_1, k_1), L_1 \rightarrow \ldots \rightarrow H_{16} \oplus f(L_{16}, k_{16}), L_{16} \rightarrow IP^{-1}(2^{32}*L_{17}+H_{17}), \qquad (8.3)$$

$$T = DES^{-1}(C) = IP(C) \rightarrow H_1 \oplus f(L_1, k_{16}), L_1 \rightarrow \ldots \rightarrow H_{16} \oplus f(L_{16}, k_1), L_{16} \rightarrow IP^{-1}(2^{32}*L_{17}+ H_{17}). \qquad (8.4)$$

Как видно схема алгоритма остается неизменной, меняется лишь порядок использования ключевых элементов. Таким образом, для расшифрования необходимо «прогнать» DES с тем же ключом, но использовать ключевые элементы в обратном порядке.

Использование различных методов шифрования:

- замена - функция расширения **E**, узлы замены **S**;

- перестановка – перестановки **IP**, $\mathbf{IP^{-1}}$, **P**, $\mathbf{PC_1}$, $\mathbf{PC_2}$, чередование $\mathbf{L_i}$ и $\mathbf{H_i}$, циклический сдвиг;

- гаммирование – $\oplus$.

Из-за небольшого числа возможных ключей (всего 256), появляется возможность их полного перебора на быстродействующей вычислительной технике за реальное время. В 1998 г. Electronic Frontier Foundation используя специальный компьютер DES-Cracker, удалось взломать DES за 3 дня. По неподтвержденным данным, Агентство национальной безопасности США уже в 1996 г. могло вскрывать ключ DES за 3-15 мин. с помощью устройства стоимостью 50000 долларов.

Режим CBC (сцепление блоков шифра - Cipher Block Chaining).

Схемы алгоритмов представлены на следующем рисунке. По каждой из стрелок передается 64 бита информации.

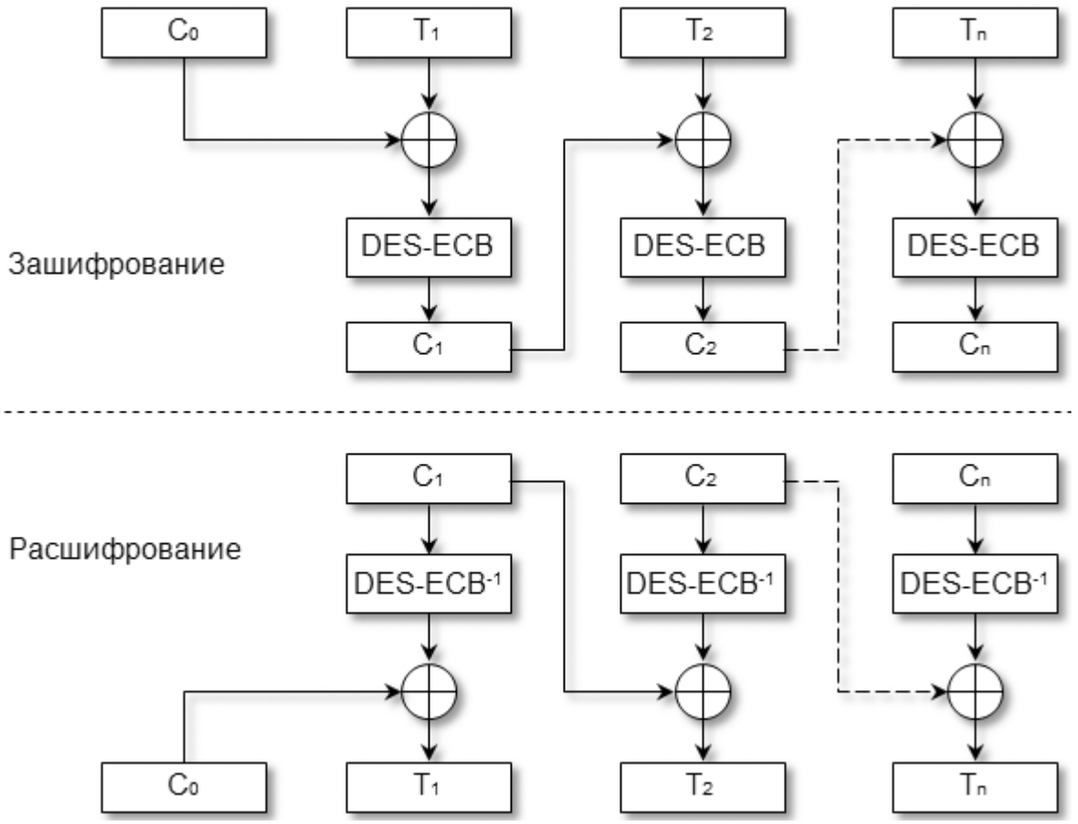


Рис.8.8. Схемы алгоритмов DES в режиме сцепления блоков шифра (CBC)

1. Исходное сообщение разбивается на 64-битовые блоки: $\mathbf{T_1, T_2, ..., T_n}$.

2. Первый блок $\mathbf{T_1}$ складывается по модулю 2 с 64-битовым начальным вектором $\mathbf{C_0}$, который меняется и держится в секрете ($\mathbf{C_0}$ - **синхропосылка**, по сути, второй ключ).

3. Полученная сумма затем шифруется с использованием ключа DES.

4. 64-битовый шифр $\mathbf{C_1}$ складывается по модулю 2 со вторым блоком текста, ре¬зультат шифруется и получается второй 64-битовый шифр $\mathbf{C_2}$, и т.д.

5. Процедура повторяется до тех пор, пока не будут обработаны все блоки текста. Т.о., для всех **i** = 1...n (**n** - число блоков) результат шифрования $\mathbf{C_i}$ определяется следующим образом: $\mathbf{C_i = DES(T_i \oplus C_{i-1})}$.

Очевидно, что последний 64-битовый блок шифртекста является функцией секретного ключа, начального вектора и каждого бита открытого текста независимо от его длины. Этот блок шифртекста называют **кодом аутентификации сообщения** (**КАС**). Код КАС может быть легко проверен получателем, владеющим секретным ключом и начальным вектором, путем повторения процедуры, выполненной отправителем. Достоинство данного режима в том, что он не позволяет накапливаться ошибкам при передаче. Блок $\mathbf{T_i}$ является функцией только $\mathbf{C_{i-1}}$ и $\mathbf{C_i}$. Поэтому ошибка при передаче приведет к потере только двух блоков исходного текста.

Режимы CFB и OFB (обратная связь по шифртексту - Cipher Feed Back; обратная связь по выходу - Output Feed Back).

В алгоритмах CFB и OFB используется такой же подход, что и в алгоритме CBC, когда результат шифрования (расшифрования) блока на предыдущем шаге используется для шифровки (расшифровки) очередного блока. В этих режимах размер входного блока может быть меньше 64 бит.

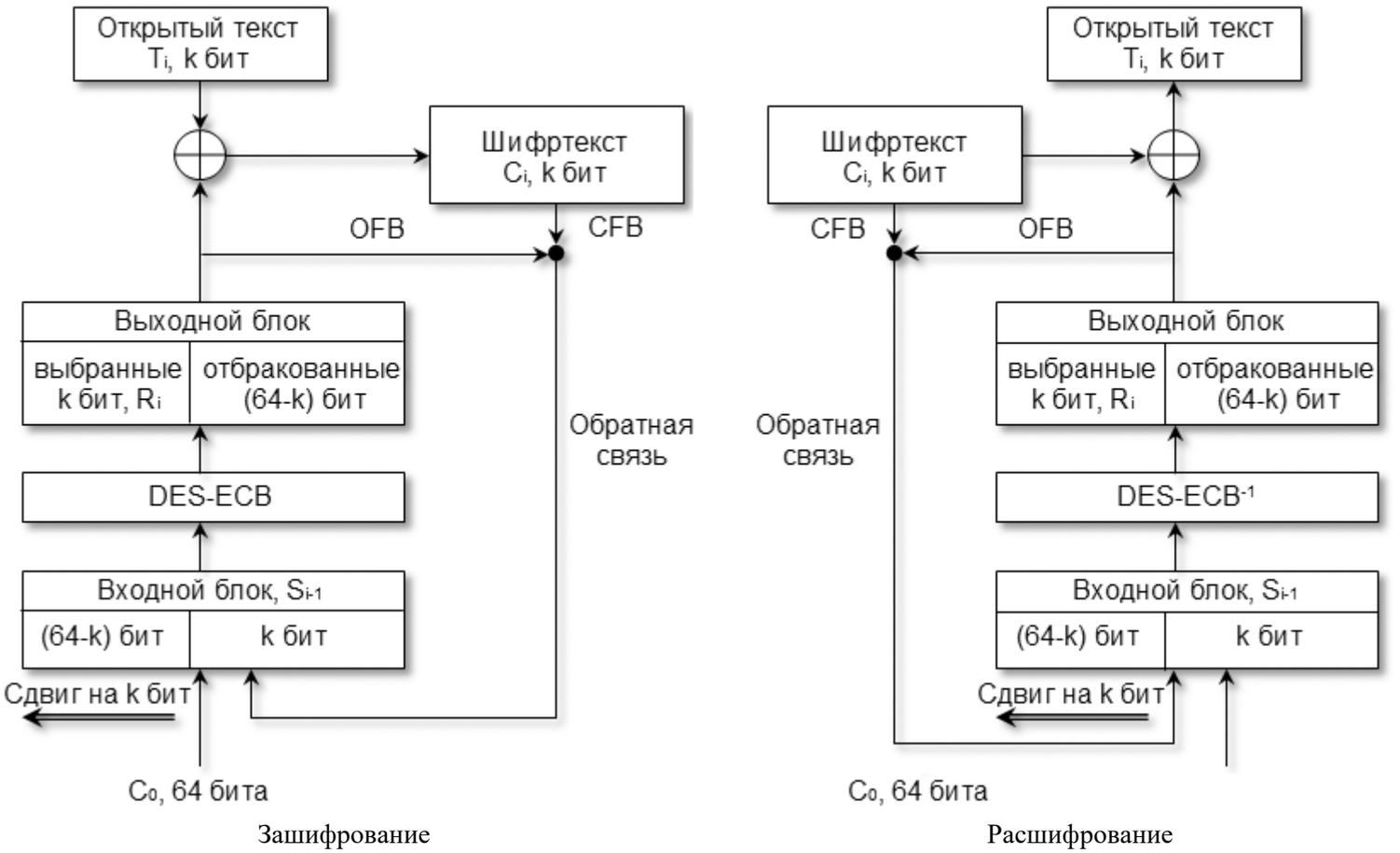


Рис.8.9. Схемы алгоритмов DES в режимах CFB и OFB

Алгоритм шифрования.

1. Исходной сообщение разбивается на блоки, длиной **k** бит: $\mathbf{T_1, T_2, ..., T_n}$.

2. Синхропосылка шифруется с использованием ключа DES.

3. Из полученного результата выбираются старшие **k** бит, которые складываются с блоком открытого текста $\mathbf{T_i}$ по модулю 2 для получения шифртекста $\mathbf{C_i}$.

4. Процедура повторяется до тех пор, пока не будут обработаны все блоки текста. При этом, содержимое входного блока сдвигается на **k** бит влево (причем сдвиг не циклический) и младшие **k** бит заполняются либо шифртекстом (CFB), либо старшими **k** бит выходного блока (OFB).

Алгоритм расшифрования аналогичен алгоритму шифрования и отличается только входом и выходом стрелок в блок сложения по модулю 2.

Тройной DES.

При тройном DES текст шифруется 3 раза DES. Таким образом, длина ключа возрастает до 168-бит (56x3). Однако, применение тройного DES не всегда означает увеличение уровня безопасности сообщения.

Типы тройного шифрования DES:

- DES-EEE3: Шифруется 3 раза с 3 различными ключами.

- DES-EDE3: 3 DES операции шифровка-расшифровка-шифровка с 3 различными ключами.

- DES-EEE2 и DES-EDE2: Как и предыдущие, за исключением того, что первая и третья операции используют одинаковый ключ.

Сферы применения разных режимов DES.

Электронная кодовая книга ECB рекомендуется использовать для шифрования сообщений длиной в один блок (как правило, ключей). Недостаток ECB, в сравнении с другими режимами шифрования - сохранение статистических особенностей открытого текста [17].

оригинальная битовая карта изображения

криптограмма в режиме ECB (сохранились статистические особенности)

другие режимы шифрования (псевдослучайная последовательность)

Рис.8.10. Сохранение статистических особенностей открытого текста

Сцепление блоков шифра CBC и тройной DES могут использоваться для шифрование больших объемов данных (в частности тройной DES используется для эмиссии и обработки кредитных карт VISA, EuroPay и других платежных систем).

Обратная связь по шифртексту CFB и обратная связь по выходу OFB рекомендуется для шифрования отдельных символов или небольших сообщений.

8.4. ГОСТ 28147-89

ГОСТ 28147-89 - бывший советский и российский стандарт симметричного шифрования, введённый в 1990 г.(«ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования»). С момента опубликования ГОСТа на нём стоял ограничительный гриф «Для служебного пользования», и формально шифр был объявлен «полностью открытым» только в мае 1994 г. [17].

В алгоритме, лежащем в основе ГОСТ, используются методы замены, перестановки и гаммирования (сложение по модулю 2 и 2^{32}).

Открытое сообщение разбивается на блоки длиной 64 бита. Если длина сообщения не кратна 64, оно дополняется справа недостающим количеством битов.

Длина ключа – 256 битов.

Рабочие режимы:

- простой замены;

- гаммирования;

- гаммирования с обратной связью;

- имитовставки.

Режим простой замены.

Открытое сообщение разбивают на 64-битовые блоки. Каждый из них шифруют независимо с использованием одного и того же ключа шифрования. Перед шифрованием 256-битный ключ разбивается на восемь 32-битных ключевых элементов (блоков) $\mathbf{k_1}$, $\mathbf{k_2}$, …, $\mathbf{k_8}$.

Общая схема шифрования блока изображена на следующем рисунке.

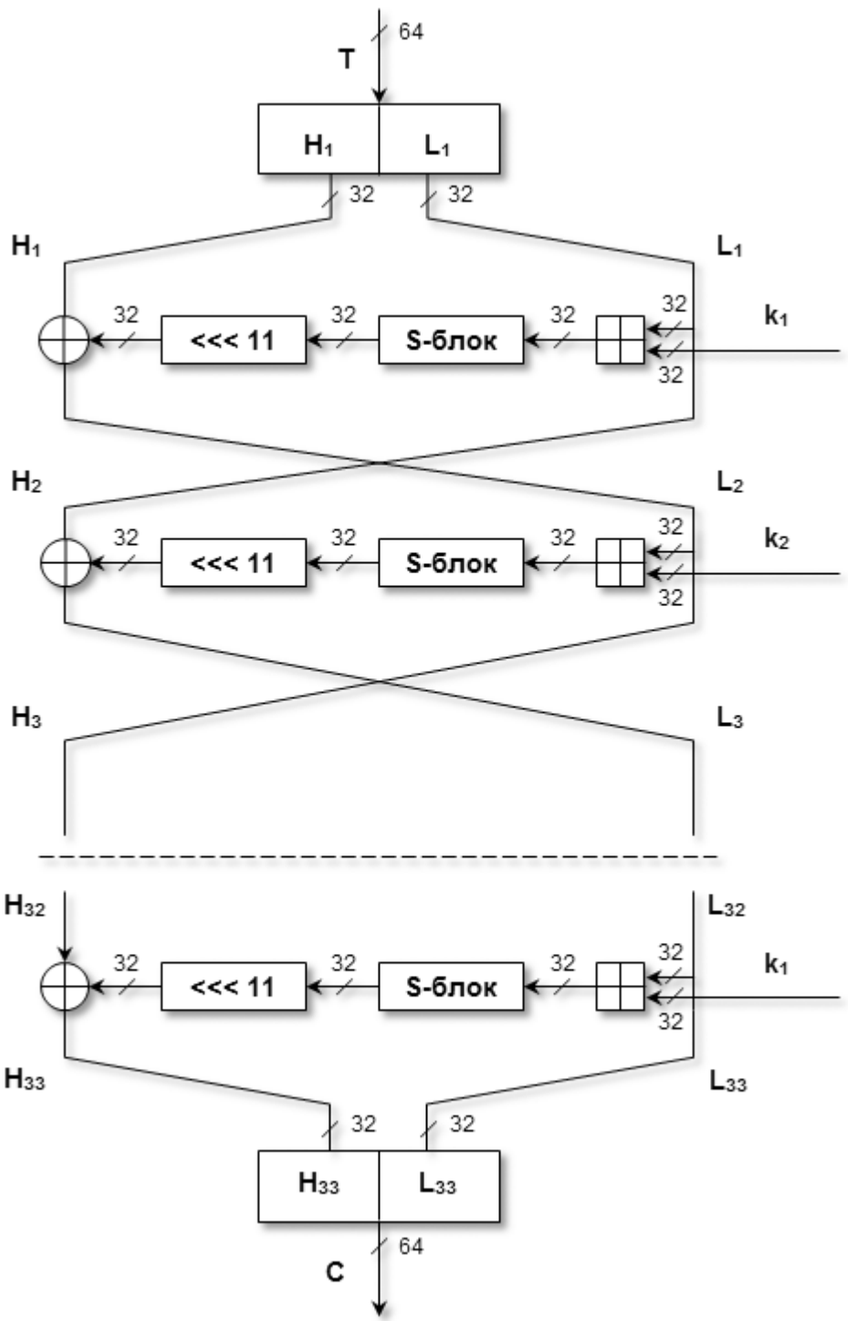


Рис.8.11. Схема шифрования блока

1. Шифрование 64-битового блока данных **T** начинается с его разбиения на две 32-битовые части $\mathbf{H_1}$ и $\mathbf{L_1}$, с которыми выполняются 32 раунда преобразования.

2. В каждом раунде **i**:

2.1. Младшая половина $\mathbf{L_i}$ складывается с ключевым элементом $\mathbf{k_i}$ по модулю 2^{32} ($\boxplus$).

$$X_i = (L_i + k_i) \bmod 2^{32}. \qquad (8.5)$$

Порядок использования ключевых элементов указан в следующей таблице.

Таблица 8.8. Порядок использования ключевых элементов

Раунд	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Ключевой элемент	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Раунд	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Ключевой элемент	1	2	3	4	5	6	7	8	8	7	6	5	4	3	2	1

2.2. Результат сложения $\mathbf{X_i}$ разбивается на восемь 4-битных фрагментов $\mathbf{x_{ij}}$ (j = 1..8), каждый из которых поступает на вход своего **S**-блока (узла замены) и заменяется на другой 4-битовый фрагмент $\mathbf{y_{ij}}$ (j = 1..8). В самом ГОСТ **S**-блоки не приведены. Указывается только, что они должны быть откуда-то взяты. В ГОСТ Р 34.11-94 «Информационная технология. Криптографическая защита информации. Функция хэширования» в приложении А приведены «проверочные» **S**-блоки, которые использовал ЦБ РФ в своих приложениях.

Таблица 8.9. Значения y_{ij}

S-блок (S_j)	Значение x_{ij}															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
S_2	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
S_3	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
S_4	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
S_5	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
S_6	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
S_7	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
S_8	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

Так, если третий фрагмент $x_{i3} = 6_{10}$ (0110_2), то он заменяется на $y_{i3} = 4_{10}$ (0100_2).

2.3. Замененные фрагменты $\mathbf{y_{ij}}$ объединяются обратно в 32-битовый блок $\mathbf{Y_i}$. После чего выполняется циклический сдвиг влево на 11 бит – $\mathbf{Z_i = Y_i <<< 11}$.

2.4. Старшая половина $\mathbf{H_i}$ блока модифицируется путем побитового прибавления к ней по модулю 2 ($\oplus$) результата сдвига $\mathbf{Z_i}$.

2.5. Между раундами старшая и младшая половины блока меняются местами.

3. Полублоки $\mathbf{H_{33}}$ и $\mathbf{L_{33}}$ объединяются в зашифрованный блок **C**.

Расшифрование отличается зашифрования обратным порядком использования ключевых элементов, представленного в табл.8.8.

Использование различных методов шифрования:

- замена - узлы замены (**S**-блоки);

- перестановка – чередование $\mathbf{H_i}$ и $\mathbf{L_i}$, циклический сдвиг на 11 бит;

- гаммирование – по модулю 2^{32} (⊞) и 2 (⊕).

Основные различия между DES и ГОСТ [8]:

- DES использует сложную процедуру для генерации подключей из ключей. В ГОСТ эта процедура очень проста;

- в DES 56-битовый ключ, а в ГОСТ - 256-битовый;

- у **S**-блоков DES 6-битовые входы и 4-битовые выходы, а у **S**-блоков ГОСТ 4-битовые входы и выходы. В обоих алгоритмах используется по восемь **S**-блоков;

- в DES используются перестановка, названная P-блоком, а в ГОСТ используется 11-битовый циклический сдвиг влево;

- в DES 16 этапов, а в ГОСТ - 32.

По мнению одного из авторитетных специалистов в области криптографии Брюса Шнайера увеличенная длина ключа и использование в вдвое больше раундов шифрования делает ГОСТ

> *... более устойчивым к дифференциальному и линейному криптоанализу, чем DES. ... Разработчики ГОСТ пытались достигнуть равновесия между безопасностью и эффективностью. Они изменили идеологию DES так, чтобы создать алгоритм, который больше подходит для программной реализации. Они, по видимому, менее уверены в безопасности своего алгоритма и попытались скомпенсировать это очень большой длиной ключа, сохранением в секрете S-блоков и удвоением количества итераций. Вопрос, увенчались ли их усилия созданием более безопасного, чем DES, алгоритма, остается открытым.*
>
> *Брюс Шнайер. «Прикладная криптография»*

8.5. AES [57]

AES (Advanced Encryption Standard, расширенный стандарт шифрования) - американский стандарт симметричного блочного шифрования, введённый в действие с 26 мая 2002 г. (FIPS PUB 197 «Announcing the ADVANCED ENCRYPTION STANDARD (AES)»). Является упрощенной версией алгоритма Rijndael (авторы Винсент Рэймен и Йоан Даймен), поддерживающего более широкий набор длин блоков (от 128 до 256 битов с шагом в 32 бита).

По типу относится к подстановочно-перестановочным сетям (англ. Substitution-Permutation network, SP-сеть). В алгоритме используются методы замены, перестановки и гаммирования (сложение по модулю 2).

Основные параметры AES приведены в следующей таблицы.

Таблица 8.10. Основные параметры AES

Тип шифра	Длина ключа N_k, битов (слов[1])	Размер блока зашифрования / расшифрования N_b, битов (слов)	Количество раундов N_r	Количество ключевых элементов[2] $N_w = N_b (N_r + 1)$
AES-128	128 (4)		10	44

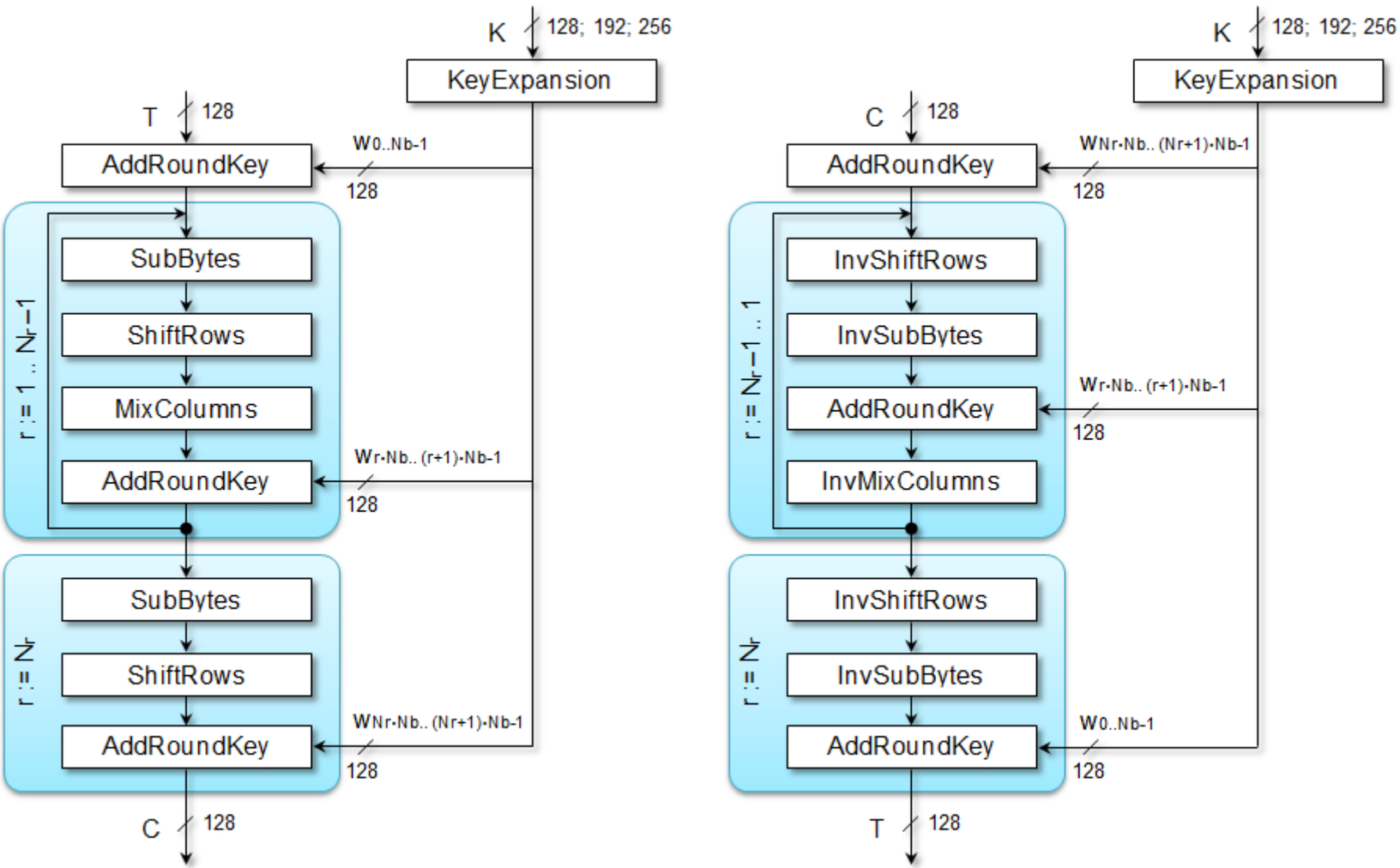
AES-192	192 (6)	128 (4)	12	52
AES-256	256 (8)		14	60

Примечания.

[1]Слово (англ. word) – последовательность из 4 байтов (32 битов).

[2]Ключевой элемент – 4-байтовое слово, формируемое из ключа.

Схемы зашифрования и расшифрования одного блока изображены на следующем рисунке.



а) зашифрование

б) расшифрование

Рис.8.12

А. Формирование ключевых элементов – KeyExpansion.

Перед процедурой зашифрования или расшифрования из ключа **K** формируются набор ключевых элементов $\mathbf{w_i}$. На следующем рисунке отображена схема формирования ключевых элементов при длине ключа $\mathbf{N_k}$ = 128 битов. При большей длине ключа (192 и 256 битов) имеются особенности получения $\mathbf{w_i}$ (см. спецификацию [57]).

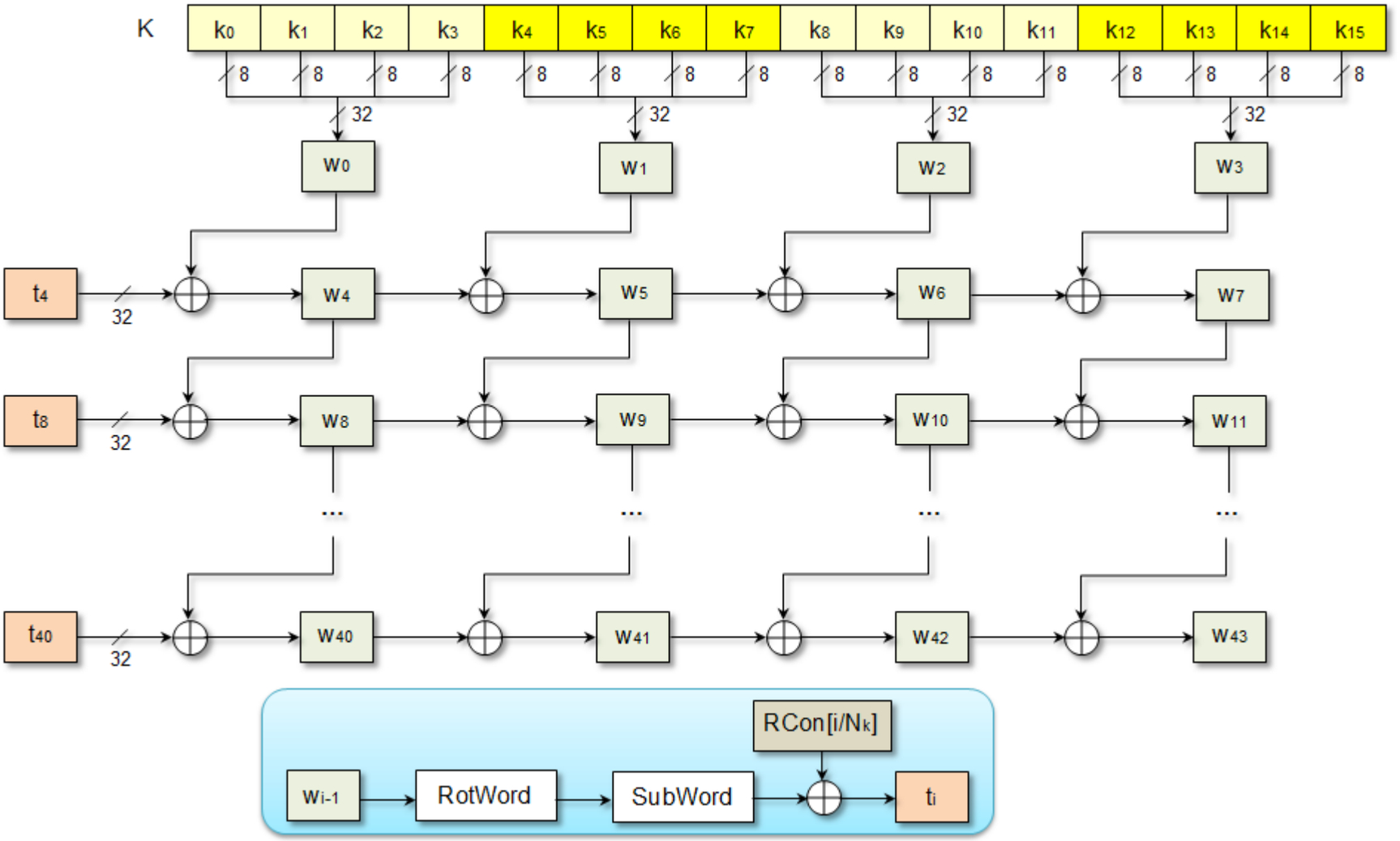
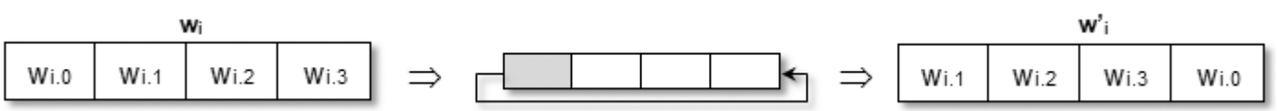


Рис.8.13. Схема формирования ключевых элементов (AES-128)

A.1. RotWord.

Циклический сдвиг в слове $\mathbf{w_i}$ на один байт влево.



A.2. SubWord.

Замена каждого байта слова $\mathbf{w_i}$ на байт **S-box** в соответствии со следующей таблицей.

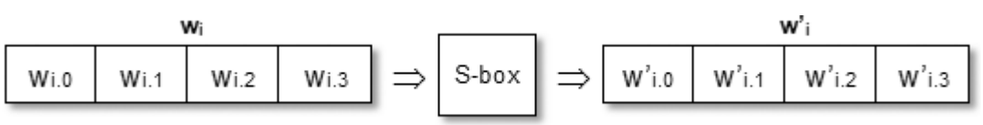


Таблица 8.11. S-box
(16-теричное представление)

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76

1_	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2_	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3_	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4_	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5_	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6_	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7_	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8_	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9_	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A_	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B_	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C_	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D_	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E_	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F_	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

При этом байт в 16-теричном представлении состоит из двух символов, первый из которых указывает на номер строки **S-box**, а второй – на номер столбца. Например, если один из байтов $\mathbf{w_i} = C5_{16}$, то он заменяется на $\mathbf{w'_i} = A6_{16}$.

Иначе, таблицу **S-box** можно представить в виде одномерного массива. Тогда $\mathbf{w_i}$ – это индекс элемента массива, который служит заменой – $\mathbf{w'_i = S\text{-}box[w_i]}$.

A.3. RCon.

4-байтовая константа.

Rcon

$2^{i / Nk - 1}$	0	0	0

B. Зашифрование.

Изначально шифруемый блок **T** представляется массивом байтов **state** размером 4x4. При этом согласно спецификации AES [57] байты выписываются в массив по столбцам.

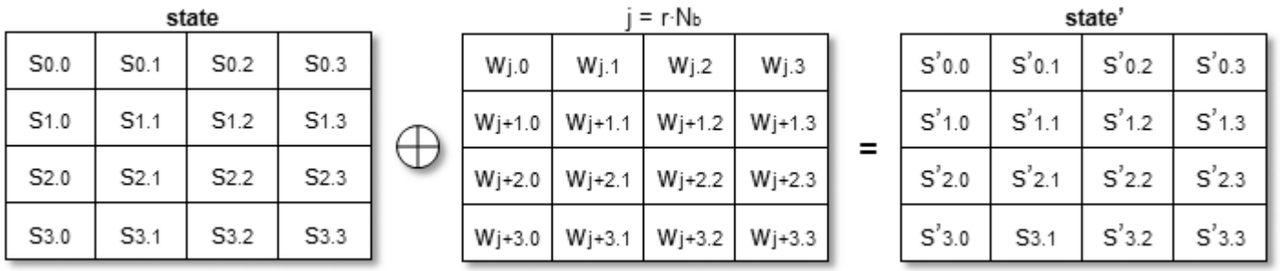
$T = state_0 =$

S0.0	S0.1	S0.2	S0.3
S1.0	S1.1	S1.2	S1.3
S2.0	S2.1	S2.2	S2.3
S3.0	S3.1	S3.2	S3.3

Далее над массивом **state** выполняются последовательные преобразования.

B.1. AddRoundKey.

Сложение **state** по модулю 2 с четырьмя 4-байтовыми ключевыми элементами.

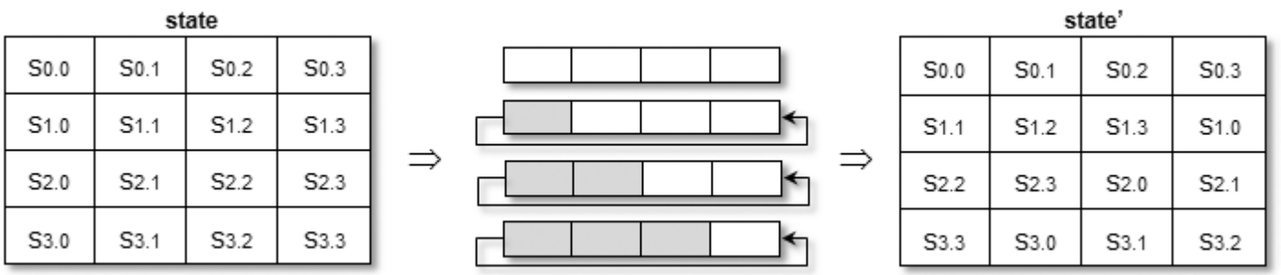


B.2. SubBytes.

Замена байтов **state** по аналогии с **SubWord** в соответствии с табл.8.11.

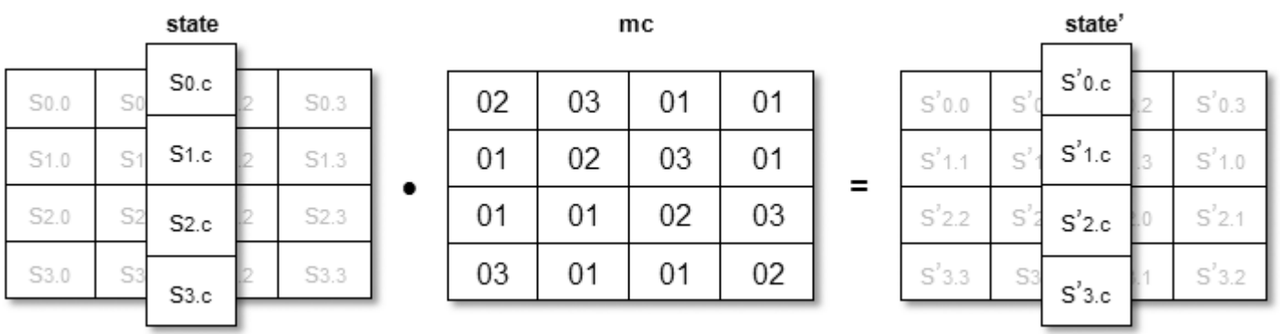
B.3. ShiftRows.

Построчный циклический сдвиг байтов **state** влево на 0, 1, 2 и 3 байта.



B.4. MixColumns.

Смешивание байтов в столбцах **state**. Достигается за счет умножения вектора-столбца **state** на квадратную матрицу **mc**. Умножение выполняется в поле Галуа $GF(2^8)$ по модулю $x^8 + x^4 + x^3 + x + 1$ – обозначается жирной точкой (•).



В результате выполняется замена байтов по формулам:

$$s'_{0.c} = (02 \bullet s_{0.c}) \oplus (03 \bullet s_{1.c}) \oplus s_{2.c} \oplus s_{3.c}$$

$$s'_{1.c} = s_{0.c} \oplus (02 \bullet s_{1.c}) \oplus (03 \bullet s_{2.c}) \oplus s_{3.c}$$

$$s'_{2.c} = s_{0.c} \oplus s_{1.c} \oplus (02 \bullet s_{2.c}) \oplus (03 \bullet s_{3.c})$$

$$s'_{3.c} = (03 \bullet s_{0.c}) \oplus s_{1.c} \oplus s_{2.c} \oplus (02 \bullet s_{3.c})$$

Операция **s • mc** выполняется по правилам полиномиальной арифметики. Для этого элементы вектора и матрицы представляются в двоичном виде. Например, $[s_{0.0}, s_{1.0}, s_{2.0}, s_{3.0}] = [D4, BF, 5D, 30]_{16} = [11010100, 10111111, 1011101, 110000]_2$ и $[mc_{0.0}, mc_{0.1}, mc_{0.2}, mc_{0.3}] = [02, 03, 01, 01]_{16} = [10, 11, 01, 01]_2$.

Дальнейшие преобразования иллюстрируются на примере умножения и деления в столбик. Умножение выполняется классически,

но со сложением единиц в столбцах по модулю 2.

```
*   11010100      *   10111111
          10                11
  ----------        ----------
⊕   00000000      ⊕   10111111
   11010100          10111111
  ----------        ----------
   110101000         111000001
```

Взятие остатка от произведения по модулю $x^8 + x^4 + x^3 + x + 1$ (= 100011011_2) в поле Галуа GF(2^8) выполняется делением в столбик до тех пор, пока остаток превышает 8 битов. При этом вместо вычитания делителя из делимого также используется сложение по модулю 2.

```
⊕  110101000      ⊕  111000001
   100011011         100011011
  ----------        ----------
    10110011          11011010
```

Для $s_{2.0} = 1 \bullet s_{2.0} = 1011101_2$ и $s_{3.0} = 1 \bullet s_{3.0} = 110000_2$ умножение и взятие остатка не требуется, так как их длина не превышает 8 битов. Более сложный пример деления числа $C543_{16} = 1100010101000011_2$ по модулю $x^8 + x^4 + x^3 + x + 1$ в поле Галуа GF(2^8) показан ниже.

```
⊕  1100010101000011
   100011011
⊕   100100011
    100011011
⊕      111000000
       100011011
⊕       110110110
        100011011
⊕        101011011
         100011011
            10000001 = 81_16
```

Окончательное значение элемента $\mathbf{s'_{0.0}}$ вектора **state’** получается сложением по модулю 2 полученных остатков.

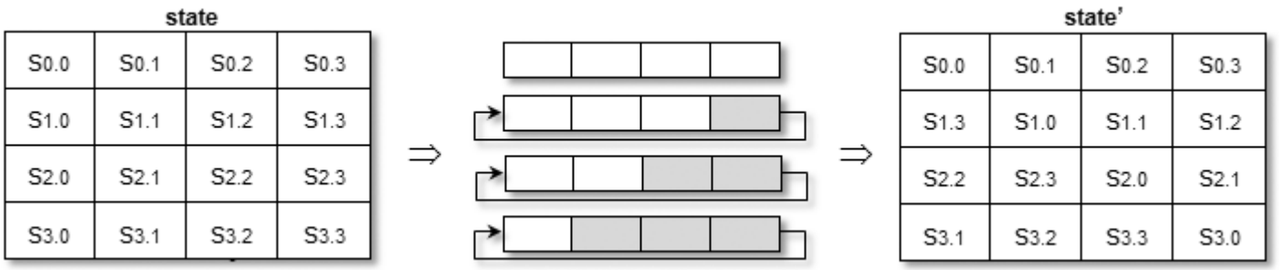
```
   10110011
   11011010
⊕   1011101
     110000
  ----------
   00000100 = 04_16 = s'_0.0
```

C. Расшифрование.

При расшифровании меняется не только порядок использования ключевых элементов (как в DES и ГОСТ 28147-89), но и выполняется инверсия всех операций.

C.1. InvShiftRows.

Построчный циклический сдвиг байтов **state** вправо на 0, 1, 2 и 3 байта.



C.2. InvSubBytes.

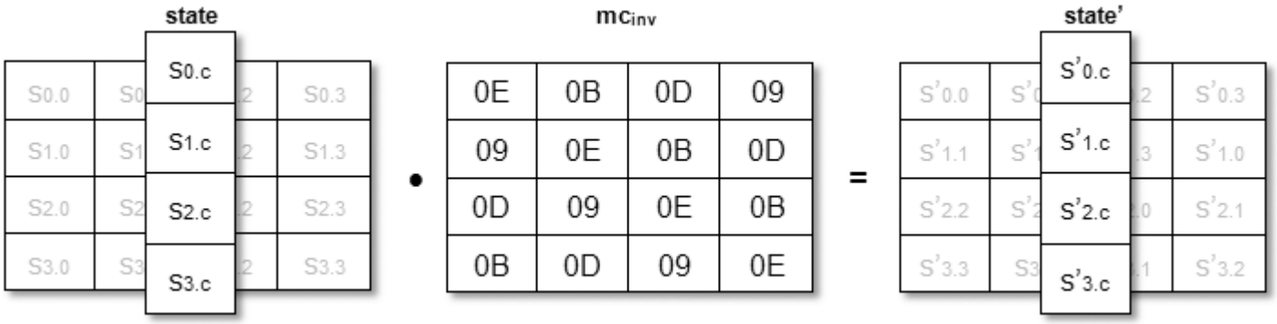
Замена байтов **state** в соответствии со следующей таблицей.

Таблица 8.12. Inverse S-box
(16-теричное представление)

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1_	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2_	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3_	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4_	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5_	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6_	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7_	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8_	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9_	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A_	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B_	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C_	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D_	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E_	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F_	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

C.3. InvMixColumns.

Смешивание байтов в столбцах – умножение вектора-столбца **state** на квадратную матрицу $\mathbf{mc}_{\mathbf{inv}}$ с взятием остатков по модулю $x^8 + x^4 + x^3 + x + 1$ в поле Галуа $GF(2^8)$.



Замена байтов выполняется по формулам:

$$s'_{0.c} = (0E \bullet s_{0.c}) \oplus (0B \bullet s_{1.c}) \oplus (0D \bullet s_{2.c}) \oplus (09 \bullet s_{3.c})$$

$$s'_{1.c} = (09 \bullet s_{0.c}) \oplus (0E \bullet s_{1.c}) \oplus (0B \bullet s_{2.c}) \oplus (0D \bullet s_{3.c})$$

$$s'_{2.c} = (0D \bullet s_{0.c}) \oplus (09 \bullet s_{1.c}) \oplus (0E \bullet s_{2.c}) \oplus (0B \bullet s_{3.c})$$

$$s'_{3.c} = (0B \bullet s_{0.c}) \oplus (0D \bullet s_{1.c}) \oplus (09 \bullet s_{2.c}) \oplus (0E \bullet s_{3.c})$$

В заключение несколько слов о стойкости шифра. В отличие от большинства других шифров, AES имеет простое математическое описание. Это обстоятельство беспокоит многих криптографов. Вот что по этому поводу написали известные популяризаторы криптографии Нильс Фергюсон и Брюс Шнайер.

> *У нас есть одно критическое замечание к AES: мы не совсем доверяем его безопасности. Что беспокоит нас больше всего в AES, так это его простая алгебраическая структура… Ни один другой блочный шифр не имеет столь простого алгебраического представления. Мы понятия не имеем, ведёт это к атаке или нет, но незнание этого является достаточной причиной, чтобы скептически относиться к использованию AES.*
>
> *Нильс Фергюсон, Брюс Шнайер. «Практическая криптография»*

8.6. ГОСТ 34.12-2015 [58]

ГОСТ 34.12-2015 – российский стандарт симметричного блочного шифрования, введённый в действие 19 июня 2015 г. («ГОСТ 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры»).

Содержит описание двух блочных шифров:

- **«Магма»** («Magma») – с длиной блока n = 64 бит;

- **«Кузнечик»** («Kuznyechik») – с длиной блока n = 128 бит.

Длина ключа для обоих шифров – K = 256 бит.

«Магма».

Полностью соответствует ГОСТ 28147-89. В отличие от старого ГОСТ в новом явно определены значения **S-блоков** (подстановок **π**).

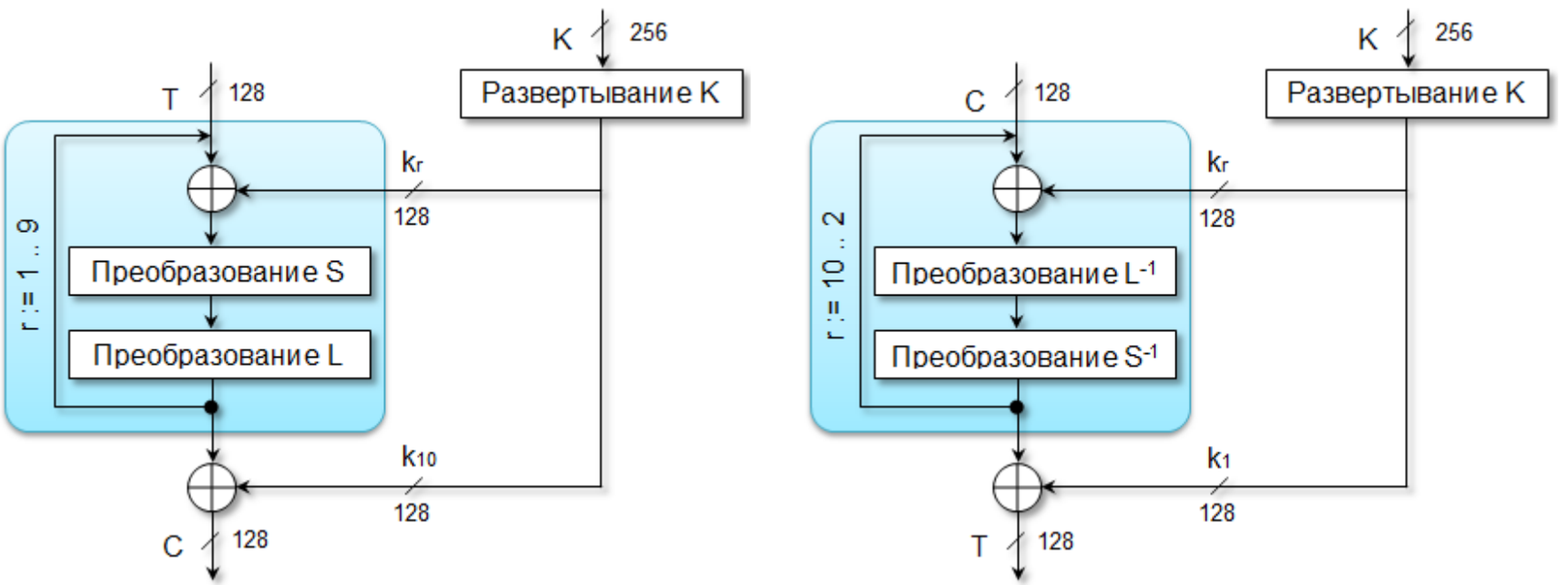
Таблица 8.13. Значения y_{ij} (π)

S-блок (S_j)	Значение x_{ij}															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	12	4	6	2	10	5	11	9	14	8	13	7	0	3	15	1
S_2	6	8	2	3	9	10	5	12	1	14	4	7	11	13	0	15
S_3	11	3	5	8	2	15	10	13	14	1	7	4	12	9	6	0
S_4	12	8	2	1	13	4	15	6	7	0	10	5	3	14	9	11
S_5	7	15	5	10	8	1	6	13	0	9	3	14	11	4	2	12
S_6	5	13	15	6	9	2	12	10	11	7	8	1	4	3	14	0
S_7	8	14	2	5	6	9	1	12	15	4	11	0	13	10	3	7
S_8	1	7	14	13	0	5	8	3	4	15	10	6	9	12	11	2

«Кузнечик».

По типу, как и AES, относится к подстановочно-перестановочным сетям. В алгоритме развертывания ключа (процедуре формирования итерационных ключей – ключевых элементов) используется сеть Фейстеля.

Схема зашифрования / расшифрования одного блока изображена на следующем рисунке.



а) зашифрование

б) расшифрование

Рис.8.14

А. Развертывание K.

Формирование 10 итерационных ключей (ключевых элементов) выполняется по следующей схеме.

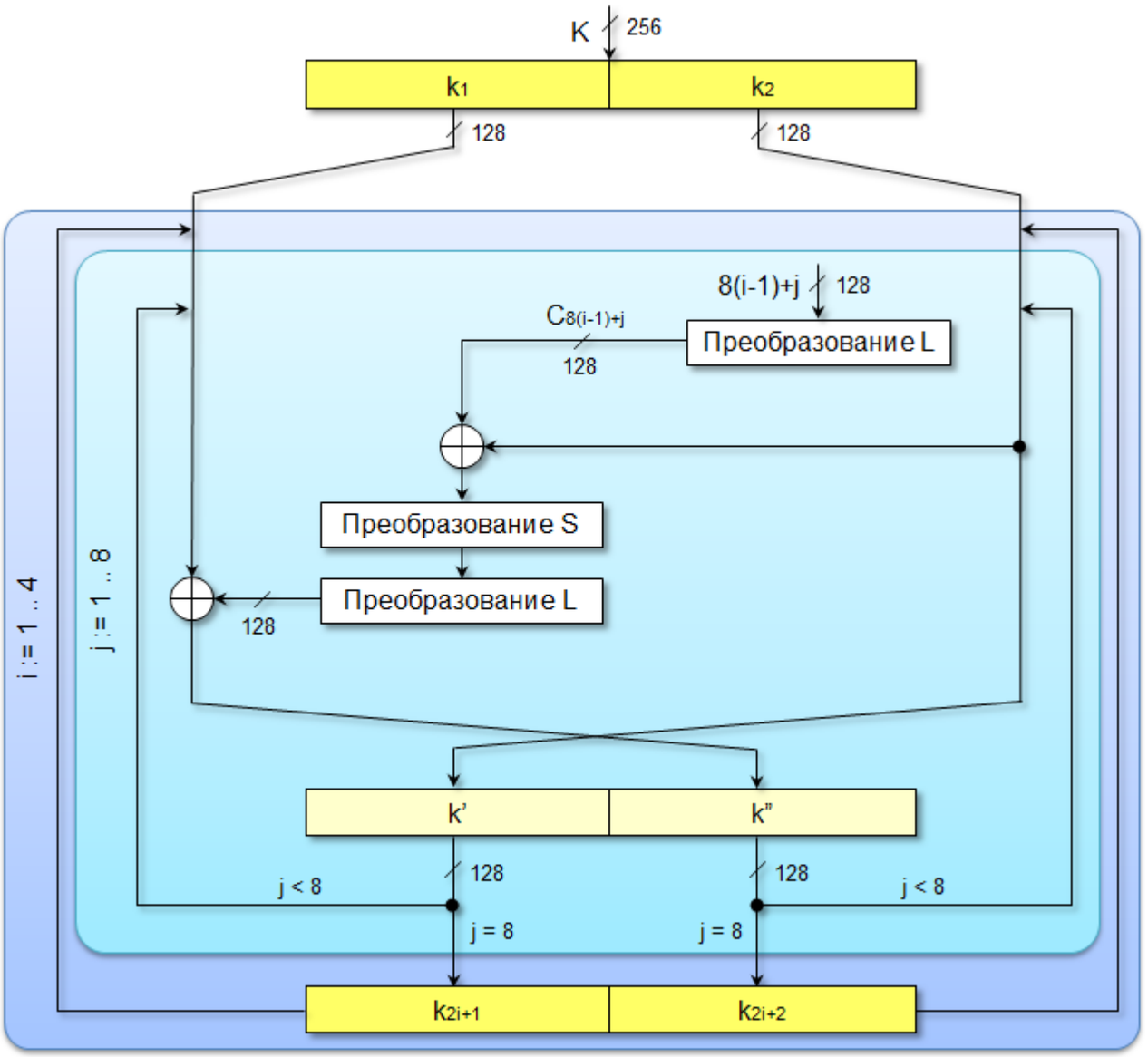


Рис.8.15. Развертывание K

Изначально ключ делится на две половины по 128 битов с получение первых двух итерационных ключей ($\mathbf{k_1}$ и $\mathbf{k_2}$).

Далее, в течение 4 раундов **i**, формируются остальные 8 итерационных ключей ($\mathbf{k_{2i+1}}$ и $\mathbf{k_{2i+2}}$). Каждая последующая пара итерационных ключей получается в результате прогона предыдущей пары через сеть, состоящую из 8 ячеек Фейстеля (в ГОСТ ячейка обозначена **F**). При этом в начале каждой итерации **j** (ячейки **F**) должна быть получена итерационная константа $\mathbf{C_{8(i-1)+j}}$ (не путать с обозначением блока шифрограммы **C** на рис.8.14). Для этого целое число **8(i-1)+j**, представленное в двоичном виде на 128 битов, должно пройти преобразование **L**.

A.1. Линейное преобразование L.

Преобразование байтов $\mathbf{a_i}$ входного 128-битового (16-байтового) массива **V** в течение 16 раундов по следующей схеме.

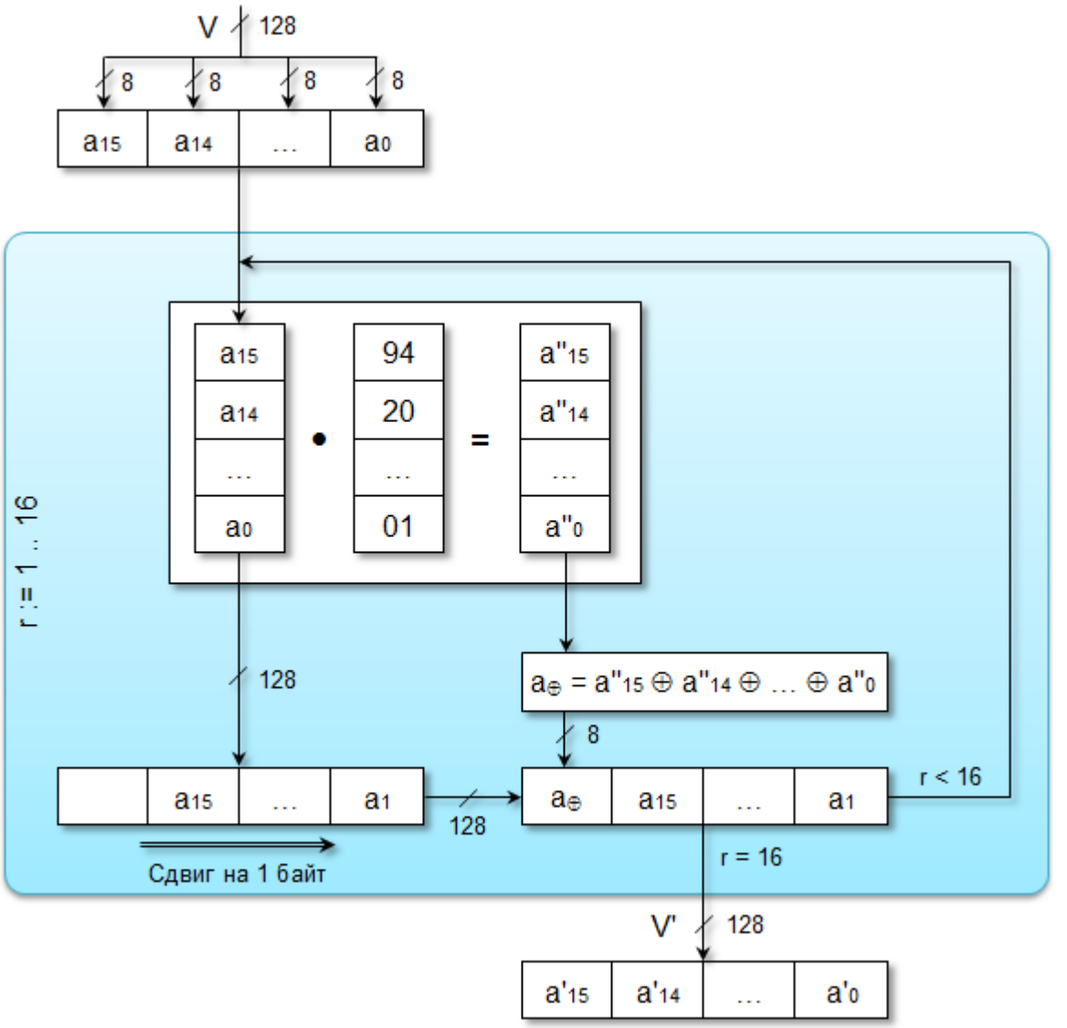


Рис.8.16. Линейное преобразование L

Один раунд состоит из следующей последовательности операций:

- умножение (•) каждого байта $\mathbf{a_i}$ массива **V** на соответствующую константу, определенную в ГОСТ;

- сложение по модулю 2 результатов умножения ($\mathbf{a_{\oplus}}$);

- сдвиг вправо элементов массива **V** на 1 байт;

- занесение $\mathbf{a_{\oplus}}$ в качестве первого (старшего) байта массива **V**.

Массив констант по ГОСТ представлен в следующей таблице.

Таблица 8.14. Массив констант

№ п/п	Основание системы счисления		
	16	10	2
1	94	148	10010100
2	20	32	00100000
3	85	133	10000101
4	10	16	00010000

5	C2	194	11000010
6	C0	192	11000000
7	01	1	00000001
8	FB	251	11111011
9	01	1	00000001
10	C0	192	11000000
11	C2	194	11000010
12	10	16	00010000
13	85	133	10000101
14	20	32	00100000
15	84	148	10010100
16	01	1	00000001

Проиллюстрируем выполнение операций на примере 4 раунда определения итерационной константы $\mathbf{C_1}$ (см. п.А.1.4 приложения А к ГОСТ 34.12-2015).

Массив **V** в начале итерации – $[84, 94, 01, 00, 00, \ldots, 00]_{16} = [10000100, 10010100, 00000001, 00000000, 00000000, \ldots, 00000000]_2$.

Умножение (•) байта на константу выполняется в поле Галуа $GF(2^8)$ по модулю $x^8 + x^7 + x^6 + x + 1$ ($=111000011_2 = 1C3_{16}$) в соответствии с правилами полиномиальной арифметики (см. функцию MixColumns шифра AES):

- полиномиальное умножение:

```
          84 * 94                  94 * 20                  01 * 85
  *    10000100            *    10010100            *    00000001
       10010100                 00100000                 10000101
       --------                 --------                 --------
       00000000                 00000000                 00000001
      00000000                 00000000                 00000000
     10000100                 00000000                 00000001
⊕   00000000             ⊕   00000000             ⊕   00000000
    10000100                 00000000                 00000000
   00000000                 10010100                 00000000
  00000000                 00000000                 00000000
 10000100                 00000000                 00000001
 --------                 --------                 --------
100100001010000 = 4850₁₆  001001010000000 = 1280₁₆  000000010000101 = 85₁₆
```

- полиномиальное деление (для 85_{16} не требуется, т.к. число не превышает 8 битов):

```
          4850 mod 1C3                    1280 mod 1C3
⊕  100100001010000               ⊕  1001010000000
   111000011                        111000011
   -----------                      -----------
⊕    111000100                   ⊕    111010110
     111000011                        111000011
     ---------                        ---------
     11110000 = F0₁₆                  10101000 = A8₁₆
```

Сложение по модулю 2 результатов умножения - $\mathbf{a}_{\oplus}$.

```
   11110000
   10101000
⊕  10000101
   00000000
```

```
  ...
 00000000
 11011101 = DD16
```

Сдвиг массива **V** вправо – $[\ \_, 84, 94, 01, 00, \ldots, 00]_{16}$.

Занесение $\mathbf{a}_{\oplus}$ в качестве первого (старшего) байта массива **V** – $[DD, 84, 94, 01, 00, \ldots, 00]_{16}$.

A.2. Нелинейное биективное преобразование S.

Замена каждого каждого байта $\mathbf{a_i}$ входного 128-битового (16-байтового) массива **V** на байт подстановки $\boldsymbol{\pi}$ в соответствии со следующей таблицей.

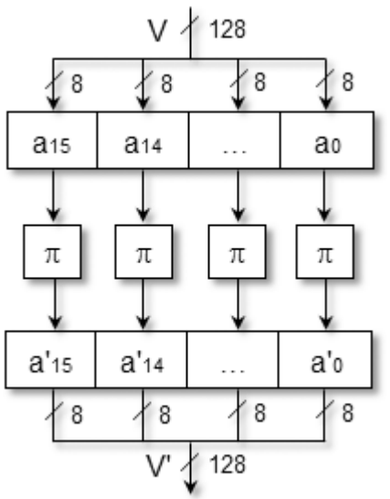


Таблица 8.15. Подстановки π
(16-теричное представление)

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	FC	EE	DD	11	CF	6E	31	16	FB	C4	FA	DA	23	C5	04	4D
1_	E9	77	F0	DB	93	2E	99	BA	17	36	F1	BB	14	CD	5F	C1
2_	F9	18	65	5A	E2	5C	EF	21	81	1C	3C	42	8B	01	8E	4F
3_	05	84	02	AE	E3	6A	8F	A0	06	0B	ED	98	7F	D4	D3	1F
4_	EB	34	2C	51	EA	C8	48	AB	F2	2A	68	A2	FD	3A	CE	CC
5_	B5	70	0E	56	08	0C	76	12	BF	72	13	47	9C	B7	5D	87
6_	15	A1	96	29	10	7B	9A	C7	F3	91	78	6F	9D	9E	B2	B1
7_	32	75	19	3D	FF	35	8A	7E	6D	54	C6	80	C3	BD	0D	57
8_	DF	F5	24	A9	3E	A8	43	C9	D7	79	D6	F6	7C	22	B9	03
9_	E0	0F	EC	DE	7A	94	B0	BC	DC	E8	28	50	4E	33	0A	4A
A_	A7	97	60	73	1E	00	62	44	1A	B8	38	82	64	9F	26	41
B_	AD	45	46	92	27	5E	55	2F	8C	A3	A5	7D	69	D5	95	3B
C_	07	58	B3	40	86	AC	1D	F7	30	37	6B	E4	88	D9	E7	89
D_	E1	1B	83	49	4C	3F	F8	FE	8D	53	AA	90	CA	D8	85	61
E_	20	71	67	A4	2D	2B	09	5B	CB	9B	25	D0	BE	E5	6C	52
F_	59	A6	74	D2	E6	F4	B4	C0	D1	66	AF	C2	39	4B	63	B6

При этом байт $\mathbf{a_i}$ в 16-теричном представлении состоит из двух символов, первый из которых указывает на номер строки подстановки $\boldsymbol{\pi}$, а второй – на номер столбца. Например, если один из байтов $\mathbf{a_i} = C5_{16}$, то он заменяется на $\mathbf{a'_i} = AC_{16}$.

Иначе, таблицу подстановок $\boldsymbol{\pi}$ можно представить в виде одномерного массива. Тогда $\mathbf{a_i}$ – это индекс элемента массива, который служит заменой – $\mathbf{a'_i = \pi[a_i]}$.

B. Зашифрование.

Зашифрование представляется собой последовательное применение к исходному блоку текста **T** в течение 9 раундов **r** гаммирования по модулю 2 с итерационными ключами $\mathbf{k_r}$, а также преобразований **S** и **L** (см. **A.1** и **A.2**). Для окончательного получения блока шифрограммы **C** результат раундовых преобразований складывается по модулю 2 с последним итерационным ключом $\mathbf{k_{10}}$.

C. Расшифрование.

При расшифровании меняется не только порядок использования итерационных ключей (как в DES и ГОСТ 28147-89), но и выполняется инверсия всех операций.

C.1. Линейное преобразование $\mathbf{L^{-1}}$.

Преобразование байтов $\mathbf{a_i}$ входного 128-битового (16-байтового) массива **V** в течение 16 раундов по следующей схеме.

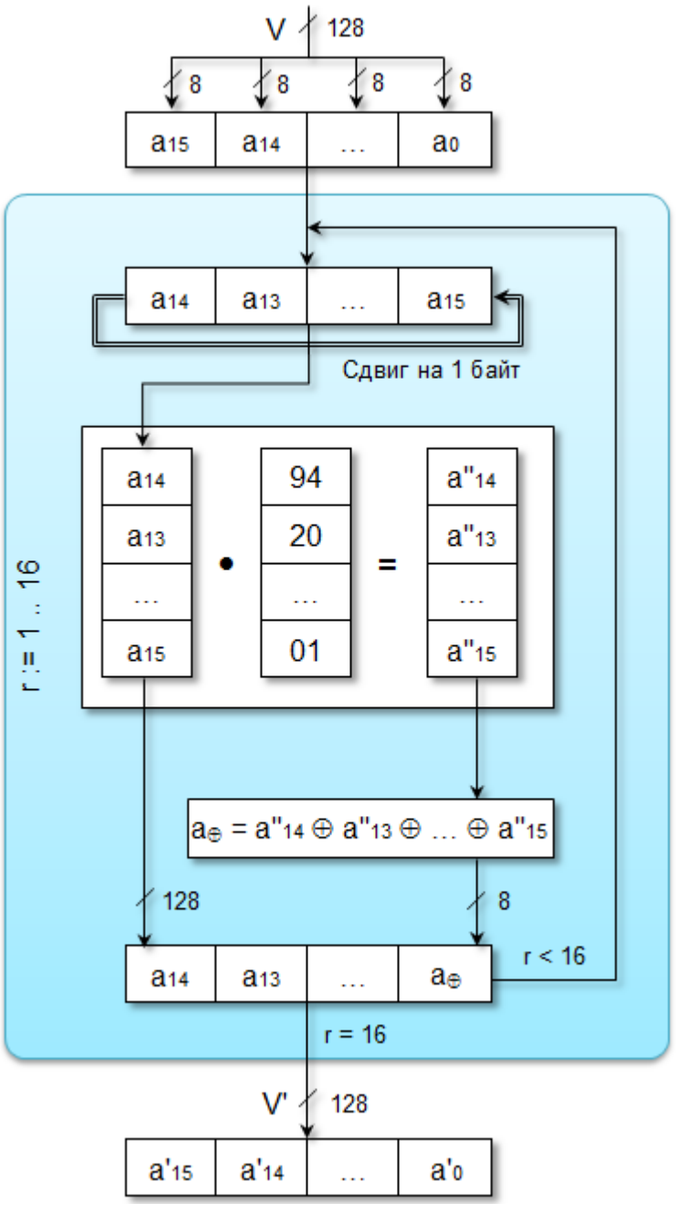


Рис.8.17. Линейное преобразование L^{-1}

Обратное линейное преобразование $\mathbf{L^{-1}}$ выполняется по тем же правилам, что и прямое **L**. К особенностям преобразования $\mathbf{L^{-1}}$ следует отнести:

- сдвиг выполняется в начале итерации;

- используется циклический сдвиг влево вместо простого сдвига вправо;

- результат сложения $\mathbf{a_{\oplus}}$ заносится как последний (младший) байт массива **V**.

С.2. Нелинейное биективное преобразование S^{-1}.

Замена байтов $\mathbf{a_i}$ входного 128-битового массива (16-байтового) **V** на байт подстановки $\boldsymbol{\pi^{-1}}$ в соответствии со следующей таблицей.

Таблица 8.16. Подстановки π^{-1}
(16-теричное представление)

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F

0_	A5	2D	32	8F	0E	30	38	C0	54	E6	9E	39	55	7E	52	91
1_	64	03	57	5A	1C	60	07	18	21	72	A8	D1	29	C6	A4	3F
2_	E0	27	8D	0C	82	EA	AE	B4	9A	63	49	E5	42	E4	15	B7
3_	C8	06	70	9D	41	75	19	C9	AA	FC	4D	BF	2A	73	84	D5
4_	C3	AF	2B	86	A7	B1	B2	5B	46	D3	9F	FD	D4	0F	9C	2F
5_	9B	43	EF	D9	79	B6	53	7F	C1	F0	23	E7	25	5E	B5	1E
6_	A2	DF	A6	FE	AC	22	F9	E2	4A	BC	35	CA	EE	78	05	6B
7_	51	E1	59	A3	F2	71	56	11	6A	89	94	65	8C	BB	77	3C
8_	7B	28	AB	D2	31	DE	C4	5F	CC	CF	76	2C	B8	D8	2E	36
9_	DB	69	B3	14	95	BE	62	A1	3B	16	66	E9	5C	6C	6D	AD
A_	37	61	4B	B9	E3	BA	F1	A0	85	83	DA	47	C5	B0	33	FA
B_	96	6F	6E	C2	F6	50	FF	5D	A9	8E	17	1B	97	7D	EC	58
C_	F7	1F	FB	7C	09	0D	7A	67	45	87	DC	E8	4F	1D	4E	04
D_	EB	F8	F3	3E	3D	BD	8A	88	DD	CD	0B	13	98	02	93	80
E_	90	D0	24	34	CB	ED	F4	CE	99	10	44	40	92	3A	01	26
F_	12	1A	48	68	F5	81	8B	C7	D6	20	0A	08	00	4C	D7	74

8.7. ГОСТ 34.13-2015 [59]

ГОСТ 34.13-2015 – российский стандарт симметричного блочного шифрования, введённый в действие 19 июня 2015 г. («ГОСТ 34.13-2015. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров»).

Является дополнением к ГОСТ 34.12-2015 и определяет режимы работы блочных шифров:

- простой замены (Electronic Codebook, ECB);

- гаммирования (Counter, CTR);

- гаммирования с обратной связью по выходу (Output Feedback, OFB);

- простой замены с зацеплением (Cipher Block Chaining, CBC);

- гаммирования с обратной связью по шифртексту (Cipher Feedback, CFB);

- выработки имитовставки (Message Authentication Code algorithm).

Для каждого из режимов в стандарте приведены схемы зашифрования / расшифрования, которые в качестве базовых операций используют процедуры зашифрования / расшифрования по ГОСТ 34.12-2015.

Вопросы для самопроверки

1. Дайте описание ячейки Фейстеля.

2. Назовите основные режимы DES.

3. Перечислите методы шифрования, используемые в DES-ECB.

4. Приведите схему алгоритма DES в режиме сцепления блоков шифра.

5. Назовите сферы применения разных режимов DES

6. Какова длина ключа шифра ГОСТ и как генерируются ключевые элементы.

7. Назовите основные различия между DES и ГОСТ.

<

9. ШИФРОВАНИЕ С ОТКРЫТЫМ КЛЮЧОМ

9.1. Основы шифрования

Главная проблема использования одноключевых (симметричных) криптосистем заключается в распределении ключей. Для того, чтобы был возможен обмен информацией между двумя сторонами, ключ должен быть сгенерирован одной из них, а затем в конфиденциальном порядке передан другой. Особую остроту данная проблема приобрела в наши дни, когда криптография стала общедоступной, вследствие чего количество пользователей больших криптосистем может исчисляться сотнями и тысячами.

Начало асимметричным шифрам было положено в работе «Новые направления в современной криптографии» Уитфилда Диффи и Мартина Хеллмана, опубликованной в 1976 г. Находясь под влиянием работы Ральфа Меркла (Ralph Merkle) о распространении открытого ключа, они предложили метод получения секретных ключей для симметричного шифрования, используя открытый канал. В 2002 г. Хеллман предложил называть данный алгоритм «Диффи - Хеллмана - Меркла», признавая вклад Меркла в изобретение криптографии с открытым ключом [17].

Хотя работа Диффи-Хеллмана создала большой теоретический задел для открытой криптографии, первой реальной криптосистемой с открытым ключом считают алгоритм RSA (названный по имени авторов - Рон Ривест (Ronald Linn Rivest), Ади Шамир (Adi Shamir) и Леонард Адлеман (Leonard Adleman)).

Справедливости ради следует отметить, что в декабре 1997 г. была обнародована информация, согласно которой британский математик Клиффорд Кокс (Clifford Cocks), работавший в центре правительственной связи (GCHQ) Великобритании, описал систему, аналогичную RSA, в 1973 г., а несколькими месяцами позже в 1974 г. Малькольм Вильямсон изобрел математический алгоритм, аналогичный алгоритму Диффи – Хеллмана - Меркла.

Суть шифрования с открытым ключом заключается в том, что для шифрования данных используется один ключ, а для расшифрования другой (поэтому такие системы часто называют **асимметричными**).

Основная предпосылка, которая привела к появлению шифрования с открытым ключом, заключалось в том, что отправитель сообщения (тот, кто зашифровывает сообщение), не обязательно должен быть способен его расшифровывать. Т.е. даже имея исходное сообщение, ключ, с помощью которого оно шифровалось, и зная алгоритм шифрования, он не может расшифровать закрытое сообщение без знания ключа расшифрования.

Первый ключ, которым шифруется исходное сообщение, называется **открытым** и может быть опубликован для использования всеми пользователями системы. Расшифрование с помощью этого ключа невозможно. Второй ключ, с помощью которого дешифруется сообщение, называется **секретным** (закрытым) и должен быть известен только законному получателю закрытого сообщения.

Алгоритмы шифрования с открытым ключом используют так называемые необратимые или односторонние функции. Эти функции обладают следующим свойством: при заданном значении аргумента **x** относительно просто вычислить значение функции **f(x)**, однако, если известно значение функции **y = f(x)**, то нет простого пути для вычисления значения аргумента **x**. Например, функция **SIN**. Зная x, легко найти значение **SIN(x)** (например, x = π, тогда SIN(π) = 0). Однако, если SIN(x) = 0, однозначно определить **x** нельзя, т.к. в этом случае **x** может быть любым числом, определяемым по формуле **i * π**, где **i** – целое число.

Однако не всякая необратимая функция годится для использования в реальных криптосистемах. В их числе и функция **SIN**. Следует также отметить, что в самом определении необратимости функции присутствует неопределенность. Под необратимостью понимается не теоретическая необратимость, а практическая невозможность вычислить обратное значение, используя современные вычислительные средства за обозримый интервал времени.

Поэтому чтобы гарантировать надежную защиту информации, к криптосистемам с открытым ключом предъявляются два важных и очевидных **требования**.

1. Преобразование исходного текста должно быть условно необратимым и исключать его восстановление на основе открытого ключа.

2. Определение закрытого ключа на основе открытого также должно быть невозможным на современном технологическом уровне.

Все предлагаемые сегодня криптосистемы с открытым ключом опираются на один из следующих **типов односторонних преобразований**.

1. Разложение больших чисел на простые множители (алгоритм RSA).

2. Вычисление дискретного логарифма или дискретное возведение в степень (алгоритм Диффи-Хелмана-Меркла, схема Эль-Гамаля).

3. Задача об укладке рюкзака (ранца) (авторы Хелман и Меркл).

4. Вычисление корней алгебраических уравнений.

5. Использование конечных автоматов (автор Тао Ренжи).

6. Использование кодовых конструкций.

7. Использование свойств эллиптических кривых.

9.2. Алгоритм RSA

Стойкость RSA основывается на большой вычислительной сложности известных алгоритмов разложения числа на простые сомножители (делители). Например, легко найти произведение двух простых чисел 7 и 13 даже в уме – 91. Попробуйте в уме найти два простых числа, произведение которых равно 323 (числа 17 и 19). Конечно, для современной вычислительной техники найти два простых числа, произведение которых равно 323, не проблема. Поэтому для надежного шифрования алгоритмом RSA, как правило, выбираются простые числа, количество двоичных разрядов которых равно нескольким сотням.

[44] В августе 1977 г. знаменитый американский писатель и популяризатор науки Мартин Гарднер озаглавил свою колонку по занимательной математике в журнале Scientific American так: «Новый вид шифра, на расшифровку которого потребуются миллионы лет». После объяснения принципа системы шифрования с открытым ключом он показал само зашифрованное сообщение и открытый ключ N, используемый в этом шифре:

N = 114 381 625 757 888 867 669 235 779 976 146 612 010 218 296 721 242 362 562 561 842 935 706 935 245 733 897 830 597 123 563 958 705 058 989 075 147 599 290 026 879 543 541.

Гарднер призвал читателей попробовать расшифровать сообщение, используя предоставленную информацию, и даже дал подсказку: для решения необходимо разложить число N на простые множители p и q. Более того, Гарднер назначил приз в размере $100 (приличная сумма на тот момент) тому, кто первым получит правильный ответ. Каждый, кто захочет побольше узнать о шифре, писал Гарднер, может обратиться к создателям шифра — Рону Ривесту, Ади Шамиру и Леонарду Адлеману из Лаборатории информации Массачусетского технологического института.

Правильный ответ был получен лишь через 17 лет. Он стал результатом сотрудничества более чем 600 человек. Ключами оказались p = 32 769 132 993 266 709 549 961 988 190 834 461 413 177 642 967 992 942 539 798 288 533 и q = 3 490 529 510 847 650 949 147 849 619 903 898 133 417 764 638 493 387 843 990 820 577, а зашифрованная фраза звучала так: «Волшебные слова — это брезгливый ягнятник».

Авторы RSA поддерживали идею её активного распространения. В свою очередь, Агентство национальной безопасности (США), опасаясь использования этого алгоритма в негосударственных структурах, на протяжении нескольких лет безуспешно требовало прекращения распространения системы. Ситуация порой доходила до абсурда. Например, когда программист Адам Бек (Adam Back) описал на языке Perl алгоритм RSA, состоящий из пяти строк, правительство США запретило распространение этой программы за пределами страны. Люди, недовольные подобным ограничением, в знак протеста напечатали текст этой программы на своих футболках [17].

Первым этапом любого асимметричного алгоритма является создание получателем шифрограмм пары ключей: открытого и секретного. Для алгоритма RSA этап создания ключей состоит из следующих операций.

Таблица 9.1. Процедура создания ключей

№ п/п	Описание операции	Пример
1	Выбираются два простых числа[1] **p** и **q**.	p=7, q=13
2	Вычисляется произведение **n** = p * q.	n=91
3	Вычисляется **функция Эйлера**[2] **φ(n)**.	φ(n) = (7-1)(13-1) = 91-7-13+1 = 72
4	Выбирается открытый ключ **e** - произвольное число (0 < e < n), взаимно простое[3] с результатом функции Эйлера (e ⊥ φ(n)).	e=5
5	Вычисляется закрытый ключ **d** - обратное число[4] к **e** по модулю φ(n) из соотношения (d * e) mod φ(n) = 1.	(d*5) mod 72 = 1, d = 29
	Публикуются открытый ключ (**e**, **n**) в специальном хранилище, где исключается возможность его подмены (**общедоступном**	

6	**сертифицированном справочнике**).	

Примечания.

1) **Простое число** — натуральное число, большее единицы и не имеющее других натуральных делителей, кроме самого себя и единицы.

2) Результат расчета **функции Эйлера φ(n)** равен количеству положительных чисел, не превосходящих **n** и взаимно простым с **n**. Некоторые случаи и способы расчета функции Эйлера приведены в следующей таблице.

Таблица 9.2. Способы расчета функции Эйлера

Расчетный случай	Формула	Пример (число / расчетная формула / список взаимно простых чисел)
n простое число	$\varphi(n) = n - 1$	n = 7 φ(7) = 7 – 1 = 6 {1, 2, 3, 4, 5, 6}
n = p q произведение двух простых чисел	φ(n) = φ(p) φ(q) = = (p - 1) (q - 1) = n – p – q + 1 (за исключением случая p = q = 2)	n = 15 = 3 * 5 φ(15) = φ(3) φ(5) = (3 - 1) (5 - 1) = 15 - 3 - 5 + 1 = 8 {1, 2, 4, 7, 8, 11, 13, 14}
$n = p^q$ простое число в степени	$\varphi(n) = p^q - p^{q-1}$	n = 9 = 3^2 φ(9) = $3^2 - 3^{2-1}$ = 9 - 3 = 6 {1, 2, 4, 5, 7, 8}
$n = p_1^{q1} p_2^{q1} \dots p_k^{qk}$ разложение числа согласно основной теоремы арифметики (общий случай)	$\varphi(n) = \varphi(p_1^{q1})\varphi(p_2^{q2}) \dots \varphi(p_k^{qk}) =$ $= p_1^{q1}\left(1-\frac{1}{p_1}\right)p_2^{q2}\left(1-\frac{1}{p_2}\right)\dots p_k^{qk}\left(1-\frac{1}{p_k}\right) =$ $= n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\dots\left(1-\frac{1}{p_k}\right)$	n = 84 = 2^2 3^1 7^1 $\varphi(84) = 84\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right)\left(1-\frac{1}{7}\right) = 24$ {1, 5, 11, 13, 17, 19, 23, 25, 29, 31, 37, 41, 43, 47, 53, 55, 59, 61, 65, 67, 71, 73, 79, 83}

3) **Взаимно простые числа** – числа, не имеющие общих делителей, кроме 1, т.е. наибольший общий делитель которых равен 1.

4) **Обратными числами по модулю m** называются такие числа **n** и $\mathbf{n^{-1}}$, для которых справедливо выражение $\mathbf{(n * n^{-1}) \bmod m = 1}$. Для вычисления обратных чисел по модулю обычно используется расширенный алгоритм Евклида. В безмодульной математике **обратное число** $\mathbf{n^{-1}}$ (обратное значение, обратная величина) - число, на которое надо умножить данное число **n**, чтобы получить единицу ($\mathbf{n * n^{-1} = 1}$). Пара чисел, произведение которых равно единице, называются **взаимно обратными**. Например: 5 и 1/5, -6/7 и -7/6.

Процедуры шифрования и дешифрования выполняются по следующим формулам

$$C = T^e \bmod n, \qquad (9.1)$$

$$T = C^d \bmod n. \qquad (9.2)$$

где T, C - числовые эквиваленты символов открытого и шифрованного сообщения.

Пример шифрования по алгоритму RSA приведен в следующей таблице. Коды букв соответствуют их положению в русском алфавите (начиная с 1).

Таблица 9.3. Пример шифрования по алгоритму RSA

Открытое сообщение, T	Символ	А	Б	Р	А	М	О	В
	Код	1	2	18	1	14	16	3

Шифрограмма, $C = T^5 \bmod 91$	1	32	44	1	14	74	61
Открытое сообщение, $T = C^{29} \bmod 91$	1	2	18	1	14	16	3

Следует отметить, что **p** и **q** выбираются таким образом, чтобы **n** было больше кода любого символа открытого сообщения. В автоматизированных системах исходное сообщение переводиться в двоичное представление, после чего шифрование выполняется над блоками бит равной длины. При этом длина блока должна быть меньше, чем длина двоичного представления **n**.

[44] Алгоритм RSA требует много машинного времени и очень мощных процессоров. До 1980-х гг. только правительства, армия и крупные предприятия имели достаточно мощные компьютеры для работы с RSA. В результате у них была фактически монополия на эффективное шифрование. Летом 1991 г. Филипп Циммерман, американский физик и борец за сохранение конфиденциальности, предложил бесплатную систему шифрования PGP (англ. Pretty Good Privacy — «достаточно хорошая степень конфиденциальности»), алгоритм которой мог работать на домашних компьютерах. PGP использует классическое симметричное шифрование, что и обеспечивает ей большую скорость на домашних компьютерах, но она шифрует ключи по асимметричному алгоритму RSA.

Циммерман объяснил причины этой меры в открытом письме, которое заслуживает быть процитированным здесь, по крайней мере, частично из-за пророческого описания того, как мы живем, работаем и общаемся два десятилетия спустя.

> *Это личное. Это конфиденциальное. И это только ваше дело и ничье другое. Вы можете планировать политическую кампанию, обсуждать ваши налоги или иметь тайную любовную связь. Или вы можете заниматься тем, что вам не кажется незаконным, хотя таковым является. Что бы то ни было, вы не хотите, чтобы ваши личные электронные письма или конфиденциальные документы были прочитаны кем-то еще. Нет ничего плохого в том, чтобы охранять вашу частную жизнь. Частная жизнь неприкосновенна, как Конституция...*
>
> *Мы движемся к будущему, где мир будет опутан волоконно-оптическими сетями высокой емкости, связывающими наши повсеместно распространенные персональные компьютеры. Электронная почта станет нормой для всех, а не новинкой, как сегодня. Правительство будет защищать наши электронные сообщения государственными протоколами шифрования. Наверное, большинство людей примет это. Но, возможно, некоторые захотят иметь свои собственные защитные меры... Если конфиденциальность признать вне закона, только люди вне закона будут ею обладать.*
>
> *Спецслужбы обладают лучшими криптографическими технологиями. Как и торговцы оружием и наркотиками. Как и военные подрядчики, нефтяные компании и другие корпорации-гиганты. Но обычные люди и общественные организации практически не имеют недорогих защитных криптографических технологий с открытым ключом. До сих пор не имели.*
>
> *PGP дает людям возможность самим защищать свою конфиденциальность. Сегодня существует растущая социальная потребность в этом. Вот почему я написал PGP.*

В заключении следует отметить стойкость данного алгоритма. В 2003 г. Ади Шамир и Эран Тромер разработали схему устройства TWIRL, которое при стоимости $ 10 000 может дешифровать 512-битный ключ за 10 минут, а при стоимости $ 10 000 000 – 1024-битный ключ меньше, чем за год. В настоящее время Лаборатория RSA рекомендует использовать ключи (параметр **n**) размером 2048 битов.

9.3. Алгоритм на основе задачи об укладке ранца

В 1978 г. Меркл и Хеллман предложили использовать задачу об укладке ранца (рюкзака) для асимметричного шифрования [8]. Она относится к классу NP-полных задач и формулируется следующим образом. Дано множество предметов различного веса. Спрашивается, можно ли положить некоторые из этих предметов в ранец так, чтобы его вес стал равен определенному значению? Более формально задача формулируется так: дан набор значений $\mathbf{M_1, M_2, ..., M_n}$ и суммарное значение **S**; требуется вычислить значения $\mathbf{b_i}$ такие что

$$S = b_1M_1 + b_2M_2 + ... + b_nM_n, \qquad (9.3)$$

где n – количество предметов;
b_i - бинарный множитель. Значение $b_i = 1$ означает, что предмет i кладут в рюкзак, $b_i = 0$ - не кладут.

Например, веса предметов имеют значения 1, 5, 6, 11, 14, 20, 32 и 43. При этом можно упаковать рюкзак так, чтобы его вес стал равен 22, использовав предметы весом 5, 6 и 11. Невозможно упаковать рюкзак так, чтобы его вес стал равен 24.

В основе алгоритма, предложенного Мерклом и Хеллманом, лежит идея шифрования сообщения на основе решения серии задач укладки ранца. Предметы из кучи выбираются с помощью блока открытого текста, длина которого (в битах) равна количеству предметов в куче. При этом биты открытого текста соответствуют значениям **b**, а текст является полученным суммарным весом. Пример шифрограммы, полученной с помощью задачи об укладке ранца, показан в следующей таблице.

Таблица 9.4. Пример шифрования на основе задачи об укладке ранца

Открытый текст	1	1	1	0	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0
Рюкзак (ключ)	1	5	6	11	14	20	32	43	1	5	6	11	14	20	32	43	1	5	6	11	14	20	32	43
Шифрограмма	32 (1+5+6+20)								73 (5+11+14+43)								0							

Суть использования данного подхода для шифрования состоит в том, что на самом деле существуют две различные задачи укладки ранца - одна из них решается легко и характеризуется линейным ростом трудоемкости, а другая, как принято считать, нет. Легкий для укладки ранец можно превратить в трудный. Раз так, то можно применить в качестве открытого ключа **трудный** для укладки ранец, который легко использовать для шифрования, но невозможно - для дешифрования. А в качестве закрытого ключа применить **легкий** для укладки ранец, который предоставляет простой способ дешифрования сообщения.

В качестве закрытого ключа (легкого для укладки ранца) используется сверхвозрастающая последовательность. **Сверхвозрастающей** называется **последовательность**, в которой каждый последующий член больше суммы всех предыдущих. Например, последовательность {2, 3, 6, 13, 27, 52, 105, 210} является сверхвозрастающей, а {1, 3, 4, 9, 15, 25, 48, 76} - нет.

Решение для сверхвозрастающего ранца найти легко. В качестве текущего выбирается полный вес, который надо получить, и сравнивается с весом самого тяжелого предмета в ранце. Если текущий вес меньше веса данного предмета, то его в рюкзак не кладут, в противном случае его укладывают в рюкзак. Уменьшают текущий вес на вес положенного предмета и переходят к следующему по весу предмету в последовательности. Шаги повторяются до тех пор, пока процесс не закончится. Если текущий вес уменьшится до нуля, то решение найдено. В противном случае, нет.

Например, пусть полный вес рюкзака равен 270, а последовательность весов предметов равна {2, 3, 6, 13, 27, 52, 105, 210}. Самый большой вес – 210. Он меньше 270, поэтому предмет весом 210 кладут в рюкзак. Вычитают 210 из 270 и получают 60. Следующий наибольший вес последовательности равен 105. Он больше 60, поэтому предмет весом 105 в рюкзак не кладут. Следующий самый тяжелый предмет имеет вес 52. Он меньше 60, поэтому предмет весом 52 также кладут в рюкзак. Аналогично проходят процедуру укладки в рюкзак предметы весом 6 и 2. В результате полный вес уменьшится до 0. Если бы этот рюкзак был бы использован для дешифрования, то открытый текст, полученный из значения шифртекста 270, был бы равен 10100101.

Открытый ключ представляет собой не сверхвозрастающую (нормальную) последовательность. Он формируется на основе закрытого ключа и, как принято считать, не позволяет легко решить задачу об укладке ранца. Для его получения все значения закрытого ключа умножаются на число **n** по модулю **m**. Значение модуля **m** должно быть больше суммы всех чисел последовательности, например, 420 (2+3+6+13+27+52+105+210=418). Множитель **n** должен быть взаимно простым числом с модулем **m**, например, 31. Результат построения нормальной последовательности (открытого ключа) представлен в следующей таблице.

Таблица 9.5. Пример получения открытого ключа

Закрытый ключ, k_i	2	3	6	13	27	52	105	210
Открытый ключ, $(k_i*n) \bmod m = (k_i*31) \bmod 420$	62	93	186	403	417	352	315	210

Для шифрования сообщение сначала разбивается на блоки, по размерам равные числу элементов последовательности в рюкзаке. Затем, считая, что единица указывает на присутствие элемента последовательности в рюкзаке, а ноль — на его отсутствие, вычисляются полные веса рюкзаков – по одному рюкзаку для каждого блока сообщения.

В качестве примера возьмем открытое сообщение «АБРАМОВ», символы которого представим в бинарном виде в соответствии с таблицей кодов символов Windows 1251. Результат шифрования с помощью открытого ключа {62, 93, 186, 403, 417, 352, 315, 210} представлен в следующей таблице.

Таблица 9.6. Пример шифрования

Открытое сообщение		Сумма весов	Шифрограмма (рюкзак), c_i
Символ	Bin-код		

А	1100 0000	62+93	155
Б	1100 0001	62+93+210	365
Р	1101 0000	62+93+403	558
А	1100 0000	62+93	155
М	1100 1100	62+93+417+352	924
О	1100 1110	62+93+417+352+315	1239
В	1100 0010	62+93+315	470

Для расшифрования сообщения получатель должен сначала определить обратное число $\mathbf{n^{-1}}$, такое что $\mathbf{(n * n^{-1}) \bmod m = 1}$. После определения обратного числа каждое значение шифрограммы умножается на $\mathbf{n^{-1}}$ по модулю **m** и с помощью закрытого ключа определяются биты открытого текста.

В нашем примере сверхвозрастающая последовательность равна {2, 3, 6, 13, 27, 52, 105, 210}, m = 420, n = 31. Значение n^{-1} равно 271 (31*271 mod 420 = 1).

Таблица 9.7. Пример расшифрования

Шифрограмма (рюкзак), c_i	$(c_i*n^{-1}) \bmod m = (c_i*271) \bmod 420$	Сумма весов	Открытое сообщение	
			Символ	Bin-код
155	5	2+3	1100 0000	А
365	215	2+3+210	1100 0001	Б
558	18	2+3+13	1101 0000	Р
155	5	2+3	1100 0000	А
924	84	2+3+27+52	1100 1100	М
1239	189	2+3+27+52+105	1100 1110	О
470	110	2+3+105	1100 0010	В

В своей работе авторы рекомендовали брать длину ключа, равную 100 (количество элементов последовательности). В заключении следует отметить, что задача вскрытия данного способа шифрования успешно решена Шамиром и Циппелом в 1982 г.

9.4. Вероятностное шифрование

Вероятностное шифрование является разновидностью криптосистем с открытым ключом (авторы - Шафи Гольдвассер (Shafi Goldwasser) и Сильвио Микали (Silvio Micali)). Данный вид шифрования относят к допускающим **неоднозначное вскрытие**. Основной целью вероятностного шифрования является устранение утечки информации в криптографии с открытым ключом. Поскольку криптоаналитик всегда может зашифровать случайные сообщения открытым ключом, он может получить некоторую информацию. При условии, что у него есть шифртекст **C** ($\mathbf{C = E_{k1}(T)}$) и он пытается получить открытый текст **T**, он может выбрать случайное сообщение **T'** и зашифровать: $\mathbf{C' = E_{k1}(T')}$. Если **C' = C**, то он угадал правильный открытый текст. В противном случае он делает следующую попытку.

Вероятностное шифрование такую попытку атаки шифртекста делает бессмысленной. Другими словами, при шифровании с помощью одного и того же открытого ключа можно получить разные шифртексты, которые при расшифровке дают один и тот же открытый текст.

$$C_1 = E_{k1}(T), C_2 = E_{k1}(T), C_3 = E_{k1}(T), ..., C_N = E_{k1}(T), \qquad (9.4)$$

$$T = D_{k2}(C_1) = D_{k2}(C_2) = D_{k2}(C_3) = ... = D_{k2}(C_N). \qquad (9.5)$$

В результате, даже если у криптоаналитика имеется шифртекст $\mathbf{C_i}$ и он угадает **T**, то в результате операции шифрования получиться $\mathbf{C_j = E_{k1}(T)}$. Вероятность того, что $\mathbf{C_i = C_j}$

крайне низка. Таким образом, криптоаналитик даже не узнает, была ли правильной его догадка относительно **T** или нет.

Ниже рассматриваются два алгоритма вероятностного шифрования:

- алгоритм шифрования Эль-Гамаля;

- алгоритм на основе эллиптических кривых.

9.5. Алгоритм шифрования Эль-Гамаля

Схема была предложена Тахером Эль-Гамалем в 1984 г. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и обеспечения аутентификации. Стойкость данного алгоритма базируется на сложности решения задачи дискретного логарифмирования.

Суть задачи заключается в следующем. Имеется уравнение

$$g^x \bmod p = y. \qquad (9.6)$$

Требуется по известным **g**, **y** и **p** найти целое неотрицательное число **x** (**дискретный логарифм**).

Порядок создания ключей приводится в следующей таблице.

Таблица 9.8. Процедура создания ключей

№ п/п	Описание операции	Пример
1	Выбирается простое число **p**.	p = 37
2	Выбирается число **g**, являющееся первообразным корнем по модулю **p**[*)] и меньшее **p**.	g = 2
3	Выбираются произвольное число **x**, меньшее **p**.	x = 5
4	Вычисляется **y** = $g^x \bmod p$.	$y = 2^5 \bmod 37 = 32 \bmod 37 = 32$
5	Открытый ключ - **y**, **g** и **p**. Причем **g** и **p** можно сделать общими для группы пользователей. Закрытый ключ - **x**.	

*) **Первообразный (примитивный) корень по модулю p** – наименьшее положительное число g такое, что

$$g^{\varphi(p)} \bmod p = 1$$

и

$$g^i \bmod p \neq 1, \quad \text{для } 1 \leq i < \varphi(p)$$

где φ(p) – функция Эйлера (т.к. p – простое число, то φ(p) = p - 1).

Проверка. При g = 2 и p = 37, φ(37) = 37 – 1 = 36.

$2^{36} \bmod 37 = 1$;
$2^1 \bmod 37 = 2 (\neq 1)$;
$2^2 \bmod 37 = 4 (\neq 1)$;
$2^3 \bmod 37 = 8 (\neq 1)$;
$2^4 \bmod 37 = 16 (\neq 1)$;
...;

$2^{34} \bmod 37 = 28 (\neq 1)$;
$2^{35} \bmod 37 = 19 (\neq 1)$.

Для шифрования каждого отдельного блока исходного сообщения должно выбираться случайное число **k** ($1 < k < p - 1$). После чего шифрограмма генерируется по следующим формулам

$$a = g^k \bmod p, \qquad (9.7)$$

$$b = (y^k T) \bmod p, \qquad (9.8)$$

где T – исходное сообщение;
(a, b) – зашифрованное сообщение.

Дешифрование сообщения выполняется по следующей формуле

$$T = (b\ (a^x)^{-1}) \bmod p \qquad (9.9)$$

или

$$T = (b\ a^{p-1-x}) \bmod p, \qquad (9.10)$$

где $(a^x)^{-1}$ – обратное значение числа a^x по модулю p.

Пример шифрования и дешифрования по алгоритму Эль-Гамаля при k = 7 приведен в таблице, хотя для шифрования каждого блока (в нашем случае буквы) исходного сообщения надо использовать свое случайное число **k**.

Первая часть шифрованного сообщения – $a = 2^7 \bmod 37 = 17$.

$a^x = 17^5 = 1419857$, $(a^x)^{-1} = 2$ ($1419857 * 2 \bmod 37 = 1$) или $a^{p-1-x} = 17^{37-1-5} \approx 1.3928892 * 10^{38}$.

Таблица 9.9. Пример шифрования по алгоритму Эль-Гамаля (при k = const)

Открытое сообщение, T	Символ	А	Б	Р	А	М	О	В
	Код	1	2	18	1	14	16	3
Шифрование	Случайное число k	7	7	7	7	7	7	7
	Первая часть шифрограммы, $a = 2^k \bmod 37$	17	17	17	17	17	17	17
	Вторая часть шифрограммы, $b = (32^k * T) \bmod 37$	19	1	9	19	7	8	20
Расшифрование	Обратное значение числа a^x по модулю p, $(a^x)^{-1}$	2	2	2	2	2	2	2
	Открытое сообщение, $T = (b * (a^x)^{-1}) \bmod 37$	1	2	18	1	14	16	3

Ввиду того, что число **k** является произвольным, то такую схему еще называют схемой **вероятностного шифрования**. Вероятностный характер шифрования является преимуществом для схемы Эль-Гамаля, т.к. у схем вероятностного шифрования наблюдается большая стойкость по сравнению со схемами с определенным процессом шифрования. Недостатком схемы шифрования Эль-Гамаля является удвоение длины зашифрованного текста по сравнению с начальным текстом. Для схемы вероятностного шифрования само сообщение **T** и ключ не определяют шифртекст однозначно. В схеме Эль-Гамаля необходимо использовать различные значения случайной величины **k** для шифровки различных сообщений **T** и **T'**. Если использовать одинаковые **k**, то для соответствующих шифртекстов (**a**, **b**) и (**a'**, **b'**) выполняется соотношение $\mathbf{b\ (b')^{-1} = T\ (T')^{-1}\ (mod\ p)}$. Из этого выражения можно легко вычислить **T**, если известно **T'**.

Пример. Предположим злоумышленник перехватил зашифрованное сообщение $\mathbf{C = ((a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n))}$, для которого использовалось одно и тоже случайное число **k**. Он знает один из блоков $\mathbf{T_i = E_{k1}(C_i)}$ или при известном открытом ключе (**y**, **g**, **p**) ему удалось подобрать **k'**, которое совпало с используемым при шифровании **k**. Например по второму варианту, для шифрования символа **X** (T' = 22) злоумышленник использовал k' = 7 (равное k). Тогда, b(X) = b' = ($32^7 * 22$) mod 37 = 11, $(b')^{-1} = 27$, $(T')^{-1} = 32$. Расшифрование перехваченного сообщения приведено в следующей таблице.

Таблица 9.10. Пример расшифрования перехваченного сообщения

Вторая часть перехваченной шифрограммы, b		19	1	9	19	7	8	20
$L = (b * (b')^{-1}) \bmod p = (b * 27) \bmod 37$		32	27	21	32	4	31	22
Вскрытое открытое сообщение, T	Код, определяемый по уравнению $(T * (T')^{-1}) \bmod p = L$ $(T * 32) \bmod 37 = L$	1	2	18	1	14	16	3
	Символ	А	Б	Р	А	М	О	В

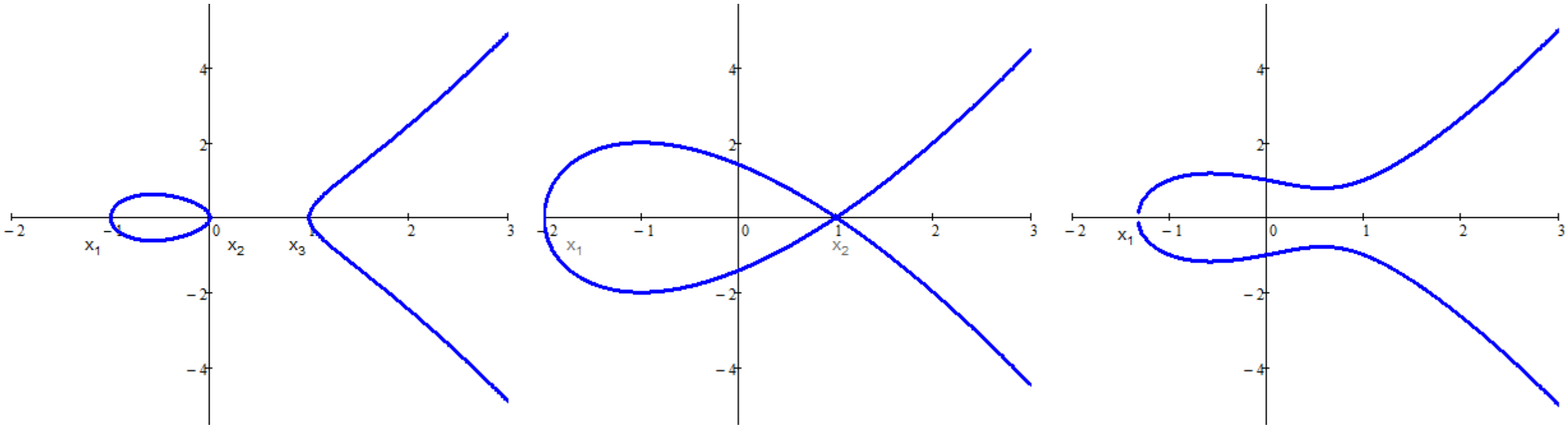
9.6. Алгоритм на основе эллиптических кривых

Использование эллиптических кривых для создания криптосистем было независимо предложено Нилом Коблицем (Neal Koblitz) и Виктором Миллером (Victor Miller) в 1985 г. [8, 17]. При использовании алгоритмов на эллиптических кривых полагается, что не существует быстрых алгоритмов для решения задачи дискретного логарифмирования в группах их точек. В настоящий момент известны лишь экспоненциальные алгоритмы вычисления обратных функций для эллиптических кривых. По сравнению с субэкспоненциальными алгоритмами разложения числа на простые сомножители (см. криптосистему RSA), это позволяет при одинаковом уровне стойкости уменьшить размерность ключа в несколько раз, а, следовательно, упростить программную и аппаратную реализацию криптосистем.

Эллиптической кривой **E** называется множество точек (**x**, **y**), удовлетворяющих однородному уравнению Вейерштрасса:

$$y^2 + a_1xy + a_2y = x^3 + a_3x^2 + a_4x + a_5, \qquad (9.11)$$

где a_i - коэффициенты уравнения.



а) $y^2 = x^3 - x$ б) $y^2 = x^3 - 3x + 2$ в) $y^2 = x^3 - x + 1$

Рис.9.1. Примеры эллиптических кривых

В криптографии эллиптические кривые рассматриваются над двумя типами конечных полей: простыми полями нечётной характеристики ($\mathbb{Z}_n$, где **n** > 3 - простое число) и полями характеристики 2 (**GF(2^m)**).

Эллиптические кривые над полями нечётной характеристики $\mathbb{Z}_n$ можно привести к виду, называемому эллиптической кривой в короткой форме Вейерштрасса:

$$y^2 = x^3 + Ax + B \pmod n, \qquad (9.12)$$

где A, B $\in \mathbb{Z}_n$ - коэффициенты эллиптической кривой, удовлетворяющие $4A^3 + 27B^2 \neq 0 \pmod n$.

Поскольку $y = \pm\sqrt{x^3 + Ax + B}$, график кривой симметричен относительно оси абсцисс. Чтобы найти точки его пересечения с осью абсцисс, необходимо решить кубическое уравнение

$$x^3 + Ax + B = 0. \qquad (9.13)$$

Это можно сделать с помощью известных формул Кардано. Дискриминант этого уравнения

$$\Delta = \left(\frac{A}{3}\right)^3 + \left(\frac{B}{2}\right)^2 = \frac{4A^3 + 27B^2}{108}. \qquad (9.14)$$

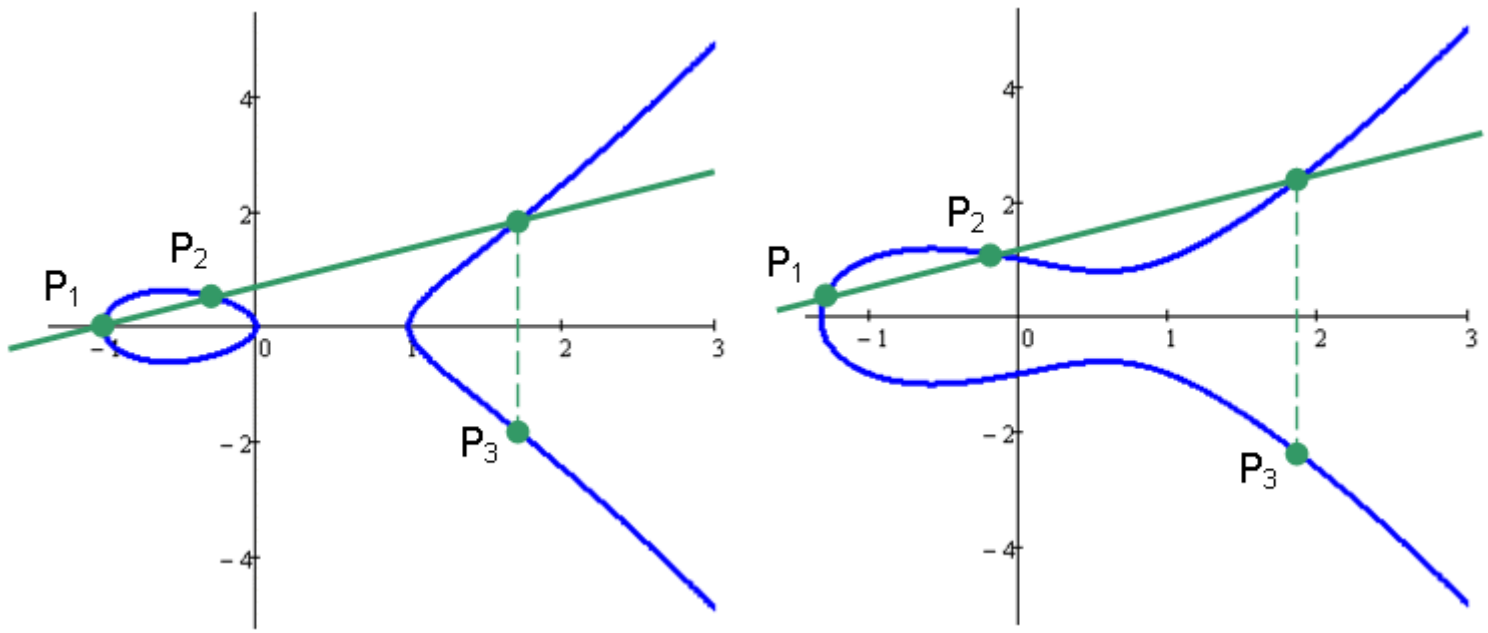
Если $\Delta > 0$, то уравнение имеет три различных действительных корня (см. рис. 9.1а – x_1, x_2 и x_3). Если $\Delta = 0$, то уравнение имеет три действительных корня, по крайней мере, два из которых равны (см. рис. 9.1б – x_1 и x_2). Если $\Delta < 0$, то уравнение имеет один действительный корень (см. рис. 9.1в – x_1) и два комплексно сопряженных.

Используемые в криптографии кривые не должны иметь особых точек. Геометрически это значит, что график не должен иметь точек возврата и самопересечений (см. рис. 9.1б). Если кривая не имеет особых точек, то её график имеет две части при положительном дискриминанте (см. рис. 9.1а), и одну - при отрицательном (см. рис. 9.1в). Например, для графиков выше в первом случае дискриминант равен 64, а в третьем он равен -368.

Следует отметить, что в $\mathbb{Z}_n$ у каждого ненулевого элемента есть либо два квадратных корня, либо нет ни одного, поэтому точки эллиптической кривой разбиваются на пары вида P = (x, y) и -P = (x, -y). Например, эллиптическая кривая $y^2 = x^3 + 3x + 2$ при x = 1 и n = 5 имеет две точки в качестве решения: P = (1, 1) и -P = (1, -1), т.к. $1^2 \equiv 1^3 + 3 * 1 + 2 \pmod 5$ и $(-1)^2 \equiv 1^3 + 3 * 1 + 2 \pmod 5$.

Введем две операции, которые можно выполнять над точками кривой.

Сложение точек $P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2)$.



а) $y^2 = x^3 - x$ б) $y^2 = x^3 - x + 1$

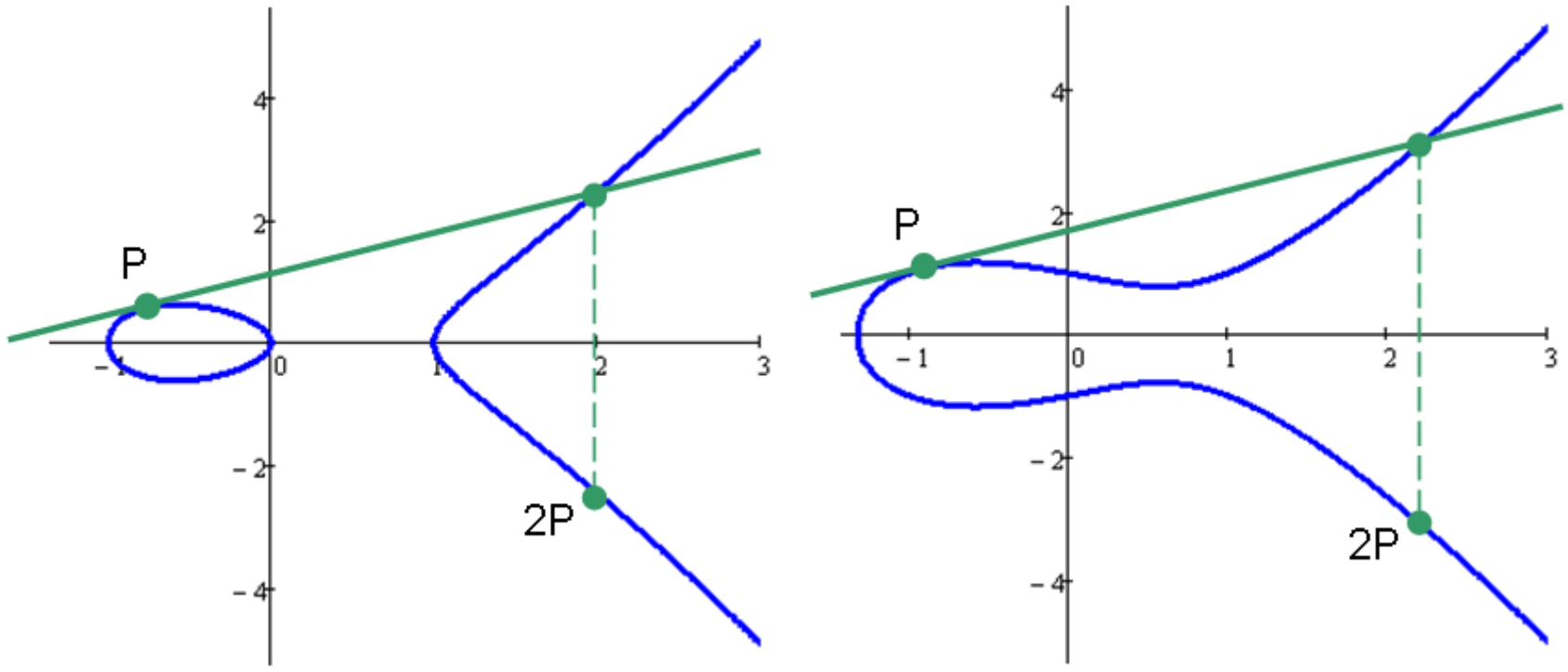
Рис.9.2. Сложение точек

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & x_1 \neq x_2 \\ \dfrac{3x_1^2 + A}{2y_1}, & x_1 = x_2 \end{cases} \qquad (9.15)$$

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}. \qquad (9.16)$$

В формулах 9.15 и 9.16 λ - угловой коэффициент прямой, проходящей через точки P_1 и P_2. Если $P_1 = P_2$, то λ равен значению производной в точке P_1.

Умножение точки на число $P_k(x_k, y_k) = k * P(x, y)$.



а) $y^2 = x^3 - x$ б) $y^2 = x^3 - x + 1$

Рис.9.3. Удвоение точки

А) вариант 1 – выполнить сложение точки **P k** раз

$$P_k(x_k, y_k) = k \cdot P(x,y) = \underbrace{P + P + P + \ldots + P}_{k}. \qquad (9.17)$$

В) вариант 2 – с использование двоичного представления числа $\mathbf{k = (b_L, \ldots, b_2, b_1)}$ и операции удвоения точки. Например, $k = 110_{10} = 1101110_2$, тогда $P_k = 64P + 32P + 8P + 4P + 2P$.

Алгоритм, вычисления $\mathbf{P_k}$ может выглядеть следующим образом.

```
1. Pk := null
2. Q := P
3. Цикл. Для i := 1 до L
      3.1. Если b  = 1
```

```
                i
          то Если P_k = null
             то P_k = Q
             иначе P_k = P_k + Q
    3.2. Q = 2 * Q     // ≈ Q = Q + Q
```

Для k = 110 вместо 109 сложений будет 1 присваивание, 4 сложения и 6 удвоений (сложений).

Рассмотрим процедуру создания ключей.

Таблица 9.11. Процедура создания ключей

№ п/п	Описание операции	Пример
1	Выбирается модуль эллиптической кривой - простое число **n** ($n > 2^{255}$ - ГОСТ).	n = 41
2	Выбираются коэффициенты эллиптической кривой **A** и **B**. Должно соблюдаться условие $(4 A^3 + 27 B^2) \bmod n \neq 0$, в противном случае меняются параметры эллиптической кривой n, A или B.	A = 3, B = 7 $(4 * 3^3 + 27 * 7^2) \bmod 41 = 37$
3	Определяется точка эллиптической кривой $\mathbf{P(x_p, y_p)}$ и порядок циклической подгруппы группы точек эллиптической кривой **q***). Принимается произвольное x_p ($0 < x_p < n$) и определяется y_p из уравнения эллиптической кривой. Должны соблюдаться условия: - для x_p должен существовать y_p (не для всякого x_p при данных параметрах кривой (A, B, n) может существовать y_p); - $P \neq O$ и $q P = O$, где O - нулевая точка эллиптической кривой ($P + (-P) = O$); - q – простое число; - $n^t \bmod q \neq 1$, для всех целых $t = 1, 2, \ldots, T$, где $T \leq 31$; - $2^{254} < q < 2^{256}$ (ГОСТ). В противном случае меняются либо параметры эллиптической кривой n, A или B либо выбирается другая точка P.	$x_p = 7$ ($(7^3+3*7+7) \bmod 41 = 2$), $y_p = 17$ ($17^2 \bmod 41 = 2$) q = 47
4	Выбирается закрытый ключ **d** ($0 < d < q$).	d = 10
5	Определяется точка эллиптической кривой $\mathbf{Q(x_q, y_q)}$: $Q(x_q, y_q) = d * P(x_p, y_p)$.	$x_q = 36$, $y_q = 20$
6	Публикуется открытый ключ $\mathbf{[(A, B), P(x_p, y_p), n, Q(x_q, y_q)]}$ в специальном хранилище, где исключается возможность его подмены (**общедоступном сертифицированном справочнике**). Для выработки и проверки электронной цифровой подписи q является частью открытого ключа вместо n.	

*) Прямой способ вычисления порядка группы точек эллиптической кривой **q**.

1) Рассчитываются координаты первой точки из выражения

$$y^2 = x^3 + Ax + B \pmod n > y^2 = x^3 + 3x + 7 \pmod{41}.$$

Примем $x_1 = 7$, тогда ${y_1}^2 \bmod 41 = (7^3 + 3 * 7 + 7) \bmod 41 = 2$, откуда $y_1 = 17$.

$$P(x_p, y_p) = P(x_1, y_1) = P(7, 17).$$

2) Находим координаты второй точки. Для этого вначале вычисляется коэффициент λ

$$\lambda = \frac{(3x_1^2 + a)}{2y_1} (\text{mod } n) = \frac{3 * 7^2 + 3}{2 * 17} (\text{mod } 41) = \frac{150}{34} (\text{mod } 41) \Rightarrow$$

$$34\lambda \text{ mod } 41 = 150 \text{ mod } 41 \Rightarrow$$

$$34\lambda \text{ mod } 41 = 27$$

Решая последнее уравнение, получаем $\lambda = 2$ ($34 * 2 \text{ mod } 41 = 27$).

Координаты второй точки получаем путем удваивания первой из выражений

$x_2 = (\lambda^2 - 2x_1) \text{ mod } n = (2^2 - 2 * 7) \text{ mod } 41 = -10 \text{ mod } 41 = 31,$

$y_2 = (\lambda(x_1 - x_2) - y_1) \text{ mod } n = (2\ (7 - 31) - 17) \text{ mod } 41 = -65 \text{ mod } 41 = 17.$

3) Каждую следующую точку рассчитываем по формулам, пока в знаменателе первой формулы не будет получен 0:

$$\lambda_i = \frac{y_{i-1} - y_1}{x_{i-1} - x_1} (\text{mod } n), \qquad (9.18)$$

$$\begin{cases} x_i = (\lambda_i^2 - x_1 - x_{i-1}) \text{ mod } n \\ y_i = (\lambda_i(x_1 - x_i) - y_1) \text{ mod } n \end{cases}. \qquad (9.19)$$

Получаем:

- $\lambda_3 = 0$, $x_3 = 3$, $y_3 = 24$;
- $\lambda_4 = 29$, $x_4 = 11$, $y_4 = 31$;
- $\lambda_5 = 24$, $x_5 = 25$, $y_5 = 2$;
- …;
- $\lambda_{46} = 2$, $x_{46} = 7$, $y_{46} = 24$.

К полученному числу точек добавляем точку **O**, в результате чего $q = 46 + 1 = 47$. Эта точка есть результат сложения любых двух точек $P(x_i, y_i)$ и $-P(x_i, -y_i)$ и представляет собой бесконечно удаленную точку, в которой гипотетически сходятся все вертикальные кривые.

Процедура шифрования отдельного блока выполняется следующим образом [25].

Таблица 9.12. Процедура шифрования отдельного блока (буквы)

№ п/п	Описание операции	Пример
1	Определяется десятичное представление буквы **t**.	Буква «К» t = 12
2	Выбирается случайное число **k** ($0 < k < n$).	k = 5
3	Определяется точка $\mathbf{P_k(x_{pk}, y_{pk})} = k * P$.	$P_k(25, 2)$
4	Определяется точка $\mathbf{Q_k(x_{qk}, y_{qk})} = k * Q$.	$Q_k(3, 24)$
5	Вычисляется $\mathbf{c} = (t * x_{qk}) \text{ mod } n$.	$c = (12 * 3) \text{ mod } 41 = 36$
6	Шифрограмма – пара $[P_k, c]$.	$[P_k(25, 2), 36]$

Процедура расшифрования отдельного блока выполняется следующим образом.

Таблица 9.13. Процедура расшифрования отдельного блока (буквы)

№ п/п	Описание операции	Пример

1	Вычисляется точка $\mathbf{D(x_d, y_d)} = d * P_k$.	D(3, 24)
2	Вычисляется десятичное представление зашифрованной буквы $\mathbf{t} = (c * x_d^{-1}) \bmod n$, где x_d^{-1} – обратное число к x_d по модулю n.	$x_d^{-1} = 14$ [(3 * 14) mod 41 = 1] t = (36 * 14) mod 41 = 12
3	Определяется исходное сообщение по ее десятичному представлению.	Буква «К»

Приведенный выше способ шифрования является вариацией шифрования Эль-Гамаля. Если стойкость алгоритма шифрования Эль-Гамаля базируется на сложности решения задачи дискретного логарифмирования, то стойкость шифрования с помощью эллиптических кривых базируется на сложности нахождения множителя **k** точки **P** по их произведению. Т.е. если **Q = k P**, то зная **P** и **k** довольно легко вычислить **Q**. Эффективное решение обратной задачи (найти **k** при известных **P** и **Q**) на текущий момент пока не опубликовано.

Вопросы для самопроверки

1. В чем заключается суть и основная предпосылка появления шифрования с открытым ключом?
2. Основные требования, предъявляемые к криптосистемам с открытым ключом.
3. Перечислите типы односторонних преобразований, применяемых при асимметричном шифровании.
4. Дайте краткую характеристику алгоритма RSA.
5. В чем отличие сверхвозрастающей последовательности от обыкновенной?
6. Что означает обратное число по модулю?
7. В чем отличие вероятностного шифрования с открытым ключом от детерминированного?
8. В чем суть задачи дискретного логарифмирования?
9. Приведите уравнение эллиптической кривой в короткой форме Вейерштрасса.

10. ХЕШ-ФУНКЦИИ

10.1. Основные понятия

Хеширование (иногда хэширование, англ. hashing) - преобразование входного массива данных произвольной длины в выходную строку фиксированной длины. Такие преобразования также называются **хеш-функциями** или **функциями свёртки**, входной массив – **прообразом**, а результаты преобразования - **хешем, хеш-кодом, хеш-образом, цифровым отпечатком** или **дайджестом сообщения** (англ. message digest).

Хеш-функция – легко вычислимая функция, преобразующая исходное сообщения произвольной длины (прообраз) в сообщение фиксированное длины (хеш-образ), для которой не существует эффективного алгоритма поиска коллизий.

Коллизией для функции **h** называется пара значений **x, y, x ≠ y**, такая, что **h(x) = h(y)**. Т.о. хеш-функция должна обладать следующими свойствами:

- для данного значения **h(x)** невозможно найти значение аргумента **x**. Такие хеш-функции называют **стойкими в смысле обращения** или **стойкими в сильном смысле**;

- для данного аргумента **x** невозможно найти другой аргумент **y** такой, что **h(x) = h(y)**. Такие хеш-функции называют **стойкими в смысле вычисления коллизий** или **стойкими в слабом смысле**.

В случае, когда значение хеш-функции зависит не только от прообраза, но и закрытого ключа, то это значение называют **кодом проверки подлинности сообщений (Message Authentication Code, MAC), кодом проверки подлинности данных (Data Authentication Code, DAC)** или **имитовставкой**.

На практике хеш-функции используют в следующих целях:

- для ускорения поиска данных в БД;

- для проверки целостности и подлинности сообщений;

- для создания сжатого образа, применяемого в процедурах ЭЦП;

- для защиты пароля в процедурах аутентификации.

Ускорения поиска данных. Например, при записи текстовых полей в базе данных может рассчитываться их хеш-код и данные могут помещаться в раздел, соответствующий этому хеш-коду. Тогда при поиске данных надо будет сначала вычислить хеш-код текста и сразу станет известно, в каком разделе их надо искать, т.е. искать надо будет не по всей базе, а только по одному её разделу (это сильно ускоряет поиск).

Бытовым аналогом хеширования в данном случае может служить размещение слов в словаре по алфавиту. Первая буква слова является его хеш-кодом, и при поиске мы просматриваем не весь словарь, а только раздел с нужной буквой.

Процедура вычисления (стандартная схема алгоритма) хеш-функции представлена на следующем рисунке.

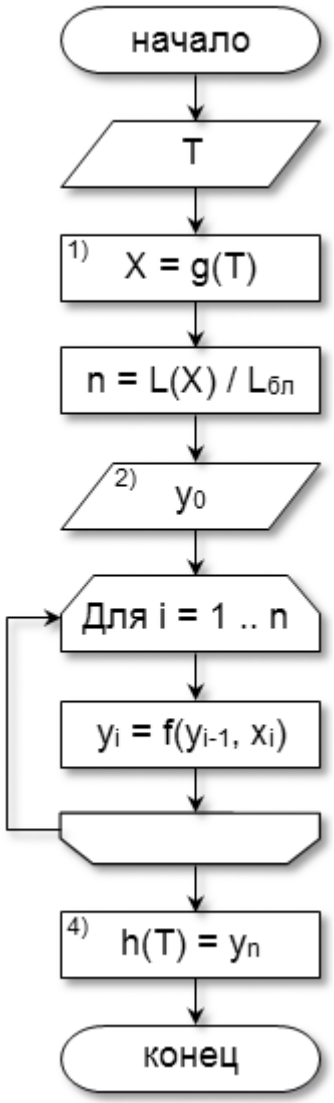


Рис.10.1. Процедура вычисления значения хеш-функции

1) К исходному сообщению **T** добавляется вспомогательная информация (например, длина прообраза, вспомогательные символы и т.д.) так, чтобы длина прообраза **X** стала кратной величине $\mathbf{L_{бл}}$, определенной спецификацией (стандартом) хеш-функции.

2) Для инициализации процедуры хеширования используется синхропосылка $\mathbf{y_0}$.

3) Прообраз **X** разбивается на **n** блоков $\mathbf{x_i}$ (i = 1 .. n) фиксированной длины $\mathbf{L_{бл}}$, над которыми выполняется однотипная процедура хеширования $\mathbf{f(y_{i-1}, x_i)}$, зависящая от результата хеширования предыдущего блока $\mathbf{y_{i-1}}$.

4) Хеш-образом **h(T)** исходного сообщения **T** будет результат процедуры хеширования $\mathbf{y_n}$, полученный после обработки последнего блока $\mathbf{x_n}$.

10.2. MD5

MD5 (англ. Message Digest 5) – 128-битный алгоритм хеширования, разработанный профессором Рональдом Л. Ривестом из Массачусетского технологического института (Massachusetts Institute of Technology, MIT) в 1991 г. Является улучшенной в плане безопасности версией MD4 [17].

Ниже приведен алгоритм вычисления хеша.

1. Выравнивание потока.

В конец исходного сообщения, длиной **L**, дописывают единичный бит, затем необходимое число нулевых бит так, чтобы новый размер **L'** был сравним с 448 по модулю 512 (L' mod 512 = 448). Добавление нулевых бит выполняется, даже если новая длина, включая единичный бит, уже сравнима с 448.

2. Добавление длины сообщения.

К модифицированному сообщению дописывают 64-битное представление длины данных (количество бит в сообщении). Т.е. длина сообщения **T** становится кратной 512 (T mod 512 = 0). Если длина исходного сообщения превосходит 2^{64} - 1, то дописывают только младшие 64 бита. Кроме этого, для указанного 64-битного представления длины вначале записываются младшие 32 бита, а затем старшие 32 бита.

3. Инициализация буфера.

Для вычислений инициализируются 4 переменных размером по 32 бита и задаются начальные значения (шестнадцатеричное представление):

A = 67 45 23 01;
B = EF CD AB 89;
C = 98 BA DC FE;
D = 10 32 54 76.

В этих переменных будут храниться результаты промежуточных вычислений. Начальное состояние **ABCD** называется инициализирующим вектором.

4. Вычисление хеша в цикле.

Исходное сообщение разбивается на блоки **T**, длиной 512 бит. Для каждого блока в цикле выполняется процедура, приведенная на рис.10.2. Результат обработки всех блоков исходного сообщения в виде объединения 32-битных значений переменных **ABCD** и будет являться хешем.

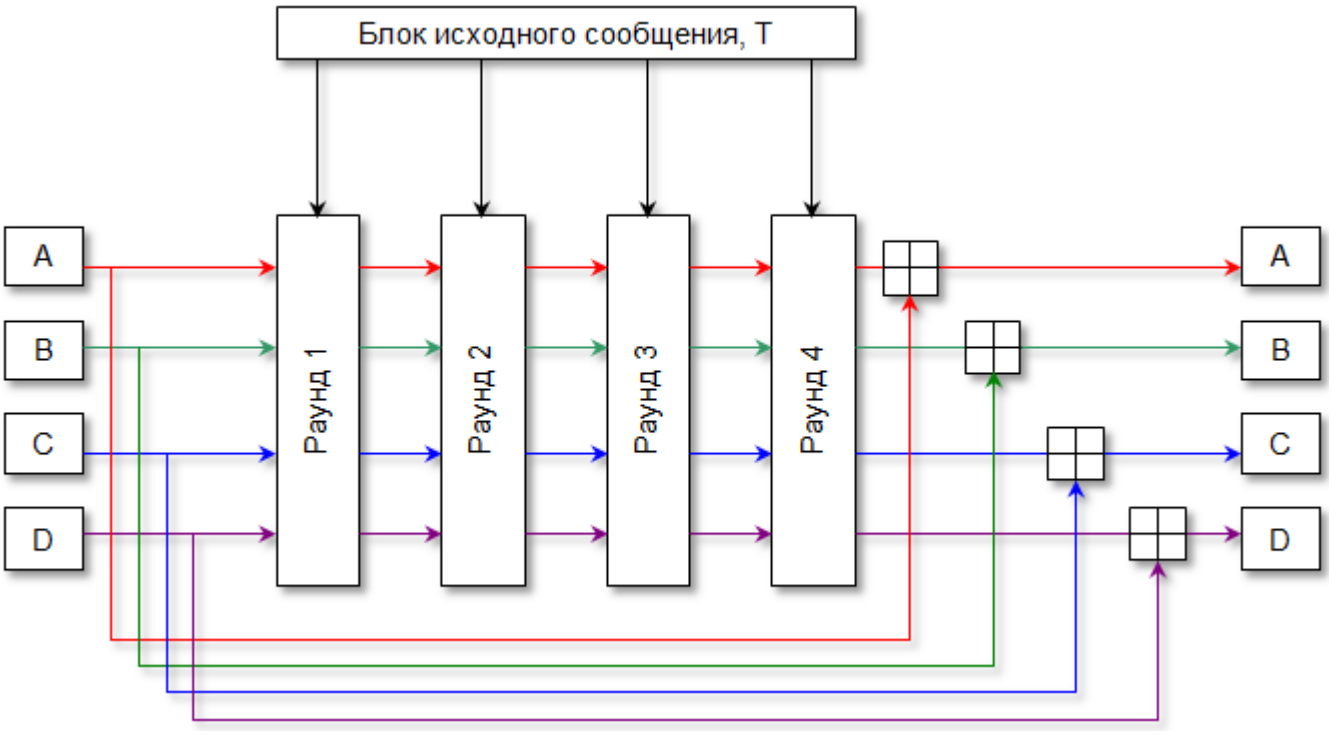


Рис.10.2. Шаг основного цикла вычисления хеша

В каждом раунде над переменными **ABCD** и блоком исходного текста **T** в цикле (16 итераций)

выполняются однотипные преобразования по следующей схеме.

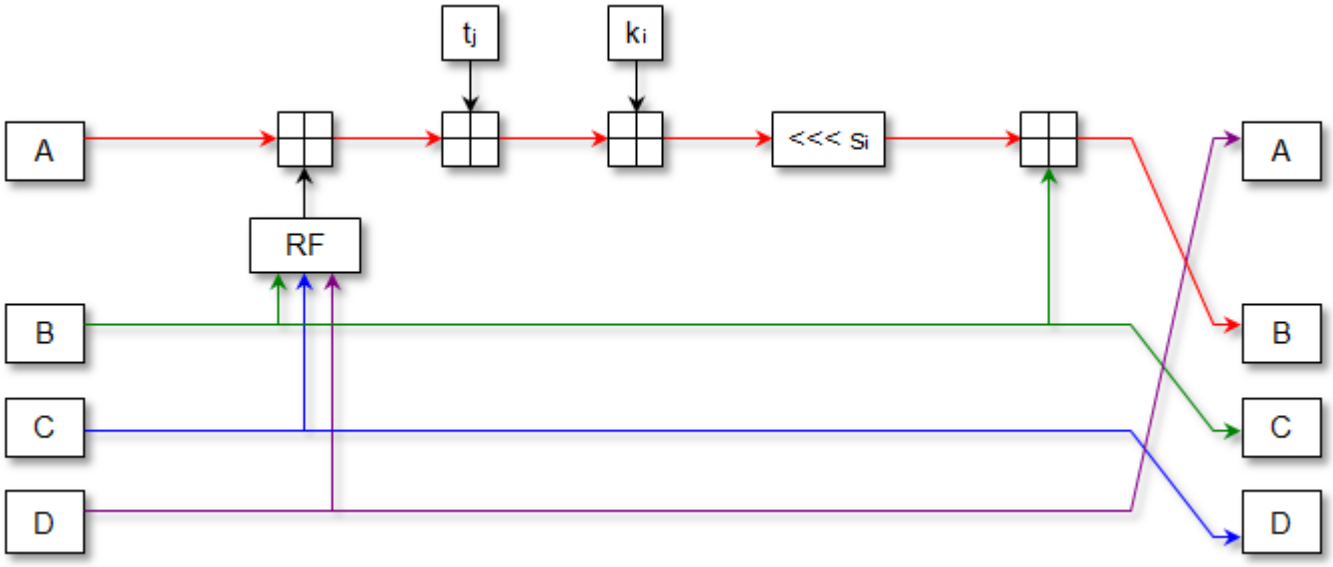


Рис.10.3. Одна итерация цикла раунда

Условные обозначения.

1) **RF** - раундовая функция, определяемая по следующей таблице.

Таблица 10.1. Раундовые функции RF

№ раунда	Обозначение функции	Формула расчета
1	F	F(B, C, D) = (B ∧ C) ∨ (¬B ∧ D)
2	G	G(B, C, D) = (B ∧ D) ∨ (¬D ∧ C)
3	H	H(B, C, D) = B ⊕ C ⊕ D
4	I	I(B, C, D) = C ⊕ (¬D ∨ B)

2) $\mathbf{t_j}$ - j-ая 32-битовая часть блока исходного сообщения **T** с обратным порядком следования байт;

3) $\mathbf{k_i}$ - целая часть константы, определяемой по формуле

$$k_i = 2^{32} * | \sin(i + 16 * (r - 1)) |, \qquad (10.1)$$

где i – номер итерации цикла (i = 1..16);
r – номер раунда (r = 1..4).

Аргумент функции sin измеряется в радианах.

4) ⊞ – сложение по модулю 2^{32}.

5) <<< $\mathbf{s_i}$ – циклический сдвиг влево на s_i разрядов.

Используемая 32-битовая часть блока исходного сообщения $\mathbf{t_j}$ и величина циклического сдвига влево $\mathbf{s_i}$ зависят от номера итерации и приведены в следующей таблице.

Таблица 10.2. Величины, используемые на шаге цикла раунда

№ итерации		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Раунд 1	t_j	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}
	s_i	7	12	17	22	7	12	17	22	7	12	17	22	7	12	17	22
Раунд 2	t_j	t_2	t_7	t_{12}	t_1	t_6	t_{11}	t_{16}	t_5	t_{10}	t_{15}	t_4	t_9	t_{14}	t_3	t_8	t_{13}
	s_i	5	9	14	20	5	9	14	20	5	9	14	20	5	9	14	20
Раунд 3	t_j	t_6	t_9	t_{12}	t_{15}	t_2	t_5	t_8	t_{11}	t_{14}	t_1	t_4	t_7	t_{10}	t_{13}	t_{16}	t_3
	s_i	4	11	16	23	4	11	16	23	4	11	16	23	4	11	16	23
Раунд 4	t_j	t_1	t_8	t_{15}	t_6	t_{13}	t_4	t_{11}	t_2	t_9	t_{16}	t_7	t_{14}	t_5	t_{12}	t_3	t_{10}
	s_i	6	10	15	21	6	10	15	21	6	10	15	21	6	10	15	21

После 4 раундов новое (модифицированное) значение каждой из переменных **ABCD** складывается (⊞) с исходным (значением переменной до 1-го раунда).

5. Перестановка байт в переменных ABCD. После обработки всех блоков исходного сообщения для каждой переменной выполняется обратная перестановка байт.

Поиск коллизий.

В 2004 г. китайские исследователи Ван Сяоюнь (Wang Xiaoyun), Фен Дэнгуо (Feng Dengguo), Лай Сюэцзя (Lai Xuejia) и Юй Хунбо (Yu Hongbo) объявили об обнаруженной ими уязвимости в алгоритме, позволяющей за небольшое время (1 час на кластере IBM p690) находить коллизии.

10.3. Применение шифрования для получения хеш-образа

Для выработки устойчивого к коллизиям хеш-образа могут применяться специальные режимы, предусмотренные в блочных шифрах (например, сцепление блоков шифра у DES), или в самой хеш-функции, как составная часть, может использоваться один из режимов блочного шифра (например, составной часть хеш-функции по ГОСТ 34.11-94[1] является режим простой замены алгоритма криптографического преобразования по ГОСТ 28147-89[2]).

Напомним что в случае, когда значение хеш-функции зависит не только от прообраза, но и закрытого ключа, то хеш-образ называют **кодом проверки подлинности сообщений (Message Authentication Code, MAC), кодом проверки подлинности данных (Data Authentication Code, DAC)** или **имитовставкой**.

В качестве примера приведем режим DES-CBC (сцепление блоков шифра - Cipher Block Chaining).

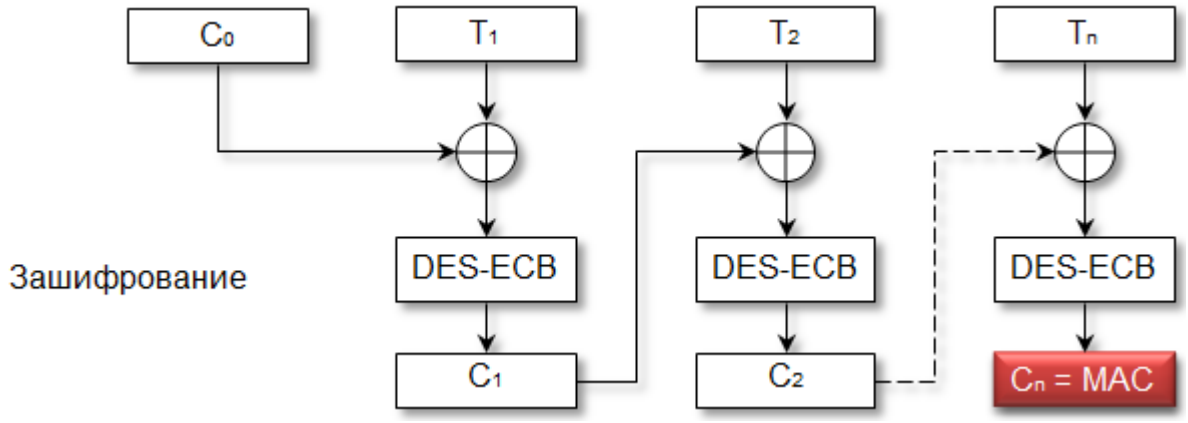


Рис.10.4. Схема алгоритма DES в режиме сцепления блоков шифра

Последний зашифрованный блок $\mathbf{C_n}$ и есть хеш-образ сообщения $\mathbf{T = \{T_1, T_2, \ldots, T_n\}}$.

---

[1]ГОСТ 34.11-94 «Информационная технология. Криптографическая защита информации. Функция хэширования».

[2]ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования».

Вопросы для самопроверки

1. Дайте определение понятиям: «хеширование», «хеш-функция», «коллизия».
2. В каких целях используют хеш-функции на практике?
3. Приведите стандартную схему алгоритма генерации хеш-образа.
4. Как называется хеш-образ, полученный с применением закрытого ключа шифрования?

СПИСОК ЛИТЕРАТУРЫ

1. Конституция Российской Федерации (от 12.12.1993 г.).

2. Федеральный закон Российской Федерации «Об информации, информационных технологиях и о защите информации» (№ 149-ФЗ от 27.07.2006 г.).

3. Доктрина информационной безопасности Российской Федерации (№ Пр-1895 от 06.09.2000 г.).

4. Федеральный закон Российской Федерации «О государственной тайне» (№ 5485-1 от 06.10.1997 г.).

5. Указ Президента РФ «Перечень сведений конфиденциального характера» (№ 188 06.03.1997 г.).

6. Федеральный закон Российской Федерации «О персональных данных» (№ 152 от 27.07.2006 г.).

7. Федеральный закон Российской Федерации «Об электронной цифровой подписи» (№ 1-ФЗ от 26.12.2001 г.).

8. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер – М.: ТРИУМФ, 2002. – 816 с.

9. Введение в криптографию / Под. ред. В.В. Ященко. – СПб.: Питер, 2001. – 288 с.

10. Коутинхо, С. Введение в теорию чисел. Алгоритм RSA / С. Коутинхо. – М.: Постмаркет, 2001. – 328 с.

11. Барычев, С. Основы современной криптографии / С. Барычев, Р. Серов. – М.: 2001. – 152 с.

12. Яковлев, В.В. Информационная безопасность и защита информации в корпоративных сетях железнодорожного транспорта: Учебник для вузов ж.-д. транспорта / В.В. Яковлев, А.А. Корниенко. – М.: УМК МПС России, 2002. – 328 с.

13. Кан, Д. Взломщики кодов / Д. Кан – М.: Центрполиграф, 2000. – 594 с.

14. Бабаш, А.В. Криптография. Под. ред. В.П. Шерстюка, Э.А. Применко / А.В. Бабаш, Г.П. Шанкин – М.: СОЛОН-ПРЕСС, 2007. – 512 с.

15. Алферов, А.П. Основы криптографии / А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин, А.В. Черемушкин – М.: Гелиос АРВ, 2002. – 480 с.

16. National Institute of Standards and Technology (NIST). FIPS Pub 46-3 (Federal information processing standards publication): Data Encryption Standard (DES). Oct. 1999. csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

17. Википедия. ru.wikipedia.org.

18. Галатенко, В.А. Основы информационной безопасности. www.intuit.ru.

19. Галатенко, В.А. Информационная безопасность: основные стандарты и спецификации. www.intuit.ru.

20. Басалова, Г.В. Основы криптографии. www.intuit.ru.

21. Ботт, Э. Эффективная работа: Безопасность Windows / Э. Ботт, К. Зихерт. – СПб.: Питер, 2003. – 682 с.

22. Малюк, А.А. Введение в защиту информации в автоматизированных системах / А.А. Малюк, С.В.

Пазинин, Н.С. Погожин. – М.: Горячая линия - Телеком, 2001. – 148 с.

23. Фомичев, В.М. Дискретная математика и криптология / В.М. Фомичев. – М.: ДИАЛОГ-МИФИ, 2003. – 400 с.

24. Винокуров, А.Ю. Традиционные криптографические алгоритмы. www.enlight.ru/crypto/algorithms/algs.htm.

25. CryptoBloG. Прикладные методы защиты информации. crypto-blog.ru.

26. Скудис, Э. Противостояние хакерам / Эд Скудис. – М.: ДМК Пресс, 2003. – 512 с.

27. Андреев, А.Г. и др. Microsoft Windows 2000 Professional. Русская версия / Под. ред. А.Н. Чекмарева и Д.Б. Вишнякова - СПб.: БВХ - Санкт-Петербург, 2000. – 752 с.

28. Барсуков, В.С. Безопасность: технологии, средства, услуги. / В.С. Барсуков - М.: КУДИЦ - ОБРАЗ, 2001. – 496 с.

29. Гордиенко, И. ID-cards: о старый, новый дивный мир! Компьютерра - 1999. - №10 (288).

30. Берд, К. Война — это мир, паспорт — это свобода. www.computerra.ru.

31. Кохно, В.О. Информационные системы медицинского страхования на основе пластиковых карт

32. Макаров, О. Отдай голос машине: голоса из урны. Популярная механика – 2012. – № 113

33. ЦИК РФ. cikrf.ru.

34. Кнут, Д.Э. Искусство программирования, т.2. Получисленные алгоритмы / Д.Э. Кнут - М.: Издательский дом «Вильямс», 2000. – 832 с.

35. Петров, А.А. Компьютерная безопасность. Криптографические методы защиты / А.А. Петров - М.: ДМК, 2000. – 448 с.

36. Портал ОАО «Универсальная электронная карта». www.uecard.ru.

37. Palaciosa, M.A. and etc. InfoBiology by printed arrays of microorganism colonies for timed and on-demand release of messages. www.pnas.org/content/early/2011/09/19/1109554108.

38. Шелков, В.А. История "микроточки". Специальная техника и связь – 2012. – № 3.

39. Получение микроизображения (микроточки). www.photohistory.ru/Mikrot-manuf.html.

40. Малюк, А.А. Введение в защиту информации в автоматизированных системах / А.А. Малюк, С. В. Пазизин, Н. С. Погожин - М.: Горячая линия-Телеком, 2001. – 148 с.

41. Риджуэй, А. Целую книгу закодировали в ДНК. Наука в фокусе - 2012 - №11 (013).

42. www.technologyreview.com/view/515016/one-time-pad-reinvented-to-make-electronic-copying-impossible.

43. Риксон, Фред Б. Коды, шифры, сигналы и тайная передача информации / Фред Б. Риксон - М.: АСТ: Астрель, 2011. – 656 с.

44. Мир математики. Т.2: Жуан Гомес. Математики, шпионы и хакеры. Кодирование и криптография. - М.: Де Агостини, 2014. - 144 с.

45. RFIDjournal.com

46. Мошенский, С.З. Эволюция векселя / С.З. Мошенский. - Ровно: «Планета-друк», 2005. – 446 с.

47. Автономов, А.С. Сравнительное избирательное право / А.С. Автономов, Ю.А. Веденеев, В.В. Луговой. - М.: Издательство НОРМА, 2003. — 208 с.

48. Автономов, А.С. Зарубежное избирательное право / А.С. Автономов, Ю.А. Веденеев, О.В. Дегтярева, В.В. Луговой, В.И. Лысенко. - М.: Издательство НОРМА, 2003. — 288 с.

49. Веденеев, Ю.А. Очерки по истории выборов и избирательного права / Ю.А. Веденеев, И.В. Зайцев, В.Е. Кораблин, В.В. Луговой, В.В. Тылкин. - Калуга : Калужский обл. фонд возрождения историко-культурных и духовных традиций «Символ», 2002. - 692 с.

50. Морозова, О.С. Обязательное голосование: мировой опыт и особенности избирательных практик. Каспийский регион: политика, экономика, культура - 2013. - № 3 (36).

51. Чуров, В.Е. Российский опыт использования технических средств в ходе региональных и муниципальных выборов 2008–2011 годов / В.Е. Чуров, Н.Е. Конкин, Г.И. Райков, А.В. Иванченко. — М.: РЦОИТ, 2011. — 328 с.

52. Мир математики. Т.3: Энрике Грасиан. Простые числа. Долгая дорога к бесконечности. - М.: Де Агостини, 2014. - 144 с.

53. Мир математики. Т.21: Ламберто Гарсия дель Сид. Замечательные числа: Ноль, 666 и другие бестии. - М.: Де Агостини, 2014. - 160 с.

54. Мир математики. Т.31: Хоакин Наварро. Тайная жизнь чисел. Любопытные разделы математики. - М.: Де Агостини, 2014. - 160 с.

55. Fujioka, A., Okamoto, T., and Ohta, K. A Practical Secret Voting Scheme for Large Scale Elections. In Proceedings of the Workshop on the theory and Application of Cryptographic Techniques: Advances in Cryptology (December 13 - 16, 1992). J. Seberry and Y. Zheng, Eds. Lecture Notes In Computer Science, vol. 718. Springer-Verlag, London, 244-251.

56. Cranor, L. and Cytron, R., 1997. Sensus: A Security-Conscious Electronic Polling System for the Internet. Proc. of the Hawaii International Conference on System Sciences. Wailea, Hawaii.

57. Federal Information Processing Standards Publication 197. Announcing the ADVANCED ENCRYPTION STANDARD (AES). November 26, 2001. nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

58. ГОСТ 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры. www.tc26.ru/standard/gost/GOST_R_3412-2015.pdf.

59. ГОСТ 34.13-2015. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров. www.tc26.ru/standard/gost/GOST_R_3413-2015.pdf.

60. Bitcoin Developer Reference. bitcoin.org/en/developer-reference.

61. Bitcoin Wiki. en.bitcoin.it/wiki/Main_Page.

62. Технологии криптовалют. www.intuit.ru/studies/courses/3643/885/info.

63. J. von Neumann. Various techniques used in connection with random digits. Collected Works, vol. 5, pp. 768–770, Pergamon Press, 1963. also: in Monte Carlo Method, National Bureau of Standards Series, Vol. 12, pp. 36-38, 1951.

64. Круглов, П. Поиск генераторов истинных случайных чисел [Электронный ресурс]. — Режим доступа: habr.com/ru/company/mailru/blog/408181 (дата обращения: 10.12.2019).

65. Intel Digital Random Number. Software Implementation Guide. - Intel Corporation. August 7, 2012. - 35 p.

66. R.C. Fairfield, R.L. Mortenson, K.B. Coulthart. 1985. An LSI random number generator (RNG). In

Proceedings of CRYPTO 84 on Advances in cryptology, G R Blakley and David Chaum (Eds.). Springer-Verlag New York, Inc., New York, NY, USA, 203-230.

67. David H. Bailey, Peter B. Borwein, Simon Plouffe. On the Rapid Computation of Various Polylogarithmic Constants // Mathematics of Computation. – 1997. – Т. 66, вып. 218. – С. 903–913