Н. Н. Токарева

Симметричная криптография

Краткий курс

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Механико-математический факультет
Кафедра теоретической кибернетики

Н. Н. Токарева

Симметричная криптография. Краткий курс

Учебное пособие

Новосибирск

2012

УДК 519.7
ББК 22.1
Т 510

ISBN 978-5-4437-0067-0

Токарева Н. Н. Симметричная криптография. Краткий курс: учебное пособие / Новосиб. гос. ун-т. Новосибирск, 2012. 234 с.

Учебное пособие представляет собой введение в современные методы симметричной криптографии и служит учебным материалом для спецкурса «Криптография и криптоанализ», читаемого автором для студентов ММФ НГУ и учащихся СУНЦ НГУ. В пособии отражены такие направления, как история криптографии в России, криптографические свойства булевых функций, алгоритмы блочного и поточного шифрования, статистические и алгебраические методы криптоанализа симметричных шифров.

Предназначено для школьников старших классов, студентов и преподавателей.

Рецензент — канд. физ.-мат. наук, доц. А. Л. Пережогин

Издание подготовлено в рамках реализации Программы развития государственного образовательного учреждения высшего профессионального образования «Новосибирский государственный университет» на 2009–2018 годы.

ISBN 978-5-4437-0067-0

Оглавление

Вместо предисловия

Криптография — наука о секретной передаче информации. На самом деле науки две. Собственно *криптография*, наука о шифровании, и *криптоанализ*, наука о взломе шифров. Оба направления имеют долгую и драматичную историю и до сих пор остаются «наиболее важными формами разведки в современном мире».

Научные методы криптографии — это методы дискретной математики, алгебры, теории вероятностей и математической статистики. И в то же время криптография — это искусство. Объекты, которые она исследует, могут не подчиняться математическим законам, а научные методы иногда не давать результата.

И тогда остаётся только *воображение*.

Истории криптографии в России, до сих пор остающейся мало изученной и во многом засекреченной, посвящена первая глава пособия. В дальнейшем основное внимание уделяется современным методам симметричной криптографии, в частности — использованию в конструкциях шифров криптографических булевых функций. Рассматриваются математические свойства таких булевых функций и способы их построения, приводятся необходимые утверждения дискретной математики и алгебры. Многие приводимые результаты о криптографических булевых функциях получены совсем недавно и пока не нашли отражения в учебной литературе. В пособии сделан обзор современных методов криптоанализа шифров. Приводится серия криптографических и комбинаторных задач, а также нерешённых проблем в области теоретической криптографии.

Автор выражает свою глубокую благодарность А. Д. Коршунову, А. Л. Пережогину и В. М. Фомичёву за ценные замечания и обсуждения.

Наталья Токарева
Академгородок, Новосибирск

Криптография незаметно проникает в кровь
и делает странные вещи с людьми.

Герберт Ярдли,
американский криптограф,
XX век

1. Из истории криптографии в России

Небольшие очерки, которые приводятся в данной главе, относятся в основном к истории криптографии в России. До сих пор она остаётся мало изученной. Иногда мы будем отступать от краткого хронологического изложения и останавливаться на отдельных событиях и людях, повлиявших на ход истории криптографии или своеобразно его отразивших.

1.1 Первые шифры

Появление первых зашифрованных текстов можно отнести к шестому тысячелетию до нашей эры. Это был текст «Великие Арканы Таро», написанный древними египтянами на символическом языке. Текст о принципах мироздания, об абсолютной и относительной истинах, дешифрованный лишь в 1998 году [9]. К V веку до н. э. относится шифр «сцитала», изобретенный в древней Спарте. С рассказа именно об этом шифре начинаются многие книги по криптографии. Сцитала — это цилиндр специального диаметра, послуживший первым шифровальным прибором. На цилиндр аккуратно наматывалась тонкая лента стык в стык, и на неё выписывалось секретное сообщение вдоль оси сциталы. Затем лента сматывалась и передавалась адресату, имеющему под рукой сциталу такого же диаметра и длины. Криптоанализ этого спартанского шифра предложил древнегреческий философ Аристотель (IV век до н. э.), став при этом первым криптоаналитиком. Аристотель использовал длинное конусообразное копье, наматывал на него ленту и сдвигал её по копью до тех пор, пока не удавалось прочесть осмысленный текст.

Один из наиболее известных древних шифров — это шифр Цезаря (I век до н. э.), которым великий полководец пользовался в своей секретной переписке. Шифрование заключалось в циклическом сдвиге каждой буквы сообщения на три позиции вправо по алфавиту. Например, применительно к русскому алфавиту, слово ЦЕЗАРЬ было бы зашифровано как ЩИКГУЯ. Для зашифрования и расшифрования достаточно было иметь перед глазами оригинальный алфавит и его сдвиг:

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
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

Несложно понять, что этот шифр может быть легко дешифрован.

Ещё одним известным примером древнего шифра служит «квадрат Полибия» (Древняя Греция, II век до н. э.). Как и шифр Цезаря, он является криптографически очень слабым. Однако шифр приобрел огромную популярность в российских и советских тюрьмах за удобный и быстрый способ кодирования сообщений. Другое его название — тюремная азбука. Считается, что в России в активное употребление этот шифр ввёл декабрист М. А. Бестужев. Для шифрования запоминалась следующая таблица размера 6×5

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

Заключённые перестукивались через стенку камеры так: сначала редким стуком указывался номер строки, затем через короткую паузу — номер столбца более частым стуком. Потом шла следующая буква и т. д. Таким способом, например, пользовался один из героев романа А. Рыбакова «Дети Арбата».

Большое значение иносказанию, которое можно считать одним из криптографических методов, придавали древнегреческие философы. Так, ещё в VI веке до н. э., как пишет Э. Шюре в своей книге «Великие посвящённые», «философы испытывали потребность для своего учения в двух различных доктринах, в одной — открытой для всех и в другой — тайной, которые передавали бы одну и ту же истину, но под различными формами и в мере, приспособленной для степени развития их учеников». Там же упоминается, что Пифагор «никогда не записывал своё эзотерическое учение иначе, как тайными знаками и под различными символами» [81].

Тайнописью активно пользовались и религиозно-мистические организации, так называемые «мистерии», послужившие впоследствии прообразом масонских обществ. «В мистериях существовал свой язык, доступный только посвящённым. В нём использовались мифы, метафоры, иносказания, сокращения, специальная символика и т. д.» [9].

1.2 Появление криптографии в России

Первые профессиональные криптографы на Руси появились при Иване Грозном (1530–1584). Они находились на службе в Посольском приказе, созданном им в 1549 году и отвечавшем за внешнюю политику страны. Криптографы разрабатывали так называемые «азбуки», «цифири», «цифры» или шифры, как они стали называться позднее. Вначале это были простые шифры замены.

Но всё-таки, как отмечает в своей книге [66] исследователь истории шифровального дела Т. А. Соболева, первым из российских государей, осознавшим всю важность криптографии для безопасности страны, стал Пётр I (1672–1725). Он поставил шифровальную службу действительно на профессиональную основу.

Г. И. Головкин

П. П. Шафиров

С 1700 года вся работа по созданию шифров и шифрованию велась в цифирном отделении Посольского приказа, а позднее, с 1709 года, — в Посольской канцелярии. Криптографическая служба в это время находилась под постоянным и непосредственным контролем государственного канцлера Гавриила Ивановича Головкина и вице-канцлера Петра Павловича Шафирова. Ими же заслушивались отчёты о перехваченных иностранных шифрах, что может свидетельствовать и о начале криптоаналитической деятельности. Затем криптографическая работа велась в Первой экспедиции Коллегии иностранных дел, где она стала строго регламентироваться и засекречиваться.

Типичным шифром того времени был шифр простой замены: каждая буква алфавита заменялась новым знаком, буквой или сочетанием букв. Кроме того, добавлялись «пустышки» — незначащие символы, а также вводились специальные обозначения для часто употребимых в определённом контексте слов или словосочетаний (такой словарь назывался «суплемент»). Авторство некоторых цифирей принадлежало лично Петру I.

У Петра I имелся даже специальный блокнот с шестью шифрами, которыми он активно пользовался. Однако в переписке случались и некоторые казусы. В книге [66] приводится такой пример. Пётр I часто употреблял французские шифры. В одном из писем фельдмаршал Г. Б. Огильви жаловался Головкину: «Французские цифирные грамотки нихто читать не может, тако не знаю, что на них ответствовать...» и писал напрямую Петру I: «...никого здесь нет, который бы французское ваше мог разуметь, понеже Рен ключ от того потерял... Изволте ко мне через цифирь мою писать, чтоб я мог разуметь...», на что Пётр I отвечал: «Французскою азбукою к вам писали для того, что иной не было. А которую вы перво прислали, и та не годна, понеже так, как простое письмо, честь можно. А когда другую прислал, то от тех пор ею, а не французскою к вам пишем. А и французской ключ послан.». Кажется, что потеря ключа в то время не сильно озадачивала переписчиков. Однако, позднее к Огильви был приставлен А. И. Репнин, доверенное лицо Петра I, которому было поручено наблюдать за действиями фельдмаршала.

1.3 Чёрные кабинеты

Криптографическую службу России продолжал курировать вице-канцлер. С 1725 года этот пост занимал Андрей Иванович Остерман. При нём шифры становятся неалфавитными — кодируются уже комбинации букв, а в качестве шифробозначений теперь используются исключительно цифры. В 1741 году с приходом к власти Елизаветы Петровны (1709–1761) вице-канцлером и главным директором почт назначается Алексей Петрович Бестужев-Рюмин. С его именем связано появление в России службы перлюстрации (тайного вскрытия почты). Осуществляется эта деятельность в «чёрных кабинетах» — тайных комнатах, имевшихся во всех крупных почтовых отделениях.

Вскрывать письма было непросто. Нужна была необыкновенная аккуратность и изобретательность. А иной раз ничего и не выходило. Например, об одной своей неудаче сообщал перлюстратор Ф. Аш в письме к Бестужеву-Рюмину: «...на письмах нитка таким образом утверждена была, что оный клей от пара кипятка, над чем письма я несколько часов держал, никак распуститься и отстать не мог. Да и тот клей, который под печатями находился (кои я хотя искусно снял), однако ж

не распустился. Следовательно же я к превеликому моему соболезнованию никакой возможности не нашел оных писем распечатать без совершенного разодрания кувертов. И тако я оные паки запечатал и стафету в ея дорогу отправить принуждён был...» [66]. Со временем в чёрных кабинетах появился целый штат сотрудников: одни вскрывали и запечатывали письма, другие прошивали их ниткой и подделывали печати, третьи копировали содержимое, четвёртые переводили, пятые занимались дешифрованием и т. д. Их деятельность держалась в строгом секрете.

Государственные интересы оказывались выше доводов морали. И не только в России. Надо сказать, что в европейских странах чёрные кабинеты начали свою работу лет на сто раньше.

А. И. Остерман

А. П. Бестужев-Рюмин

В 1742 году на «особливую должность» в Коллегию иностранных дел был принят первый профессиональный криптоаналитик. Им стал математик Христиан Гольдбах (1690–1764), получивший через год первые успехи на новом поприще. Он дешифровал ряд французских дипломатических шифров. Позднее криптоанализом занимались математики Ф. Эпинус и И. Тауберт; русские криптографы братья Ерофей и Фёдор Коржавины и другие, [54].

1.4 Первая половина XIX века

С начала XIX века вся криптографическая деятельность, а также руководство службой перлюстрации осуществляются в Канцелярии только что созданного (1802 г.) Министерства иностранных

дел. Непосредственно руководит Канцелярией (с 1810 года) статс–секретарь Карл Васильевич Нессельроде, позднее ставший министром иностранных дел и государственным канцлером.

К. В. Нессельроде

К числу ярких успехов того времени относится дешифрование военной переписки Наполеона. Этот факт сыграл важную роль в исходе Отечественной войны 1812 года и поражении наполеоновской армии.

– Нам очень сильно помогло то, что мы всегда знали намерения вашего императора из его же собственных депеш <...>

– Я считаю очень странным, что Вы смогли их прочесть. Кто-нибудь, наверное, выдал Вам ключ?

– Отнюдь нет! Я даю Вам честное слово, что ничего подобного не имело места. Мы просто дешифровали их.

(из разговора, состоявшегося после войны между императором Александром I и командующим одним из корпусов армии Наполеона маршалом Макдональдом)

Интересно, что выдающийся русский учёный Павел Львович Шиллинг (1786–1837) — электротехник, изобретатель первого электромагнитного телеграфа[1] и электрической мины, собиратель ценнейших коллекций китайских и японских рукописей, учёный–востоковед, отважный военный, участвовавший в сражениях Отечественной войны и награждённый одной из самых почётных наград — саблей с надписью «За храбрость», блестящий игрок в шахматы, «весельчак,

[1]Получивший распространение телеграф С. Морзе был запатентован в 1837 году — спустя пять лет после изобретения П. Л. Шиллинга.

отличный говорун» и, кстати, петербургский знакомый А. С. Пушкина и К. Н. Батюшкова — был, кроме того, одним из крупнейших криптографов XIX века! И эта особая сторона его многогранной деятельности долгое время оставалась засекреченной. Даже сейчас она ещё не исследована должным образом.

Павел Львович Шиллинг — крупнейший криптограф XIX века. Портрет и мини-потрет на советской марке

Известно, что с 1803 года П. Л. Шиллинг работал в МИД, в которое он вернулся и после окончания Отечественной войны. Именно он организовал министерскую литографию — способ плоской печати, только что входивший в употребление в Европе. До этого шифрдокументы копировались от руки. С 1818 года барон Шиллинг стал заведующим цифирной экспедицией Канцелярии МИД, занимающейся разработкой шифров. Ему принадлежит изобретение биграммного шифра, в котором шифровались не отдельные буквы, а их двойные сочетания, биграммы. При этом некоторые статистические зависимости снимались за счёт того, что в биграммы объединялись буквы, находящиеся друг от друга на большом расстоянии [66].

1.5 Шифры второй половины XIX века

Во второй половине XIX века криптографическая служба России перестала быть привилегией МИД и была создана ещё в двух ведомствах: военном и Министерстве внутренних дел. Тем самым суще-

ственно расширялись сферы использования криптографии, её значение в жизни государства неуклонно росло. Появилась классификация шифров по их назначению и области применения. Были выделены шифры военного ведомства (включая императорские), шифры жандармерии, гражданские шифры (например, Министерства финансов), агентурные шифры, предназначенные для связи с разведчиками.

Активно использовались биграммные шифры, введённые бароном Шиллингом; биклавные шифры — шифры многозначной замены, определяемой двумя ключами (автор — барон Н. Ф. Дризен); шифровальные коды — шифры, использовавшие цифры в качестве шифралфавита; коды с перешифровкой[2].

Термины «шифр» и «ключ» тогда ещё были синонимами. Ключом назывался, по сути, принцип шифрования сообщения, его алгоритм. Раскрытие ключа было равносильно гибели всей криптографической системы. Не так будет обстоять дело в XX веке, когда ключ станет сменной частью сложной криптосистемы. Раскрытие ключа не будет приводить к краху шифра, а будет означать лишь то, что ключ необходимо поменять.

Цифирный комитет устанавливал предельный срок действия каждого шифра в среднем от трёх до шести лет. Но любопытно, что многие шифры использовались и по истечении их «срока годности», что, несомненно, сказывалось на тайне переписки. Кроме того, имели место серьёзные нарушения. Так, представлялось возможным снова использовать скомпрометированные шифры в других регионах или спустя некоторое время.

Например, русский биграммный ключ №356 использовался почти 25 лет! История его такова. Он был введен в действие в 1869 году «в консульствах на Востоке». В 1888 году его экземпляр был украден из Российской миссии в Пекине. Вследствие этого шифр был выведен из употребления, но лишь на некоторое время. Несмотря на очевидность компрометации в начале 90-х годов XIX века шифр вновь ввели в действие, но уже в другом регионе. Он был направлен в Амстердам, Гаагу, Берн, Женеву, Стокгольм и другие города. В 1898 году произошла ещё одна компрометация этого шифра: один его экземпляр был потерян начальником Адриатической эскадры. Вероятно, именно это событие, как пишет Т. А. Соболева [66], наконец заставило руководителей шифрслужбы окончательно изъять ключ №356 из употребления. В

[2]Отметим, что сейчас термин «код» в криптографии почти не используется. «Коды» изучаются в *теории кодирования*, основной задачей которой является не сокрытие информации от злоумышленника, а защита её от помех при передаче по каналу связи. Таким образом, свой «секретный» смысл термин «код» утратил. А вот «шифры» криптография продолжает изучать.

соответствующем заключении было указано: «вследствие почти 1/4-векового всемирного использования». Лучше и не скажешь.

Но в целом криптографическая служба России в то время находилась на достаточно высоком профессиональном уровне. Очень быстро и эффективно работали чёрные кабинеты.

«Сохранить тайну шифра в Петербурге особенно трудно», — отмечал канцлер Германской империи Отто фон Бисмарк [22].

Один из бывших сотрудников спецслужбы перлюстратор С. Майский вспоминал: «Иностранная дипломатическая переписка попадала в руки российских специалистов практически полностью. «Чёрные кабинеты», разумеется, существовали везде, даже в самых демократических республиках Америки и Старого Света. Но справедливость требует сказать, что нигде в мире «чёрный кабинет» не работал так чисто, как в России, и в особенности в Петрограде» [22].

Интересно, что император Александр III в течение всего своего правления отказывался читать выписки из писем, добытые в чёрном кабинете. После вступления на престол и знакомства со службой перлюстрации он заявил: «Мне этого не нужно». Не так поступали другие императоры.

В Военном ведомстве второй половины XIX века чаще всего использовались «словарные ключи». Работали они так. Составлялся словарь небольшого объёма (до 1 000 словарных величин), каждому слову которого соответствовал код — трёх- или четырёхзначное число. Шифрование велось непосредственной заменой по словарю. Такие «военные ключи» действовали длительное время, при этом относительно часто менялся словарь.

Подобными (словарными) шифрами пользовался и Николай II. Примечательно, что словари Его Императорского Величества, предназначенные для деловой переписки, содержали множество слов с эмоциональной окраской. Например, такие: «бескорыстный», «безотрадный», «благородный», «болезненный», «ни под каким видом», «молва», «нелепый», «неправдоподобный» и др. [66].

Особое положение занимали *агентурные шифры*, использовавшиеся разведчиками и агентами царской охранки. Одним из основных требований, предъявляемых к таким шифрам, была «скрываемость», «безуликовость» их документации. Ключ должен был запоминаться или легко извлекаться из «окружающих предметов» (например, распространённых книг), наличие которых никак не компрометировало агента. Сам процесс шифрования должен был быть быстрым и простым. К числу таких шифров относились варианты шифра Цезаря, книжные шифры, шифры перестановок.

Книжный шифр при правильном использовании мог быть действительно очень надёжным и безуликовым. Выбиралась определённая книга, в которой номера страниц, строк и букв в строках служили шифробозначениями для шифруемых букв. Подобные шифры массово использовались и в русском подполье конца XIX века. Однако сотрудники «чёрных кабинетов» обнаружили несколько «зацепок» к раскрытию таких шифров. Оказалось, что корреспонденты предпочитали находить в книгах буквы, стоящие неподалеку от начала строки или страницы. Так, подсчёт номера буквы занимал меньше времени, да и риск ошибки был ниже. А вот редкие буквы обычно имели большие номера, так как в начале строк они попросту не попадались. Другой «зацепкой» было изъятие и внимательное изучение личной библиотеки каждого подозреваемого.

Подробнее о российской криптографии XIX века и начала XX века можно прочитать в статьях [10]–[14], [23]–[30].

1.6 Криптограф-соловчанин

В 1898 году сотрудник российской криптографической службы, коллежский регистратор, Владимир Иванович Кривош (1865–1942) был послан в Париж для изучения иностранного опыта в делах перлюстрации. В том числе устройства местного чёрного кабинета.

Любопытно описание этого учреждения, которое приведём по книге [66]. «Парижский чёрный кабинет был устроен аналогично петербургскому. Эта «секретная часть» находилась в частном доме. Официальная вывеска на нём гласила, что здесь располагается землемерный институт. Один из служащих «секретной части» действительно знал толк в лесоводстве, и если какой-то частный человек туда забредал, то ему давалась вполне квалифицированная справка. В передней комнате, куда мог прийти с улицы кто угодно, на стенах висели карты, планы земельных участков, а на столах лежали свежие газеты и письменные принадлежности. Из этой комнаты была дверь в следующую, в которой также не было ничего секретного, но был шкаф, служивший дверью в третью комнату. Таким образом, чтобы пройти в действительно секретную часть, необходимо было идти через шкаф, зная как его открыть (наступить одновременно на две дощечки на полу и нажать одно из украшений шкафа). Дверь автоматически сама запиралась за прошедшим через неё. В следующей комнате была перлюстрационная часть, имевшая сообщение пневматической почтой с главным почтамтом. Все прибывающие в Париж дипломатические пост-пакеты прежде всего отправлялись сюда. Здесь проводилась их регистрация и передача в кабинеты дешифровальщикам, занимавшимся с ними по двое. После дешифрования и фотографирования письма вновь заклеивались и отправлялись по той же трубе пневматическим способом на почтамт. Для президента ежедневно выпускался «ли-

сток» со всеми полученными за сутки сведениями — нечто вроде дипломатической газеты.»

Поездка не прошла даром. Вместе с французскими специалистами были раскрыты шифры, использовавшиеся Японией, Англией и Германией. В. И. Кривош, словак по происхождению, стал одним из ведущих российских криптографов. За предложенные им усовершенствования российской криптографической службы он получил орден Святого Владимира 4-й степени из рук П. А. Столыпина. Владимир Иванович постоянно приглашался для ведения заседаний государственных комиссий различного уровня секретности.

Удивительна судьба этого человека. В. И. Кривош родился в одной из деревень Австро-венгерской монархии. Он рос вблизи строящейся железной дороги и мечтал: когда дорога будет достроена, он уедет в далёкие края. Какими же суровыми и фантастическими они оказались...

Святое озеро. Соловецкий монастырь

Его родители были мелкими предпринимателями, которым хватило средств отправить на учёбу только одного сына, Владимира. Его одарённость открыла ему поистине удивительные перспективы. Сначала были гимназии: немецко-словацко-венгерская и итальянско-хорватская. Потом с поразительной быстротой — Королевская Ориентальная Академия, Петербургский университет, парижская Сорбонна. В 1890 году Владимиру Ивановичу — 25 лет. Он блестяще

образован, изучил математику и статистику, владеет пятнадцатью языками (к концу его жизни это число достигнет сорока!), написал диссертацию по арабской литературе и уже стал незаменимым российским специалистом в области криптографии и стенографии. Вскоре он становится Главным цензором газет и журналов Российской Империи и занимает множество «особых» и «сверхсекретных» должностей. Царское правительство использует его талант в самых разных областях.

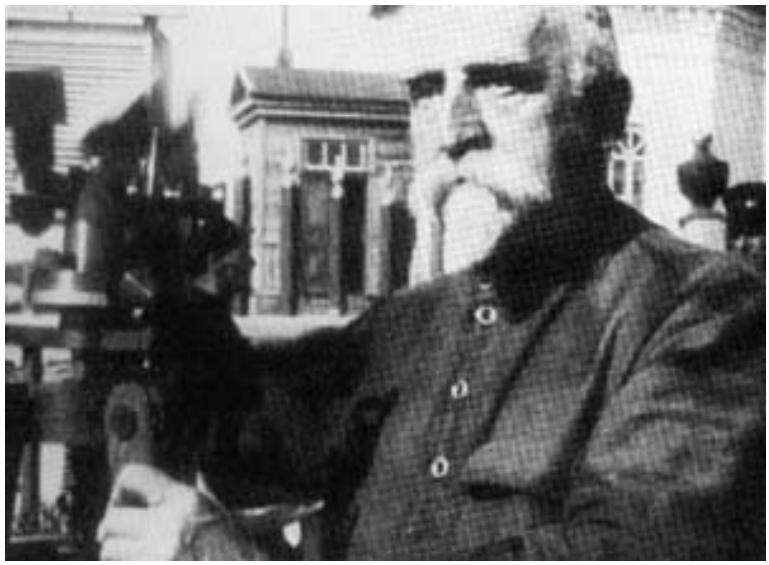
Однако в 1915 г. он попадает под подозрение в шпионаже. За этим следует разжалование и ссылка в Сибирь. И возвращение в революционный Петербург в 1917-м. Встреча и работа с В. И. Лениным, который зачисляет В. И. Кривоша в состав наркомата иностранных дел. Так начинается новый, советский, период его жизни.

А дальше, как пишет Любомир Гузи [32], исследователь жизни и творчества выдающегося криптографа, «жизненный путь этого человека напоминает шутку потерявшего всякую объективность биографа-графомана». Кривоша арестовывают: его прошлое дискредитирует советскую власть. Но расстрелять лучшего специалиста не решаются. Из тюрьмы он переводится на службу в разведку, потом становится переводчиком-дешифровальщиком Особого отдела ВЧК. Новый арест. И последовавший затем перевод в Спецотдел на разработку сложнейших шифров и их дешифрование. Вскоре «за принятие мер к выезду из страны» Кривош арестовывается и приговаривается к расстрелу. Но снова помилован. В мае 1922-го — очередное освобождение и очередное назначение в контрразведку. Год спустя — опять арест «за несанкционированные контакты с представителями чехословацкой миссии». В тюрьме он ожидает то расстрела, то ссылки в лагерь. Кривош временно теряет зрение, но, когда оно возвращается, с радостью читает тюремную библиотеку и занимается переводами. Он приговаривается к 10-летнему заключению в концлагерь, который ему разрешают выбрать самому — выбирает Соловки[3]. Главным образом потому, что первую волну заключённых составили преимущественно интеллектуалы бывшего режима.

На Соловках Кривош выбирает псевдоним «Тот, у которого ничего нет», что по-словацки звучит как «Нема нич». Он работает ботаником, зоологом, орнитологом, переводчиком, преподаёт иностранные

[3]Соловки, или СЛОН — Соловецкий лагерь особого назначения — первый концентрационный лагерь Советской России. Организован в 1923 году на базе древнего Соловецкого монастыря, расформирован в 1933-м.

языки, основывает оркестр, становится председателем научной комиссии по фауне и флоре Севера России.



Узник СЛОНа Владимир Иванович Кривош-Неманич, выдающийся криптограф царской и советской России

Фрагмент стены Соловецкого монастыря

В 1928 году Кривош-Неманич выходит на свободу. Дома его встречает жена, которая не отказалась от мужа во время всех преследований. До 1936 года этот удивительный человек (принявший фамилию Кирпичников) работает в Министерстве иностранных дел, но вернуться на словацкую родину ему не удается. Во время войны он живет в эвакуации в Уфе, где преподаёт иностранные языки. Умер Кривош-Неманич в августе 1942-го. Хоронил его сын, однако через несколько лет останки выдающегося криптографа царской и советской России были перемещены в братскую могилу, следы которой затерялись [69]...

1.7 Первая мировая война

Общий недостаточный уровень подготовки России к войне отразился и на работе криптографической службы.

В то время в российской армии практически отсутствовала надёжная проволочная телеграфная связь, поэтому основное взаимодействие между частями велось по радиосвязи. Но никакого отлаженного механизма использования шифрованной радиосвязи не было. Вследствие беспорядка с распределением и согласованием шифров радиостанции часто «не понимали» друг друга. Им приходилось передавать свои сообщения открытым текстом...

«Такое легкомыслие очень облегчало нам ведение войны на Востоке, иногда лишь благодаря ему и вообще возможно было вести операции», — вспоминал немецкий военачальник М. Гофман, позднее — командующий германскими войсками на Восточном фронте.

«Русские пользовались своими аппаратами так легкомысленно, как если бы они не предполагали, что в распоряжении австрийцев имеются такие же приемники, которые без труда настраивались на соответствующую волну. Австрийцы пользовались своими радиостанциями гораздо экономнее и осторожнее и, главным образом, для подслушивания, что им с успехом удавалось. Иногда расшифровка удавалась путём догадок, а иногда при помощи прямых запросов по радио во время радиопередачи. Русские охотно помогали «своим», как они считали, коллегам.» — из отзыва М. Ронге — начальника разведывательного бюро австрийского генштаба, см. [22].

Всё это очень грустно.

Ошибка с передачей специального военного шифра сыграла определённую роль в поражении армии А. В. Самсонова на Мазурских островах у Танненберга [66]. Во время восточно-прусской операции в августе 1914 года две армии (Самсонова и Ренненкампфа), выступив до завершения мобилизации, должны были оттянуть на

себя часть немецких сил, тем самым сорвав основное наступление Германии против Франции. Но сценарий реализовался другой. При взаимодействии двух армий оказалось, что в армии П. К. Ренненкампфа новый шифр уже получен, а старый уничтожен, а в армии Самсонова ещё действовал старый шифр. Поэтому радиопереговоры между ними велись в открытую, чем не могло не воспользоваться немецкое командование. Кроме того, армия Самсонова не имела запасов телеграфной проволоки, командованию и разведке приходилось использовать для связи даже телефоны местных жителей. В то же время посылаемые приказы командующего фронтом о своевременном отходе армий к определённым рубежам просто не доходили до Самсонова. Его армия попала в окружение и героически сражалась, оставшись без какой-либо поддержки. К ней на помощь должна была прийти армия Ренненкампфа, но не пришла: по оценкам историков, это было фактическое предательство. В результате, армия Самсонова была уничтожена. Потери составили десятки тысяч убитыми, ранеными и пленными [66]...

Русские офицеры. Фото времен Первой мировой войны

В сентябре 1914 года российскому командованию всё-таки удалось обеспечить войска шифровальными средствами. Однако новый шифр был без труда раскрыт дешифровальной службой Австро-Венгрии уже через пять дней после его введения! Наши противники бесперебойно читали шифрпереписку русской армии. Потом они настолько привыкли к этому, что даже не отдавали приказов до тех пор, пока не получали очередной порции информации от своих дешифровальщиков.

В целом «войну в эфире» мы проиграли. Причинами этого послужили плохая организация шифрованной радиосвязи царских армий,

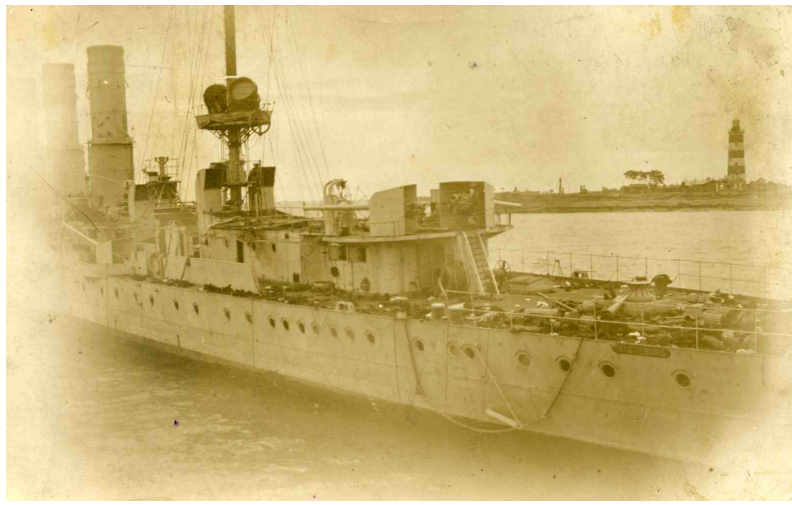
слабость российских шифров и нарушения в их использовании. К этому необходимо добавить и то, что до войны в России не существовало военных дешифровальных отделений (они были только у Франции и Австро-Венгрии). Когда такие отделения были созданы, им не хватало соответствующих специалистов и оборудования — например радиостанций пеленгации и перехвата.

Отметим и ряд успехов нашей криптографической службы. Перед войной и во время войны дешифровальная служба МИД работала довольно результативно. Ею читалась переписка многих иностранных государств (в первую очередь, Австрии, Германии, Болгарии, Италии, Турции и др.). Позднее число перехваченных и дешифрованных телеграмм снизилось в связи с тем, что Германия и Австро-Венгрия стали чаще использовать телеграф, а не радиосвязь. В недавно созданных военных дешифровальных отделениях достаточно быстро вскрывались ключи немецкого морского шифра, что позволяло читать немецкие сообщения и приказы.

К числу успешных относится и операция по захвату кодовых книг с затонувшего немецкого крейсера «Магдебург» в 1914 году.

Приведём эту историю, следуя книге [22]. «В августе 1914 года наскочил на мель в восточной части Балтийского моря у острова Оденсхольм лёгкий немецкий крейсер «Магдебург». Русские моряки сумели достать с этого крейсера кодовые книги ВМС Германии. Для того чтобы скрыть факт захвата кодовых книг с «Магдебурга» от немцев, русские провели следующую операцию. Немцы не знали, что командир «Магдебурга» Хабенихт при аварии был тяжело ранен и умирал в госпитале. В операции было решено использовать двойника командира немецкого крейсера. В Шлиссельбурге под охраной жил офицер русского флота И. И. Ренгартен. Он свободно говорил по-немецки и был внешне похож на Хабенихта. Как и рассчитывало русское командование Балтфлотом, немцы сумели выйти с ним на связь. Это было сделано с помощью немецких газет, которые «командир» заказывал в шведском посольстве. Над буквами одной из статей Ренгартен обнаружил еле видные точки. Помеченные буквы складывались в следующий текст: «Где книги? Если уничтожили их, сообщите так: если утопили, попросите журнал «Иллюстрированные новости», если сожгли, то «Шахматный журнал Кагана» — номер, соответствующий номеру котла на "Магдебурге"». Ренгартен заказал «Шахматный журнал Кагана» номер 14. Именно в этом котле крейсера русскими были сожжены фальшивые кодовые книги и подлинные обложки в свинцовом переплёте. На следующий день к сидящему на камнях «Магдебургу» подошла немецкая подводная лодка. Высадившаяся из неё на крейсер группа извлекла пепел от «сгоревших кодовых книг», остатки переплёта и кожи от обложек. Русские подводную лодку «не заметили». Так немцы убедились в том, что кодовые книги с «Магдебурга» уничтожены. В результате свой код они не сменили.» [22].

Примечание автора к электронной версии (2014). После выхода данного пособия автору стало известно о том, что приведённые выше детали захвата крейсера «Магдебург» являются вымышленными. Обсуждению подлинной истории крейсера «Магдебург» и мифам вокруг неё посвящены статьи *«Рифы и мифы острова Оденсхольм. К истории захвата секретных документов германского флота на крейсере "Магдебург"» в августе 1914 года*, *«"Магдебургская" история — "работа над ошибками"»* (автор – М. А. Партала) в журнале «Защита информации. Inside» (2007, 2014). Выражаю свою благодарность автору статей за указание на эти работы.



Крейсер «Магдебург», севший на мель. Справа виден маяк о. Оденсхольм (северное побережье современной Эстонии)

Для России это был большой успех. Захваченными шифрами русские поделились с англичанами. Уинстон Черчиль, получивший доступ к этим документам, назвал их «бесценными». Англичане эффективно использовали русский подарок. Они не только дешифровывали ценные телеграммы, но и посылали сообщения от имени германского командования. Одно из таких сообщений привело к крупной победе англичан на море: была уничтожена немецкая эскадра под командованием генерала Шпее осенью 1914 г. недалеко от Южной Америки.

1.8 «На грани крушения»

Глубокий кризис, который переживала российская криптографическая служба, особенно остро ощущался самими криптографами. Многие из них искренно переживали за судьбу не то́лько своей службы, но и России в целом. Юрий Александрович Колемин, управляющий шифровальной частью МИД, писал в своей докладной записке министру иностранных дел С. Д. Сазонову о необходимости немедленной реорганизации криптографической службы, находящейся, по его словам, «на грани крушения». Он писал о ничтожных окладах её сотрудников, об их «второсортном положении», а по сути об их ненужности государству. «На каком именно основании, — пишет Колемин, — они должны чувствовать солидарность с интересами сво-

его дела?», ведь «добросовестность нельзя безнаказанно эксплуатировать». В его записке встречаются такие слова, как «крах», «безнадёжность», «банкротство»... Пользуясь активной поддержкой своих служащих, он предлагает организацию нового Отделения, детально разрабатывает принципы его устройства, вкладывает в это дело «всю свою душу»! Но этим планам не суждено было реализоваться. На пороге стоял 1917 год. И история начиналась совсем другая.

1.9 Глеб Бокий и начало советской криптографии

«По ночам Самбикин долго не мог заснуть от воображения труда на советской земле, освещённого сейчас электричеством. Он вставал с кровати, зажигал свет и ходил в волнении, желая предпринять что-либо немедленно. Он включал радио и слышал, что музыка уже не играет, но пространство гудит в своей тревоге, будто безлюдная дорога, по которой хотелось уйти.»

А.Платонов, «Счастливая Москва».

Пятого мая 1921 года при ВЧК был создан Спецотдел, заведовавший криптографическими делами. Отдел находился на особом положении: его действия координировались непосредственно Политбюро. Распоряжения Спецотдела по всем вопросам шифрования были обязательными к исполнению всеми ведомствами РСФСР. Возглавил новую криптографическую службу Глеб Иванович Бокий (1879–1937), соратник В. И. Ленина.

Об этом человеке трудно найти какую-либо информацию. Особенно непротиворечивую. Историк Т. А. Соболева в своей книге [66] пишет: «Даже в моём собственном окружении, в той самой службе КГБ, которая была детищем Бокия и которую он возглавлял 17 лет, о нём почти никто ничего не знал».

Дворянин Г. И. Бокий вступил в Российскую социал-демократическую рабочую партию (РСДРП) в 1900 году. Кстати, его партийный билет был номер 7. «Бокий, как и Сталин, в предреволюционный и революционный период входил в ядро, руководящую верхушку большевистской партии» [66]. На протяжении 20 лет (с 1897 по 1917 год) он являлся одним из руководителей петербургского большевистского подполья. За это время Бокий двенадцать раз подвергался арестам, провёл полтора года в одиночной камере, два с половиной года — в сибирской ссылке, от побоев в тюрьме получил травматический туберкулёз [56]. Параллельно с революционной деятельностью

он учился в Петербургском горном институте, работал гидротехником и горным инженером. Основательно изучал философию и политэкономию. «Работал над своим образованием настолько упорно, что позволял себе спать не более четырёх часов в сутки» [66].

Глеб Иванович Бокий

Максим Горький писал о нём так: «Человек из породы революционеров-большевиков старого, несокрушимого закала. Я знаю почти всю его жизнь, всю работу и мне хотелось бы сказать ему о моём уважении к людям его типа, о симпатии лично к нему. Он, вероятно, отнёсся бы к такому «излиянию чувств» недоумённо, оценил бы это как излишнюю и, пожалуй, смешную сентиментальность». [31].

В советский период Г. И. Бокий не только руководил Спецотделом, но и был членом ВЧК, затем коллегии ОГПУ и НКВД, входил в состав «троек» ОГПУ, приговаривавших людей к расстрелам — часто заочно: без участия обвиняемых, свидетелей и защиты. Он был одним из активных создателей системы ГУЛАГ. В частности, Соловецкого лагеря, уже упоминавшегося в этой главе.

Именем Бокия был назван пароход, в трюме которого в Соловки привозили новых заключённых. Известный советский учёный–филолог Дмитрий Сергеевич Лихачёв, узник Соловков, вспоминал свою поездку на пароходе так: «Вывели нас на пристань с вещами, построили, пересчитали. Потом стали выносить трупы задохшихся в трюме или тяжело заболевших: стиснутых до перелома костей, до кровавого поноса...» [48]. А однажды, на пароходе прибыл и сам «куратор Соловков». Но это был его рабочий визит. Заключённые лагеря сочинили тогда такие строки [69]: «В волненье все, но я спокоен. // Весь шум мне кажется нелеп: // Уедет так же, как приехал, // На «Глебе Боком» — Бокий Глеб.»

На службе у Бокия работали некоторые криптографы царской России. Был здесь В. И. Кривош-Неманич, И. А. Зыбин, дешифровавший в своё время переписку Ленина, И. М. Ямченко, бывший начальник врангелевской радиостанции. В создании службы участвовали и люди, ранее не работавшие в области шифрования. Зять Бокия, писатель Лев Разгон, вспоминал: «В спецотделе работало множество самого разного народа, так как криптографический талант — талант от Бога. Были старые дамы с аристократическим прошлым, был немец с бородой почти до ступней» и множество других непонятных людей.

К работе на Спецотдел Бокий привлек и учёного-мистика А. Барченко, исследовавшего биоэлектрические явления в жизни клетки, в работе мозга и в живом организме в целом. Свои лабораторные опыты Барченко совмещал с должностью эксперта Бокия по психологии и парапсихологии. В частности, им разрабатывалась методика выявления лиц, склонных к криптографической работе. Учёный выступал консультантом при обследовании всевозможных знахарей, шаманов, гипнотизёров и прочих людей, утверждавших, что они общаются с призраками. С конца 1920-х годов Спецотдел активно использовал их в своей работе. Как отмечается в книге [56], исследования и методика Барченко применялись и в особенно сложных случаях дешифрования вражеских сообщений — в таких ситуациях проводились сеансы связи с духами.

Первый успех новой криптографической службы относится к 1921 году: был раскрыт немецкий дипломатический код. С этого времени и вплоть до 1933 года контролировалась переписка многих линий дипломатической связи Германии и её консульств в СССР. С 1921 года читалась переписка внутренних линий связи Турции. В 1924 году были вскрыты два шифра польского разведотдела генерального штаба, которые использовались для связи с военными атташе в Москве, Париже, Лондоне, Вашингтоне и Токио. В 1927 году началось чтение японской переписки, в 1930 году — переписки некоторых линий связи США. Разрабатывались коды и других стран.

Одновременно со «взломом» чужих шифров шла напряжённая работа по созданию своих. В 1924 году на основе 52 различных шифров был создан так называемый «русский код», дешифровать который не удалось никому. В литературе по истории криптографии об этом коде нет информаций. По одним источникам, «на десятилетия он стал основным шифром для всех служб СССР», по другим — такого кода никогда не было.

Несмотря на большой спектр решаемых задач спецотдел ВЧК, а затем ОГПУ был в их структуре самым засекреченным. Его сотруд-

никам запрещалось даже родным говорить, где они работают.

В 30-е годы руководство криптографической службой сменилось, а Глеб Бокий был расстрелян.

Его жизнь окружена множеством легенд. Кто-то их подтверждает, кто-то яростно опровергает. Часто говорится о связи Г. И. Бокия с представителями тайных обществ, о его поисках Шамбалы — страны вечных мудрецов, по преданию затерянной где-то в Азии. В 1925 году он даже планировал туда научную экспедицию, но запретило Политбюро.

Одни считали Бокия «страшным человеком», устраивавшим тайные оргии на своей даче. Вспоминали, что некоторые сотрудники спецотдела, принимавшие в них участие, потом заканчивали жизни самоубийством. Атмосферу созданной им «дачной коммуны» сравнивали с атмосферой Великого бала у сатаны в романе его современника М. А. Булгакова «Мастер и Маргарита». Только в действительности, вспоминали очевидцы, было ещё страшнее. Другие с возмущением отвергали подобные «байки», считая их «версией, которую пустили в обиход после ареста Бокия». Эти люди вспоминали Бокия как «интеллигентного и весьма скромного человека, никогда и никому не пожимавшего руки и отказывавшегося от всех привилегий». Как человека, который «на сделку с совестью не шёл никогда».

1.10 Секретная связь во время Великой Отечественной войны

В этом и следующих разделах речь пойдёт о методах секретной связи и криптографии в СССР во время Великой Отечественной войны и в период подготовки к ней. Это и секретная телефонная связь, и радиосвязь, и создание текстовых шифраторов.

Первые разработки аппаратов секретного телефонирования в СССР относятся к 1927–1928 гг., когда в Научно-исследовательском институте связи РККА были изготовлены для погранохраны и войск ОГПУ 6 телефонных аппаратов ГЭС (конструктор Н. Г. Суэтин).

В 1930-х годах в области секретной телефонии вели работы семь организаций: НИИ НКПиТ (наркомата почт и телеграфа), НИИС РККА, завод имени Коминтерна, завод «Красная Заря», НИИ связи и телемеханики ВМФ, НИИ №20 Наркомата электропромышленности (НКЭП), лаборатория НКВД.

ВЧ-связь. В 1930 году заработали первые линии междугородной правительственной высокочастотной связи (ВЧ-связи) Москва — Ленинград и Москва — Харьков. Отметим, что сама технология ВЧ-связи без применения аппаратуры шифрования была совершенно ненадёжна и могла защитить только от прямого прослушивания. В

1935–1936 годах на заводе «Красная Заря» было создано устройство автоматического засекречивания телефонных переговоров — *инвертор ЕС* (названный по фамилиям разработчиков К. П. Егорова и Г. В. Старицына) — и налажен его выпуск для каналов телефонной ВЧ-связи. Практически на всём протяжении Великой Отечественной войны и позднее для организации ВЧ-связи успешно использовались устройства этого типа.

К 1941 году в СССР функционировало 116 ВЧ-станций и 39 трансляционных пунктов, а число абонентов высшего партийного и государственного руководства достигло 720.

К первому периоду войны относится разработка портативной, исполненной в виде чемодана, засекречивающей аппаратуры СИ-15 («Синица») и САУ-16 («Снегирь»), которая использовалась в основном при выездах высшего командного состава в пункты, не имевшие ВЧ-станций.

В 1938–1939 гг. в Центральном научно-исследовательском институте связи были созданы две лаборатории по засекречиванию телеграфной и телефонной информации. Возглавил их выдающийся учёный Владимир Александрович Котельников. Это человек, сыгравший ключевую роль в организации надёжной секретной связи самого высокого уровня во время Великой Отечественной войны и после неё.

Секретная телеграфная связь. В. А. Котельниковым впервые в СССР были разработаны принципы построения телеграфной засекречивающей аппаратуры путём наложения на сообщение знаков гаммы (аппаратура «Москва»).

Как приводится в статье [15], «Сам шифратор, сконструированный на электромеханических узлах, был сложным и громоздким. В основе конструкции лежал барабан, заполненный шариками. При вращении барабана через систему штырей из щелей шарики случайным образом скатывались по шести вертикальным трубкам на две движущиеся телеграфные ленты, которые были наложены одна на другую через «копирку». В результате на обеих лентах получался одинаковый рисунок — «дорожки» из случайно расположенных пятен. Затем по этим меткам ленты перфорировались. Эти ленты образовывали случайный ключ и рассылались на пункты установки аппаратуры.».

Сама схема наложения гаммы на открытый текст была уже хорошо известна к тому времени благодаря изобретению Гильберта Вернама 1917 года. Она оказалась очень привлекательной и долгое время использовалась в аппаратуре последующих поколений.

Владимир Александрович Котельников

Секретная телефонная связь. В 1939 году В. А. Котельникову было поручено решение важной государственной задачи — создание шифратора для засекречивания речевых сигналов с повышенной стойкостью к дешифрованию.

В лаборатории Котельникова было установлено, что для хорошей маскировки речевого сигнала необходимо использовать *частотные преобразования* и *временные перестановки* отрезков речи одновременно [18]. Эти принципы легли в основу новой разработанной под руководством Котельникова сложной засекречивающей аппаратуры С-1 («Соболь») [38], которая стала широко использоваться в действующей армии. Несмотря на все трудности уже к осени 1942 года сотрудники лаборатории Котельникова изготовили несколько образцов оборудования «Соболь-П». Согласно статье [15] это была самая сложная аппаратура засекречивания информации, не имевшая аналогов в мире. «Соболь-П» использовался для обеспечения секретной связи самого высокого уровня (Ставки Верховного главнокомандующего со штабами фронтов), причём впервые такая связь осуществлялась с помощью радиоканала. Заменить относительно безопасный проводной канал связи на радиоканал для связи такого уровня оказалось возможно только благодаря *исключительной стойкости* использованного шифрования.

Как вспоминали ветераны ВОВ, применение шифраторов Котельникова в ходе решающих боев на Курской дуге в значительной степени определило успешный исход

битвы [15]. По сведениям советской разведки, А. Гитлер заявлял, что за одного криптоаналитика, способного «взломать» советскую радиосвязь, он не пожалел бы трёх отборных дивизий.

Уже в то время В. А. Котельников понимал, что для обеспечения высокой стойкости и уровня маскировки речевого сигнала необходимо сначала проводить сжатие речи [18]. Поэтому параллельно с разработкой «Соболя-П» В. А. Котельников проводил работы по созданию *вокодера* — устройства, обеспечивавшего компрессию спектра речи примерно в десять раз. К октябрю 1941 года вокодер начал «говорить». В ноябре 1941-го лаборатория продолжила свою работу в Уфе, куда была эвакуирована.

За создание шифраторов В. А. Котельников и его коллеги по лаборатории (И. С. Нейман, Д. П. Горелов, А. М. Трахтман, Н. Н. Найдёнов) получили в марте 1943 года Сталинские премии I степени.

Все разработки были жёстко засекречены. Сотрудник лаборатории Е. В. Руднев передает атмосферу секретности в своих стихах–воспоминаниях [60]: «В 43-м весною, по радио // Мы узнали — трудились не зря. // Золотыми нагрудными знаками // Удостоена эта работа была. // Первый том был написан В. А. // Мой четвёртый — последний, // Между ними два тома Д. Б. и Ю. С. // Все четыре — под грифом С. С.»

Аппаратура «Соболь-П» очень активно использовалась в дальнейшем. После окончания Второй мировой войны она получила применение и на дипломатических линиях связи Москвы с Хельсинки, Парижем и Веной при проведении переговоров по заключению мирных договоров, а также при проведении Тегеранской, Ялтинской и Потсдамской конференций и для связи с Москвой нашей делегации во время принятия капитуляции Германии в мае 1945 года. За дальнейшие разработки в области шифраторов Котельникову и его группе в 1946 году была повторно присуждена Сталинская премия I степени.

* * *

Одновременно с созданием аппаратуры засекречивания в СССР проводились и работы по её дешифрованию. Было установлено, что аналоговая аппаратура шифрования мозаичного типа теоретически дешифруема. Для того чтобы получить недешифруемую аппаратуру засекречивания телефонных переговоров, речь необходимо переводить в *цифровую форму*.

В 1941 году Владимир Александрович доказал, что можно создать *математически недешифруемую систему* засекречивания, если каждый знак сообщения будет засекречиваться выбираемым случайно и равновероятно знаком гаммы. Параллельно и независимо к этим идеям пришёл выдающийся американский учёный Клод Шеннон. Подобные системы он стал называть *совершенно секретными шифрами*. Такие системы, как показал В. А. Котельников, должны быть цифровыми, а преобразование аналогового сигнала в цифровую форму должно основываться на доказанной им *теореме отсчётов* (другое название — *теорема дискретизации*). Эта теорема, как и другие результаты В. А. Котельникова, прочно вошла в Золотой фонд классических результатов современной теории информации.

Теорема Котельникова. *Если аналоговый сигнал $s(t)$ имеет ограниченный спектр, то он может быть восстановлен однозначно и без потерь по своим дискретным отсчётам, взятым с частотой не менее удвоенной максимальной частоты спектра.*

Другими словами, теорема говорит о возможности восстановления непрерывных функций с ограниченным спектром по их значениям через определённые интервалы времени. Заслуга Котельникова состоит в том, что он первый осознал возможности приложений этого математического факта и осуществил удачный выбор класса функций для дискретизации. В зарубежной литературе эта теорема более известна как теорема Найквиста – Шеннона.

С этих результатов В. А. Котельникова, представленных в секретной научной работе «Основные положения автоматической шифровки» (1941) и независимо полученных К. Шенноном в 1945 году, и началась криптография как наука. Для криптографических методов впервые за всю историю их развития был разработан строгий математический аппарат.

Необходимо отметить, что результаты К. Шеннона были опубликованы в открытой печати, тогда как работы В. А. Котельникова долгое время оставались засекреченными. Поэтому приоритет в этой области по праву закреплён за К. Шенноном.

* * *

После войны, 21 января 1948 года была создана секретная Марфинская лаборатория [38] для работы над следующими проблемами:

- дискретизация непрерывного речевого сигнала;
- увеличение скорости передачи двоичных сигналов;
- разработка высокоскоростного шифратора;

- создание нового направления — криптографического анализа.

Однако В. А. Котельников отказался возглавить лабораторию; он только её курировал. В это время он продолжал работать в Московском энергетическом институте (МЭИ). Руководил Марфинской лабораторией А. М. Васильев.

В новой лаборатории вместе с вольными работали заключённые спецтюрьмы №16 МГБ СССР. На проведение всех работ руководство страны поставило сверхкороткий срок — полтора года! Атмосфера была очень напряжённой. Марфинская криптографическая лаборатория описана в романе А. И. Солженицына «В круге первом».

«Марфинская шарашка», спецтюрьма №16 МГБ СССР

Правда, как вспоминал в 2003 году В. А. Котельников, «Солженицын не слишком правильно описал атмосферу в лаборатории. «Вольным» приходилось работать больше, чем заключённым, и кормили их много хуже. Усложняла работу и обстановка излишней секретности. Закрытость и секретность вообще много вреда принесли нашей науке» [36].

1.11 В. А. Котельников

Владимир Александрович Котельников (1908–2005) родился в Казани, в семье университетского профессора — известного математика — Александра Петровича Котельникова. Его дед, Пётр Иванович Котельников, также всю жизнь проработал математиком в Казанском университете и между прочим был первым математиком, который

открыто поддержал смелые работы Н. И. Лобачевского о неевклидовой геометрии.

В 1926 году Владимир Александрович поступил в Московское высшее техническое училище им. Баумана, на последних курсах перешел в отделившийся от училища Московский энергетический институт (МЭИ), который и окончил в 1930 году, получив звание инженера-электрика. Однако научная деятельность В. А. Котельникова началась раньше — в 1930 году в НИИ связи Красной армии, куда он был зачислен в качестве инженера. Одновременно с научной деятельностью с 1931 по 1941 г. В. А. Котельников преподавал на кафедре радиотехники МЭИ. В конце 30-х гг. и в годы войны Владимир Александрович занимался разработкой специальной шифровальной аппаратуры связи. Достигнутые им криптографические успехи без преувеличения внесли огромный вклад в нашу победу, а сам В. А. Котельников снискал себе негласный титул «патриарха секретной телефонии».

В конце 40-х годов Владимир Александрович организовал Особое конструкторское бюро МЭИ, ставшее впоследствии одним из ведущих предприятий в области космической техники. С 1954 года В. А. Котельников возглавил недавно созданный Институт радиотехники и электроники (ИРЭ), получивший при нём мощное развитие.

В 90-е годы В. А. Котельников был одним из шести основателей Академии Криптографии. Он принимал активное участие в работе её советов и комиссий.

Обширные научные интересы Владимира Александровича включали в себя общую и прикладную физику; радиофизику, радиотехнику и электронику; теорию информации, в частности — методы защиты информации от помех в системах радиосвязи и криптографию, практические вопросы разработки специальной аппаратуры связи; исследования космоса, в особенности созданное им направление планетной радиолокации и многое, многое другое. Во всех этих областях В. А. Котельников получил существенные результаты. В этом пособии мы коснулись лишь его криптографической деятельности.

В. А. Котельников — лауреат Ленинской премии, дважды лауреат Государственной премии СССР, дважды Герой Социалистического труда. Он награждён шестью орденами Ленина, орденом «За заслуги перед Отечеством» I степени, другими орденами и медалями, в частности орденом «За заслуги перед Москвой». Хоть и с большим опозданием его научные заслуги были признаны во всем мире. В 1999 году ему были присуждены высшие международные награды — премия Э. Рейна

и Золотая медаль А. Белла. Решением Международного астрономического союза астероид №2726 носит имя «Kotelnikov».

Президент Международного Института электронной и электрической инженерии Брюс Эйзенштейн (США) признавал самый существенный вклад Котельникова в развитие радиосвязи и криптографии, он говорил: «Over the years the West had its Shannon; and the East had its Kotelnikov.»

Владимир Александрович Котельников

Полученные звания и награды не стали препятствием основному делу его жизни: Владимир Александрович сохранил научную активность до самого последнего времени. Его творческий и земной путь завершился на 97-м году жизни почти законченной, но не опубликованной работой «Модельная квантовая механика».

Коллеги В. А. Котельникова отмечали его выдающиеся личные качества. «Прежде всего, это необычайная серьёзность в подходе к решению любого вопроса, будь то государственная проблема или личная проблема сотрудника. Далее, неизменная доброжелательность, обязательность в выполнении обещанного, стремление всегда решить вопрос не откладывая на завтра — вот те замечательные качества, которые характеризовали Владимира Александровича как руководителя и как человека» (директор ИРЭ академик Ю. В. Гуляев [33]).

На работе и в жизни Владимир Александрович был скромен и прост. Он умел видеть главное и быть требовательным. Читая воспоминания [40] тех, кому посчастливилось работать с ним, отмечаешь, что в присутствии Владимира Александровича людям неожиданно

становилось... стыдно. За себя, за недостатки в своей работе. И они старались работать лучше, не позволяя себе «халтуры», и сами менялись к лучшему.

1.12 Немного о советских шифрмашинах

Когда речь заходит о шифровальной технике времен Второй мировой войны, то, как правило, вспоминают знаменитую немецкую шифровальную машину «Энигма» — изобретение инженера Артура Шербиуса 1918 года. Этим дисковым шифратором с 1926 года стали оснащаться вооружённые силы и спецслужбы Германии. Наиболее востребованной «Энигма» стала после прихода к власти А. Гитлера и особенно во время войны. По некоторым оценкам для вооружения немецкой армии было выпущено до 100 000 её экземпляров.

Вспоминают тайную работу по дешифрованию «Энигмы», которая велась в разных странах и увенчалась успехом. На протяжении войны немецкие сообщения тайно читались англичанами, американцами, русскими и др.

Первый математический аппарат для дешифрования «Энигмы» разработали выпускники Познаньского университета (Польша) Мариан Раевский, Генрих Зыгальский и Ежи Розицкий (см. подробнее [39], [22], [65]). В 2008 году в Познани о них был снят документальный фильм [113]. Результатами польских криптоаналитиков воспользовались англичане. Ими была спланирована масштабная операция «Ультра», нацеленная как на аналитическое дешифрование сообщений «Энигмы», так и на захват её действующих кодовых книг. Об этой успешной операции и её главном криптографическом центре в Блетчли–парке написано довольно много. Большой объём сведений можно найти и о самой «Энигме».

А какими были советские шифрмашины?

До последнего времени информации о них практически не было.

* * *

«...кто возьмет в плен русского шифровальщика, либо захватит русскую шифровальную технику, будет награждён Железным крестом, отпуском на родину и обеспечен работой в Берлине, а после окончания войны — поместьем в Крыму.».

«Эти проклятые русские шифровальные машины, мы никак не можем их расколоть!».

Многое дают понять эти слова А. Гитлера. Он инициировал настоящую охоту за советскими шифровальщиками. Однако немцам так и не удалось дешифровать сообщения, зашифрованные с помощью советской техники. С 1942 года эти сообщения перестали перехватывать. Благодаря разработанной перед войной шифровальной технике Советскому Союзу удалось скрыть свои стратегические планы. Это был огромный успех нашей шифровальной службы!

В подтверждение приведём несколько цитат.

«Ни одно донесение о готовящихся военно-стратегических операциях нашей армии не стало достоянием фашистских разведок» (начальник Генштаба Маршал Советского Союза А. М. Василевский).

«Хорошая работа шифровальщиков помогла выиграть не одно сражение» (зам. Верховного Главнокомандующего Маршал Советского Союза Г. К. Жуков).

В своих показаниях начальник штаба при ставке верховного главнокомандования немецких вооружённых сил генерал-полковник А. Йодль сообщал: «Радиоразведка играла особую роль в самом начале войны, но и до последнего времени не теряла своего значения. Правда нам никогда не удавалось перехватить и расшифровать радиограммы вашей ставки, штабов фронтов и армий. Радиоразведка, как и все прочие виды разведок, ограничивалась только тактической зоной» [46].

Первая попытка создать текстовый электро́механический шифратор в СССР была предпринята в 1923 году в Особом техническом бюро по военным изобретениям специального назначения. Найти какую-либо информацию об этой попытке достаточно трудно.

В 30-е годы образцы советской шифровальной техники создавались под руководством талантливого инженера Ивана Павловича Волоска. Шифрмашины того времени реализовывали наложение случайной последовательности (гаммы) на открытое текстовое сообщение. Даже сейчас такой подход абсолютно современный и при выполнении некоторых условий может обеспечивать гарантированную стойкость шифрования.

В-4, М-100 — одни из первых советских шифрмашин, реализующих шифры гаммирования. В 1938 году началось их серийное производство.

«Шифровальная машина М-100 состояла из трёх основных узлов: клавиатуры с контактными группами, лентопротяжного механизма с трансмиттером и приспособления, устанавливаемого на клавиатуру пишущей машинки, и семи дополнительных блоков. Общий вес комплекта достигал 141 кг. Только одни аккумуляторы для автономного питания электрической части машины весили 32 кг. Тем не менее, данная техника выпускалась серийно и в 1938 году была успешно испытана в боевых условиях во время гражданской войны в Испании (1936–1939 гг.), на Халхин-Голе (1939), во

время советско-финской войны (1939–1940 гг.). Шифрованная связь в этих военных конфликтах осуществлялась в звене Генеральный штаб — Штаб армии» [15].

Позднее появились и более компактные машины. Например, К-37 («Кристалл»), М-101 («Изумруд») и другие. Наряду с шифрами гаммирования применялись и шифры многоалфавитной замены. После войны в СССР использовались такие шифрмашины, как М-105 «Агат», М-125 «Фиалка» и другие.

Широко использовалось и ручное шифрование. Телеграммы отправлялись с помощью лёгких, весом в три килограмма, радиостанций «Север». Или — «Северок», как их ласково называли военные связисты. Эта техника, быстро завоевавшая симпатии наших разведчиков и партизан, выпускалась в блокадном Ленинграде.

Советский связист. Минута отдыха

На машинную шифросвязь в годы войны легла основная нагрузка при передаче секретных телеграмм. И с этой нагрузкой, как уже отмечалось выше, наша шифровальная служба справилась блестяще. Только в 8-м Управлении Красной Армии за период с 1941 по 1945 год было обработано свыше 1,6 миллионов шифротелеграмм.

Не будем забывать про эти успехи.

Известно много примеров героического поведения наших шифровальщиков на войне [46]. Офицеры спецсвязи на грани жизни и смерти, часто с тяжелейшими ранениями, уничтожали шифровальные документы перед приходом врага. В большинстве случаев это было то последнее, что они успевали сделать перед смертью. Советские шифровальщики под страшными пытками не выдавали ни наших кодовых таблиц, ни особенностей использования нашей шифровальной техники.

Без этого личного героизма секретная работа даже самой надёжной шифровальной техники была бы невозможна.

Успехи нашей дешифровальной службы во время войны требуют отдельного исследования. Приведём лишь один пример. Как отмечает в своём интервью Н. Н. Андреев, бывший руководителем 8-го Главного Управления КГБ, с 1992 по 1998 год — президент Академии криптографии РФ, «во время войны мы читали японскую дипломатическую переписку, анализ которой позволил сделать вывод о том, что Япония не намерена начинать военные действия против СССР, что дало возможность перебросить значительные силы на германский фронт» [37].

1.13 После войны

О том, как развивалась отечественная криптография после войны, есть только обрывочные сведения.

Ещё с 1 сентября 1939 года в СССР велась подготовка военных криптографов. С момента образования Спецотдела ВЧК и вплóть до Великой Отечественной войны каких-либо стационарных учебных заведений (кроме краткосрочных курсов и школ) для подготовки профессионалов-криптографов не было. В эти годы основы криптографии преподавались в Военно-инженерной академии имени В. В. Куйбышева и в Военной академии связи в Ленинграде.

В Спецотделе разрабатывались и первые учебники по криптографии. Среди них — пособие «Шифры и их применение» (1933), учебник «Криптография (шифрование и дешифрование)» (авторы: А. И. Копытцев, С. Г. Андреев, С. С. Толстой, Б. А. Аронский, 1939). В эти же годы был подготовлен, а в 1951 году издан учебник «Введение в криптографию» (автор — М. С. Одноробов). Учебник состоял из четырёх частей (общим объёмом — 737 страниц), содержал большое количество примеров, построенных на реальных материалах. Подробнее см. [59].

В 1940 году при ОГПУ была создана криптографическая школа особого назначения (ШОН), которая с началом войны перебазировалась в Уфу.

Именно там, в криптографической школе, преподавал иностранные языки удивительный В. И. Кривош-Неманич. Кстати, языковой подготовке в СССР уделялось очень большое внимание. Считалось, что каждому криптографу следует владеть хотя бы одним иностранным языком.

В начале 1946 года при Высшей школе НКГБ были организованы криптографические курсы. Эти курсы позднее послужили основой подготовки криптографов на закрытом отделении механико-математического факультета МГУ. Такое отделение было создано в 1949 году и просуществовало до 1957 года; его возглавлял Георгий Иванович Пондопуло (1910–1996).

Как вспоминает в своей статье [63] выпускник закрытого отделения мех-мата В. Н. Сачков, «в послевоенные годы в связи с резким увеличением информационного обмена и необходимостью его надёжной защиты, а также с целью повышения эффективности дешифровальной работы возникла потребность существенного усиления криптографических служб ведущих держав. С этой целью в Советском Союзе в 1949 году было создано Главное управление специальной службы (ГУСС), а в США в 1952 году — Агентство национальной безопасности (АНБ). Деятельность как ГУСС, так и АНБ протекала в условиях строгой секретности.»

19 октября 1949 года, в год четырёхсотлетия российской криптографической службы, были созданы Главное управление Специальной службы (ГУСС) и *Высшая школа криптографов* (ВШК). ВШК стала первым и единственным в стране высшим учебным заведением такого профиля. Впоследствии она была преобразована в Высшую школу криптографов 8-го Управления МВД СССР, затем — в Высшую школу криптографов КГБ при Совете Министров СССР, затем — в технический факультет Высшей школы 8-го Главного управления КГБ при СМ СССР, а в 1992 году — в *Институт криптографии, связи и информатики* (ИКСИ). Все эти годы днём рождения ИКСИ и его предшественников считается 19 октября [59].

Позволим небольшое отступление от хронологии. 19 октября — это день рождения ещё одного учебного заведения — Царскосельского лицея, особо почитаемый его первыми выпускниками. Помните у А. С. Пушкина: «Кому ж из нас под старость день лицея // Торжествовать придется одному?». «Торжествовать» пришлось Александру Михайловичу Горчакову (1798–1883), российскому дипломату, министру иностранных дел России с 1856 по 1882 г. Любопытно вспомнить его здесь как человека, на самом высоком уровне причастного к криптографической деятельности. Ведь именно министр иностранных дел в XIX веке контролировал шифровальную службу.

С 40-х годов XX века криптографические задачи стали существенно сложнее и математичнее. В ряде институтов, таких как математи-

ческий институт им. В. А. Стеклова, были созданы закрытые отделы для их решения.

Например, в терминах теории вероятностей и математической статистики решалась задача о критерии открытого текста. Она заключалась в том, чтобы из всевозможных «хаотических» вариантов, которые получаются при дешифровании перехваченного сообщения, выделить единственный верный вариант. Для этого нужно было очень тонко учитывать статистические особенности, присущие неизвестному «правильному» тексту, составленному на том или ином языке.

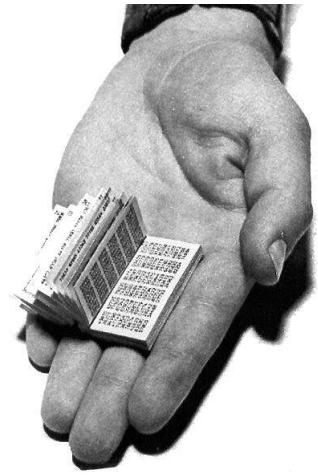
Как отмечается в книге [59], в пятидесятые годы в Высшей школе активизировалась работа по подготовке специалистов–криптографов. В 1955 году в ВШ был создан Учёный совет, стали готовиться научные работы и диссертации. Большой творческий вклад в подготовку научных кадров внёс член-корреспондент Академии наук СССР Владимир Яковлевич Козлов (1914–2007) — им подготовлено более 25 кандидатов и докторов наук [8]. Забота о становлении, развитии и укреплении дневного отделения подготовки криптографов (создано в 1962 году) легла на плечи Ивана Яковлевича Верченко (1907–1995), очень уважаемого студентами преподавателя, декана технического факультета Высшей школы КГБ [62].

По заказам оборонных предприятий криптографические исследования проводились во многих вузах страны (см. например, [5]). Результаты исследований публиковались в закрытых сборниках, о которых и сейчас мало информации. Попытки опубликовать криптографические результаты в открытой печати, предпринимавшиеся отдельными исследователями, были безуспешны.

Так, рукопись А. Д. Закревского «Метод автоматической шифрации сообщений» 1959 года была не принята к печати из-за её высокой секретности; автору пришлось сменить область своих исследований. Спустя 50 лет рукопись была опубликована в журнале «Прикладная дискретная математика» [34].

В целом специалисты–криптографы высоко оценивали работу нашей шифровальной службы послевоенного времени. Американский криптограф Дэвид Кан так описывает криптографические успехи СССР 50–60-х годов: «Россия сама по себе остаётся загадкой, овеянной тайной из тайн. То же самое касается и её средств связи. Одноразовые шифрблокноты обеспечивают надёжную защиту для сообщений российских разведчиков, военных, дипломатов и работников тайной политической полиции. Грамотно сконструированные шифраторы навечно сохраняют в секрете от врагов России её наиболее важную дипломатическую, агентурную и военную переписку. В пери-

од «холодной войны» русские сумели вскрыть шифры американского посольства в Москве. Такие подвиги свидетельствуют об их осведомлённости, базирующейся на глубоком понимании шифровального дела и криптоанализа. Так или иначе русские вознесли достижения своей страны в криптологии до высоты полёта её космических спутников» [39].



1) Шифрмашина М-125 «Фиалка»;
2) Кодовая книга советского разведчика (одноразовый блокнот)

Однако не всё было гладко. С начала 40-х годов и вплоть до 1 октября 1980 года в контрразведке США действовал проект Venona, направленный на дешифрование советских сообщений. Успехи американцев привели к раскрытию нескольких советских разведчиков и утечке секретных сведений. Официально проект был рассекречен в США в 1995 году.

В отличие от других стран, в СССР все исследования по криптографии были сильно засекречены. Почти вплоть до распада Союза «упоминание слова «криптография» в открытой печати даже в невинном контексте часто вызывало в инстанциях резкие возражения и обычно под тем или иным предлогом приводило к запрещению этого упоминания» [45].

До сих пор большинство архивов по истории российских спецслужб (в том числе по истории криптографии) остаются закрытыми. Более того, как отмечается в книге А. Солдатова и И. Бороган [67],

«некоторые архивы, открытые в 1990-е, были вновь засекречены в недавнее время».

1.14 Современность

«Остановиться бы тогда, в конце 80-х годов, снять с глаз тёмные очки и оглядеться вокруг на окружающую действительность. Была же ведь реальная возможность побороться за мировые рынки сбыта наукоёмкой криптографической продукции, программ и алгоритмов, была возможность даже в каком-то смысле стать законодателями криптографической моды. Были и идеи, и отличные молодые специалисты...» — так рассуждает в своей книге «Криптография и свобода» [52] криптограф М. Е. Масленников, выпускник 4-го факультета Высшей школы, с 1979 по 1993 г. сотрудник 8-го Главного управления КГБ, ныне — свободный житель Южной Кореи.

Но возможность была упущена. Тогда, в конце 80-х, криптографическая служба России была не готова к переменам. Как в далёком 1917 году, она переживала тяжёлые времена.

* * *

И всё же те времена миновали. Страна выдержала. Выдержала и обновилась криптографическая служба. Начиная с 90-х годов в России стала развиваться гражданская криптография. Теперь криптография изучается в гражданских вузах, многие статьи и книги по криптографии публикуются в открытой печати, широко используются криптографические средства защиты информации.

Отметим лишь некоторые события последних десятилетий в области российской криптографии и защиты информации.

В 70—80-х годах был разработан блочный шифр ГОСТ 28147-89, ставший с 1990 года государственным стандартом России. Он был рассекречен в 1994 году.

В 1991 году было создано Федеральное агентство правительственной связи и информации при Президенте РФ (ФАПСИ). Упразднено в 2003 году.

В 1992 году в России была создана Академия криптографии. Её первым президентом стал Н. Н. Андреев. Идею создания Академии поддержали академики и член-корреспонденты РАН: В. А. Котельников, Ю. В. Прохоров, В. Я. Козлов, В. К. Левин, Б. А. Севастьянов.

Академия решает задачи государственной важности по обеспечению национальной безопасности и обороноспособности страны [37].

В 90-е и 2000-е годы защиту информации и криптографию начали изучать в гражданских вузах страны.

В 1995 году был издан Указ Президента РФ от 03.04.1995 №334 о мерах по соблюдению законности в области разработки, производства, реализации и эксплуатации шифровальных средств, а также предоставления услуг в области шифрования информации. Согласно Указу был наложен запрет на использование, разработку, производство, реализацию и эксплуатацию шифровальных средств юридическими и физическими лицами без наличия лицензий ФАПСИ.

В настоящее время криптографическая деятельность в России также подлежит обязательному лицензированию, см. Приказ ФСБ России от 09.02.2005 №66 «Об утверждении положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации (положение пкз-2005)», Постановление Правительства РФ от 29.12.2007 №957 «Об утверждении положений о лицензировании отдельных видов деятельности, связанных с шифровальными (криптографическими) средствами».

В 1997 году была образована «Лаборатория Касперского». Её возглавляет Евгений Касперский — выпускник 1987 года технического факультета Высшей школы КГБ СССР. Сейчас компания является одним из мировых лидеров в сфере программных решений для информационной защиты конечных пользователей.

В 1999 году Институт криптографии, связи и информатики отметил своё пятидесятилетие. Ко дню рождения вышла книга [59] об истории ИКСИ, ставшая библиографической редкостью.

В 1999 году была создана Российская криптографическая ассоциация «РусКрипто», аналог международной организации IACR.

В 2003 году на базе ФАПСИ была создана Служба специальной связи и информации Федеральной службы охраны РФ (Спецсвязь России).

В 2010 году шифр ГОСТ был заявлен в качестве участника конкурса Международной организации по стандартизации (ISO) на приобретение статуса Всемирного стандарта шифрования (Worldwide Industrial Encryption Standard).

В апреле 2011 года вступил в силу закон об электронной подписи в РФ, согласно которому каждый гражданин России может завести себе электронную подпись.

В 2013 году в 15-й и соответственно в 12-й раз пройдут российские конференции РусКрипто и SIBECRYPT по криптографии.

* * *

О советской криптографии сняты несколько документально-публицистических фильмов. Среди них:

- «Открытая закрытая связь» (режиссёр Д. Скворцов, 2006) — фильм об истории создания секретной телефонной связи в СССР;
- «Код Верченко» (режиссёр А. Трофимов, 2007) — фильм о советском криптографе И. Я. Верченко (1907–1995), сотруднике Марфинской лаборатории, позднее — декане технического факультета Высшей школы КГБ.

Для более глубокого знакомства с историей криптографии в России можно рекомендовать книги Т. А. Соболевой «История шифровального дела в России» [66], Ю. И. Гольева, Д. А. Ларина, А. Е. Тришина, Г. П. Шанкина «Криптография: страницы истории тайных операций» [22], А. В. Бабаша, Г. П. Шанкина «История криптографии» (часть 1) [9], Д. Кана «Взломщики кодов» [39], С. Сингха «Книга шифров. Тайная история шифров и их расшифровки» [65], главы по истории в книгах А. П. Алфёрова, А. Ю. Зубова, А. С. Кузьмина, А. В. Черёмушкина [7], В. И. Нечаева [53], В. М. Фомичёва [76]. Российской криптографии XIX века и начала XX века посвящены статьи А. В. Бабаша, Ю. И. Гольева, Д. А. Ларина, А. Е. Тришина, Г. П. Шанкина [10]–[14], [23]–[30]. О вкладе Х. Гольдбаха и Ф. Эпинуса в российскую криптографию можно прочитать в работе В. К. Новика [54]. Дополнительно о В. И. Кривоше—Неманиче см. публикации [32], [69]. О криптографии во время Великой Отечественной войны, В. А. Котельникове, марфинской лаборатории можно прочитать в статье Д. А. Ларина [46], сборнике «В. А. Котельников. Судьба, охватившая век» [40], книге К. Ф. Калачёва [38]. О криптографии в XX веке — в сборнике об истории ИКСИ [59], статьях и интервью Н. Н. Андреева [37], [8], В. А. Котельникова [36], В. Н. Сачкова [63], А. Д. Закревского [34], Г. П. Агибалова [5], сайте фонда им. И. Я. Верченко [62], публикациях сайта [1], отчёте лаборатории МГУ [45], книге [52] и других. На английском языке истории российской криптографии посвящены некоторые публикации журнала «Cryptologia», среди них — статьи T. R. Hammant [116]–[118], D. Kahn [121], D. Schimmelpenninck [138], J. Bury [91], Z. J. Kapera [122] и другие.

2. Первое приближение

2.1 Криптографические термины

Для знакомства с криптографической терминологией очень полезным может оказаться словарь терминов сайта `www.cryptofaq.ru`. Приведём лишь основные определения. Лучше сразу запоминать их вместе с английскими терминами.

Открытый текст — секретное сообщение, как правило, это последовательность двоичных битов; *plaintext*.

Шифртекст — результат зашифрования; *ciphertext*.

Зашифрование — процесс преобразования открытого текста в шифрованный с помощью шифра; *encryption*.

Шифр — семейство обратимых отображений множества последовательностей открытых текстов в множество последовательностей шифртекстов. Каждое отображение определяется параметром, называемым *ключом*. Ключ является сменной частью шифра. В зависимости от способа представления открытых текстов различают *поточные* и *блочные* шифры. Поточные шифры осуществляют зашифрование отдельных символов (битов) открытого текста, тогда как блочные шифры обрабатывают блоки фиксированной длины. Соответствующие английские термины — *cipher*; *key*; *stream cipher*; *block cipher*.

Расшифрование — процесс, обратный зашифрованию, реализуемый при известном значении ключа; *deciphering*, decryption.

Дешифрование — процесс получения открытого текста без предварительного знания ключа, взлом; *decryption*.

Криптография — научная и практическая деятельность, связанная с разработкой криптографических средств защиты информации, а также анализом и обоснованием их криптографической стойкости. В отличие от организационных и других способов защиты информации, под криптографическими понимаются такие, которые используют *математические методы преобразования* информации; *cryptography*.

Криптоанализ — научная и практическая деятельность по исследованию криптографических алгоритмов с целью получения обоснованных оценок их криптографической стойкости; *cryptanalysis*.

Криптология — понятие, объединяющее криптографию и криптоанализ; *cryptology*.

Для более глубокого знакомства с основами криптологии можно порекомендовать книги [7], [75], [2], [61], [80], [50], [21] и другие.

2.2 Правило стойкости

Голландский криптограф Огюст Керкгоффс (1835–1903) в своей монографии «Военная криптография» впервые сформулировал *правило стойкости* шифрсистемы, которое и сейчас остаётся актуальным:

1) весь механизм преобразований шифрсистемы не должен требовать секретности; надо полагать, что он известен злоумышленнику;

2) надёжность шифрсистемы должна определяться только неизвестным значением секретного ключа.

Таким образом, Керкгоффс впервые разделил понятия криптографического *алгоритма* и *ключа*, чётко определив их роли. Алгоритм — это «долгоиграющий» элемент системы. Он тщательно разрабатывается и меняется в редких случаях, его раскрытие не снижает стойкости системы. Ключ — «легко сменяемый» элемент шифрсистемы, который и обеспечивает её надёжность. Он предназначен для частого модифицирования в соответствии с некоторым порядком.

Правило Керкгоффса стимулировало появление более качественных алгоритмов шифрования. Во-первых, потому что была осознана необходимость испытания шифрсистемы в условиях, благоприятных для злоумышленника. Во-вторых, на разработку алгоритма теперь можно потратить больше времени и средств, так как при взломе системы её надёжность легко восстанавливается заменой ключа.

В XIX веке Керкгоффсом заложен первый элемент стандартизации в криптографии, поскольку он выступил за разработку *открытых способов* криптографических преобразований. Их стойкость определяется не секретностью, а математическими характеристиками использованных методов. Керкгоффс опередил своё время. В реальности открытые стандарты в криптографии стали появляться лишь в конце XX века.

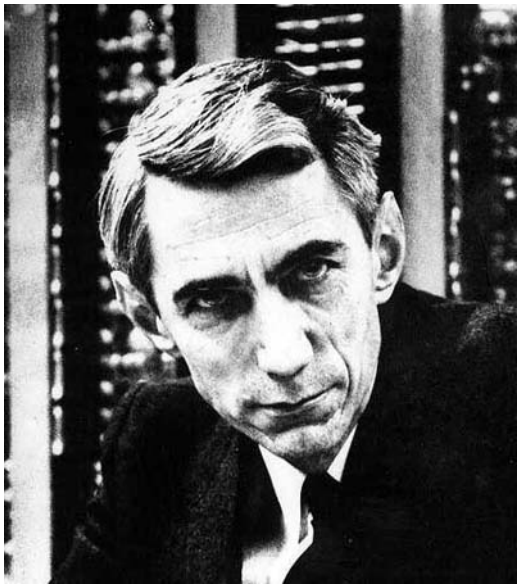
2.3 Принципы Шеннона

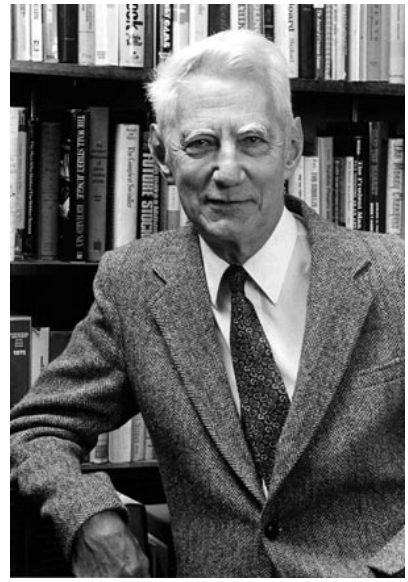
Как наука криптография возникла после фундаментальных работ американского математика и электротехника Клода Шеннона (1916–

2001). В его работах «Математическая теория связи» и «Теория связи в секретных системах» (1949) содержится обобщение большого опыта создания шифров, накопленного до него, и разрабатывается полноценный математический аппарат для криптографических задач [79]. Напомним, что многие результаты Шеннона независимо были получены В. А. Котельниковым ещё в 1941 году, но были строго засекречены.

Анализируя ранее существовавшие шифры, Клод Шеннон пришёл к выводу, что большинство из них (даже самые сложные шифры) сконструированы из простых типичных компонент, осуществляющих *замену* и *перестановку*. Более глубокое понимание того, как должны строиться надёжные шифры, привело Шеннона к выделению двух общих принципов построения криптографических преобразований: *перемешивание* и *рассеивание* (*confusion and diffusion*).

Перемешивание означает усложнение всевозможных связей между битами открытого и шифрованного текстов. *Рассеивание* подразумевает распространение влияния одного бита открытого текста на большое число битов шифрованного текста. Как реализовать эти принципы в конкретной системе, разработчики каждый раз решают заново: в криптографии, как и в других науках, нет универсальных приёмов.





Клод Шеннон

Клод Шеннон впервые математически строго сформулировал вопросы о *теоретической стойкости* шифров. А именно, насколько устойчивой является шифрсистема для злоумышленника, обладающего неограниченными ресурсами (временем, памятью и т. д.)?

Какой минимальный объём шифртекста злоумышленнику необходимо перехватить, чтобы однозначно восстановить по нему исходный открытый текст? Этот объём в среднем, т. е. длина шифртекста, называется *расстоянием единственности* шифра. Клод Шеннон показал, что расстояние единственности прямо пропорционально длине ключа и обратно пропорционально избыточности исходного открытого текста.

А существуют ли шифрсистемы, в которых злоумышленник не получит никакой информации, сколько бы он ни перехватывал шифртекст? Ответ оказался неожиданным: да! Шифрсистемы, обладающие таким свойством, называются *совершенно секретными*.

Шеннон доказал, что необходимое и достаточное условие совершенной секретности состоит в том, чтобы условная вероятность получить шифртекст C при условии, что он соответствует некоторому открытому тексту P, не зависела от выбора текста P. Эта вероятность должна быть равна вероятности просто получить шифртекст C.

К числу таких совершенно секретных систем относится, например, шифр «одноразовый блокнот» (или *шифр Вернама*). В этом шифре открытый текст $P = (p_1, \ldots, p_n)$ переводится в шифртекст $C = (c_1, \ldots, c_n)$ путём наложения на него секретной гаммы $K = (k_1, \ldots, k_n)$, а именно

$$C = P + K = (p_1 + k_1, \ldots, p_n + k_n).$$

Сложение выполняется по модулю некоторого целого числа. Например, если открытый текст (0110) сложить по модулю 2 с секретной гаммой (1011), то шифртекстом будет вектор (1101). Нетрудно понять, что вероятность получить такой шифртекст C *не зависит* от текста P. Шифртекст C в равной степени может соответствовать *любому* открытому тексту.

Но криптография на результатах о совершенной секретности не остановилась. Наоборот! Совершенно секретные системы очень редко используются на практике, так как длина секретного ключа (гаммы) слишком велика. Она должна совпадать с длиной открытого текста, что практически невыполнимо при современных объёмах информации. Так возникает задача построения шифрсистем, в которых

надёжность зашифрования больших объёмов информации обеспечивалась бы относительно малым размером ключа. Например таких, чтобы гигабайты информации можно было шифровать с помощью нескольких килобайтов ключа. Конечно, такие системы заведомо не будут теоретически стойкими, но их *практическая* стойкость может оказаться достаточно высокой.

2.4 Виды криптографии

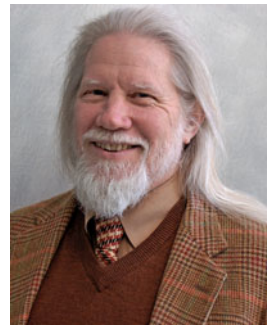
Условно говоря, вся криптография делится на симметричную криптографию (с закрытым ключом) и асимметричную (с открытым ключом). Различаются они по типу используемых шифрсистем.

Симметричная шифрсистема — система шифрования, в которой ключи зашифрования и расшифрования совпадают, либо легко определяются один по другому. Перед использованием симметричной шифрсистемы абонентам *необходимо* заранее договариваться о едином секретном ключе.

Асимметричная шифрсистема — система шифрования, в которой используются ключи двух видов — открытые ключи и секретные ключи. *Открытый ключ* применяется в процессе зашифрования и, как правило, является общедоступным. *Секретный ключ* используется в процессе расшифрования сообщения и должен храниться в тайне получателем сообщения. Криптографическая стойкость асимметричной системы определяется трудоемкостью, с которой злоумышленник может вычислить секретный ключ исходя из знания открытого ключа и другой дополнительной информации о шифрсистеме. Основным преимуществом асимметричной шифрсистемы является то, что абонентам *не нужно* заранее договариваться об общем секретном ключе. На практике активно используются шифрсистемы обоих видов. У каждого из них есть свои преимущества и недостатки.

До 1976 года все шифрсистемы относились к симметричной криптографии. Но неожиданно всё изменилось. Асимметричные криптосистемы — изобретение американских криптографов У. Диффи, М. Хеллмана и Р. Меркля — оказались основаны совсем на иных принципах. Здесь нашла своё применение математическая теория сложных задач. Оказалось возможным строить такие математические функции, которые «легко» вычисляются, а обращаются «очень трудно»: для этого необходимы нереальные вычислительные ресур-

сы и время. Однако если для подобной функции известна некоторая дополнительная информация (её часто называют «лазейкой»), то обратить функцию можно тоже «легко». Такие функции называются *односторонними* или *односторонними с лазейкой*.



Р. Меркль, М. Хеллман, У. Диффи — 1977 год и современность

Используя одностороннюю функцию f с лазейкой s, абонент A может построить асимметричную шифрсистему, например так. Алгоритм вычисления функции f объявить открытым ключом и сделать общедоступным. «Лазейку» s назвать своим секретным ключом и сохранить в тайне. Любой другой абонент, скажем B, используя открытый ключ, может зашифровать секретное сообщение m для абонента A. Для этого он вычисляет значение $f(m)$, которое и передает абоненту A по открытому каналу связи. Злоумышленник, перехватив значение $f(m)$, не может восстановить сообщение m, так как задача обращения функции f «очень трудна». Только абонент A может

справиться с задачей обращения, так как ему известна «лазейка» s. Вычисляя $f^{-1}(m, s)$, он «легко» восстанавливает сообщение m.

До сих пор существование односторонних функций строго не доказано. Но имеется несколько функций-кандидатов, обладающих свойствами односторонних функций. Они используются для построения современных асимметричных шифрсистем.

Среди них, например, функция произведения двух простых чисел, скажем, p и q. Обратить такую функцию, т. е. решить *задачу факторизации* (разложения на множители числа $n = pq$), очень сложно, если p и q достаточно большие, например имеют в десятичной записи не менее 100 знаков. Дополнительной информацией для быстрого обращения может служить, например, один из делителей или значение функции Эйлера от числа n. На основе функции произведения двух простых чисел строится криптосистема RSA.

Другая функция-кандидат — возведение в степень по модулю. Пусть n и g — целые числа такие, что $2 \leqslant g \leqslant n-1$. Пусть функция f сопоставляет произвольному целому числу m, $1 \leqslant m \leqslant n-1$ значение $x = g^m \mod n$. Восстановить m по значениям x, g и n — это тоже очень трудная задача, которая называется *задачей дискретного логарифмирования*. Криптосистема ЭльГамаля строится на её основе.

Краткое описание двух самых известных асимметричных шифрсистем приводится в следующем разделе, далее будут рассматриваться только методы симметричной криптографии.

Основательное знакомство с методами асимметричной криптографии (и не только) можно получить, прочитав книгу Б. Я. Рябко и А. Н. Фионова «Основы современной криптографии и стеганографии» [61].

2.5 Криптосистемы RSA и ElGamal

RSA — самая популярная криптосистема с открытым ключом. Она основана на сложности задачи факторизации числа и была предложена в 1978 году Рональдом Ривестом, Ади Шамиром и Леонардом Адлеманом. Приведём её краткое описание. Пусть

p, q — большие простые числа;

$n = pq$;

$\varphi(n) = (p-1)(q-1)$ — функция Эйлера числа n;

e, d — натуральные числа такие, что e, d взаимно просты с $\varphi(n)$ и выполняется $ed \equiv 1 \mod \varphi(n)$.

Первый абонент (Алиса) формирует свой открытый ключ $K_{pub} = \{n, e\}$ и секретный ключ $K_{priv} = \{p, q, d\}$. Открытый ключ Алисы сообщается всем абонентам. Любой другой абонент (Боб) может зашифровать секретное сообщение m для Алисы, воспользовавшись её открытым ключом. Получив шифртекст c, Алиса расшифрует его с помощью своего секретного ключа.

Зашифрование $c = E(m, K_{pub}) = m^e \mod n$.

Расшифрование $m = D(c, K_{priv}) = c^d \mod n$.

Корректность расшифрования, т. е. того, что после применения функции D к шифртексту c будет получено именно сообщение m, несложно доказать, используя теорему Эйлера.

А. Шамир, Р. Ривест, Л. Адлеман

Задача 1. Функция Эйлера. Пусть n — натуральное число, разложение которого на простые множители имеет вид

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k},$$

где $p_1, p_2, \ldots, p_k$ — простые числа. Пусть $\varphi(n)$ — количество натуральных чисел, не превышающих n и взаимно простых с n (это и есть *функция Эйлера* числа n). Докажите, что

$$\varphi(n) = n \cdot (1 - \frac{1}{p_1}) \cdot (1 - \frac{1}{p_2}) \cdot \ldots \cdot (1 - \frac{1}{p_k}).$$

Задача 2. Теорема Эйлера. Докажите, что если любые целые числа a и n взаимно просты, то $a^{\varphi(n)} \equiv 1 \mod n$.

Задача 3. Малая теорема Ферма. Докажите, что если целое число a не делится на простое число p, то $a^{p-1} \equiv 1 \mod p$.

Задача 4. Корректность RSA. Докажите, что при расшифровании сообщения c с помощью секретного ключа Алисы будет восстановлено исходное сообщение m.

Для стойкости криптосистемы RSA числа p и q рекомендуется выбирать достаточно большими (их длина в десятичной записи должна быть не менее 100 знаков). Они не должны быть близкими друг к другу, но и не должны сильно друг от друга отличаться. Необходимо, чтобы $gcd(p-1, q-1)$ был небольшой, равный, например, двум. Кроме того, числа p и q должны быть сильно простыми.

Число r называется *сильно простым*, если

а) число $r+1$ имеет большой простой делитель;

б) число $r-1$ имеет большой простой делитель s;

в) число $s-1$ имеет достаточно большой простой делитель.

При несоблюдении указанных рекомендаций по выбору p и q, число n может быть эффективно разложено на множители.

Для повышения скорости шифрования с помощью RSA рекомендуют выбирать малую шифрующую экспоненту e. Однако если число e достаточно мало, то есть риск, что найдутся e абонентов с тем же значением шифрующей экспоненты. Тогда, если некоторое секретное сообщение m было зашифровано и передано всем e абонентам, то злоумышленник может однозначно восстановить его, решая систему

$$x \equiv c_1 \mod n_1,\ x \equiv c_2 \mod n_2,\ \ldots,\ x \equiv c_e \mod n_e,$$

где $x = m^e$. Китайская теорема об остатках позволяет ему это сделать.

Задача 5. Китайская теорема об остатках*. Пусть $n_1, n_2, \ldots, n_s$ — попарно взаимно простые числа. Пусть $M = n_1 \cdot \ldots \cdot n_s$. Докажите, что для любых целых чисел $a_1, a_2, \ldots, a_s$ сравнения

$x \equiv a_1 \mod n_1$

$x \equiv a_2 \mod n_2$

$\ldots$

$x \equiv a_s \mod n_s$

имеют в интервале $0 \leqslant x \leqslant M-1$ единственное решение вида

$$x = \sum_{j=1}^{s} a_j \cdot N_j \cdot M_j \mod M,$$

где $M_j = M/n_j$ и $N_j = M_j^{-1} \mod n_j$.

При относительно малом значении расшифровывающей экспоненты d, а именно при $d < \sqrt[4]{n}/3$, существует эффективный метод восстановления секретного ключа d (метод Винера). Он связан с разложением вещественного числа e/n в цепную дробь. Из установленного ограничения на d следует, что знаменатель одной из подходящих дробей в этом разложении будет совпадать с секретной экспонентой d. При этом достаточно просмотреть около $O(\log n)$ таких дробей.

Задача 6. Криптоанализ RSA. Используя метод Винера, восстановите расшифровывающую экспоненту d, если известно, что выполняется $d < \sqrt[4]{n}/3$ и $n = 9\,449\,868\,410\,449$, $e = 6\,792\,605\,526\,025$.

Криптосистему **ElGamal** предложил в 1985 году Тахер Эль-Гамаль.

Т. Эль-Гамаль

Кратко опишем её. Пусть

p — большое простое число,

$\mathbb{Z}_p$ — поле вычетов по модулю p,

g — целое число, $2 \leqslant g \leqslant p-2$, такое, что числа $g^0, g^1, \ldots, g^{p-2}$, взятые по модулю p, попарно различны, т. е. g — порождающий элемент мультипликативной группы $\mathbb{Z}_p$.

Алиса выбирает целое число a такое, что $2 \leqslant a \leqslant p-2$, и вычисляет $d = g^a \mod p$. Открытый ключ Алисы — это $K_{pub} = \{p, g, d\}$, её секретный ключ — $K_{priv} = \{a\}$.

Пусть s — случайное число, которое выбирает Боб, $2 \leqslant s \leqslant p-2$. Тогда зашифрование и расшифрование секретного сообщения m от Боба Алисе осуществляется так.

Зашифрование $c = E(m, K_{pub}) = (r, e)$, где

$$r = g^s \mod p, \quad e = m \cdot d^s \mod p.$$

Расшифрование $m = D(c, K_{priv}) = e \cdot r^{p-1-a} \mod p$.

В асимметричной криптографии возникает много теоретико-числовых задач. Например, определение простоты числа, эффективное разложение числа на множители при выполнении ряда условий и др. Для их решения необходимо основательное погружение в теоретико-числовые методы криптографии. Отличной книгой по данному направлению служит учебное пособие М. М. Глухова, И. А. Круглова, А. Б. Пичкура и А. В. Черёмушкина [21], вышедшее в 2011 году.

Задача 7. Проверка простоты числа. Применяя любой из методов проверки простоты числа, определите, являются ли числа простыми:

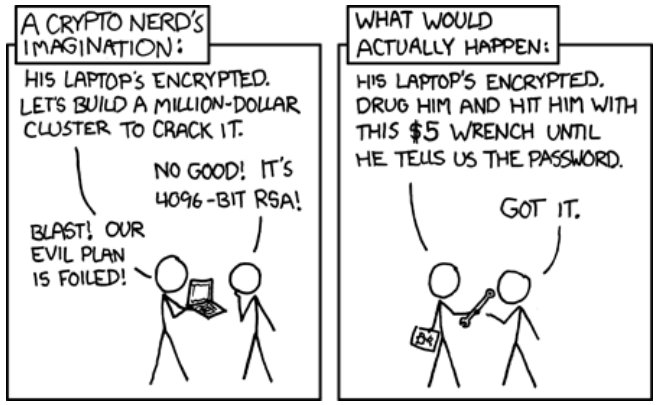
а) 7 079; б) 12 827; в) 50 819; г) 13 667 809;
д) 50 821; е) 162 401; ж) 252 601; з) 1 111 111 111 111 111 111.

Задача 8. Факторизация числа. Применяя (P-1)-метод Полларда, найдите разложение числа $n = 719\,088\,091$ на простые множители. При каких значениях порога показательной гладкости это возможно сделать?

2.6 Криптографические протоколы

Всё большее значение в криптографии приобретает разработка не отдельных шифрсистем, а сложных протоколов взаимодействия абонентов. Под *криптографическим протоколом* понимается описание распределённого алгоритма, в процессе выполнения которого два (или более) участника последовательно выполняют определённые действия и обмениваются сообщениями. При этом цель выполнения протокола криптографическая, т. е. связана с передачей и защитой той

или иной информации пользователей. В ходе выполнения протокола пользователи применяют различные шифрсистемы, но шифрование является лишь составной частью системы безопасности, и во многих случаях не самой важной. При неправильной организации взаимодействия пользователей, при выборе плохих параметров работы криптографических алгоритмов даже самая надёжная система шифрования не защитит вас от злоумышленника. Всё чаще стали разрабатываться атаки, направленные не на взлом шифрсистемы, а на грамотное использование слабостей криптографического протокола, или (что ещё более опасно!) ошибок в его реализации. Часто применяются и совсем ненаучные методы. Разрешите привести известную карикатуру о том, как «иногда в реальности» происходит «взлом» криптографического протокола, например 4096-битного RSA[1].



Но мы будем рассматривать только честные математические методы криптографии и криптоанализа. Для этого нам понадобятся строгие математические определения и результаты. Прежде всего начнём с булевых функций, ведь именно с их помощью можно описать многие составные элементы шифра. Изучая математические свойства конкретных булевых функций, можно оценивать криптографическую стойкость шифра, в котором они используются. Известно много примеров, когда неудачный выбор булевых функций приводил к успешным атакам на шифр (см., например, атаки на шифры DES (1993), Grain (2011) и другие). Поэтому в области булевых функций очень важны теоретические исследования.

[1]Пусть это будет и небольшим упражнением по английскому языку.

3. Булевы функции. Комбинаторный подход

Материал, представленный в этой главе, предполагает знакомство читателя с основными методами комбинаторики. Познакомиться с ними можно, например, решив серию задач из приложения к данному пособию.

3.1 Определение

Пусть множество $\{0, 1\}$ обозначается через $\mathbb{Z}_2$. Множество всех двоичных векторов $x = (x_1, \ldots, x_n)$ длины n обозначим через $\mathbb{Z}_2^n$.

Произвольная функция f из множества $\mathbb{Z}_2^n$ в множество $\mathbb{Z}_2$ называется *булевой функцией от n переменных*. Свое название булевы функции получили в честь известного английского математика и философа Джорджа Буля (1815–1864).

Каждая булева функция может быть задана с помощью *таблицы истинности*, т. е. таблицы вида

$x_1 \ldots x_n$	$f(x_1, \ldots, x_n)$
$0 \ldots 0$	$*$
$\vdots$	$\vdots$
$1 \ldots 1$	$*$

,

где в первом столбце приводятся всевозможные векторы длины n, а во втором столбце вместо $*$ указываются конкретные значения булевой функции на этих векторах. Здесь и далее будем считать, что все векторы в первом столбце лексикографически упорядочены.

Задача 9. Чему равно число булевых функций от n переменных?

Приведём несколько простейших булевых функций от двух переменных. Часто их называют *бинарными операциями*, а участвующие в них переменные — *операндами*. Каждая из операций имеет своё название.

	AND	OR	XOR	NAND	NOR
x_1x_2	$x_1 \cdot x_2$	$x_1 \vee x_2$	$x_1 \oplus x_2$	$x_1 \mid x_2$	$x_1 \downarrow x_2$
0 0	0	0	0	1	1
0 1	0	1	1	1	0
1 0	0	1	1	1	0
1 1	1	1	0	0	0

По-русски эти функции называются соответственно: И, ИЛИ, исключающее ИЛИ, штрих Шеффера, стрелка Пирса.

Бинарные операции XOR и AND иногда называют *сложением* и *умножением* по модулю 2. Знак $\cdot$ для умножения часто опускается. А вот ещё несколько функций:

x	$\overline{x}$
0	1
1	0

			Эквивалентность	Импликация
x_1x_2	0	1	$x_1 \sim x_2$	$x_1 \to x_2$
0 0	0	1	1	1
0 1	0	1	0	1
1 0	0	1	0	0
1 1	0	1	1	1

.

Нетрудно доказать, что операции $\cdot$, $\vee$, $\oplus$, $\sim$, $|$, $\downarrow$ *коммутативны*, т. е. операнды в них можно переставлять местами. Операции $\cdot$, $\vee$, $\oplus$ являются *ассоциативными*. Например, для конъюнкции это означает, что $(x \cdot y) \cdot z = x \cdot (y \cdot z)$. Аналогичные тождества выполняются и для операций $\vee$, $\oplus$.

Пользуясь ассоциативностью некоторых операций, можно свободнее раскрывать скобки. При этом необходимо учитывать приоритеты операций. Самый высокий приоритет имеет унарная операция отрицания, $\overline{x}$, на втором месте — операция умноже́ния, за́тем следуют остальные операции.

Задача 10. Докажите, что имеют место *законы дистрибутивности*:
а) $(x \vee y)z = xz \vee yz$;
б) $(x \cdot y) \vee z = (x \vee z) \cdot (y \vee z)$;
в) $(x \oplus y)z = xz \oplus yz$.

Каждая булева функция может иметь несколько представлений, полученных с помощью суперпозиции различных булевых функций. Все эти представления (или *формулы*) для булевой функции называются *эквивалентными*. Например, функция $f(x_1, x_2, x_3) = (x_1 \vee x_2)x_3$ имеет следующие эквивалентные представления $f(x_1, x_2, x_3) = x_1x_3 \vee x_2x_3 = x_1x_2x_3 \oplus x_1x_2 \oplus x_2x_3$.

Задача 11. Докажите равенства:

а) $x\overline{x} = 0$ (закон противоречия);

б) $x \vee \overline{x} = 1$ (закон исключенного третьего);

в) $\overline{\overline{x}} = x$ (закон снятия двойного отрицания);

г) $\overline{x \cdot y} = \overline{x} \vee \overline{y}$, $\overline{x \vee y} = \overline{x} \cdot \overline{y}$ (законы де Моргана).

Задача 12. Докажите равенства:

а) $x \cdot x = x$; б) $x \vee x = x$;

в) $x \cdot 1 = x$; г) $x \cdot 0 = 0$;

д) $x \vee 1 = 1$; е) $x \vee 0 = x$;

ж) $x \oplus x = 0$; з) $x \oplus 0 = x$;

и) $\overline{x} = x \oplus 1$; к) $x \vee y = xy \oplus x \oplus y$;

л) $x \to y = \overline{x} \vee y$; м) $x \sim y = \overline{x \oplus y}$;

н) $x \oplus y = \overline{x}y \vee x\overline{y}$; о) $x \oplus y = (\overline{x} \vee \overline{y})(x \vee y)$;

п) $x|y = \overline{xy}$; р) $x \downarrow y = \overline{x \vee y}$.

Подфункцией булевой функции f называется функция от меньшего числа переменных, полученная фиксацией значений некоторых её переменных. Например, подфункция $g(x_1, x_3) = f(x_1, 0, x_3, 1, 0)$ получена из функции f фиксацией трёх переменных. Подфункцию булевой функции принято обозначать как $f^{a_1,\ldots,a_{n-k}}_{i_1,\ldots,i_{n-k}}$, где $i_1, \ldots, i_{n-k}$ — номера фиксированных переменных и $a_1, \ldots, a_{n-k}$ — фиксированные значения. При этом $f^{a_1,\ldots,a_{n-k}}_{i_1,\ldots,i_{n-k}}$ — функция от k переменных.

3.2 Алгебраическая нормальная форма

Любую булеву функцию можно задать формулой от её переменных, в которой используются только операции сложения и умножения, а также константы 0 и 1. Такое представление называется *алгебраической нормальной формой* функции[1] (АНФ).

Например, в таком виде представлены булевы функции

$$g(x_1, x_2) = x_1 \oplus x_2 \oplus 1,$$

$$h(x_1, x_2, x_3) = x_1x_2x_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3.$$

Докажем корректность и однозначность та́кого представления.

[1] В отечественной литературе АНФ часто называют *полиномом Жегалкина*.

Теорема 1. *Каждая булева функция от n переменных единственным образом представляется в виде АНФ*:

$$f(x_1,\dots,x_n)=\Big(\bigoplus_{k=1}^{n}\bigoplus_{i_1,\dots,i_k} a_{i_1,\dots,i_k}x_{i_1}\cdot\ldots\cdot x_{i_k}\Big)\oplus a_0,$$

где при каждом k все индексы $i_1,\dots,i_k$ различны и параметры a_0, $a_{i_1,\dots,i_k}$ принимают значения 0 *или* 1.

Доказательство. Определим число различных полиномов Жегалкина от n переменных. Каждый коэффициент полинома Жегалкина независимо принимает значение 0 или 1. Поскольку число коэффициентов равно числу всех сочетаний из n по k, а именно числу 2^n (см. задачу 73), то число различных полиномов Жегалкина равно 2^{2^n}. Согласно задаче 9 это число совпадает с числом всех булевых функций от n переменных.

Осталось показать, что различные полиномы Жегалкина определяют различные булевы функции. Действительно, предположим, что некоторая булева функция определяется двумя различными полиномами P_1 и P_2. Тогда их сумма $P_1\oplus P_2$ является ненулевым полиномом, но соответствует тождественно нулевой булевой функции. Покажем, что такое невозможно. Так как $P_1\oplus P_2$ — ненулевой полином, то он содержит хотя бы одно слагаемое. Пусть $x_{i_1}\dots x_{i_k}$ — слагаемое наименьшей длины, которое встречается в полиноме $P_1\oplus P_2$. Тогда на векторе переменных x таком, что $x_i=1$, если $i\in\{i_1,\dots,i_k\}$, и $x_i=0$ при остальных значениях i, полином $P_1\oplus P_2$ принимает значение 1.

Таким образом, *каждая* булева функция имеет единственное представление в виде полинома Жегалкина. □

Для того чтобы найти представление функции в АНФ, часто используют *метод неопределённых коэффициентов*.

Например, для функции $f(x_1,x_2)=x_1\to x_2$ найдем представление в виде $f(x_1,x_2)=a_{12}x_1x_2\oplus a_1x_1\oplus a_2x_2\oplus a_0$, где коэффициенты a_0, a_1, a_2, a_{12} нам пока неизвестны. Метод состоит в их последовательном вычислении путём подстановки значений функции: $a_0=f(00)$, $a_1=f(10)\oplus a_0$, $a_2=f(01)\oplus a_0$, $a_{12}=f(11)\oplus a_1\oplus a_2\oplus a_0$. А именно получаем $a_0=1$, $a_1=1$, $a_2=0$, $a_{12}=1$. Значит, $f(x_1,x_2)=x_1x_2\oplus x_1\oplus 1$.

Можно получить это представление и по-другому. Например, воспользовавшись доказанными ранее тождествами. Используя пункты

л), к), и), в), ж) задачи 12, получаем цепочку равенств: $f(x_1, x_2) = x_1 \to x_2 = \overline{x_1} \vee x_2 = \overline{x_1}x_2 \oplus \overline{x_1} \oplus x_2 = (x_1 \oplus 1)x_2 \oplus (x_1 \oplus 1) \oplus x_2 = x_1x_2 \oplus x_2 \oplus x_1 \oplus 1 \oplus x_2 = x_1x_2 \oplus x_1 \oplus 1$.

Задача 13. Представьте следующие функции в АНФ:

а) $f(x_1, x_2) = x_1 \sim x_2$; б) $f(x_1, x_2) = x_1|x_2$;
в) $f(x_1, x_2, x_3) = x_1 \vee x_2 \vee x_3$; г) $f(x_1, x_2, x_3) = (x_1 \sim x_2) \sim x_3$;
д) $f(x_1, x_2, x_3) = x_1x_2x_3$; е) $f(x_1, x_2, x_3) = (x_1 \vee x_2)x_3 \to x_2$.

Алгебраической степенью булевой функции f называется число переменных в самом длинном слагаемом её АНФ. Степень функции обозначается через $deg(f)$. Например, для функций, приведённых в начале параграфа, $deg(g) = 1$, $deg(h) = 3$.

Задача 14. Упростите и найдите алгебраическую степень функции:

а) $x_1x_2(x_1x_2x_3 \oplus x_1) \oplus x_3(x_1x_2 \oplus 1)$;
б) $(x_1x_2 \oplus x_3x_4 \oplus x_1)(x_1 \oplus x_3)(x_2 \oplus x_4)$;
в) $x_1(x_1x_2x_3 \oplus x_2x_3 \oplus x_1 \oplus 1)$;
г) $x_1(x_2x_3x_4 \oplus x_2x_3 \oplus x_1 \oplus 1)$;

Как видно из примеров задачи 14 определить степень произведения двух булевых функций довольно трудно. Например, если $deg(f) = k > 0$, $deg(g) = \ell > 0$, то степень функции $f \cdot g$ может оказаться любым числом, не превосходящим $k\ell$. Степень может даже оказаться равной нулю! И к этой особенности булевых функций нужно привыкнуть.

Булева функция степени 1 называется *аффинной*. Если, кроме того, она принимает значение 0 на нулевом векторе, то функция называется *линейной*. Если степень булевой функции равна $2, 3, \ldots$, то функция называется *квадратичной*, *кубической* и т. д.

Задача 15. Определите число линейных и аффинных булевых функций от n переменных. Сколько существует булевых функций от n переменных степени не выше k?

Задача 16. Покажите, что булева функция f линейна тогда и только тогда, когда $f(x \oplus y) = f(x) \oplus f(y)$ для любых x, y. Аналогично проверьте, что функция аффинная тогда и только тогда, когда $f(x \oplus y) = f(x) \oplus f(y) \oplus f(0)$.

3.3 Векторные булевы функции

Функция вида $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ называется *векторной булевой функцией*, или (n, m)-*функцией*. Интерес к таким функциям в криптографии особенно высок, так как они определяют нелинейные компоненты блочных и поточных шифров.

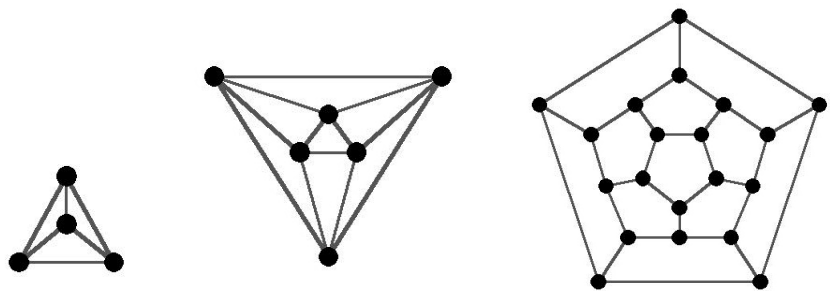
Векторную булеву функцию F от n переменных всегда можно представить в виде $F(x) = (f^1(x), \ldots, f^m(x))$, где f^j — обычная булева функция от n переменных. Функции f^j называются *координатными*. Если каждую функцию f^j задать в АНФ, то получим *алгебраическую нормальную форму* векторной функции F, т. е. представление в виде

$$F(x_1, \ldots, x_n) = \Big(\bigoplus_{k=1}^{n} \bigoplus_{i_1, \ldots, i_k} b_{i_1, \ldots, i_k} x_{i_1} \cdot \ldots \cdot x_{i_k} \Big) \oplus b_0,$$

где при каждом k все индексы $i_1, \ldots, i_k$ различны и параметры b_0, $b_{i_1, \ldots, i_k}$ принимают значения из множества $\mathbb{Z}_2^m$. *Алгебраической степенью* векторной булевой функции называется максимальная из степеней её координатных функций; она обозначается $deg(F)$. Под *минимальной степенью* функции понимается минимальная из этих степеней.

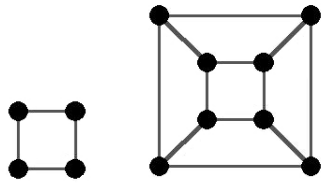
3.4 Булев куб

Напомним, что *графом* G называется пара множеств (V, E), где V — *множество вершин* и E — *множество ребер*, т. е. подмножество неупорядоченных пар элементов из V. Граф имеет наглядную интерпретацию. На рисунке представлено несколько примеров графов с числом вершин 4, 6 и 20 соответственно.

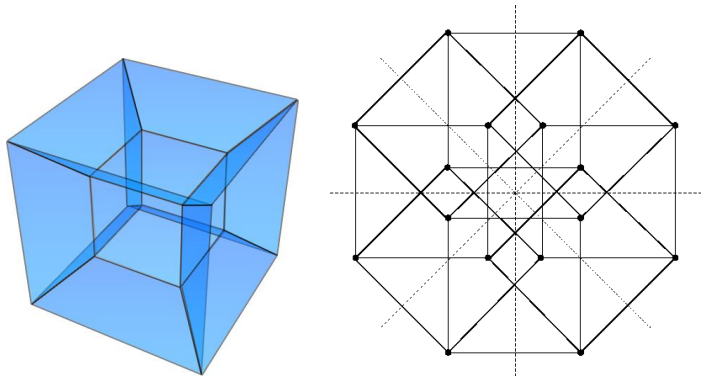


Например, для первого графа множества вершин и ребер имеют вид $V = \{1, 2, 3, 4\}$, $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$.

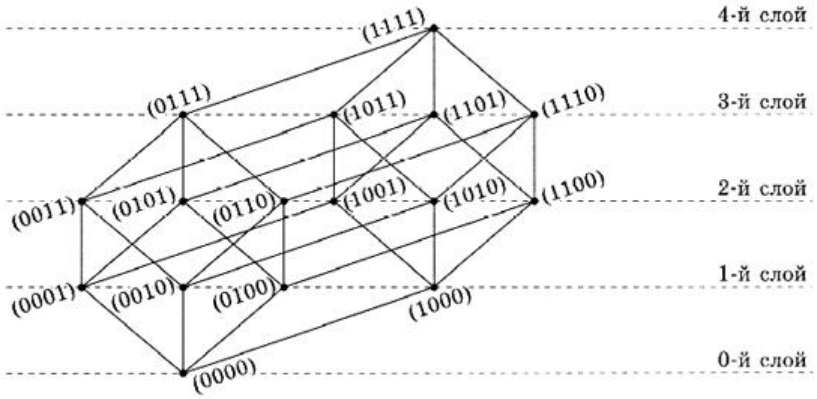
Нас будет интересовать особый граф E^n, который называется *булевым кубом*. Его вершинами являются все двоичные векторы длины n, т. е. $V = \{(x_1, \ldots, x_n) : x_i \in \mathbb{Z}_2\}$, а ребрами соединяются только те векторы, которые различаются ровно в одной координате. Число n называется *размерностью* булева куба. Например, двумерный и трёхмерный кубы выглядят так.



Четырёхмерный булев куб выглядит чуть сложнее:



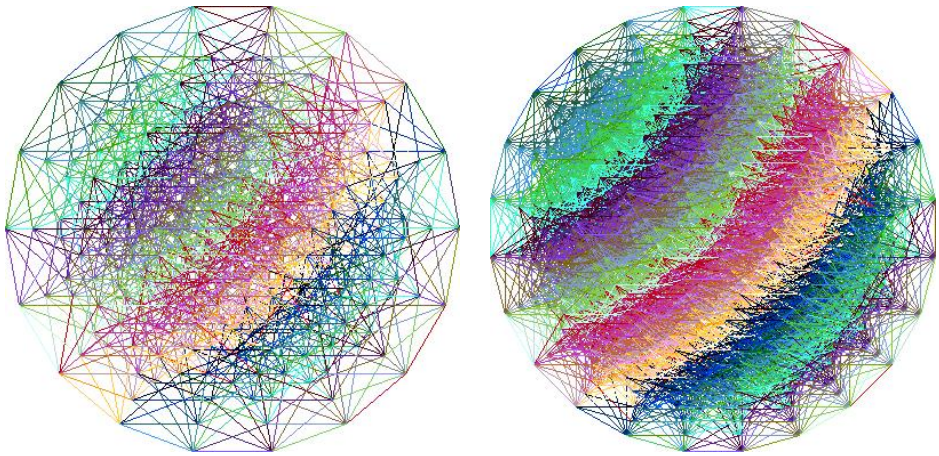
Для того чтобы лучше разобраться в его устройстве, достаточно внимательно изучить следующую картинку.



Несложно заметить, что булев куб E^4 можно получить из двух булевых кубов E^3 таким образом: ко всем векторам первого и второго кубов добавляем новую координату (например, слева). Эту координату полагаем равной нулю для первого куба и единице для второго. После этого соединяем ребрами соответствующие вершины первого и второго кубов. Таким же способом можно построить любой булев куб E^n из двух кубов E^{n-1}.

Векторы булева куба часто располагают по слоям. Все векторы, имеющие ровно k ненулевых координат, образуют *k-й слой* куба. Нетрудно понять, что мощность k-го слоя n-мерного булева куба равна числу сочетаний из n по k, а именно $C_n^k = \left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$.

При больших n булев куб, скорее, напоминает шар. Этот эффект уже хорошо заметен на кубах размерности 9 и 12:



Для наглядности ребра этих кубов раскрашены в разные цвета. Упрощенно можно считать, что ребра одного цвета соединяют вершины двух соседних слоев куба.

Важным свойством булева куба является его *однородность*. Это означает, что в нём нет особых вершин: из любой вершины куб «смотрится» одинаково. На математическом языке такое свойство графа называется *вершинной транзитивностью*.

3.5 Расстояние Хэмминга

На множестве всех векторов длины n определяется *метрика*, т. е. корректный способ измерения расстояния между векторами. Наибо-

лее распространенной является *метрика Хэмминга*, названная так в честь американского математика и философа Ричарда Хэмминга (1915–1998).

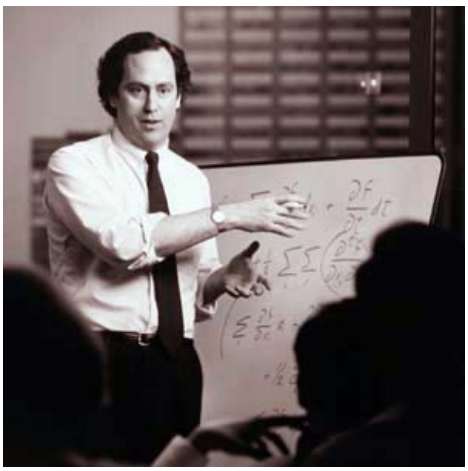
Любопытно, что Хэмминг вошел в историю теории информации благодаря единственной работе в этой области. В дальнейшем он занимался совсем другими задачами, например участвовал в разработке атомной бомбы («Манхэттенский проект»), а его «метрика Хэмминга» и «коды Хэмминга» продолжают жить.

Расстоянием Хэмминга между двумя векторами x и y длины n называется число координат, в которых они различаются. Оно обозначается через $d(x, y)$. Например, если $x = (011001)$ и $y = (110010)$, то $d(x, y) = 4$. *Весом* $wt(x)$ двоичного вектора x называется число его ненулевых координат. Например, для $z = (010110)$ справедливо $wt(z) = 3$. Нетрудно проверить, что для любых векторов x и y выполняется равенство $d(x, y) = wt(x \oplus y)$.

Метрика Хэмминга играет важную роль при работе с булевым кубом. Можно заметить, что множество ребер булева куба эквивалентно определяется как множество $E = \{(x, y) : d(x, y) = 1\}$.

Задача 17. Найдите число вершин и ребер в E^n. Чему равно число пар векторов $x, y \in E^n$ таких, что $d(x, y) = k$?

Задача 18. Найдите мощности *сферы* $S_r(x) = \{y : d(x, y) = r\}$ и *шара* $B_r(x) = \{y : d(x, y) \leqslant r\}$ радиуса r в n-мерном булевом кубе.



Ричард Хэмминг

3.6 Грани и подпространства

Гранью размерности k в булевом кубе E^n называется множество $\Gamma^{a_1,\ldots,a_{n-k}}_{i_1,\ldots,i_{n-k}}$ всех векторов вида $(*,\ldots,*,a_1,*,\ldots,*,a_{n-k},*,\ldots,*)$, где в координатах с номерами $i_1,\ldots,i_{n-k}$ зафиксированы значения $a_1,\ldots,a_{n-k}$, при этом значения в остальных координатах произвольны (они помечены $*$). Нетрудно понять, что мощность такой грани равна 2^k. Множество $\{i_1,\ldots,i_{n-k}\}$ называется *направлением* грани. Например, в четырёхмерном кубе грани $\Gamma^{0,1}_{1,2}$ и $\Gamma^{1,1,1}_{1,2,3}$ состоят из векторов (0100), (0101), (0110), (0111) и (1110), (1111) соответственно.

Задача 19. Чему равно число различных граней в E^n фиксированного направления $i_1,\ldots,i_{n-k}$? Докажите, что эти грани не пересекаются и при объединении дают весь куб E^n.

Задача 20. Определите число всех граней размерности k в E^n. Чему равно число всех граней в E^n?

Задача 21. Чему равно число граней в E^n размерности k, содержащих заданную вершину x? А заданную грань размерности ℓ?

Линейным подпространством булева куба E^n называется такое его подмножество L, что сумма $x \oplus y$ любых двух его элементов x, y принадлежит L. Заметим, что любое линейное подпространство содержит нулевой вектор. Максимальная по включению система линейно независимых векторов из L образует *базис* подпространства; число векторов в базисе называется *размерностью* подпространства и обозначается через $dim(L)$. *Аффинным подпространством* называется любой смежный класс линейного пространства, т. е. множество вида $M = z \oplus L$, где z — некоторый вектор. Отметим, что́ грань — это частный случай аффинного подпространства булева куба.

3.7 Булевы функции и куб

Каждую булеву функцию f от n переменных можно определить множеством всех тех векторов длины n, на которых f принимает значение 1. Это множество называется *носителем* булево́й функции и обозначается $supp(f)$. Например, носитель функции $g(x_1,x_2,x_3) = x_1 \oplus x_2 \oplus x_3 \oplus 1$ состоит из векторов (000), (011), (101), (110). Очевидно, что разные булевы функции всегда имеют разные носители. Так

возникает следующая интерпретация. Каждое подмножество вершин n-мерного булева куба является носителем некоторой булевой функции от n переменных.

С другой стороны, каждую булеву функцию от n переменных можно однозначно определить вектором её значений длины 2^n. При этом важно фиксировать порядок, в котором перечисляются значения. Обычно значения перебираются согласно лексикографическому порядку векторов длины n.

Например, при $n = 3$ лексикографический порядок векторов такой: $(000), (001), (010), (011), (100), (101), (110), (111)$. Тогда вектор значений функции состоит по порядку из значений $g(000)$, $g(001)$, $g(010)$, $g(011)$, $g(100)$, $g(101)$, $g(110)$, $g(111)$. Получаем, что приведённой выше функции g соответствует вектор (10010110).

Заметим, что каждый вектор длины 2^n является вектором значений некоторой булевой функции. Так, каждая вершина булева куба размерности 2^n определяет булеву функцию от n переменных. И это другая часто используемая интерпретация.

Весом Хэмминга булевой функции называется вес её вектора значений и обозначается через $wt(f)$. А именно $wt(f) = |supp(f)|$.

Кстати, для булевых функций «работает» и метрика, которая определялась для булева куба. Под *расстоянием Хэмминга* между двумя булевыми функциями от n переменных понимается расстояние между векторами их значений.

3.8 Аффинно эквивалентные функции

Часто на множестве булевых функций вводится отношение эквивалентности. С помощью него всё множество булевых функций от n переменных делится на классы, в каждом из которых функции эквивалентны, а значит, многие их свойства одинаковы. Такой подход упрощает изучение всех булевых функций: достаточно рассматривать только представителей классов эквивалентности.

В качестве отношения эквивалентности часто рассматривают аффинную эквивалентность. Приведём необходимое определение.

Булевы функции f и g от n переменных *аффинно эквивалентны*, если существуют невырожденная матрица A порядка n, векторы b, c длины n и константа $\lambda \in \mathbb{Z}_2$ такие, что

$$g(x) = f(Ax \oplus b) \oplus \langle c, x \rangle \oplus \lambda.$$

Например, функции $f(x_1, x_2, x_3, x_4) = x_1x_2 \oplus x_3x_4$ и $g(x_1, x_2, x_3, x_4) = x_1x_2 \oplus x_1x_3 \oplus x_2x_4 \oplus x_3$ аффинно эквивалентны. Действительно, $g(x) = f(Ax \oplus b) \oplus \langle c, x\rangle \oplus \lambda$, где

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, b = (0011), c = (0001), \lambda = 1.$$

Покажем подробнее:

$f(Ax \oplus b) \oplus \langle c, x\rangle \oplus \lambda = f(x_1 \oplus x_4, x_2 \oplus x_3, x_3 \oplus 1, x_4 \oplus 1) \oplus x_4 \oplus 1 =$
$= (x_1 \oplus x_4)(x_2 \oplus x_3) \oplus (x_3 \oplus 1)(x_4 \oplus 1) \oplus x_4 \oplus 1 =$
$= x_1x_2 \oplus x_1x_3 \oplus x_2x_4 \oplus x_3 = g(x).$

3.9 Преобразование Уолша — Адамара

Это один из удобных и современных инструментов для работы с булевыми функциями (в частности, в криптографии).

Скалярным произведением $\langle x, y\rangle$ двух векторов x, y длины n называется число $x_1y_1 \oplus \ldots \oplus x_ny_n$. Два вектора *ортогональны*, если $\langle x, y\rangle = 0$. Заметим, что любая аффинная булева функция от n переменных представима в виде $\langle a, x\rangle \oplus b$ для подходящего вектора a длины n и константы b.

Задача 22. Для фиксированного вектора x длины n определите количество ортогональных и неортогональных ему векторов.

Каждой булевой функции f удобно сопоставить некоторую *целочисленную* функцию W_f, которая более наглядно отражает (а лучше сказать — проецирует) свойства исходной булевой функции.

Преобразованием Уолша — Адамара булевой функции f от n переменных называется целочисленная функция W_f, заданная на множестве $\mathbb{Z}_2^n$ равенством

$$W_f(y) = \sum_{x \in \mathbb{Z}_2^n} (-1)^{\langle x,y\rangle \oplus f(x)}.$$

Числа $W_f(y)$ называются *коэффициентами Уолша — Адамара* булевой функции f. Набор $W_f = \{W_f(x_1, \ldots, x_n) : x \in \mathbb{Z}_2^n\}$, где векторы x перебираются в лексикографическом порядке, называется *спектром*

Уолша — Адамара функции f. Как будет показано далее, булева функция однозначно восстанавливается по спектру своих коэффициентов.

Например, для функции $g(x_1, x_2) = x_1 x_2$ имеем

$$W_g(00) = (-1)^{\langle 00,00\rangle\oplus 0} + (-1)^{\langle 01,00\rangle\oplus 0} + (-1)^{\langle 10,00\rangle\oplus 0} + (-1)^{\langle 11,00\rangle\oplus 1} = 2,$$

$$W_g(01) = (-1)^{\langle 00,01\rangle\oplus 0} + (-1)^{\langle 01,01\rangle\oplus 0} + (-1)^{\langle 10,01\rangle\oplus 0} + (-1)^{\langle 11,01\rangle\oplus 1} = 2,$$

$$W_g(10) = (-1)^{\langle 00,10\rangle\oplus 0} + (-1)^{\langle 01,10\rangle\oplus 0} + (-1)^{\langle 10,10\rangle\oplus 0} + (-1)^{\langle 11,10\rangle\oplus 1} = 2,$$

$$W_g(11) = (-1)^{\langle 00,11\rangle\oplus 0} + (-1)^{\langle 01,11\rangle\oplus 0} + (-1)^{\langle 10,11\rangle\oplus 0} + (-1)^{\langle 11,11\rangle\oplus 1} = -2.$$

Так, её спектр Уолша — Адамара имеет вид $W_g = \{2, 2, 2, -2\}$.

Теорема 2. *Для любой булевой функции f от n переменных справедлива формула обращения*

$$(-1)^{f(z)} = \frac{1}{2^n} \sum_{y\in\mathbb{Z}_2^n} W_f(y)(-1)^{\langle z,y\rangle}.$$

Доказательство. Подставляя выражение для коэффициента Уолша — Адамара $W_f(y)$ в правую часть формулы, имеем

$$\frac{1}{2^n} \sum_{y\in\mathbb{Z}_2^n} \sum_{x\in\mathbb{Z}_2^n} (-1)^{\langle x,y\rangle\oplus f(x)} (-1)^{\langle z,y\rangle}.$$

Меняя порядок суммирования, получаем

$$\frac{1}{2^n} \sum_{x\in\mathbb{Z}_2^n} (-1)^{f(x)} \sum_{y\in\mathbb{Z}_2^n} (-1)^{\langle x\oplus z,y\rangle}.$$

Заметим, что для любого вектора u выполняется

$$\sum_{y\in\mathbb{Z}_2^n} (-1)^{\langle u,y\rangle} = \begin{cases} 0 & \text{при } u \neq 0; \\ 2^n & \text{при } u = 0. \end{cases} \tag{3.1}$$

Используем это соотношение при $u = x \oplus z$. Тогда правая часть доказываемого соотношения приобретает вид

$$\frac{1}{2^n}(-1)^{f(z)} 2^n,$$

откуда и следует утверждение теоремы. □

В теореме 2 в точности утверждается, что любая булева функция однозначно определяется своими коэффициентами Уолша — Адамара.

Докажем некоторые свойства спектра Уолша — Адамара.

Теорема 3. *Для любой булевой функции f от n переменных справедливо равенство*

$$\sum_{y\in\mathbb{Z}_2^n} (W_f(y))^2 = 2^{2n},$$

которое называется равенством Парсеваля.

Доказательство. Снова воспользуемся определением $W_f(y)$. Левая часть доказываемого равенства имеет вид

$$\sum_{y\in\mathbb{Z}_2^n}\Big(\sum_{x\in\mathbb{Z}_2^n}(-1)^{\langle x,y\rangle\oplus f(x)}\Big)\Big(\sum_{z\in\mathbb{Z}_2^n}(-1)^{\langle z,y\rangle\oplus f(z)}\Big).$$

Раскрывая скобки и меняя порядок суммирования, получаем

$$\sum_{x\in\mathbb{Z}_2^n}\sum_{z\in\mathbb{Z}_2^n}(-1)^{f(x)\oplus f(z)}\sum_{y\in\mathbb{Z}_2^n}(-1)^{\langle x\oplus z,y\rangle}.$$

Применяя равенство (3.1) при $u = x \oplus z$, приходим к выражению

$$\sum_{x\in\mathbb{Z}_2^n}(-1)^{f(x)\oplus f(x)}2^n,$$

из которого и получаем требуемое число 2^{2n}. □

Поскольку число всех коэффициентов $W_f(y)$ булевой функции f равно 2^n, из равенства Парсеваля следует, что максимум модуля коэффициента Уолша — Адамара не может быть меньше величины $2^{n/2}$. А именно справедливо

Утверждение 1. *Выполняется* $\max_{y\in\mathbb{Z}_2^n}|W_f(y)| \geqslant 2^{n/2}$.

Этот факт нам пригодится в дальнейшем.

4. Булевы функции. Алгебраический подход

Другим удобным способом представления булевой функции является её *трейс–форма*. Это алгебраическое представление, в котором булева функция рассматривается как функция над конечным полем.

4.1 Конечное поле

Приведём необходимые определения.

Полем называется множество F с определёнными на нём бинарными операциями сложения $(+)$ и умножения $(\cdot)$ такими, что

1) F — абелева группа относительно операции сложения, т. е. для любых элементов $a, b, c \in F$ выполнены условия:

• элемент $a + b$ принадлежит множеству F (корректность);

• $a + (b + c) = (a + b) + c$ (ассоциативность);

• в F существует *нулевой элемент* 0 такой, что $a + 0 = 0 + a = a$ (существование нуля);

• в F существует элемент $(-a)$ такой, что $(-a) + a = a + (-a) = 0$ (существование обратного элемента);

• $a + b = b + a$ (коммутативность);

2) множество $F^* = F \setminus \{0\}$ — абелева группа относительно умножения, т. е. для любых $a, b, c \in F^*$ выполняются условия:

• элемент $a \cdot b$ принадлежит множеству F^* (корректность);

• $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ (ассоциативность);

• в F^* существует *единичный элемент* 1 такой, что $a \cdot 1 = 1 \cdot a = a$ (существование единицы);

• в F^* существует элемент a^{-1} такой, что $a^{-1} \cdot a = a \cdot a^{-1} = 1$ (существование обратного элемента);

• $a \cdot b = b \cdot a$ (коммутативность);

3) справедливо $0 \cdot a = a \cdot 0 = 0$;

4) выполнен закон дистрибутивности, т. е. для любых $a, b, c \in F$ справедливо равенство $a \cdot (b + c) = a \cdot b + a \cdot c$.

Если вы столкнулись с определением поля первый раз, то пусть оно не покажется вам слишком сложным. Можно считать, что все приведённые условия на операции вводятся для того, чтобы с новым

сложением/умножением было «привычно» работать. Например, раскрывать скобки, менять местами слагаемые или множители.

Поле называется *конечным*, если F — конечное множество. Число элементов q в множестве F называется *порядком поля*. Конечные поля часто называют *полями Галуа* и обозначают $GF(q)$ (сокращенно от Galois Field) в честь талантливого французского математика и революционера Эвариста Галуа (1811–1832).

Винсент Ван Гог. Пшеничное поле с заходящим солнцем

О конечных полях доказаны следующие важные утверждения.

Теорема 4. *Конечное поле $GF(q)$ существует тогда и только тогда, когда q является степенью простого числа, $q = p^m$.*

Число p называется *характеристикой* поля $GF(p^m)$.

Теорема 5. *Конечное поле $GF(q)$ единственно с точностью до изоморфизма.*

Обозначим через $GF^*(q)$ множество $GF(q) \setminus \{0\}$.

Теорема 6. *Множество $GF^*(q)$ образует циклическую мультипликативную группу порядка $q-1$, т. е. существует элемент α такой, что $GF^*(q) = \{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}$ и $\alpha^{q-1} = 1$.*

Элемент α, указанный в теореме 6, называется *примитивным*. По его определению любой ненулевой элемент поля β представим в виде $\beta = \alpha^k$ для подходящего целого k. Под *степенью* α^k элемента α понимается его k-кратное произведение $\alpha \cdot \ldots \cdot \alpha$ самого на себя либо 1 в случае $k = 0$. С помощью α поле $GF(q)$ удобно представлять в виде

$$GF(q) = \{0, \alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{q-2}\}.$$

В таком представлении легко осуществляется операция умножения элементов, так как выполняется

$$\alpha^i \cdot \alpha^j = \alpha^{(i+j) \mod (q-1)}.$$

Теорема 7. *Каждый элемент β поля $GF(q)$ удовлетворяет тождеству $\beta^q = \beta$, т. е. является корнем уравнения $x^q = x$.*

Задача 23. Пусть a_j — произвольные (не обязательно различные, но не все одновременно равные нулю) элементы поля $GF(q)$, где $j = 0, \ldots, q-1$. Докажите, что найдётся элемент $\beta \in GF(q)$ такой, что $a_0\beta^0 + a_1\beta^1 + a_2\beta^2 + \ldots + a_{q-1}\beta^{q-1} \neq 0$.

Особый интерес для нас будет представлять случай $q = 2^n$.

4.2 Поле $GF(2^n)$

Покажем, как на множестве $\mathbb{Z}_2^n$ двоичных векторов длины n корректно определить операции сложения и умножения.

Нетрудно проверить, что множество $\mathbb{Z}_2 = \{0, 1\}$ с операциями $\oplus$ и $(\cdot)$ является полем, а именно $GF(2)$.

Пусть x — *переменная*, некоторый формальный символ, неизвестная величина, не принадлежащая множеству $\mathbb{Z}_2$. *Многочленом* (или *полиномом*) степени k над $\mathbb{Z}_2$ называется формальное выражение

$$a(x) = a_1x^k + a_2x^{k-1} + \ldots + a_{k-1}x^2 + a_kx + a_{k+1},$$

где все коэффициенты a_i принадлежат $\mathbb{Z}_2$ (и $a_1 \neq 0$). Многочлены над $\mathbb{Z}_2$ можно складывать, при этом их сумма — это многочлен, полученный путём сложения по модулю два коэффициентов при соответствующих степенях x. Многочлен называется *неприводимым*, если его нельзя разложить на множители. Например, многочлен $x^3 + x + 1$

неприводим, тогда как многочлен $x^3 + 1$ приводим: его можно представить в виде произведения $(x + 1)(x^2 + x + 1)$.

Пусть задан неприводимый многочлен $g(x)$ степени n. Каждому двоичному вектору $c = (c_1, \ldots, c_n)$ длины n поставим в соответствие многочлен $c(x)$ степени $n - 1$ над $\mathbb{Z}_2$ вида

$$c(x) = c_1 x^{n-1} + c_2 x^{n-2} + \ldots + c_{n-1} x + c_n.$$

Определим операции сложения и умножения векторов так:

$$\begin{array}{ll} c + c' = d, & \text{где } d(x) = c(x) + c'(x); \\ c \cdot c' = d, & \text{где } d(x) = c(x) \cdot c'(x) \mod g(x). \end{array}$$

Относительно этих операций множество $\mathbb{Z}_2^n$ является полем.

Таким образом, поле $GF(2^n)$ состоит из всевозможных многочленов от переменной x степени меньше n с коэффициентами из множества $\{0, 1\}$. Многочлены складываются путём сложения по модулю 2 соответствующих коэффициентов, а перемножаются по модулю некоторого неприводимого многочлена $g(x)$ степени n. Вообще говоря, многочлен $g(x)$ может быть произвольным неприводимым. Поля, построенные с помощью различных многочленов, изоморфны (см. теорему 5).

В следующем параграфе будет приведён пример поля $GF(2^3)$.

Задача 24. Покажите, что для любых $a, b \in GF(2^n)$ и любого натурального k выполняется равенство $(a + b)^{2^k} = a^{2^k} + b^{2^k}$.

4.3 След из поля в подполе

В конечном поле определяется специальная функция, которую называют *след* или trace. Применительно к полю $GF(2^n)$, это функция из $GF(2^n)$ в $GF(2^n)$ следующего вида:

$$tr(c) = c + c^2 + c^{2^2} + c^{2^3} + c^{2^4} + \ldots + c^{2^{n-1}}.$$

Удивительным свойством следа, особо выделяющим эту функцию среди других, является то, что для любого элемента c значение $tr(c)$ всегда принадлежит простому подполю $GF(2)$, т. е. всегда равно либо нулевому, либо единичному элементу поля! Можно сказать, что след является своеобразной проекцией элемента c, его простым отражением, чем и оправдывает своё название.

Если теперь нулевому элементу поля поставить в соответствие число 0, а единичному — число 1, то функцию tr можно отождествить с соответствующей булевой функцией от n переменных. Действительно, ведь каждому входному элементу c соответствует двоичный вектор $(c_1, \ldots, c_n)$ длины n: рассматриваем его как вектор переменных.

Приведём пример для $n = 3$. Пусть поле $GF(2^3)$ порождается с помощью неприводимого многочлена $g(x) = x^3 + x + 1$. Можно проверить, что в этом поле элемент $x+1$ будет примитивным (для этого достаточно перебрать его степени с первой по шестую и убедиться, что все они различны). Обозначим $x + 1$ через α и приведём различные представления элементов поля.

вектор	многочлен	α^k	след $tr(c) = c + c^2 + c^4$	f
(000)	0	-	0	0
(001)	1	α^0	$\alpha^0 + \alpha^{0\cdot 2} + \alpha^{0\cdot 4} = 1$	1
(010)	x	α^5	$\alpha^5 + \alpha^{5\cdot 2} + \alpha^{5\cdot 4} = \alpha^5 + \alpha^3 + \alpha^6 = 0$	0
(011)	$x + 1$	α^1	$\alpha^1 + \alpha^{1\cdot 2} + \alpha^{1\cdot 4} = 1$	1
(100)	x^2	α^3	$\alpha^3 + \alpha^{3\cdot 2} + \alpha^{3\cdot 4} = \alpha^3 + \alpha^6 + \alpha^5 = 0$	0
(101)	$x^2 + 1$	α^2	$\alpha^2 + \alpha^{2\cdot 2} + \alpha^{2\cdot 4} = \alpha^2 + \alpha^4 + \alpha^1 = 1$	1
(110)	$x^2 + x$	α^6	$\alpha^6 + \alpha^{6\cdot 2} + \alpha^{6\cdot 4} = \alpha^6 + \alpha^5 + \alpha^3 = 0$	0
(111)	$x^2 + x + 1$	α^4	$\alpha^4 + \alpha^{4\cdot 2} + \alpha^{4\cdot 4} = \alpha^4 + \alpha^1 + \alpha^2 = 1$	1

В таблице также показано, как вычисляется след каждого элемента. В последнем столбце приводится набор значений булевой функции, которая отождествляется со следом. Заметим, что $f(x_1, x_2, x_3) = x_3$. На самом деле след всегда определяет линейную булеву функцию. Её конкретный вид зависит от неприводимого многочлена $g(x)$, выбранного при построении поля.

Задача 25. Докажите, что $(tr(c))^2 = tr(c^2) = tr(c)$ для любого c.

Задача 26. Докажите, что функция $tr(c)$ принимает значения только из множества $GF(2)$.

Задача 27. Докажите, что функция $tr(c)$ линейна, т. е. для любых элементов c', c'' выполняется $tr(c' + c'') = tr(c') + tr(c'')$.

Задача 28. (*) Покажите, что существует c такой, что $tr(c) = 1$.

Задача 29. Покажите, что функция $tr(c)$ принимает значения 0 и 1 одинаково часто.

4.4 Описание линейных функций

Булеву функцию $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ будем отождествлять с функцией, определённой над полем, а именно с функцией $f : GF(2^n) \to GF(2)$.

Теорема 8. *Множеству всех линейных булевых функций от n переменных соответствует множество всех функций из $GF(2^n)$ в $GF(2)$ вида $\ell_a(c) = tr(a \cdot c)$, где a пробегает поле $GF(2^n)$.*

Доказательство. Покажем, что каждая функция $\ell_a(c)$ определяет линейное отображение из $GF(2^n)$ в $GF(2)$. Сначала убедимся в том, что $tr(c)$ — линейное отображение. Имеем

$$tr(c' + c'') = (c' + c'') + (c' + c'')^2 + (c' + c'')^{2^2} + \ldots + (c' + c'')^{2^{n-1}}.$$

Из равенства $(c' + c'')^{2^k} = c'^{2^k} + c''^{2^k}$ (см. задачу 24) получаем

$$tr(c' + c'') = c' + c'^2 + \ldots + c'^{2^{n-1}} + c'' + c''^2 + \ldots + c''^{2^{n-1}} = tr(c') + tr(c'').$$

Теперь несложно получить линейность ℓ_a. Действительно, согласно задаче 16 булева функция f линейна тогда и только тогда, когда $f(x \oplus y) = f(x) \oplus f(y)$ для любых x, y. Проверим это свойство для функции ℓ_a. Имеем

$$\ell_a(c' + c'') = tr(a \cdot c' + a \cdot c'') = tr(a \cdot c') + tr(a \cdot c'') = \ell_a(c') + \ell_a(c'').$$

Итак, функция ℓ_a линейна.

Число всех функций ℓ_a равно 2^n, что совпадает с числом всех линейных функций. Осталось показать, что функции ℓ_a и ℓ_b при $a \neq b$ соответствуют различным линейным функциям. Рассмотрим разность

$$\ell_a(c) - \ell_b(c) = tr((a - b)c) = tr(a' \cdot c),$$

где a' — подходящий элемент поля. Поскольку $a' \neq 0$, элемент $a' \cdot c$ пробегает всё поле $GF(2^n)$. А значит, как следует из задачи 28, найдётся такой элемент c, что $tr(a' \cdot c) \neq 0$. Таким образом, отображения ℓ_a, ℓ_b различны. Следовательно, различны и соответствующие им булевы функции. □

Для того чтобы привести описание произвольной (а не только линейной) булевой функции с помощью следа, необходимо сначала рассмотреть алгебраическое представление векторной булевой функции, а также определить группу автоморфизмов конечного поля.

4.5 Векторные булевы функции

Векторные булевы функции (см. раздел 3.3), которые мы также отождествим с функциями над конечным полем, можно представлять с помощью одномерных многочленов над $GF(2^n)$. Отметим, что это представление существует только для функций вида $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, т. е. в случае $n = m$.

Теорема 9. *Любая функция $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ может быть однозначно представлена в форме одномерного многочлена (ОМФ):*

$$F(c) = \sum_{j=0}^{2^n-1} a_j c^j, \quad \text{где } a_j \in GF(2^n).$$

Доказательство. Действительно, число всех векторных функций $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ равно $(2^n)^{2^n}$. Число различных многочленов $\sum_{j=0}^{2^n-1} a_j x^j$ также равно $(2^n)^{2^n}$. Осталось заметить, что различные многочлены определяют различные векторные функции, что несложно следует из задачи 23. □

4.6 Группа автоморфизмов поля $GF(p^n)$

Автоморфизмом поля $GF(p^n)$ над $GF(p)$ называется такое взаимнооднозначное отображение φ поля на себя, что

1) φ оставляет элементы $GF(p)$ неподвижными;

2) φ сохраняет операции в поле, а именно для любых $a, b \in GF(p^n)$:

$$\varphi(a + b) = \varphi(a) + \varphi(b);$$

$$\varphi(ab) = \varphi(a)\varphi(b).$$

Множество всех автоморфизмов поля образует группу относительно операции суперпозиции. Эта группа называется *группой Галуа* поля $GF(p^n)$ и обозначается через $Aut(GF(p^n))$.

Теорема 10. *Группа автоморфизмов поля $GF(p^n)$ является циклической порядка n и порождается отображением $\varphi : a \to a^p$.*

Другими словами,

$$Aut(GF(p^n)) = \{e, \varphi, \varphi^2, \ldots, \varphi^{n-1}\},$$

где e — тождественное отображение. Например, для поля $GF(2^3)$, построенного в разделе 4.3, группа автоморфизмов имеет вид

$$Aut(GF(2^3)) = \{e, a \to a^2, a \to a^2 + a\},$$

поскольку a^4 равно a^2+a по модулю выбранного порождающего многочлена.

4.7 Трейс-форма булевой функции

Итак, как было показано в разделе 4.4, $tr(c)$ является линейной булевой функцией, т. е. можно выбрать ненулевой вектор b такой, что

$$tr(c) = \langle b, c\rangle = b_1c_1 \oplus \ldots \oplus b_nc_n.$$

Каждый раз в подобных равенствах надо понимать, в каком контексте рассматривается элемент c. В данном случае: слева он — элемент поля, справа — двоичный вектор. Приравнивая левую и правую части, мы неявно̀ отождествляем элементы $GF(2)$ с числами 0, 1.

Рассмотрим векторную булеву функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$. Её всегда можно представить в виде $F(c) = (f^1(c), \ldots, f^n(c))$, где f^j — обычная булева функция. Тогда

$$tr(F(c)) = b_1f^1(c) \oplus \ldots \oplus b_nf^n(c).$$

Поскольку b — ненулевой вектор, всегда найдётся номер i такой, что $b_i \neq 0$. Зафиксируем этот номер. Тогда любую булеву функцию f от n переменных можно представить в виде

$$f(c) = tr(F(c))$$

для подходящей векторной функции F, например, пусть

$$F(c) = (0, \ldots, 0, f(c), 0, \ldots, 0),$$

где $f(c)$ расположена на i-м месте. Вообще говоря, таких функций F можно выбрать очень много.

Согласно теореме 9 о представлении векторной функции F в форме одномерного многочлена имеем

$$f(c) = tr\left(\sum_{j=0}^{2^n-1} a_j c^j\right). \tag{4.1}$$

Докажем следующее утверждение.

Утверждение 2. *Для любого элемента a из поля $GF(2^n)$ найдётся элемент $d \in GF(2^n)$ такой, что $tr(ac^2) = tr(dc)$ для любого $c \in GF(2^n)$.*

Доказательство. Случай $a = 0$ тривиален. Пусть $a \neq 0$. По теореме 10 отображение $\varphi : \beta \to \beta^2$ является автоморфизмом поля $GF(2^n)$. Так как φ — взаимно-однозначное отображение, то для любого ненулевого элемента поля $GF(2^n)$ существует прообраз относительно φ. Это означает, что для любого ненулевого a найдётся элемент d такой, что $a = d^2$. Тогда в силу задачи 25 для любого $c \in GF(2^n)$ имеем $tr(ac^2) = tr(d^2c^2) = tr(dc)$. □

Используя утверждение 2, можно сократить число слагаемых в скобках выражения (4.1), а именно оставить суммирование только по таким j, j', которые не связаны соотношениями вида $j = 2j'$.

Опишем это более строго.

Множество $\{k, 2k, 2^2k, 2^3k, \ldots\}$ называется *циклотомическим классом* по модулю $2^n - 1$ с представителем k. Например, при $n = 4$ есть пять циклотомических классов $\{0\}$, $\{1, 2, 4, 8\}$, $\{3, 6, 12, 9\}$, $\{5, 10\}$, $\{7, 14, 13, 11\}$. Их представителями являются числа 0, 1, 3, 5 и 7. При этом считаем, что $c^0 = 1$ для любого $c \in GF(2^n)$. Заметим, что $c^{2^n-1} = 1$ для всех $c \in GF^*(2^n)$ и $c^{2^n-1} = 0$, если $c = 0$.

Тогда согласно (4.1) и утверждению 2 справедлива

Теорема 11. *Любая функция $f : GF(2^n) \to GF(2)$ может быть представлена в виде*

$$f(c) = tr\left(\sum_{j \in CS} a_j c^j\right) + tr(a_{2^n-1} c^{2^n-1}) = \left(\sum_{j \in CS} tr(a_j c^j)\right) + tr(a_{2^n-1} c^{2^n-1}),$$

для подходящих элементов $a_j \in GF(2^n)$, где CS — множество представителей циклотомических классов по модулю $2^n - 1$.

Это представление называется *трейс-формой* булевой функции. Важно отметить, что оно не однозначное! Однако можно наложить дополнительные условия на выбор коэффициентов a_j так, чтобы булева функция f всегда имела только одно такое представление. Как это сделать, можно прочитать в статье [41] А. С. Кузьмина и др.

Трейс-формы — один из наиболее удобных и современных инструментов для работы с булевыми функциями. Некоторые криптографические характеристики булевой функции особенно естественно выражаются только с помощью следа. Отметим также, что наиболее интересные конструкции криптографических булевых функций были найдены с участием именно этого алгебраического представления.

Задача 30. Трейс-форма. Пусть поле $GF(2^3)$ построено с помощью порождающего многочлена $g(x) = x^3 + x + 1$. Найдите трейс-формы следующих булевых функций:

1) $f(x_1, x_2, x_3) = 0$;
2) $f(x_1, x_2, x_3) = 1$;
3) $f(x_1, x_2, x_3) = x_1$;
4) $f(x_1, x_2, x_3) = x_1x_2x_3$;
5) $f(x_1, x_2, x_3) = x_1x_2 \oplus x_1x_3 \oplus x_2x_3$.

4.8 Мономиальные булевы функции

Рассмотрим булевы функции, трейс-формы которых имеют самый простой вид, а именно состоят только из одного монома. Булева функция f от n переменных называется *мономиальной*, если она представима в виде

$$f(c) = tr(ac^d)$$

для подходящего $a \in GF(2^n)$ и целого числа d, где $1 \leqslant d \leqslant 2^n - 1$.

При $d = 1$ класс мономиальных функций совпадает с множеством всех линейных функций (см. теорему 8).

Если число d взаимно просто с $2^n - 1$, то функции вида $tr(ac^d)$ называются *собственными мономиальными*.

Теорема 12. *Степень булевой функции $f(c) = tr(ac^d)$, где $a \neq 0$, равна весу Хэмминга двоичного представления числа d.*

Опишем все собственные мономиальные булевы функции при $n = 3$. Пусть $g(x) = x^3 + x + 1$ — порождающий многочлен поля $GF(2^3)$. Пусть, как и раньше, выбран примитивный элемент $\alpha = x + 1$.

Тогда все линейные функции представляются так:

вектор	a	$tr(c)$	$tr(ac)$	значения	$f(x_1, x_2, x_3)$
(000)	0	0	$tr(0)$	00000000	0
(001)	1	1	$tr(c)$	01010101	x_3
(010)	α^5	0	$tr(\alpha^5 c)$	00001111	x_1
(011)	α^1	1	$tr(\alpha^1 c)$	01011010	$x_1 \oplus x_3$
(100)	α^3	0	$tr(\alpha^3 c)$	00110011	x_2
(101)	α^2	1	$tr(\alpha^2 c)$	01100110	$x_2 \oplus x_3$
(110)	α^6	0	$tr(\alpha^6 c)$	00111100	$x_1 \oplus x_2$
(111)	α^4	1	$tr(\alpha^4 c)$	01101001	$x_1 \oplus x_2 \oplus x_3$

Несложно понять, что вектор значений функции $tr(\alpha^k c)$ получается из вектора значений $tr(c)$ путём переупорядочивания отдельных значений. Этот порядок и определяется множителем α^k.

Рассмотрим также случай $d = 3$. Это функции степени 2, так как число 3 имеет двоичное представление (11) веса 2.

вектор	a	$tr(ac^3)$	значения	$f(x_1, x_2, x_3)$
(000)	0	$tr(0)$	00000000	0
(001)	1	$tr(c^3)$	01101010	$x_1x_2 \oplus x_1 \oplus x_2 \oplus x_3$
(010)	α^5	$tr(\alpha^5 c^3)$	00011110	$x_2x_3 \oplus x_1$
(011)	α^1	$tr(\alpha^1 c^3)$	01110100	$x_1x_2 \oplus x_2x_3 \oplus x_2 \oplus x_3$
(100)	α^3	$tr(\alpha^3 c^3)$	00100111	$x_1x_3 \oplus x_2x_3 \oplus x_2$
(101)	α^2	$tr(\alpha^2 c^3)$	01001101	$x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1 \oplus x_3$
(110)	α^6	$tr(\alpha^6 c^3)$	00111001	$x_1x_3 \oplus x_1 \oplus x_2$
(111)	α^4	$tr(\alpha^4 c^3)$	01010011	$x_1x_2 \oplus x_1x_3 \oplus x_3$

Мономиальные булевы функции понадобятся при рассмотрении ряда криптографических свойств булевых функций (см. далее главу 9 и, в частности, раздел 9.9).

5. Криптографические свойства булевых функций

В этой главе мы кратко рассмотрим ряд свойств булевой функции, которыми она должна обладать для использования её в криптографических приложениях.

Эти свойства были определены в 80-х, 90-х годах XX века и в начале XXI века путём детального анализа успешных способов взлома блочных и поточных шифров. Анализ показал, что во многих случаях успех достигался за счёт нахождения слабостей в математических свойствах булевых функций, описывающих компоненты шифра. Так возникла необходимость выработать строгие критерии для криптографических булевых функций.

Другой вопрос — как совместить различные криптографические свойства в одной булевой функции? Дело в том, что многие свойства противоречат друг другу. Поэтому разработчику шифров постоянно приходится находить между ними непростой компромисс (так называемый trade-off), порой не достигая максимума ни в одном криптографическом качестве.

5.1 Высокая алгебраическая степень

Параметр $deg(f)$ булевой функции должен быть большим. Для блочных шифров это условие накладывается, как правило, для того, чтобы система уравнений на биты ключа, построенная путём анализа структуры шифра — в том числе функции f, использующейся в качестве его компоненты, — имела высокую степень. Чем выше степень системы, тем сложнее её решить, а значит, определить ключ.

В поточных шифрах использование булевой функции высокой степени (a Boolean function with a high algebraic degree) в качестве *комбинирующей* (сочетающей выходы различных регистров сдвига) или *фильтрующей* (применяемой к битам одного регистра) повышает линейную сложность генерируемой последовательности. Отметим, что линейная сложность последовательности — одна из важнейших характеристик генератора; от неё напрямую зависит криптографическая стойкость поточного шифра (см. подробнее главу 7).

5.2 Высокая нелинейность

Булева функция должна находиться на достаточно большом расстоянии Хэмминга от множества всех аффинных булевых функций. Это расстояние называют *нелинейностью* (nonlinearity) булевой функции и обозначают через N_f. Чем выше нелинейность функции, используемой в качестве компоненты шифра, тем выше стойкость шифра к линейному криптоанализу (см. раздел 8.4). Если кратко, то суть линейного криптоанализа состоит в *подмене* сложной булевой функции, описывающей некоторое нелинейное преобразование шифра, простой линейной функцией. Конечно, при этом получается не исходный шифр, а его приближение, но, во-первых, оно гораздо проще исходного шифра и легче поддается анализу, а во-вторых, это приближение может оказаться весьма удачным и может позволить *свести* проблему криптоанализа исходного шифра к анализу его модификации с некоторой (допустимой) погрешностью.

Так что, чем выше нелинейность булевой функции, тем сильнее функция *отличается* от любой линейной функции и тем «неудачней» будет подобная замена, предложенная криптоаналитиком. Высокая нелинейность функции является одним из основных её криптографических свойств. Булевы функции от чётного числа переменных, обладающие максимальной нелинейностью, а именно равной $2^{n-1} - 2^{(n/2)-1}$, называются *бент-функциями* (bent functions). Им посвящена отдельная глава 9.

Задача 31. Докажите, что расстояние от булевой функции f от n переменных до аффинной функции $\ell_{a,b}(x) = a_1x_1 \oplus \ldots \oplus a_nx_n \oplus b$, где $a \in \mathbb{Z}_2^n$, $b \in \mathbb{Z}_2$, вычисляется по формулам

$$d(f, \ell_{a,0}) = 2^{n-1} - \frac{1}{2}W_f(a),$$

$$d(f, \ell_{a,1}) = 2^{n-1} + \frac{1}{2}W_f(a).$$

Задача 32. Докажите, что нелинейность произвольной булевой функции f определяется по формуле $N_f = 2^{n-1} - \frac{1}{2}\max_y |W_f(y)|$.

Задача 33. Докажите, что булева функция f — бент-функция тогда и только тогда, когда $W_f(y) = \pm 2^{n/2}$ для каждого y.

5.3 Сбалансированность

Булева функция называется *сбалансированной* (или *уравновешенной*, balanced function), если она принимает значения 0 и 1 одинаково часто. Это необходимое условие для снятия статистических зависимостей между входом функции и её выходом. Если несбалансированная функция f используется в качестве некоторого криптографического примитива, то для случайного (неизвестного) вектора x криптоаналитик может полагать, скажем, что $f(x) = 1$ с вероятностью, строго большей, чем $1/2$. Подобные вероятностные допущения могут быть использованы в статистических методах криптоанализа.

Задача 34. Определите число сбалансированных функций от n переменных.

5.4 Устойчивость

Пусть r — неотрицательное число, меньшее n. Булева функция f от n переменных называется *r-устойчивой* (r-resilient), если любая её подфункция, полученная фиксацией не более r переменных является сбалансированной. Булевы функции с таким более сильным (по сравнению со сбалансированностью) свойством используются, например, в качестве комбинирующих функций в поточных генераторах, повышая их стойкость к *корреляционному* криптоанализу.

Устойчивые функции тесно связаны с корреляционно-иммунными функциями.

5.5 Корреляционная иммунность

Булева функция f от n переменных называется *корреляционно-иммунной порядка r* (correlation immune of order r), $1 \leqslant r \leqslant n$, если для любой её подфункции $f^{a_1,\ldots,a_r}_{i_1,\ldots,i_r}$, полученной фиксацией r переменных, выполняется равенство

$$wt(f^{a_1,\ldots,a_r}_{i_1,\ldots,i_r}) = \frac{wt(f)}{2^r}.$$

Нетрудно доказать, что корреляционно-иммунная функция порядка r является корреляционно-иммунной и любого меньшего порядка.

Заметим, что r-устойчивую функцию можно эквивалентно определить как сбалансированную корреляционно-иммунную функцию порядка r.

Задача 35. Докажите, что булева функция f — r-устойчивая тогда и только тогда, когда она сбалансированная и корреляционно-иммунная порядка r.

В 1984 году Т. Зигенталер установил связь между алгебраической степенью булевой функции от n переменных и параметром её корреляционной иммунности. Получите и вы следующие результаты.

Задача 36. Теорема Зигенталера I. Докажите, что если f — корреляционно-иммунная порядка r, то выполняется $deg(f) + r \leqslant n$.

Задача 37. Теорема Зигенталера II. Докажите, что если f является r-устойчивой и $r \leqslant n - 2$, то выполняется $deg(f) + r \leqslant n - 1$.

Доказательство теорем Зигенталера можно найти, например, в книге Г. П. Агибалова [2].

Удобным эквивалентным определением корреляционно-иммунной функции служит следующее.

Задача 38. Докажите, что булева функция f — корреляционно-иммунная порядка r тогда и только тогда, когда $W_f(y) = 0$ для каждого вектора y такого, что $1 \leqslant wt(y) \leqslant r$.

В 2007 году Д. Г. Фон–Дер–Флаасс [114] получил верхнюю оценку корреляционной иммунности несбалансированной функции.

Задача 39. Теорема Фон-Дер–Флаасса*. Пусть f — несбалансированная корреляционно-иммунная функция порядка r отличная от константы. Докажите, что $r \leqslant (2n/3) - 1$.

Задача 40. (*) Докажите, что для r-устойчивой булевой функции f такой, что $r \leqslant n - 2$, справедливо неравенство $N_f \leqslant 2^{n-1} - 2^{r+1}$.

Решение последней задачи можно найти в монографии [104].

5.6 Алгебраическая иммунность

Это свойство было введено в 2004 году. *Алгебраической иммунностью* (algebraic immunity) булевой функции f называется минимальное число d такое, что существует булева функция g степени d, не тождественно равная нулю, для которой выполняется равенство $fg = 0$ или $(f \oplus 1)g = 0$. Алгебраическую иммунность принято обозначать через $AI(f)$. Функция g называется *аннулирующей* для f. Чем больше значение алгебраической иммунности булевой функции, тем более стойким к алгебраическому криптоанализу является шифр, в котором она используется. Подробнее об алгебраическом криптоанализе см. в параграфе 8.9.

Например, алгебраическая иммунность функции $f(x) = x_1x_2x_3 \oplus x_1$ равна 1, так как аннулирующей функцией можно выбрать $g(x) = x_1 \oplus 1$. Действительно, $fg = (x_1x_2x_3 \oplus x_1)(x_1 \oplus 1) = 0$.

Задача 41. Граница алгебраической иммунности. Докажите, что алгебраическая иммунность любой булевой функции от n переменных не превышает числа $\lceil n/2 \rceil$.

Задача 42. Простые свойства. Докажите, что для любых булевых функций f, g от n переменных выполняются:

а) $AI(f) \leqslant deg(f)$;

б) $AI(f \cdot g) \leqslant AI(f) + AI(g)$;

в) $AI(f \oplus g) \leqslant AI(f) + AI(g)$;

г) $AI(f) = AI(g)$, если g получена из f аффинным преобразованием переменных, а именно $g(x) = f(Ax \oplus b)$, где A — невырожденная матрица порядка n, вектор b — произвольный.

Задача 43. Определите значения $AI(f)$ для следующих функций f. Найдите вид аннулирующих функций.

а) $f(x) = x_1x_2x_4 \oplus x_1x_2 \oplus 1$;

б) $f(x) = 0$;

в) $f(x) = 1$;

г) $f(x) = x_1 \cdot \ldots \cdot x_k$, где $k = 1, 2, \ldots, n$;

д) $f(x) = x_1 \oplus \ldots \oplus x_n$;

е) $f(x) = x_1x_2 \oplus \ldots \oplus x_{n-1}x_n$ (сумма всех попарных произведений);

ж) $f(x) = x_1x_2x_3x_4 \oplus x_5x_6$;

Задача 44. Приведите пример булевой функции f с наименьшим числом переменных такой, что $AI(f) = d$, где $d = 1, 2, \ldots, k$.

Подробнее об алгебраически иммунных функциях можно прочитать в обзоре М. Э. Тужилина [71].

5.7 Дифференциальная равномерность

Это свойство определяется только для векторных булевых функций. Векторная булева функция $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ называется *дифференциально δ-равномерной* (differentially δ-uniform), если при любом ненулевом векторе a и произвольном векторе b уравнение

$$F(x) \oplus F(x \oplus a) = b$$

имеет не более δ решений, где δ — целое число. Кстати, заметим, что если уравнение имеет одно решение (скажем, x), то решением будет и вектор $x \oplus a$. Чем меньше δ у функции, определяющей нелинейное преобразование шифра, тем более стойким к дифференциальному криптоанализу является шифр (см. раздел 8.8). Векторная булева функция называется *APN-функцией*, если она дифференциально равномерна с минимальным возможным значением δ. В этом случае $\delta = 2$. Сложным вопросом в этой области остаётся вопрос о построении взаимно однозначных APN-функций при чётном n. Подробнее об APN-функциях можно прочитать в работе М. Э. Тужилина [72].

В общих словах идея дифференциального криптоанализа шифра заключается в следующем. Например, вы подаете на вход шифра два блока, которые отличаются в некоторых фиксированных битах, и затем анализируете, насколько отличаются соответствующие им выходные блоки. Иногда можете обнаружить такую деталь: при фиксированной разности входных блоков некоторые разности на выходе не встречаются, а какие-то встречаются чаще других. Подобные «отклонения от нормы» сразу используются криптоаналитиками.

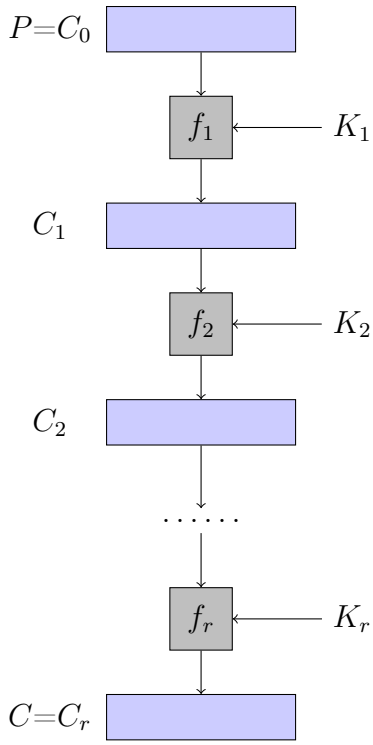
Так что APN-функции являются «совершенно равномерными» в этом отношении, а дифференциально δ-равномерные допускают лишь небольшие «отклонения» в пределах выбранного параметра. Дифференциально 4-равномерная функция, например, используется в S-блоке шифра AES — американского стандарта симметричного шифрования.

Задача 45. Докажите, что функция $S : \mathbb{Z}_2^8 \to \mathbb{Z}_2^8$, задающая S-блок шифра AES, является дифференциально 4-равномерной.

6. Блочные шифры

6.1 Принципы построения

Пусть шифруемое сообщение представлено в виде последовательности нулей и единиц. Если перед шифрованием эта последовательность разбивается на блоки фиксированной длины n и затем каждый блок шифруется отдельно, то такое шифрование называется *блочным*.



Блочный шифр

Каждый блок принято называть *открытым текстом* (plaintext) и обозначать через P. Размер блока, как правило, составляет несколько десятков битов (например, 64, 128, 256 бит). Напомним, что блоки, или векторы, подлежащие шифрованию, — это не что иное, как элементы хорошо знакомого нам булева куба размерности n. *Ключом* (key) K называется секретная последовательность битов длины m, используемая при шифровании некоторого (достаточно большого) числа блоков. Современными длинами ключей являются, например, 128, 256, 512 бит. Блочный шифр состоит из нескольких *раундов*, на каждом из которых происходит обратимое преобразование блока длины n в новый блок той же длины. Выходной блок i-го раунда называется *промежуточным шифртекстом* и обозначается C_i. Он является входным блоком следующего $i+1$-го раунда. Входным блоком первого раунда служит $C_0 = P$. *Шифртекстом* (ciphertext) C называется выходной блок последнего раунда, $C = C_r$. Число раундов r (на практике, от 8 до 64) должно быть, с одной стороны, достаточно большим для обеспечения высокой криптостойкости шифра, а с другой стороны, достаточно малым для того, чтобы шифрование было быстрым. Часто на всех раундах шифра используется одно и то же преобразование,

зависящее лишь от разных битов ключа. А именно из ключа формируется r *подключей* $K_1, \ldots, K_r$, каждый из которых используется на определённом раунде.

6.2 Примитивные операции

Приведём ряд примитивных операций, которые часто встречаются в конструкциях шифров.

Пусть A, B — двоичные блоки длины n. Пусть D — двоичный блок длины m. Будем считать, что в каждом блоке биты нумеруются слева направо, начиная с первого бита.

Каждый блок A является двоичным представлением некоторого целого числа $r(A)$ от 0 до $2^n - 1$, а именно, если $A = (a_1, \ldots, a_n)$, то

$$r(A) = a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0.$$

Сложение по модулю 2 или XOR. Обозначается эта операция через $\oplus$. Результатом сложения $A \oplus B$ является блок длины n, каждая компонента которого получена сложением по модулю 2 соответствующих компонент A и B. Например, $(1101) \oplus (1011) = (0110)$. Заметим, что сложение по модулю 2 совпадает с вычитанием по модулю 2, поэтому можно не вводить операцию $\ominus$.

Сложение и вычитание по модулю 2^n. Операции обозначаются соответственно через $A \boxplus B$ и $A \boxminus B$. Результатом операции $A \boxplus B$ служит блок длины n, являющийся двоичным представлением числа $r(A)+r(B)$, взятым по модулю 2^n. Например, $(1101) \boxplus (1011) = (1000)$. Действительно, (1101) и (1011) — двоичные представления чисел 13 и 11, их сумма 24 по модулю 16 равна числу 8, которое представляется в виде (1000). Аналогично определяется операция вычитания $\boxminus$.

Циклический сдвиг влево. Он обозначается через $A <<< k$ и означает побитовый циклический сдвиг блока A влево на k битов. Например, $(11010000) <<< 3 = (10000110)$. Допустима также запись $A <<< D$, что означает сдвиг блока A влево на число бит, определяемое блоком D. Например, значением $(10111110) <<< (001)$ является вектор (01111101). Сдвиг влево был на одну позицию, так как (001) — двоичное представление числа 1. Аналогично $(01100000) <<< (101) = (00001100)$.

Объединение блоков (или *конкатенация*). Блок A можно объединить с блоком D и получить блок длины $n+m$. Будем обозначать его через AD.

Стандартное кодирование. Для компактности двоичные блоки часто представляют в шестнадцатеричной системе. Например, 32-битный блок (10101111100110110000001111010101) представляют в виде `af9b03d5`. Здесь каждый символ кодирует полубайт. Принято считать, что старшие биты располагаются слева, а младшие биты — справа. Так, например, блоку (1010) соответствует символ `a` в шестнадцатеричном представлении.

6.3 Сеть Фейстеля и SP-сеть

Основными разновидностями блочных шифров являются *сеть Фейстеля* и *подстановочно-перестановочная сеть* (SP-сеть).

Сетью Фейстеля называется блочный шифр, устроенный следующим образом. Пусть входной блок C_i каждого раунда делится на левую L_i и правую R_i половины. Тогда блок C_{i+1} получается объединением $L_{i+1} = R_i$ и $R_{i+1} = L_i \oplus F(R_i, K_{i+1})$, где F — *раундовая функция* и K_{i+1} — подключ данного раунда.

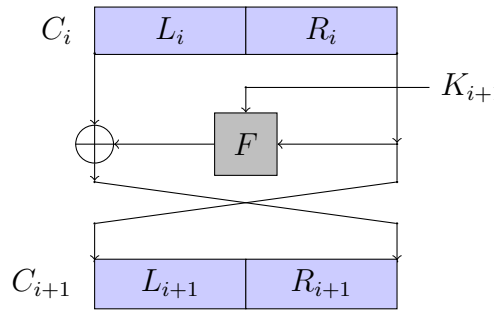
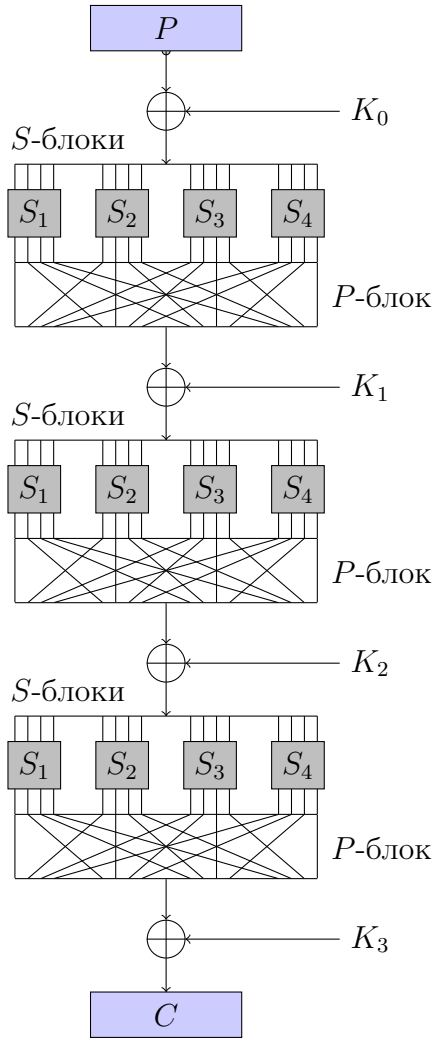


Схема Фейстеля

Сеть Фейстеля примечательна тем, что при её расшифровании не требуется обращать раундовую функцию F, т. е. находить функцию F^{-1}. Действительно, значения L_i и R_i могут быть восстановлены как $L_i = R_{i+1} \oplus F(L_{i+1}, K_{i+1})$ и $R_i = L_{i+1}$. Это свойство существенно. Оно означает, что в сети Фейстеля функция F может быть устроена как угодно сложно (обратная функция F^{-1} может даже не существовать!), что тем не менее не усложняет процедуру расшифрования по сравнению с шифрованием. Примерами сетей Фейстеля являются шифры DES, ГОСТ 28147-89, TwoFish, RC-6, MARS, Camellia.

SP-сетью (*подстановочно-перестановочной сетью*, Substitution-Permutation network) называется блочный шифр, в котором преобразование каждого раунда является комбинацией *подстановок* (S-блоков) и *перестановок* (P-блоков).



SP-сеть

S-блок (S-box, узел замены) — это отображение из множества двоичных векторов длины n в множество двоичных векторов длины m. При этом числа n и m относительно малы, например 4, 6, 8, 16, 32. Очень часто рассматриваются S-блоки, являющиеся взаимно однозначными преобразованиями (one-to-one substitution), в этом случае

$n = m$. S-блоки представляют собой основные нелинейные преобразования шифра. Как правило, криптостойкость всего шифра существенно зависит от их криптографических характеристик. Поэтому проектироваться S-блоки должны наиболее тщательно. Математически S-блок является *векторной булевой функцией* (см. раздел 3.3).

P-блок (P-box, permutation) — перестановка на множестве координат двоичного вектора длины n. Количество битов n, как правило, достаточно большое. Часто оно совпадает с длиной шифруемого блока. Например, равно 64, 128 и т. п.

На рисунке приведён пример трёхраундовой SP-сети, осуществляющей зашифрование блока открытого текста длины 16. В сети имеются четыре S-блока типа $4 \to 4$ и один P-блок $16 \to 16$.

В структуре SP-сети наглядно могут быть реализованы два фундаментальных принципа построения криптографических преобразований: *перемешивание* и *рассеивание* (confusion and diffusion), предложенные Клодом Шенноном в 1949 году (см. раздел 2.3). Напомним, что перемешивание означает усложнение всевозможных связей между открытым и шифрованным текстами. Локальную функцию перемешивания осуществляют S-блоки. Рассеивание заключается в распространении влияния одного бита открытого текста на большое число битов шифрованного текста. Локально рассеивание осуществляется с помощью P-блоков.

Другой отличительной особенностью SP-сетей является их *внутренний параллелизм* (inherent parallelism), позволяющий реализовывать их с меньшими вычислительными затратами по сравнению с сетями Фейстеля.

Примерами SP-сетей служат шифры IDEA, AES (Rijndael), Serpent.

6.4 DES — бывший стандарт США

Рассмотрим наиболее известный пример сети Фейстеля — шифр DES, бывший с 1980 по 1998 г. американским стандартом симметричного шифрования данных для защиты важной, но несекретной информации в государственных и коммерческих организациях.

В 1973 году Национальное бюро стандартов США объявило первый открытый конкурс на единый стандарт шифрования. К тому времени отсутствие такого стандарта уже привело к серьёзной проблеме несовместимости между различными средствами шифрования.

Конкурс был объявлен, а конкурсантом оказался только один шифр — Lucifer, разработанный фирмой IBM в 70-х годах. Он и послужил основой для DES. Этот шифр был детально проанализирован Агентством национальной безопасности США (АНБ или NSA — National Security Agency) и несколько изменен. В частности, была уменьшена длина ключа (со 128 до 56 битов) и были выбраны другие S-блоки. После изменений шифр получил название DES — Data Encryption Standard.

S-блоки являются единственными нелинейными компонентами этого шифра, от которых существенным образом зависит его криптостойкость. Замена S-блоков не была достаточно аргументирована АНБ. Официально считается, что она была сделана в целях повышения криптостойкости шифра. По другой версии — для вставки «закладок», существенно ослабляющих шифр для посвященных [80].

Штаб-квартира АНБ, штат Мэриленд, США

Из-за повышенной секретности Агентства национальной безопасности его аббревиатуру NSA иногда в шутку расшифровывают как «No such Agency». Да и само здание штаб-квартиры АНБ с чёрными зеркальными стенами словно растворяется в воздухе.

Перейдем к описанию шифра. DES осуществляет шифрование блоков длины 64 бита. Длина ключа составляет 56 бит. Число раундов равно 16. Последовательно приведём способ выбора раундовых ключей, общую схему шифрования DES и непосредственно саму раундовую функцию.

Раундовые ключи. Будем считать, что в каждом блоке биты нумеруются слева направо, начиная с первого бита. Ключ K обычно представляют блоком длины 64 бита, в котором каждый восьмой бит зависимый — он равен сумме по модулю 2 предыдущих семи битов плюс 1 (такие зависимые биты называются *проверками на чётность*). На подготовительном этапе из ключа K формируются 16 подключей $K_1, \ldots, K_{16}$ длины 48. А именно сначала по ключу K строятся блоки C_0 и D_0, состоящие каждый из 28 битов:

C_0	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
D_0	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

Например, первый бит блока C_0 — это 57-й бит ключа K, второй бит — 49-й бит ключа и т. д. слева направо и сверху вниз. Аналогично, первым битом блока D_0 является 63-й бит ключа K и т. д. Обратим внимание, что в блоки C_0 и D_0 не входят зависимые биты ключа, т. е. биты с номерами 8, 16, 24, 32, 40, 48, 56 и 64.

Далее, начиная с блоков C_0 и D_0, индуктивно строятся блоки C_i и D_i, где $i = 1, \ldots, 16$. Блоки C_i и D_i получаются из блоков C_{i-1} и D_{i-1} их циклическим сдвигом влево на один или два бита в зависимости от индекса i:

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

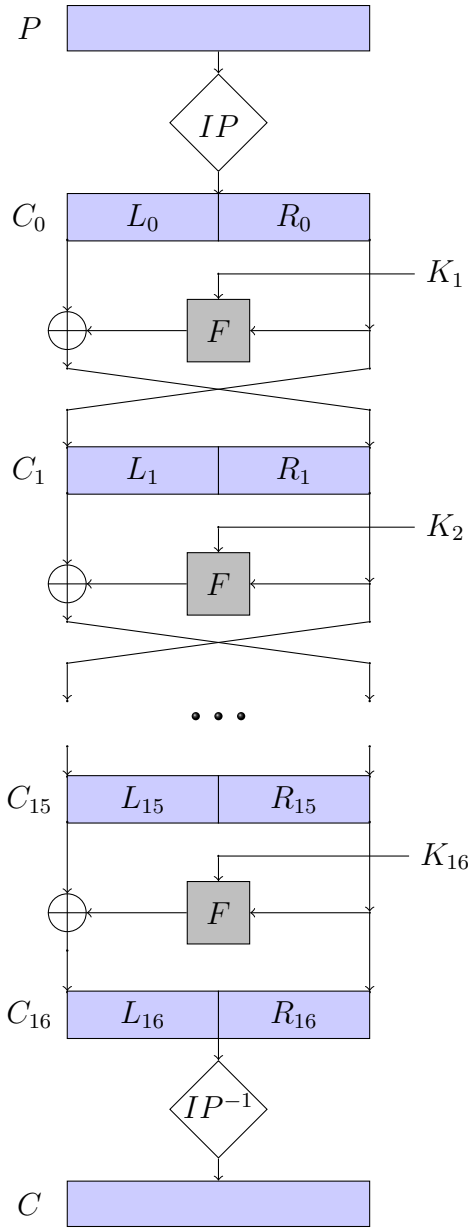
.

Затем подключ K_i для i-го раунда формируется из блока C_iD_i длины 56 выбором следующих 48 битов:

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

.

Таким образом, первый бит K_i — это 14-й бит блока C_iD_i и т. д. После того как получены 16 раундовых подключей можно переходить к шифрованию.



Шифр DES

Шифрование DES заключается в следующем. Блок открытого текста P подается на вход начальной перестановки IP. После перестановки 58-й бит открытого текста становится первым битом, 50-й — вторым и т. д. слева направо и сверху вниз.

IP — начальная перестановка DES

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

Полученный блок разбивается на левую и правую половины L_0 и R_0. Далее для $i = 1, \ldots, 15$ итеративно вычисляются блоки L_i и R_i по схеме Фейстеля: $L_i = R_{i-1}$ и $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$. Раундовая функция F будет определена ниже. Последний, шестнадцатый, раунд проходит несколько иначе: в нём левый и правый блоки не меняются местами, $L_{16} = L_{15} \oplus F(R_{15}, K_{16})$ и $R_{16} = R_{15}$.

Затем к блоку $L_{16}R_{16}$ применяется обратная к начальной перестановка IP^{-1}. В результате будет получен шифртекст. Так, первым битом шифртекста становится 40-й бит блока $L_{16}R_{16}$ и т. д.

IP^{-1} — обратная перестановка DES

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

Отметим, что процедуры шифрования и расшифрования для DES совпадают. Только раундовые ключи необходимо использовать в обратном порядке.

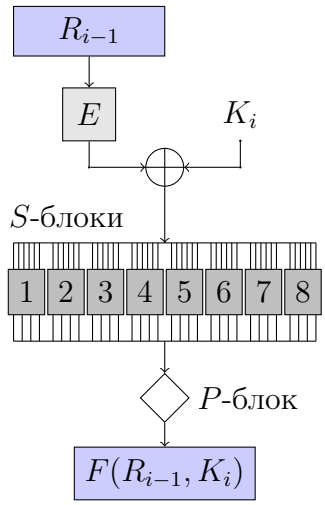
Раундовая функция. На каждом раунде DES используется одна и та же раундовая функция F. Приведём её описание. На вход функции подается правая часть R_{i-1} промежуточного шифртекста (блок из 32 битов) и подключ K_i i-го раунда (блок из 48 битов). На выходе функции F будет получен блок длиной 32 бита.

Сначала к блоку R_{i-1} длины 32 применяется расширяющая подстановка E, переводящая его в блок длины 48.

E — расширяющая подстановка DES

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

Так, первым битом расширенного блока будет 32-й бит блока R_{i-1} и т. д. При этом некоторые биты входного блока (например, с номерами 32, 1, 4, 5 и др.) дублируются. Полученный блок складывается побитово с подключом K_i.



Раундовая функция DES

Новый блок разбивается на восемь блоков длины 6, каждый из которых подается на вход соответствующего S-блока. Выходом каждого S-блока является вектор длины 4. Объединением выходов всех S-блоков получается новый блок длины 32, к которому применяется перестановка (P-блок).

Перестановка в P-блоке DES

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

Полученный блок является выходом раундовой функции.

S-блоки DES

	0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
00	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
01	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
10	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
11	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
00	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
01	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
10	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
11	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
00	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
01	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
11	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
00	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
01	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
10	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
00	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
01	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
10	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
11	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
00	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
01	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
10	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
11	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
00	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
01	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
10	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
11	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Все S-блоки шифра DES приведены выше. Они представлены в следующем виде. Пусть $(a_1, a_2, a_3, a_4, a_5, a_6)$ — вход некоторого S-блока. Значения битов a_1, a_6 определяют строку в таблице, а значения битов a_2, a_3, a_4, a_5 определяют некоторый её столбец. Двоичное

представление (b_1, b_2, b_3, b_4) числа, находящегося на пересечении выбранных строки и столбца, является выходом S-блока. Например, если на вход третьего S-блока подается вектор (100101), то соответствующим выходом будет вектор (1101). Действительно, в таблице выбираем третью часть (она относится к третьему S-блоку), в ней строку, определяемую значениями 11, и столбец 0010. На пересечении строки и столбца находится число 13. Его двоичным представлением и является вектор (1101).

Режимы работы. Шифр DES, как и любой блочный шифр, может использоваться в разных режимах. Обозначим через $\mathrm{DES}(P, K)$ результат применения алгоритма DES к блоку P при ключе K. Пусть $P_1, \ldots, P_n$ — последовательность открытых текстов, подлежащих шифрованию. Кратко опишем, как формируются соответствующие шифртексты $C_1, \ldots, C_n$ в разных режимах.

• ECB — *электронная кодовая книга* (Electronic Code Book). Это обычный режим, при котором $C_i = \mathrm{DES}(P_i, K)$. Основное достоинство режима — простота реализации. Недостаток — одинаковые блоки открытого текста шифруются одинаково, что приводит к возможности «криптоанализа со словарем».

• CBC — *сцепление блоков шифртекста* (Cipher Block Chaining). В этом режиме $C_i = \mathrm{DES}(C_{i-1} \oplus P_i, K)$ при $i = 1, \ldots, n$, где C_0 — некоторый начальный вектор. Сцепление выражается в том, что каждый следующий шифртекст зависит не только от открытого текста, но и от предыдущего шифртекста. Так устраняется недостаток первого режима.

• CFB — *обратная связь по шифртексту* (Cipher Feed Back). В этом случае $C_i = \mathrm{DES}(C_{i-1}, K) \oplus P_i$ при $i = 1, \ldots, n$, где C_0 — некоторый начальный вектор. Режим напоминает поточное шифрование. Он заключается в выработке гаммы и побитового наложения её на последовательность открытых текстов. Похож на предыдущий режим тем, что блоки в гамме сцеплены.

• OFB — *обратная связь по выходу* (Output Feed Back). На открытые тексты также накладывается гамма, $C_i = Z_i \oplus P_i$, где $Z_i = \mathrm{DES}(Z_{i-1}, K)$ при $i = 1, \ldots, n$ и Z_0 — начальный вектор. Отличие от режима CFB состоит в том, что вырабатываемая гамма не зависит от последовательности открытых текстов. В этом случае требуется отдельный анализ качества гаммы при различных начальных векторах Z_0. Для надёжного шифрования период гаммы должен быть достаточно большим.

Использование. Шифр DES получил очень широкое международное распространение. Он был реализо́ван в большинстве коммерческих продуктов, а также в приложениях почти всех правительственных агентств США. Однако слишком малая длина ключа шифра DES стала причиной того, что в 1998 году он утратил статус стандарта. Незадолго до этого сотрудники компании RSA Laboratory дешифровали DES менее чем за три дня, используя специальный суперкомпьютер DES Cracker. Они успешно осуществили так называемую *атаку грубой силой* (brute-force attack), основанную на полном переборе 2^{56} возможных вариантов ключа. Смене статуса шифра DES предшествовали также теоретические методы его криптоанализа, такие как дифференциальный (E. Biham, A. Shamir, 1991, [86]) и линейный (M. Matsui, 1993, [126]), которые будут рассмотрены далее. Вместо DES было предложено использовать Triple DES (или 3DES) — трёхкратное шифрование DES с помощью различных ключей. Таким образом, длина ключа в 3DES была увеличена до $168 = 3 \times 56$ битов. Этот медленный шифр и сейчас считается достаточно стойким. Он используется в банковских операциях, например в системах обслуживания кредитных карт VISA, во многих Интернет-приложениях, таких как PGP и др.

Задача 46. Подумайте, почему на практике не применяется Double DES — двукратное шифрование DES с разли́чными ключами.

6.5 Российский стандарт ГОСТ 28147-89

ГОСТ — один из наиболее загадочных шифров, возникший где-то в 70-х годах в одном из закрытых научно-исследовательских институтов СССР, подведомственных Восьмому главному управлению КГБ. Авторы его неизвестны. С 1990 года этот шифр является государственным стандартом России. Это означает, что во всех государственных организациях РФ и в коммерческих организациях, имеющих лицензию ФСБ, в качестве алгоритма шифрования (если он требуется) *должен* использоваться ГОСТ. При этом в стандарте не накладывается ограничений на степень секретности защищаемой информации. Предполагается, что она́ может быть любой.

Полностью ГОСТ 28147-89 был рассекречен и опубликован только в 1994 году. Приведём его описание. ГОСТ является классической

сетью Фейстеля и осуществляет шифрование блоков длиной 64 бита. Длина ключа в ГОСТ равна 256 бит. Число раундов шифрования равно 32.

С 1919 по 1991 гг. главное здание органов госбезопасности СССР и РСФСР на Лубянке в Москве

Раундовые ключи. На каждом раунде требуется один раундовый ключ длиной 32 бита. Выбор раундовых ключей в ГОСТе очень прост. Исходный ключ K, состоящий из 256 бит, разбивается на восемь блоков по 32 бита каждый:

$$K = K^{(1)}K^{(2)}K^{(3)}K^{(4)}K^{(5)}K^{(6)}K^{(7)}K^{(8)}.$$

Последовательность из тридцати двух раундовых ключей выглядит следующим образом:

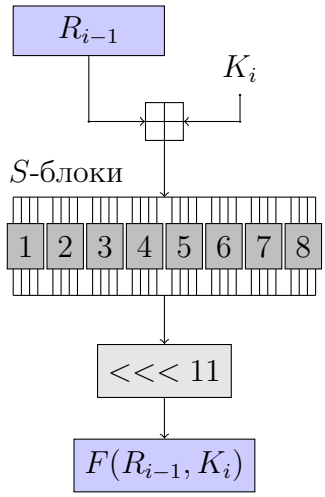
$$K^{(1)}, \ldots, K^{(8)}, K^{(1)}, \ldots, K^{(8)}, K^{(1)}, \ldots, K^{(8)}, K^{(8)}, \ldots, K^{(1)},$$

т. е. первым раундовым ключом будет $K^{(1)}$, вторым — $K^{(2)}$ и т. д., а последним раундовым ключом — снова $K^{(1)}$.

Шифрование ГОСТ. Блок открытого текста разбивается на левую и правую половины L_0 и R_0. Далее для $i = 1, \ldots, 31$ итеративно вычисляются блоки L_i и R_i по схеме Фейстеля: $L_i = R_{i-1}$ и

$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$. Раундовая функция F определяется ниже. Последний, тридцать второй, раунд проходит немного иначе: в нём левый и правый блоки не меняются местами, $L_{32} = L_{31} \oplus F(R_{31}, K_{32})$ и $R_{32} = R_{31}$.

Раундовая функция. На вход раундовой функции на i-м раунде, $i = 1, \ldots, 32$, подается блок R_{i-1} длиной 32 бита и i-й раундовый ключ K_i также длины 32.



Раундовая функция ГОСТ

Сначала блок R_{i-1} складывается с блоком K_i по модулю 2^{32}. Биты полученного вектора $R_{i-1} \boxplus K_i$ разбиваются на восемь групп по четыре бита в каждой. Биты j-й группы подаются на вход j-го S-блока, выходом которого являются новые четыре бита. Выходы всех S-блоков объединяются в новый 32-битный блок. Затем этот блок циклически сдвигается влево на 11 позиций. Полученный блок является выходом раундовой функции.

Особенностью ГОСТа является то, что в стандарте не указан конкретный вид его S-блоков. Предполагается, что для различных систем компетентные органы могут предложить к использованию разные наборы S-блоков. Разные в том числе и по своим криптографическим свойствам. Так, шифрование по единому государственному стандарту в одних организациях может оказаться более сильным, в других — более слабым.

Использование. ГОСТ не представляет собой сверхбыстрый программный шифр, но с точки зрения аппаратной реализации его конструкция очень удачна даже по современным оценкам. Большая длина ключа придает шифру достаточный иммунитет против атаки грубой силой. В России этот алгоритм до сих пор считается криптографически стойким. Но согласны с этим не все. Например, разработчики свободно распространяемой библиотеки криптографических схем `Crypto++` относят ГОСТ к категории «insecure or obsolescent algorithms retained for backwards compatibility and historical value» (небезопасных или устаревших алгоритмов, поддерживаемых для совместимости или имеющих историческое значение) [103]. И всё же практически реализованного криптоанализа этого шифра нет.

За более чем 20 лет существования стандарта было предложено много работ по его анализу, но в реальности ни одна из них серьёзной угрозы ГОСТу пока не представила. В 2000 году для 13-раундового шифра ГОСТ был предложен дифференциальный криптоанализ [137]. Интересно, что в 2001 году появилась работа [139], в которой доказывается стойкость ГОСТа к дифференциальному криптоанализу уже после семи раундов. Для 30 раундов ГОСТа рассматривались рефлексивные и слайдовые атаки. В ряде работ описаны классы слабых ключей ГОСТа, но их число $2^{128} + 2^{224}$ пренебрежительно мало по сравнению с объёмом всего ключевого пространства. Предложен также ряд атак, использующих связанные ключи. Среди них выделяется атака, в которой с помощью 18 определённым образом связанных ключей, удается определить весь ключ ГОСТа за 2^{26} операций. Но предположение о наличии связанных ключей является очень сильным и практически не реализуемым.

В 2010 году ГОСТ был заявлен в качестве участника конкурса Международной организации по стандартизации (ISO) на приобретение статуса Всемирного стандарта шифрования (Worldwide Industrial Encryption Standard). Осенью 2011 года ГОСТ находился в процессе стандартизации, а именно — на стадии рассылки комментариев. В 2012 году статус шифра был изменен на deleted. Документацию можно найти на сайте `www.iso.org`, см. ISO/IEC 18033-3:2010/WD Amd 1. По-видимому, с этой попыткой стандартизации и связан большой всплеск интереса к ГОСТу в последнее время. В 2011 году на него заявлены по крайней мере две серьёзные атаки:

- Рефлексивная встреча посередине (T. Isobe, материалы конференции FSE, 2011 [120]).

• Дифференциальный криптоанализ (N. Courtois, M. Misztal, см. `http://eprint.iacr.org/2011/312` [101]).

По поводу первой атаки (T. Isobe) приведём комментарии [19] экспертов сайта `www.cryptofaq.ru`: «И вот только в 2011 году появились исследования [120], действительно снижающие стойкость ГОСТа. Предложена новая атака на блочные шифры: «рефлексивная встреча посередине» (reflection-meet-in-the-middle). Отметим, что применение этой атаки к ГОСТу потребовало существенной модификации этапа́ встречи посередине путём использования эквивалентных ключей для четырёх итераций. Параметры предложенной атаки составляют 2^{225} операций, 2^{32} известных пар блоков открытого/шифртекста и 2^{64} ячеек памяти. Каковы последствия появления данной атаки? Во-первых, стойкость ГОСТа снижена, пусть и незначительно: с 2^{256} до 2^{225}. Но всё же это говорит о том, что ГОСТ более слабый шифр, нежели предполагалось ранее. В то же время, предложенная атака имеет исключительно теоретическое значение. Таким образом, с точки зрения практических аспектов использования ГОСТа не произошло ровным счётом ничего. А вот попытки стандартизации ГОСТа в организации ISO могут оказаться под угрозой.».

Вторая атака — дифференциальный криптоанализ *полнораундового* ГОСТа — заявлена в июне 2011 года в работе [101]. В работе, однако, пока не приводится точных характеристик быстродействия метода, но утверждается, что «GOST is NOT SECURE even against differential cryptanalysis». Во всем этом ещё предстоит разобраться в ближайшем будущем. Дальнейшее развитие дифференциального криптоанализа для ГОСТ на основе сгруппированных разностей приводится в [102]. При этом в качестве использующейся слабости алгоритма ГОСТ отмечают [129] *постоянный* сдвиг влево на 11 позиций в раундовой функции.

На конференции FSE в 2012 году была представлена новая работа по криптоанализу ГОСТа «Improved Attacks on Full GOST» (авторы I. Dinur, O. Dunkelman, A. Shamir), в которой стойкость ГОСТа была снижена до 2^{196}. В марте 2012 года появилась статья «An Improved Differential Attack on Full GOST» (автор N. Courtois) [100], в которой заявлено снижение стойкости полнораундового ГОСТа до 2^{178}.

6.6 Канадский шифр CAST-128

Этот блочный шифр предложили в 1997 году канадские криптографы **C**arlisle **A**dams (Университет Оттавы) и **S**tafford **T**avares (Университет Куинс). Именно этот шифр был поддержан правительством Канады в качестве официальной замены шифру DES. И хотя шифр CAST-128 запатентован компанией Entrust Technologies, его можно бесплатно использовать для коммерческих и некоммерческих целей во всем мире. В настоящее время CAST-128 широко распространен. Например, это шифр по умолчанию в некоторых версиях

программ GPG и PGP. Кроме того, CAST-128 — предшественник известного шифра CAST-256, участника конкурса AES.

CAST-128 является сетью Фейстеля. Шифрование осуществляется блоками по 64 бита. Длина ключа равна 128 бит[1], а число раундов равно 16.

Университет Оттавы, Канада

Queens University, Канада

[1]На самом деле размер ключа может варьироваться от 40 до 128 бит с шагом 8 бит. Все короткие ключи расширяются до 128 битов приписыванием нулей. Число раундов равно 12 (при длине ключа $\leqslant 80$ бит) или 16 (если длина ключа > 80 бит).

Раундовые ключи. Сначала из ключа K формируются 16 пар 32-битовых раундовых ключей $\{Km_i, Kr_i\}$. Ключи Km_i называются *маскирующими*, а Kr_i — *вращательными* (от слов masking и rotate). Способ получения этих ключей из первоначального 128-битного ключа приводится в приложении 10.4. Отметим, что при шифровании требуются только младшие пять битов подключей Kr_i.

Шифрование состоит в следующем. Входной блок разбивается на левую и правую половины, L_0 и R_0 — блоки длины 32 бита. Далее на каждом i-м раунде, $i = 1, \ldots, 16$, итеративно вычисляются $L_i = R_{i-1}$ и $R_i = L_{i-1} \oplus F_i(R_{i-1}, Km_i, Kr_i)$. Выходным блоком служит пара $R_{16}L_{16}$.

Раундовая функция F_i в шифре CAST зависит от номера раунда и может быть одного из трёх типов. На первом раунде используется первый тип, на втором — второй и т. д. с простым чередованием. На вход F_i подается правая часть R_{i-1} промежуточного шифртекста (блок из 32 битов) и пара подключей $\{Km_i, Kr_i\}$ для i-го раунда (блоки по 32 бита каждый). На выходе функции F_i будет получен блок длиной 32 бита.

Тип 1 определяется для раундов $i = 1, 4, 7, 10, 13, 16$. Сначала вычисляется вспомогательный 32-битовый блок

$$I = ((Km_i \boxplus R_{i-1}) <<< Kr_i).$$

Напомним, что сложение $\boxplus$ здесь рассматривается по модулю 2^{32}, а операция $<<<$ означает побитовый циклический сдвиг влево (см. подробнее раздел 6.2). Пусть I_a, I_b, I_c, I_d — соответственно первый, второй, третий и четвёртый байты блока I. Тогда выходом раундовой функции F_i служит блок

$$F_i = \big((S_1(I_a) \oplus S_2(I_b)) \boxminus S_3(I_c)\big) \boxplus S_4(I_d).$$

Здесь S_1, S_2, S_3 и S_4 — операции подстановки байтов с помощью S-блоков типа $8 \to 32$. Таблицы для них приводятся в приложении.

Тип 2 для раундов $i = 2, 5, 8, 11, 14$. Функция имеет вид

$$I = ((Km_i \oplus R_{i-1}) <<< Kr_i),$$

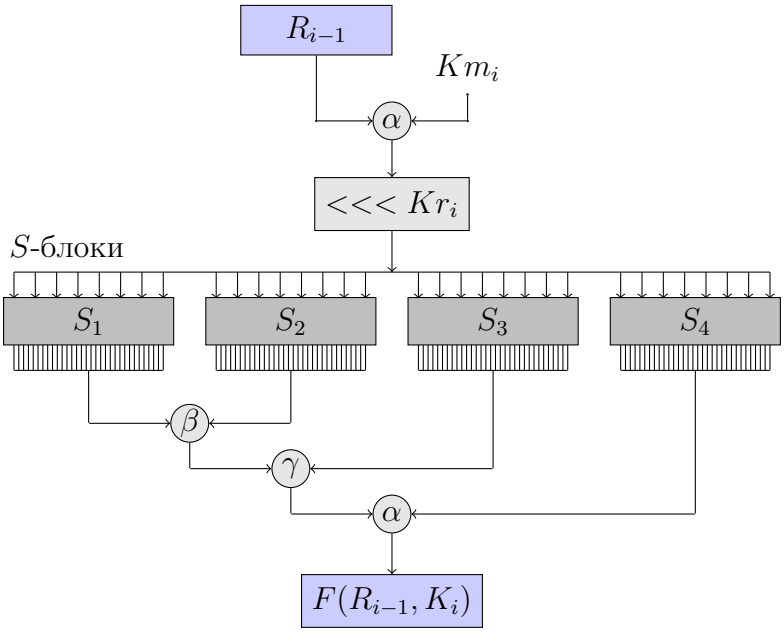
$$F_i = \big((S_1(I_a) \boxminus S_2(I_b)) \boxplus S_3(I_c)\big) \oplus S_4(I_d).$$

Тип 3 для раундов $i = 3, 6, 9, 12, 15$. Функция определяется как

$$I = ((Km_i \boxminus R_{i-1}) <<< Kr_i),$$

$$F_i = \big((S_1(I_a) \boxplus S_2(I_b)) \oplus S_3(I_c)\big) \boxminus S_4(I_d).$$

Несложно заметить, что все типы отличаются друг от друга лишь разным порядком операций $\boxplus$, $\boxminus$ и $\oplus$. На следующем рисунке приводится общий вид раундовой функции для CAST.



$\boxplus$, $\oplus$, $\boxminus$ для $i = 1, 4, 7, 10, 13, 16$

$\alpha\ \beta\ \gamma$ означают $\oplus$, $\boxminus$, $\boxplus$ для $i = 2, 5, 8, 11, 14$

$\boxminus$, $\boxplus$, $\oplus$ для $i = 3, 6, 9, 12, 15$

Булевы функции в шифре CAST. Каждый S-блок шифра CAST — это векторная булева функция, $S_j : \mathbb{Z}_2^8 \to \mathbb{Z}_2^{32}$. Удобно представить её в виде набора 32 обычных булевых функций $f_k^{(j)}$, а именно

$$S_j(x_1, \ldots, x_8) = (y_1, \ldots, y_{32}),\ j = 1, \ldots, 4,$$

где

$$y_k = f_k^{(j)}(x_1, \ldots, x_8),\ k = 1, \ldots, 32.$$

В шифре CAST все функции $f_k^{(j)}$ являются бент-функциями, т. е. обладают самыми сильными свойствами нелинейности (см. подробнее главу 9). Кроме того, для каждого S-блока они подобраны так,

что любая их линейная комбинация также имеет достаточно «хорошие» свойства нелинейности, хоть и не является бент-функцией в общем случае. Такой способ построения S-блоков придает шифру CAST стойкость к линейному криптоанализу (см. раздел 8.4). Подбор S-блоков для CAST совершался на обычном компьютере за несколько недель его работы. Подробнее о построении S-блоков CAST можно прочитать в статье [83].

6.7 AES — текущий стандарт США

Конкурс AES. В 1997 году Институт стандартизации США объявил конкурс на новый стандарт блочного шифра. Было рассмотрено 15 заявок из разных научно–исследовательских институтов всего мира. В финал вышли пять блочных шифров. Перечислим их с именами создателей:

MARS — C. Burwick и др. (фирма IBM, США);
RC6 — R. Sidney, R. Rivest, M. Robshaw, Y. Yin (RSA Security, EMC Corporation, США);
Rijndael — J. Daemen, V. Rijmen (лаборатория COSIC, Katholieke Universiteit Leuven, Бельгия);
Serpent — R. Anderson (University of Cambridge, Великобритания), E. Biham (Technion, Израиль), L. Knudsen (Technical University of Denmark, Дания);
Twofish — B. Schneier и др. (Counterpane Internet Security, Inc., США).

Все эти шифры были признаны криптографически стойкими. Победителем конкурса стал шифр Rijndael (произносится «Рэйндал»), создатели которого — молодые бельгийские учёные из Католического университета г. Лёвена — 32-летний Йоан Даймен и 27-летний Винсент Рэймен. После небольшой доработки шифр был принят 26 мая 2002 года в качестве нового стандарта шифрования США. Он получил официальное название AES — Advanced Encryption Standard.

AES представляет собой классическую SP-сеть. Длина шифруемого блока в AES равна 128 бит, длина ключа может составлять 128, 192 или 256 бит. В зависимости от длины ключа выделяют AES-128, AES-192 и AES-256. Число раундов шифрования в них соответственно равно 10, 12 и 14.

Йоан Даймен и Винсент Рэймен — создатели Rijndael

Рассмотрим случай, когда длина ключа равна 128 бит, а число раундов равно 10. Каждый промежуточный шифртекст удобно называть *состоянием* и представлять его в виде массива байтов 4×4:

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

Для 10-раундового шифрования требуется 11 раундовых подключей длины 128. О том, как выбирать раундовые ключи из секретного ключа, будет рассказано далее. Каждый раундовый ключ также удобно представлять в виде матрицы:

k_{00}	k_{01}	k_{02}	k_{03}
k_{10}	k_{11}	k_{12}	k_{13}
k_{20}	k_{21}	k_{22}	k_{23}
k_{30}	k_{31}	k_{32}	k_{33}

Все блоки a_{ij}, k_{ij} состоят из восьми битов. Вначале блок открытого текста складывается по модулю 2 с нулевым раундовым подключом. Полученный блок C_0 подается на вход первого раунда.

Шифрование AES заключается в многократном применении раундового преобразования к блоку C_0. Выход C_i каждого i-го раунда является входом следующего и т. д. Каждый i-й раунд, где $i = 1, \ldots, 9$, состоит из четырёх простых преобразований:

SubByte(C_{i-1}) — побайтовая подстановка с помощью S-блока;

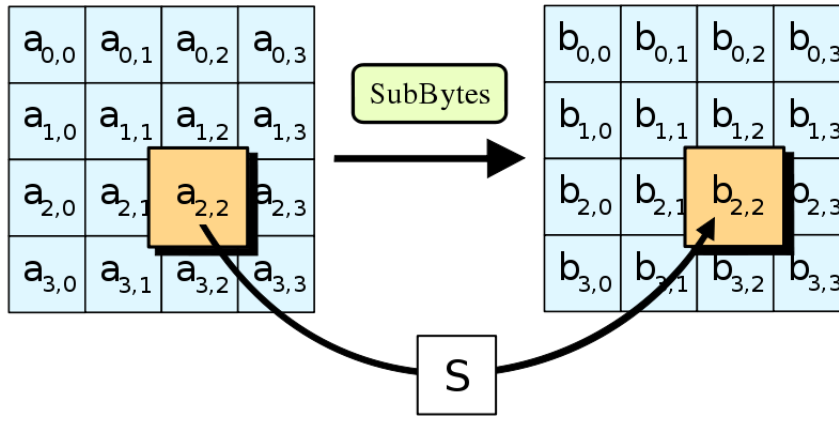
ShiftRows(C_{i-1}) — циклический сдвиг байтов состояния;

MixColumns(C_{i-1}) — домножение столбцов состояния;

AddRoundKey(C_{i-1}, K_i) — сложение по модулю 2 с раундовым ключом.

Последний 10-й раунд отличается от предыдущих только тём, что преобразование MixColumns() на нём не выполняется. Рассмотрим все преобразования более подробно.

SubByte. Каждый байт состояния заменяется на другой байт с помощью S-блока $8 \to 8$.



Этот S-блок фиксирован и известен. Он определяется приведённой ниже таблицей. В ней используется шестнадцатеричное представление: каждый байт представляется в виде двухразрядного шестнадцатеричного числа. Например, байты (00001000), (00111110) представ-

ляются числами `08`, `3e` и заменяются в S-блоке на числа `30`, `b2`, которые соответствуют байтам (00110000), (10110010). Другие примеры:

$$(00010010) = \texttt{12} \longrightarrow \textbf{S-box} \longrightarrow \texttt{c9} = (11001001)$$
$$(11111101) = \texttt{fd} \longrightarrow \textbf{S-box} \longrightarrow \texttt{54} = (01010100)$$

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

S-блок $8 \to 8$ **шифра AES**

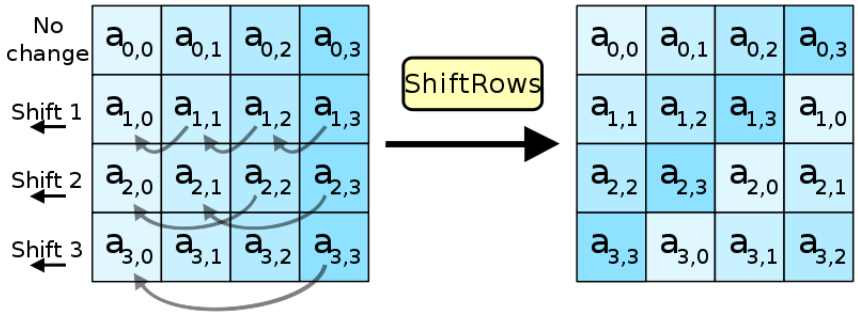
Отметим, что приведённая таблица замены не представляет собой хаотический набор шестнадцатеричных чисел и, конечно, возникла не случайно. Преобразование, осуществляемое S-блоком, имеет красивое алгебраическое представление.

А именно пусть

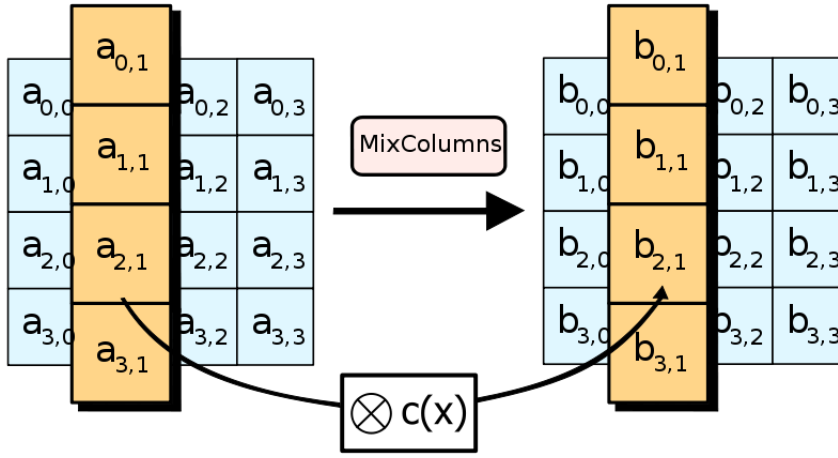
$$A = \begin{pmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{pmatrix}, \quad y = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

Установим соответствие между байтами и элементами поля Галуа $GF(2^8)$, построенного с помощью неприводимого многочлена $g(x) = x^8+x^4+x^3+x+1$. Напомним, что такое соответствие устанавливается просто: байту $(c_7c_6c_5c_4c_3c_2c_1c_0)$ соответствует многочлен $c_7x^7+c_6x^6+c_5x^5+c_4x^4+c_3x^3+c_2x^2+c_1x+c_0$. Например, байту (11100001) отвечает многочлен $x^7+x^6+x^5+1$. Будем интерпретировать S-блок как подстановку на элементах поля: каждый ненулевой элемент поля β заменяется на элемент $A\beta^{-1}+y$. Элемент 0 заменяется на y. Нелинейность S-блока (а по сути и всего шифра AES) «спрятана» в использовании обратного элемента β^{-1} вместо β. Все вычисления производятся по модулю выбранного неприводимого многочлена. Однако при первом знакомстве с AES можно не вникать в алгебраическое представление его S-блока и пользоваться приведённой выше таблицей.

ShiftRows. На следующем этапе байты состояния циклически сдвигаются: i-я строка байтов, где $i = 0, 1, 2, 3$, циклически сдвигается влево на i позиций.



MixColumns. Это преобразование заключается в замене каждого j-го столбца байтов на новый столбец,

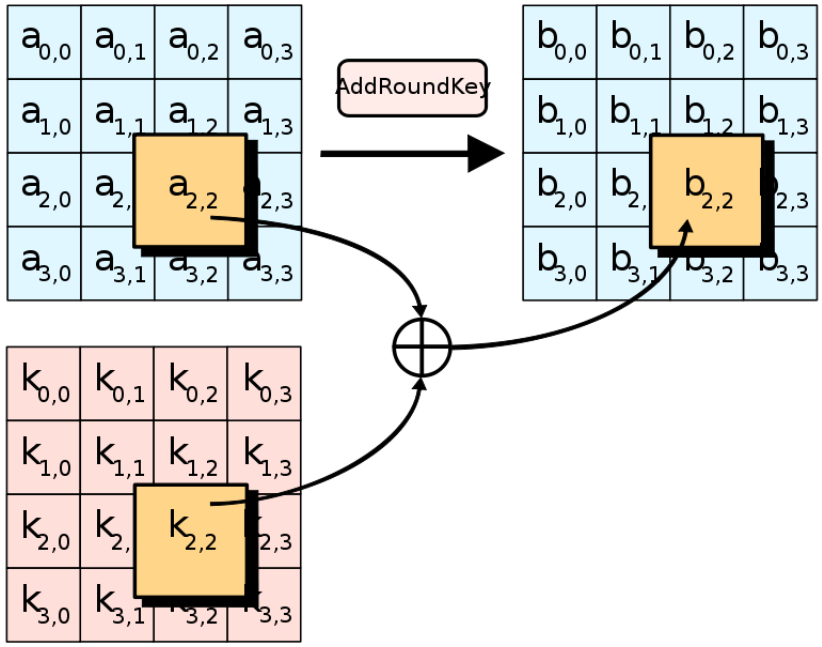


который получается следующим образом:

$$\begin{pmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} a_{0,j} \\ a_{1,j} \\ a_{2,j} \\ a_{3,j} \end{pmatrix},$$

где каждый элемент матрицы — это шестнадцатеричное представление соответствующего элемента поля $GF(2^8)$. Умножение и сложение рассматриваются как операции в поле.

AddRoundKey. Пожалуй, самое простое преобразование: биты состояния складываются по модулю 2 с соответствующими битами раундового ключа.



Раундовые ключи. Напомним, что для 10-раундового AES требуется 11 раундовых ключей длины 128. Один — для первоначального сложения с блоком открытого текста и десять — собственно для раундов. Общая длина всех раундовых ключей составляет 1 408 битов. Из ключа K построим *расширенный ключ* — новую последовательность битов именно такой длины.

Для этого ключ K разобьём на четыре блока $K = (w_0, w_1, w_2, w_3)$. Каждый блок w_i имеет длину 32 бита; удобно называть его *словом*. Далее итеративно определим слова $w_4, \ldots, w_{43}$ по некоторому правилу (см. ниже) и получим расширенный ключ $K' = (w_0, \ldots, w_{43})$ длины $44 \times 32 = 1408$ бит. Разделяя K' на блоки длины 128 бит, получаем раундовые ключи K_i, где $i = 0, 1, \ldots, 11$. Другими словами, $K' = (K_0, K_1, \ldots, K_{10})$, или иначе $K_i = (w_{4i}, w_{4i+1}, w_{4i+2}, w_{4i+3})$.

Осталось итеративно определить w_j, где $j = 4, \ldots, 43$:

$$w_j = \begin{cases} w_{j-4} \oplus \mathrm{SubWord}(\mathrm{RotWord}(w_{j-1})) \oplus v_{j/4}, & \text{если } j \text{ делится на 4.} \\ w_{j-4} \oplus w_{j-1}, & \text{иначе.} \end{cases}$$

Поясним определение. Преобразование $\mathrm{RotWord}(w_{j-1})$ заключается в побайтовом циклическом сдвиге влево на один байт, т. е. если $w_{j-1} = (d_0, d_1, d_2, d_3)$, то $\mathrm{RotWord}(w_{i-1}) = (d_1, d_2, d_3, d_0)$.

Преобразование $\mathrm{SubWord}(w_{i-1})$ состоит в замене каждого байта слова w_{i-1} на новый с помощью S-блока из функции SubBytes.

Слова v_n длины 32 бита, где $n = 1, \ldots, 11$, представляют собой константы и определяются следующим образом: $v_n = (z_n, 00, 00, 00)$, где байт z_n принимает значения из массива

$$\{\texttt{01, 02, 04, 08, 10, 20, 40, 80, 1b, 36, 6c}\}.$$

Мы используем шестнадцатеричное представление. Например, $v_{11} = (\texttt{6c}, \texttt{00}, \texttt{00}, \texttt{00}) = (0110\ 1100\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000)$.

Таким образом, раундовые ключи определены. Необходимо отметить, что при реализации шифрования и при выборе раундовых ключей нужно строго следовать стандарту. Любое отклонение от него не позволит утверждать, что реализован именно AES. Это важно.

Для **расшифрования** AES требуется провести обратные преобразования, применяя раундовые ключи в обратном порядке. Обратным преобразованием для MixColumns служит умножение каждого столбца байтов на обратную матрицу, т. е.

$$\begin{pmatrix} a_{0,j} \\ a_{1,j} \\ a_{2,j} \\ a_{3,j} \end{pmatrix} = \begin{pmatrix} \texttt{0e} & \texttt{0b} & \texttt{9d} & \texttt{09} \\ \texttt{09} & \texttt{0e} & \texttt{0b} & \texttt{0d} \\ \texttt{0d} & \texttt{09} & \texttt{0e} & \texttt{0b} \\ \texttt{0b} & \texttt{0d} & \texttt{09} & \texttt{0e} \end{pmatrix} \cdot \begin{pmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{pmatrix}.$$

Для преобразования, обратного к SubByte, следует использовать инверсный S-блок.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
10	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
20	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
30	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
40	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
50	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
60	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
70	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
80	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
90	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a0	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b0	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c0	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d0	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e0	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f0	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Инверсный S-блок $8 \to 8$ шифра AES

Получить обратные преобразования для ShiftRows и AddRoundKey не представляет труда.

Подробнее об особенностях шифра Rijndael можно прочитать в книге его авторов «The Design of Rijndael: AES — The Advanced Encryption Standard» [105].

6.8 Учебный шифр S-AES

В 2002 году в журнале «Cryptologia» появилась статья [130], в которой вниманию читателей предлагался упрощенный вариант шифра AES, так называемый S-AES (Simplified AES). Этот шифр, в деталях сильно напоминающий американский стандарт, можно использовать в учебных целях, пробуя на нём различные методы криптоанализа. Авторы статьи так и называют его «a testbed for cryptanalysis students».

Рассмотрим этот шифр. Длина шифруемого блока в нём равна 16 битам, длина ключа может быть равна 16 или 48 битам. В шифре имеется всего два раунда.

Каждый промежуточный шифртекст называем *состоянием* и представляем его в виде массива полубайтов 2×2.

a_{00}	a_{01}
a_{10}	a_{11}

Каждый раундовый ключ также удобно представлять в таком виде.

k_{00}	k_{01}
k_{10}	k_{11}

Все блоки a_{ij}, k_{ij} состоят из четырёх битов. Для двухраундового шифрования требуется три раундовых подключа K_0, K_1, K_2 длины 16. Первый вариант — раундовые ключи являются разными частями 48-битного секретного ключа. Второй вариант — все раундовые ключи совпадают с 16-битным секретным ключом[2].

[2]Есть и более нетривиальный способ выбора раундовых ключей, схожий с их выбором в AES. Мы не приводим здесь этот способ, его можно найти, например, в [16].

Вначале блок открытого текста складывается по модулю 2 с раундовым ключом K_0. Полученный блок C_0 подается на вход первого раунда. Выход первого раунда является входом второго. Каждый раунд состоит из четырёх простых преобразований:

SubHalfByte — подстановка на каждом полубайте с помощью S-блока $(9, 14, 5, 1, 8, 11, 13, 10, 6, 7, 15, 3, 12, 4, 0, 2)$.

ShiftRows — циклический сдвиг полубайтов состояния, а именно

$$\begin{array}{|c|c|} \hline a_{00} & a_{01} \\ \hline a_{10} & a_{11} \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|} \hline a_{00} & a_{01} \\ \hline a_{11} & a_{10} \\ \hline \end{array}.$$

MixColumns — домножение столбцов состояния. Это преобразование заключается в замене каждого j-го столбца байтов на новый столбец, который получается следующим образом:

$$\begin{pmatrix} b_{0,j} \\ b_{1,j} \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} a_{0,j} \\ a_{1,j} \end{pmatrix} = \begin{pmatrix} x \cdot a_{0,j} + x \cdot a_{1,j} + a_{0,j} \\ x \cdot a_{0,j} + x \cdot a_{1,j} + a_{1,j} \end{pmatrix},$$

где элементы $a_{i,j}$ надо представить как многочлены от x с двоичными коэффициентами, а умножение провести по модулю многочлена $g(x) = x^4 + x + 1$. Замена полубайта на многочлен проводится очень просто. Например, полубайт (0111) заменяется на многочлен x^2+x+1 и т. п.

Это преобразование можно описать в терминах умножения байта $a_{0,j}a_{1,j}$ на двоичную матрицу M, а именно

$$\begin{pmatrix} b_{0,j} \\ b_{1,j} \end{pmatrix} = M \cdot \begin{pmatrix} a_{0,j} \\ a_{1,j} \end{pmatrix},$$

где матрица M имеет вид

$$M = \begin{pmatrix} 1100 & 0100 \\ 0110 & 0010 \\ 1011 & 1001 \\ 1001 & 1000 \\ 0100 & 1100 \\ 0010 & 0110 \\ 1001 & 1011 \\ 1000 & 1001 \end{pmatrix}.$$

Заметим, что $M^{-1} = M$.

AddRoundKey — биты состояния складываются по модулю 2 с соответствующими битами раундового ключа.

Второй раунд отличается от первого тем, что на нём не выполняется преобразование MixColumns().

6.9 Шифр SMS4 — стандарт Китайской Республики

В 2006 году был рассекречен блочный шифр SMS4 — стандарт Китайской Народной Республики для защищенных беспроводных сетей. Он был изобретен китайским профессором Ш. В. Лу. Приведём описание этого шифра, следуя работе [106].

Шифр SMS4 является модифицированной сетью Фейстеля. Шифрование в SMS4 осуществляется блоками длины 128 битов, длина ключа также равна 128 битам. Число раундов равно 32.

По ключу шифрования K строится последовательность раундовых подключей $K_0, K_1, \ldots, K_{31}$ каждый длины 32. Способ построения будет приведён ниже. Для удобства каждый двоичный блок длины 32 будем называть *словом*.

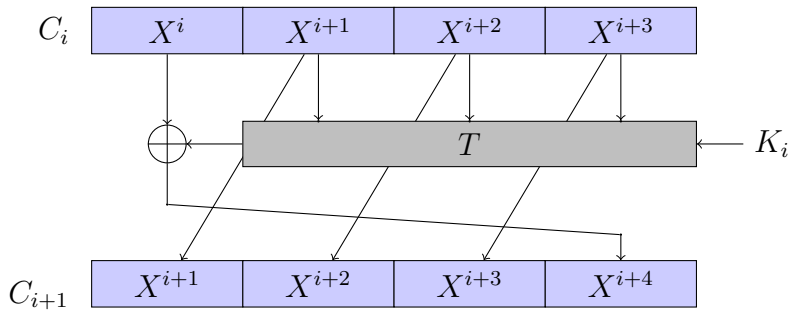
Входной блок P разбивается на четыре слова, а именно,

$$P = (X^0, X^1, X^2, X^3).$$

Шифрование. На каждом i-м раунде, где $i = 0, \ldots, 31$, вычисляется новое слово X^{i+4} по правилу

$$X^{i+4} = X^i \oplus T(X^{i+1} \oplus X^{i+2} \oplus X^{i+3} \oplus K_i),$$

где $T : \mathbb{Z}_2^{32} \to \mathbb{Z}_2^{32}$ — раундовая функция, являющаяся обратимым преобразованием. Приведём схему раунда.



Один раунд шифра SMS4

Шифртекст получается как инвертированный набор последних четырёх слов в этой последовательности. А именно

$$C = (X^{35}, X^{34}, X^{33}, X^{32}).$$

Раундовая функция. Как уже было отмечено, раундовая функция T является обратимым преобразованием, что, вообще говоря, необязательно для сетей Фейстеля (см. раздел 6.3). Преобразование T является комбинацией нелинейной подстановки τ и линейной подстановки L. А именно, $T(\cdot) = L(\tau(\cdot))$.

Нелинейная подстановка определяется так:

$$\tau(X) = \tau(a_0, a_1, a_2, a_3) = (Sbox(a_0), Sbox(a_1), Sbox(a_2), Sbox(a_3)),$$

где каждый a_i обозначает байт слова X, а $Sbox$ обозначает используемый S-блок типа $8 \to 8$. Его таблица приводится ниже.

Линейная подстановка имеет вид

$$L(X) = X \oplus (X <<< 2) \oplus (X <<< 10) \oplus (X <<< 18) \oplus (X <<< 24).$$

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	d6	90	e9	fe	cc	e1	3d	b7	16	b6	14	c2	28	fb	2c	05
10	2b	67	9a	76	2a	be	04	c3	aa	44	13	26	49	86	06	99
20	9c	42	50	f4	91	ef	98	7a	33	54	0b	43	ed	cf	ac	62
30	e4	b3	1c	a9	c9	08	e8	95	80	df	94	fa	75	8f	3f	a6
40	47	07	a7	fc	f3	73	17	ba	83	59	3c	19	e6	85	4f	a8
50	68	6b	81	b2	71	64	da	8b	f8	eb	0f	4b	70	56	9d	35
60	1e	24	0e	5e	63	58	d1	a2	25	22	7c	3b	01	21	78	87
70	d4	00	46	57	9f	d3	27	52	4c	36	02	e7	a0	c4	c8	9e
80	ea	bf	8a	d2	40	c7	38	b5	a3	f7	f2	ce	f9	61	15	a1
90	e0	ae	5d	a4	9b	34	1a	55	ad	93	32	30	f5	8c	b1	e3
a0	1d	f6	e2	2e	82	66	ca	60	c0	29	23	ab	0d	53	4e	6f
b0	d5	db	37	45	de	fd	8e	2f	03	ff	6a	72	6d	6c	5b	51
c0	8d	1b	af	92	bb	dd	bc	7f	11	d9	5c	41	1f	10	5a	d8
d0	0a	c1	31	88	a5	cd	7b	bd	2d	74	d0	12	b8	e5	b4	b0
e0	89	69	97	4a	0c	96	77	7e	65	b9	f1	09	c5	6e	c6	84
f0	18	f0	7d	ec	3a	dc	4d	20	79	ee	5f	3e	d7	cb	39	48

S-блок $8 \to 8$ шифра SMS4

Раундовые ключи. Генерация раундовых ключей в SMS4 является нелинейной процедурой.

Ключ K представляется в виде четырёх слов, а именно $K = (MK_0, MK_1, MK_2, MK_3)$. Далее генерируется последовательность раундовых подключей $K_0, K_1, \ldots, K_{31}$ следу̇ющим образом. Для каждого $i = 0, 1, \ldots, 31$ определяем

$$K_i = HK_{i+4} = HK_i \oplus T'(HK_{i+1} \oplus HK_{i+2} \oplus HK_{i+3} \oplus CK_i),$$

где $HK_i = MK_i \oplus FK_i$ при $i = 0, 1, 2, 3$. Поясним определение.

Вспомогательные блоки FK имеют вид

$$FK_0 = (\texttt{a3b1bac6}), FK_1 = (\texttt{56aa3350}),$$

$$FK_2 = (\texttt{677d9197}), FK_3 = (\texttt{b27022dc}).$$

Функция T' отличается от ранее определённого преобразования T тем, что вместо L в ней используется подстановка L':

$$L'(X) = X \oplus (X <<< 13) \oplus (X <<< 23).$$

Постоянные блоки CK_i, где $i = 0, 1, \ldots, 31$, имеют вид

```
00070e15,  1c232a31,  383f464d,  545b6269,
70777e85,  8c939aa1,  a8afb6bd,  c4cbd2d9,
e0e7eef5,  fc030a11,  181f262d,  343b4249,
50575e65,  6c737a81,  888f969d,  a4abb2b9,
c0c7ced5,  dce3eaf1,  f8ff060d,  141b2229,
30373e45,  4c535a61,  686f767d,  848b9299,
a0a7aeb5,  bcc3cad1,  d8dfe6ed,  f4fb0209,
10171e25,  2c333a41,  484f565d,  646b7279.
```

Для расшифрования требуется использовать те же раундовые ключи, но в обратном порядке.

7. Поточные шифры

7.1 Принципы построения

Использование поточных систем шифрования позволяет избежать таких недостатков блочного шифрования, как *эффект размножения ошибок*, к которому приводит потеря или искажение нескольких битов блока, а также *возможность криптоанализа «со словарем»* в режиме простой замены. Кроме того, именно поточные шифры обеспечивают максимальные скорости шифрования (см. подробнее [7]).

Очень часто поточный шифр состоит из *генератора псевдослучайной последовательности* битов и простого *алгоритма* «наложения» этой последовательности на биты открытого текста. Как правило, шифрование заключается в том, что псевдослучайная последовательность (или *гамма*) складывается по модулю 2 с битами открытого текста, т. е.

$$C = E(P, K) = P \oplus \gamma,$$

где $\gamma = (\gamma_1, \gamma_2, \ldots)$ — выход некоторого псевдослучайного генератора G, работающего на ключе K. Таким образом, свойства поточного шифра полностью определяются свойствами задействованного в нем генератора псевдослучайной последовательности.

Как правило, в поточном шифре гамма является рекуррентной последовательностью и для её порождения используются *регистры сдвига с обратной связью* (feedback shift registers, FSR), речь о которых пойдет в следующем разделе.

7.2 Регистры сдвига с обратной связью

Как пишет в своей книге [80] Брюс Шнайер, «поточные шифры на базе регистров сдвига служили рабочим инструментом военной криптографии задолго до появления электроники».

Введем необходимые определения.

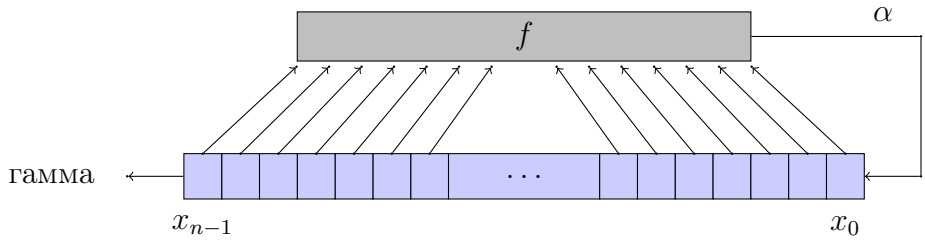
Регистр сдвига с обратной связью состоит из двух частей: двоичного вектора

$$x = (x_{n-1}, \ldots, x_0)$$

длины n и определённой на нём *функции обратной связи*

$$f : (x_{n-1}, \ldots, x_0) \to \{0, 1\}.$$

Напомним, что f является булевой функцией от n переменных (см. раздел 3.1). Сначала биты вектора x заполняются конкретными значениями. Каждое заполнение вектора называется *состоянием* регистра. Затем регистр начинает изменять своё состояние во времени с помощью функции обратной связи.



Регистр сдвига с обратной связью

А именно будем считать время дискретной величиной, измеряемой в тактах. На каждом такте вычисляется значение функции обратной связи, скажем $\alpha = f(x_{n-1}, \ldots, x_0)$, затем все биты регистра сдвигаются влево на одну позицию. При этом крайний левый бит помещается в выходную последовательность регистра (или *гамму*). Новым значением крайнего правого бита становится α.

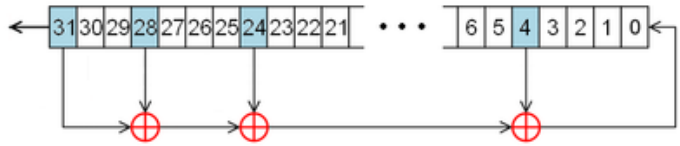
Например, пусть $n = 4$ и начальное состояние регистра — это вектор (0110). Пусть $f(x_3, x_2, x_1, x_0) = x_3x_2 \oplus x_0$. Тогда регистр функционирует во времени следующим образом:

Такт	0	1	2	3	4	5	6
Состояние	(0110)	(1100)	(1001)	(0011)	(0111)	(1111)	(1110)
Гамма	-	0	1	1	0	0	1
Такт	7	8	9	10	11	12	...
Состояние	(1101)	(1010)	(0100)	(1000)	(0000)	(0000)	(0000)
Гамма	1	1	1	0	1	0	...

Здесь приводятся состояния регистра после соответствующего числа тактов. Например, после восьми тактов состоянием регистра будет вектор (1010), а выходная гамма будет содержать биты 01100111. Несложно заметить, что после одиннадцати тактов регистр зацикливается в состоянии (0000) и в выходную гамму выводятся одни нули.

Зацикливание гаммы происходит всегда в силу конечности множества возможных состояний регистра: рано или поздно какое-то состояние регистра должно встретиться дважды. С этого момента гамма начинает повторяться. Длину повторяющегося фрагмента гаммы называют *периодом* регистра. Число битов гаммы до начала её повторения называют *предпериодом* регистра. Отметим, что период и предпериод существенно зависят от начального состояния регистра. В приведённом примере предпериод равен 11, а период равен 1.

Наибольшее распространение получили *регистры сдвига с линейными обратными связями* (кратко — РСЛОС, или LFSR — Linear Feedback Shift Register). Это регистры, в которых функция обратной связи линейная, т. е. представляет собой сумму по модулю 2 отдельных битов регистра. Например, ниже изображен регистр сдвига длины 32 с функцией обратной связи $f(x_{31}, \ldots, x_0) = x_{31} \oplus x_{28} \oplus x_{24} \oplus x_4$.



Часто РСЛОС длины n задают с помощью *многочлена обратной связи* (feedback polynomial). Это многочлен степени n, определяющий биты, которые нужно суммировать. Если

$$f(x_{n-1}, \ldots, x_0) = a_0 x_{n-1} \oplus a_1 x_{n-2} \oplus \ldots \oplus a_{n-1} x_0,$$

то многочлен обратной связи определяется так:

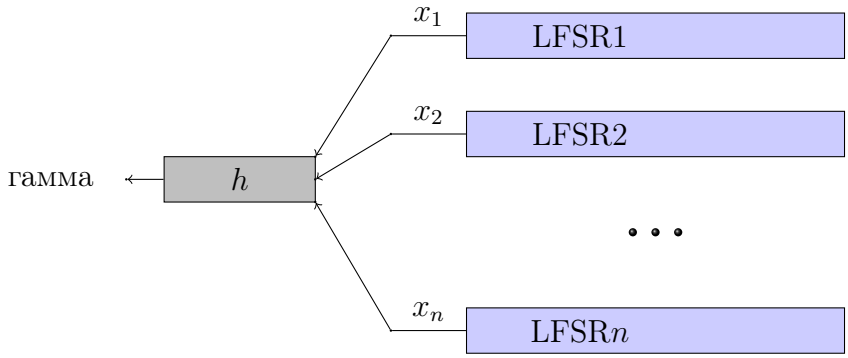
$$p(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1} x + 1.$$

Например, многочлен обратной связи приведённого выше регистра имеет вид $x^{32} + x^{29} + x^{25} + x^5 + 1$. Наличие $+1$ в этом многочлене означает, что результат суммирования направляется в нулевую ячейку.

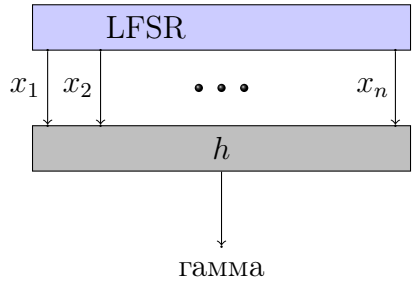
7.3 Комбинирующая и фильтрующая модели поточных генераторов

Генераторы псевдослучайных последовательностей в поточных шифрах часто строятся с использованием нескольких регистров сдвига и нелинейного усложнения.

Выделяют две основные модели псевдослучайных генераторов на основе РСЛОС: *комбинирующую* и *фильтрующую* модели.



Комбинирующая модель



Фильтрующая модель

Свойства булевой функции h от n переменных во многом определяют свойства гаммы. Усложнéние, которое привносит функция h, отражается в возможности получения «хорошей» гаммы при *различных* состояниях регистров. Это важно. Ведь обеспечить определённые хорошие свойства *одной* последовательности (такие как высокая линейная сложность, большая длина периода) — лишь часть дела! Нужно, чтобы *почти все* псевдослучайные последовательности, порождаемые генератором при различных значениях ключа, обладали такими свойствами. Для этого и применяется усложнение. Например, высокая алгебраическая степень $deg(h)$ функции h в фильтрующей модели обеспечивает высокую линейную сложность порождаемых последовательностей и т. п. Подробнее о ċвойствах комбинирующей и фильṫрующей моделей можно прочитать ȧ [7].

7.4 Линейные рекуррентные последовательности

Регистры сдвига с линейными обратными связями представляют собой генераторы так называемых *линейных рекуррентных последовательностей* (кратко — ЛРП), широко применяемых в криптографии.

Двоичная последовательность $\{u_k\}$, где $k = 0, 1, 2, \ldots$, называется *линейной рекуррентной последовательностью*, если существует натуральное число n и константы $a_0, \ldots, a_{n-1} \in \mathbb{Z}_2$ такие, что для любого k выполняется

$$u_{k+n} = a_0 u_k \oplus a_1 u_{k+1} \oplus \ldots \oplus a_{n-2} u_{k+n-2} \oplus a_{n-1} u_{k+n-1}.$$

Число n называется *порядком* ЛРП. Для того чтобы полностью определить последовательность, необходимо задать конкретные значения для n первых её членов, т. е. $u_0, \ldots, u_{n-1}$. Очевидно, что линейная рекуррентная последовательность порядка n порождается с помощью РСЛОС длины n с функцией обратной связи

$$f(x_{n-1}, \ldots, x_0) = a_0 x_{n-1} \oplus a_1 x_{n-2} \oplus \ldots \oplus a_{n-1} x_0$$

и начальным заполнением регистра $(u_0, \ldots, u_{n-1})$. Напомним, что соответствующий *многочлен обратной связи* РСЛОС имеет вид

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

Характеристическим многочленом ЛРП (characteristic polynomial) называется многочлен

$$\varphi(x) = x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_1 x + a_0.$$

Минимальным многочленом ЛРП называется её характеристический многочлен наименьшей возможной степени, обозначим его через $M(x)$. Степень многочлена $M(x)$ называется *линейной сложностью* ЛРП. Другими словами, линейная сложность — это длина самого короткого регистра с линейной обратной связью, порождающего данную последовательность.

В общем случае ЛРП рассматривается над произвольным конечным полем $GF(p^m)$. Её характеристический многочлен определяется как

$$\varphi(x) = x^n - a_{n-1} x^{n-1} - a_{n-2} x^{n-2} - \ldots - a_1 x - a_0,$$

где операции $+$ и $-$ проводятся над полем $GF(p^m)$, а константы $a_0, \ldots, a_{n-1}$ являются элементами этого поля. В приведённом выше определении мы используем «плюс» вместо «минуса», так как над полем $GF(2)$ эти операции совпадают.

Периодом последовательности $\{u_k\}$ называется наименьшее натуральное число T такое, что $u_{k+T} = u_k$ для любого k, начиная с некоторого k_0. Число k_0 называется *предпериодом* последовательности. Если T существует (конечно), то последовательность называется *периодической*. Если при этом её предпериод равен нулю, то она *строго периодическая*. Нетрудно понять, что любая ЛРП порядка n периодическая, причём её период не больше $2^n - 1$. Действительно, число различных возможных состояний соответствующего РСЛОС равно 2^n, при этом нулевое состояние порождает лишь нулевую последовательность.

ЛРП порядка n, имеющая период $2^n - 1$, называется *максимальной*. Её линейная сложность также максимальна и равна n. Максимальные ЛРП особенно интересны для криптографических приложений, поскольку они «наиболее сложные» среди других последовательностей того же порядка.

Свойства минимального многочлена во многом определяют свойства соответствующей ему линейной последовательности, например её период.

Теорема 13. *ЛРП имеет максимальный период тогда и только тогда, когда ее минимальный многочлен является примитивным.*

Многочлен $g(x)$ степени n называется *примитивным*, если он, во-первых, неприводим (т. е. не раскладывается на множители), а во-вторых, наименьшее число s такое, что многочлен $x^s - 1$ делится на $g(x)$, равно $2^n - 1$.

Снова обратимся к алгебре. Данное нами определение примитивного многочлена является вполне строгим. Однако необходимы некоторые пояснения. Неразложимость многочлена на множители надо понимать над полем его коэффициентов. А примитивность многочлена часто определяется другим, впрочем, эквивалентным образом. Пусть $g(x)$ — неприводимый многочлен степени n над полем $GF(p^m)$. Неприводимый многочлен $g(x)$ называется *примитивным*, если его корень является примитивным элементом, т. е. порождающим элементом мультипликативной группы поля разложения этого многочлена (см. подробнее главу 4, а также [47], [20]).

Примитивным многочленом является, например, любой неприводимый многочлен степени n, если число $2^n - 1$ простое. Например, при $n = 5$ любой неприводимый многочлен степени 5 примитивен. Их всего шесть таких многочленов: x^5+x^2+1; x^5+x^3+1; $x^5+x^3+x^2+x+1$; $x^5 + x^4 + x^2 + x + 1$; $x^5 + x^4 + x^3 + x + 1$; $x^5 + x^4 + x^3 + x^2 + 1$. Однако уже при $n = 4$ среди трёх существующих неприводимых многочленов степени 4 два являются примитивными (это многочлены $x^4 + x + 1$ и $x^4 + x^3 + 1$), а один — нет. Этот последний многочлен имеет вид $x^4+x^3+x^2+x+1$. Действительно, его непримитивность сразу следует из разложения $x^5 + 1 = (x + 1)(x^4 + x^3 + x^2 + x + 1)$, т. е. число s для него равно лишь пяти.

7.5 Алгоритм Берлекэмпа — Месси

Как приводится в книге [7], линейная сложность псевдослучайной последовательности — это основной параметр, характеризующий сложность её *аналитического строения.*

Можно считать, что все последовательности, возникающие в криптографических приложениях, периодичны. Несложно понять, что любую периодическую последовательность можно рассматривать как ЛРП (хотя и очень большого порядка в общем случае).

Рассмотрим такую задачу. Пусть имеется достаточно большая часть

$$u_0, u_1, \ldots, u_{\ell-1}$$

некоторой псевдослучайной последовательности. Эту часть будем называть *отрезком* длины ℓ. Как по данному отрезку восстановить *закон рекурсии* всей последовательности? А именно: как построить регистр сдвига с линейной обратной связью минимальной длины, порождающий данную последовательность?

Алгоритм Берлекэмпа — Месси — это эффективный способ решения данной задачи. За полиномиальное время по отрезку последовательности длины ℓ алгоритм позволяет найти её минимальный многочлен, а значит, полностью восстановить её.

Приведём описание этого алгоритма, следуя книге [7]. Для простоты будем рассматривать только двоичные последовательности.

Будем говорить, что многочлен $G(x)$ некоторой степени m вида

$$G(x) = x^m + b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + \ldots + b_1x + b_0 \qquad (7.1)$$

вырабатывает отрезок $u_0, u_1, \ldots, u_{\ell-1}$, если

$$u_{k+m} = b_{m-1}u_{k+m-1} + b_{m-2}u_{k+m-2} + \ldots + b_1u_1 + b_0u_0 \qquad (7.2)$$

для каждого $k = 0, \ldots, \ell-m-1$. Другими словами, исходный отрезок является отрезком некоторой ЛРП с характеристическим многочленом $G(x)$.

Операция *умножения последовательности на многочлен* определяется так. Пусть v — некоторая последовательность,

$$v = \{v_k\} = (v_0, v_1, v_2, v_3, \ldots).$$

Тогда определим

$$1 \cdot v = (v_0, v_1, v_2, v_3, \ldots),$$

$$x \cdot v = (v_1, v_2, v_3, v_4, \ldots),$$

$$x^2 \cdot v = (v_2, v_3, v_4, v_5, \ldots),$$

$$\ldots$$

Другими словами, умножение на x^i означает *сдвиг* всей последовательности влево на i позиций. Тогда результатом умножения многочлена $h(x)$ на последовательность v будет последовательность w, полученная покомпонентным сложением последовательностей $x^j \cdot v$, отвечающих тем степеням x^j, которые входят в $h(x)$ с ненулевыми коэффициентами. Например,

$$(x^3 + x + 1) \cdot v = (v_3 \oplus v_1 \oplus v_0, v_4 \oplus v_2 \oplus v_1, \ldots).$$

Заметим, что многочлен $G(x)$ вида (7.1) вырабатывает ℓ знаков последовательности u, если $G(x) \cdot u = (0, \ldots, 0, 1, *, *, \ldots)$, где первые $\ell-m$ элементов равны нулю, элемент с номером $\ell-m$ равен единице, а значения на местах, помеченных $*$, произвольны. Действительно, нетрудно проверить, что для такого многочлена $G(x)$ выполняются соотношения (7.2).

Индуктивно построим последовательность многочленов

$$G_0(x), G_1(x), G_2(x) \ldots$$

неубывающих степеней $0 = m_0 < m_1 \leqslant m_2 \leqslant \ldots$ такую, чтобы для некоторого j многочлен $G_j(x)$ вырабатывал отрезок $u_0, \ldots, u_{\ell-1}$.

Удобно использовать следующие обозначения. Пусть

$$u = (u_0, u_1, \ldots, u_{\ell-1}, 0, 0, \ldots).$$

Последовательность, полученную умножением $G_i(x)$ на u, где $i = 0, 1, 2, \ldots$, обозначим через

$$u^{(i)} = (u_0^{(i)}, u_1^{(i)}, u_2^{(i)}, \ldots).$$

Пусть k_i обозначает максимальное число первых подряд идущих нулей в последовательности $u^{(i)}$. Считаем, что k_i принимает значения в множестве $\mathbb{N} \cup \{0, +\infty\}$.

Итак, начнём.

Алгоритм Берлекэмпа — Месси

Этап 0. Полагаем $G_0(x) = 1$, $m_0 = 0$. Если G_0 вырабатывает отрезок $u_0, u_1, \ldots, u_{\ell-1}$, то искомый многочлен найден, иначе переходим на следующий этап.

Этап 1. Полагаем $G_1(x) = x^{k_0+1} + u_{k_0+1}^{(0)} x^{k_0} G_0(x)$. Тогда $m_1 = k_0+1$. Если G_1 не вырабатывает отрезок $u_0, u_1, \ldots, u_{\ell-1}$, то переходим на следующий этап.

Этап $t+1$. Пусть многочлены $G_0(x), G_1(x), \ldots, G_t(x)$ уже построены. По построению, $0 = m_0 < m_1 \leqslant m_2 \leqslant \ldots \leqslant m_t$. Определим максимальное число s такое, что $m_t = m_{t-1} = \ldots = m_{s+1} > m_s$. Такое число s найдётся, поскольку $m_1 > m_0$.

Если $k_t \leqslant k_s$, то полагаем $G_{t+1} = G_t(x) + x^{k_s - k_t} u_{k_t}^{(t)} G_s(x)$.

Если $k_t > k_s$, то полагаем $G_{t+1} = x^{k_t - k_s} G_t(x) + u_{k_t}^{(t)} G_s(x)$.

Проверяем, если первые ℓ элементов последовательности $u^{(t+1)}$ равны нулю (т. е. G_{t+1} вырабатывает отрезок $u_0, u_1, \ldots, u_{\ell-1}$), то G_{t+1} — искомый многочлен. Иначе переходим на Этап $t+2$.

Сложность алгоритма равна $6m^2(1+o(1))$ операций в поле $GF(2)$.

Можно проверить, что каждый следующий многочлен $G_i(x)$ вырабатывает хотя бы на один элемент последовательности u больше, чем предыдущий. Поэтому число шагов алгоритма ограничено сверху числом ℓ. Кроме того, каждый многочлен, вырабатывающий столько же знаков последовательности u, что и многочлен $G_j(x)$, имеет степень не меньше, чем степень многочлена $G_j(x)$. Из этих свойств вытекает следующая теорема, которая означает, что алгоритм Берлекэмпа — Месси действительно эффективно строит минимальный многочлен ЛРП.

Теорема 14. *Если* $\{u_k\}$ *— ЛРП с минимальным многочленом* $M(x)$ *степени* m*, то для некоторого подходящего значения* $t \leqslant 2m-1-k_0$ *выполняется равенство* $M(x) = G_t(x)$*.*

Алгоритм Берлекэмпа — Месси в общем виде (над конечными кольцами, модулями и бимодулями) подробно описан в статье В. Л. Куракина [44].

Задача 47. Определите минимальные многочлены последовательностей, отрезки которых имеют вид:

а) $00000\ldots01$ (n первых подряд идущих нулей);
б) 1010;
в) 101011110;
г) 1010111100010011010;
д) 01011011101111011101111011101111.

Задача 48. Пусть степень минимального многочлена $M(x)$ некоторой ЛРП равна m. Определите, какой длины отрезок последовательности достаточно рассмотреть, чтобы восстановить $M(x)$ с помощью алгоритма Берлекэмпа — Месси.

7.6 Шифрование в сотовой связи

В качестве практического примера использования поточных шифров рассмотрим шифр A5, входящий в систему безопасности стандарта GSM цифровой сотовой связи.

Кратко рассмотрим основные этапы развития сотовой связи, а потом остановимся на вопросах её безопасности.

Первой практически реализованной системой мобильной связи следует считать, согласно [51], систему диспетчерской связи полиции города Детройта (США), внедренную ещё в 1921 году. Однако в течение последующих пятидесяти лет сотовая связь развивалась очень медленно. Широкое распространение она получила только с конца 70-х годов XX века, что было связано с изобретением метода повторного использования рабочих частот, позволившего существенно увеличить число абонентов системы. Первая автоматическая система сотовой связи появилась в Чикаго в 1979 году. В России сотовая связь стала внедряться лишь с 1990 года (отечественные разработки велись с 1952 года).

Системы сотовой связи принято относить к разным *поколениям*.

Все системы (или *стандарты*) первого поколения были *аналоговыми*, т. е. передаваемые в них сигналы имели непрерывную область определения. К числу таких систем относились: AMPS (использовалась в США, Канаде, Австралии, Южной Америке), TACS (Англия, Италия, Испания, Австрия, Ирландия, Япония), NMT

(Скандинавския страны, Россия и др. страны). Однако частотное разделение каналов (FDMA), применявшееся в аналоговых системах, всё ещё не позволяло им поддерживать большое число абонентов.

Этот недостаток был преодолен в системах второго поколения, которые стали *цифровыми*. Самым распространенным стандартом второго поколения стал стандарт GSM — Global System for Mobile Communications. Он был разработан под эгидой Европейского института стандартизации электросвязи (ETSI) в конце 80-х годов и начал внедряться в 1992 году. В GSM применяется временно́е разделение каналов (TDMA). На июль 2010 года общее число подключений к GSM превысило 5 миллиардов. По данным ассоциации GSM на данный стандарт приходится 82 % мирового рынка мобильной связи, 29 % населения земного шара использует глобальные технологии GSM (кстати, общая численность населения Земли на май 2012 года составляла около $7,01$ миллиардов человек).

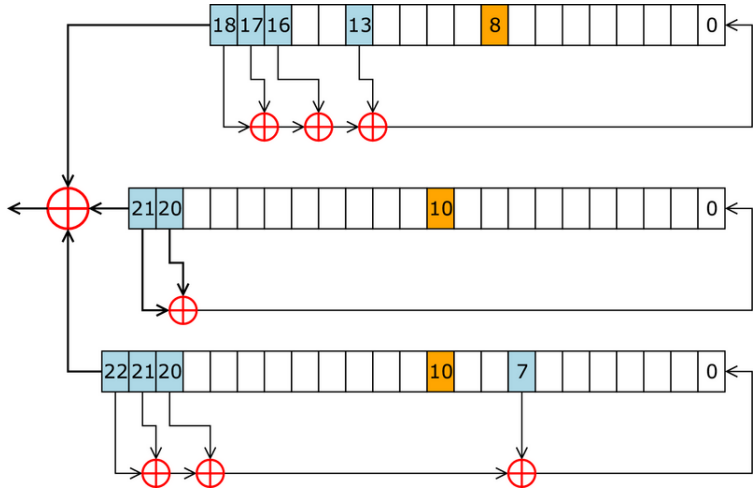
Системы третьего поколения позволят объединять существующие сотовые системы с информационными службами XXI века. «Они будут иметь архитектуру единой сети и предоставлять связь абонентам в различных условиях, включая движущийся транспорт, жилые помещения, офисы и т. д.». Предполагается, что функциональные возможности существующих цифровых систем будут объединены в одну наземную систему мобильной общественной связи будущего (FPLMTS — Future Public Land Mobile Telephone System). Однако широкое распространение систем второго поколения весьма затруднит внедрение систем третьего. По оценкам авторов [51] «в ближайшее десятилетие технологии GSM будут по-прежнему доминировать».

Рассмотрим подробнее систему безопасности GSM. Её основу составляют три алгоритма, официально не раскрытые до сих пор, сообщаемые лишь тем, кому это требуется по необходимости поставщикам оборудования, операторам связи и т. д.: A3 — алгоритм аутентификации; A8 — алгоритм генерации ключа, по сути дела, однонаправленная функция, которая превращает фрагмент выхода от A3 в сеансовый ключ для A5; A5 — собственно алгоритм шифрования оцифрованной речи для обеспечения конфиденциальности переговоров между абонентом и базовой станцией. В GSM используются две основные разновидности алгоритма: A5/1 — сильная версия шифра для избранных стран (в их числе и Россия) и A5/2 — его ослабленная версия для всех остальных. Мобильные станции (телефоны) снабжены смарт-картой, содержащей A3 и A8, а в самом телефоне имеется ASIC-чип с алгоритмом A5. Базовые станции также снабжены ASIC-чипом с A5 и «центром аутентификации», использующим алгоритмы A3, A5 и A8 для идентификации мобильного абонента и генерации сеансового ключа. Более подробно о GSM можно прочитать в [17] и [51].

Первые утечки информации о шифровании в GSM стали появляться в 90-х годах. В 1994 году алгоритм A5 появился в Интернете.

7.7 Алгоритм A5/1 из группы GSM

A5/1 — это поточный алгоритм шифрования, в котором псевдослучайная последовательность порождается на основе трёх регистров сдвига с линейной обратной связью. Регистры имеют длины 19, 22 и 23 бита соответственно. Сдвигами управляет функция большинства (также известная как majority: $m(a_1, a_2, a_3) = a_1a_2 \vee a_2a_3 \vee a_1a_3$) — в каждом регистре есть контрольный бит: восьмой в первом регистре (обозначим a_1), десятый во втором (обозначим a_2) и в третьем (обозначим a_3). Нумерация битов ведется справа налево, начинается с нуля. При очередном такте сдвигаются состояния только тех регистров, у которых значения контрольных бит совпадают со значением функции m. Функция управления сдвигом и $m(a_1, a_2, a_3)$ связаны следующим соотношением: $c(x) = (a_1 \equiv m, a_2 \equiv m, a_3 \equiv m)$. Последние биты регистров суммируются по модулю два. Результат сложения становится новым битом *гаммы*. Гамма накладывается на открытый текст, вследствие чего получается шифртекст. На одном ключе генерируется 114 бит гаммы.



Линейные функции обратной связи удобно представлять с помощью полиномов, сопоставляя каждому биту регистра соответствующую степень переменной x. В шифре A5/1 функции обратной связи задаются следующими полиномами:

$$x^{19} + x^{18} + x^{17} + x^{14} + 1,$$

$$x^{22} + x^{21} + 1,$$

$$x^{23} + x^{22} + x^{21} + x^8 + 1.$$

Например, в первом регистре суммируются биты с номерами 18, 17, 16 и 13 (биты нумеруются справа налево, начиная с нуля). Результат становится новым значением крайнего правого бита (см. рисунок). На нём биты трёх регистров с номерами 8, 10 и 10 (биты a_1, a_2 и a_3 соответственно) являются контрольными.

8. Криптоанализ

Напомним, что под *криптоанализом* понимается научная и практическая деятельность по исследованию криптографических алгоритмов с целью получения обоснованных оценок их криптографической стойкости. Термин «криптоанализ» был введен в 20-х годах XX века известным американским криптографом Уильямом Фридманом.

Элизебет и Уильям Фридман, 1957 год

Путь генетика Уильяма Фридмана в криптографию начался с интереса к задаче о дешифровании рукописей Шекспира и со знакомства со своей будущей женой — криптографом Элизебет Смит. Позднее У. Фридман организовал службу сигнальной разведки США, занимался разработкой статистических методов криптоанализа и дешифрованием шифров различных стран.

Алгоритм считается *криптографически стойким*, если метод криптоанализа, основанный на полном переборе ключа, — так называемый *метод «грубой силы»* (или brute force attack) — для него неэффективен, и не существует других более быстрых методов его криптоанализа. Получение любого метода криптоанализа шифра, более эффективного, чем полный перебор, снижает криптографическую стойкость шифра, даже если предложенный метод является практически нереализуемым.

8.1 Виды криптоанализа

Методы криптоанализа можно разделить на группы различными способами. Например, по *имеющейся у криптоаналитика информации*. При таком подходе выделяют:

- криптоанализ на основе только известного шифртекста;
- криптоанализ на основе известного открытого текста;
- криптоанализ на основе выбранного открытого текста;
- криптоанализ на основе выбранного шифртекста.

По используемым *математическим методам* выделяют статистические методы (такие как линейный, дифференциальный) и аналитические методы криптоанализа (например, алгебраический).

По *характеру применимости* метода различают универсальные и неуниверсальные методы криптоанализа. К числу универсальных, т. е. применимых ко многим шифрам, относятся:

- метод полного перебора;
- анализ на основе словаря;
- использование «парадокса дней рождения»;
- метод «встречи посередине»;
- метод «разделяй и властвуй».

Подробнее об этих и других методах можно прочитать в книге [16]. В этой главе мы остановимся на методах линейного, дифференциального и алгебраического криптоанализа.

8.2 Парадокс дней рождения

Пусть в комнате собрались n случайных людей. Каким наименьшим должно быть число n, чтобы с вероятностью больше $1/2$ среди них нашлись двое, родившиеся в один день?

Для простоты будем считать, что в каждом году $N = 365$ дней. Определим вероятность $p(k)$ того, что k случайных людей родились в разные дни года. Нетрудно заметить, что

$p(2) = 1 - \frac{1}{N}$;
$p(3) = (1 - \frac{1}{N})(1 - \frac{2}{N})$;
$p(4) = (1 - \frac{1}{N})(1 - \frac{2}{N})(1 - \frac{3}{N})$;
...
$p(k) = (1 - \frac{1}{N})(1 - \frac{2}{N})(1 - \frac{3}{N}) \cdot \ldots \cdot (1 - \frac{k-1}{N})$.

Воспользуемся известным неравенством $(1 + s) \leqslant e^s$, выполняющимся при малых значениях s. Здесь e обозначает натуральную экспоненту, $e \approx 2,7$. Тогда

$$p(k) \leqslant e^{-1/N} e^{-2/N} \cdot \ldots \cdot e^{-(k-1)/N} = e^{-\frac{(k-1)k}{2N}}.$$

Определим теперь, при каких значениях k правая часть этого неравенства меньше или равна $1/2$. Решаем неравенство

$$-\frac{(k-1)k}{2N} \leqslant -\ln 2.$$

Имеем $k^2 - k - 2N \ln 2 \geqslant 0$, откуда $k \geqslant \frac{1}{2} + \sqrt{\frac{1}{4} + 2N \ln 2}$. Итак, наименьшее целое значение k, при котором $p(k) \leqslant 1/2$, равно 23. Следовательно, *уже в компании из* 23 *человек с вероятностью больше* $1/2$ *найдётся пара людей, родившихся в один день*. Сам «парадокс дней рождений» заключается в том, что интуитивно число 23 кажется недостаточным; кажется́, что людей в компании должно быть больше!

Применение в криптографии. Приведём несколько примеров.

Пример 1. Пусть $F_K : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ — некоторое взаимно однозначное отображение. Вообще говоря, оно может быть нам неизвестно. Например, это может быть раундовая функция шифра, конкретный вид которой зависит от неизвестного ключа K.

Пусть $A \subset \mathbb{Z}_2^n$ — произвольное подмножество двоичных ве́кторов длины n. Определим, какую минимальную мощность s должно иметь множество A, чтобы с вероятностью больше $1/2$ в нём нашлись элементы x, y такие, что $F_K(x) = y$.

Если A содержит все векторы длины n, то нужная пара, очевидно, всегда будет найдена в нем. Однако в криптографических приложениях множество A часто представляет собой множество набранной

статистики. Поскольку криптоаналитику важно получить результат при самом малом наборе статистики, мощность s множества A желательно минимизировать.

Определим минимально возможный размер s этого множества. Пусть x, y — произвольные двоичные векторы. Так же как в случае с днями рождения, вероятность того, что эти векторы связаны соотношением $F_K(x) = y$, равна $1/2^n$. Действительно, вектор $F_K(x)$ совпадет с конкретным вектором y только в одном из 2^n случаев. Следовательно, вероятность того, что $F_K(x) \neq y$, равна $1 - 1/2^n$. Тогда вероятность того, что в случайном множестве A нет ни одной пары, связанной соотношением $F_K(x) = y$, равна

$$(1 - \frac{1}{2^n})^{s^2},$$

поскольку число различных упорядоченных пар в этом множестве равно s^2. Оценим

$$(1 - \frac{1}{2^n})^{s^2} \leqslant (e^{-1/2^n})^{s^2}$$

и решим неравенство $(e^{-1/2^n})^{s^2} \leqslant 1/2$. Получаем $s^2 \geqslant 2^n \ln 2$. Следовательно, $s \geqslant 2^{n/2}\sqrt{\ln 2} \approx 2^{n/2} \cdot 0,83$. Часто просто полагают $s = 2^{n/2}$.

Таким образом, в произвольном множестве $A \subset \mathbb{Z}_2^n$ мощности $2^{n/2}$ с вероятностью больше $1/2$ найдётся хотя бы одна пара векторов x, y, связанных соотношением $F_K(x) = y$.

Пример 2. Пусть $F_K : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ — некоторая случайная функция, пусть $m < n$. Её конкретный вид можно по-прежнему считать неизвестным. Например, F_K может быть некоторой ключевой хэш-функцией. Пусть $A \subset \mathbb{Z}_2^n$ — произвольное подмножество двоичных векторов длины n. Какую минимальную мощность s должно иметь множество A, чтобы с вероятностью большей $1/2$ в нём нашлись элементы x, x' такие, что $F_K(x) = F_K(x')$? В терминах хэш-функций такая пара векторов x, x' называется *коллизией второго рода*.

Как в предыдущем примере, минимальная мощность множества A равна $2^{m/2}$. *Таким образом, в случайном множестве $A \subset \mathbb{Z}_2^n$ мощности $2^{m/2}$ можно с вероятностью больше $1/2$ найти коллизию второго рода для функции F_K.* Обратим внимание, что минимальная мощность множества A зависит только от мощности множества значений функции. Во многих случаях именно на этом «строит» свой выигрыш криптоаналитик.

8.3 Метод «встречи посередине»

Пусть процедуру зашифрования можно представить в виде последовательного применения двух функций E_1, E_2. А именно пусть

$$C = E(P, K) = E_2(E_1(P, K_1), K_2),$$

где ключ K — объединение независимых подключей K_1 и K_2 длин m_1 и m_2 соответственно.

Пусть криптоаналитику известна пара (P, C) открытого и шифрованного текстов соответственно. Метод «встречи посередине» заключается в следующем.

Шаг 1. Производим частичное зашифрование открытого текста P с помощью функции E_1 при всех возможных значениях подключа K_1. Результаты сохраняем в массиве $U_1, U_2, \ldots, U_{2^{m_1}}$.

Шаг 2. Производим частичное расшифрование шифртекста C, обращая функцию E_2 при всех возможных значениях подключа K_2. Результаты сохраняем в массиве $V_1, V_2, \ldots, V_{2^{m_2}}$.

Шаг 3. Анализируя массивы U, V, формируем возмóжные пары подключей $K = (K_1, K_2)$. Пара является *возможной*, если промежуточный шифртекст U_i, полученный из P с помощью подключа K_1, совпадает с шифртекстом V_j, полученным из C при подключе K_2. Другими словами, при ключе K «встреча посередине» происходит.

Шаг 4. Повторяем Шаги 1, 2, 3 для нескольких (двух, трёх) известных пар (P, C), сокращая при этом множество возможных ключей. Как правило, число повторений очень мало.

Заметим, что сложность метода полного перебора равна $2^{m_1+m_2}$, тогда как сложность метода «встречи посередине» существенно меньше и составляет $a \cdot 2^{m_1} + b \cdot 2^{m_2}$, где a, b — некоторые константы. Метод наиболее эффективен в случае $m_1 = m_2$.

Именно ввиду простоты этого метода в криптографии не используют никакие *двойные* алгоритмы шифрования (т. е. алгоритмы, основанные на двукратном применении к открытому тексту одного и того же алгоритма зашифрования с разными ключами), так как сложность их криптоанализа мало отличается от сложности анализа «одинарного» алгоритма. Например, никогда не используют алгоритм Double DES: применяют либо DES, либо — тройной DES, вспомним задачу 46 и раздел 6.4.

8.4 Линейный криптоанализ. Алгоритмы

Этот статистический метод криптоанализа предложил в 1992–1993 гг. японский криптограф Мицуру Мацуи. Сначала он вместе с А. Ямагиши исследовал этот метод для блочного шифра FEAL (1992), а позднее применил его к шифру DES (1993).

Мицуру Мацуи

Для проведения криптоанализа необходимо знать структуру шифра, а также иметь достаточный объём статистики, состоящей из пар открытого и шифрованного текстов, полученных на одном и том же *неизвестном* криптоаналитику ключе.

Суть метода состоит в *возможности замены* сложной булевой функции, описывающей нелинейное преобразование шифра, простой линейной функцией. При этом, естественно, получается не исходный шифр, а его приближение, но во-первых, оно гораздо проще исходного шифра и легче поддается анализу, а во-вторых, это приближение может оказаться весьма удачным и позволить *свести* проблему криптоанализа исходного шифра к анализу его модификации — с некоторой (допустимой) погрешностью.

Напомним, что $\langle x, y\rangle = x_1y_1 \oplus \ldots \oplus x_ny_n$ обозначает скалярное произведение двоичных векторов по модулю 2.

Основная трудность линейного криптоанализа заключается в поиске удачного линейного приближения. От того, с какой вероятностью найденная линейная функция приближает шифр, зависит необходимый для атаки объём статистики, который желательно сделать минимальным.

Пусть рассматривается некоторый алгоритм блочного шифрования с r раундами. Пусть P, C, K — открытый текст, шифртекст и ключ соответственно. *Линейным приближением шифра* называется соотношение L вида

$$\langle P, \alpha \rangle \oplus \langle C, \beta \rangle = \langle K, \gamma \rangle,$$

которое выполняется с вероятностью $1/2 + \varepsilon$, отличной от $1/2$. Величина ε называется *преобладанием* линейного соотношения. Для успешного криптоанализа модуль преобладания должен быть как можно больше.

Алгоритм 1. Определение одного бита ключа

Шаг 1. Найти *линейное* соотношение L на биты открытого текста, шифртекста и ключа, выполняющееся с вероятностью $p = 1/2 + \varepsilon$, достаточно сильно отличающейся от $1/2$.

Шаг 2. При фиксированном неизвестном ключе K собрать статистику из N пар открытого и соответствующего шифрованного текстов. На её основе с учетом знака ε произвести различение двух простых статистических гипотез: выполняется ли соотношение L для *данного* неизвестного ключа K или нет. А именно:

- для каждой пары (P, C) статистики вычислить значение левой части соотношения L. Пусть N_0 и N_1 обозначают соответственно количества пар статистики, для которых левая часть соотношения равна нулю и единице, $N_0 + N_1 = N$.
- полагаем $\langle K, \gamma \rangle = \begin{cases} 0, & \text{если } (N_0 - N_1) \cdot \varepsilon > 0; \\ 1 & \text{в другом случае;} \end{cases}$
- с учетом полученного соотношения подбираем ключ.

В результате на биты ключа K устанавливается новое вероятностное соотношение, позволяющее определить один бит ключа. Можно установить несколько таких соотношений, используя различные линейные приближения шифра и один и тот же набор статистики.

Однако наибольшее распространение получил улучшенный вариант линейного криптоанализа, позволяющий с помощью одного линейного приближения находить несколько битов̇ ключа.

Пусть C_i обозначает промежуточный шифртекст, где $i = 0, 1, \ldots, r$, и выполняется $C_0 = P$, $C_r = C$.

Алгоритм 2. Определение нескольких битов ключа

Шаг 1. Установить *линейное* соотношение L' на биты промежуточных шифртекстов C_1, C_{r-1} и ключа K, выполняющееся с вероятностью $p = 1/2 + \varepsilon'$, достаточно сильно отличающейся от $1/2$. Число ε' называется *теоретическим преобладанием* соотношения L'. Пусть соотношение L' имеет вид

$$\langle a, C_1\rangle \oplus \langle b, C_{r-1}\rangle = \langle d, K\rangle.$$

Левая часть этого соотношения зависит только от битов шифртекста и не зависит от ключа.

Шаг 2. При фиксированном неизвестном ключе K собрать статистику из N пар открытого и шифрованного текстов. Пусть это пары

$$(P^{(1)}, C^{(1)}), \ldots, (P^{(N)}, C^{(N)}).$$

Шаг 3. Пусть K' обозначает такую минимальную часть ключа K, знание которой позволяет найти значение линейной комбинации $\langle a, C_1\rangle$ по известному открытому тексту P. Для того чтобы определить, какие биты ключа K входят в K', достаточно проанализировать первый раунд шифрования и выявить те биты ключа, от которых существенно зависят биты шифртекста C_1, входящие в комбинацию $\langle a, C_1\rangle$ с ненулевыми коэффициентами.

Аналогично, пусть K'' — такая минимальная часть ключа K, зная которую можно однозначно найти значение линейной комбинации $\langle b, C_{r-1}\rangle$ по известному шифртексту C.

Биты ключа, входящие в $K' \cup K''$, называются *активными* (или иногда — *эффективными*).

Шаг 4. Осуществить перебор по всем возможным вариантам части $K' \cup K''$. Для каждой фиксированной части $K' \cup K''$ провести следующие действия.

• для каждой пары $(P^{(i)}, C^{(i)})$ с использованием частей ключа K', K'' вычислить значение $\langle a, C_1\rangle \oplus \langle b, C_{r-1}\rangle$. Пусть среди N пар ровно для N_0 пар это значение оказалось равным нулю, для N_1 — равным единице, $N = N_0 + N_1$.

• вычислить *экспериментальное преобладание* — это величина $\widetilde{\varepsilon}$, определяемая равенством $1/2 + \widetilde{\varepsilon} = N_0/N$.

Шаг 5. Части ключей $K' \cup K''$, для которых полученные экспериментальные преобладания достаточно сильно отличаются от теоретического преобладания, полагаются неправильными. Оставшиеся

части ключа заносятся в массив. Как правило, для работы алгоритма вводится *допустимая погрешность* δ, определяющая возможное различие между преобладаниями. Если $|\varepsilon' - \widetilde{\varepsilon}| \leqslant \delta$, то различие допустимо, иначе часть ключа $K' \cup K''$ считается неправильной.

Шаг 6. Для каждой допустимой части ключа $K' \cup K''$ перебором восстановить оставшиеся биты ключа K.

Основная идея алгоритма 2, которая позволяет, кажется, «совершить невероятное» — с помощью *одного* соотношения восстановить группу битов ключа, — заключается в том, что установленный «всеобщий закон»[1] на биты C_1, C_{r-1}, K должен статистически подтверждаться, если при его проверке используется правильная группа битов ключа, и должен в основном нарушаться, если группа битов ключа выбрана неверно.

Для оценки качества работы статистического метода криптоанализа (в данном случае — линейного) используется параметр надёжности. Он необходим, поскольку в статистических методах всегда остаётся нежелательная возможность выбрать статистику неудачно: таким образом, что работа алгоритма на ней приведет к неправильному решению.

Надежностью ξ_0 алгоритма, основанного на процедуре статистической классификации, называется математическое ожидание вероятности его корректной работы. В данном случае — математическое ожидание вероятности того, что с помощью алгоритма будет найден правильный ключ. При этом предполагается, что искомый ключ выбран во всем пространстве ключей случайно, равновероятно и независимо от выбора открытых текстов для статистики. А именно

$$\xi_0 = \mathbf{E}\{\xi(K)\} = \frac{1}{2^m} \sum_{K \in \mathbb{Z}_2^m} \xi(K),$$

где $\xi(K)$ — вероятность выбора открытых текстов $P^{(1)}, \ldots, P^{(N)}$ таких, что по ходу работы алгоритма будет установлено верное соотношение на биты ключа K.

Надежность работы ξ_0 метода линейного криптоанализа можно оценить с помощью функции нормального распределения

$$\xi_0 \simeq \Phi_{0,1}(-2|\varepsilon|\sqrt{N}) = \int_{-2|\varepsilon|\sqrt{N}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy.$$

Отсюда заключаем, что *для надёжной работы метода линейного криптоанализа мощность статистики N должна быть пропорциональна величине $|\varepsilon|^{-2}$.*

В заключение приведём численный пример из книги [50] зависимости надёжности ξ_0 от мощности статистики N.

[1] Т. е. линейное соотношение, выполняющееся с фиксированной вероятностью.

N	$\frac{1}{4}\lvert\varepsilon\rvert^{-2}$	$\frac{1}{2}\lvert\varepsilon\rvert^{-2}$	$\lvert\varepsilon\rvert^{-2}$	$2\lvert\varepsilon\rvert^{-2}$
ξ_0	$0,841$	$0,921$	$0,977$	$0,998$

8.5 Линейный криптоанализ: «от простого — к сложному»

Трудность линейного криптоанализа заключается в том, как находить линейные приближения шифра

$$\langle P, \alpha\rangle \oplus \langle C, \beta\rangle = \langle K, \gamma\rangle,$$

выполняющиеся с большим преобладанием. В общем случае необходимо рассматривать всевозможные значения векторов α, β и γ и отдельно находить вероятности выполнения каждого соотношения. Но, как правило, это требует огромных вычислительных затрат. Й, кроме того, неизвестен простой способ вычисления вероятности, с которой выполняется соотношение. Поэтому на практике линейные приближения получают, анализируя отдельные компоненты шифра, начиная с небольших, таких как S-блоки, и поэтапно переходят к приближению всего шифра. Такой метод можно назвать «от простого — к сложному». Опишем его более подробно.

Пусть дан блочный шифр с r раундами, в раундовой функции которого используются S- и P-блоки.

• Строим таблицу линейного преобладания для каждого S-блока. Например, пусть в шифре используется S-блок типа $4 \to 4$. Рассмотрим S-блок из статьи [119]: (`e,4,d,1,2,f,b,8,3,a,6,c,5,9,0,7`).

Приведённая запись означает, что входы `0` $= (0000)$, `1` $= (0001)$, `2` $= (0010)$ под действием S-блока переходят соответственно в векторы `e` $= (1110)$, `4` $= (0100)$, `d` $= (1101)$ и т. д. Пусть вход и выход S-блока обозначаются соответственно через $x = (x_1, x_2, x_3, x_4)$ и $y = (y_1, y_2, y_3, y_4)$. *Таблицей линейного преобладания* S-блока называется следующая таблица, в которой на пересечении строки u и столбца v находится число λ такое, что соотношение $u \cdot x = v \cdot y$ выполняется с вероятностью $(8 + \lambda)/16$.

Например, соотношение $\texttt{0} \cdot x = \texttt{0} \cdot y$ выполняется при каждом значении входа x. Значит его вероятность равна 1. Поэтому элемент

на пересечении нулевых строк в таблице преобладания равен +8. Рассмотрим соотношение $\mathtt{6} \cdot x = \mathtt{3} \cdot y$. После перехода к двоичным представлениям элементов $\mathtt{6} = (0110)$ и $\mathtt{3} = (0011)$ соотношение приобретает вид $x_2 \oplus x_3 = y_3 \oplus y_4$. Проверяя это соотношение на каждой из 16 пар вход–выход S-блока, несложно убедиться в том, что оно выполняется для 12 таких пар. Значит, соответствующее значение в таблице преобладания должно быть равно +4.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	+8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	−2	−2	0	0	−2	+6	+2	+2	0	0	+2	+2	0	0
2	0	0	−2	−2	0	0	−2	−2	0	0	+2	+2	0	0	−6	+2
3	0	0	0	0	0	0	0	0	+2	−6	−2	−2	+2	+2	−2	−2
4	0	+2	0	−2	−2	−4	−2	0	0	−2	0	+2	+2	−4	+2	0
5	0	−2	−2	0	−2	0	+4	+2	−2	0	−4	+2	0	−2	−2	0
6	0	+2	−2	+4	+2	0	0	+2	0	−2	+2	+4	−2	0	0	−2
7	0	−2	0	+2	+2	−4	+2	0	−2	0	+2	0	+4	+2	0	+2
8	0	0	0	0	0	0	0	0	−2	+2	+2	−2	+2	−2	−2	−6
9	0	0	−2	−2	0	0	−2	−2	−4	0	−2	+2	0	+4	+2	−2
a	0	+4	−2	+2	−4	0	+2	−2	+2	+2	0	0	+2	+2	0	0
b	0	+4	0	−4	+4	0	+4	0	0	0	0	0	0	0	0	0
c	0	−2	+4	−2	−2	0	+2	0	+2	0	+2	+4	0	+2	0	−2
d	0	+2	+2	0	−2	+4	0	+2	−4	−2	+2	0	+2	0	0	+2
e	0	+2	+2	0	−2	−4	0	+2	−2	0	0	−2	−4	+2	−2	0
f	0	−2	−4	−2	−2	0	+2	0	0	−2	+4	−2	−2	0	+2	0

Таблица линейного преобладания S-блока.

- Выбирая лучшие линейные приближения S-блоков, строим приближения раундовой функции шифра.

Например, рассмотрим линейное соотношение $x_2 = y_1 \oplus y_2 \oplus y_3$ для пятого S-блока шифра DES (см. раздел 6.4), которое выполняется с вероятностью 42/64. Будем рассматривать раунд с номером i. Следуя описанию раундовой функции DES, несложно установить, что данное соотношение можно представить сначала в виде $w_{26} = z_{17} \oplus z_{18} \oplus z_{19}$, где векторы w, z длины 48 и 32 соответственно представляют общий вход и выход из S-блоков в раундовой функции. Затем, используя то, что вектор w получен из R_{i-1} путём расширения и сложения с раундовым ключом K_i, замечаем, что $w_{26} = R_{i-1}[17] \oplus K_i[26]$, где числа в квадратных скобках обозначают соответствующий бит вектора (или их сумму, если чисел несколько). С другой стороны, биты z_{17}, z_{18}, z_{19} после прохождения P-блока становятся битами вектора $F(R_{i-1}, K_i)$ с номерами 8, 14, 25. Таким образом, получаем линейное приближение

$$R_{i-1}[17] \oplus K_i[26] = F(R_{i-1}, K_i)[8, 14, 25]$$

раундовой функции DES, выполняющееся с вероятностью 42/64.

• Комбинируя различные раундовые приближения, получаем согласованную схему раундовых приближений. Например такую, что комбинация выходных битов, участвующая в приближении i-го раунда, совпадает с комбинацией входных битов в приближении $(i+1)$-го раунда. Иногда согласование проводится «более хитро» (см. далее в разделе 8.7 пример линейного криптоанализа DES).

• Складывая все соотношения из полученной схемы раундовых приближений, получаем линейное соотношение L на биты P, C и K. В это линейное соотношение не должны входить биты промежуточных шифртекстов.

• Вероятность выполнения L определяем с помощью так называемой piling-up леммы, или леммы «О набегании знаков».

Лемма 1. ***(Piling-up lemma)*** *Пусть X_i, где $1 \leqslant i \leqslant n$, — независимые случайные величины, принимающие значения из $\mathbb{Z}_2$. Пусть*

$$\mathbf{P}\{X_i = 0\} = 1/2 + \varepsilon_i, \text{ где } 0 \leqslant |\varepsilon_i| \leqslant 1/2.$$

Тогда случайная величина $X_1 \oplus X_2 \oplus \ldots \oplus X_n$ принимает значение 0 с вероятностью $1/2 + \varepsilon$, где $\varepsilon = 2^{n-1} \prod_{i=1}^{n} \varepsilon_i$.

Задача 49. Докажите piling-up лемму.

Воспользоваться piling-up леммой можно так. Каждое линейное соотношение из схемы раундовых приближений можно представить в виде $X_i = 0$, перенеся в левую часть все случайные величины. Тогда вероятность итогового приближения, полученного суммой соотношений, вычисляется по лемме.

Задача 50. Линейный криптоанализ простого шифра. Проведите линейный криптоанализ трёхраундового блочного шифра:

$P = (p_1, p_2, p_3, p_4, p_5, p_6)$ — открытый текст длины 6,

$K = (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9)$ — ключ шифрования длины 9,

$C = (c_1, c_2, c_3, c_4, c_5, c_6)$ — шифртекст длины 6,

Подключи

$K_1 = (k_1, k_2, k_3, k_4, k_5, k_6)$ — подключ 1-го раунда,

$K_2 = (k_4, k_5, k_6, k_7, k_8, k_9)$ — подключ 2-го раунда,

$K_3 = (k_7, k_8, k_9, k_1, k_2, k_3)$ — подключ 3-го раунда,

S-блоки: вход и выход — векторы длины 3

$S_1 = (0, 5, 7, 2, 3, 4, 1, 6)$,

$S_2 = (6, 1, 2, 0, 5, 4, 3, 7)$.
Пусть $C_0 = P$. Опишем раунд шифрования номер i, где $i = 1, 2, 3$. Полагаем $C_i = (S_1(L_i), S_2(R_i))$, где $(L_i, R_i) = C_{i-1} \oplus K_i$ и векторы L_i, R_i имеют длину 3. Результат зашифрования $C = C_3$.

Определите, какой объём статистики потребуется для работы алгоритма с надёжностью $0,8$; $0,95$; $0,99$. Предложите свои идеи по возможности нелинейного криптоанализа этого шифра.

Задача 51. Линейный криптоанализ S-AES. Проведите линейный криптоанализ шифра S-AES (см. раздел 6.8). Рассмотрите два случая: когда раундовые ключи одинаковые и когда они разные. Оцените необходимый объём статистики для надёжной работы метода.

8.6 Линейный криптоанализ. Предположения

Для работы метода линейного криптоанализа необходимо выполнение некоторых криптографических предположений.

Так, для применения леммы 1 при получении вероятности линейного приближения в методе «от простого — к сложному» требуется

Предположение 1. Считаем, что случайные величины X_i, определяющие соотношения $X_i = 0$ из схемы раундовых приближений, независимы.

Заметим, что на практике это предположение выполняется очень редко. Тем не менее, лемму 1 почти всегда применяют, поскольку в реальных шифрах влияние зависимости случайных величин оказывается очень малым.

Предположение 2. Пусть найдено линейное соотношение L вида $\langle P, \alpha\rangle \oplus \langle C, \beta\rangle = \langle K, \gamma\rangle$, выполняющееся с вероятностью $1/2 + \varepsilon$, достаточно сильно отличающейся от $1/2$ при случайном равновероятном выборе открытого текста P и ключа K. Тогда предполагаем, что для практически всех ключей K вероятность выполнения L также достаточно сильно отличается от $1/2$ при случайном равновероятном выборе открытого текста P.

Пусть при фиксированном ключе K вероятность выполнения линейного соотношения L равна $1/2+\varepsilon_K$. Тогда, предположение 2 озна-

чает, что если величина $|\varepsilon|$ достаточно большая, то для практически всех ключей K величина $|\varepsilon_K|$ также большая.

Заметим, что если линейное приближение L было получено методом «от простого — к сложному», то предположение 2 всегда выполняется, так как при любом ключе K справедливо $\varepsilon_K = \varepsilon$.

Следующее предположение необходимо для работы алгоритма 2 определения нескольких битов ключа.

Предположение 3. Пусть найдено линейное соотношение L' вида $\langle a, C_1\rangle \oplus \langle b, C_{r-1}\rangle = \langle d, K\rangle$, выполняющееся с вероятностью $1/2+\varepsilon'$. Пусть $K' \cup K''$ — минимальная часть ключа, знания которой достаточно для того, чтобы по известной паре (P, C) вычислить значение левой части соотношения L'. Пусть для L' выполняется предположение 2. Тогда предполагаем, что вероятность выполнения соотношения L' в случае, когда значение $\langle a, C_1\rangle \oplus \langle b, C_{r-1}\rangle$ вычислено по паре (P, C) с использованием правильной части ключа $K' \cup K''$, существенно сильнее отличается от $1/2$, чем вероятность выполнения того же соотношения, если его левая часть вычислена по (P, C) с использованием неправильной части ключа.

Обозначим через $P(K' \cup K'')$ вероятность выполнения соотношения L' на набранной статистике при фиксированной части ключа $K' \cup K''$. Пусть K^* обозначает неизвестную криптоаналитику правильную часть ключа. Тогда предположение 3 означает, что

$$\frac{|P(K' \cup K'',\ \text{если}\ K' \cup K'' = K^*) - 1/2|}{|P(K' \cup K'',\ \text{если}\ K' \cup K'' \neq K^*) - 1/2|} >> 1.$$

Другими словами, неправильная часть ключа при работе со статистикой проявляется в «неправильном поведении» вероятностного соотношения: оно ведет себя как случайное, вероятность его выполнения становится близкой к $1/2$. Использование же правильной части ключа приводит к тому, что вероятность выполнения L' на статистике достаточно сильно отличается от $1/2$, что подтверждает отмеченный ранее «общий закон».

В реальности удостовериться в выполнении предположения 3 очень трудно. Существуют примеры шифров, для которых оно неверно.

Задача 52. Контрпример. Постройте пример итерационного отображения $F : \mathbb{Z}_2^n \times \mathbb{Z}_2^m \to \mathbb{Z}_2^n$, где $F(P, K) = C$, для которого предположение 3 не выполнено.

8.7 Линейный криптоанализ DES

При чтении этого раздела будьте готовы к тому, что потребуется активно заглядывать в описание шифра DES (см. раздел 6.4). В частности, нам потребуется детальное описание раундовой функции DES и схемы раундов.

Идея в двух словах. Линейный криптоанализ шифра DES, предложенный Мицуру Мацуи в 1994 году, проводится по второму алгоритму и основан на следующем. Строится линейное приближение L' для 14 раундов шифра DES (без первого и последнего раундов). Определяются 12 активных битов ключа, знания которых достаточно для того, чтобы по известной паре открытого и шифрованного текстов вычислить левую часть соотношения L'. Далее линейный криптоанализ запускается параллельно 2^{12} раз и по алгоритму 2 определяется правильное значение группы активных битов ключа. Кроме того, из самого линейного соотношения определяется ещё один, 13-й, бит ключа. Затем процедура повторяется с аналогичным линейным соотношением, полученным из первого перестановкой местами открытого и шифрованного текстов. Так восстанавливаются ещё 13 битов ключа. Оставшиеся 30 битов ключа находятся перебором.

Линейное приближение для DES. Для нахождения линейного приближения по методу «от простого — к сложному» используются следующие линейные приближения S-блоков:

S-блок	соотношение	вероятность
S_5:	$x_2 = y_1 \oplus y_2 \oplus y_3$	42/64
S_1:	$x_4 = y_2 \oplus 1$	34/64
S_5:	$x_2 = y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus 1$	52/64

Сначала по методу «от простого — к сложному» каждое соотношение используется для построения приближений раундовой функции DES. Внимательно анализируя раундовую функцию, получаем приближения:

соотношение	вероятность
$R_{i-1}[17] \oplus F(R_{i-1}, K_i)[8, 14, 25] = K_i[26]$	42/64
$R_{i-1}[3] \oplus F(R_{i-1}, K_i)[17] \oplus 1 = K_i[4]$	34/64
$R_{i-1}[17] \oplus F(R_{i-1}, K_i)[8, 14, 25, 3] \oplus 1 = K_i[26]$	52/64

С помощью этих соотношений выстраиваем специальную схему приближений для четырнадцати раундов шифра DES (без первого и последнего раундов, как требуется во втором алгоритме линейного криптоанализа), см. таблицу.

После сложения всех линейных соотношений из таблицы получаем линейное приближение для раундов со второго по пятнадцатый включительно:

$$R_1[8,14,25] \oplus L_{15}[17] \oplus R_{15}[8,14,25,3] = K_3[26] \oplus K_4[4] \oplus K_5[26] \oplus$$

$$K_7[26] \oplus K_8[4] \oplus K_9[26] \oplus K_{11}[26] \oplus K_{12}[4] \oplus K_{13}[26] \oplus K_{15}[26]. \quad (8.1)$$

Это соотношение согласно piling-up лемме выполняется с вероятностью

$$\frac{1}{2} + 2^9 \cdot \frac{10}{64} \cdot \frac{2}{64} \cdot \frac{20}{64} \cdot \frac{20}{64} \cdot \frac{2}{64} \cdot \frac{10}{64} \cdot \frac{10}{64} \cdot \frac{2}{64} \cdot \frac{20}{64} \cdot \frac{20}{64} = \frac{1}{2} + \frac{5^7}{2^{37}} \approx 0,50000057.$$

Так, преобладание соотношения равно $\varepsilon' = 0,00000057$.

Напомним, что если на вход алгоритма DES подать шифртекст C, а раундовые ключи использовать в обратном порядке (т. е. сначала K_{16}, затем K_{15} и т. д.), то на выходе алгоритма получим открытый текст P. Благодаря этому свойству шифра DES по (8.1) устанавливаем ещё одно линейное соотношение

$$L_{15}[8,14,25] \oplus R_1[17] \oplus L_1[8,14,25,3] = K_{14}[26] \oplus K_{13}[4] \oplus K_{12}[26] \oplus$$

$$K_{10}[26] \oplus K_9[4] \oplus K_8[26] \oplus K_6[26] \oplus K_5[4] \oplus K_4[26] \oplus K_2[26], \quad (8.2)$$

также выполняющееся с вероятностью $1/2 + \varepsilon'$.

Оба эти соотношения будем использовать для анализа.

соотношение	p
$R_1[8,14,25] = L_2[8,14,25]$	1
$R_2[17] \oplus F(R_2,K_3)[8,14,25] = K_3[26]$	42/64
$L_2[8,14,25] \oplus F(R_2,K_3)[8,14,25] = R_3[8,14,25]$	1
$R_2[17] = L_3[17]$	1
$R_3[3] \oplus F(R_3,K_4)[17] \oplus 1 = K_4[4]$	34/64
$L_3[17] \oplus F(R_3,K_4)[17] = R_4[17]$	1
$R_3[8,14,25,3] = L_4[8,14,25,3]$	1
$R_4[17] \oplus F(R_4,K_5)[8,14,25,3] \oplus 1 = K_5[26]$	52/64
$L_4[8,14,25,3] \oplus F(R_4,K_5)[8,14,25,3] = R_5[8,14,25,3]$	1
$R_5[8,14,25,3] = L_6[8,14,25,3]$	1
$R_6[17] \oplus F(R_6,K_7)[8,14,25,3] \oplus 1 = K_7[26]$	52/64
$L_6[8,14,25,3] \oplus F(R_6,K_7)[8,14,25,3] = R_7[8,14,25,3]$	1
$R_6[17] = L_7[17]$	1
$R_7[3] \oplus F(R_7,K_8)[17] \oplus 1 = K_8[4]$	34/64
$L_7[17] \oplus F(R_7,K_8)[17] = R_8[17]$	1
$R_7[8,14,25] = L_8[8,14,25]$	1
$R_8[17] \oplus F(R_8,K_9)[8,14,25] = K_9[26]$	42/64
$L_8[8,14,25] \oplus F(R_8,K_9)[8,14,25] = R_9[8,14,25]$	1
$R_9[8,14,25] = L_{10}[8,14,25]$	1
$R_{10}[17] \oplus F(R_{10},K_{11})[8,14,25] = K_{11}[26]$	42/64
$L_{10}[8,14,25] \oplus F(R_{10},K_{11})[8,14,25] = R_{11}[8,14,25]$	1
$R_{10}[17] = L_{11}[17]$	1
$R_{11}[3] \oplus F(R_{11},K_{12})[17] \oplus 1 = K_{12}[4]$	34/64
$L_{11}[17] \oplus F(R_{11},K_{12})[17] = R_{12}[17]$	1
$R_{11}[8,14,25,3] = L_{12}[8,14,25,3]$	1
$R_{12}[17] \oplus F(R_{12},K_{13})[8,14,25,3] \oplus 1 = K_{13}[26]$	52/64
$L_{12}[8,14,25,3] \oplus F(R_{12},K_{13})[8,14,25,3] = R_{13}[8,14,25,3]$	1
$R_{13}[8,14,25,3] = L_{14}[8,14,25,3]$	1
$R_{14}[17] \oplus F(R_{14},K_{15})[8,14,25,3] \oplus 1 = K_{15}[26]$	52/64
$L_{14}[8,14,25,3] \oplus F(R_{14},K_{15})[8,14,25,3] = R_{15}[8,14,25,3]$	1
$R_{14}[17] = L_{15}[17]$	1

Схема 14-раундового приближения для DES

Активные биты ключа. По второму алгоритму определяем минимальную часть ключа, с помощью которой по известной паре (P, C) можно восстанавливать комбинации битов шифртекста, входящие в соотношение (8.1). Это следующие двенадцать битов первого и последнего подключей:

$$K' = K_1[25], K_1[26], K_1[27], K_1[28], K_1[29], K_1[30],$$

$$K'' = K_{16}[1], K_{16}[2], K_{16}[3], K_{16}[4], K_{16}[5], K_{16}[6],$$

Действительно, рассмотрим этот момент подробнее. В наше приближение для 14 раундов входят неизвестные биты $R_1[8, 14, 25]$ и $L_{15}[17]$ промежуточных шифртекстов. Их нужно определить. Заметим, что биты $R_{15}[8, 14, 25, 3]$, которые также входят в приближение, нам известны, так как $R_{15} = R_{16}$.

Чтобы восстановить биты $R_1[8, 14, 25]$ по известному открытому тексту L_0R_0, необходимо сначала найти значение $F(R_0, K_1)[8, 14, 25]$. Тогда мы получим $R_1[8, 14, 25] = L_0[8, 14, 25] \oplus F(R_0, K_1)[8, 14, 25]$. Итак, чтобы найти значение $F(R_0, K_1)[8, 14, 25]$, анализируем раундовую функцию DES. Нужные нам выходные биты 8, 14, 25 раундовой функции до прохождения P-блока имели номера 17, 18, 19, т. е. были первым, вторым и третьим битами выхода пятого S-блока DES. Значит, для их восстановления необходимо знать биты входа пятого S-блока, обозначим их x_1, x_2, x_3, x_4, x_5, x_6. Несложно заметить, что эти биты получены в результате сложения отдельных битов блока R_0 с некоторыми битами первого подключа, а именно:

$$x_1 = R_0[16] \oplus K_1[25], \quad x_2 = R_0[17] \oplus K_1[26], \quad x_3 = R_0[18] \oplus K_1[27],$$

$$x_4 = R_0[19] \oplus K_1[28], \quad x_5 = R_0[20] \oplus K_1[29], \quad x_6 = R_0[21] \oplus K_1[30].$$

Таким образом, знания отмеченных шести битов первого подключа достаточно, чтобы восстановить $R_1[8, 14, 25]$ по известному открытому тексту.

Для восстановления бита $L_{15}[17]$ по известному шифртексту необходимо вычислить бит $F(R_{15}, K_{16})[17]$. Тогда будем иметь $L_{15}[17] = F(R_{15}, K_{16})[17] \oplus L_{16}[17]$. Поскольку $R_{15} = R_{16}$, ищем 17-й бит блока $F(R_{16}, K_{16})$. Проникая внутрь функции F, замечаем, что до прохождения P-блока это был бит с номером 2, т. е. второй бит выхода первого S-блока. Значит, для его восстановления необходимы те биты

подключа, которые участвовали в первом S-блоке. Так мы определяем, что нужными битами ключа являются $K_{16}[1]$, $K_{16}[2]$, $K_{16}[3]$, $K_{16}[4]$, $K_{16}[5]$, $K_{16}[6]$.

Аналогично, для второго соотношения (8.2) активные биты ключа — это

$$K''' = K_{16}[25], K_{16}[26], K_{16}[27], K_{16}[28], K_{16}[29], K_{16}[30],$$

$$K'''' = K_1[1], K_1[2], K_1[3], K_1[4], K_1[5], K_1[6].$$

Заметим, что это другие биты, отличные от участвующих в K', K''.

Криптоанализ. Выполняется алгоритм 2 линейного криптоанализа. Параллельно 2^{12} раз запускается обработка статистики: при каждом возможном значении группы $K' \cup K''$ активных битов ключа вычисляется экспериментальное преобладание соотношения (8.1). Сравнивая его с ε', определяется правильное значение группы активных битов ключа. Кроме того, после работы алгоритма получаем линейное соотношение на биты ключа, учáствовавшие в (8.1). Так, имеем информацию о 13 битах ключа.

Далее процедура повторяется со вторым линейным соотношением (8.2), полученным из первого перестановкой местами открытого и шифрованного текстов. С помощью него восстанавливаются ещё 12 битов ключа — $K''' \cup K''''$ — и получается второе линейное соотношение на ключ.

Таким образом, получена информация о 26 битах ключа. Оставшиеся 30 битов ключа находятся перебором.

В среднем для нахождения ключа DES с помощью данного метода потребуется приблизительно 2^{43} пар открытого и шифрованного текстов. При этом успех ожидается в 85% случаев, т. е. найденный с помощью метода ключ будет правильным с вероятностью $0,85$. Отметим, что вследствие большого объёма статистики этот метод криптоанализа DES не относится к числу практичных.

Замечания. Линейный криптоанализ шифра DES оказался возможным в связи со слабыми свойствами его S-блоков. Для того чтобы шифр имел высокую стойкость к линейному криптоанализу, его S-блоки должны быть построены с использованием нелинейных булевых функций, таких как бент-функции и их обобщения. Эти функции уже упоминались в разделе 5 и именно им посвящена глава 9.

В описании криптоанализа DES мы следовали в основном статье М. Мацуи [127]. Однако чтение этой статьи затрудняется тем, что биты блоков в ней нумеруются нестандартно (справа налево и начиная с нуля). Можно обратиться также к книге Б. Шнайера [80], в которой приводится краткая идея метода, а также книге Л. К. Бабенко, Е. А. Ищуковой [16], в которой подробнее изложено построение линейных соотношений. О дальнейшем развитии общего метода можно прочитать в статье Г. П. Агибалова и И. А. Панкратовой [6]. В этой работе изучаются методы построения линейных статистических аналогов функций в итеративных блочных шифрах, рассматриваются алгоритмы криптоанализа, основанные на решении таких линейных и нелинейных аналогов. В частности, в [6] показано, как с помощью одного из алгоритмов можно найти 34 бита ключа DES, иєпользуя те же самые соотношения (8.1), (8.2), которые позволили Мацуи найти 26 битов ключа. Для дальнейшего изўчения темы отметим также статью [85]. Общий подход к использованию в криптоанализе нелинейных приближений предложили Л. Кнудсен и М. Робшау [123].

8.8 Дифференциальный криптоанализ

В 1990 году Эли Бихам и Ади Шамир предложили [86] этот метод для шифра DES. В его основе лежит анализ пар открытых текстов (P, P') и соответствующих им пар шифртекстов (C, C'), между которыми существуют определённые *разности* или *дифференциалы*, а именно $\alpha = P \oplus P'$ и $\beta = C \oplus C'$.

Эли Бихам

Ади Шамир

Алгоритм. Основная идея

Шаг 1. Пара векторов $(\alpha, \beta)_i$ называется *i-м дифференциалом шифра*, если пара открытых текстов, отличающихся на вектор α, после i раундов шифрования может перейти в пару шифртекстов, отличающихся на вектор β. Вероятность i-го дифференциала $\mathbf{P}_{(\alpha,\beta)_i} = \mathbf{P}\{\Delta C_i = \beta \mid \Delta P = \alpha\}$.

Шаг 2. Выбираем наиболее вероятный $(r-1)$-й дифференциал (α, β).

Шаг 3. При фиксированном неизвестном ключе K собираем статистику из N четверок $\{P, P', C, C'\}$ таких, что $\Delta P = \alpha$.

Шаг 4. Предполагаем, что $\Delta C_{r-1} = \beta$. Для тройки $\Delta C_{r-1}, C, C'$ находим список возможных значений ключа K.

Шаг 5. Ключ, встречающийся в таких списках наиболее часто, принимаем за основное решение.

Описание метода в общем виде, в том числе с иллюстрациями на шифре DES, можно найти в статье [4].

Задача 53. Дифференциальный криптоанализ простого блочного шифра. Реализуйте дифференциальный криптоанализ следующего четырёхраундового блочного шифра:

$P = (p_1, p_2, p_3, p_4, p_5, p_6)$ — открытый текст длины 6,

$K = (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8)$ — ключ шифрования длины 8,

$C = (c_1, c_2, c_3, c_4, c_5, c_6)$ — шифртекст длины 6.

Подключи:

$K_1 = (k_1, k_2, k_3, k_4, k_5, k_6)$ — подключ 1-го раунда,

$K_2 = (k_3, k_4, k_5, k_6, k_7, k_8)$ — подключ 2-го раунда,

$K_3 = (k_5, k_6, k_7, k_8, k_1, k_2)$ — подключ 3-го раунда,

$K_4 = (k_7, k_8, k_1, k_2, k_3, k_4)$ — подключ 4-го раунда.

S-блоки: вход и выход — векторы длины 3,

$S_1 = (0, 5, 7, 2, 3, 4, 1, 6)$,

$S_2 = (6, 1, 2, 0, 5, 4, 3, 7)$.

Пусть $C_0 = P$. Опишем раунд шифрования номер i, где $i = 1, 2, 3, 4$. Полагаем $C_i = (S_1(L_i), S_2(R_i))$, где $(L_i, R_i) = C_{i-1} \oplus K_i$ и векторы L_i, R_i имеют длину 3. Результат зашифрования $C = C_4$.

Подумайте над обобщением метода дифференциального криптоанализа, сравните его с линейным.

Задача 54. Дифференциальный криптоанализ S-AES. Проведите дифференциальный криптоанализ шифра S-AES. Рассмотрите два случая: когда раундовые ключи одинаковы и когда они разные. Оцените необходимый объём статистики для работы алгоритма с надёжностью $0,8$; $0,95$; $0,99$.

8.9 Алгебраический криптоанализ

Основная идея алгебраического криптоанализа заключается в составлении сложной системы булевых уравнений, описывающих преобразование шифра. А именно системы, которая строится на основе полностью известного алгоритма шифрования и связывает биты открытого текста, ключа и шифртекста. Небольшое число пар открытого и шифрованного текстов, полученных на неизвестном криптоаналитику ключе, позволяет провести *означивание* системы, а именно подстановку в уравнения значений битов из векторов P и C. Полученную таким образом конкретную систему, в которой неизвестными являются биты ключа, криптоаналитик пытается решить различными способами. Как правило, число уравнений в системе значительно превышает число переме́нных, а уравнения стараются строить так, чтобы система после означивания принимала наиболее простой вид.

Особенностью системы булевых уравнений, возникающей при криптоанализе, является её непротиворечивость, т. е. система заведомо имеет хотя бы одно решение. Это так поскольку существование ключа не является секретом: неизвестно лишь его конкретное значение.

Решение систем булевых уравнений — очень трудная задача, которая в общем случае NP-полна.

Существуют различные подходы к решению систем булевых уравнений. Их компактное и чёткое описание предлагается найти в статье Г. П. Агибалова [3]. Перечислим некото́рые из них.

- *Метод Гаусса* для решения систем линейных уравнений. Его основная идея — последовательное исключение переменных из уравнений системы.

- *Поиск решения путём полного или частичного опробования* возможных значений множества переменных. Если при означивании система становится противоречивой, то данный вариант отбрасывается.

• *Методы линеаризации.* Они состоят в том, чтобы путём преобразования системы (домножения уравнений на мономы определённой степени и т. п.) свести её к линейной от большего числа переменных. Как правило, чтобы полученную систему можно было решить, она должна быть переопределённой (т. е. она должна содержать гораздо больше уравнений, чем переменных).

В 2000-х годах британский криптограф Николя Куртуа разработал несколько методов линеаризации криптографических булевых систем. Среди них — XL- и XSL-методы. Н. Куртуа и Й. Пипджик осуществили первую попытку применить XSL-метод к шифру AES, а перед этим установили, что процесс зашифрования AES можно смоделировать с помощью квадратичной системы уравнений. Долгое время до этого алгебраические атаки считались нереализуемыми на практике.

Николя Куртуа

На криптографической конференции в 2004 году Винсент Рэймен (один из создателей AES) и Николя Куртуа обменялись репликами: «The XSL attack is not an attack. It is a dream» — «It will become your nightmare». И этот ответ Н. Куртуа запомнился.

Появление XSL-атаки привело к новым криптографическим условиям на используемые булевы функции. Для обеспечения стойкости функция должна обладать большой алгебраической степенью и высокой алгебраической иммунностью (см. параграф 5). Чем больше значение алгебраической иммунности булевой функции, тем более

стойким к алгебраическому криптоанализу является шифр, в котором она используется (например, если функция применяется в качестве фильтрующей в поточном генераторе и т. п.).

- *Метод линеаризационного множества.* Подмножество переменных системы называется *линеаризационным множеством*, если фиксация любых значений этих переменных превращает систему в линейную. Если полученная линейная система совместна, решаем её. При таком подходе возникает задача поиска *минимального* линеаризационного множества системы, являющаяся весьма трудной.

- *Метод Бухбергера* — обобщение метода Гаусса последовательных исключений переменных. Он связан с поиском для системы так называемого *базиса Грёбнера* — набора многочленов, полученных в результате многошаговой редукции из исходных многочленов, задающих систему. Как правило, базис Грёбнера позволяет решить нелинейную систему, однако на данный момент трудоемкость его нахождения слишком велика.

На практике эффективность алгебраического криптоанализа пока исследована очень мало. Открытыми остаются вопросы о том, в каких случаях применима эта атака. Но многие криптоаналитики [134] относят её к числу самых перспективных — потенциально, этот дамоклов меч висит над каждым симметричным шифром.

Замечания. Подробнее о методе алгебраического криптоанализа можно прочитать в статье Н. Куртуа и Й. Пипджика [98], обзоре по алгебраическому криптоанализу AES [131], статье [99], посвященной алгебраическому криптоанализу DES. Для знакомства с методами решения ċистем булевых уравнений отлично подходит уже упоминавшаяся статья Г. П. Агибалова [3].

Задача 55. Алгебраический криптоанализ SP-сети. Рассмотрим следующий блочный шифр. Пусть число раундов равно трем.

$P = (p_1, p_2, p_3, p_4, p_5, p_6)$ — открытый текст длины 6,

$K = (k_1, k_2, k_3, k_4, k_5, k_6)$ — ключ шифрования длины 6,

$C = (c_1, c_2, c_3, c_4, c_5, c_6)$ — шифртекст длины 6.

Подключи

$K_1 = (k_1, k_2, k_3, k_4, k_5, k_6)$ — подключ 1-го раунда,

$K_2 = (k_4, k_5, k_6, k_7, k_1, k_2)$ — подключ 2-го раунда,

$K_3 = (k_2, k_6, k_1, k_5, k_3, k_4)$ — подключ 3-го раунда.

S-блок: вход и выход — векторы длины 3,

$S = (1, 0, 3, 6, 7, 4, 5, 2)$.

Пусть $C_0 = P$. Опишем раунд шифрования номер $i = 1, 2, 3$.

$(L_i, R_i) = C_{i-1} \oplus K_i$, где L_i, R_i — векторы длины 3,

$V_i = (S(L_i), S(R_i))$,

$C_i = V_i <<< 2$.

Шифртекст $C = C_3$.

Методом алгебраического криптоанализа найдите ключ K, если:

а) был выполнен один раунд шифрования, в результате чего вектор $P = (101001)$ перешел в $C_1 = (110111)$;

б) были проведены два раунда шифрования, при этом вектор $P = (100001)$ перешел в $C_2 = (100111)$;

в) в результате зашифрования (все три раунда) открытый текст $P = (001011)$ перешел в шифртекст $C = (010010)$.

8.10 Слайдовые атаки

В 1999 году Алекс Бирюков и Дэвид Вагнер предложили следующий метод криптоанализа итерированных шифров, получивший название *слайдовая атака* (slide attack). Основная особенность метода состоит в том, что его успешность *не зависит* от числа раундов в шифре.

Алекс Бирюков

Дэвид Вагнер

Для применимости метода необходимо, чтобы шифр можно было разделить на полностью идентичные раунды (т. е. такие, что на

каждом из них используется один и тот же подключ K). Преобразование на одном таком раунде должно быть относительно несложным. Обозначим его через F.

Пусть шифр разделен на s таких раундов. Зашифрование осуществляется так:

$C_0 = P; \qquad C_i = F(C_{i-1}, K)$ для $i = 1, \ldots, s; \qquad C = C_s$.

Криптоанализ основан на использовании слайдовых пар. Пара открытых текстов (P, P') называется *слайдовой* (slid pair), если P, P' удовлетворяют соотношению $P' = F(P, K)$. Другими словами, один открытый текст получается из другого путём одного раундового преобразования. Заметим, что если C, C' — шифртексты, соответствующие слайдовой паре (P, P'), то они также образуют слайдовую пару, $C' = F(C, K)$.

Многие алгоритмы шифрования построены так, что функция раунда F является относительно слабой, тогда как стойкость шифрования определяется большим числом раундов. В слайдовой атаке криптоаналитик, угадав одну слайдовую пару, может получить значение ключа K как раз в силу несложности F.

Шаг 1. Угадать одну слайдовую пару (P, P') для неизвестного ключа K.

Шаг 2. Имея доступ к шифратору с неизвестным ключом K, получить необходимый объём слайдовых пар:

$I_1 = P$	$O_1 = P'$	— первая слайдовая пара;
$I_2 = E(I_1, K)$	$O_2 = E(O_1, K)$	— вторая слайдовая пара;
$\ldots$	$\ldots$	$\ldots$
$I_N = E(I_{N-1}, K)$	$O_2 = E(O_{N-1}, K)$	— N-я слайдовая пара.

Шаг 3. С помощью набранной статистики (I_j, O_j), $j = 1, \ldots, N$ — пар входа и выхода функции F — провести криптоанализ раундовой функции. Можно применить линейный, дифференциальный, алгебраический и др. методы криптоанализа.

Основная трудность слайдовой атаки состоит в шаге 1. Как угадать первую слайдовую пару, если о преобразовании $F(\cdot, K) : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ неизвестно ничего, кроме того, что оно взаимно однозначно? В этом случае на помощь приходит парадокс дней рождения (см. пример 1 в разделе 8.2). Согласно этому парадоксу в случайном подмножестве $A \subseteq \mathbb{Z}_2^n$ мощности $2^{n/2}$ с вероятностью больше $1/2$ найдутся

элементы P, P' такие, что $F(P, K) = P'$. Поэтому, перебрав все пары элементов из множества A, можно считать, что среди них есть слайдовая пара. Мы не можем явно указать такую пару, но она там есть. Поэтому часто шаги 2 и 3 слайдовой атаки выполняют параллельно для всех пар некоторого подмножества A с расчётом на то, что хотя бы одна ветвь исполнения приведет к правильному ключу шифра.

8.11 Криптоанализ на связанных ключах

По-английски этот метод называют cryptanalysis on related keys или key-schedule cryptanalysis. Он был предложен в 1992 году Эли Бихамом и основан на том, что во многих шифрах проводится расширение ключа. Очень часто ключ шифрования K преобразуется в последовательность раундовых подключей $K_1, \ldots, K_r$ путём модификации и копирования его битов. При этом способ порождения подключей должен быть относительно простым и быстрым.

Пусть $Ext : K \to (K_1, K_2, \ldots, K_{r-1}, K_r)$ — функция расширения ключа, где r — число раундов шифрования. Предположим, что существует ключ K' такой, что $Ext(K') = (K_2, K_3, \ldots, K_r, K_1)$. Тогда ключи K, K' называются *связанными*. Пара K, K' представляет собой аналог слайдовой пары в предыдущем методе криптоанализа.

Приведём действия криптоаналитика, которые во многом напоминают слайдовую атаку.

Шаг 1. На неизвестном ключе K собрать статистику из N пар открытого и шифрованного текстов: $(P_1, C_1), \ldots, (P_N, C_N)$.

Шаг 2. Предположим, что криптоаналитик имеет возможность собрать статистику и для ключа K', связанного с ключом K. Пусть статистика состоит из пар $(P'_1, C'_1), \ldots, (P'_N, C'_N)$.

Шаг 3. Пусть F обозначает раундовую функцию шифра. Для каждых двух пар статистики (P, C) и P', C' пробовать решить систему булевых уравнений

$$\begin{cases} F(P, U) = P' \\ F(C, U) = C' \end{cases}$$

относительно блока переменных U. Если для некоторых пар (P, C) и (P', C') система оказалась совместной и была решена, то с большой вероятностью открытые тексты P, P' образуют слайдовую пару, т. е.

выполняется $F(P, K_1) = P'$. В этом случае найденное решение U совпадает с подключом K_1. Так восстанавливается первый подключ.

Шаг 4. Зная K_1, восстановить весь ключ K (например, перебором оставшихся битов).

Шаг 5. Для того чтобы на шаге 3 слайдовая пара среди всех возможных N^2 пар вида (P, P') была найдена, число N по парадоксу дней рождения должно быть около $2^{n/2}$.

Для успешности метода необходимо, чтобы функция F была достаточно слабой.

Отметим, что атака на связанных ключах не считается практической: криптографические предположения, при которых она работает, слишком сильные. Однако известно, что некоторые реализации криптографических алгоритмов и протоколов могут быть подвержены этой атаке. Например, криптоаналитик может осуществить атаку на протокол обмена ключами и принудить абонента использовать связанные ключи.

Подробнее об этом и других методах можно прочитать, например, в книге С. П. Панасенко [55].

9. Нелинейные булевы функции в криптографии

9.1 Введение

Задача построения булевых функций, обладающих нелинейными свойствами, естественным образом возникает во многих областях дискретной математики. И часто наибольший интерес вызывают функции с экстремальными нелинейными свойствами. Такие булевы функции называются бент-функциями. Впервые они начали исследоваться в 60-х годах XX века в связи с приложениями в криптографии.

Бент-функцию можно определить как функцию, которая крайне плохо аппроксимируется аффинными функциями. В блочных и поточных шифрах бент-функции и их векторные аналоги способствуют предельному повышению стойкости этих шифров к линейному и дифференциальному методам криптоанализа — основным статистическим методам криптоанализа шифров. Например, в 1993 году была обнаружена существенная слабость к линейному криптоанализу шифра DES (см. раздел 8.7). Она заключалась в плохих криптографических свойствах его нелинейных компонент — S-блоков. Напомним, что S-блок — это векторная булева функция, отображающая n входных битов в m выходных битов. Именно эти булевы функции в шифре DES не отвечали необходимым криптографическим требованиям, что и послужило причиной успеха метода линейного криптоанализа. Шифр DES оказался нестойким и к дифференциальному методу криптоанализа. Причина вновь заключалась в слабых S-блоках шифра. Стойкость шифров к упомянутым методам криптоанализа достигается за счёт использования бент-функций, их аналогов и обобщений при построении S-блоков. Это было сделано, например, в канадском шифре CAST и новом американском стандарте AES.

Бент-функциям и их обобщениям посвящена книга [70].

9.2 Бент-функции

Нелинейностью булевой функции f от n переменных называется расстояние Хэмминга N_f от данной функции до множества всех

аффинных функций, а именно

$$N_f = \min_{a\in\mathbb{Z}_2^n, b\in\mathbb{Z}_2} dist(f, \ell_{a,b}),$$

где $\ell_{a,b}(x) = \langle a, x\rangle \oplus b$ — аффинная функция.

Определим нелинейность функции $f(x_1, x_2, x_3) = x_1x_2 \oplus x_1x_3$. Вектор значений этой функции имеет вид $f = (00000110)$. Перечислим векторы значений всех аффинных функций от трёх переменных:

0	00000000	1	11111111
x_1	00001111	$x_1 \oplus 1$	11110000
x_2	00110011	$x_2 \oplus 1$	11001100
x_3	01010101	$x_3 \oplus 1$	10101010
$x_1 \oplus x_2$	00111100	$x_1 \oplus x_2 \oplus 1$	11000011
$x_1 \oplus x_3$	01011010	$x_1 \oplus x_3 \oplus 1$	10100101
$x_2 \oplus x_3$	01100110	$x_2 \oplus x_3 \oplus 1$	10011001
$x_1 \oplus x_2 \oplus x_3$	01101001	$x_1 \oplus x_2 \oplus x_3 \oplus 1$	10010110

Заметим, что $N_f = 2$. Действительно, вес каждой аффинной функции от трёх переменных может быть равен 0, 4 или 8. Вес функции f равен двум. Значит, $N_f \geqslant 2$. При этом расстояние 2 достигается, например, между f и нулевой функцией. Следовательно, $N_f = 2$.

Максимально нелинейной называется булева функция f от n переменных (n любое) такая, что параметр N_f достигает своего максимально возможного значения. В случае чётного n это максимальное значение равно

$$2^{n-1} - 2^{(n/2)-1}.$$

В случае нечётного n точное значение максимального расстояния неизвестно (попробуйте его найти!).

Задача 56. Нерешённая*.** Определите максимально возможное значение нелинейности булевой функции от n переменных в случае, если n нечётно.

При чётном n максимально нелинейные функции называются *бент-функциями* (от англ. bent — изогнутый, наклоненный). Перейдем к эквивалентному определению бент-функций.

Для этого напомним, что согласно задаче 32 нелинейность произвольной булевой функции f можно определить по формуле

$$N_f = 2^{n-1} - \frac{1}{2}\max_{y\in Z_2^n} |W_f(y)|.$$

Другими словами, для нахождения N_f достаточно вычислить все коэффициенты Уолша — Адамара булевой функции f и определить среди них максимальный по модулю. Но согласно утверждению 1 из параграфа 3.9

$$\max_{y\in\mathbb{Z}_2^n} |W_f(y)| \geqslant 2^{n/2}$$

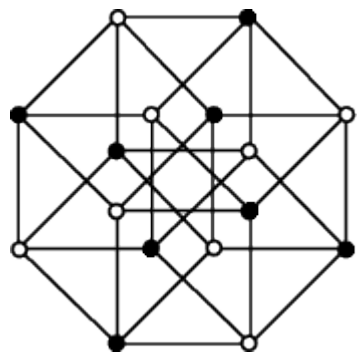
и при этом выполняется равенство Парсеваля:

$$\sum_{y\in\mathbb{Z}_2^n} (W_f(y))^2 = 2^{2n}.$$

Следовательно, максимум нелинейности булевой функции достигается при равных значениях модуля $|W_f(y)|$, для различных y, а именно при значении $|W_f(y)| = 2^{n/2}$.

Таким образом, можно говорить об эквивалентном определении. *Бент-функцией* называется булева функция от n переменных (n чётно) такая, что модуль каждого её коэффициента Уолша — Адамара равен $2^{n/2}$. Множество бент-функций от n переменных обозначим через $\mathcal{B}_n$.

На следующем рисунке изображён булев куб размерности 4, в котором разными цветами помечены вершины, соответствующие векторам значений аффинных функций и бент-функций. Поскольку $n = 2$, каждая бент-функция от двух переменных находится на расстоянии $2^{n-1} - 2^{(n/2)-1} = 1$ от множества аффинных функций. В случае $n = 2$ аффинных функций и бент-функций поровну.



Аффинные функции и бент-функции от двух переменных

9.3 Краткая история

Впервые бент-функции были введены О. Ротхаусом в 60-х годах XX века. Выпускник Принстонского университета Оскар Ротхаус (1927–2003) после службы во время Корейской войны в войсках связи поступил на работу математиком в Агентство национальной безопасности США. С 1960 по 1966 год он работал в Институте оборонного анализа (IDA).

Принстонский университет, США

Криптографические работы О. Ротхауса оценивались руководством IDA достаточно высоко. Как и его преподавательская деятельность: «He was one of the most important teachers of cryptology to mathematicians and mathematics to cryptologists».

В это время была выполнена его первая работа о бент-функциях [135]. В открытой печати она появилась только в 1976 году [136]. В ней были установлены базовые свойства бент-функций, предложены их простейшие конструкции и намечена классификация бент-функций от шести переменных. В дальнейшем Оскар Ротхаус не занимался бент-функциями. С 1966 года он работал в Корнелльском университете (штат Нью-Йорк).

В Советском Союзе тоже изучали бент-функции в 60-х годах, однако имена первых исследователей пока не переданы широкой огласке. Известно, что среди них были В. А. Елисеев и О. П. Степченков.

К числу первых относятся и исследования американских математиков Дж. Диллона [107] и Р. Л. МакФарланда [128], которые в 70-х

годах рассматривали бент-функции в связи с разностными множествами.

С 80-х годов бент-функции начинают интенсивно изучаться во всем мире. В настоящее время известны сотни работ о бент-функциях и близких вопросах. Получены серии конструкций бент-функций, но тем не менее класс всех бент-функций от n переменных до сих пор не описан, для мощности этого класса не найдена асимптотика и не установлено даже приемлемых нижних и верхних оценок. Малый прогресс в этой области связан с тем, что несмотря на простые формулировки задачи здесь очень сложны.

Интересно, что в одном из своих последних годовых отчётов (2001–2002) Оскар Ротхаус писал о том, что он вернулся к исследованиям своей молодости — к кодам и бент-функциям — и получил новые неожиданные результаты. К сожалению, они так и остались неопубликованными.

9.4 Свойства бент-функций

Пусть всюду далее n — чётное число. Перечислим ряд свойств бент-функций, которые вы можете попробовать доказать.

Булева функция $D_y f(x) = f(x) \oplus f(x \oplus y)$ называется *производной* булевой функции f *по направлению* $y \in \mathbb{Z}_2^n$.

Задача 57. Свойство производной*. Докажите, что булева функция f от n переменных является бент-функцией тогда и только тогда, когда каждая её производная $D_y f$ по ненулевому направлению y сбалансирована.

Булева функция f называется *существенно зависимой* от своей переменной x_i, если суiествуют векторы a и b, отличающиеся только значением i-й координаты такие, что $f(a) \neq f(b)$.

Задача 58. Докажите, что бент-функция существенно зависит от каждой своей переменной.

Матрицей Адамара называется квадратная матрица A порядка k с элементами ± 1 такая, что $AA^T = kE$, где E — единичная матрица. Строки и столбцы матрицы размера $2^n \times 2^n$ занумеруем векторами x, y длины n.

Задача 59. (*) Докажите, что булева функция f от n переменных является бент-функцией тогда и только тогда, когда матрица $A = (a_{x,y})$, где $a_{x,y} = \frac{1}{2^{n/2}} W_f(x \oplus y)$, является матрицей Адамара.

Задача 60. Степень бент-функции.** Докажите, что алгебраическая степень $deg(f)$ любой бент-функции f от $n \geqslant 4$ переменных не превосходит $n/2$.

Решение последней задачи можно найти, например, в книге [104].

Аффинная функция, как нетрудно видеть, не может быть бент-функцией (до класса аффинных функций она имеет расстояние 0). Сразу отметим, что бент-функции любой другой возможной степени существуют. Например, квадратичной бент-функцией при любом чётном n является функция

$$f(x_1, \ldots, x_n) = x_1 x_2 \oplus x_3 x_4 \oplus \ldots \oplus x_{n-1} x_n. \tag{9.1}$$

Интересно, что любую другую квадратичную бент-функцию g можно получить из f аффинным преобразованием. Напомним, что булевы функции f и g от n переменных *аффинно эквивалентны*, если существуют невырожденная квадратная матрица A порядка n, векторы b, c длины n и константа $\lambda \in \mathbb{Z}_2$ такие, что $g(x) = f(Ax \oplus b) \oplus \langle c, x \rangle \oplus \lambda$.

Задача 61. (**) Докажите, что любая квадратичная бент-функция от n переменных аффинно эквивалентна функции (9.1).

Задача 62. (*) Докажите, что класс $\mathcal{B}_n$ бент-функций замкнут относительно любого невырожденного аффинного преобразования переменных, а также прибавления любой аффинной функции.

Задача 63. Покажите, что бент-функция не является корреляционно иммунной никакого порядка.

Задача 64. Покажите, что алгебраическая иммунность произвольной бент-функции от n переменных, $n \geqslant 4$, не меньше двух.

9.5 Конструкции бент-функций

Очень сложно не только классифицировать бент-функции, но и предложить отдельные способы их построения. Конструкции принято делить на *первичные* (primary) и *вторичные* (secondary). К первой

группе относят те, с помощью которых бент-функции строятся напрямую, ко второй группе — конструкции, опирающиеся на уже известные бент-функции (например, от меньшего числа переменных).

Теорема 15. (Итеративная конструкция.) *Булева функция вида* $f(x', x'') = g(x') \oplus h(x'')$, *где векторы* x', x'' *имеют чётные длины* r, k *соответственно, является бент-функцией тогда и только тогда, когда функции* g, h — *бент-функции.*

Теорема 16. (Конструкция Мэйорана — МакФарланда, 1973.) *Пусть* h — *любая перестановка на множестве* $\mathbb{Z}_2^{n/2}$, *пусть* g — *произвольная булева функция от* $n/2$ *переменных. Тогда функция* $f(x', x'') = \langle x', h(x'') \rangle \oplus g(x'')$ *является бент-функцией от* n *переменных.*

Основной идеей конструкции МакФарланда служит «соединение аффинных функций». Действительно, при каждом фиксированном значении переменных из второй половины функция f является аффинной от $n/2$ первых переменных. Из теоремы легко следует, что существуют бент-функции с любой степенью d такой, что $2 \leqslant d \leqslant n/2$.

Задача 65. Докажите теоремы 15 и 16.

Следующая первичная конструкция Дж. Диллона 1974 года [108] опирается на специальные семейства подпространств n-мерного пространства и носит название *частичного расщепления* (Partial Spreads). Определение подпространства можно вспомнить, вернувшись к параграфу 3.6. Пусть $Ind_S : \mathbb{Z}_2^n \to \mathbb{Z}_2$ — характеристическая функция подмножества $S \subseteq \mathbb{Z}_2^n$, т. е. такая, что она принимает значение 1 на элементах множества S и значение 0 на остальных элементах.

Теорема 17. (Конструкция Диллона, 1974.) *Пусть число* q *равно* $2^{(n/2)-1}$ *или* $2^{(n/2)-1} + 1$. *Пусть* $L_1, \ldots, L_q$ — *линейные подпространства размерности* $n/2$ *пространства* $\mathbb{Z}_2^n$ *такие, что любые два из них пересекаются лишь по нулевому вектору. Тогда функция* $f(x) = \bigoplus_{i=1}^{q} Ind_{L_i}(x)$ *является бент-функцией.*

Случай $q = 2^{(n/2)-1}$ определяет класс бент-функций $\mathcal{PS}^-$.

Случай $q = 2^{(n/2)-1} + 1$ задает класс бент-функций $\mathcal{PS}^+$.
Вместе $\mathcal{PS}^-$ и $\mathcal{PS}^+$ составляют класс $\mathcal{PS}$.

Приведём несколько алгебраических конструкций.

Первая серия конструкций называется *степенные* или *мономиальные бент-функции* (power/monomial bent functions). Пусть векторное пространство $\mathbb{Z}_2^n$ отождествляется с полем Галуа $GF(2^n)$. Булевы функции от n переменных можно рассматривать как функции из $GF(2^n)$ в $GF(2)$, сопоставляя каждому вектору соответствующий элемент поля $GF(2^n)$, так же как в параграфе 4.2. Пусть $tr : GF(2^n) \to GF(2)$ — функция следа. Бент-функции, имеющие вид

$$f(c) = tr(ac^d),$$

где $a \in GF^*(2^n)$ — некоторый параметр, называются *степенными* или *мономиальными*, а целое число d называется *бент-показателем*. Напомним, что $GF^*(2^n)$ — множество ненулевых элементов поля. Бент-функции такого вида интересны в первую очередь для криптографических приложений в силу своей простой вычислимости. Хотя криптографы до сих пор не определились: считать простоту вычислимости бент-функции её достоинством или, скорее, недостатком [92].

Пусть $gcd(\cdot, \cdot)$ — наибольший общий делитель двух чисел.

Теорема 18. (Мономиальные конструкции.) *Следующие значения d являются бент-показателями:*
$d = 2^{n/2} - 1$;
$d = 2^i + 1$, *где* $\frac{n}{gcd(n,i)}$ *чётно;*
$d = 2^{2k} - 2^k + 1$, *где* $gcd(k, n) = 1$ *и* n *не делится на* 3;
$d = (2^k + 1)^2$, *где* $n = 4k$, k *нечётно;*
$d = 2^{2k} + 2^k + 1$, *где* $n = 6k$.

Известно, что три типа степенных бент-функций (в теореме это второй, четвёртый и пятый) можно описать с помощью конструкции Мэйорана — МакФарланда, а один тип (первый в теореме) содержится в классе $\mathcal{PS}^-$. Существуют ли степенные бент-функции с другими показателями? Можно ли для степенных бент-функций найти простое комбинаторное описание? Ответов на эти вопросы пока нет.

Вторая серия бент-функций состоит из функций вида

$$f(c) = tr(a_1 c^{d_1} + a_2 c^{d_2}) \tag{9.2}$$

для подходящих элементов $a_1, a_2 \in GF(2^n)$ и показателей d_1, d_2. Известны примеры таких функций с так называемыми *показателями Нихо* вида $d \equiv 2^i \ mod\,(2^{n/2}-1)$. Без ограничения общности [111] можно считать, что первый показатель равен $d_1 = (2^{(n/2)}-1)\frac{1}{2}+1$. Справедлива [112]

Теорема 19. *Если* $d_2 = (2^{(n/2)}-1)\lambda + 1$, *где* λ *равно* $1/6$, $1/4$ *или* 3, *то существуют элементы* $a_1, a_2 \in GF(2^n)$ *такие, что (9.2) является бент-функцией.*

Следует отметить, что алгебраические конструкции бент-функций носят весьма случайный характер: каждый раз исследуются функции лишь некоего специального вида. Общий алгебраический подход к описанию бент-функций мог бы основываться на исследовании булевых функций в трейс-форме произвольного вида (см. 4.7):

$$f(c) = tr\left(\sum_{d\in CS} a_d c^d\right) = \sum_{d\in CS} tr(a_d c^d), \tag{9.3}$$

где $a_d \in GF(2^n)$ и через CS обозначено множество представителей циклотомических классов по модулю 2^n-1. Эволюционный алгоритм на основе такого представления был предложен М. Янгом, К. Менгом и Х. Жангом [142]. На основе многочисленных компьютерных исследований авторы делают некоторые предположения относительно общего алгебраического вида бент-функций. В частности, они предполагают, что бент-функцию — представителя класса аффинной эквивалентности — можно представить в виде (9.3) с участием небольшого числа мономов. При этом более вероятными ненулевыми коэффициентами a_d в этом представлении авторы [142] считают те, для которых d является бент-показателем (см. теорему 18). Но общего подхода к алгебраическому описанию бент-функций пока нет.

Задача 66. Нерешённая*.** Предложить новые конструкции бент-функций от n переменных.

Задача 67. Нерешённая*.** Найти новые бент-показатели или доказать, что их не существует.

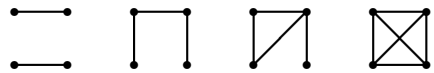
Задача 68. Нерешённая*.)** Исследовать алгебраическую иммунность бент-функций.

9.6 Малое число переменных

Задача описания всех бент-функций от n переменных решена лишь при малых значениях n. Приведём эти результаты.

$n = 2$. Функция x_1x_2 является представителем единственного класса аффинной эквивалентности. Класс $\mathcal{B}_2$ состоит из восьми функций. Это все функции, векторы значений которых содержат нечётное число единиц.

$n = 4$. Множество $\mathcal{B}_4$ состоит из 896 булевых функций, причём каждая функция является квадратичной. Все бент-функции от четырёх переменных аффинно эквивалентны функции $x_1x_2 \oplus x_3x_4$. Множество $\mathcal{B}_4$ можно разделить на 28 классов по 32 функции. Алгебраические нормальные формы функций из каждого класса обладают одинаковой квадратичной частью, произвольной линейной частью и любым свободным членом. Если рассмотреть граф на множестве переменных, а ребрами соединить те вершины, которые образуют слагаемое в квадратичной части АНФ функции, то эти 28 типов можно задать следующим образом:



(3) (12) (12) (1)

Под каждым графом указано число типов, которые он определяет. Например, имеется три типа квадратичной части, состоящей из двух слагаемых: $x_1x_2 \oplus x_3x_4$, $x_1x_3 \oplus x_2x_4$, $x_1x_4 \oplus x_2x_3$, и только один тип из шести слагаемых.

$n = 6$. Аффинная классификация бент-функций от 6 переменных была получена ещё в работе О. Ротхауса [136]: множество $\mathcal{B}_6$ состоит из четырёх классов аффинной эквивалентности, представителями которых являются следующие функции:

$x_1x_2 \oplus x_3x_4 \oplus x_5x_6$,
$x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6$,
$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_2 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_5 \oplus x_4x_5$,
$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_4 \oplus x_3x_5 \oplus x_3x_6 \oplus x_4x_5 \oplus x_4x_6$.

Доказано [107], что любая бент-функция от шести переменных аффинно эквивалентна функции из класса Мэйорана — МакФарланда.

Класс $\mathcal{B}_6$ содержит $5\,425\,430\,528 \simeq 2^{32,3}$ функций.

В работе [142] дана подобная алгебраическая классификация. Пусть $GF(2^6) = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{62}\}$, где α — корень примитивного многочлена $x^6 + x + 1$. Пусть

булева функция отождествляется с функцией $f(c) : GF(2^6) \to GF(2)$, где c рассматривается как элемент поля $GF(2^6)$. Тогда в качестве представителей классов аффинной эквивалентности множества $\mathcal{B}_6$ можно выбрать функции: $tr(c^3 + \alpha^5 c^5)$, $tr(\alpha^3 c^7 + c^9)$, $tr(\alpha c^3 + \alpha^6 c^7 + \alpha^{60} c^{13})$, $tr(c^7 + \alpha c^9 + c^{21})$, где tr — функция следа из $GF(2^6)$ в $GF(2)$, см. параграф 4.3.

$n = 8$. Бент-функции от восьми переменных степени не выше трёх делятся на 10 классов аффинной эквивалентности, представителями которых являются:

$$x_1x_2 \oplus x_3x_4 \oplus x_5x_6 \oplus x_7x_8,$$
$$x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6 \oplus x_7x_8,$$
$$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4 \oplus x_2x_6 \oplus x_1x_7 \oplus x_5x_8,$$
$$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_3 \oplus x_1x_5 \oplus x_2x_6 \oplus x_3x_4 \oplus x_7x_8,$$
$$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_2x_6 \oplus x_2x_5 \oplus x_1x_7 \oplus x_4x_8,$$
$$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_1x_3 \oplus x_1x_4 \oplus x_2x_7 \oplus x_6x_8,$$
$$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_2x_6 \oplus x_2x_5 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1x_4 \oplus x_7x_8,$$
$$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_3x_5 \oplus x_1x_6 \oplus x_2x_7 \oplus x_4x_8,$$
$$x_1x_2x_7 \oplus x_3x_4x_7 \oplus x_5x_6x_7 \oplus x_1x_4 \oplus x_3x_6 \oplus x_2x_5 \oplus x_4x_5 \oplus x_7x_8,$$
$$x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4x_7 \oplus x_3x_5 \oplus x_2x_7 \oplus x_1x_5 \oplus x_1x_6 \oplus x_4x_8.$$

В диссертации [88] также показано, что все эти функции аффинно эквивалентны функциям из класса Мэйорана — МакФарланда.

М. Янг, К. Менг и Х. Жанг [142] показали, что множество $\mathcal{B}_8$ состоит не менее чем из 129 классов аффинной эквивалентности. Представители всех найденных ими классов приводятся в их работе. Это 53 функции вида $tr(\alpha^i c^{d_1} + \alpha^j c^{d_2} + \alpha^k c^{d_3})$ и 76 функций вида $tr(\alpha^i c^{d_1} + \alpha^j c^{d_2} + \alpha^k c^{d_3} + \alpha^\ell c^{d_4})$, где $tr : GF(2^8) \to GF(2)$ — функция следа.

Недавно аффинная классификация бент-функций от восьми переменных четвёртой степени была завершена [124]. Описаны все 536 возможных вариантов для части четвёртой степени[1] АНФ бент-функции от восьми переменных. Установлено точное число всех бент-функций от восьми переменных [124]. Оно равно
$2^9 \times 193\,887\,869\,660\,028\,067\,003\,488\,010\,240 \simeq 2^{106{,}29}$.

При $n \geqslant 10$ класс $\mathcal{B}_n$ не описан, его мощность неизвестна. В статье [140] получена нижняя оценка $2^{262{,}16}$ на мощность $\mathcal{B}_{10}$. В работе [142] построено большое число бент-функций от десяти переменных; установлено, что среди них содержится как минимум несколько сотен попарно аффинно неэквивалентных функций.

[1] Под *частью степени i АНФ функции* понимаем набор всех тех слагаемых её АНФ, степень которых равна i.

9.7 Нижние и верхние оценки

Информации об оценках числа бент-функций от n переменных немного. Приведём нижнюю оценку этого числа, которую дает конструкция Мэйорана — МакФарланда.

Теорема 20. *Справедливо* $|\mathcal{B}_n| \geqslant 2^{2^{n/2}}(2^{n/2})!$.

Асимптотически, эта оценка имеет вид $(\frac{2^{(n/2)+1}}{e})^{2^{n/2}}\sqrt{2^{(n/2)+1}\pi}$, или, если совсем грубо, $2^{2^{n/2}}$. Следует отметить, что в работе [84] приводится улучшение оценки теоремы 20. Однако охарактеризовать асимптотическое поведение оценки [84] достаточно трудно.

Пусть X_n — множество всех булевых функций от n переменных, которые можно представить в виде суммы двух бент-функций, т. е.

$$X_n = \bigcup_{f \in \mathcal{B}_n} (\mathcal{B}_n + f).$$

Кратностью покрытия булевой функции h назовем число бент-функций f от n переменных таких, что h принадлежит множеству $\mathcal{B}_n + f$. Обозначим кратность функции через $m(f)$. Несложно заметить, что $\sum_{f \in X_n} m(f) = |\mathcal{B}_n|^2$. В статье [140] доказана

Теорема 21. *Справедливо* $|\mathcal{B}_n| \geqslant \sum\limits_{f \in X_{n-2}} m(f)^2 \geqslant |\mathcal{B}_{n-2}|^4/|X_{n-2}|$.

Тривиальная верхняя оценка следует из того факта, что согласно задаче 60 степень бент-функции не превышает $n/2$. Имеем

$$|\mathcal{B}_n| \leqslant 2^{1+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{n/2}} = 2^{2^{n-1}+\frac{1}{2}\binom{n}{n/2}}.$$

К. Карле и А. Клаппер в 2002 г. [96] немного улучшили эту оценку:

Теорема 22. *Пусть* $n \geqslant 6$ *и* $\varepsilon = \frac{1}{2^{O(\sqrt{2^n/n})}}$. *Тогда*

$$|\mathcal{B}_n| \leqslant 2^{2^{n-1}+\frac{1}{2}\binom{n}{n/2}-2^{n/2}+(n/2)+1}(1+\varepsilon) + 2^{2^{n-1}-\frac{1}{2}\binom{n}{n/2}}.$$

Хотя по-прежнему верхняя оценка близка к тривиальной 2^{2^n}. Верхняя оценка обсуждается также в работе [141].

Кажется интересным, что аналогичная проблема сильного разрыва между нижней и верхней оценками наблюдается и для числа других комбинаторных объектов.

Задача 69. Нерешённая*.** Улучшить существующие нижние и верхние оценки числа всех бент-функций от n переменных.

Задача 70. Нерешённая*.** Доказать (или опровергнуть) гипотезу о том, что любая булева функция от n переменных (n чётно) степени не выше $n/2$ представима в виде суммы двух бент-функций от n переменных.

9.8 Векторные бент-функции

С 90-х годов XX века стали исследоваться функции $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$, получившие название *векторных булевых функций*, или (n, m)-*функций* (см. ранее 3.3 и 4.5). Интерес к ним вызван тем, что такие нелинейные функции имеют непосредственные криптографические приложения. Например, в шифрах они используются в качестве S-блоков.

Рассмотрим нелинейные свойства векторных функций.

Преобразованием Уолша — Адамара (n, m)-функции F называется отображение $W_F^{vect} : \mathbb{Z}_2^n \times \mathbb{Z}_2^m \to \mathbb{Z}$, заданное равенством

$$W_F^{vect}(a, b) = \sum_{y \in \mathbb{Z}_2^n} (-1)^{\langle a,y\rangle \oplus \langle b,F(y)\rangle} \text{ для любых } a \in \mathbb{Z}_2^n, b \in \mathbb{Z}_2^m.$$

Нелинейностью (n, m)-функции F называется минимальная из нелинейностей булевых функций f_b от n переменных, где $f_b(y) = \langle b, F(y)\rangle$ при различных значениях $b \in \mathbb{Z}_2^m$, $b \neq 0$. Справедливо

$$N_F = \min_{b \in (\mathbb{Z}_2^m)^*} dist(f_b, \mathfrak{A}_n) = 2^{n-1} - \frac{1}{2} \max_{a \in \mathbb{Z}_2^n, b \in (\mathbb{Z}_2^m)^*} |W_F^{vect}(a, b)|.$$

Здесь через $(\mathbb{Z}_2^m)^*$ обозначено множество ненулевых двоичных векторов длины m, а через $\mathfrak{A}_n$ — множество аффинных булевых функций от n переменных. Для нелинейности векторной булевой функции имеется та же верхняя оценка, что и в случае обычной булевой функции:

$$N_F \leqslant 2^{n-1} - 2^{(n/2)-1}. \tag{9.4}$$

Векторная (n, m)-функция называется *бент-функцией*, если параметр N_F достигает своего максимально возможного значения, т. е. если каждая булева функция f_b, где $b \in (\mathbb{Z}_2^m)^*$, является бент-функцией.

Следующий важный факт о существовании векторных бент-функций получила К. Ньюберг [132] в 1991 г.

Теорема 23. *Бент (n, m)-функции существуют тогда и только тогда, когда n чётно и $m \leqslant n/2$.*

Поскольку бент (n, m)-функций не существует при $m > n/2$, то оценка (9.4) в этом случае не точна. В. М. Сидельников [64] и независимо Ф. Шабат, С. Ваденай [97] установили следующий факт.

Теорема 24. *Пусть $m \geqslant n - 1$. Тогда для любой (n, m)-функции F выполняется неравенство*

$$N_F \leqslant 2^{n-1} - \frac{1}{2}\sqrt{3(2^n) - 2 - 2\frac{(2^n - 1)(2^{n-1} - 1)}{2^m - 1}}. \tag{9.5}$$

При $n/2 < m < n - 1$ неизвестна верхняя оценка лучше, чем (9.4). Случай $n = m$ выделяется особо. При нём оценка (9.5) имеет вид

$$N_F \leqslant 2^{n-1} - 2^{(n-1)/2}.$$

Векторная (n, n)-функция F называется *почти бент-функцией* (AB function — almost bent function), если параметр N_F достигает своего максимально возможного значения, $N_F = 2^{n-1} - 2^{(n-1)/2}$. Следует отметить, что по смыслу слово «почти» здесь лишнее, поскольку речь идет о максимальном значении N_F. Но термин в таком виде уже вошел в употребление. AB-функции существуют, только если n нечётно. К. Карле, П. Шарпин и В. Зиновьев [94] доказали, что степень любой такой функции не превышает величины $(n + 1)/2$.

Более широким является класс APN-функций. Эти векторные (n, n)-функции стала рассматривать К. Ньюберг [133] при исследовании устойчивости шифров к дифференциальному криптоанализу.

Стойкость S-блока, заданного векторной функцией F, к дифференциальному криптоанализу тем выше, чем меньше значение

$$\delta_F = \max_{a \in (\mathbb{Z}_2^n)^*, b \in \mathbb{Z}_2^n} \delta_{F,a,b},$$

где через $\delta_{F,a,b}$ обозначено число решений уравнения $F(y) \oplus F(y \oplus a) = b$. Параметр δ_F и его связи с другими нелинейными характеристиками исследовались в работах [49], [82], [90]. Наименьшее возможное

значение параметра δ_F равно двум[2]. Векторная (n, n)-функция, для которой этот минимум достигается, называется *почти совершенно нелинейной* (APN function — almost perfectly nonlinear function). И снова слово «почти» здесь ни при чём. Эквивалентно APN-функция может быть определена как функция, сужение которой на любое двумерное аффинное подпространство пространства $\mathbb{Z}_2^n$ является неаффинной функцией. Подробный обзор результатов о APN-функциях приводится в обзоре М. Э. Тужилина [72] (см. также обзор в [93]).

Кайса Ньюберг

AB- и APN-функции тесно связаны.

Теорема 25. *Каждая AB-функция является APN-функцией.*

Теорема 26. *Квадратичная APN-функция является AB-функцией.*

Приведём одно определение для обычных булевых функций. Булева функция f называется *платовидной* (plateaued function), если существует натуральное число M такое, что любой коэффициент Уолша — Адамара $W_f(y)$ равен нулю или $\pm M$. Из равенства Парсеваля следует, что $M = 2^\beta$ и показатель β может принимать целые значения от $n/2$ до n. Число $2(n - \beta)$ называют *порядком* платовидной функции f. Бент-функции и аффинные функции являются крайними частными случаями платовидных функций (порядков m и 0 соответственно). Справедлива

[2]Интересно, что при рассмотрении q-значных векторных функций, $q \neq 2$, возможно и $\delta_F = 1$.

Теорема 27. *Векторная функция F является AB-функцией тогда и только тогда, когда она APN-функция и все булевы функции f_b при $b \neq 0$ являются платовидными, причём одного порядка.*

Более общим понятием по отношению к понятию APN-функции является следующее. Векторная (n, n)-функция F называется *дифференциально δ-равномерной* (differential δ-uniform), δ — целое число, если уравнение $F(y) \oplus F(y \oplus a) = b$ при любых $a \in (\mathbb{Z}_2^n)^*$, $b \in \mathbb{Z}_2^n$ имеет не более δ решений, т. е., другими словами, $\delta_F = \delta$. Такие функции уже определялись нами в параграфе 5. APN-функции представляют собой частный случай таких функций при $\delta = 2$. Дифференциально 4-равномерные функции (см., например, [87]) используются в S-блоках шифра AES.

AB, APN, δ-равномерные функции и вопросы их эквивалентности широко исследуются. В частности, уже выдвинута гипотеза, что все степенные AB- и APN-функции найдены (X. Доббертин [109]), и обозначена проблема существования новых комбинаторных конструкций таких функций (см. подробнее [89], [93]). При $n \leqslant 25$ для APN-функций и при $n \leqslant 33$ для AB-функций гипотеза Доббертина уже подтвердилась [110], [125].

9.9 Гипербент-функции

В этом разделе, который можно пропустить при первом прочтении, рассмотрим одно криптографическое обобщение бент-функций.

А. М. Йоссеф и Г. Гонг [143] в 2001 году ввели понятие гипербент-функции. Их работе предшествовала статья С. В. Голомба и Г. Гонга [115], в которой алгоритм шифрования DES рассматривался как регистр сдвига с нелинейными обратными связями и проводился анализ его S-блоков. При таком подходе авторы [115] предложили использовать для приближения координатных функций S-блоков вместо линейных булевых функций собственные мономиальные функции (см. раздел 4.8). Эта идея и была развита в [143].

Напомним, что булеву функцию от n переменных можно рассматривать как функцию из $GF(2^n)$ в $GF(2)$, сопоставляя каждому вектору x соответствующий элемент поля $GF(2^n)$. Согласно теореме 8 (см. раздел 4.4) любая линейная функция $\langle x, y \rangle$ может быть представлена как $tr(a_x y)$ для подходящего элемента $a_x \in GF(2^n)$, где $tr : GF(2^n) \to GF(2)$ — функция следа. Тогда преобразование Уол-

ша — Адамара приобретает следующий эквивалентный вид

$$W_f(y) = \sum_{x \in GF(2^n)} (-1)^{tr(yx)+f(x)}.$$

Функция вида $tr(a_x y^s)$, где целое число s такое, что $1 \leqslant s \leqslant 2^n - 1$ и $gcd\,(s, 2^n - 1) = 1$, называется *собственной мономиальной функцией. Расширенное преобразование Уолша — Адамара* булевой функции f имеет вид

$$W_{f,s}(y) = \sum_{x \in GF(2^n)} (-1)^{tr(yx^s)+f(x)}.$$

Булева функция f называется *гипербент-функцией*, если для любого $y \in GF(2^n)$ и любого целого s, $gcd\,(s, 2^n - 1) = 1$, выполняется

$$|W_{f,s}(y)| = 2^{n/2}.$$

Другими словами, гипербент-функция одинаково плохо приближается всеми собственными мономиальными функциями, её обобщенная нелинейность

$$NLG(f) = 2^{n-1} - \frac{1}{2} \max_{\substack{y,\, s \\ gcd\,(s, 2^n - 1) = 1}} |W_{f,s}(y)|$$

максимальна, т. е. равна $2^{n-1} - 2^{(n/2)-1}$. Авторы [143] для каждого четного n доказали существование гипербент-функций, предложили их векторный вариант и для малого числа переменных рассмотрели уравновешенные гипербент-функции. В 2006 году К. Карле и П. Габори [95] и независимо А. С. Кузьмин, В. Т. Марков, А. А. Нечаев и А. Б. Шишков [41] показали, что алгебраическая степень любой гипербент-функции от n переменных равна $n/2$.

А. С. Кузьмин и др. [43], [42] обобщили понятие гипербент-функции: от булевых функций авторы перешли к функциям над произвольным конечным полем характеристики 2.

А именно пусть $q = 2^\ell$. В работе [42] рассматривается задача приближения произвольной функции из $GF(q^n)$ в $GF(q)$ (как и выше, она отождествляется с функцией $f : GF(q^n) \to GF(q)$) функциями из некоторого ограниченного класса $\mathcal{A}$. Для оценки эффективности приближения функции f функцией $g \in \mathcal{A}$, вводится

параметр *согласие* $\nabla(f,g)$. Для согласия выполняется $0 \leqslant \nabla(f,g) \leqslant 1$, и при крайних значениях 0 и 1 функции f и g отличаются, соответственно, на уравновешенную функцию и на константу. При $q=2$ справедливо

$$\left| P(f=g) - \frac{1}{2} \right| = \frac{\nabla(f,g)}{2},$$

т. е. чем меньше согласие между функциями, тем ниже эффективность замены одной на другую. Пусть $\nabla(f,\mathcal{A}) = \max_{g\in\mathcal{A}} \nabla(f,g)$ — *эффективность аппроксимации* функции f функциями из $\mathcal{A}$. Тогда

• если $\mathcal{A} = Hom(A,B)$ — класс всех гомоморфизмов из A в B, то функция $f : GF(q^n) \to GF(q)$ такая, что параметр $\nabla(f, Hom(A,B))$ принимает минимальное возможное значение $q^{-n/2}$, является q-значной бент-функцией в смысле [68].

• пусть $\mathcal{A} = \mathcal{M}$ — класс всех собственных обобщенных мономиальных функций, т. е. функций вида $g(x) = h(x^s)$, где $h \in Hom(A,B)$, s — целое, $gcd\,(s, q^n-1) = 1$.

А.С.Кузьмин и др. предложили следующее определение (2007). Функция $f : GF(q^n) \to GF(q)$ называется *гипербент-функцией*, если параметр $\nabla(f,\mathcal{M})$ принимает минимальное возможное значение $q^{-n/2}$.

При $q=2$ это определение совпадает с определением, приведённым выше.

В [43] проведено детальное исследование таких обобщенных гипербент-функций. Мономиальные приближения булевых функций также изучались А. В. Ивановым [35]. Например, им было показано, что свойство бент-функции быть гипербент-функцией, вообще говоря, зависит от выбора базиса, при котором рассматривается её приведённое представление.

10. Вопросы практики

10.1 Российские вузы

Приведём список некоторых учебных заведений, в которых ведется обучение криптографии и информационной безопасности, а также работа по данной специальности.

ИКСИ — Институт криптографии, связи и информатики Академии ФСБ России (`www.academy.fsb.ru/index_i.html`).

Адрес: 19602, г. Москва, Мичуринский проспект, 70.

История ИКСИ начинается с 1949 года, когда постановлением ЦК ВКП(б) была создана Высшая школа криптографов. Современное название получил в 1992 году. Деятельность выпускников ИКСИ имеет исключительно важное значение для обеспечения безопасности государства. Подготовка специалистов ведется по следующим специальностям: криптография; прикладная математика и информатика; информационная безопасность телекоммуникационных систем; радиоэлектронные системы; компьютерная безопасность; вычислительные машины, комплексы, системы и сети. Ежегодно ИКСИ проводит олимпиаду по математике и криптографии для школьников (`www.cryptolymp.ru`), результаты выступле́ния на которой учитываются при приёме в вузы.

Спецсвязь России — Служба специальной связи и информации Федеральной службы охраны (ФСО) Российской Федерации. Обучение ведется в Академии ФСО России (`www.fso.gov.ru`). Адрес Академии: 302034, г. Орел, ул. Приборостроительная, 35, Академия ФСО России.

Служба спецсвязи была создана в 2003 году на базе Федерального агентства правительственной связи и информации (ФАПСИ). Спецсвязь России является федеральным органом специальной связи и информации, осуществляющим в пределах своих полномочий организацию и обеспечение эксплуатации, безопасности, развития и совершенствования систем правительственной и иных видов специальной связи и информации для федеральных органов государственной власти, органов государственной власти субъектов Российской Федерации и государственных органов. С 2004 года входит в состав ФСО.

МГУ — Московский государственный университет им. М. В. Ломоносова (`www.msu.ru`). А именно: механико-математический факультет (`www.mech.math.msu.su`) и факультет вычислительной математики и кибернетики (`www.cs.msu.su`). Адрес: 119991, Москва, ГСП1, Ленинские горы, Московский государственный университет имени М. В. Ломоносова.

МИФИ — Московский инженерно-физический институт. Адрес: 115409, г. Москва, Каширское шоссе, 31 (www.mephi.ru).

МГТУ — Московский государственный технический университет имени Н. Э. Баумана. Адрес: 105005, г. Москва, 2-я Бауманская ул., 5 (www.bmstu.ru).

РГГУ — Российский государственный гуманитарный университет. В его составе — Институт информационных наук и технологий безопасности (ИИНТБ). Адрес: 125993, г. Москва, ГСП-3, Миусская площадь, 6 (www.rsuh.ru).

МФТИ — Московский физико-технический институт (mipt.ru). Адрес: 141700, Московская область, г. Долгопрудный, Институтский переулок, 9.

СПбГУИТМО — Санкт-Петербургский государственный университет информационных технологий, механики и оптики. Адрес: 197101, г. Санкт-Петербург, Кронверкский просп., 49 (www.ifmo.ru).

СПбГПУ — Санкт-Петербургский государственный политехнический университет. Адрес: 195251, г. Санкт-Петербург, ул. Политехническая, 29 (www.spbstu.ru). Кафедра информационной безопасности компьютерных систем (www.ssl.stu.neva.ru).

ТГУ — Томский государственный университет. Кафедра защиты информации и криптографии. Адрес: 634050, г. Томск, пр. Ленина, 36 (www.tsu.ru).

Кафедры криптографии и информационной безопасности сейчас появляются во многих вузах. В Новосибирске криптографию изучают, например, в НГУ (www.nsu.ru) и СибГУТИ (www.sibsutis.ru).

10.2 Лицензирование криптографической деятельности

Если российское предприятие планирует проводить работы, связанные с использованием / разработкой криптографических средств, то на осуществление своей деятельности оно *обязано* получить лицензию в «Центре по лицензированию, сертификации и защите государственной тайны» ФСБ России.

А именно в соответствии с Законом Российской Федерации от 21 июля 1993 года N 5485-1 «О государственной тайне» и постановлением Правительства Российской Федерации от 15 апреля 1995 г. N 333

«О лицензировании деятельности предприятий, учреждений и организаций по проведению работ, связанных с использованием сведений, составляющих государственную тайну, созданием средств защиты информации, а также осуществлением мероприятий и (или) оказанием услуг по защите государственной тайны» лицензированию, осуществляемому Центром «ЛСЗ» ФСБ России, подлежат:

- деятельность, связанная с использованием сведений, составляющих государственную тайну;
- деятельность по осуществлению мероприятий и (или) оказанию услуг в области защиты государственной тайны;
- деятельность, связанная с созданием средств защиты информации.

В процессе получения лицензии предприятию придётся взаимодействовать с представителями самых разных министерств и служб Российской Федерации. Среди них — Федеральная служба безопасности (ФСБ), Федеральная служба по техническому и экспортному контролю (ФСТЭК), Служба внешней разведки РФ, Министерство обороны РФ и другие. Более подробно о лицензировании можно прочитать на страницах сайта ФСБ России [78].

Для использования / разработки криптографических средств частным лицом лицензии не требуется.

10.3 Криптографические стандарты РФ

Россия имеет три государственных криптографических стандарта (на алгоритм блочного шифрования, хэш-функцию и электронную цифровую подпись). В криптографической деятельности на территории России стандарты играют очень важную роль. Теоретически предприятие может получить лицензию, если разрабатываемая или используемая им криптографическая система содержит методы, отличные от стандартных. Насколько это возможно практически, не вполне ясно.

Кратко приведём действующие стандарты.

ГОСТ 28147-89. Алгоритм блочного шифрования. Полное название — «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». С 1990 года этот шифр является государственным стандартом России и СНГ. В стандарте не накладывается ограничений на степень

секретности защищаемой информации. Предполагается, что она может быть любой.

ГОСТ Р 34.11-94. Алгоритм вычисления хэш-функции. Является стандартом с 23 мая 1994 года. Обязателен для применения в качестве алгоритма хэширования в государственных организациях РФ и ряде коммерческих организаций. ЦБ РФ требует использовать ГОСТ Р 34.11-94 для электронной подписи предоставляемых ему документов.

ГОСТ Р 34.10-2001. Алгоритмы формирования и проверки электронной цифровой подписи. Принят и введен в действие Постановлением Госстандарта России от 12 сентября 2001 года вместо ГОСТ Р 34.10-94. Алгоритм ГОСТ Р 34.10-2001 основан на эллиптических кривых. Его стойкость основывается на сложности вычисления дискретного логарифма в группе точек эллиптической кривой, а также на стойкости хэш-функции по ГОСТ Р 34.11-94.

10.4 Действительность

Как справедливо отмечают в книге «Практическая криптография» [73] известные американские криптографы Нильс Фергюсон и Брюс Шнайер, «система безопасности надёжна настолько, насколько надёжно её самое слабое звено». И часто в информационной безопасности таким звеном оказывается не криптография.

В качестве иллюстрации приведём выдержки из записи на сайте `www.habrahabr.ru`. Это блог, в котором отечественные специалисты в информационных технологиях неформально обсуждают различные проблемы в сфере ИТ, в том числе — в информационной безопасности. Некий пользователь описывает работу информационной системы на одном крупном российском предприятии. Нет возможности убедиться в достоверности приводимой информации. Более того, надеемся, что она неверна. Однако заставляет задуматься один из комментариев: «Насколько можно доверять этим словам, каждый пусть судит сам. По мне, — очень даже вписывается в нашу с вами действительность».

«Давайте знакомиться: инженер группы поддержки программных средств цехов основного производства по производству *** завода ***. Некоторые факты. Все локальные системы управления отличаются как интерфейсно, так и по выбранному «железу». И построены на множестве систем от того же Step 5/7 до C++.

<...> Разрабатывая проект на своей WinXP с правами администратора, задумался ли хоть кто-нибудь о модели безопасности, хотя бы использовать стандартные ограничения прав пользователя? Нет — неудобно же. <...> Все ПК соединены в сеть с доменом, в которой кроме данных технологических ПК находятся и другие, с других участков, управляющие другими контроллерами/процессами. Доменный пользователь, под которым работает проект, обладает правами локального администратора на всех ПК. <...> Пароль к настройкам проекта состоит из русского слова, записанного английскими буквами (недлинного). На каждом ПК лежит бумажка с табельными номерами и паролями работников и контролеров. USB и DVD никак не защищены — подключай, что хочешь. Автозапуск не отключен. Порушить проект может каждый, освоивший удаление файлов в Windows. <...> До недавнего времени (конкретно — до осени прошлого, 2009, года) на ПК и серверах не было антивирусной защиты. Совсем. После занесения и лавинообразного распространения вируса kido примерно через пару недель опомнились. И в течение ещё двух месяцев поставили везде антивирус Касперского KAV 6.0. <...> «Железо» было закуплено сильно заранее и к моменту начала эксплуатации устарело. Так же как и софт. <...> Когда линии были подписаны в промышленную эксплуатацию, они ещё даже не были собраны. Когда закончился период сервисного обслуживания разработчиков, линии ещё не начинали эксплуатироваться. Это не помешало принимать высокопоставленных чиновников из Москвы и показывать им работу «линии» в начале цикла. Кстати, во время визитов «высоких гостей» подменяли и персонал. <...> Для тех, кто уже забыл начало текста, — мы производим ядерное топливо.» — цитата с сайта [77] с купюрами.

Задача 71. (**) Восстановите секретное сообщение, зашифрованное в этом пособии.

Комбинаторные задачи

Биномиальные коэффициенты

Пусть A — произвольное конечное множество из n элементов. Чему равно число его различных k-элементных подмножеств? Это число называется *числом сочетаний из n по k* или просто *биномиальным коэффициентом* и обозначается C_n^k или $\binom{n}{k}$. Например, при $n = 4$, $k = 2$ оно равно 6. Если $A = \{a, b, c, d\}$, то интересующие нас подмножества: $\{a, b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, d\}$, $\{c, d\}$.

Задача 72. Докажите, что $C_n^k = \frac{n!}{k!(n-k)!}$ при $0 \leqslant k \leqslant n$.

Определите, чему равно число *всех* различных подмножеств множества A. Эту задачу умели решать в Индии ещё во II веке до н. э.

Задача 73. Докажите, что $C_n^0 + C_n^1 + C_n^2 + ... + C_n^n = \sum\limits_{k=0}^{n} C_n^k = 2^n$.

Задача 74. Биномиальная формула. Докажите, что для любых чисел a, b и любого натурального n справедлива формула

$$(a + b)^n = \sum_{i=0}^{n} C_n^i \; a^i \; b^{n-i},$$

которая уже была, по-видимому, известна персидскому поэту, математику и философу Омару Хайяму (1048–1131).

Задача 75. Докажите, что $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$.

Задача 76. Найдите, чему равны выражения:

а) $C_n^0 - C_n^1 + C_n^2 - ... + (-1)^n C_n^n$;

б) $C_n^0 + C_n^2 + C_n^4 + ...$;

в) $C_n^1 + C_n^3 + C_n^5 + ...$

Задача 77. Докажите, что если n фиксировано, то C_n^k возрастает по k при $k \leqslant \lfloor n/2 \rfloor$ и убывает при $k > \lceil n/2 \rceil$, где запись $\lfloor x \rfloor$ означает ближайшее целое число к x снизу, а запись $\lceil x \rceil$ — ближайшее целое к x сверху (например, $\lfloor 3,14 \rfloor = 3$, $\lceil 3,14 \rceil = 4$).

Задача 78. О кодовом замке. На входных дверях Алисы и Боба — кодовые замки. Код — это комбинация различных цифр $0, 1, ..., 9$, которая нажимается одновременно. Известно, что код на двери у Алисы состоит из трёх цифр, у Боба — из двух. Злоумышленник Ева на проверку одной комбинации на кодовом замке тратит две секунды. Сколько времени ей понадобится, чтобы подобрать код Алисы и проникнуть в дом? А код Боба? Прежде чем решать, попробуйте предложить свой интуитивный ответ на этот вопрос (хватит ли Еве недели? суток? часа? десяти минут? минуты?), а потом проверьте себя. Сколько цифр должен содержать самый надёжный дверной код? Что изменится, если в кодовой комбинации учитывать порядок цифр (т. е. нажимать их последовательно)?

Задача 79. В классе учатся n девушек и m юношей. Сколькими способами можно выбрать пару из одной девушки и одного юноши, чтобы назначить их дежурными? А выбрать группу из k девушек и ℓ юношей для участия в соревнованиях?

Задача 80. О шахматном городе. Город — это $m \times n$ прямоугольных кварталов, разделённых $(n-1)$ горизонтальными и $(m-1)$ вертикальными улицами. Каждому перекрёстку отвечает пара координат (i, j). Чему равно число различных кратчайших путей из точки $(0, 0)$ в точку (m, n)?

Задача 81. Идея на прогулке. Как, размышляя о шахматном городе, доказать, что $C_{2n}^n = (C_n^0)^2 + (C_n^1)^2 + ... + (C_n^n)^2$?

Задача 82. О прямых. На плоскости проведено n прямых так, что никакие две из них не параллельны и никакие три не пересекаются в одной точке.

а) Чему равно число точек пересечения этих прямых?

б) Сколько различных треугольников образуют прямые?

в) На сколько частей эти прямые делят плоскость?

г) Сколько среди этих частей ограниченных и неограниченных?

Главное — это... порядок?

Множество называется *упорядоченным*, если некоторым образом упорядочены его элементы. И этот порядок важен. Например, упорядоченные множества $A = \{a, b, c\}$ и $B = \{b, c, a\}$ различны. Хотя

без учета порядка это одно и то же множество. Нетрудно понять, что существует ровно $n!$ различных упорядоченных множеств из одних и тех же n элементов.

Задача 83. Сколькими способами можно упорядочить множество $\{1, 2, ..., 2n\}$ так, чтобы каждое чётное число имело чётный номер?

Задача 84. О ладьях. Сколькими способами можно расположить на шахматной доске 8 ладей так, чтобы они не били друг друга?

Задача 85. Сколькими способами можно упорядочить множество $\{1, 2, ..., n\}$ так, чтобы числа $1, 2, 3$ стояли рядом и в порядке возрастания?

Задача 86. Гости за круглым столом. Сколькими способами можно рассадить n гостей за круглым столом? Способы считаются одинаковыми, если один можно получить из другого циклической перестановкой.

Полиномиальные коэффициенты

Эти коэффициенты возникают, когда мы имеем дело с задачей разбиения множества на части. А именно пусть $n = k_1 + k_2 + \ldots + k_m$. Сколькими способами можно разбить n-элементное множество A на m подмножеств $B_1, \ldots, B_m$ так, что $|B_1| = k_1, \ldots, |B_m| = k_m$? Это число называется *полиномиальным коэффициентом* и обозначается через $C_n(k_1, \ldots, k_m)$.

Задача 87. Докажите, что $C_n(k_1, \ldots, k_m) = \frac{n!}{k_1!k_2!\cdot\ldots\cdot k_m!}$.

Задача 88. Сколько различных слов можно составить, переставляя буквы слова «МАМА»?, слова «МАТЕМАТИКА»?

Решив эту задачу, несложно понять, почему полиномиальный коэффициент называется также *числом перестановок из n элементов с повторениями.*

Сочетания с повторениями

Задача 89. Сколькими способами можно выбрать 6 одинаковых или разных пирожных в кондитерской, в которой есть 11 разных сортов пирожных?

Задача 90. Сколько различных костей домино можно сделать, используя числа $0, 1, 2, \ldots, n$?

Конечному множеству из n элементов можно придать такой смысл: пусть каждый его элемент обозначает некоторый *тип* предмета. Считаем, что предметов каждого типа существует бесконечно много и между собой они неразличимы. Сколькими способами можно выбрать k предметов, каждый из которых может быть одного из n типов? Это число называется числом *сочетаний из n элементов по k с повторениями*. Докажите, что оно равно C^k_{n+k-1}.

Разное

Задача 91. Книжная полка. На полке стоят 12 книг. Сколькими способами можно выбрать 5 книг, не стоящих рядом?

Задача 92. Домино. Кости домино сделаны с использованием чисел 0, 1, 2, 3, 4, 5, 6. Сколькими способами можно выбрать пару домино так, чтобы их можно было приложить друг к другу?

Задача 93. Сколько различных матриц $m \times n$, заполненных элементами 0 и 1, можно составить? Сколько среди них матриц с попарно различными строками?

Задача 94. Друзья Боба. У Боба 6 друзей и ежедневно в течение 18 дней он приглашает к себе в гости троих из них так, что компания ни разу не повторяется. Сколькими способами он может это сделать?

Задача 95. Треугольники. Сколько существует треугольников, у которых длина каждой стороны принимает одно из четырёх значений 4, 5, 6, 7?

Задача 96. Векторы. Сколько существует различных векторов длины 20 таких, что каждый вектор содержит

(а) 5 нулей, 4 единицы, 7 двоек, а остальные — тройки?
(б) в сумме 5 нулей и единиц; 4 двойки, 7 троек, а остальные — четверки или пятерки?

Задача 97. Враждующие рыцари*. За круглым столом собрались 12 рыцарей. Каждые два соседних рыцаря враждуют между собой. Сколькими способами можно выбрать 5 попарно невраждующих друг с другом рыцарей?

Метод включения и исключения

Задача 98. Полиглоты. Из 100 студентов английский язык знают 28, немецкий — 30, итальянский — 42, английский и немецкий — 8, английский и итальянский — 10, немецкий и итальянский — 5, все три языка знают 3 студента, а декан факультета свободно говорит по-испански. Сколько студентов не знают ни один язык?

Пусть $A_1, \ldots, A_m$ — некоторые подмножества n-элементного множества A. Определим n_0 — число элементов множества A, *не принадлежащих* ни одному из подмножеств. Для этого вычислим значения:

b_i — число элементов, принадлежащих множеству A_i;

b_{i_1,i_2} — число элементов, принадлежащих множеству $A_{i_1} \cap A_{i_2}$;

$\ldots$

$b_{i_1,\ldots,i_k}$ — число элементов в множестве $A_{i_1} \cap \ldots \cap A_{i_k}$;

После этого вычислим:

$n_1 = b_1 + \ldots + b_m$ (сумма всех b с одним индексом);
$n_2 = \sum b_{i_1,i_2}$ (сумма всех b с двумя индексами);
$\ldots$
$n_k = \sum b_{i_1,\ldots,i_k}$ (сумма всех b с k индексами);

Тогда $n_0 = n - n_1 + n_2 - n_3 + \ldots + (-1)^m n_m$.

Можно понимать задачу так. Есть набор из n элементов. Есть m определённых свойств. Каждый элемент может обладать какими-то из них, а может и не обладать. Удобно считать, что элемент обладает i-м свойством, если и только если он принадлежит множеству A_i. Тогда сколько элементов не обладают ни одним свойством? По методу включения и исключения это число равно n_0. А сколько элементов обладают хотя бы одним свойством? Всё просто: $n - n_0$.

Задача 99. Ладьи на прогулке. Сколькими способами можно разместить 8 ладей на шахматной доске так, чтобы они, во-первых, не могли бить друг друга, а во-вторых, чтобы ни одна из ладей не стояла бы на белой главной диагонали?

Задача 100. Ящики и предметы I. Сколькими способами можно разместить m различных предметов по n ящикам так, чтобы ни один ящик не был пуст?

Задача 101. Ящики и предметы II. Сколькими способами можно разместить m различных предметов по n ящикам так, чтобы ровно ℓ ящиков оказались пустыми?

Задача 102. Ящики и предметы III. Сколькими способами можно разместить m различных предметов по n ящикам так, чтобы не менее ℓ ящиков оказались пустыми?

Задача 103. Перестановки цифр*. Сколькими способами можно переставить в числе 12341234 цифры так, чтобы никакие две одинаковые цифры не шли друг за другом?

Задача 104. Функция Эйлера*. Пусть n — натуральное число, разложение которого на простые множители имеет вид

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k},$$

где $p_1, p_2, \ldots, p_k$ — простые числа. Пусть $\varphi(n)$ — количество натуральных чисел, не превышающих n и взаимно простых с n (это и есть *функция Эйлера* числа n). Докажите, что

$$\varphi(n) = n \cdot (1 - \frac{1}{p_1}) \cdot (1 - \frac{1}{p_2}) \cdot \ldots \cdot (1 - \frac{1}{p_k}).$$

Задача 105. О беспорядках*. Беспорядок на n элементах — это перестановка чисел $1, 2, \ldots, n$ такая, что ни одно число не стоит на своём месте. Например, (34152) — беспорядок, а (34125) — нет. Чему равно число беспорядочных перестановок на n элементах?

Задача 106. Задача мажордома.** К обеду за круглым столом приглашены n пар враждующих рыцарей ($n \geqslant 2$). Требуется рассадить их так, чтобы никакие два врага не сидели рядом. Докажите, что это можно сделать $\sum_{k=0}^{n} (-1)^k C_n^k 2^k (2n - k - 1)!$ способами.

Вместо заключения

Сколькими способами можно...	число
выбрать q-значный вектор длины n?	q^n
упорядочить n-элементное множество?	$n!$
выбрать k различных элементов из n-элементного множества?	C_n^k
разбить n-элементное множество на подмножества мощностей $k_1, \ldots, k_m$?	$C_n(k_1, \ldots, k_m)$
выбрать k элементов, каждый из которых может быть одного из n типов?	C_{n+k-1}^k
выбрать элемент из n-элементного множества, не принадлежащий ни одному из подмножеств $A_1, \ldots, A_m$?	$n - n_1 + \ldots (-1)^m n_m.$

Криптография в литературе

Криптографические методы нередко упоминаются в художественной литературе. Попробуйте угадать, из каких произведений взяты следующие отрывки и с чем они связаны.

Отрывок 1. «Савелий показал Саше, как надо перестукиваться: алфавит делится на шесть рядов, по пять букв в каждом. Первые удары означают ряд, вторые — место буквы в ряду. Между ударами короткие паузы — это ряд; между буквами паузы чуть длиннее, между словами ещё длиннее, царапание по стене — «кончил!» или «стоп!» или «повторите!». Паузы и интервалы совсем крошечные, у опытных заключенных они измеряются долями секунды. В паузах и главная трудность — если их не уловить, звуки сливаются, получается не та буква и теряется смысл.

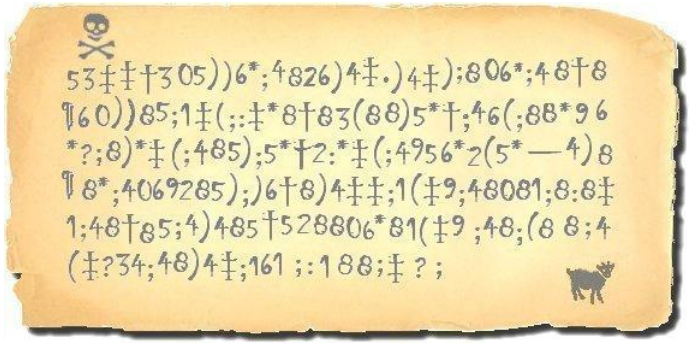
Обгоревшей спичкой Саша написал алфавит на картоне от папиросной коробки и начал перестукиваться. Стучал он медленно, с большими паузами, лежа на койке, прикрывшись одеялом, чтобы не услышал надзиратель. Сосед понимал его, но Саша понимал плохо, путал буквы, просил повторить, хотя сосед стучал чётко, ясно, с длинными паузами.»

Отрывок 2. «Штирлиц достал из книжного шкафа томик Монтеня, перевел цифры в слова и соотнёс эти слова с кодом, скрытым среди мудрых истин великого и спокойного французского мыслителя. «Кем они считают меня? — подумал он. — Гением или всемогущим? Это же немыслимо...» Думать так у Штирлица были все основания, потому что задание, переданное ему через московское радио, гласило: ...»

Отрывок 3. «Долгий овал его лица, острый профиль, властная манера разговаривать с тюремной администрацией и ещё тот едва голубоватый свет выцветших глаз, который дается только абстрактным умам, — всё это странно делало Челнова похожим не то на Декарта, не то на Архимеда. Он был прислан для разработки математических оснований абсолютного шифратора, то есть, прибора, который своим механическим вращением мог бы обеспечить включение и переключение множества реле, так запутывающих порядок посылки прямоугольных импульсов изуродованной речи, чтобы даже сотни людей, поставив аналогичные приборы, не могли бы расшифровать

разговора, идущего по проводам. В конструкторском бюро своим чередом шли поиски конструктивного решения подобного шифратора.»

Отрывок 4. «Через несколько минут, когда сковорода накалилась, я вынул пергамент и с невыразимым восторгом увидел, что кое-где на нём появились знаки, напоминавшие цифры и расположенные в строку. Я снова положил пергамент на сковороду и подержал ещё над огнем. Тут надпись выступила вся целиком — сейчас я вам покажу.



— Что ж! — сказал я, возвращая Леграну пергамент, — меня это не подвинуло бы ни на шаг. За все алмазы Голконды я не возьмусь решать подобную головоломку.»

Отрывок 5. «Андрей жил теперь на сумском подворье. Встретясь первый раз с Ваничкой и узнав, что работа близка к концу, дня два осталось, он, успокоенный, решил эти два дня посвятить учению: почитать с толком книгу «Кожевенное производство», купленную ещё в Петербурге. Читать было всё недосуг, а нужно. Вдруг — сообщение, Колька принес, Ваничкин подручный, писано шифром. Здесь, в Харькове, ключевым словом было «ШТУНДИСТЫ». Андрей ещё не привык читать сразу, в уме, пришлось набросать сетку: «Штундисты» написать колом, по-китайски, и затем к каждой букве приписать девять, следующих по алфавиту. В результате прочитал: «Срочно искать другое место пять на горке.»

Отрывок 6. В каком романе приводится следующий зашифрованный отрывок X главы «Евгения Онегина»? Дешифруйте его.

1. Властитель слабый и лукавый	1. Нечаянно пригретый славой
2. Его мы очень смирным знали	2. Орла двуглавого щипали
3. Гроза двенадцатого года	3. Остервенение народа
4. Но Бог помог — стал ропот ниже	4. Мы очутилися в Париже
5. И чем жирнее, тем тяжеле	5. Скажи, зачем ты в самом деле
6. Авось, о Шиболет народный	6. Но стихоплет великородный
7. Авось, аренды забывая	7. Авось по манью Николая
8. Сей муж судьбы, сей странник бранный	8. Сей всадник, папою венчанный
9. Тряслися грозно Пиринеи	9. Безрукий князь друзьям Мореи
10. Я всех уйму с моим народом	10. А про себя и в ус не дует
11. Потешный полк Петра Титана	11. Предавших некогда тирана
12. Россия присмирела снова	12. Но искра пламени иного
13. У них свои бывали сходки	13. Они за рюмкой русской водки
14. Витийством резким знамениты	14. У беспокойного Никиты
15. Друг Марса, Вакха и Венеры	15. Свои решительные меры
16. Так было над Невою льдистой	16. Блестит над каменкой тенистой
17. Плешивый щеголь, враг труда	17. Над нами царствовал тогда
18. Когда не наши повара	18. У Бонапартова шатра
19. Настала — кто тут нам помог?	19. Барклай, зима иль русский бог?
20. И скоро силою вещей	20. А русский царь главой царей
21. О русский глупый наш народ	21. ...
22. Тебе б я оду посвятил	22. Меня уже предупредил
23. Ханжа запрется в монастырь	23. Семействам возвратит Сибирь
24. Пред кем унизились цари	24. Исчезнувший как тень зари
25. Волкан Неаполя пылал	25. Из Кишинёва уж мигал
26. Наш царь в конгрессе говорил	26. Ты александровский холоп (?)
27. Дружина старых усачей	27. Свирепой шайке палачей
28. И пуще царь пошел кутить	28. Уже издавна, может быть
29. Они за чашею вина	29. ...
30. Сбирались члены сей семьи	30. У осторожного Ильи
31. Тут Лунин дерзко предлагал	31. И вдохновенно бормотал
32. Но там, где ранее весна	32. И над холмами Тульчина

Отрывок 7. «В штабном вагоне, где разместились офицеры маршевого батальона, с начала поездки царила странная тишина. Большинство офицеров углубилось в чтение небольшой книжки в полотняном переплете, озаглавленной «Die Sunden der Vater». Novelle von Ludwig Ganghofer[1]. Все одновременно сосредоточенно изучали стра-

[1] «Грехи отцов» Роман Людвига Гангофера (нем.)

ницу сто шестьдесят первую. Командир батальона капитан Сагнер стоял у окна и держал в руке ту же книжку, открытую на той же сто шестьдесят первой странице. <...>

— Also, meine Herren![2] — продолжал он торжественно. — Перед нами совершенно секретная информация, касающаяся новой системы шифровки полевых депеш. <...> Система, которую я вам объяснил, является не только одной из лучших, но, можно сказать, одной из самых непостигаемых. Все отделы контрразведки вражеских штабов теперь могут заткнуться, они скорее лопнут, чем разгадают наш шифр. Это нечто совершенно новое. Подобных шифров ещё никогда не бывало.

Дотошный кадет Биглер многозначительно кашлянул.

— Позволю себе обратить ваше внимание, господин капитан, — сказал он, — на книгу Керкгоффса о военной шифровке. Книгу эту каждый может заказать в издательстве Военного научного словаря. Там подробно описывается, господин капитан, метод, который вы нам только что объяснили.»

Отрывок 8. «Память, если донимать её вопросами, уподобляется луковице: при чистке обнаруживаются письмена, которые можно читать — букву за буквой. Только смысл редко бывает однозначным, а письмена выполнены зеркальным шрифтом или ещё как-нибудь зашифрованы.

Под первой, сухо шуршащей пергаментной кожицей находится следующая, которая, едва отслоившись, открывает влажную третью, под ней, перешептываясь, ждут свой черед четвёртая, пятая... И на каждой плёночке проступают давно хранившиеся слова или витиеватые знаки, будто некий тайнописец начертал их тогда, когда луковица ещё только нарождалась.

Сразу возникает тщеславное желание разгадать эти закорючки, взломать секретный код. И вот уже опровергается то, что ещё недавно претендовало на правду: ведь ложь и её младшая сестра подделка составляют наиболее устойчивую часть воспоминаний; если записать их на бумаге, они выглядят достоверными и не скупятся на подробности, убеждающие своей фотографической точностью: разогретый июльским солнцем толь на крыше сарая во дворике нашего дома пахнет в безветренные дни солодовыми леденцами...»

[2]Итак, господа! (нем.)

Ответы к задачам

Криптография в задачах

1. Указание: воспользуйтесь методом включений и исключений.
7. а) Да. б) Нет: $101 \cdot 127$. в) Нет: $517 \cdot 89$. г) Нет: 3697^2. д) Да. е) Нет: $17 \cdot 41 \cdot 233$. ж) Нет: $41 \cdot 61 \cdot 101$. з) Да.
9. 2^{2^n}.
13. а) $x_1 \oplus x_2 \oplus 1$; б) $x_1x_2 \oplus 1$; в) $x_1x_2x_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1 \oplus x_2 \oplus x_3$; г) $x_1 \oplus x_2 \oplus x_3$; д) $x_1x_2x_3$; е) $x_1x_2x_3 \oplus x_1x_3 \oplus 1$.
14. 2; 3; 0; 4.
15. 2^n (линейных); 2^{n+1} (аффинных); 2^s, где $s = \sum_{i=0}^{k} C_n^i$ (функций, степень которых не превосходит k).
17. 2^n — число вершин, $n2^{n-1}$ — число ребер. $C_n^k 2^{n-1}$.
18. C_n^r; $\sum_{i=0}^{r} C_n^i$.
19. 2^{n-k}.
20. $C_n^k 2^{n-k}$; 3^n.
21. C_n^k; $C_{n-\ell}^{n-k}$.
22. Если $x = 0$, то 2^n и 0 соответственно; если $x \neq 0$, то 2^{n-1} и 2^{n-1}.
23. Указание: воспользуйтесь теоремой 7.
28. Указание: докажите методом от противного.
34. $C_{2^n}^{2^{n-1}}$.
43. 1; 0; 0; 1; 1; если $n = 2$, то $AI(f) = 1$, иначе $AI(f) = 2$; 2.
55. а) (001111).
71. Подсказка: в этом сообщении ровно 120 букв.
76. 0; 2^{n-1}; 2^{n-1}.
80. C_{n+m}^n.
83. $(n!)^2$.
84. 8!.
85. $(n-2)!$.
86. $(n-1)!$.
88. 6; $10!/24$.
89. C_{16}^6.
98. 20.
99. 14 833.
100. $\sum_{k=0}^{n} (-1)^k C_n^k (n-k)^m$.
101. $C_n^\ell \sum_{k=0}^{n-\ell} (-1)^k C_{n-\ell}^k (n-\ell-k)^m$.
102. $\sum_{s=\ell}^{n} C_n^s \sum_{k=0}^{n-s} (-1)^k C_{n-s}^k (n-s-k)^m$.
103. 864.

105. $n!\sum_{k=0}^{n}(-1)^k/k!$, что при $n \to \infty$ равно $n!/e$.

Криптография в литературе

Отрывок 1. А. Н. Рыбаков, «Дети Арбата». Арестованный — студент Саша Панкратов. Описанный шифр — квадрат Полибия, или — тюремный шифр.
Отрывок 2. Ю. С. Семенов, «Семнадцать мгновений весны». Книжный шифр.
Отрывок 3. А. И. Солженицын, «В круге первом». Работа Марфинской шарашки.
Отрывок 4. Э. А. По, «Золотой жук». Зашифрован текст «A good glass in the bishop's hostel in the devil's seat twenty one degrees and thirteen minutes northeast and by north main branch seventh limb east side shoot from the left eye of the death's head a bee line from the tree through the shot fifty feet out.», его перевод «Хорошее стекло в трактире епископа на чёртовом стуле двадцать один градус и тринадцать минут северо-северо-восток главный сук седьмая ветвь восточная сторона стреляй из левого глаза мёртвой головы прямая от дерева через выстрел на пятьдесят футов.» В тексте указан путь к сокровищам капитана Кидда.
Отрывок 5. Ю. В. Трифонов, «Нетерпение». Ваничка — Иван Окладников, провокатор «Народной воли». Время подготовки несостоявшегося покушения на Александра II на железной дороге под г. Александровском.
Отрывок 6. В. А. Каверин, «Исполнение желаний». Ключ к дешифрованию:

1-левая, 17-левая, 1-правая, 17-правая
2-левая, 18-левая, 2-правая, 18-правая и т. д.

Отрывок 7. Я. Гашек, «Приключения бравого солдата Швейка». Капитан Сагнер объясняет офицерам «совершенно новый» «дополнительный цифровой метод», который, как выясняется, был взят из известной в то время книги Керкгоффса по военной криптографии. Кроме того, вместо нужного для шифрования второго тома книги «Грехи отцов» офицерами был получен первый том.
Отрывок 8. Г. Грасс, «Луковица памяти».

Приложение

CAST. S-блоки

Восемь S-блоков шифра CAST-128. Каждый вида $8 \to 32$.

Первый S-блок шифра CAST

30fb40d4	9fa0ff0b	6beccd2f	3f258c7a	1e213f2f	9c004dd3	6003e540	cf9fc949
bfd4af27	88bbbdb5	e2034090	98d09675	6e63a0e0	15c361d2	c2e7661d	22d4ff8e
28683b6f	c07fd059	ff2379c8	775f50e2	43c340d3	df2f8656	887ca41a	a2d2bd2d
a1c9e0d6	346c4819	61b76d87	22540f2f	2abe32e1	aa54166b	22568e3a	a2d341d0
66db40c8	a784392f	004dff2f	2db9d2de	97943fac	4a97c1d8	527644b7	b5f437a7
b82cbaef	d751d159	6ff7f0ed	5a097a1f	827b68d0	90ecf52e	22b0c054	bc8e5935
4b6d2f7f	50bb64a2	d2664910	bee5812d	b7332290	e93b159f	b48ee411	4bff345d
fd45c240	ad31973f	c4f6d02e	55fc8165	d5b1caad	a1ac2dae	a2d4b76d	c19b0c50
882240f2	0c6e4f38	a4e4bfd7	4f5ba272	564c1d2f	c59c5319	b949e354	b04669fe
b1b6ab8a	c71358dd	6385c545	110f935d	57538ad5	6a390493	e63d37e0	2a54f6b3
3a787d5f	6276a0b5	19a6fcdf	7a42206a	29f9d4d5	f61b1891	bb72275e	aa508167
38901091	c6b505eb	84c7cb8c	2ad75a0f	874a1427	a2d1936b	2ad286af	aa56d291
d7894360	425c750d	93b39e26	187184c9	6c00b32d	73e2bb14	a0bebc3c	54623779
64459eab	3f328b82	7718cf82	59a2cea6	04ee002e	89fe78e6	3fab0950	325ff6c2
81383f05	6963c5c8	76cb5ad6	d49974c9	ca180dcf	380782d5	c7fa5cf6	8ac31511
35e79e13	47da91d0	f40f9086	a7e2419e	31366241	051ef495	aa573b04	4a805d8d
548300d0	00322a3c	bf64cddf	ba57a68e	75c6372b	50afd341	a7c13275	915a0bf5
6b54bfab	2b0b1426	ab4cc9d7	449ccd82	f7fbf265	ab85c5f3	1b55db94	aad4e324
cfa4bd3f	2deaa3e2	9e204d02	c8bd25ac	eadf55b3	d5bd9e98	e31231b2	2ad5ad6c
954329de	adbe4528	d8710f69	aa51c90f	aa786bf6	22513f1e	aa51a79b	2ad344cc
7b5a41f0	d37cfbad	1b069505	41ece491	b4c332e6	032268d4	c9600acc	ce387e6d
bf6bb16c	6a70fb78	0d03d9c9	d4df39de	e01063da	4736f464	5ad328d8	b347cc96
75bb0fc3	98511bfb	4ffbcc35	b58bcf6a	e11f0abc	bfc5fe4a	a70aec10	ac39570a
3f04442f	6188b153	e0397a2e	5727cb79	9ceb418f	1cacd68d	2ad37c96	0175cb9d
c69dff09	c75b65f0	d9db40d8	ec0e7779	4744ead4	b11c3274	dd24cb9e	7e1c54bd
f01144f9	d2240eb1	9675b3fd	a3ac3755	d47c27af	51c85f4d	56907596	a5bb15e6
580304f0	ca042cf1	011a37ea	8dbfaadb	35ba3e4a	3526ffa0	c37b4d09	bc306ed9
98a52666	5648f725	ff5e569d	0ced63d0	7c63b2cf	700b45e1	d5ea50f1	85a92872
af1fbda7	d4234870	a7870bf3	2d3b4d79	42e04198	0cd0ede7	26470db8	f881814c
474d6ad7	7c0c5e5c	d1231959	381b7298	f5d2f4db	ab838653	6e2f1e23	83719c9e
bd91e046	9a56456e	dc39200c	20c8c571	962bda1c	e1e696ff	b141ab08	7cca89b9
1a69e783	02cc4843	a2f7c579	429ef47d	427b169c	5ac9f049	dd8f0f00	5c8165bf

Второй S-блок шифра CAST

1f201094	ef0ba75b	69e3cf7e	393f4380	fe61cf7a	eec5207a	55889c94	72fc0651
ada7ef79	4e1d7235	d55a63ce	de0436ba	99c430ef	5f0c0794	18dcdb7d	a1d6eff3
a0b52f7b	59e83605	ee15b094	e9ffd909	dc440086	ef944459	ba83ccb3	e0c3cdfb
d1da4181	3b092ab1	f997f1c1	a5e6cf7b	01420ddb	e4e7ef5b	25a1ff41	e180f806
1fc41080	179bee7a	d37ac6a9	fe5830a4	98de8b7f	77e83f4e	79929269	24fa9f7b
e113c85b	acc40083	d7503525	f7ea615f	62143154	0d554b63	5d681121	c866c359
3d63cf73	cee234c0	d4d87e87	5c672b21	071f6181	39f7627f	361e3084	e4eb573b
602f64a4	d63acd9c	1bbc4635	9e81032d	2701f50c	99847ab4	a0e3df79	ba6cf38c
10843094	2537a95e	f46f6ffe	a1ff3b1f	208cfb6a	8f458c74	d9e0a227	4ec73a34
fc884f69	3e4de8df	ef0e0088	3559648d	8a45388c	1d804366	721d9bfd	a58684bb
e8256333	844e8212	128d8098	fed33fb4	ce280ae1	27e19ba5	d5a6c252	e49754bd
c5d655dd	eb667064	77840b4d	a1b6a801	84db26a9	e0b56714	21f043b7	e5d05860
54f03084	066ff472	a31aa153	dadc4755	b5625dbf	68561be6	83ca6b94	2d6ed23b
eccf01db	a6d3d0ba	b6803d5c	af77a709	33b4a34c	397bc8d6	5ee22b95	5f0e5304
81ed6f61	20e74364	b45e1378	de18639b	881ca122	b96726d1	8049a7e8	22b7da7b
5e552d25	5272d237	79d2951c	c60d894c	488cb402	1ba4fe5b	a4b09f6b	1ca815cf
a20c3005	8871df63	b9de2fcb	0cc6c9e9	0beeff53	e3214517	b4542835	9f63293c
ee41e729	6e1d2d7c	50045286	1e6685f3	f33401c6	30a22c95	31a70850	60930f13
73f98417	a1269859	ec645c44	52c877a9	cdff33a6	a02b1741	7cbad9a2	2180036f
50d99c08	cb3f4861	c26bd765	64a3f6ab	80342676	25a75e7b	e4e6d1fc	20c710e6
cdf0b680	17844d3b	31eef84d	7e0824e4	2ccb49eb	846a3bae	8ff77888	ee5d60f6
7af75673	2fdd5cdb	a11631c1	30f66f43	b3faec54	157fd7fa	ef8579cc	d152de58
db2ffd5e	8f32ce19	306af97a	02f03ef8	99319ad5	c242fa0f	a7e3ebb0	c68e4906
b8da230c	80823028	dcdef3c8	d35fb171	088a1bc8	bec0c560	61a3c9e8	bca8f54d
c72feffa	22822e99	82c570b4	d8d94e89	8b1c34bc	301e16e6	273be979	b0ffeaa6
61d9b8c6	00b24869	b7ffce3f	08dc283b	43daf65a	f7e19798	7619b72f	8f1c9ba4
dc8637a0	16a7d3b1	9fc393b7	a7136eeb	c6bcc63e	1a513742	ef6828bc	520365d6
2d6a77ab	3527ed4b	821fd216	095c6e2e	db92f2fb	5eea29cb	145892f5	91584f7f
5483697b	2667a8cc	85196048	8c4bacea	833860d4	0d23e0f9	6c387e8a	0ae6d249
b284600c	d835731d	dcb1c647	ac4c56ea	3ebd81b3	230eabb0	6438bc87	f0b5b1fa
8f5ea2b3	fc184642	0a036b7a	4fb089bd	649da589	a345415e	5c038323	3e5d3bb9
43d79572	7e6dd07c	06dfdf1e	6c6cc4ef	7160a539	73bfbe70	83877605	4523ecf1

Третий S-блок шифра CAST

8defc240	25fa5d9f	eb903dbf	e810c907	47607fff	369fe44b	8c1fc644	aececa90
beb1f9bf	eefbcaea	e8cf1950	51df07ae	920e8806	f0ad0548	e13c8d83	927010d5
11107d9f	07647db9	b2e3e4d4	3d4f285e	b9afa820	fade82e0	a067268b	8272792e
553fb2c0	489ae22b	d4ef9794	125e3fbc	21fffcee	825b1bfd	9255c5ed	1257a240
4e1a8302	bae07fff	528246e7	8e57140e	3373f7bf	8c9f8188	a6fc4ee8	c982b5a5
a8c01db7	579fc264	67094f31	f2bd3f5f	40fff7c1	1fb78dfc	8e6bd2c1	437be59b
99b03dbf	b5dbc64b	638dc0e6	55819d99	a197c81c	4a012d6e	c5884a28	ccc36f71
b843c213	6c0743f1	8309893c	0feddd5f	2f7fe850	d7c07f7e	02507fbf	5afb9a04
a747d2d0	1651192e	af70bf3e	58c31380	5f98302e	727cc3c4	0a0fb402	0f7fef82
8c96fdad	5d2c2aae	8ee99a49	50da88b8	8427f4a0	1eac5790	796fb449	8252dc15
efbd7d9b	a672597d	ada840d8	45f54504	fa5d7403	e83ec305	4f91751a	925669c2
23efe941	a903f12e	60270df2	0276e4b6	94fd6574	927985b2	8276dbcb	02778176
f8af918d	4e48f79e	8f616ddf	e29d840e	842f7d83	340ce5c8	96bbb682	93b4b148
ef303cab	984faf28	779faf9b	92dc560d	224d1e20	8437aa88	7d29dc96	2756d3dc
8b907cee	b51fd240	e7c07ce3	e566b4a1	c3e9615e	3cf8209d	6094d1e3	cd9ca341
5c76460e	00ea983b	d4d67881	fd47572c	f76cedd9	bda8229c	127dadaa	438a074e
1f97c090	081bdb8a	93a07ebe	b938ca15	97b03cff	3dc2c0f8	8d1ab2ec	64380e51
68cc7bfb	d90f2788	12490181	5de5ffd4	dd7ef86a	76a2e214	b9a40368	925d958f
4b39fffa	ba39aee9	a4ffd30b	faf7933b	6d498623	193cbcfa	27627545	825cf47a
61bd8ba0	d11e42d1	cead04f4	127ea392	10428db7	8272a972	9270c4a8	127de50b
285ba1c8	3c62f44f	35c0eaa5	e805d231	428929fb	b4fcdf82	4fb66a53	0e7dc15b
1f081fab	108618ae	fcfd086d	f9ff2889	694bcc11	236a5cae	12deca4d	2c3f8cc5
d2d02dfe	f8ef5896	e4cf52da	95155b67	494a488c	b9b6a80c	5c8f82bc	89d36b45
3a609437	ec00c9a9	44715253	0a874b49	d773bc40	7c34671c	02717ef6	4feb5536
a2d02fff	d2bf60c4	d43f03c0	50b4ef6d	07478cd1	006e1888	a2e53f55	b9e6d4bc
a2048016	97573833	d7207d67	de0f8f3d	72f87b33	abcc4f33	7688c55d	7b00a6b0
947b0001	570075d2	f9bb88f8	8942019e	4264a5ff	856302e0	72dbd92b	ee971b69
6ea22fde	5f08ae2b	af7a616d	e5c98767	cf1febd2	61efc8c2	f1ac2571	cc8239c2
67214cb8	b1e583d1	b7dc3e62	7f10bdce	f90a5c38	0ff0443d	606e6dc6	60543a49
5727c148	2be98a1d	8ab41738	20e1be24	af96da0f	68458425	99833be5	600d457d
282f9350	8334b362	d91d1120	2b6d8da0	642b1e31	9c305a00	52bce688	1b03588a
f7baefd5	4142ed9c	a4315c11	83323ec5	dfef4636	a133c501	e9d3531c	ee353783

Четвёртый S-блок шифра CAST

9db30420	1fb6e9de	a7be7bef	d273a298	4a4f7bdb	64ad8c57	85510443	fa020ed1
7e287aff	e60fb663	095f35a1	79ebf120	fd059d43	6497b7b1	f3641f63	241e4adf
28147f5f	4fa2b8cd	c9430040	0cc32220	fdd30b30	c0a5374f	1d2d00d9	24147b15
ee4d111a	0fca5167	71ff904c	2d195ffe	1a05645f	0c13fefe	081b08ca	05170121
80530100	e83e5efe	ac9af4f8	7fe72701	d2b8ee5f	06df4261	bb9e9b8a	7293ea25
ce84ffdf	f5718801	3dd64b04	a26f263b	7ed48400	547eebe6	446d4ca0	6cf3d6f5
2649abdf	aea0c7f5	36338cc1	503f7e93	d3772061	11b638e1	72500e03	f80eb2bb
abe0502e	ec8d77de	57971e81	e14f6746	c9335400	6920318f	081dbb99	ffc304a5
4d351805	7f3d5ce3	a6c866c6	5d5bcca9	daec6fea	9f926f91	9f46222f	3991467d
a5bf6d8e	1143c44f	43958302	d0214eeb	022083b8	3fb6180c	18f8931e	281658e6
26486e3e	8bd78a70	7477e4c1	b506e07c	f32d0a25	79098b02	e4eabb81	28123b23
69dead38	1574ca16	df871b62	211c40b7	a51a9ef9	0014377b	041e8ac8	09114003
bd59e4d2	e3d156d5	4fe876d5	2f91a340	557be8de	00eae4a7	0ce5c2ec	4db4bba6
e756bdff	dd3369ac	ec17b035	06572327	99afc8b0	56c8c391	6b65811c	5e146119
6e85cb75	be07c002	c2325577	893ff4ec	5bbfc92d	d0ec3b25	b7801ab7	8d6d3b24
20c763ef	c366a5fc	9c382880	0ace3205	aac9548a	eca1d7c7	041afa32	1d16625a
6701902c	9b757a54	31d477f7	9126b031	36cc6fdb	c70b8b46	d9e66a48	56e55a79
026a4ceb	52437eff	2f8f76b4	0df980a5	8674cde3	edda04eb	17a9be04	2c18f4df
b7747f9d	ab2af7b4	efc34d20	2e096b7c	1741a254	e5b6a035	213d42f6	2c1c7c26
61c2f50f	6552daf9	d2c231f8	25130f69	d8167fa2	0418f2c8	001a96a6	0d1526ab
63315c21	5e0a72ec	49bafefd	187908d9	8d0dbd86	311170a7	3e9b640c	cc3e10d7
d5cad3b6	0caec388	f73001e1	6c728aff	71eae2a1	1f9af36e	cfcbd12f	c1de8417
ac07be6b	cb44a1d8	8b9b0f56	013988c3	b1c52fca	b4be31cd	d8782806	12a3a4e2
6f7de532	58fd7eb6	d01ee900	24adffc2	f4990fc5	9711aac5	001d7b95	82e5e7d2
109873f6	00613096	c32d9521	ada121ff	29908415	7fbb977f	af9eb3db	29c9ed2a
5ce2a465	a730f32c	d0aa3fe8	8a5cc091	d49e2ce7	0ce454a9	d60acd86	015f1919
77079103	dea03af6	78a8565e	dee356df	21f05cbe	8b75e387	b3c50651	b8a5c3ef
d8eeb6d2	e523be77	c2154529	2f69efdf	afe67afb	f470c4b2	f3e0eb5b	d6cc9876
39e4460c	1fda8538	1987832f	ca007367	a99144f8	296b299e	492fc295	9266beab
b5676e69	9bd3ddda	df7e052f	db25701c	1b5e51ee	f65324e6	6afce36c	0316cc04
8644213e	b7dc59d0	7965291f	ccd6fd43	41823979	932bcdf6	b657c34d	4edfd282
7ae5290c	3cb9536b	851e20fe	9833557e	13ecf0b0	d3ffb372	3f85c5c1	0aef7ed2

Пятый S-блок шифра CAST

7ec90c04	2c6e74b9	9b0e66df	a6337911	b86a7fff	1dd358f5	44dd9d44	1731167f
08fbf1fa	e7f511cc	d2051b00	735aba00	2ab722d8	386381cb	acf6243a	69befd7a
e6a2e77f	f0c720cd	c4494816	ccf5c180	38851640	15b0a848	e68b18cb	4caadeff
5f480a01	0412b2aa	259814fc	41d0efe2	4e40b48d	248eb6fb	8dba1cfe	41a99b02
1a550a04	ba8f65cb	7251f4e7	95a51725	c106ecd7	97a5980a	c539b9aa	4d79fe6a
f2f3f763	68af8040	ed0c9e56	11b4958b	e1eb5a88	8709e6b0	d7e07156	4e29fea7
6366e52d	02d1c000	c4ac8e05	9377f571	0c05372a	578535f2	2261be02	d642a0c9
df13a280	74b55bd2	682199c0	d421e5ec	53fb3ce8	c8adedb3	28a87fc9	3d959981
5c1ff900	fe38d399	0c4eff0b	062407ea	aa2f4fb1	4fb96976	90c79505	b0a8a774
ef55a1ff	e59ca2c2	a6b62d27	e66a4263	df65001f	0ec50966	dfdd55bc	29de0655
911e739a	17af8975	32c7911c	89f89468	0d01e980	524755f4	03b63cc9	0cc844b2
bcf3f0aa	87ac36e9	e53a7426	01b3d82b	1a9e7449	64ee2d7e	cddbb1da	01c94910
b868bf80	0d26f3fd	9342ede7	04a5c284	636737b6	50f5b616	f24766e3	8eca36c1
136e05db	fef18391	fb887a37	d6e7f7d4	c7fb7dc9	3063fcdf	b6f589de	ec2941da
26e46695	b7566419	f654efc5	d08d58b7	48925401	c1bacb7f	e5ff550f	b6083049
5bb5d0e8	87d72e5a	ab6a6ee1	223a66ce	c62bf3cd	9e0885f9	68cb3e47	086c010f
a21de820	d18b69de	f3f65777	fa02c3f6	407edac3	cbb3d550	1793084d	b0d70eba
0ab378d5	d951fb0c	ded7da56	4124bbe4	94ca0b56	0f5755d1	e0e1e56e	6184b5be
580a249f	94f74bc0	e327888e	9f7b5561	c3dc0280	05687715	646c6bd7	44904db3
66b4f0a3	c0f1648a	697ed5af	49e92ff6	309e374f	2cb6356a	85808573	4991f840
76f0ae02	083be84d	28421c9a	44489406	736e4cb8	c1092910	8bc95fc6	7d869cf4
134f616f	2e77118d	b31b2be1	aa90b472	3ca5d717	7d161bba	9cad9010	af462ba2
9fe459d2	45d34559	d9f2da13	dbc65487	f3e4f94e	176d486f	097c13ea	631da5c7
445f7382	175683f4	cdc66a97	70be0288	b3cdcf72	6e5dd2f3	20936079	459b80a5
be60e2db	a9c23101	eba5315c	224e42f2	1c5c1572	f6721b2c	1ad2fff3	8c25404e
324ed72f	4067b7fd	0523138e	5ca3bc78	dc0fd66e	75922283	784d6b17	58ebb16e
44094f85	3f481d87	fcfeae7b	77b5ff76	8c2302bf	aaf47556	5f46b02a	2b092801
3d38f5f7	0ca81f36	52af4a8a	66d5e7c0	df3b0874	95055110	1b5ad7a8	f61ed5ad
6cf6e479	20758184	d0cefa65	88f7be58	4a046826	0ff6f8f3	a09c7f70	5346aba0
5ce96c28	e176eda3	6bac307f	376829d2	85360fa9	17e3fe2a	24b79767	f5a96b20
d6cd2595	68ff1ebf	7555442c	f19f06be	f9e0659a	eeb9491d	34010718	bb30cab8
e822fe15	88570983	750e6249	da627e55	5e76ffa8	b1534546	6d47de08	efe9e7d4

Шестой S-блок шифра CAST

f6fa8f9d	2cac6ce1	4ca34867	e2337f7c	95db08e7	016843b4	eced5cbc	325553ac
bf9f0960	dfa1e2ed	83f0579d	63ed86b9	1ab6a6b8	de5ebe39	f38ff732	8989b138
33f14961	c01937bd	f506c6da	e4625e7e	a308ea99	4e23e33c	79cbd7cc	48a14367
a3149619	fec94bd5	a114174a	eaa01866	a084db2d	09a8486f	a888614a	2900af98
01665991	e1992863	c8f30c60	2e78ef3c	d0d51932	cf0fec14	f7ca07d2	d0a82072
fd41197e	9305a6b0	e86be3da	74bed3cd	372da53c	4c7f4448	dab5d440	6dba0ec3
083919a7	9fbaeed9	49dbcfb0	4e670c53	5c3d9c01	64bdb941	2c0e636a	ba7dd9cd
ea6f7388	e70bc762	35f29adb	5c4cdd8d	f0d48d8c	b88153e2	08a19866	1ae2eac8
284caf89	aa928223	9334be53	3b3a21bf	16434be3	9aea3906	efe8c36e	f890cdd9
80226dae	c340a4a3	df7e9c09	a694a807	5b7c5ecc	221db3a6	9a69a02f	68818a54
ceb2296f	53c0843a	fe893655	25bfe68a	b4628abc	cf222ebf	25ac6f48	a9a99387
53bddb65	e76ffbe7	e967fd78	0ba93563	8e342bc1	e8a11be9	4980740d	c8087dfc
8de4bf99	a11101a0	7fd37975	da5a26c0	e81f994f	9528cd89	fd339fed	b87834bf
5f04456d	22258698	c9c4c83b	2dc156be	4f628daa	57f55ec5	e2220abe	d2916ebf
4ec75b95	24f2c3c0	42d15d99	cd0d7fa0	7b6e27ff	a8dc8af0	7345c106	f41e232f
35162386	e6ea8926	3333b094	157ec6f2	372b74af	692573e4	e9a9d848	f3160289
3a62ef1d	a787e238	f3a5f676	74364853	20951063	4576698d	b6fad407	592af950
36f73523	4cfb6e87	7da4cec0	6c152daa	cb0396a8	c50dfe5d	fcd707ab	0921c42f
89dff0bb	5fe2be78	448f4f33	754613c9	2b05d08d	48b9d585	dc049441	c8098f9b
7dede786	c39a3373	42410005	6a091751	0ef3c8a6	890072d6	28207682	a9a9f7be
bf32679d	d45b5b75	b353fd00	cbb0e358	830f220a	1f8fb214	d372cf08	cc3c4a13
8cf63166	061c87be	88c98f88	6062e397	47cf8e7a	b6c85283	3cc2acfb	3fc06976
4e8f0252	64d8314d	da3870e3	1e665459	c10908f0	513021a5	6c5b68b7	822f8aa0
3007cd3e	74719eef	dc872681	073340d4	7e432fd9	0c5ec241	8809286c	f592d891
08a930f6	957ef305	b7fbffbd	c266e96f	6fe4ac98	b173ecc0	bc60b42a	953498da
fba1ae12	2d4bd736	0f25faab	a4f3fceb	e2969123	257f0c3d	9348af49	361400bc
e8816f4a	3814f200	a3f94043	9c7a54c2	bc704f57	da41e7f9	c25ad33a	54f4a084
b17f5505	59357cbe	edbd15c8	7f97c5ab	ba5ac7b5	b6f6deaf	3a479c3a	5302da25
653d7e6a	54268d49	51a477ea	5017d55b	d7d25d88	44136c76	0404a8c8	b8e5a121
b81a928a	60ed5869	97c55b96	eaec991b	29935913	01fdb7f1	088e8dfa	9ab6f6f5
3b4cbf9f	4a5de3ab	e6051d35	a0e1d855	d36b4cf1	f544edeb	b0e93524	bebb8fbd
a2d762cf	49c92f54	38b5f331	7128a454	48392905	a65b1db8	851c97bd	d675cf2f

Седьмой S-блок шифра CAST

85e04019	332bf567	662dbfff	cfc65693	2a8d7f6f	ab9bc912	de6008a1	2028da1f
0227bce7	4d642916	18fac300	50f18b82	2cb2cb11	b232e75c	4b3695f2	b28707de
a05fbcf6	cd4181e9	e150210c	e24ef1bd	b168c381	fde4e789	5c79b0d8	1e8bfd43
4d495001	38be4341	913cee1d	92a79c3f	089766be	baeeadf4	1286becf	b6eacb19
2660c200	7565bde4	64241f7a	8248dca9	c3b3ad66	28136086	0bd8dfa8	356d1cf2
107789be	b3b2e9ce	0502aa8f	0bc0351e	166bf52a	eb12ff82	e3486911	d34d7516
4e7b3aff	5f43671b	9cf6e037	4981ac83	334266ce	8c9341b7	d0d854c0	cb3a6c88
47bc2829	4725ba37	a66ad22b	7ad61f1e	0c5cbafa	4437f107	b6e79962	42d2d816
0a961288	e1a5c06e	13749e67	72fc081a	b1d139f7	f9583745	cf19df58	bec3f756
c06eba30	07211b24	45c28829	c95e317f	bc8ec511	38bc46e9	c6e6fa14	bae8584a
ad4ebc46	468f508b	7829435f	f124183b	821dba9f	aff60ff4	ea2c4e6d	16e39264
92544a8b	009b4fc3	aba68ced	9ac96f78	06a5b79a	b2856e6e	1aec3ca9	be838688
0e0804e9	55f1be56	e7e5363b	b3a1f25d	f7debb85	61fe033c	16746233	3c034c28
da6d0c74	79aac56c	3ce4e1ad	51f0c802	98f8f35a	1626a49f	eed82b29	1d382fe3
0c4fb99a	bb325778	3ec6d97b	6e77a6a9	cb658b5c	d45230c7	2bd1408b	60c03eb7
b9068d78	a33754f4	f430c87d	c8a71302	b96d8c32	ebd4e7be	be8b9d2d	7979fb06
e7225308	8b75cf77	11ef8da4	e083c858	8d6b786f	5a6317a6	fa5cf7a0	5dda0033
f28ebfb0	f5b9c310	a0eac280	08b9767a	a3d9d2b0	79d34217	021a718d	9ac6336a
2711fd60	438050e3	069908a8	3d7fedc4	826d2bef	4eeb8476	488dcf25	36c9d566
28e74e41	c2610aca	3d49a9cf	bae3b9df	b65f8de6	92aeaf64	3ac7d5e6	9ea80509
f22b017d	a4173f70	dd1e16c3	15e0d7f9	50b1b887	2b9f4fd5	625aba82	6a017962
2ec01b9c	15488aa9	d716e740	40055a2c	93d29a22	e32dbf9a	058745b9	3453dc1e
d699296e	496cff6f	1c9f4986	dfe2ed07	b87242d1	19de7eae	053e561a	15ad6f8c
66626c1c	7154c24c	ea082b2a	93eb2939	17dcb0f0	58d4f2ae	9ea294fb	52cf564c
9883fe66	2ec40581	763953c3	01d6692e	d3a0c108	a1e7160e	e4f2dfa6	693ed285
74904698	4c2b0edd	4f757656	5d393378	a132234f	3d321c5d	c3f5e194	4b269301
c79f022f	3c997e7e	5e4f9504	3ffafbbd	76f7ad0e	296693f4	3d1fce6f	c61e45be
d3b5ab34	f72bf9b7	1b0434c0	4e72b567	5592a33d	b5229301	cfd2a87f	60aeb767
1814386b	30bcc33d	38a0c07d	fd1606f2	c363519b	589dd390	5479f8e6	1cb8d647
97fd61a9	ea7759f4	2d57539d	569a58cf	e84e63ad	462e1b78	6580f87e	f3817914
91da55f4	40a230f3	d1988f35	b6e318d2	3ffa50bc	3d40f021	c3c0bdae	4958c24c
518f36b2	84b1d370	0fedce83	878ddada	f2a279c7	94e01be8	90716f4b	954b8aa3

Восьмой S-блок шифра CAST

e216300d	bbddfffc	a7ebdabd	35648095	7789f8b7	e6c1121b	0e241600	052ce8b5
11a9cfb0	e5952f11	ece7990a	9386d174	2a42931c	76e38111	b12def3a	37ddddfc
de9adeb1	0a0cc32c	be197029	84a00940	bb243a0f	b4d137cf	b44e79f0	049eedfd
0b15a15d	480d3168	8bbbde5a	669ded42	c7ece831	3f8f95e7	72df191b	7580330d
94074251	5c7dcdfa	abbe6d63	aa402164	b301d40a	02e7d1ca	53571dae	7a3182a2
12a8ddec	fdaa335d	176f43e8	71fb46d4	38129022	ce949ad4	b84769ad	965bd862
82f3d055	66fb9767	15b80b4e	1d5b47a0	4cfde06f	c28ec4b8	57e8726e	647a78fc
99865d44	608bd593	6c200e03	39dc5ff6	5d0b00a3	ae63aff2	7e8bd632	70108c0c
bbd35049	2998df04	980cf42a	9b6df491	9e7edd53	06918548	58cb7e07	3b74ef2e
522fffb1	d24708cc	1c7e27cd	a4eb215b	3cf1d2e2	19b47a38	424f7618	35856039
9d17dee7	27eb35e6	c9aff67b	36baf5b8	09c467cd	c18910b1	e11dbf7b	06cd1af8
7170c608	2d5e3354	d4de495a	64c6d006	bcc0c62c	3dd00db3	708f8f34	77d51b42
264f620f	24b8d2bf	15c1b79e	46a52564	f8d7e54e	3e378160	7895cda5	859c15a5
e6459788	c37bc75f	db07ba0c	0676a3ab	7f229b1e	31842e7b	24259fd7	f8bef472
835ffcb8	6df4c1f2	96f5b195	fd0af0fc	b0fe134c	e2506d3d	4f9b12ea	f215f225
a223736f	9fb4c428	25d04979	34c713f8	c4618187	ea7a6e98	7cd16efc	1436876c
f1544107	bedeee14	56e9af27	a04aa441	3cf7c899	92ecbae6	dd67016d	151682eb
a842eedf	fdba60b4	f1907b75	20e3030f	24d8c29e	e139673b	efa63fb8	71873054
b6f2cf3b	9f326442	cb15a4cc	b01a4504	f1e47d8d	844a1be5	bae7dfdc	42cbda70
cd7dae0a	57e85b7a	d53f5af6	20cf4d8c	cea4d428	79d130a4	3486ebfb	33d3cddc
77853b53	37effcb5	c5068778	e580b3e6	4e68b8f4	c5c8b37e	0d809ea2	398feb7c
132a4f94	43b7950e	2fee7d1c	223613bd	dd06caa2	37df932b	c4248289	acf3ebc3
5715f6b7	ef3478dd	f267616f	c148cbe4	9052815e	5e410fab	b48a2465	2eda7fa4
e87b40e4	e98ea084	5889e9e1	efd390fc	dd07d35b	db485694	38d7e5b2	57720101
730edebc	5b643113	94917e4f	503c2fba	646f1282	7523d24a	e0779695	f9c17a8f
7a5b2121	d187b896	29263a4d	ba510cdf	81f47c9f	ad1163ed	ea7b5965	1a00726e
11403092	00da6d77	4a0cdd61	ad1f4603	605bdfb0	9eedc364	22ebe6a8	cee7d28a
a0e736a0	5564a6b9	10853209	c7eb8f37	2de705ca	8951570f	df09822b	bd691a6c
aa12e4f2	87451c0f	e0f6a27a	3ada4819	4cf1764f	0d771c2b	67cdb156	350d8384
5938fa0f	42399ef3	36997b07	0e84093d	4aa93e61	8360d87b	1fa98b0c	1149382c
e97625a5	0614d1b7	0e25244b	0c768347	589e8d82	0d2059d1	a466bb1e	f8da0a82
04f19130	ba6e4ec0	99265164	1ee7230d	50b2ad80	eaee6801	8db2a283	ea8bf59e

CAST. Раундовые подключи

Пусть $K =$`x0x1x2x3x4x5x6x7x8x9xAxBxCxDxExF` — 128-битный ключ шифрования, где `x0` представляет крайний левый байт ключа, а `xF` — крайне правый. Через `z0 ... zF` обозначим вспомогательные байты. Маскирующие $Km_1, \ldots, Km_{16}$ и вращательные $Kr_1, \ldots, Kr_{16}$ раундовые ключи формируются следующим образом.

Маскирующие ключи.

`z0z1z2z3 = x0x1x2x3` ⊕ `S5[xD]` ⊕ `S6[xF]` ⊕ `S7[xC]` ⊕ `S8[xE]` ⊕ `S7[x8]`
`z4z5z6z7 = x8x9xAxB` ⊕ `S5[z0]` ⊕ `S6[z2]` ⊕ `S7[z1]` ⊕ `S8[z3]` ⊕ `S8[xA]`
`z8z9zAzB = xCxDxExF` ⊕ `S5[z7]` ⊕ `S6[z6]` ⊕ `S7[z5]` ⊕ `S8[z4]` ⊕ `S5[x9]`
`zCzDzEzF = x4x5x6x7` ⊕ `S5[zA]` ⊕ `S6[z9]` ⊕ `S7[zB]` ⊕ `S8[z8]` ⊕ `S6[xB]`
`Km1 = S5[z8]` ⊕ `S6[z9]` ⊕ `S7[z7]` ⊕ `S8[z6]` ⊕ `S5[z2]`
`Km2 = S5[zA]` ⊕ `S6[zB]` ⊕ `S7[z5]` ⊕ `S8[z4]` ⊕ `S6[z6]`

```
Km3 = S5[zC] ⊕ S6[zD] ⊕ S7[z3] ⊕ S8[z2] ⊕ S7[z9]
Km4 = S5[zE] ⊕ S6[zF] ⊕ S7[z1] ⊕ S8[z0] ⊕ S8[zC]
x0x1x2x3 = z8z9zAzB ⊕ S5[z5] ⊕ S6[z7] ⊕ S7[z4] ⊕ S8[z6] ⊕ S7[z0]
x4x5x6x7 = z0z1z2z3 ⊕ S5[x0] ⊕ S6[x2] ⊕ S7[x1] ⊕ S8[x3] ⊕ S8[z2]
x8x9xAxB = z4z5z6z7 ⊕ S5[x7] ⊕ S6[x6] ⊕ S7[x5] ⊕ S8[x4] ⊕ S5[z1]
xCxDxExF = zCzDzEzF ⊕ S5[xA] ⊕ S6[x9] ⊕ S7[xB] ⊕ S8[x8] ⊕ S6[z3]
Km5 = S5[x3] ⊕ S6[x2] ⊕ S7[xC] ⊕ S8[xD] ⊕ S5[x8]
Km6 = S5[x1] ⊕ S6[x0] ⊕ S7[xE] ⊕ S8[xF] ⊕ S6[xD]
Km7 = S5[x7] ⊕ S6[x6] ⊕ S7[x8] ⊕ S8[x9] ⊕ S7[x3]
Km8 = S5[x5] ⊕ S6[x4] ⊕ S7[xA] ⊕ S8[xB] ⊕ S8[x7]
z0z1z2z3 = x0x1x2x3 ⊕ S5[xD] ⊕ S6[xF] ⊕ S7[xC] ⊕ S8[xE] ⊕ S7[x8]
z4z5z6z7 = x8x9xAxB ⊕ S5[z0] ⊕ S6[z2] ⊕ S7[z1] ⊕ S8[z3] ⊕ S8[xA]
z8z9zAzB = xCxDxExF ⊕ S5[z7] ⊕ S6[z6] ⊕ S7[z5] ⊕ S8[z4] ⊕ S5[x9]
zCzDzEzF = x4x5x6x7 ⊕ S5[zA] ⊕ S6[z9] ⊕ S7[zB] ⊕ S8[z8] ⊕ S6[xB]
Km9 = S5[z3] ⊕ S6[z2] ⊕ S7[zC] ⊕ S8[zD] ⊕ S5[z9]
Km10 = S5[z1] ⊕ S6[z0] ⊕ S7[zE] ⊕ S8[zF] ⊕ S6[zC]
Km11 = S5[z7] ⊕ S6[z6] ⊕ S7[z8] ⊕ S8[z9] ⊕ S7[z2]
Km12 = S5[z5] ⊕ S6[z4] ⊕ S7[zA] ⊕ S8[zB] ⊕ S8[z6]
x0x1x2x3 = z8z9zAzB ⊕ S5[z5] ⊕ S6[z7] ⊕ S7[z4] ⊕ S8[z6] ⊕ S7[z0]
x4x5x6x7 = z0z1z2z3 ⊕ S5[x0] ⊕ S6[x2] ⊕ S7[x1] ⊕ S8[x3] ⊕ S8[z2]
x8x9xAxB = z4z5z6z7 ⊕ S5[x7] ⊕ S6[x6] ⊕ S7[x5] ⊕ S8[x4] ⊕ S5[z1]
xCxDxExF = zCzDzEzF ⊕ S5[xA] ⊕ S6[x9] ⊕ S7[xB] ⊕ S8[x8] ⊕ S6[z3]
Km13 = S5[x8] ⊕ S6[x9] ⊕ S7[x7] ⊕ S8[x6] ⊕ S5[x3]
Km14 = S5[xA] ⊕ S6[xB] ⊕ S7[x5] ⊕ S8[x4] ⊕ S6[x7]
Km15 = S5[xC] ⊕ S6[xD] ⊕ S7[x3] ⊕ S8[x2] ⊕ S7[x8]
Km16 = S5[xE] ⊕ S6[xF] ⊕ S7[x1] ⊕ S8[x0] ⊕ S8[xD]
```

Вращательные ключи формируются аналогично, но только на основе уже измененного на предыдущем шаге блока `x0 ... xF`.

```
z0z1z2z3 = x0x1x2x3 ⊕ S5[xD] ⊕ S6[xF] ⊕ S7[xC] ⊕ S8[xE] ⊕ S7[x8]
z4z5z6z7 = x8x9xAxB ⊕ S5[z0] ⊕ S6[z2] ⊕ S7[z1] ⊕ S8[z3] ⊕ S8[xA]
z8z9zAzB = xCxDxExF ⊕ S5[z7] ⊕ S6[z6] ⊕ S7[z5] ⊕ S8[z4] ⊕ S5[x9]
zCzDzEzF = x4x5x6x7 ⊕ S5[zA] ⊕ S6[z9] ⊕ S7[zB] ⊕ S8[z8] ⊕ S6[xB]
Kr1 = S5[z8] ⊕ S6[z9] ⊕ S7[z7] ⊕ S8[z6] ⊕ S5[z2]
Kr2 = S5[zA] ⊕ S6[zB] ⊕ S7[z5] ⊕ S8[z4] ⊕ S6[z6]
Kr3 = S5[zC] ⊕ S6[zD] ⊕ S7[z3] ⊕ S8[z2] ⊕ S7[z9]
Kr4 = S5[zE] ⊕ S6[zF] ⊕ S7[z1] ⊕ S8[z0] ⊕ S8[zC]
x0x1x2x3 = z8z9zAzB ⊕ S5[z5] ⊕ S6[z7] ⊕ S7[z4] ⊕ S8[z6] ⊕ S7[z0]
x4x5x6x7 = z0z1z2z3 ⊕ S5[x0] ⊕ S6[x2] ⊕ S7[x1] ⊕ S8[x3] ⊕ S8[z2]
x8x9xAxB = z4z5z6z7 ⊕ S5[x7] ⊕ S6[x6] ⊕ S7[x5] ⊕ S8[x4] ⊕ S5[z1]
xCxDxExF = zCzDzEzF ⊕ S5[xA] ⊕ S6[x9] ⊕ S7[xB] ⊕ S8[x8] ⊕ S6[z3]
Kr5 = S5[x3] ⊕ S6[x2] ⊕ S7[xC] ⊕ S8[xD] ⊕ S5[x8]
```

```
Kr6 = S5[x1] ⊕ S6[x0] ⊕ S7[xE] ⊕ S8[xF] ⊕ S6[xD]
Kr7 = S5[x7] ⊕ S6[x6] ⊕ S7[x8] ⊕ S8[x9] ⊕ S7[x3]
Kr8 = S5[x5] ⊕ S6[x4] ⊕ S7[xA] ⊕ S8[xB] ⊕ S8[x7]
z0z1z2z3 = x0x1x2x3 ⊕ S5[xD] ⊕ S6[xF] ⊕ S7[xC] ⊕ S8[xE] ⊕ S7[x8]
z4z5z6z7 = x8x9xAxB ⊕ S5[z0] ⊕ S6[z2] ⊕ S7[z1] ⊕ S8[z3] ⊕ S8[xA]
z8z9zAzB = xCxDxExF ⊕ S5[z7] ⊕ S6[z6] ⊕ S7[z5] ⊕ S8[z4] ⊕ S5[x9]
zCzDzEzF = x4x5x6x7 ⊕ S5[zA] ⊕ S6[z9] ⊕ S7[zB] ⊕ S8[z8] ⊕ S6[xB]
Kr9 = S5[z3] ⊕ S6[z2] ⊕ S7[zC] ⊕ S8[zD] ⊕ S5[z9]
Kr10 = S5[z1] ⊕ S6[z0] ⊕ S7[zE] ⊕ S8[zF] ⊕ S6[zC]
Kr11 = S5[z7] ⊕ S6[z6] ⊕ S7[z8] ⊕ S8[z9] ⊕ S7[z2]
Kr12 = S5[z5] ⊕ S6[z4] ⊕ S7[zA] ⊕ S8[zB] ⊕ S8[z6]
x0x1x2x3 = z8z9zAzB ⊕ S5[z5] ⊕ S6[z7] ⊕ S7[z4] ⊕ S8[z6] ⊕ S7[z0]
x4x5x6x7 = z0z1z2z3 ⊕ S5[x0] ⊕ S6[x2] ⊕ S7[x1] ⊕ S8[x3] ⊕ S8[z2]
x8x9xAxB = z4z5z6z7 ⊕ S5[x7] ⊕ S6[x6] ⊕ S7[x5] ⊕ S8[x4] ⊕ S5[z1]
xCxDxExF = zCzDzEzF ⊕ S5[xA] ⊕ S6[x9] ⊕ S7[xB] ⊕ S8[x8] ⊕ S6[z3]
Kr13 = S5[x8] ⊕ S6[x9] ⊕ S7[x7] ⊕ S8[x6] ⊕ S5[x3]
Kr14 = S5[xA] ⊕ S6[xB] ⊕ S7[x5] ⊕ S8[x4] ⊕ S6[x7]
Kr15 = S5[xC] ⊕ S6[xD] ⊕ S7[x3] ⊕ S8[x2] ⊕ S7[x8]
Kr16 = S5[xE] ⊕ S6[xF] ⊕ S7[x1] ⊕ S8[x0] ⊕ S8[xD]
```

Криптографические термины на английском языке

affine approximation	аффинное приближение
algebraic cryptanalysis	алгебраический криптоанализ
algebraic immune function	алгебраически иммунная функция
almost perfect nonlinear function	почти совершенно нелинейная функция
authentication	аутентификация
balanced function	сбалансированная функция (или уравновешенная функция)
bent function	бент-функция
birthday paradox	парадокс дней рождения
block cipher	блочный шифр
Boolean function	булева функция
brute-force attack	атака грубой силой (метод полного опробования ключей)
checksum	контрольная сумма
chosen plaintext attack	атака с выбранным открытым текстом
cipher	шифр
cipher block chaining, CBC	сцепление блоков шифртекста
cipher feed back, CFB	обратная связь по шифртексту
ciphertext	шифртекст
collision	коллизия хэш-функции
confusion	перемешивание
correlation cryptanalysis	корреляционный криптоанализ
correlation immune function	корреляционно-иммунная функция
cryptanalysis	криптоанализ
cryptanalysis on related keys	криптоанализ на связанных ключах
cryptographic protocol	криптографический протокол
cryptography	криптография
cryptology	криптология
cryptosystem	криптосистема

deciphering	расшифрование
decryption	дешифрование (реже — расшифрование)
differential cryptanalysis	дифференциальный криптоанализ
differentially δ-uniform function	дифференциально δ-равномерная функция
diffusion	рассеивание
digital signature	цифровая подпись
distinguishing attack	атака различения
electronic code book, ECB	электронная кодовая книга
encryption	зашифрование
encryption function	функция зашифрования
entropy	энтропия
exhaustive key search	атака грубой силой (метод полного опробования ключей)
feedback polynomial	многочлен обратной связи
feedback shift register, FSR	регистр сдвига с обратной связью
Feistel cipher	шифр Фейстеля
fingerprint	контрольная сумма; отпечаток
Hamming distance	расстояние Хэмминга
Hamming weight	вес Хэмминга
hash function	хэш-функция
high algebraic degree	высокая алгебраическая степень
identification	идентификация
integrity	целостность
integrity protection	имитозащита
IV, initial vector	начальный вектор (как правило, для хэш-функций)
Kerckhoffs' Requirements	правила Керкгоффса
key	ключ
key distribution problem	проблема распределения ключей

key length	длина ключа
key space	ключевое пространство
key-schedule cryptanalysis	криптоанализ на связанных ключах
keystream	гамма, ключевая последовательность
known plaintext attack	атака с известным открытым текстом
linear cryptanalysis	линейный криптоанализ
linear feedback shift register, LFSR	регистр сдвига с линейной обратной связью, РСЛОС
linear function	линейная функция
linear probability bias	линейное преобладание
meet-in-the-middle attack	криптоанализ метод встречи по середине
message authentication code	код аутентичности сообщения
monomial (power) function	мономиальная (степенная) функция
munge a file	применить к файлу хэш-функцию
nonlinearity	нелинейность
one time pad	одноразовый блокнот
one-to-one function	взаимно однозначная функция
one-way function	односторонняя функция
output feed back, OFB	обратная связь по выходу
P-box	P-блок
perfect secure system	совершенно секретная система
permutation	перестановка
piling-up lemma	лемма о набегании знаков
plaintext	открытый текст

preimage	прообраз
pretty good privacy, PGP	PGP
privacy	конфиденциальность информации
private key	секретный, закрытый ключ
pseudorandom generator, PRNG	генератор псевдослучайных последовательностей
pseudorandom sequence	псевдослучайная последовательность
public key	публичный, открытый ключ
public-key cryptography	асимметричная криптография
random key	случайный ключ
resilient function	устойчивая функция
resistance	стойкость
round	раунд
S-box	S-блок
security	безопасность
session key	сессионный ключ
slid pair	слайдовая пара
slide attack	слайдовая атака
SP network	SP-сеть
stream cipher	поточный шифр
subkey	подключ; раундовый ключ
substitution	подстановка
symmetric cipher	симметричный шифр
symmetric-key cryptography	симметричная криптография
time-space trade-off	компромисс время–память
trap-door one-way function	односторонняя функция с лазейкой
unicity distance	расстояние единственности
untracebility	неотслеживаемость
Walsh—Hadamard transform	преобразование Уолша — Адамара

Предметный указатель

Литература

1. *Агентура.ру* Российский интернет-ресурс, посвященный проблемам спецслужб, разведки и борьбы с терроризмом. URL: `www.agentura.ru`.

2. *Агибалов Г. П.* Избранные теоремы начального курса криптографии. Томск: Томский государственный университет, 2005. 116 с.

3. *Агибалов Г. П.* Методы решения систем полиномиальных уравнений над конечным полем // Вестник Томского государственного университета. Приложение. 2006. № 17. С. 4–9.

4. *Агибалов Г. П.* Элементы теории дифференциального криптоанализа итеративных блочных шифров с аддитивным раундовым ключом // Прикл. дискретная математика, 2008. № 1. С. 34–42.

5. *Агибалов Г. П.* 50 лет криптографии в Томском государственном университете // Прикл. дискретная математика, 2009. № 2. С. 104–126.

6. *Агибалов Г. П., Панкратова И. А.* Элементы теории статистических аналогов дискретных функций с применением в криптоанализе итеративных блочных шифров // Прикл. дискретная математика, 2010. № 3. С. 51–68.

7. *Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черёмушкин А. В.* Основы криптографии. М.: Гелиос АРВ, 2005. 480 с.

8. *Андреев Н.Н., Зубков А.М., Ивченко Г.И., Колчин В.Ф., Максимов Ю.И., Медведев Ю.И., Погорелов Б.А., Прохоров Ю.В., Сачков В.Н., Севастьянов Б.А.* Владимир Яковлевич Козлов (к девяностолетию со дня рождения) // Дискрет. математика, 2004. Т. 16. № 2, С. 3–6.

9. *Бабаш А. В., Шанкин Г. П.* История криптографии. Ч. 1. М.: Гелиос АРВ. 2002. 240 с.

10. *Бабаш А. В., Гольев Ю. И., Ларин Д. А., Шанкин Г. П.* О развитии криптографии в XIX веке // Защита информации. Конфидент. 2003. № 5. С. 90–96.

11. *Бабаш А. В., Гольев Ю. И., Ларин Д. А., Шанкин Г. П.* Криптографические идеи XIX века // Защита информации. Конфидент. 2004. № 1. С. 88–95; 2004. № 2. С. 92–96.

12. *Бабаш А. В., Гольев Ю. И., Ларин Д. А., Шанкин Г. П.* Криптографические идеи XIX века. Русская Криптография // Защита информации. Конфидент. 2004. № 3. С. 90–96.

13. *Бабаш А. В., Гольев Ю. И., Ларин Д. А., Шанкин Г. П.* Шифры революционного подполья России XIX века // Защита информации. Конфидент. 2004. № 4. С. 82–87.

14. *Бабаш А. В., Гольев Ю. И., Ларин Д. А., Шанкин Г. П.* Криптография в XIX веке // Информатика, 2004. Т. 466, № 33. М.: Издательский дом «Первое сентября». С. 17–23.

15. *Бабиевский В. В., Бутырский Л. С., Ларин Д. А., Шанкин Г. П.* Советская шифровальная служба: 1920 – 40-е В. // Статья в рамках совместного проекта с журналом «Защита информации. INSIDE». URL: `http://www.agentura.ru`

16. *Бабенко Л. К., Ищукова Е. А.* Современные алгоритмы блочного шифрования и методы их анализа. М.: Гелиос АРВ, 2006. 376 с. ISBN 5-85438-149-4.

17. Безопасность GSM: история, анализ, вскрытие // URL: `kiwibyrd.chat.ru/gsm/gsm-pap.htm` (дата обращения 05.05.2012).

18. *Букашкин С. А.* В. А. Котельников — основоположник секретной телефонии // Сб. «В. А. Котельников. Судьба, охватившая век»: в 2 т. М.: ФИЗМАТЛИТ, 2011. Т. 1 Воспоминания коллег. С. 21–24.

19. Гид в мире криптографии // URL: `www.cryptofaq.ru`

20. *Глухов М. М., Елизаров В. П., Нечаев А. А.* Алгебра: В 2 т. М.: Гелиос АРВ, 2003.

21. *Глухов М. М., Круглов И. А., Пичкур А. Б., Черёмушкин А. В.* Введение в теоретико-числовые методы криптографии. СПб.: Лань, 2011. 400 с.

22. *Гольев Ю. И., Ларин Д. А., Тришин А. Е., Шанкин Г. П.* Криптография: страницы истории тайных операций. М.: Гелиос АРВ. 2008. 288 с. ISBN 978-5-85438-177-2.

23. *Гольев Ю. И., Ларин Д. А., Тришин А. Е., Шанкин Г. П.* Криптографическая деятельность в период наполеоновских войн // Защита информации. Конфидент. 2004. № 5. С. 90–95.

24. *Гольев Ю. И., Ларин Д. А., Тришин А. Е., Шанкин Г. П.* Научно-технический прогресс и криптографическая деятельность в России XIX века // Защита информации. INSIDE. 2005. № 2. С. 67–75.

25. *Гольев Ю. И., Ларин Д. А., Тришин А. Е., Шанкин Г. П.* Начало войны в эфире // Защита информации. INSIDE. 2005. № 3. С. 89–96.

26. *Гольев Ю. И., Ларин Д. А., Тришин А. Е., Шанкин Г. П.* Криптографическая деятельность во время гражданской войны в России // Защита информации. INSIDE. 2005. № 4. С. 89–96.

27. *Гольев Ю. И., Ларин Д. А., Тришин А. Е., Шанкин Г. П.* Криптографическая деятельность революционеров в 20-х – 70-х годах XIX века в России: успехи и неудачи // Защита информации. INSIDE. 2005. № 5. С. 90–96.

28. *Гольев Ю. И., Ларин Д. А., Шанкин Г. П.* Криптографическая деятельность организаций «Земля и Воля» и «Народная Воля» в России в 1876–1881 годах // Защита информации. INSIDE. 2005. № 6. С. 80–87.

29. *Гольев Ю. И., Ларин Д. А., Шанкин Г. П.* Криптографическая деятельность революционеров в России. 1881–1887 годы: агония «Народной Воли» // Защита информации. INSIDE. 2006. № 2. С. 88–96.

30. *Гольев Ю. И., Ларин Д. А., Шанкин Г. П.* Криптографическая деятельность революционеров в России в 90-е годы XIX века // Защита информации. INSIDE. 2006. № 4. С. 84–91.

31. *Горький М.* «По союзу советов». Очерк V «Соловки» // Собрание сочинений в тридцати томах. М.: гос. изд. худ. лит., Т. 17. 1952. С. 201–220.

32. *Гузи Л.* Узник Соловецких островов Владимир Кривош-Неманич // Кафедра русистики, ФФ ПУ Прешов. Словакия. Специально для «Соловки Энциклопедия». 15.11.2005. URL `http://www.solovki.ca/camp_20/scientists.php` (дата обращения 07.08.2011).

33. *Гуляев Ю. В.* Краткая научная биография академика В. А. Котельникова. URL: `www.cplire.ru/alt/Kotelnikov/index.html` (дата обращения 01.01.2012).

34. *Закревский А. Д.* Метод автоматической шифрации сообщений // Прикл. дискретная математика. 2009. Т. 4, № 2. С. 127–137.

35. *Иванов А. В.* Близость к классу мономиальных аппроксимаций приведённого представления булевой функции в зависимости от выбора базиса, в котором оно задано // Прикл. дискретная математика. 2009. Приложение. № 1. С. 7–9.

36. Интервью: «Владимир Котельников: «Радио — главное открытие XX века»» // С. Лесков. URL: `fbm2000.ru/tp/in/rd.htm` (дата обращения 02.02.2012).

37. Интервью: «Н. Н. Андреев: Россия остаётся в числе лидеров мировой криптографии» // URL: `http://www.ssl.stu.neva.ru/psw/crypto/Andreev23.html` (дата обращения 15.10.2012).

38. *Калачёв К. Ф.* В круге третьем: Воспоминания и размышления о работе Марфинской лаборатории в 1948–1951 годах. М., 1999. 129 с.

39. *Кан Д.* Взломщики кодов. М.: Центрполиграф, 2000. Перевод книги Kahn D. The codebreakers, 1967.

40. «В. А. Котельников. Судьба, охватившая век»: В 2 т. М.: ФИЗМАТЛИТ, 2011. 312 с. ISBN 978-5-9221-1309-0.

41. *Кузьмин А. С., Марков В. Т., Нечаев А. А., Шишков А. Б.* Приближение булевых функций мономиальными // Дискретная математика. 2006. Т. 18, № 1. С. 9–29.

42. *Кузьмин А. С., Марков В. Т., Нечаев А. А., Шишкин В. А., Шишков А. Б.* Бент-функции и гипербент-функции над полем из 2^l элементов // Проблемы передачи информ. 2008. Т. 44, вып. 1. С. 15–37.

43. *Кузьмин А. С., Нечаев А. А., Шишкин В. А.* Бент- и гипербент-функции над конечным полем // Тр. по дискрет. матем. 2007. Т. 10. М.: Физматлит. С. 97–122.

44. *Куракин В. Л.* Алгоритм Берлекэмпа — Месси над конечными кольцами, модулями и бимодулями // Дискрет. матем. 1998. Т. 10, № 4. С. 3–34.

45. Лаборатория МГУ по математическим проблемам криптографии 1990–2000. Материалы к заседанию межведомственного междисциплинарного семинара по научным проблемам информационной безопасности 30 ноября 2000 г. М.: МГУ, 2000. 48 с.

46. *Ларин Д. А.* Советская шифровальная служба в годы Великой Отечественной войны // Проблемы образования, науки и культуры, 2011. Т. 86, № 1. С. 69–80. URL: `proceedings.usu.ru` (дата обращения 03.03.2012).

47. *Лидл Р., Нидеррайтер Г.* Конечные поля: В 2 т. М.: Мир, 1988. 822 с.

48. *Лихачёв Д. С.* Избранное: Великое наследие. Заметки о русском. СПб.: Logos, 1998. 560 с.

49. *Логачёв О. А., Сальников А. А., Ященко В. В.* Некоторые характеристики «нелинейности» групповых отображений // Дискрет. анализ и исслед. операций. Сер. 1. 2001. Т. 8, № 1. С. 40–54.

50. *Логачёв О. А., Сальников А. А., Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: Московский центр непрерывного математического образования, 2004. 470 с.

51. *Макаров С., Певцов Н., Попов Е., Сиверс М.* Телекоммуникационные технологии: введение в технологии GSM. М: Академия. 2008. ISBN: 978-5-7695-4770-6. 256 с.

52. *Масленников М. Е.* Криптография и свобода. Доступна по URL: `http://lib.rus.ec/b/145611` (дата обращения 05.05.2012).

53. *Нечаев В. И.* Элементы криптографии (Основы теории защиты информации) // М.: Высш. шк., 1999. 109 с. ISBN 5-06-003644-8.

54. *Новик В. К.* Христиан Гольдбах и Франц Эпинус (из истории шифровальных служб России XVIII века) // Математика и безопасность информационных технологий. Материалы конференции в МГУ 23–24 октября 2003 г. М.: МЦНМО, 2004. С. 87–110.

55. *Панасенко С. П.* Алгоритмы шифрования. Специальный справочник. СПб.: БХВ-Петербург, 2009. 576 с. ISBN 978-5-9775-0319-8.

56. *Первушин А. И.* Оккультный Сталин. М.: Яуза. 2006. ISBN: 5-87849-202-4.

57. Положение ФАПСИ «Система сертификации средств криптографической защиты информации». Октябрь 1993. Доступно по URL: `http://www.ancud.ru`

58. Приказ ФАПСИ от 23.09.1999 г. № 158 «Об утверждении положения о порядке разработки, производства, реализации и использования средств криптографической защиты информации с ограниченным доступом, не содержащей сведений, составляющих государственную тайну» // Российская газета. 2000. № 18.

59. Пятьдесят лет Институту криптографии, связи и информатики. Исторический очерк — Под ред. Б. А. Погорелова, М., 1999. 272 с.

60. *Руднев Е. В.* О нашей юности и сверстнике моём // Сб. «В. А. Котельников. Судьба, охватившая век»: в 2 т. М.: ФИЗМАТЛИТ, 2011. Т. 1 Воспоминания коллег. С. 19–20.

61. *Рябко Б. Я., Фионов А. Н.* Основы современной криптографии и стеганографии // М.: Горячая линия-Телеком, 2010. 232 с. ISBN 978-5-9912-0150-6.

62. Сайт благотворительного фонда им. И. Я. Верченко // URL: `www.verchenko.ru` (дата обращения 21.10.2012).

63. *Сачков В. Н.* Вклад выпускников МГУ в развитие теоретической криптографии в России во второй половине XX века // Московский университет и развитие криптографии в России. Материалы конференции в МГУ 17–18 октября 2002 г. М.: МЦНМО, 2003. С. 250–257.

64. *Сидельников В. М.* О взаимной корреляции последовательностей // Проблемы кибернетики. 1971. Т. 24. С. 15–42.

65. *Сингх С.* Книга шифров. Тайная история шифров и их расшифровки. М.: АСТ Астрель, 2006. 447 с.

66. *Соболева Т. А.* История шифровального дела в России. М.: ОЛМА-ПРЕСС, 2002. 512 с. ISBN 5-224-03634-8.

67. *Солдатов А., Бороган И.* Новое дворянство: Очерки истории ФСБ. М.: ООО «Юнайтед Пресс», 2011. 298 с. ISBN 978-5-4295-0023-2.

68. *Солодовников В. И.* Бент-функции из конечной абелевой группы в конечную абелеву группу // Дискретная математика. 2002. Т. 14, № 1. С. 99–113.

69. Соловки-энциклопедия. Digest project // URL: `www.solovki.ca` (дата обращения 05.05.2011).

70. *Токарева Н. Н.* Нелинейные булевы функции: бент-функции и их обобщения. LAP LAMBERT Academic Publishing (Saarbrucken, Germany), 2011. 170 с. ISBN: 978-3-8433-0904-2.

71. *Тужилин М. Э.* Алгебраический иммунитет булевых функций // Прикл. дискретная математика. 2008. № 2. С. 18–22.

72. *Тужилин М. Э.* Почти совершенные нелинейные функции // Прикл. дискретная математика. 2009. № 3. С. 14–20.

73. *Фергюсон Н., Шнайер Б.* Практическая криптография. М.: Вильямс, 2005. 424 с. ISBN 5-8459-0733-0.

74. Федеральная служба охраны Российской Федерации. URL: `www.fso.gov.ru`

75. *Фомичёв В. М.* Методы дискретной математики в криптологии. М.: Диалог-МИФИ, 2010. 424 с. ISBN 978-5-86404-234-2.

76. *Фомичёв В. М.* Исторические очерки о криптографии // М.: 2012.

77. Хабрахабр.ру, Блог. URL: `www.habrahabr.ru`.

78. Центр по лицензированию, сертификации и защите государственной тайны ФСБ России. URL: www.fsb.ru/fsb/supplement/contact/lsz.htm (дата обращения 01.01.2012).

79. *Шеннон К.* Работы по теории информации и кибернетике. М.: Изд-во иностр. лит., 1963. 832 с.

80. *Шнайер Б.* Прикл. криптография: Протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф. 2002. 816 с. ISBN 5-89392-055-4.

81. *Шюре Э.* Великие посвященные. Издательство: Книга–Принтшоп, 1990. 420 с. (Оригинал — 1914 г.). ISBN: 5-7160-0007-X.

82. *Ященко В. В.* О двух характеристиках нелинейности булевых отображений // Дискрет. анализ и исслед. операций. Сер. 1. 1998. Т. 5, № 2. С. 90–96.

83. *Adams C. M.* Constructing symmetric ciphers using the CAST design procedure // Designs, Codes and Cryptography. 1997. V. 12, N 3. P. 283–316.

84. *Agievich S. V.* On the representation of bent functions by bent rectangles // Fifth Int. Petrozavodsk conf. on probabilistic methods in discrete mathematics (Petrozavodsk, Russia, June 1–6, 2000). Proc. Boston: VSP, 2000. P. 121–135. URL: http://arxiv.org/abs/math/0502087 (дата обращения 01.01.2012).

85. *Baignères T., Junod P., Vaudenay S.* How far can we go beyond linear cryptanalysis? // Advances in Cryptology — ASIACRYPT'04, 10th International Conference on the Theory and Applications of Cryptology and Information Security (Jeju Island, Korea. December 5–9, 2004). Proc. Springer. 2004. P. 432–450 (Lecture Notes in Comput. Sci. V. 3329).

86. *Biham E., Shamir A.* Differential cryptanalysis of DES-like cryptosystems // J. Cryptology. 1991. V. 4, N 1. P. 3–72.

87. *Bracken C., Leander G.* New families of functions with differential uniformity of 4 // Fourth Intern. Conf. BFCA — Boolean Functions: Cryptography and Applications (Copenhagen, Denmark, May 19–21, 2008). Proc. to appear. P. 190–194.

88. *Braeken A.* Cryptographic properties of Boolean functions and S-boxes// Ph. D. Thesis. Katholieke Univ. Leuven, 2006. URL: http://www.cosic.esat.kuleuven.be/publications/thesis-129.pdf (дата обращения 02.10.2012).

89. *Budaghyan L., Carlet C., Leander G.* On inequivalence between known power AP№ functions // Fourth Intern. Conf. BFCA — Boolean Functions: Cryptography and Applications (Copenhagen, Denmark, May 19–21, 2008). Proc. P. 3–15.

90. *Budaghyan L., Pott A.* On differential uniformity and nonlinearity of functions // Discrete Math. 2009. V. 309, N 1. P. 371–384.

91. *Bury J.* Operation Stonka. An Ultimate Deception Spy Game // Cryptologia. 2011. V. 35, N 4. P. 297–327.

92. *Carlet C.* Boolean Functions for Cryptography and Error Correcting Codes // Chapter of the monograph «Boolean Methods and Models», Cambridge Univ. Press (P. Hammer, Y. Crama eds.). URL: www-rocq.inria.fr/secret/Claude.Carlet/chap-fcts-Bool.pdf (дата обращения 01.10.2012).

93. *Carlet C.* Vectorial Boolean Functions for Cryptography // Chapter of the monograph «Boolean Methods and Models», Cambridge Univ. Press (P. Hammer, Y. Crama eds.), to appear. URL: www-rocq.inria.fr/secret/Claude.Carlet/chap-vectorial-fcts.pdf (дата обращения 01.10.2012).

94. *Carlet C., Charpin P., Zinoviev V.* Codes, bent functions and permutations suitable for DES-like cryptosystems // Designs, Codes and Cryptography. 1998. V. 15, N 2. P. 125–156.

95. *Carlet C., Gaborit P.* Hyper-bent functions and cyclic codes // J. Combin. Theory. Ser. A. 2006. V. 113, № 3. P. 466–482.

96. *Carlet C., Klapper A.* Upper bounds on the numbers of resilient functions and of bent functions // 23rd Symposium on Information Theory (Benelux, Belgium. May, 2002). Proc. 2002. P. 307–314. URL: http://www.cs.engr.uky.edu/∼klapper/ps/bent.ps.

97. *Chabaud F., Vaudenay S.* Links between Differential and Linear Cryptanalysis // Advances in Cryptology — EUROCRYPT '94,

Intern. Conf. on the Theory and Application of Cryptographic Techniques. (Perugia, Italy. May 9–12, 1994) Proc. Springer, 1995. P. 356–365 (Lecture Notes in Comput. Sci. V. 950).

98. *Courtois N., Pieprzyk J.* Cryptanalysis of block ciphers with overdefined systems of equations // ASIACRYPT'02 — Advances in Cryptology. P. 267–287 (Lecture Notes in Comput. Sci. V. 2501).

99. *Courtois N., Bard G.* Algebraic cryptanalysis of Data Encryption Standard // IMA Int. Conf. 2007. P. 152–169.

100. *Courtois N.* An Improved Differential Attack on Full GOST // Cryptology ePrint Archive, Report 2012/138. `eprint.iacr.org`.

101. *Courtois N., Misztal M.* Differential cryptanalysis of GOST // Cryptology ePrint Archive, Report 2011/312. `eprint.iacr.org`.

102. *Courtois N., Misztal M.* Aggregated differentials and cryptanalysis of PP-1 and GOST // Proc. of CECC'2011 — Central European Conference on Cryptology, 2011 (Debrecen, Hungary, 30 June — 2 July, 2001).

103. Crypto++ Library — a free C++ class library of cryptographic schemes. URL: `www.cryptopp.com`.

104. *Cusick T. W., Stănică P.* Cryptographic Boolean Functions and Applications. Acad. Press. Elsevier. 2009. 245 p.

105. *Daemen J., Rijmen V.* The Design of Rijndael: AES — Advanced Encryption Standard. Springer. 2002. 256 p. ISBN-10: 3540425802.

106. *Diffie W.* SMS4 Encryption Algorithm for Wireless Networks // translated from Chinese. 2008.

107. *Dillon J. F.* A survey of bent functions // The NSA Technical J. 1972. Special Issue. P. 191–215.

108. *Dillon J. F.* Elementary Hadamard Difference sets // Ph. D. Thesis. Univ. of Maryland, 1974.

109. *Dobbertin H.* Almost perfect nonlinear power functions over $GF(2^n)$: the Niho case // Inform. and Comput. 1999. V. 151, N 1, 2. P. 57–72.

110. *Dobbertin H.* Almost perfect nonlinear power functions over $GF(2^n)$: a new case for n divisible by 5 // Finite Fields and Applications FQ5 (Augsburg, Germany, August 2–6, 2000). Proc. Springer. Eds. D. Jungnickel, H. Niederreiter. 2000. P. 113–121.

111. *Dobbertin H., Leander G.* A survey of some recent results on bent functions // Sequences and their applications — SETA 2004. 3rd Int. conference (Seoul, Korea, October 24–28, 2004). Revised papers. Berlin: Springer, 2005. P. 1–29 (Lecture Notes in Comput. Sci. V. 3486).

112. *Dobbertin H., Leander G., Canteaut A. et al.* Construction of Bent Functions via Niho Power Functions // J. Combin. Theory. Ser. A. 2006. V. 113. N 5. P. 779–798.

113. ENIGMA. Poznan mathematicians success // Познаньский университет, Польша. Фильм, 2008. Реж. J. Malinowska.

114. *Fon-Der-Flaass D. G.* A bound on correlation immunity // Siberian Elektron. Mat. Izv., 4 (2007), 133–135.

115. *Gong G., Golomb S. W.* Transform domain analysis of DES // IEEE Trans. Inform. Theory. 1999. V. 45, № 6. P. 2065–2073.

116. *Hammant T. R.* Russian and Soviet cryptology I — Some communications intelligence in tsarist Russia // Cryptologia. 2000. V. 24, N 3. P. 235–249.

117. *Hammant T. R.* Russian and Soviet cryptology II — The Magdeburg incident: the Russian view // Cryptologia. 2000. V. 24, N 4. P. 333–338.

118. *Hammant T. R.* Russian and Soviet cryptology III — Soviet Comint and the Civil War, 1918–1921 // Cryptologia. 2001. V. 25, N 1. P. 50–60.

119. *Heys H. M.* A Tutorial on Linear and Differential Cryptanalysis // Cryptologia. 2002. V. 26, N 3. P. 189–221.

120. *Isobe T.* A Single-Key Attack on the Full GOST Block Cipher // Fast Software Ecnryption — FSE, 2011. (Lyngby, Denmark, February 13–16, 2011). Proc. Springer-Verlag. 2011. P. 290–305 (Lecture Notes in Comput. Sci. V. 6733).

121. *Kahn D.* Soviet Comint in the Cold War // Cryptologia. 1998. V. 22, N 1. P. 1–24.

122. *Kapera Z. J.* Summary Report of the State of the Soviet Military Sigint in November 1942 Noticing «ENIGMA» // Cryptologia. 2011. V. 35, N 3, P. 247–256.

123. *Knudsen L. R., Robshaw M. J. B.* Non-linear approximation in linear cryptanalysis // Advances in Cryptology — EUROCRYPT'96. Workshop on the theory and application of cryptographic techniques (Saragossa, Spain. May 12–16, 1996). Proc. Springer-Verlag. 1996. P. 224–236 (Lecture Notes in Comput. Sci. V. 1070).

124. *Langevin P., Leander G.* Counting all bent functions in dimension eight 99270589265934370305785861242880 // Designs, Codes and Crypt. 2011. V. 59. P. 193–205.

125. *Leander N. G., Langevin P.* On exponents with highly divisible Fourier coefficients and conjectures of Niho and Dobbertin // Algebraic Geometry and its applications (France, May 7–11, 2007). Proc. 2008. P. 410–418.

126. *Matsui M.* Linear cryptanalysis method for DES cipher // Advances in Cryptology — EUROCRYPT'93. Workshop on the theory and application of cryptographic techniques (Lofthus, Norway, May 23–27, 1993). Proc. Berlin: Springer, 1994. P. 386–397 (Lecture Notes in Comput. Sci. V. 765).

127. *Matsui M.* The First Experimental Cryptanalysis of Data Encryption Standard // Advances in Cryptology — CRYPTO'94 (Santa Barbara, USA, August 21–25, 1994). Proc. Berlin: Springer, 1994. P. 1–11 (Lecture Notes in Comput. Sci. V. 839).

128. *McFarland R. L.* A family of difference sets in non-cyclic groups // J. Combin. Theory. Ser. A. 1973. V. 15, N 1. P. 1–10.

129. *Misztal M.* Частное сообщение. 2011.

130. *Musa M. A., Schaefer E. F., Weding S.* A simplified AES algorithm and its linear and differential cryptanalyses // Cryptologia. 2003. V. 17, N 2. P. 148–177.

131. *Nover H.* Algebraic cryptanalysis of AES: an overview // University of Wisconsin, USA, 2005.

132. *Nyberg K.* Perfect nonlinear S-boxes // Advances in cryptology — EUROCRYPT'1991. Int. conference on the theory and application of cryptographic techniques (Brighton, UK, April 8–11, 1991). Proc. Berlin: Springer, 1991. P. 378–386 (Lecture Notes in Comput. Sci. V. 547).

133. *Nyberg K.* Differentially uniform mappings for cryptography// Advances in cryptology — EUROCRYPT'1993. Int. conference on the theory and application of cryptographic techniques (Lofthus, Norway. May 23–27, 1993). Proc. Berlin: Springer, 1994. P. 55–64 (Lecture Notes in Comput. Sci. V. 765).

134. *Preneel B.* Частное сообщение. 2011.

135. *Rothaus O.* On bent functions // IDA CRD W.P. N 169. 1966.

136. *Rothaus O.* On bent functions // J. Combin. Theory. Ser. A. 1976. V. 20, N 3. P. 300–305.

137. *Seki H., Kaneko T.* Differential cryptanalysis of reduced rounds of GOST // Selected areas in cryptography — SAC 2000. Berlin: Springer, 2000. P. 315–323 (Lecture Notes in Comput. Sci. V. 2012).

138. *Schimmelpenninck van der Oye D.* Tsarist Codebreaking some Background and some examples // Cryptologia. 1998. V. 22, N 4. P. 342–353.

139. *Shorin V. V., Jelezniakov V. V., Gabidulin E. M.* Linear and Differential Cryptanalysis of Russian GOST // Preprint submitted to Elsevier Preprint, 4 April 2001.

140. *Tokareva N.* On the number of bent functions from iterative constructions: lower bounds and hypotheses // Advances in Mathematics of Communications (AMC). 2011. V. 5, N 4. P. 609–621.

141. *Wang L., Zhang J.* A best possible computable upper bound on bent functions// J. West of China. 2004. V. 33, N 2. P. 113–115.

142. *Yang M., Meng Q., Zhang H.* Evolutionary design of trace form bent functions// Cryptology ePrint Archive, Report 2005/322. URL: `http://eprint.iacr.org/` (дата обращения 01.01.2012).

143. *Youssef A. and Gong G.* Hyper-bent functions // Advances in cryptology — EUROCRYPT'2001. Int. conference on the theory and application of cryptographic techniques (Innsbruk, Austria. May 6–10, 2001). Proc. Berlin: Springer, 2001. P. 406–419 (Lecture Notes in Comput. Sci. V. 2045).

Учебное издание

Токарева Наталья Николаевна

СИММЕТРИЧНАЯ КРИПТОГРАФИЯ. КРАТКИЙ КУРС

Учебное пособие

Редактор *Н. В. Осипова*

Подписано в печать 24.10.2012 г.
Формат 70×100 1/16. Уч.-изд. л. 14,6. Усл. печ. л. 18,8.
Тираж 400 экз. Заказ № 267

Редакционно-издательский центр НГУ.
630090, Новосибирск, ул. Пирогова, 2.