ПРИОРИТЕТНЫЙ НАЦИОНАЛЬНЫЙ ПРОЕКТ «ОБРАЗОВАНИЕ»
РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

А.А. ВАРФОЛОМЕЕВ

СОВРЕМЕННАЯ ПРИКЛАДНАЯ КРИПТОГРАФИЯ

Учебное пособие

Москва
2008

Инновационная образовательная программа
Российского университета дружбы народов

«Создание комплекса инновационных образовательных программ и формирование инновационной образовательной среды, позволяющих эффективно реализовывать государственные интересы РФ через систему экспорта образовательных услуг»

Экспертное заключение –

кандидат технических наук, доцент *С.В. Запечников*

Варфоломеев А.А.

Современная прикладная криптография: Учеб. пособие. – М.: РУДН, 2008. – 218 с.: ил.

В учебном пособии представлены основные понятия криптографии и ее методы обеспечения безопасности информации. Приводятся и сравниваются современные отечественные и зарубежные криптографические стандарты. Основное внимание уделяется применению достижений криптографии для решения прикладных задач.

Учебное пособие входит в серию пособий по информационной безопасности и расширяет материал пособий «Основы информационной безопасности», «Защита информации с использованием интеллектуальных карт».

Учебное пособие выполнено в рамках инновационной образовательной программы Российского университета дружбы народов, направление «Комплекс экспортоориентированных инновационных образовательных программ по приоритетным направлениям науки и технологий», и входит в состав учебно-методического комплекса, включающего описание курса, программу и электронный учебник.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Долгое время криптография в основном применялась для обеспечения секретности переписки в дипломатии, военном деле, спецслужбами, и ее методы были известны только узкому кругу профессионалов-криптографов. Но в связи с развитием информационных технологий, в связи с возрастающей зависимостью от них жизни современного общества и необходимостью обеспечения информационной безопасности применение криптографических методов стало актуальным для всего общества. При этом обеспечение секретности не всегда является самым актуальным и уступает по значимости обеспечению целостности, подлинности и другим аспектам безопасности. Изобретение новых принципов криптографии и появление так называемой криптографии с открытым (или общедоступным) ключом дало мощный импульс для широкого использования криптографии для нужд гражданского общества, для нужд бизнеса, банковского дела и позволило обеспечить безопасность взаимодействия широкому кругу не обязательно доверяющих друг другу субъектов.

Необходимо различать теоретическую и прикладную криптографию. Для глубокого изучения вопросов теоретической криптографии приходится знакомиться с большим кругом математических дисциплин: теорией вероятностей и статистики, высшей алгеброй, теорией чисел и другими дисциплинами. Прикладная криптография, что соответствует названию, больше занимается вопросами применения достижений теоретической криптографии для нужд конкретных применений на практике.

Криптография в настоящее время – достаточно интенсивно развивающаяся область человеческих знаний. Об этом свидетельствует большой поток открытых научных публикаций, многочисленные научно-технические конференции: Eurocrypt, Crypto, Asiacrypt и др.

В связи со все большей интеграцией нашей страны в международное экономическое пространство особую роль играют общепринятые стандартные процедуры обеспечения защиты информации. Поэтому уделено внимание организациям, занимающимся разработкой и внедрением стандартов в интересующей нас области, сравнением отечественных стандартов с прошедшими широкую международную экспертизу соответствующими аналогами алгоритмов цифровой подписи и функции хэширования.

Основная масса литературы по данным вопросам опубликована на английском языке, для многих терминов еще нет общепринятых отечественных эквивалентов, к некоторым трудно найти подходящий удобный перевод.

В основу курса положен материал из известных и проверенных учебников по криптографии с добавлением последних результатов в этой области. Для безопасного применения криптографических методов защиты очень важно знать, даже не глубоко разбираясь, о последних достижениях математиков и криптографов в анализе криптографических систем и оценке их стойкости.

Раздел 1. ВВЕДЕНИЕ В КРИПТОЛОГИЮ

1.1. Первые понятия криптологии. Этапы развития

Криптология – наука о создании и анализе систем безопасной связи. Это одно из определений. Долгое время, говоря о криптологии, имели в виду не безопасную, а секретную связь, хотя секретность является только одним из аспектов безопасности. В настоящее время криптология занимается аспектами целостности, подлинности (аутентичности), неотказуемости, анонимности, а также всеми вопросами, возникающими при работе с документами. Современная криптология уже часто рассматривается как наука о методах создания и анализа систем, устойчивых к попыткам отклонения их от предписанного функционирования.

Криптологию принято делить на две части: криптографию и криптоанализ, в соответствии с аспектами синтеза и анализа. **Криптография** – наука о методах обеспечения безопасности, то есть более занимается вопросами синтеза систем. **Криптоанализ** – наука о методах атак на безопасность данных. У нас в стране имеет место и другая терминология, когда термин «криптография» использовался для названия всей науки, а криптоанализ назывался **дешифрованием**.

Цели криптографии менялись на протяжении всей ее истории. Сначала она служила больше для обеспечения секретности, чтобы препятствовать несанкционированному раскрытию информации, передаваемой по открытым каналам связи. С началом информационного века обнаружилась потребность применения криптографии и в частном секторе. Количество конфиденциальной информации огромно: истории болезней, юридические, финансовые документы и т.д.

Рассмотрим сначала вопрос обеспечения секретности информации.

Для сохранения тайны сообщения помимо криптографических способов применяются способы физической защиты и стеганографии. **Стеганография** занимается методами сокрытия самого факта передачи сообщения. Например,

можно писать скрытое сообщение (на свободном месте фактического сообщения) молоком, специальными чернилами и т.п. К методам стеганографии можно отнести шумоподобные методы радиопередачи. В настоящее время получила развитие **компьютерная стеганография**.

Как показала практика, наиболее эффективная защита информации обеспечивается на основе криптографических способов и, как правило, в сочетании с другими способами.

Исходное сообщение, которое должен защищать криптограф, называется **открытым текстом.**

Важным является понятие «шифра», которое иногда путают с понятием «код». **Шифр** – это множество обратимых преобразований (отображений) открытого текста, проводимых с целью его защиты.

Процесс применения конкретного преобразования шифра к открытому тексту называется **зашифрованием**, а результат этого преобразования – **шифртекстом** или **криптограммой.** Соответственно процесс обратного преобразования шифртекста в открытый текст называется **расшифрование**. Следует различать понятия «расшифрование» и «дешифрование». **Дешифрование** связано с действиями злоумышленника, противника, который желает нарушить безопасность информации. Совокупность данных, определяющих конкретное преобразование из множества преобразований шифра, называется **ключом.** Часто (но не всегда) ключ передается отправителем получателю каким-либо надежно защищенным способом заранее, до момента, когда возникает необходимость отправления сообщения по открытому каналу.

Раньше (XVII в.) для специального типа шифра использовалось название код. **Код** – это своего рода словарь, где элементы открытого текста (буквы, сочетания букв, слова и даже короткие фразы) – так называемые **кодвеличины** – заменяются группами символов (букв, цифр, других знаков). Эти группы символов называются **кодобозначениями.** В настоящее время под кодом, как правило, имеют в виду одно фиксированное преобразование

открытого текста, проводимое не с целью его защиты, а с целью устранения избыточности или для обнаружения и устранения искажений при передаче в канале с шумом. Этим занимается современная **теория кодирования**.

Важным понятием криптографии является понятие стойкости. **Стойкость** – это способность шифра противостоять попыткам хорошо вооруженного современной техникой и знаниями криптоаналитика нарушить секретность сообщения, дешифровать перехваченное сообщение, раскрыть ключи шифра или нарушить целостность и (или) подлинность информации. Далее это понятие будет уточняться. Понятие стойкости распространяется и на другие аспекты безопасности информации.

Можно выделить следующие **периоды развития криптологии**. Первый период – эра донаучной криптологии, когда она являлась ремеслом, уделом узкого круга искусных умельцев. Началом второго периода можно считать 1949 г., когда появилась работа К. Шеннона «Теория связи в секретных системах», в которой проведено фундаментальное научное исследование шифров и важнейших вопросов оценки их стойкости. Благодаря этому труду, криптология оформилась как прикладная математическая наука. И, наконец, начало третьему периоду было положено появлением в 1976 г. работы У. Диффи, М. Хеллмана «Новые направления в криптографии», где показано, что секретная связь возможна без полного доверия сторон и предварительного обмена ключами. Так началось и продолжается до настоящего времени бурное развитие наряду с обычной классической криптографией и криптографии с открытым ключом. Оба направления необходимы для практики, так как обладают своими сильными и слабыми сторонами.

Можно дать и другую классификацию этапов развития криптологии, например, по уровню развития используемых технических средств, начиная от ручной криптографии с различными вспомогательными устройствами до современных специализированных электронных шифрмашин и компьютеров.

1.2. Некоторые поясняющие примеры из истории

Приводимые далее примеры тайнописи служат для пояснения введенных понятий и для введения новых. Большинство из них можно найти в работах [2, 15, 34, 40]. Многие методы, примененные в далекие времена, используются до сих пор (например, перешифрование, рандомизация…).

Еще несколько веков назад само применение письменности можно было рассматривать как способ сокрытия информации, так как владение письменностью было уделом немногих.

XX в. до н.э. Один из самых древних шифртекстов был найден при раскопках в Месопотамии. Он был написан клинописью на глиняной табличке и содержал рецепт глазури для покрытия гончарных изделий, что, по-видимому, было коммерческой тайной. Известны древнеегипетские религиозные тексты и медицинские рецепты.

Середина IX в. до н.э. Именно в это время, как сообщает Плутарх, использовалось шифрующее устройство – скиталь, которое реализовывало так называемый **шифр перестановки.** При шифровании слова писались на узкую ленту, намотанную на цилиндр, вдоль образующей этого цилиндра (скиталя). После написания лента разматывалась, и на ней оставались переставленные буквы открытого текста. Неизвестным параметром – ключом – в данном случае служил диаметр этого цилиндра. Известен и метод дешифрования такого шифртекста, предложенный Аристотелем, который наматывал ленту на конус, и то место, где появлялось читаемое слово или его часть, определяло неизвестный диаметр цилиндра.

56 г. н.э. Во время войны с галлами Ю. Цезарь использует другую разновидность шифра – **шифр замены.** Под алфавитом открытого текста подписывлся тот же алфавит со сдвигом (у Цезаря на три позиции) по циклу. При шифровании буквы открытого текста из верхнего алфавита заменялись на буквы нижнего алфавита. Хотя этот шифр был известен до Ю. Цезаря, тем не менее, шифр был назван его именем.

Другим более сложным шифром замены является греческий шифр – «квадрат Полибия». Алфавит записывается в виде квадратной таблицы 5x5. При шифровании буквы открытого текста заменялись на пару чисел – номера столбца и строки этой буквы в таблице. При произвольном расписывании алфавита по таблице и шифровании такой таблицей короткого сообщения этот шифр является стойким даже по современным понятиям.

	1	2	3	4	5
1	A	B	C	D	T
2	F	G	H	I,J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Рис. 1. Квадрат Полибия для английского языка

Крах Римской империи в V в. н.э. сопровождался закатом искусства и наук, в том числе и криптографии. Церковь в те времена преследовала тайнопись, сокрытие мыслей за шифрами не позволяло церкви контролировать эти мысли.

Р. Бэкон (1214–1294) – францисканский монах и философ – описал семь систем секретного письма. Большинство шифров в те времена применялись для закрытия научных записей.

Середина XV в. Изобретение И. Гутенбергом книгопечатания привело к росту грамотности и увеличению числа людей, которые могли вести переписку. Развиваются межгосударственные отношения, тайнопись становится крайне необходимой.

Вторая половина XV в. Леон Баттиста Альберти – архитектор и математик. Работал в Ватикане. Автор книги о шифрах, где описал шифр замены на основе двух концентрических кругов, по периферии которых были

нанесены на одном круге – алфавит открытого текста, а на другом – алфавит шифрованного текста. Важно, что шифралфавит был не последовательным и мог быть смещен на любое количество шагов.

Именно Альберти впервые применил для дешифрования свойство неравномерности встречаемости различных букв в языке. Также он впервые предложил для повышения стойкости применять повторное шифрование с помощью разных шифрсистем. К перешифрованию надо относиться критически, так как в некоторых случаях оно не приводит к повышению стойкости.

Интересен факт, что король Франции Франциск I в 1546 г. издал указ, запрещающий подданным использование шифров. Хотя шифры того времени были исключительно простыми, они считались нераскрываемыми. В настоящее время в разных странах существуют ограничения на использование шифров.

Франсуа Виет (1540–1630) – французский математик. Всем известна со школьной скамьи теорема его имени. Он же был искусным дешифровальщиком, находясь при дворе короля Генриха IV.

Иоганн Тритемий (1462–1516) – монах-бенедиктинец, живший в Германии. Написал один из первых учебников по криптографии. Предложил оригинальный шифр многозначной замены под названием «Ave Maria». Каждая буква открытого текста имела не одну замену, а несколько, по выбору шифровальщика. Причем буквы заменялись буквами или словами так, что получался некоторый псевдооткрытый текст. То есть применялась стеганография вместе с криптографической защитой. Разновидность шифра многозначной замены.

Джироламо Кардано – итальянский математик, механик, врач. Изобрел свою систему шифрования, так называемую **решетку Кардано**. В куске картона с размеченной решеткой определенным образом прорезались отверстия, нумерованные в произвольном порядке. Чтобы получить шифртекст, нужно положить этот кусок картона на бумагу и вписывать в

отверстия буквы в выбранном порядке. После снятия картона промежутки бессмысленного набора букв дописывались до псевдоосмысловых фраз, так что можно было скрыть факт передачи секретного сообщения. Это легко достигается, если промежутки между отверстиями большие. Неудобство шифра в том, что кусок картона надо хранить в тайне.

XVI в. Шифры замены получили развитие в работах итальянца Джованни Батиста Порты и француза Блеза де Вижинера. Рассмотрим систему шифрования Вижинера подробно ввиду ее важности для дальнейшего изложения.

Система Вижинера требует для зашифрования ключ – в виде ключевого слова, которое подписывается сверху над открытым текстом периодически. Каждая буква открытого текста заменяется на букву, стоящую в алфавите далее ее (по циклу) на число позиций, определяемое номером в алфавите стоящей сверху буквы ключевого слова (минус 1). Для сравнения заметим, что в «шифре Цезаря» это ключевое слово состоит из одной буквы, стоящей на третьем месте в алфавите. Для удобства пользования изготовливалась легко запоминающаяся таблица.

Буквы ключевого слова

	A	B	C	D	. . . . .	X	Y	Z
A	A	B	C	D	. . . . .	X	Y	Z
B	B	C	D	E	. . . . .	Y	Z	A
C	C	D	E	G	. . . . .	Z	A	B
.	.	.	.	.		.	.	.
.	.	.	.	.		.	.	.
.	.	.	.	.		.	.	.
X	X	Y	Z	A	. . . . .	U	V	W
Y	Y	Z	A	B	. . . . .	V	W	X
Z	Z	A	B	C	. . . . .	W	X	Y

Буквы открытого текста A, B, C, …, Z.

Рис. 2. Таблица Вижинера относительно английского алфавита

Легко видеть, что буква открытого текста заменяется на букву шифртекста одним из 26 способов, в зависимости от соответствующей буквы ключевого слова. Периодичность выбора этих способов является слабостью метода. Поэтому Вижинер предложил в качестве текущей ключевой буквы брать последнюю букву шифртекста (первые несколько букв надо оговаривать заранее). Однако при такой модернизации допущенная ошибка распространяется на весь последующий текст. Обозначив буквы алфавитов целыми числами от 0 до 25, описанное преобразование зашифрования открытого текста $[m_i]$ и переобразование расшифрования шифрованного текста $[c_i]$ могут быть представлены в виде соотношений

$c_i = m_i + k_i \pmod{26}$ и $m_i = c_i - k_i \pmod{26}$ для всех i.

(Для однообразия можно использовать похожие преобразования вида

$c_i = k_i - m_i \pmod{26}$ и $m_i = k_i - c_i \pmod{26}$)

XVII в. Кардинал Ришелье (министр при короле Франции Людовике XIII) создал первую в мире шифрслужбу. Эту службу возглавлял Антуан Россиньоль (1590–1673).

Лорд Френсис Бэкон (1562–1626) был первым, кто обозначил буквы 5-значным двоичным кодом: A= 00001. B = 00010, ... и т.д. Правда, Бэкон никак не обрабатывал этот код, поэтому такое закрытие было совсем нестойким. Тут уместно вспомнить коды Морзе, Бодо, международный телеграфный код № 2, код **ASCII**, также представляющие собой простую замену.

В этом же веке были изобретены так называемые **словарные шифры.** При шифровании буквы открытого текста обозначались двумя числами – номером строки и номером буквы в строке на определенной странице какой-нибудь выбранной распространенной книги. Эта система является довольно стойкой, но книга может попасть в руки противника.

В конце XVIII в. в переписке французской метрополии с колониями стали применяться в основном трехзначные **коды** на несколько сот кодвеличин. Обычно при шифровании пользуются кодкнигой, где для

удобства все кодвеличины стоят в алфавитном порядке. Если при кодировании нужного слова не окажется среди кодвеличин, то оно кодируется побуквенно. Код имеет только один ключ – долговременный – содержание кодкниги. Главный недостаток такого шифрования – ограниченная стойкость, особенно при длительной и интенсивной переписке. Криптоаналитики противника обычно предполагают состав кодвеличин, а кодобозначения они могут узнать из перехвата шифрпереписки. Остается только правильно привязать их друг к другу. Для этого анализируются действия применяющего коды, сопоставляются даты, названия населенных пунктов, имен и т.п., чтобы строить гипотезы о соответствии.

К. Гаусс (1777–1855) – великий математик, он тоже не обошел своим вниманием криптологию. Он создал шифр, который ошибочно считал нераскрываемым. При его создании использовался интересный прием – **рандомизация** (random – случайный) открытого текста. Открытый текст можно преобразовать в другой текст, содержащий символы большего алфавита, путем замены часто встречающихся букв случайными символами из соответствующих определенных им групп. В получающемся тексте все символы большого алфавита встречаются с примерно одинаковой частотой. Зашифрование такого текста противостоит методам дешифрования на основе анализа частот отдельных символов. После расшифрования законный получатель легко снимает рандомизацию. Такие шифры называют «шифрами с многократной подстановкой» или «равночастотными шифрами».

Итог многовекового противостояния разработчика шифра – криптографа и его оппонента – криптоаналитика, дешифровальщика подвел голландец Керкхоффс (Kerckhoffs, 1835–1903), который сформулировал правила этого противостояния. Основное **правило Керкхоффса** состоит в том, что при разработке и применении шифра надо исходить из того, что весь механизм шифрования, множество используемых правил или алгоритмов, рано или поздно становятся известными криптоаналитику, а стойкость шифра должна определяться только секретностью ключа [6, 40].

Середина XIX в. Изобретение телеграфа и других технических видов связи дало новый толчок развитию криптологии. Информация передается в виде токовых и бестоковых посылок, т.е. представляется в двоичном виде. Поэтому возникла проблема сжатия информации, которая решалась опять же с помощью кодов, чтобы одно слово или даже целую фразу можно было передать двумя-тремя знаками.

Передача шифрсообщений по таким линиям связи, как телеграф и особенно радио, дала криптоаналитику относительно легкую возможность получать передаваемые шифртексты. Вообще, можно предложить **классификацию методов дешифрования** по уровню доступной для криптоаналитика информации.

1. Методы дешифрования на основе знания только шифртекстов.

2. Методы дешифрования при известном открытом и соответствующем шифрованном текстах. Такая ситуация вполне реальна, так как иногда приходится шифровать общеизвестные данные, например, дипломатические документы, секретные только до их опубликования. Можно также предполагать наличие в открытом тексте вероятных слов и выражений.

3. Методы дешифрования по выбранным открытым и соответствующим шифртекстам (chosen plaintext attack). Такая ситуация возникает, когда криптоаналитик имеет доступ к шифраппарату или шифрующему преобразованию и может получить шифртекст для любого выбранного им открытого текста (например, в криптосистемах с открытым ключом).

4. Методы дешифрования по выбранным шифртекстам и соответствующим открытым текстам (chosen cipher text attack). Такая ситуация возможна, когда криптоаналитик имеет доступ к преобразованию расшифрования (как к черному ящику) и может получать для выбранных шифртекстов соответствующие открытые тексты.

Продолжим исторический экскурс и проиллюстрируем сделанное отступление о методах дешифрования примерами. Так, во времена Первой мировой войны некоторые военачальники, не понимая тонкостей

криптографии, требовали от шифровальщиков повторных передач своих донесений или приказов, а иногда из-за низкого уровня связи эта же информация передавалась еще и в открытом виде. Это приводило к вскрытиям применяемых систем из-за их нестойкости к методам по открытому и соответствующему шифрованному текстам. Вообще, история изобилует примерами превосходства криптоаналитиков над криптографами.

Вместе с тем, имеются примеры и другого рода. В 1917 г. инженер американской компании AT&T Г.С. Вернам опубликовал замечательную систему побитового шифрования открытого текста, представленного в коде Бодо, когда каждый бит преобразуется с использованием соответствующего ему бита ключа по следующему алгоритму:

$$1 \oplus 1 = 0,\ 0 \oplus 1 = 1,\ 1 \oplus 0 = 1,\ 1 \oplus 1 = 0.$$

Это так называемое сложение по модулю два («XOR»). Расшифрование осуществляется той же операцией, что очень удобно для реализации. Вернам предлагал (точнее, после совета Моборна) использовать ключ только один раз (one - time pad), несмотря на трудности передачи по секретному каналу этого ключа, длина которого равна длине шифруемого открытого текста. Однако это дает, как показал впоследствии К. Шеннон, действительно нераскрываемый шифр. Сам Вернам, хотя и считал, что его шифр нераскрываем, не представил доказательств этого. Казалось бы, что после создания такого шифра все проблемы для криптографов решены. Но это не так. Как показала дальнейшая практика, использование шифра Вернама требует решения проблем выработки длинных двоичных последовательностей ключей, контроля за их качеством, проблем хранения, уничтожения и транспортировки. На каждом из этапов существования ключей (жизненного цикла) есть угроза их безопасности. Все это делает систему Вернама непрактичной, дорогостоящей, и она применяется в исключительных случаях.

Все приведенные выше шифры можно назвать **ручными шифрами** (Paper-and-Pencil Cipher). Такие шифры разрабатываются и используются до сих пор.

В начале XX в. были созданы **механические шифрмашины**, вырабатывающие шифр с помощью набора колес, которые, находясь на одной оси, дискретно перемещались одно относительно другого, создавая на каждом такте уникальное сочетание из всех возможных сочетаний угловых положений. Первые такие машины были сконструированы на основе принципов, заложенных в кассовые аппараты, арифмометры, торговые автоматы и т.п. Все эти машины реализовывали шифр замены. Такой же принцип сохранился в электрических машинах, получивших название **дисковых (роторных).** Колеса этих машин (диски) изготавливались из электроизолирующего материала и имели вид узкого цилиндра, в оба основания которого были запрессованы латунные контакты, соответствующие буквам алфавита.

При шифровании на один из контактов первого диска подается напряжение, которое последовательно передается на связанный контакт последнего диска, соответствующий некоторой букве шифртекста. При большом количестве дисков (5–10) и правильно выбранном законе псевдослучайного движения дисков система обеспечивала весьма высокую стойкость.

В книге [40] описаны преобразования этого шифра в виде подстановок. Если обозначить через T подстановку (i → i+1), то после поворота на 1 угловое положение один диск реализует подстановку вида

T^{-1} X T = (i →(i-1)) ((i-1) →(x_{i-1}) (x_(i-1) → (x_(i-1) → (x_(i-1)+1)) = (i → (x_(i-1)+1)).

Это легко видеть из рисунков.

А) положение диска до поворота

Розетка		Диск				Розетка
0	→	0		X_1	→	X_1
1	→	1			→	*
2	→	2		X_0	→	X_0
*		*		*		*
*		*		*		*
n - 1	→	n - 1		*	→	*

Б) после поворота на одно угловое положение

Розетка		Диск				Розетка
0	→	n - 1		*	→	
1	→	0		X_1	→	X_1 +1
2	→	1		*		
*		*		X_0	→	X_0+1
*		*		*		*
n - 1	→	n-2		*	→	*

Рис. 3. Положение диска роторной машины до и после поворота

После поворота на m угловых положений будет реализована подстановка T^{-m} X T^{m} . Если в начальный момент N дисков реализуют подстановки X_1, … , X_N, то при повороте их соответственно на k_1, …, k_N положений будет реализована подстановка

$$TX_1T^{k_1-k_2}X_2...T^{k_{N-1}-k_N}T^{k_N}$$

На примере дисковой машины можно показать типы ключей. Долговременным ключом системы, меняющимся весьма редко, является комплект подвижных дисков. Обычно количество таких дисков в наборе превышает количество дисков, устанавливаемых в машину для шифрования. Суточным ключом является выбор дисков, устанавливаемых в машину на

текущий день из всего комплекта дисков, и порядок установки их в машину. И, наконец, для зашифрования каждого отдельного сообщения применялся сеансовый (разовый) ключ, которым в данном случае является начальное угловое положение колес.

Принцип работы дисков был почти одновременно открыт четырьмя изобретателями из разных стран. Это американец Э. Хеберн (1918 г.), голландец Х. Кох (1919 г.), швед А. Дамм (1919 г.), немец А. Шербиус (1927 г.). Последний сконструировал известную немецкую дисковую машину «Энигма» (Загадка). Из четырех изобретателей только А. Дамму удалось добиться коммерческого успеха. Его фирма «Криптография», под управлением Бориса Хагелина, выпустила весьма компактную и простую в работе шифрмашину, известную под названием «Хагелин». Шифрмашины этой фирмы и их модификации были изготовлены в огромном количестве. Так, только США в период Второй мировой войны заказали несколько тысяч таких машин под наименованием «Конвертер М-209». Далее после войны Б. Хагелин перенес штабквартиру фирмы в Швейцарию, где она функционирует до сих пор в г. Цуг, правда под другим названием – «Crypto AG».

История криптологии исключительно интересна. Она насыщена многочисленными фактами противоборства криптографов и криптоаналитиков (дешифровальщиков). Причем, чем дальше от наших дней рассматриваемый отрезок времени, тем больше этих фактов опубликовано и известно. Такова деликатная природа этой науки. Для более детального ознакомления с историей криптологии можно порекомендовать книги [2, 15, 34, 40] и др.

В настоящее время вместо понятия шифра часто используется понятие криптографической системы с секретным ключом (secret key cryptosystem), [6, 12], которая задается следующими пятью компонентами:

пространством открытых текстов, M;

пространством шифрованных текстов, C,

пространством ключей, K;

множеством преобразований зашифрования $\{E_k, k \in K\}$;

E_k : м $\rightarrow$ с, где $k \in k$;

множеством преобразований расшифрования $\{D_k, k \in K\}$;

D_k : с $\rightarrow$ м, где $k \in k$;

Преобразования D_k и E_k для всех $k \in K$ и любого открытого текста $m \in M$ должны удовлетворять следующему условию:

$$D_k\left[E_k(m)\right] = m.$$

Согласно основному правилу Керкхоффса, множества преобразований $\{E_k; k \in K\}$ и $\{D_k; k \in K\}$ могут быть известны не только законному пользователю шифра, но и криптоаналитику. Секретность же открытого текста обеспечивается сокрытием того, какое именно преобразование из известного множества преобразований использовалось для зашифрования. Заметим также, что знание ключа k дает возможность легко указать соответствующие ему преобразования E_k и D_k, однако обратное не всегда верно.

Криптосистемы с секретным ключом подразделяются на два вида: блочные (block) и поточные (stream) криптосистемы.

Блочная криптосистема (**блочный шифр**) разбивает открытый текст M на последовательные блоки M_1, M_2 .... и зашифровывает каждый блок с помощью одного и того же обратимого преобразования E_k, выбранного в соответствии с ключом k. Например, таким образом

$$E_k(M) = E_k\left(M_1\right), E_k\left(M_2\right) \ldots .$$

Но, как легко видеть, при таком методе применения преобразования E_k необходимо обеспечить целостность передаваемого шифрованного текста. Иногда преобразование E_k называют базовым блочным преобразованием или алгоритмом.

Примерами блочных систем являются: DES (режим электронной кодовой книги – ECB), ГОСТ 28147-89 (режим простой замены [20,76]. Размер блоков открытого и шифрованного текста в этих криптосистемах равен 64 битам. В американском стандарте шифрования AES размер блока равен 128. В настоящее время такой размер блоков более актуален.

ГОСТ 28147-89

Рассмотрим подробнее отечественный стандарт **ГОСТ 28147-89**, используемый с 1989 года до настоящего времени.

Всего у стандарта имеется четыре режима: режим простой замены; режим гаммирования; режим гаммирования с обратной связью; режим имитовставки.

Начнем изложение с **режима простой замены**. Описываемый режим работы соответствует режиму электронной кодовой книги ECB (Electronic Codebook) для алгоритма DES по стандарту FIPS PUB 81 (I980). Этот режим используется при построении других режимов и иногда называется **базовым алгоритмом блочного шифрования.**

Режим простой замены российского ГОСТа имеет много общего с американским стандартом шифрования DES, а именно совпадает принцип их построения в соответствии со схемой Фейстеля (Feistel H.), обеспечивающей обратимость преобразования зашифрования. (Другой принцип, SP-сети, использован в ряде других блочных шифров – RC2, AES.)

Введем необходимые понятия и обозначения.

Ключом (сеансовым, разовым) является 256-битная последовательность W = [W_1, …, W_{256}], которая разбивается на восемь 32-битных блоков следующим образом:

$$X_0 = \left[W_{32},....,W_1\right],\ X_1 = \left[W_{64},....,W_{33}\right],\ \dots\ ,\ X_7 = \left[W_{256},....,W_{225}\right].$$

Эти блоки используются в **процедуре размножения ключа** (key schedule – в литературе) для получения тридцати двух 32-битных **цикловых ключей** $y_0,....,y_{31}$ по правилу

$$\left[y_0,....y_{31}\right]=\left[x_0,....,x_7,.x_0,....x_7,.x_0,....,x_7,.x_7,....,x_0\right].$$

Долговременным ключом К является набор из восьми таблиц замены $K_1,....,K_8$ (S-boxes – в литературе) 4-битных двоичных векторов, которые разбиваются на восемь последовательно идущих 4-битных векторов, каждый из которых преобразуется в 4-битный вектор по соответствующей таблице замены из $K_1,....,K_8$ (через Т обозначим преобразование циклического сдвига на 11 разрядов 32-битных двоичных векторов в сторону старшего разряда).

Каждый 64-битный блок открытого текста M_i (i=1,2,…) разбивается на две равные части $M_1=\left[a_1(i),....,a_{32}(i),.b_1(i),....,b_{32}(i)\right]$, которые определяют начальные векторы u_0 и u_1 вида

$$u_0=\left[u_0^{32},....,u_0^1\right]=\left[b_{32}(i),....,b_1(i)\right],\ u_1=\left[u_1^{32},....,u_1^1\right]=\left[a_{32}(i),....,a_1(i)\right],$$

для рекурренты второго порядка

$$u_{j+2}=u_j\oplus TK\left[u_{j+1}+u_j\right],\ \text{j} = 0,1,\ ...\ 31.$$

Через «⊕» и «+» обозначены соответственно операции побитового сложения двоичных векторов по модулю два и сложения целых чисел по модулю 2^{32}. Здесь и далее не различаем целые числа и их двоичное представление, все определяется операциями, которые к ним применяются.

В приведенных обозначениях 64-битный блок шифртекста C_i получается из значений $u_{32}=\left[u_{32}^{32},....,u_{321}^1\right]$ и $u_{33}=\left[u_{33}^{32},....,u_{33}^1\right]$, описанной выше рекурренты, следующей перестановкой бит

$$c_1=\left[u_{33}^1,....,u_{33}^{32},.u_{32}^1,....,u_{32}^{32}\right].$$

Расшифрование производится в обратном порядке.

В стандарте указано, что этот режим используется только для выработки и зашифрования таблиц замены $K_1,....,K_8$ при их передаче по каналам связи с обеспечением имитозащиты. Также он используется как базовый блочный шифр для построения остальных 3-х режимов, которые опишем позже. Менее известен отечественный стандарт ГОСТ Р ИСО/МЭК 10116-93 «Режимы работы для алгоритма n-разрядного блочного шифрования», который фактически повторяет 4 режима алгоритма DES (ECB, CBC, CFB, OFB), описанных в FIPS PUB 81 (I980). В настоящее время они немного изменены и описаны в спецификации SP 800-38A. Всего в этом документе описаны 5 режимов, помимо названных добавлен еще один режим – CTR – the Counter Mode. Все 5 режимов служат для обеспечения секретности. Для обеспечения аутентификации было разработано много режимов, но NIST принял режимы, описанные в SP 800-38B. Наконец режимы обеспечения шифрования с аутентификацией (authenticated encryption mode), принятые NIST, можно найти в SP 800-38C и SP 800-38D.

Как уже было сказано, аналогом режима простой замены является режим ECB для DES. Здесь мы не будем приводить описание DES, так как оно хорошо известно, и к тому же сейчас не рекомендуется для использования (стойким и используемым на практике является тройной DES).

Описание тройного DES алгоритма (3DES, Triple DES, TDES)

В американских стандартах различают понятия алгоритма и стандарта. На слуху все же больше звучит «стандарт DES», чем «алгоритм DEA». После отказа от DES и до принятия стандарта AES использовался тройной DES – 3DES. Обычно производители средств защиты указывают на реализацию 3DES, не уточняя деталей, которые весьма существенны. Далее описание дается по **FIPS PUB 46-3**.

Пусть ***EK(I)*** и ***DK(I)*** обозначают соответственно преобразование зашифрования и расшифрования блока ***I*** алгоритмом DES на ключе ***K***. Преобразование зашифрования/расшифрования по алгоритму TDEA (как

определено в ANSI X9.52) является композицией из преобразований зашифрования/расшифрования по алгоритму DEA следующим образом.

1. Преобразование зашифрования по алгоритму TDEA: преобразование 64-битового блока ***M*** в 64-битовый блок ***C*** следующим образом:

$$C = EK_3(DK_2(EK_1(M))).$$

2. Преобразование расшифрования по алгоритму TDEA: преобразование 64-битового блока ***M*** в 64-битовый блок ***C*** следующим образом:

$$C = D\,K_1\,(EK_2(DK_3(M)))$$

Стандарт определяет следующие возможности для выбора ключей (K_1, K_2, K_3)

1. Выбор 1: K_1, K_2 и K_3 являются независимыми ключами;
2. Выбор 2: K_1 и K_2 – независимые ключи и $K_3 = K_1$;
3. Выбор 3: $K_1 = K_2 = K_3$.

Когда используется Выбор 3 ключей, то режимы шифрования для TECB, TCBC, TCFB и TOFB совпадают с режимами ECB, CBC, CFB, OFB обычного стандарта DES соответственно.

Заметим, что желание ускорить процесс шифрования и использование двойного DES не приводит к увеличению стойкости, так как из соотношения $C = E\,K_2\,(E\,K_1\,(M))$ следует соотношение $D\,K_2\,(C) = E\,K_1\,(M)$, которое можно использовать в методе согласования при известном открытом и шифрованном текстах.

Режим сцепления блоков шифра (Cipher Block Chaining – CBC).

Если обозначить через E_k (D_k), преобразование зашифрования (расшифрования), то получение блоков шифрованного текста C_1, C_2 … из блоков M_1, M_2, … открытого текста описывается соотношением

$$C_1 = E_k\left[C_{i-1} \oplus M_1\right], \text{ i=1,2,…}$$

Начальное значение C_{-1}, так называемый вектор инициализации – IV, является несекретным и передается вместе с шифртекстом. Но к нему предъявляется требование быть непредсказуемым для противника.

Процесс расшифрования легко выводится из предыдущих соотношений применением к обеим частям равенства преобразования D_k и определяется соотношением $M_1 = D_k\left[C_1\right] \oplus C_{i-1}$, i=1,2, … .

Иногда используются подобные CBC режимы, например режим PCBC (Propagating Cipher Block Chaning) и режим PBC (Plaintext Block Chaning), соотношения зашифрования и расшифрования для которых имеют соответственно вид

$$PCBC: \begin{cases} c_i = E_k(M_i \oplus C_{i-1} \oplus M_{i-1}), \\ M_i = C_i \oplus M_{i-1} \oplus D_k(C_i) \end{cases} \quad PBC: \begin{cases} C_i = E_k(M_i \oplus M_{i-1}), \\ M_i = M_{i-1} \oplus D_k(C_i). \end{cases}$$

Режим PCBC используется, например, в системе **Kerberos** [48].

Поточная криптосистема (поточный шифр) разбивает открытый текст M на порции, буквы (знаки) или биты $m_1, m_2, \ldots$ и зашифровывает каждый знак m_1 с помощью обратимого преобразования E_k, выбранного в соответствии со знаком k_1 **ключевого потока (key stream)** $k_1, k_2, \ldots$.

В отечественной литературе (см. например, ГОСТ 28147-89) такие системы называются **системами гаммирования,** а последовательность $k_1, k_2, \ldots$ называется **гаммой.**

Примерами поточных криптосистем являются:

- система Вернама (где $E_k(m_1) = m_1$ XOR $k_1, m_1\, k_1, \in \overline{0,1}$),

- ГОСТ 28147-89 (режимы гаммирования и гаммирования с обратной связью),

- DES (режимы CFB и OFB).

В свою очередь поточные криптосистемы подразделяются на **синхронные поточные системы** (synchronous stream cipher) и **самосинхронизирующиеся поточные системы** (self synchronous stream cipher). Первые отличаются тем, что ключевой поток в них получается независимо от открытого и шифрованного текстов.

Алгоритм, который вырабатывает ключевой поток (гамму), должен быть либо детерминистическим, чтобы можно было воспроизвести одинаковые

потоки на приемном и передающем концах связи, либо случайным, что часто непрактично в силу необходимости передачи и хранения ключевой информации большого объема. Этот алгоритм называют **генератором ключевого потока (**running key generator-RKG, key stream generator**).** Если генератор детерминистический, то он должен зависеть от секретного ключа.

Для описания работы генераторов удобно использовать язык **теории автоматов** [27, 40, 52]. Если обозначить через S_i состояние генератора в i-тый момент времени, то работу синхронной поточной системы можно описать соотношениями

$$\begin{cases} S_{i+1} = F(k, S_i) \\ k_i = f(k, S_i) \end{cases},$$

где функции F и f называются соответственно **функцией переходов** состояний и **функцией выхода.** Начальное состояние S_0 может быть функцией от ключа k.

Одна из основных сложностей в построении таких систем заключается в построении **безопасных генераторов ключей (случайных или псевдослучайных последовательностей)**, удовлетворяющих ряду требований. Эти требования можно найти в литературе. Важно, помимо других требований, тебование непредсказуемости, чтобы нельзя было по имеющемуся отрезку или части членов последовательности восстановить недостающие члены. Рекомендации по построению хороших генераторов содержатся в ряде американских и международных стандартов. Часто при построении генераторов ключей используют преобразования блочных шифров.

Российских стандартов для безопасных генераторов нет.

Режим обратной связи по входу (Output Feedback – OFB).

Этот режим характеризует независимость функции выхода *f* от ключа. Поэтому его также называют режимом с внутренней обратной связью (Internal Feedback).

Если обозначить через E_k – преобразование зашифрования блочного шифра на ключе k, то уравнение переходов в этом режиме принимает вид $S_{i+1} = E_k[S_i]$, а в качестве знаков k_i ключевого потока могут использоваться какие-нибудь (чаще всего старшие) биты вектора состояния S_i (Режим OFB для DES). В спецификации SP 800-38A используются все биты, а начальное состояние – IV – вектор инициализации, не обязан быть непредсказуемым, но обязан быть уникальным (например, номером сообщения).

Иногда рассматривают вариант режима OFB, где ключ k определяет только начальное состояние генератора ключевого потока S_0 и $S_{i+1} = F[S_i]$, $k = f[S_i]$.

Примером синхронного поточного шифра является **режим гаммирования** в ГОСТ 28147-89.

Состояния счетчика (Y_i, Z_i) изменяются по правилу

$(Y_i, Z_i) = (Y_{i-1} + N_1, Z_{i-1} + N_2)$ для всех i=1, …, n.

Константы N_1 и N_2 фиксированы и прибавляются соответственно по модулю 2^{32} и $(2^{32}-1)$. Начальное состояние счетчика $(Y_0, Z_0) = E_w(S)$ получается из 64-битного вектора S, называемого синхропосылкой, с помощью его шифрования в режиме простой замены на ключе w. Число n равно числу 64-битовых блоков открытого текста.

Блоки C_i шифртекста получаются из блоков M_i открытого текста по правилу $C_i = M_i$ XOR $E_w((Y_i, Z_i))$ для всех i=1, …, n. Синхропосылка передается вместе с блоками C_i. Никаких требований к выработке синхропосылки в стандарте не установлено.

Счетчиковый метод (CTR - the Counter mode).

Метод предложен Диффи и Хеллманом [3, 10]. В общем случае работа генератора ключей описывается соотношениями

$$\begin{cases} S_{i+1} = F(S_i) \\ k_i = f(k, S_i) \end{cases}.$$

Функция переходов F не зависит от ключа, но преобразование F выбирается так, что гарантирует перебор всех или почти всех возможных состояний генератора при изменении времени i = 0,1, ... В качестве F часто используются преобразования регистров сдвига с линейной обратной связью, вырабатывающие последовательности максимального периода [6, 17, 29], или просто счетчики, состояние которых меняется прибавлением единицы по некоторому модулю (периоду состояний счетчика), как в SP 800-38A. В качестве функции выхода используется преобразование зашифрования E_k блочного шифра.

Достоинство счетчикового метода в том, что можно получить произвольный знак k_i ключевой последовательности без получения предыдущих значений этой последовательности (в отличие от режима OFB, например), если известно состояние счетчика.

Самосинхронизирующиеся поточные системы характеризуются тем, что каждый знак ключевого потока в любой момент времени определяется фиксированным числом предшествующих знаков шифртекста, то есть они описываются соотношениями

$$S_i = F\left[C_{i-1}, C_{i-2}, \ldots, C_{i-n}\right]$$

$$k_i = f[k, S_i], \; i = 1,2, \ldots .$$

Заметим, что в этих системах вход (знаки шифртекста) и выход (ключевой поток) могут быть известны криптоаналитику как при атаке с известным открытым текстом (known plaintext attack).

Примером таких поточных систем является режим гаммирования с обратной связью в ГОСТ 28147-89 и режим CFB для DES.

Режим обратной связи по шифртексту (Cipher Feedback – CFB) для DES.

Приведенные выше соотношения при использовании блочного шифра и проходного регистра сдвига принимают следующий вид:

$S_i = [C_{i-1}]$, $i = 1, \ldots n$.

$k_i = E_k [S_i]$, i = 1, … n. C_0 = IV – вектор инициализации.

Тогда блоки C_i шифртекста получаются из блоков M_i открытого текста по правилу C_i XOR M_i.

В **режиме гаммирования с обратной связью в ГОСТ 28147-89** блоки C_i шифртекста получаются из блоков M_i открытого текста по правилу

$C_1 = M_1$ xor E_k (S), $C_i = M_i$ xor $E_k [C_{i-1}]$, для i= 2, …, n.

Здесь S – 64-битовая синхропосылка, передаваемая вместе с блоками шифртекста $[C_i]$.

Самосинхронизирующиеся поточные системы хороши тем, что как только на приемном конце принят правильный неискаженный отрезок шифртекста длины n, то значение соответствующего знака ключевого потока также будет определено правильно.

Блочные и поточные криптосистемы имеют ряд достоинств и недостатков друг перед другом, некоторые из которых приведены в работах [12, 27].

Американский стандарт шифрования AES.

В конце ноября 2001 г. NIST принял новый стандарт симметричного шифрования Advanced Encryption Standard (AES) (**FIPS PUB 197**). Этим завершилась процедура выбора нового перспективного стандарта шифрования на смену устаревшего стандарта DES (DES-1977), начатая в сентябре 1997 г. после опубликования требований к участникам конкурса.

Алгоритм стандарта DES считается устаревшим по следующим причинам: малая длина ключа в 56 бит, неудобство реализации на современных процессорах, относительно малое быстродействие, относительно малый размер блока шифрования в 64 бита. Он имеет лишь одно достоинство – стойкость. За прошедшие годы интенсивного криптоанализа не было найдено методов вскрытия этого алгоритма, существенно отличающихся по эффективности от полного перебора всех ключей.

Минимальные требования к новому алгоритму шифрования были следующие: алгоритм должен быть блочным шифром с секретным ключом; алгоритм должен поддерживать следующие комбинации пар размеров ключа и блока шифрования в битах – **128-128, 192-128, 256-128.** Возможна поддержка и других дополнительных комбинаций, помимо перечисленных.

Кроме того, предполагалось, что алгоритм должен быть открыто опубликован; удобен как для аппаратной, так и для программной реализации; не должен быть запатентован.

Алгоритмы, участвующие в конкурсе, предполагалось сравнивать по следующим характеристикам в порядке убывания значимости: стойкости, стоимости, гибкости.

Стойкость – это самый важный критерий в оценке алгоритма. Оценивались способность шифра противостоять различным методам криптоанализа, а также его статистические характеристики. Кроме того, учитывалась стойкость к атаке методом полного перебора с учетом прогнозируемого роста мощности вычислительной техники.

Стоимость – не менее важный критерий, учитывая одну из основных целей NIST – широкая область использования и доступность AES. Стоимость зависит от простоты реализации и вычислительной эффективности (в первую очередь от быстродействия) на различных платформах в сочетании с низкими требованиями к используемой памяти.

Гибкость – включает способность алгоритма использовать ключи различного размера, превышающего установленный минимум (128 бит), надежность и эффективность реализации в разных средах, возможность создания на базе этого алгоритма других криптографических примитивов, например поточного шифра, генератора псевдослучайных чисел, функции хэширования и др. Другими словами, AES должен быть существенно более эффективным с точки зрения практической реализации (в первую очередь скорости шифрования и формирования ключей), чем TripleDES, при этом не уступая ему в стойкости.

Всего к конкурсу AES были допущены 15 алгоритмов, разработанных криптографами из 12 стран. В финал конкурса вышли следующие алгоритмы: MARS (США), RC6 (США), RIJNDAEL (Бельгия), SERPENT (Великобритания, Израиль, Норвегия), TWOFISH (США).

Победителем конкурса стал криптоалгоритм RIJNDAEL.

Важно заметить, что алгоритм RIJNDAEL допускает различные размеры пар (блока – ключа) в битах, но в стандарте FIPS PUB 197 приняты только следующие: **128-128, 192-128, 256-128.** Далее описание будет в основном излагаться в терминах RIJNDAEL.

Для понимания работы алгоритма RIJNDAEL напомним некоторые математические основы. В RIJNDAEL операции выполняются над байтами, которые рассматриваются как элементы поля $GF(2^8)$. Элементами $GF(2^8)$ являются двоичные многочлены степени N < 8, которые могут быть заданы строкой своих коэффициентов. Так, например, байту 01010111 (‘57h’ в шестнадцатеричной форме) соответствует многочлен $x^6 + x^4 + x^2 + x + 1.$

Умножение в поле $GF(2^8)$ может быть реализовано как операция умножения соответствующих многочленов с последующим приведением результата по модулю некоторого неприводимого двоичного многочлена $\varphi(x)$ восьмой степени. В RIJNDAEL для построения поля $GF(2^8)$ используют неприводимый многочлен показателя 51 - $\varphi(x) = x^8 + x^4 + x^3 + x + 1$.

В алгоритме RIJNDAEL авторы вводят операцию $xtime(\alpha)$. $xtime(\alpha)$ – это операция умножения элемента поля α на x. Тогда умножение на x^n можно осуществить путем n-кратного повторения операции $xtime(\alpha)$, а умножение на произвольный элемент поля можно осуществить, складывая результаты операций $xtime(\alpha)$.

Структура шифра

RIJNDAEL – это итерационный блочный шифр, имеющий следующие варьируемые параметры: длина блоков, длина ключа. Длина ключа и длина

блока могут быть равны независимо друг от друга 128, 192 или 256 битам (в AES это не так).

Промежуточные результаты преобразований, выполняемых в рамках алгоритма, называются ***состояниями.*** Состояние можно представить в виде прямоугольного массива байтов. Этот массив имеет 4 строки, а число столбцов равно длине блока, деленной на 32.

Ключ шифрования также представлен в виде прямоугольного массива с четырьмя строками. Число столбцов равно длине ключа, деленной на 32.

a_{00}	a_{01}	a_{02}	a_{03}	a_{04}	a_{05}
a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}

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

Таблица 1. Пример представления состояния ($N_b = 6$) и ключа шифрования

Входные данные для шифра обозначаются как байты состояния в порядке a_{00}, a_{10}, a_{20}, a_{30}, a_{01}, a_{11}, a_{21}, a_{31}, a_{41} ... После завершения действия шифра выходные данные получаются из байтов состояния в том же порядке.

Число циклов N_r зависит от значений N_b и N_k – числа 32-битных слов в информационном блоке и, соответственно, основном ключе алгоритма:

N_r	$N_b = 4$	$N_b = 6$	$N_b = 8$
$N_k = 4$	**10**	**12**	**14**
$N_k = 6$	**12**	**12**	**14**
$N_k = 8$	**14**	**14**	**14**

Таблица 2. Зависимость числа циклов от числа столбцов в таблице состояний и длины ключа шифрования

Цикловое преобразование

Любой цикл шифрования, кроме последнего, состоит из четырех различных преобразований. На псевдо-Си это выглядит следующим образом:

```
Round (State, RoundKey)
{
    ByteSub(State); // замена байтов
    ShiftRow(State); // сдвиг строк
    MixColumn(State); // перемешивание столбцов
    AddRoundKey(State, RoundKey); // доб. цикл. ключа
}
```

Последний цикл шифрования немного отличается от остальных. Вот как он выглядит:

```
FinalRound(State, RoundKey)
{
    ByteSub(State); // замена байтов
    ShiftRow(State); // сдвиг строк
    AddRoundKey(State, RoundKey); // доб. цикл. ключа
}
```

Каждое из приведенных преобразований определено далее.

Замена байтов (ByteSub)

Заметим, что в AES употребляются несколько видоизмененные названия для этого и других преобразований, что можно рассматривать как еще одно различие в описании (именно SubBytes, ShiftRows, MixColumns, …). Излагать будем как в описании Rijndael.

Преобразование ByteSub является S-блоком и представляет собой нелинейную замену байтов, выполняемую независимо для каждого байта состояния. Таблицы замены S-блока являются инвертируемыми и построены как композиция двух преобразований:

переход к обратному элементу относительно умножения в поле GF(2^8), при этом нулевой элемент '00h' переходит сам в себя;

применение аффинного преобразования над 8-мерным двоичным вектором:

$$\begin{vmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{vmatrix} \begin{vmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{vmatrix} + \begin{vmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{vmatrix}.$$

Применение описанного выше S-блока ко всем байтам состояния обозначено как ByteSub(State). Операция обратная к ByteSub – это замена байтов с использованием инвертированной таблицы. Обратимость операции ByteSub следует из обратимости аффинного преобразования и наличия у каждого элемента поля $GF(2^8)$ обратного элемента.

Преобразование сдвига строк (ShiftRow)

Строки матрицы состояния (кроме нулевой строки) циклически сдвигаются на различное число байт. Строка 1 сдвигается на C_1 байт, строка 2 – на C_2 байт и строка 3 - на C_3 байт. Значения сдвигов C_1, C_2 и C_3 зависят от длины блока . Их величины приведены ниже в таблице.

N_b	C_1	C_2	C_3
4	**1**	**2**	**3**
6	**1**	**2**	**3**
8	**1**	**3**	**4**

Таблица 3. Величина сдвига для разной длины блоков

Операция сдвига последних 3 строк состояния на определенную величину обозначена как ShiftRow (State).

Преобразование перемешивания столбцов (MixColumn)

В этом преобразовании столбцы состояния рассматриваются как многочлены над GF(2^8) и умножаются по модулю $M(x) = x^4 + 1$ на многочлен $c(x) = '03h' x^3 + '01h' x^2 + '01h' x + '02'.$

Многочлен $c(x)$ взаимно прост с многочленом $M(x)$, и, следовательно, операция умножения на $c(x)$ по модулю $M(x)$ обратима.

Это может быть представлено в матричном виде следующим образом:

$$\begin{vmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{vmatrix} = \begin{vmatrix} 02h & 03h & 01h & 01h \\ 01h & 02h & 03h & 01h \\ 01h & 01h & 02h & 03h \\ 03h & 01h & 01h & 02h \end{vmatrix} \begin{vmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{vmatrix}.$$

Применение этой операции ко всем четырем столбцам состояния обозначено как MixColumn(State).

Для обращения операции MixColumn необходимо умножить столбец, к которому применялась эта операция, на многочлен обратный к $c(x)$ по модулю $x^4 + 1$, т.е. на многочлен $d(x) = '0Bh' x^3 + '0Dh' x^2 + '09h' x + '0Eh'$ ($c(x) \otimes d(x) = 1 \bmod (x^4 + 1)$).

Добавление циклового ключа (AddRoundKey)

В данной операции цикловой ключ добавляется к матрице состояния посредством простого поразрядного XOR. Цикловой ключ вырабатывается из ключа шифрования посредством алгоритма выработки ключей (key schedule). Длина циклового ключа должна быть равна длине блока. Преобразование, состоящее в добавлении посредством XOR циклового ключа к матрице состояния, обозначено как AddRoundKey(State, RoundKey).

При добавлении ключа цикловой ключ складывается посредством операции *XOR* с матрицей состояния. Эта операция, очевидным образом, обратима.

Алгоритм выработки ключей (Key Schedule)

Цикловые ключи получаются из ключа шифрования с помощью алгоритма выработки ключей. Он состоит из двух частей:

- расширение ключа (Key Expansion),
- выбор циклового ключа (Round Key Selection).

Основные принципы алгоритма выглядят следующим образом:

- общее число бит цикловых ключей равно длине блока, умноженной на число циклов шифрования плюс 1 (например, для длины блока 128 бит и 10 циклов требуется 1408 бит циклового ключа);
- ключ шифрования расширяется в *расширенный ключ* (*Expanded Key*);
- цикловые ключи берутся из расширенного ключа следующим образом: первый цикловой ключ содержит первые Nb слов, второй – следующие Nb слов и т.д.

Расширение ключа (Key Expansion)

Расширенный ключ представляет собой линейный массив 4-байтовых слов и обозначен как $W[N_b(N_r+1)]$.

Первые N_k. слов содержат ключ шифрования. Все остальные слова определяются рекурсивно из слов с меньшими индексами. Алгоритм выработки ключей зависит от величины N_k.. Ниже приведена версия для N_k. меньшего или равного 6 и версия для Nk большего 6.

Для N_k.<=6 имеем:

```
KeyExpansion(CipherKey,W)
{
    for (i = 0; i < Nk; i++) W[i] = CipherKey[i];
    for (j = Nk; j < Nb*(Nk+1); j+=Nk)
    {
        W[j] = W[j-Nk] ^ SubByte( Rotl( W[j-1] ) ) ^
    Rcon[j/Nk];
```

```
        for (i = 1; i < Nk && i+j < Nb*(Nr+1); i++)
        W[i+j] = W[i+j-Nk] ^ W[i+j-1];
    }
}
```

Как можно заметить, первые N_k слов заполняются ключом шифрования. Каждое последующее слово $W[i]$ получается посредством сложения (XOR) предыдущего слова $W[i\text{-}1]$ и слова $W[i-N_k]$, находящегося на N_k позиций левее. Для слов, позиция которых кратна N_k, перед операцией XOR применяется преобразование к $W[i\text{-}1]$, а затем еще прибавляется цикловая константа. Преобразование содержит циклический сдвиг байтов в слове, обозначенный как Rotl, затем следует SubByte – замена байт, описанная выше.

Для $N_k > 6$ имеем:

```
KeyExpansion(CipherKey,W)
{
    for (i=0; i<Nk; i++) W[i]=CipherKey[i];
    for (j=Nk; j<Nb*(Nk+1); j+=Nk)
    {
        W[j] = W[j-Nk] ^ SubByte(Rotl(W[j-1])) ^
    Rcon[j/Nk];
        for (i=1; i<4; i++) W[i+j] = W[i+j-Nk] ^ W[i+j-
    1];
        W[j+4] = W[j+4-Nk] ^ SubByte(W[j+3]);
        for (i=5; i<Nk; i++) W[i+j] = W[i+j-Nk] ^ W[i+j-
    1];
    }
}
```

Отличие по сравнению с ранее рассмотренной схемой состоит в применении SubByte для каждого 4-го байта из N_k.

Цикловая константа не зависит от N_k. и определяется следующим образом:

Rcon[i] = (RC[i], '00' , '00' , '00'), где

RC[0]='01'

RC[i]=xtime(Rcon[i-1])

Выбор циклового ключа

i-й цикловой ключ получается из слов массива циклового ключа от $W[N_b * i]$ и до $W[N_b * (i+1)]$, как показано на Рис. 4 (белым цветом обозначено начальное заполнение; серым цветом обозначены ключи, получаемые только посредством вычислений внутри итерации Б; черным цветом обозначены ключи, к которым применяется дополнительное преобразование, выполняемое перед итерацией Б).

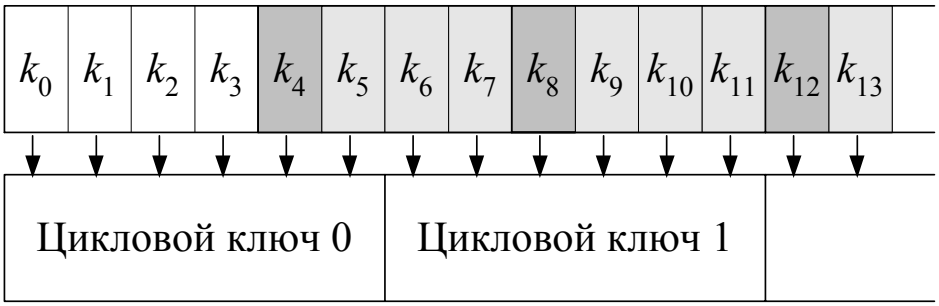

Рис. 4. **Расширение ключа и выбор циклового ключа**

Примечание. Алгоритм выработки ключей можно осуществлять и без использования массива $W[N_b(N_r+1)]$. Для реализаций, в которых существенно требование к занимаемой памяти, цикловые ключи могут вычисляться на лету посредством использования буфера длиной N_k слов.

Описание алгоритма

Для шифрования открытого текста выполняем следующие действия:

- разбиваем текст на блоки (текст в блок записывается по столбцам);
- шифруем каждый блок.

Шифрование блока состоит из следующих этапов: начального добавления циклового ключа; циклов шифрования; заключительного цикла шифрования.

На псевдо-Си это выглядит следующим образом:

```
Rijndael (State, CipherKey)
{
    KeyExpansion(CipherKey, ExpandedKey); // Расширение ключа
    AddRoundKey(State, ExpandedKey); // Добавление цикл. ключа
    For ( i=1 ; i<Nr ; i++) Round(State,ExpandedKey+Nb*i); // циклы
    FinalRound(State, ExpandedKey+Nb*Nr); // заключ. цикл
}
```

Если предварительно выполнена процедура расширения ключа, то RIJNDAEL будет выглядеть следующим образом:

```
Rijndael (State, CipherKey)
{
    AddRoundKey(State, ExpandedKey);
    For ( i=1 ; i<Nr ; i++) Round(State,ExpandedKey+Nb*i);
    FinalRound(State, ExpandedKey+Nb*Nr);
}
```

Расширенный ключ должен всегда получаться из ключа шифрования и никогда не указывается напрямую. На выбор ключа шифрования никаких ограничений не накладывается.

Для расшифрования текста необходимо разбить текст на блоки и к каждому блоку применить обратный алгоритм.

Расшифрование

При расшифровании порядок выполнения цикловых преобразований – обратный, следовательно, он имеет следующую структуру (ниже приведены только два цикла):

```
InvRound (State, RoundKey)
{
    AddRoundKey(State, RoundKey); // добавл. цикл. ключа
    InvMixColumn(State); // перемешивание столбцов
    InvShiftRow(State); // сдвиг строк
    InvByteSub(State); // замена байтов
}
InvFinalRound(State, RoundKey)
{
    AddRoundKey(State, RoundKey); // добавл. цикл. ключа
    InvShiftRow(State); // сдвиг строк
    InvByteSub(State); // замена байтов
}
```

Т.е. необходимо выполнить процедуру InvFinalRound, а затем определенное число раз процедуру InvRound.

Рассмотрим алгебраические свойства обратного преобразования шифра.

Во-первых, порядок операций InvShiftRow и InvByteSub безразличен, т.к. InvShiftRow переставляет байты, но не изменяет их значения, а InvByteSub работает со значениями байтов и не зависит от их позиций.

Во-вторых, последовательность операций:

```
AddRoundKey(State, RoundKey); // добавл. цикл. ключа
InvMixColumn(State); // перемешивание столбцов
```

может быть заменена последовательностью:

```
InvMixColumn(State); // перемешивание столбцов
AddRoundKey(State, InvRoundKey); // доб. цик. ключа,
```

где InvRoundKey получен применением InvMixColumn к соответствующему цикловому ключу.

Итак, используя свойства, описанные выше, инверсия двухциклового алгоритма будет следующей:

AddRoundKey(*State*, *ExpandedKey+2*Nb*); //
добавление циклового ключа

InvByteSub(*State*); // замена байтов

InvShiftRow(*State*); // сдвиг строк

InvMixColumn(*State*); // перемешивание столбцов

AddRoundKey(*State*, *I_ExpandedKey+Nb*); //
добавление циклового ключа

InvByteSub(*State*); // замена байтов

InvShiftRow(*State*); // сдвиг строк

AddRoundKey(*State*, *ExpandedKey*); // доб. цикл. ключа

Таким образом, получаем обратный шифр, имеющий ту же структуру, как и прямой шифр. Это может быть обобщено на произвольное число циклов. В результате, обратный шифр RIJNDAEL можно записать следующим образом:

```
I_Rijndael (State, CipherKey)
{
    I_KeyExpansion(CipherKey, I_ExpandedKey); //
    AddRoundKey(State, I_ExpandedKey+Nb*Nr); //
    For (i=Nr-1 ; i>0; i--) Round(State, I_ExpandedKey+Nb*i); //
    FinalRound(State, I_ExpandedKey); //
}
```

А процедуру расширения ключа можно записать так:

```
I_KeyExpansion(CipherKey, I_Expanded)
{
KeyExpansion(CipherKey, I_ExpandedKey);
for(i=1; I<Nr; i++)
InvMixColumn(I_ExpandedKey+Nb*i);
}
```

Следует отметить тот факт, что алгоритм расшифрования работает медленнее, чем алгоритм зашифрования. Это связано с тем, что в нем для

получения расширенного ключа на каждом шаге необходимо дополнительно применять достаточно медленную операцию MixColumn.

Новые режимы использования блочных шифров

Если проследить историю развития режимов шифрования, становится очевидной ее связь с развитием IT индустрии, в особенности с развитием аппаратной базы. Долгое время во всем мире возможность распараллелить вычисления была малодоступна. В основном использовались режим шифрования CBC и режим аутентификации сообщений – CBC-MAC. В качестве нового направления некоторые производители выбрали распараллеливание задач (процессоры с технологией HyperThreading), а другие стали более бурно развивать 64-битные платформы (Athlon64, Opteron). В настоящее время уже существуют режимы, ориентированные как на распараллеливание операций, так и режимы, которые быстро оперируют большими числами (PMAC/add).

Опишем основные свойства этих новых режимов и укажем на цели, которые преследовали их авторы, и проблемы, которые данные режимы пытаются решить.

Прежде всего, необходимо дать определение, что такое режим работы блочного шифра и какие в связи с этим понятия или термины нам придется ввести. Под **режимом работы блочного шифра** понимается алгоритм применения **базового алгоритма блочного шифра**, используемый в целях защиты информации (в том числе подсчет MAC, MDC, …). Как видно из определения, фактически сам блочный шифр (называемый теперь базовым алгоритмом блочного шифра) теперь является лишь частью другого алгоритма – алгоритма режима работы блочного шифра. Это обусловлено тем, что блочный шифр оперирует только с отдельным *блоком* данных, в то время как алгоритм *режима работы блочного шифра* имеет дело уже с целым *сообщением*, которое может состоять из некоторого числа n блоков. Более того, сообщение вообще не всегда можно разбить на *целое* число n

блоков. В этом случае в разных режимах шифрования приходится дополнять сообщение различным количеством бит.

Кроме задачи обеспечения конфиденциальности информации, обеспечение ее целостности также является важной задачей. Существуют два основных способа обеспечения целостности информации – вычисление MAC (message authentication code) и MDC (manipulation detection code). Эти режимы используются совместно с режимами шифрования.

Для повышения производительности в последнее время были созданы режимы, обеспечивающие как конфиденциальность, так и целостность информации (Authenticated Encryption Modes).

Режимы шифрования

Примеры уже были даны выше, например режимы – ECB, CBC, OFB, CFB, CTR. Каждый из этих режимов обладает своими достоинствами и недостатками: ECB – идеален для распараллеливания вычислений; с помощью него можно легко добиться больших скоростей шифрования. CBC более стойкий ко многим известным типам атак; OFB и CFB дают возможность работать с потоками информации, а не с блоками. Время выдвигает новые требования, и новые режимы шифрования пытаются их решить.

Режимы аутентификации

На практике уже давно используются такие режимы, как, например, режим CBC-MAC, аналогом которого является режим имитовставки российского ГОСТ 28147-89. Имитостойкость – это способность противостоять таким активным действиям нарушителя, как имитация и подмена. Термин «имитовставка» используется в ГОСТ как синоним кода аутентификации сообщения. Текст M разбивается на блоки по 64 бита $M_1 \| M_2 \| \ldots \| M_n$ и преобразуется по правилам

C_1 = E(16)_k (M_1), C_i = E(16)_k (C_(i-1) xor M_i), для i=1, 2…, n

Здесь E(16)_k обозначает преобразование зашифрования по ГОСТ 28147-89 на ключе k в режиме простой замены, но с 16 циклами вместо 32 для ускорения. Имитовставка – это часть бит блока C_n, как правило 32 младших.

Но подобные режимы имеют недостатки [23].

Приводимый далее режим явился прототипом принятого американского стандарта CMAC (Cipher-based MAC), точнее, он совпадает с OMAC1, который в свою очередь является модернизацией базового режима CBC-MAC и XCBC.

Режим OMAC: One-Key CBC MAC

Базовая конструкция семейства OMAC. Семейство OMAC определяется блочным шифром E: $K_E{}^x\{0,1\}^n \to \{0,1\}^n$, n-битной константой Cst, универсальной хэш-функцией H: $\{0,1\}^n{}^x X \to \{0,1\}^n$ и двумя явными константами Cst1, Cst2 $\in X$, где X – конечная область H.

H, Cst1 и Cst2 должны удовлетворять следующим условиям, пока Cst является произвольной. Будем писать $H_L(\cdot)$ для $H(L,\cdot)$

1. $\forall$ y$\in\{0,1\}^n$, число L$\in\{0,1\}^n$: H_L(Cst1)=y не более $\varepsilon_1 \cdot 2^n$ для достаточно малого ε_1.

2. $\forall$ y$\in\{0,1\}^n$, число$\in\{0,1\}^n$: H_L(Cst2)=y is не более $\varepsilon_2 \cdot 2^n$ для достаточно малого ε_2.

3. $\forall$ y$\in\{0,1\}^n$, число L$\in\{0,1\}^n$: H_L(Cst1)$\oplus$ H_L(Cst2)=y не более $\varepsilon_3 \cdot 2^n$ для достаточно малого ε_3.

4. $\forall$ y$\in\{0,1\}^n$, число L$\in\{0,1\}^n$: H_L(Cst1)$\oplus$ L=y не более $\varepsilon_4 \cdot 2^n$ для достаточно малого ε_4.

5. $\forall$ y$\in\{0,1\}^n$, число L$\in\{0,1\}^n$: H_L(Cst2)$\oplus$ L=y не более $\varepsilon_5 \cdot 2^n$ для достаточно малого ε_5.

6. $\forall$ y$\in\{0,1\}^n$, число L$\in\{0,1\}^n$: H_L(Cst1)$\oplus$ H_L(Cst2)$\oplus$ L=y не более $\varepsilon_6 \cdot 2^n$ для достаточно малого ε_6.

Заметим, что свойства 1 и 2 говорят, что $H_L(Cst1)$ и $H_L(Cst2)$ почти равномерно распределены. Свойство 3 удовлетворено AXU (almost XOR universal) хэш-функцией. Свайства 4, 5, 6 – новые вводимые требования.

Преимущества метода:

1. Минимальная длина ключа. Минимальная длина ключа в OMAC равна k-бит, тогда как в XCBC она равна (k+2n)-бит.
2. Произвольная длина сообщений. Область определения в OMAC равна {0,1} и $|M|$ не обязательно быть множеством блока длиной n.
3. Оптимальное количество применения блочного шифра. Для генерации тэга для любого непустого сообщения $M \in \{0,1\}$; OMAC требуется $\lceil |M|/n \rceil$ применений блочного шифра. Оптимальное количество выработки ключей для блочного шифра. OMAC требуется однократный запрос ключа.
4. Доказуемая стойкость. Авторы доказывают, что OMAC – псевдорандомизованная функция изменяемой вводной длины VIPRF с фиксированной исходящей длиной, из чего следует, что данный блочный шифр – псевдослучайная перестановка(PRP).
5. Не используется модуль расшифрования. Как и в любых других CBC MAC шифрах, OMAC не использует модуль расшифрования блочного шифра.
6. Простота. Благодаря простоте OMAC нетребователен ни к программным, ни к аппаратным средствам.

Также в марте 2003 г. после анализа OMAC T. Iwata и K. Kurosawa предложили новый вариант OMAC, названный OMAC1.

Режим CMAC, по рекомендации NIST

Подробное описание режима содержится в документе SP 800-38B. Как уже сказано, режим рекомендуется NIST для использования. CMAC назван по аналогии с HMAC (FIPS Pub 198, 2002).

Здесь приведем две основные процедуры (6.1) и (6.2), используемые для получения выходной строки T (MAC) из битовой строки M (сообщения) с использованием ключа K.

6.1. Генерация подключей K_1 и K_2 из ключа K.

1. $L = CIPH_K(0^b)$.(Здесь CIPH – преобразование зашифрования на рекомендованном NIST блочном шифре, b – размер входного блока шифра).

2. Если $MSB_1(L)=0$, то $K_1=L<<1$. Иначе $K_1=(L<<1)xor(R_b)$, R_b – постоянный вектор.

Здесь X<<1 есть $LSB_{Xlen}(X$ II $0)$, то есть младшие Xlen бит вектора, полученного сдвигом X на 1 знак влево и добавлением 0, Xlen – длина вектора X.

3. Если $MSB_1(K_1)=0$ то $K_2=K_1<<1$. Иначе $K_2=(K_1<<1)$ xor R_b.

6.2. Генерация кода аутентификации сообщения (MAC).

1. Получить по ключу K подключи K_1 и K_2, как в п. 6.1.

2. Если Mlen=0, то n=1, иначе n равен потолку числа Mlen/b.

3. $M= M_1 || M_2 || \dots M_{n-1} || M_n^*$

4. Если M_n^*- полный блок, то $M_n = K_1$ XOR $(M_n)^*$, иначе $M_n = K_2$ XOR $((M_n)^* ||0:o)$,где o = ти-Ьдут-1ю

5. $C_0= 0^b$.

6. Для i=1 до n , $C_i = CIPH_K(C_{i-1}$ xor $M_i)$.

7. Результирующий MAC равен $T=MSB_{Tlen}(C_n)$.

К выбору длины Tlen вектора T нужно подойти ответственно в случае, если ее выбрать меньше 64 (см. приложение A.2).

Режимы шифрования и аутентификации

При создании систем передачи информации очень часто встает задача обеспечить как конфиденциальность, так и целостность данных одновременно. Для этого можно применять два различных режима работы блочного шифра – один для шифрования, другой для проверки целостности, а можно воспользоваться специально разработанными режимами, которые сочетают в себе возможность шифрования данных с проверкой целостности

одновременно. Применение таких режимов оправдано для больших объемов передаваемых данных.

Режим Counter with CBC-MAC (CCM)

Данный режим описан в документе SP 800-38C, подготовленном NIST, и рекомендуется для использования (другой такой режим описан в SP 800-38D и называется GCM – Galois/Counter Mode)/.

Это фактически комбинация режима CTR и CBC-MAC.

Счетчик с режимом CBC-MAC (CCM) разработан, чтобы использовать блочный шифр AES (Advanced Encryption Standard) или любые другие блочные шифры с размером блока 128 битов и больше, для обеспечения аутентификации и шифрования, используя единственный ключ блочного шифра, установленный заранее.

CCM предназначен для использования в пакетной среде: ввод открытого текста включает заголовок, который заверен, но не зашифрован, и полезный груз, который заверен и зашифрован.

CCM оперирует целыми пакетами; он не поддерживает ни частичную, ни потоковую обработку данных. Пакет должен быть целым числом октетов. Каждый пакет задается как единственное значение, так называемое «случайное число». Размер случайного числа определяет максимум количества пакетов, которые могут быть заверены и зашифрованы одним ключом блочного шифра.

Обработка CCM увеличивает размер пакета, добавляя в конце зашифрованный опознавательный тэг. Успешная проверка опознавательного тэга гарантирует, что пакет преобразован из источника с использованием ключа блочного шифра. Следовательно, успешная проверка опознавательного тэга также гарантирует, что пакет не был изменен после генерирования тэга. Неудавшаяся проверка тэга выявляет как намеренные, неавторизованные изменения пакета, так и случайные изменения.

CCM допускает предвычисление ключевого потока, если известно значение случайного числа. Это позволяет вполовину снизить вычислительную нагрузку и увеличить эффективность работы.

В режиме CCM процессы зашифрования и расшифрования используют только модуль шифрования блочного шифра. В AES процессы зашифрования и расшифрования имеют некоторые существенные различия. Таким образом, использование только модуля шифрования может вести к существенному уменьшению размера кода и снижению требований к аппаратным средствам.

Спецификация CCM

CCM зашифрование может быть получено в итоге следующим образом. Во-первых, вычисляется CBC-MAC тэг T функции кодирования $\beta(N,H,M)$, где N – вводимое случайное число фиксированной битовой длины $k_n<k_b$, где k_b – битовая длина блока шифрования псевдослучайной функции E, на которой базируется CCM, H – заголовок, M – сообщение. Во-вторых, сообщение M зашифровывается в режиме CTR с блоками CTR, сгенерированными из случайного числа N с помощью π. И, наконец, тэг T зашифровывается с единственным CTR блоком.

Формально зашифрование CCM определено следующим образом.

1. Вычисление CBC-MAC:

   - Let $B_0.B_1.\cdots.B_r=\beta(N,H,M)$.

   - Let $Y_0= E_k(B_0)$.

   - For $1\le i\le r$,let $Y_i= E_k(Y_{i-1}\oplus B_i)$.

   - Let T be equal to the k_t leftmost bits of Y_r.

2. Шифрование CTR:

   - Let $\mu=[|M|/ k_b]$.

   - For $0\le i\le\mu$, let $A_i=\pi(i,N,H,|M|)$.

   - For $0\le i\le\mu$, let $S_i= E_k(A_i)$.

- Let S be equal to the |M| leftmost bits of $S_1.S_2.\cdots.S_\mu$ and let S' be equal to the |T| leftmost bits of S_0.

- Let C=[M⊕S].[T⊕S'].

3. Output C.

Расшифрование CCM шифр-текста C со случайным числом N и заголовком H определено очевидным образом: сначала, чтобы получить сообщение M и тэг CBC-MAC T, применяется к C обратный шаг 2; затем применяется CBC-MAC к β(N,H,M) как в шаге 1 для получения тэга T'. Если T=T', тогда C – достоверно и сообщение M выходит. Иначе C – недостоверно и на выходе получается ошибка. Заметим, что операция расшифрования не должна «выпускать» сообщение или любую его часть, пока тэг не будет проверен.

1.3. Теория информации в криптологии

В 1949 г. К. Шеннон заложил теоретическую базу для криптографии, опубликовав свою работу «Теория связи в секретных системах», где строго доказал безопасность некоторых криптосистем при определенных условиях. Окончился этап донаучной криптографии, основанной на вере. Результатам Шеннона во многом способствовало развитие теории информации. Остановимся на некоторых основных моментах. Материал составлен в основном по работам [6, 7].

Формально количество информации в послании измеряется **энтропией**.

Пусть $x_1 ..... x_n$ есть n возможных сообщений, появляющихся с вероятностями

$P(x_1),\ldots,P(x_n), \sum_{i=1}^{n} p(x_i)=1.$

Тогда **энтропия** данного сообщения x есть

$H(x) = - \sum_{i=1}^{n} p(x_i) \log p(x_i),$

или будем записывать ее как сумму по всем сообщениям x:

$$H(x) = -\sum_{i=1} p(x_i)\log_2 p(x) = -\sum_{i=1} p(x_i)\log_2 p(\frac{1}{p(x)}).$$

Аналогично, энтропия сообщения x при известном Y (точнее, средняя энтропия) есть

$$H(x/y) = -\sum_{x,y} p(x,y)\log_2 p(x/y) = \sum_y p(y) \sum_x p(x/y)\log_2\left[\frac{1}{p(x/y)}\right],$$

где p(x/y) – условная вероятность сообщения x при известном y. Энтропии (неопределенности) подчиняются таким правилам, как

$H(x/y) = H(x) – H(y/x)$.

Например, необходимость только одного бита в разделе «Пол» подтверждается следующим: P(муж.) = P(жен.) = 0,5. Тогда

$$H(x) = 0{,}5 \cdot \log_2 + 0{,}5 \cdot \log_2 2 = 1.$$

Интуитивно, каждый член $\log_2(1/p(x))$ в выражении для H(x) представляет число битов, необходимое для кодирования сообщения x оптимальным образом. Взвешенное среднее H(x) дает ожидаемое число битов при оптимальном кодировании. Так как 1/p(x) уменьшается при увеличении p(x), то при оптимальном кодировании используются короткие кодобозначения для часто появляющихся сообщений.

Для данного n величина H(x) принимает максимальное значение, равное $\log_2 n$, при выполнении соотношений $P(X_1) = ... = P(X_n) = 1/n$, то есть когда все сообщения одинаково вероятны.

Неопределенность H(x) уменьшается, когда распределение вероятностей сообщений становится более отличным от равновероятного и достигает минимума H(x) = 0, когда P(X_I) = 1 для некоторого сообщения X_I.

Теория информации имеет отношение к двум связанным проблемам: проблеме передачи по каналу с шумом и проблеме секретности передачи. В канале с шумом получатель должен по полученному сообщению восстановить истинное сообщение. В криптосистемах наложение шума соответствует шифрующему преобразованию, а полученное сообщение – шифртексту. Энтропия сообщения измеряет его неопределенность

(uncertainty) в числе бит информации, которая должна быть восстановлена, когда сообщение было изменено в канале с шумом или скрыто для криптоаналитика в шифртексте.

К. Шеннон различал **теоретическую** и **практическую стойкость** криптосистем. Криптосистема называется теоретически стойкой, если криптоаналитик противника не может уточнять распределение вероятностей возможных открытых текстов по имеющемуся у него шифртексту, даже если он не ограничен временем для анализа и обладает всеми необходимыми средствами, в том числе и неограниченной компьютерной мощностью. При этом предполагается, что секретный ключ используется только один раз.

По Шеннону **совершенная стойкость (секретность)** криптосистемы (perfect secrecy) означает, что открытый текст M и шифртекст C статистически независимы, то есть совпадают вероятности

$$P(M/C) = P(M)$$

для всех возможных значений м, с. Получение шифртекста не дает криптоаналитику дополнительной информации о посланном открытом тексте.

Определение совершенной стойкости можно представить и в терминах энтропии – должно выполняться равенство: H(M/C) = H(M).

Пусть P(C/M) – условная вероятность получения шифртекста C при условии, что известно, что зашифрован текст M на некотором неизвестном ключе. Тогда

$$P(C/M) = \sum_{k:E(M)=C} P(K),$$

где P(k) – вероятность использования ключа k, E_k – преобразование зашифрования на ключе k.

Обычно существует, по крайней мере, один ключ k, такой, что $E_k(M) = C$ для данных M и C, но в некоторых системах текст M может быть зашифрован в текст C при различных ключах.

Необходимым и достаточным условием для совершенной стойкости является то, что для каждого C и для всех M выполнено P(C/M) = P(C).

Последнее равенство означает, что вероятности получения конкретного шифртекста C при условии, что был зашифрован текст M, одинаковы для всех M.

Используя тот факт, что уменьшение объема известных сведений может лишь увеличить неопределенность, получаем соотношения

H(M/C) ≤ H(M,K/C) = H(K/C) + H(H/C,K) = H(K/C) ≤ H(K).

(В последнем равенстве, естественно, H(H/C,K) = 0.) Другими словами, для совершенно стойкого шифра неопределенность секретного ключа должна быть не меньше неопределенности шифруемого с его помощью открытого текста. Отсюда можно сделать вывод, что размер секретного ключа не должен быть меньше размера открытого текста. Это, конечно, практически неудобно.

Примером совершенно стойкой криптосистемы является система Вернама. В варианте шифрования двоичного открытого текста M =(M_1, ..., M_t) используется двоичный ключ K = (K_1, ..., K_t) той же длины для получения шифртекста по правилу C = M $\oplus$ K = $[M_1 \oplus K_1,....,M_t \oplus K_t]$. По формуле условной вероятности P(M=m/C=c) = P(M=m, C=c)/P(C=c). Раскроем значения числителя и знаменателя:

P(C=c)= $\sum_m$ [P(C=c/M=m) P(M=m)]= $\sum_m$ [P(K=c-m) P(M=m)] =

1/|K| $\sum_m$ [P(M=m)] = 1/|K| , т.к. P(K=c-m)=1/|K| .

P(M=m, C=c)= P(M=m, K=c-m)=P(M=m) P(K=c-m), т.к. величины K и M независимы. Используя эти выражения, получим для любых m и c требуемое равенство P(M=m/C=c) = P(M=m).

Помимо неудобств, связанных с большим объемом ключа, совершенно секретные системы могут быть не стойкими при известном открытом тексте. Кроме того, чрезмерно предположение о противнике с неограниченными вычислительными мощностями, да и сама информация может иметь

ограниченную временную ценность.

Поэтому К. Шеннон, помимо теоретической стойкости криптосистем, рассматривал и практическую стойкость. Для этого он ввел так называемую **рабочую характеристику** w(n) – среднее количество работы (измеренное в удобных единицах), требуемое для нахождения ключа на основе знания n знаков шифртекста с помощью наилучшего криптоаналитического алгоритма. Обычно криптосистемы оценивают с помощью достигнутой оценки рабочей характеристики $W(\infty)$ при использовании наилучшего из известных методов дешифрования. Криптосистемы называются **практически стойкими,** если они не могут быть вскрыты в течение реального времени ($W(\infty) \geq \text{const}$) всеми общедоступными методами. Хотя К. Шеннон исходил из анализа криптосистем лишь по шифртексту, определение подразумевает все типы криптоанализа.

При построении криптосистем на практике используют именно понятие практической стойкости. Однако при этом следует иметь в виду, что не все методы дешифрования, известные противнику, известны и создателю криптосистемы. Таким образом, квалификация создателя криптосистемы играет исключительно важную роль. Кроме того, новые исследования, возможно, могут открыть новые методы, которые сделают криптосистему практически нестойкой. Поэтому очень важно периодически проводить **контрольные исследования** уже созданных систем на основе вновь полученных результатов и разработанных методов.

Важным понятием при изучении несовершенно стойких шифров является введенное К. Шенноном понятие **расстояния единственности (unicity distance)** шифра. Оно определяется как наименьшее число знаков шифртекста, необходимое для однозначного определения ключа, то есть когда неопределенность H(K|c) при известном шифртексте данной длины равна (близка) нулю. (Величина H(K|c) – key equivocation – иногда переводится как «**ненадежность ключа».**) Системы, которые не обладают совершенной стойкостью, но, тем не менее, не вскрываемы, так как не дают

достаточной информации в шифртексте для однозначного определения ключа, называются «**идеально стойкими (секретными)» (ideal secrecy).** У идеально стойких систем неопределенность H(K|C) не достигает нуля ни при какой длине шифртекста. Для «**строго идеально стойких систем»** выполняется соотношение H(K|C) = H(k). По определению, для величины H(K|c) имеем соотношение

$$H(K|c) = -\sum_{c} P(c) \sum_{k} P(K|c) \log_2 P(K|c),$$

где P(K|c) – вероятность того, что данный ключ k использовался при получении данного шифртекста C. Большинство криптосистем слишком сложны, чтобы для них можно было определить все вероятности P(K|c), встречающиеся в последнем соотношении. Тем не менее, Шеннон показал, что возможно оценить H(K|c), используя так называемую **модель случайного шифра (random cipher model),** для которого при любых K и C из соответствующих пространств значение $D_k(c)$ является независимой случайной величиной, равномерно распределенной на пространстве всех открытых текстов (в действительности полной независимости быть не может, так как $D_k(c) \neq D_k(c')$ при $c \neq c'$ и любом k в силу однозначности зашифрования).

Пусть, для простоты изложения, мощности алфавитов открытого и шифрованного текстов равны L. Тогда существует $L^N = 2^{RN}$ последовательностей длин N, где $R = \log_2 N$ (**absolute rate of language**). Например, при *L=26* (английский язык), $R = \log_2 26 \approx 4{,}7$. Эти 2^{RN} последовательностей разбиваются на два множества: множество из $(2^r)^N$ осмысленных текстов длины N и множество из $2^{RN} - 2^{rN}$ бессмысленных текстов. Здесь r (**rate of the language**) обозначает **энтропию источника сообщений на один знак**. Например, для английского языка при больших значениях N величина r заключена в границах 1 бит/буква и 1,5 бит/буква. Предполагается, что все осмысленные тексты имеют одинаковую вероятность 2^{-rN}, тогда как все бессмысленные – нулевую вероятность.

Также естественно считать, что все $2^{-H(K)}$ ключей одинаково вероятны (P(K)= $2^{-H(K)}$) для использования, H(K) энтропия ключа (число битов в ключе).

Если шифртекст $c = E_k(m)$ получен при каких-то истинных значениях k и m, то криптоаналитик противника может принять ложный ключ k за истинный в следующих двух случаях:

$c = E_k(m')$,

или $c = E_k,(m')$,

при том же или другом осмысленном тексте m'. Так как каждый открытый текст одинаково вероятен, то вероятность получить осмысленный открытый текст при случайно выбранном для расшифрования ключе из множества ($2^{H(K)}$-1) ложных ключей равна $2^{rN} / 2^{RN} = 2^{(r-R)N} = 2^{-DN}$, где величина D = (N-r) называется **избыточностью языка** (**redundency of the language**). Таким образом, число ложных решений можно оценить величиной

$(2^{H(K)} - 1)2^{-DN} = 2^{H(K)-D(N)}$.

Отсюда видно, что число ложных решений равно нулю (единице), а значит, и неопределимость H(K|c) ключа тоже равна нулю при H(K)-DN=O или при N=N(K)/D. Величина H(k)/D является оценкой для расстояния единственности.

В качестве примеров можно посчитать расстояние единственности для широко известных криптосистем Hagelin H-209, DES и SKIPJACK, у которых величина H(к) равна 131, 56 и 80 бит соответственно. Приняв для английского языка D=3,2, получим значения расстояния единственности соответственно в 40,9; 17,5; 25 букв. Для шифра простой замены расстояние единственности будет равно 23,1 букве, но найти ключ для него гораздо проще, чем для DES.

Из отношения H(K)/D для расстояния единственности можно сделать некоторые выводы. Во-первых, если при любом N число возможных ключей так же велико, как число осмысленных открытых текстов, то

$$H(K)=-\sum_{2^{rN}}\frac{1}{2^{rN}}\log_2{}^{-rN}=rN,$$

и если $H(K)-DN = (r-D)N \neq 0$, то система является теоретически невскрываемой. Подобный принцип лежит в основе шифра Вернама (one-tine pad).

Во-вторых, при фиксированном размере ключа H(к) и нулевой избыточности языка D = R-r, расстояние единственности равно бесконечности, и криптоаналитик никогда не сможет раскрыть криптосистему, даже если число ключей много меньше числа осмысленных открытых текстов и совершенная секретность не имеет места. Действительно, для любого шифртекста с все тексты $D_k(c)$ при всех ключах к будут восприниматься криптоаналитиком как осмысленные открытые тексты. На основании этих рассуждений К. Шеннон предложил устранять избыточность открытого текста перед зашифрованием. Этого можно достичь, например, с помощью специального кодирования.

При выводе оценки расстояния единственности использовалась модель случайного шифра. Тем не менее, Шеннон отмечал, что эта оценка верна и для обычных классических криптосистем с секретными ключами. Проверка на примерах подтверждала этот вывод.

1.4. Теория сложности и криптология

Стойкость криптосистем определяется вычислительной сложностью (coeputatlonal complexity) алгоритмов, применяемых криптоаналитиками для дешифрования этих систем.

Вычислительная сложность алгоритма в свою очередь измеряется его **временной (T)** и **емкостной (S)** сложностями в зависимости от размера входных данных [7, 11, 42]. **Временная сложность** – это время, затрачиваемое алгоритмом для решения задачи, рассматриваемое как функция размера задачи или меры количества входных данных (например, размером задачи умножения матриц может быть наибольший размер матриц-

сомножителей). Аналогично, **емкостная сложность** – это объем необходимой машинной памяти. Поведение этих сложностей в пределе при увеличении размера задачи называется **асимптотическими сложностями**. Эти сложности алгоритма определяют в итоге размер задач, которые можно решить этим алгоритмом.

Если при данном размере входа в качестве меры сложности берется наибольшая из сложностей по всем входам этого размера, то она называется **сложностью в худшем случае.** Если в качестве меры сложности берется «средняя» сложность по всем входам данного размера, то она называется **средней** или **усредненной сложностью**. Обычно среднюю сложность найти труднее, чем сложность в худшем случае.

Теория сложности в основном имеет дело со сложностью задач в худшем случае. Однако в криптологии более важна средняя сложность задачи, а алгоритм с наименьшей сложностью в худшем случае не обязательно имеет наименьшую среднюю сложность. Например, как установлено в работах Кли и Минти, так называемый симплекс-метод для решения задач линейного программирования имеет большую (экспоненциальную) временную сложность, но в то же время этот метод хорошо работает на практике. Другой пример: алгоритмы ветвей и границ [7], успешно решающие так называемую задачу о рюкзаке (см. далее), имеют также большую (экспоненциальную) временную сложность.

Естественно, для анализа работы алгоритма нужны модели вычислительных машин, достаточно простых для анализа, но в то же время точно отражающих основные черты реальных машин. Так, рассматриваются модели, включающие машину с произвольным доступом к памяти, машину с произвольным доступом к памяти и хранимой программой, детерминированную и недетерминированную машину Тьюринга и др.

Хотя отмечается, что общая тенденция в разработке программ для описания алгоритма состоит в использовании языков высокого уровня, следует иметь в виду, что в криптологии важно учитывать и реальный

выигрыш во времени работы алгоритма от использования машинно-ориентированных языков и более сложных специальных моделей вычислительных машин.

Если алгоритм обрабатывает входы размера n за время cn^2, где c – некоторая постоянная, то говорят, что временная сложность этого алгоритма есть $O(n^2)$ (читается «порядка n^2.» Точнее, неотрицательная функция g(n) есть O(f(n)) (пишут g(n)=O(f(n)), если существуют постоянные c и n_0, для которых

$g(n) \le c|f(n)|$ при $n \ge n_0$.

Если $g(n) = a_n n^m + a_{m-1} n^{m-1} + a_1 n + a_0$ является полиномом степени m (deg g(n) = m), то $g(n)=0(n^m)$. Действительно, легко видеть, что

$|g(n)| \le |a_0|+|a_n|*n+\ldots+|a_m| \cdot n^m + \ldots + a_1 n + a_0$ при $n \ge 1$.

Напомним некоторые операции с символом «большое О». Так,

g(n) = O(g(n)); c•O(g(n)) = O(g(n)), если c = const ;

0(g(n))•0(g(n)) = O(g(n)); O(O(g(n))) = O(g(n));

O(f(n))•O(g(n)) = 0(f(n)g(n)); 0(f(n)g(n)) = f(n)0(g(n)).

Написанные равенства всегда односторонни. Так, пишется $3n^2+n = 0(n^2)$, но никогда не пишется $0(n^2) = 3n^2+n$

Полиномиальным алгоритмом или **алгоритмом полиномиальной временной сложности** называется алгоритм, у которого временная сложность равна T = 0(p(n)), где p(n) – некоторый полином, а n – размер входа. Алгоритм, временная сложность которого есть $T = 0\left[c^{p(n)}\right]$, где c = const, p(n) – полином, называется **экспоненциальным алгоритмом.**

В следующей таблице из работы [7] приведены времена для различных классов алгоритмов при $n = 10^6$ на последовательной машине, выполняющей 10^6 операций в секунду (1980 г.).

Класс	Сложность алгоритма	Число операций для $n = 10^6$	Реальное время
Полиномиальный			
Линейный	O(n)	10^6	1 с
Квадратичный	$O(n^2)$	10^{12}	10 дней
Кубичный	$O(n^3)$	10^{18}	27397 лет
Экспоненциальный	$O(2^n)$	10^{301030}	10^{301016} лет

Таблица 4

Задачи, которые решаются в полиномиальное время, называются **решаемыми (tractable),** так как они обычно могут быть решены для входа достаточно реального размера. Задачи, которые не могут быть систематически решаемыми за полиномиальное время, называют **нерешаемыми (intractable),** или просто **трудными (hard).**

Следует заметить, что существуют и **алгоритмически неразрешимые задачи (undecidable).** Они настолько трудны, что невозможно создать алгоритм для их решения. Такова, например, десятая проблема Гильберта: существует ли алгоритм, решающий в целых числах уравнение $p[x_1,....,x_n] = 0$ для многочлена p с целыми коэффициентами. Ю.В. Матиясевич показал, что такого алгоритма в принципе не существует. Другие примеры неразрешимых задач были ранее указаны Тьюрингом.

Далее приведена классификация задач по сложности и дано наглядное возможное их соотношение друг с другом (такое соотношение неизвестно). Схема приведена по работе [7], но несколько изменена для удобства изображения.

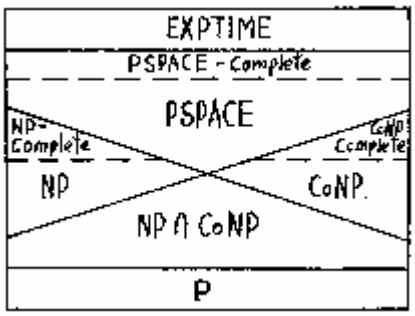

Рис. 5

Класс P или P-TIME (Polynomial) состоит из всех задач, решаемых за полиномиальное время. Примером может быть задача решения системы линейных уравнений от n неизвестных алгоритмом Гаусса.

Класс NP или **NP-TIME** (**Nondeterministic Polynomial**) состоит из всех задач, решаемых за полиномиальное время на недетерминированной машине Тьюринга, способной параллельно выполнять неограниченное количество независимых вычислений. Это означает, что если вариант решения проверяется, то он может быть проверен на машине Тьюринга за полиномиальное время. Конечно, это не означает решить задачу, так как нет гарантии, что машина угадает правильное решение. Чтобы математически решать задачу из NP класса, надо, по-видимому, тратить экспоненциальное время. Примером такой задачи является «**задача о рюкзаке» («knapsack problem»)**: дано множество из n целых чисел A = $\{a_1,....,a_n\}$ и целое число S, требуется определить, существует ли подмножество чисел из A, сумма которых равна S. Ясно, что эта задача принадлежит NP классу, так как для любого подмножества чисел в A легко проверить, равна ли их сумма числу S. Найти же подмножество чисел, сумма которых равна S, гораздо сложнее. Существует всего 2^n таких подмножеств (включая пустое множество), и проверка их всех имеет временную сложность T = O(2^n).

Другим примером задачи, которая имеет экспоненциальную временную сложность, является «**задача выполнимости» («Satisfiability problem»),**

заключающаяся в проверке существования выполняющего набора булевых переменных $v_1,....v_n$ для набора дизъюнкций над этим множеством переменных.

Класс NP включает класс P, так как любая задача, полиномиально решаемая на детерминированной машине Тьюринга, полиномиально решаема и на недетерминированной. Если бы все задачи из NP-класса полиномиально решались на детерминированной машине, то было бы верно равенство P=NP. Хотя многие задачи из NP-класса кажутся более сложными, чем задачи из P (например, задача о рюкзаке, задача о выполнимости), никто еще не доказал, что $P \neq NP$.

Основной метод, используемый для доказательства близости задач, состоит в «сведении» их друг к другу с помощью конструктивного преобразования, которое позволяет превратить любой алгоритм решения одной задачи в алгоритм решения другой. Известно много примеров подобной сводимости. В 1971 г. Кук (Cook) показал, что задача о выполнимости имеет свойство, что каждая другая задача из NP может быть сведена к ней за полиномиальное время. Далее Карп (Karp) доказал, что многие хорошо известные комбинаторные задачи, включая «задачу о коммивояжере», столь же трудны, как задача о выполнимости.

Класс эквивалентности, состоящий из всех «самых трудных» задач NP, включающий задачу о выполнимости, получил название **NP-полных задач (NP-complete).** Если какая-нибудь из NP-полных задач окажется в P, то будет доказано равенство NP = P.

Класс Co-NP состоит из всех задач, являющихся дополнительными некоторых задач из NP. Если задача в NP имеет вид «определить, существует ли решение», то задача из Co-NP имеет вид «показать, что решений нет». Неизвестно, верно ли равенство NP = Co-NP, но есть задачи, принадлежащие пересечению классов NP и Co-NP. Примером такой задачи является «**задача о разложении целых чисел»**: дано целое n, определить, существуют ли

делители p и q, такие, что n = pq. Задача нахождения делителей, однако, сложнее, чем задача установления разложимости числа n.

Класс PSPACE состоит из задач, требующих полиномиальный объем машинной памяти, но не обязательно решаемых в полиномиальное время. Он включает классы NP и Co-NP, но есть задачи в PSPACE, которые, по-видимому, труднее, чем задачи в NP и Co-NP.

Класс PSPACE-полных задач (PSPACE-complete) состоит из задач, имеющих свойство, что если какая-нибудь из них окажется принадлежащей классу NP, то PSPACE = NP, или, если какая-нибудь из них окажется в P, то PSPACE = P.

Наконец, **класс EXP-TIME** состоит из задач, решаемых в экспоненциальное время, и включает класс PSPACE.

В заключение уместно вспомнить слова К. Шеннона о том, что «проблема создания хорошего шифра является, по существу, проблемой нахождения наиболее сложных задач, удовлетворяющих определенным условиям. Можно составить наш шифр таким образом, чтобы раскрытие его было эквивалентно (или включало в себя) решению некоторой проблемы, про которую известно, что для ее решения требуется большой объем работы» [1].

1.5. Некоторые необходимые сведения из теории чисел

Следуя традиции ряда учебников по криптографии [23, 61], далее приводятся сведения из теории чисел, необходимые для понимания излагаемого материала.

Для данных целых чисел a, b и $m \neq 0$ говорят, что числа a и b **сравнимы по модулю m**, если оба числа при делении на m дают один и тот же остаток, или, что равносильно, их разность a-b делится на m (обозначается m|a-b), или, если существует целое число t, такое, что a = b +tm. Все определения эквивалентны.

Сравнимость является **бинарным отношением** на множестве целых чисел Z.

Для обозначения факта сравнимости пишут $a \equiv b(\mathrm{mod}\, m)$. При этом говорят, что b является **вычетом (residue) числа a по модулю m**. Из определения сразу следуют соотношения для произвольных целых чисел:

$a \equiv a(\mathrm{mod}\ m)$, (рефлективность);

$a \equiv b(\mathrm{mod}\ m) \Rightarrow b=a(\mathrm{mod}\ m)$, (симметричность);

$a \equiv b(\mathrm{mod}\ m), b=c(\mathrm{mod}\ m) \Rightarrow a \equiv c(\mathrm{mod}\ m)$, (транзитивность),

которые означают, что отношение «сравнимость по модулю m» является **отношением эквивалентности** на множестве целых чисел Z. В соответствии с отношением эквивалентности множество Z разбивается на классы эквивалентности, которые в данном случае называются **классами вычетов по модулю m**. Число классов равно m, оно совпадает с числом различных остатков при делении на число m.

Множество из m целых чисел $r_1,.....,r_m$ называется **полной системой вычетов по модулю m**, если для любого целого числа a существует точно одно число r_i из этого множества, такое, что $a=r_i$ (modm). Примером полной системы вычетов служит множество {0,1,..., m-1}.

Операции сложения и умножения в Z индуцируют операции сложения и умножения над классами вычетов, когда за результат операции над двумя классами $\bar{a}$ и $\bar{b}$ берется класс $\bar{C}$, содержащий результат операции над любыми представителями (например, a и b) из этих классов. Иногда вычисления над классами вычетов целых чисел называют **модульной арифметикой (modular arithmetic).**

Множество классов вычетов Z/m образует относительно этих операций (как и Z) коммутативное кольцо с единицей.

Кольцом называется множество с двумя операциями (обозначаемыми обычно «+» (сложение) и «•» (умножение)), относительно которых выполняется ряд следующих аксиом.

По операции «+» множество образует **абелеву (коммутативную)** группу, то есть для любых элементов a, b, c выполнено: операция «+» всюду определена; a+(b+c) = (a+b)+c (ассоциативность); a+b = b+a

(коммутативность); существует нейтральный элемент по сложению 0, такой, что a+0 = 0+a = a; существует обратный элемент -a, такой, что a+(-a) = (-a)+a = 0 (эту аксиому можно заменить на разрешимость уравнения a+x = b относительно неизвестного x).

По операции «•» множество является **полугруппой,** то есть операция эта всюду определена, и выполняется ассоциативность a+(b-c) = (a-b)-c.

Наконец, операции «+» и «•» связаны дистрибутивным законом, то есть a(b+c) = ab + ac.

Если в кольце выполнен коммутативный закон для умножения (ab=ba), то говорят о **коммутативном** кольце. В большинстве приложений рассматриваются кольца, в которых существует нейтральный элемент (единица) по умножению ($1 \cdot a = a \cdot 1 = a$). В этом случае кольцо называется **кольцом с единицей.**

В дальнейшем из алгебраических структур с двумя бинарными операциями, помимо кольца, потребуется тело и поле. **Телом** называется кольцо с единицей, в котором для каждого элемента $a \neq 0$ существует обратный a^{-1} по умножению элемент, $a \cdot [a^{-1}] = 1$. **Поле** – это тело, операция умножения в котором коммутативна, a*b = b*a.

Отображение кольца целых чисел Z *в* кольцо Z/m классов вычетов по модулю m, при котором каждому числу a из Z ставится в соответствие класс его содержащий, является **гомоморфизмом** колец, то есть отображением, для которого вычисление над образами элементов дает тот же результат, что и вычисление над соответствующими элементами-прообразами с последующим отображением результата.

Для произвольных элементов $a_1 = t_1 m + r$, $a_2 = t_2 m + r_2$ из классов $\overline{a_1}$ и $\overline{a_2}$ имеем

$$a_1 + a_2 (\mathrm{mod}\, m) = \left[(t_1 + t_2)m + (r_1 + r_2)\right] \mathrm{mod}\, m = r_1 + r_2 (\mathrm{mod}\, m) =$$
$$= \left[a_1 (\mathrm{mod}\, m) + a_2 (\mathrm{mod}\, m)\right] (\mathrm{mod}\, m)$$
$$a_1 \cdot a_2 (\mathrm{mod}\, m) = \left[(t_1 \cdot t_2 m + r_1 t_2 + r_2 t_2)m + r_1 r_2\right] \mathrm{mod}\, m =$$
$$= \left[r_1 \cdot r_2\right] \mathrm{mod}\, m = \left[a_1 (\mathrm{mod}\, m) \cdot a_2 (\mathrm{mod}\, m)\right] (\mathrm{mod}\, m)$$

Отсюда видно, что множество классов вычетов Z/m, как и Z, является коммутативным кольцом. Но в общем случае оно не является телом и полем. Кольцо Z/m является полем тогда и только тогда, когда число m – простое число.

Простым числом называется целое число большее 1, которое делится только на 1 и на себя. Остальные числа – **составные.** Число 2 является простым. Каждое целое число $m > 1$ допускает единственное представление в виде $m = p_1^{a1} \ldots p_r^{ar}$, где $P_1,\ldots P_r$ – простые числа, $a_1,\ldots,a_r$ – целые числа >= 0 (основная теорема арифметики). Существует бесконечно много простых чисел. Для функции n(x), определяемой как число простых положительных целых чисел, меньших некоторого целого x, К. Гаусс обнаружил соотношение n(x)/(x/logx) $\rightarrow$ 1, при x $\rightarrow \infty$ (доказано Адамаром и Валле – Пуссеном). Чебышев П.Л. показал в 1850 г., что

$$0.89 \cdot \frac{x}{\log x} < \pi(x) < 1{,}11 \cdot \frac{x}{\log x} \cdot$$

Дальнейшие результаты в этой области связаны с Селбергом, Эрдешем, Риманом. Простые числа находят широкое применение в криптографии.

Наибольшим общим делителем (НОД) чисел $a_1,\ldots a_r$ называется такой их общий делитель d[d|$a_1,\ldots,$d|a_r], который делится на все другие общие делители чисел $a_1,\ldots a_r$. По равносильному определению НОД $[a_1,\ldots a_2]$ – это наибольшее целое число, делящее все числа $a_1,\ldots a_r$.

Процесс нахождения НОД d произвольного числа чисел $a_1,\ldots a_r$ $(r \geq 2)$ сводится к нахождению НОД двух чисел следующим образом:

$d_1 = НОД[d_{i-1}, a_i]$, $i = \overline{1,r}$; $d_1 = a_1$, $d = d_r$.

Если известно разложение двух чисел $a = p_1^{k1} \ldots p_r^{kr}$ и $b = p_1^{I1} \ldots p_s^{Is}$ по степеням простых чисел, то НОД (a,b)= $\prod_{P_i} P_i^{\min(k_i, I_i)}$

Если таких представлений нет, то для нахождения НОД используется **алгоритм Евклида**, состоящий в последовательном определении для чисел a последовательности чисел $r_0 = a$, $r_2 r_3$,…., таких, что r_{i+1} равно остатку от

деления r_{i-1}, на r_i до тех пор, пока не получено $r_k = 0$. Тогда НОД (a,b) = r_{k-1}, то есть совпадает с последним ненулевым остатком. Число делений с остатком в алгоритме Евклида не превосходит 5*1g(max(a,b)) . Наибольшее число делений реализуется для двух последовательных чисел Фибоначчи $u_k = a$, $u_{k-1} = b$, где $u_{i+1} = u_i + u_{i-1}$, i = 1,2, …, $u_0 = u_1 = 1$.

Расширенный алгоритм Евклида позволяет находить такие целые числа x и y, что xa + yb = НОД (a.b). Для этого находятся последовательно числа x_i и y_i, с условием $r_i = x_i r_0 + y_i r_i$ полагая $x_0 = y_1 = 1$, $x_1 = y_0 = 0$,

а для i ≥ 1 $x_{i+1} = x_{i-1} - qx_i$, $Y_{i+1} = Y_{i-1} - qy_i$, где q определено из равенства $r_{i+1} = r_{i-1} - qr_i$.

Трудоемкость этих алгоритмов оценивается как O((log m)^3).

Числа $a_1,....,a_r$ называются **взаимно простыми,** если НОД ($a_1,....,a_r$) = 1. Аналогично определяется **наименьшее общее кратное (НОК)** чисел $a_1,....,a_r$ как такое их общее кратное, которое делит все другие общие кратные (или как наименьшее положительное целое число, делящееся одновременно на все числа $a_1,....,a_r$).

Если известно разложение двух чисел $a = p_i^{k1} ... p_r^{kr}$ и $b = p_1^{l1} ... p_s^{ls}$ на простые сомножители, то НОК[a,b] = $\prod_{pi} p_i^{\max(k_i, I_i)}$.

Если таких представлений нет, то для нахождения НОК(a,b) можно воспользоваться соотношением

НОК[a,b]= $\frac{a \cdot b}{НОД(a,b)}$, $a, b \neq 0$.

В коммутативном кольце ненулевой элемент a называется **делителем нуля,** если существует другой ненулевой элемент b, такой, что $a \cdot b = 0$. Если в кольце нет делителей нуля, то оно называется **кольцом без делителей нуля** или **областью целостности.** Примером области целостности является кольцо целых чисел Z. Никакой делитель нуля кольца не может иметь обратного по умножению элемента. Отсюда, никакое тело (поле) не содержит делителей

нуля. Если m – составное число, то кольцо Z/m содержит делители нуля (если $m = r \cdot s$, то $\bar{r} \cdot \bar{s} = \bar{0}$).

Все представители какого-либо класса $\bar{r} \in$ Z/m имеют один и тот же НОД с модулем m, поэтому имеет смысл говорить о **НОД класса $\bar{r}$ с модулем m**. Особую роль играют классы взаимно-простые с модулем. Произведение двух классов взаимно-простых с модулем является классом взаимно-простым с модулем.

Для произвольного модуля m каждый ненулевой класс вычетов либо является делителем нуля, либо он обратим. Обратимыми являются в точности все классы взаимно-простые с модулем m. Они образуют группу по умножению, которая называется **мультипликативной группой кольца Z/m** и обозначается (Z/m)*.

Если число a взаимно-просто с модулем m, то с помощью расширенного алгоритма Евклида находятся числа x и y такие, что $a \cdot x + m \cdot y = 1$. Отсюда в кольце Z/m выполняется соотношение $\bar{a} \cdot \bar{x} = \bar{1}$. Это есть способ вычисления обратных по умножению элементов в кольце Z/m.

Число классов из Z/m, взаимно-простых с модулем, равное порядку группы (Z/m)*, обозначается через $\varphi(m)$. Числовая функция $\varphi(m)$ называется **функцией Эйлера**. Для простого числа m = p очевидно, что $\varphi(m)$ = p-1. Если известно разложение числа m = $p_1^{k1} \ldots p_s^{ks}$ на простые сомножители, то справедливо равенство $\varphi(m)$ =[1-1/p_1]…[1-1/p_1].

Теорема Эйлера. Для любых целых чисел a и m, таких, что
НОД (a.m) = 1, верно соотношение $a^{\varphi(m)} = 1 (\mathrm{mod}\, m)$, или, что эквивалентно, в кольце Z/m верно равенство $a^{-\varphi(m)} = \bar{1}$.

Обощением теоремы Эйлера является **теорема Кармайкла**, где вместо функции Эйлера используется **функция Кармайкла**.

При m = p, где p – простое число, из теоремы Эйлера следует так называемая **малая теорема Ферма**: для простого числа p и любого числа a,

такого, что НОД(a,p) = 1, выполнено соотношение $a^{p-1} \equiv 1(\mathrm{mod}\, p)$. Следствием является то, что для произвольного a выполнен $a^p = a \pmod p$.

Нужно заметить, что выполнение условия $a^{m-1} \equiv 1(\mathrm{mod}\, p)$ не обеспечивает простоты числа m, но его невыполнение устанавливает, что m – составное число. Число m называют **псевдослучайным по основанию a**, если НОД (a.m) = 1 и $a^{m-1} \equiv 1(\mathrm{mod}\, p)$. Существуют составные числа, являющиеся псевдослучайными для всех a, взаимно-простых с модулем m. Такие числа называются **числами Кармайкла** (Carnichael). Например, m=56=3*11*17, 1105 = 5*13*17. Если m – составное число, то оно является псевдослучайным лишь для менее чем половины возможных оснований a, и если t различных оснований $a_i, \ldots, a_t$ выбраны независимо и случайно, то составное число m выдержит **тест Ферма** $a_i^{m-1} \equiv 1$ (modm), i= $\overline{1,t}$ с вероятностью не более 2^{-t}. Такие «вероятностные тесты» впервые предложили Соловэй (Solovay) и Штрассен (Strassen). Далее они были развиты [64].

Теорема Эйлера может быть также использована для нахождения обратного элемента $a^{-1} = a^{\varphi(m)-1} \bmod m$ к элементу a, НОД(a,m) = 1. Но, в отличие от метода с применением алгоритма Евклида, требуется знание функции $\varphi(m)$. Если m = p – простое, то $\varphi(m) = p - 1$. Нахождение обратного элемента есть решение частного случая **линейного сравнения 1-й степени** вида $ax \equiv b$ относительно неизвестного x.

Теорема. Пусть d = НОД(a.m). Если d $\mid$b, то сравнение $ax \equiv b \pmod m$ будет иметь d решений вида

$$f(x) \equiv [\mathrm{mod}\, m]\ x = \left[\left\langle \frac{a}{b} \right\rangle + t\left\langle \frac{m}{d} \right\rangle\right] \mathrm{mod}\ \text{ m для t = 0,1,….,d-1,}$$

где x_0 – решение сравнения $\left[\frac{a}{b}\right] x \equiv 1 \pmod m$. Если d не делит b, то решений нет.

Пусть f(x) = $c_n x^n + c_{n-1} x^{n-1} + \ldots + c_1 x + c_0$ многочлен с целыми коэффициентами. Два решения x_1 и x_2 сравнения n-й степени $f(x) \equiv 0 \pmod m$

называются эквивалентными, если $x_1 \equiv x_2 (\mathrm{mod\ m})$. Число решений сравнения определяется числом неэквивалентных решений. Классы эквивалентных решений сравнения находятся во взаимно-однозначном соответствии с решениями уравнения $\overline{f}(x) = \overline{c_n} x^n + ... + \overline{c_1} + \overline{i_0} = \overline{0}$ в кольце Z/m.

Теорема. Пусть $m_1, ..., m_r$ попарно взаимно простые числа ([НОД($m_1 m_j$)] = 1) и $m = m_1 \cdot ..... \cdot m_r$. Тогда:

1) сравнение $f(x) \equiv 0[\mathrm{mod}\, m]$ (*) равносильно системе сравнений

$f(x) \equiv [\mathrm{mod}\, m_1], ..., f(x) \equiv 0[\mathrm{mod}\, m_r]$;

2) число решений сравнения $f(x) \equiv 0[\mathrm{mod}\, m]$ равно произведению числа решений отдельных сравнений указанной выше системы.

Китайская теорема об остатках. Пусть $m_1 ... m_r$ – попарно взаимно простые числа. И пусть $a_1 ... a_r$ – целые числа. Тогда существует число x, такое, что $x \equiv a_1 [\mathrm{mod}\, m], ..., x \equiv [\mathrm{mod}\, m_r]$. Решение этой системы однозначно определено по модулю $m = m_1 ... m_r$.

Если обозначить через $n_j = m / m_r$, то НОД[n_j, m_r] = 1, и по расширенному алгоритму Евклида находятся числа t_j и s_j, такие, что $t_j m_j + s_j n_j$ = 1. Пусть $e_j = s_j n_j$. Тогда $e_j = 1[\mathrm{mod}\, m_j]$ и $e_j = 0[\mathrm{mod}\, m_i]$ при i ≠ j. Решение x системы, указанной в китайской теореме об остатках, имеет вид **x =** $a_1 e_1 + ... a_r e_r$**.**

Обобщением предыдущей теоремы служит следующая.

Теорема. Следующие утверждения эквивалентны:

(1) Система $x \equiv a_1 [\mathrm{mod}\, m_1], \ldots, x \equiv a_r [\mathrm{mod}\, m_r]$ имеет решение.

(2) Для любой пары индексов 1 ≤ i, J ≤ n верно сравнение $a_i \equiv a_j \left[НОД(m_i, m_j) \right]$, кроме того, решение, если оно существует, единственно по модулю $НОК[m_1, ..., m_r]$.

В случае, когда m = p – простое число, кольцо вычетов целых чисел Z/p, как было уже отмечено выше, является полем и в нем любой элемент, отличный от нуля, имеет обратный. Это поле называется **полем Галуа GF(p) (Galois Field).** При этом мультипликативная группа (Z/p)* ненулевых

элементов является **циклической группой**, то есть она совпадает с множеством $\{a, a^2, ..., a^{p-1}\}$ степеней некоторого (неоднозначно определенного) элемента a.

Наименьшее из чисел $\gamma : a^\gamma \equiv 1$ mod m называется показателем числа a по модулю m [НОД(a,m) $\equiv 1$]. Существование таких чисел γ обеспечивается теоремой Эйлера. Если показатель числа a по модулю m равен δ, то $a^\gamma \equiv a^\beta$ (modm) тогда и только тогда, когда $\gamma \equiv \beta (\mathrm{mod}\, \delta)$. Показатель числа a по модулю m является делителем числа $\varphi(m)$, что сразу ограничивает число возможных значений показателя. Если показатель числа a по модулю m равен δ, то показатель числа a^k равен $\delta / НОД(\delta, k)$ для произвольного целого числа k. В частном случае, когда показатель числа a по модулю m равен $\varphi(m)$, число a называется **первообразным (примитивным) корнем по модулю m**.

При m = p существует $\varphi(p-1)$ первообразных корней. По **теореме Гаусса** первообразные корни существуют только по модулю вида $m = 2, 4p^\alpha, 2p^\alpha$ (p – нечетное простое число). Если p имеет вид p = 2q + 1 при некотором простом числе q, то показатель любого числа a легко найти, так как он равен одному из делителей числа φ(p) = 2q. Также легко найти первообразные корни по модулю p = 2q + 1, число которых равно φ(p-1) = q-1 = (p-3)/2, и при больших p, случайно выбранный элемент имеет вероятность, близкую к 1/2 быть первообразным корнем.

Если p – простое число и число a является первообразным корнем по модулю p, то любой элемент $b \in \overline{1, p-1}$ имеет однозначное представление в виде $b = a^x \bmod p$ для некоторого целого числа $x \in \overline{0, p-1}$. Число x при этом называется **дискретным логарифмом (или индексом) числа b по основанию a.** Понятие дискретного логарифма можно ввести для любой группы, но от вида группы зависит сложность нахождения логарифма. Сложность вычисления дискретных логарифмов в поле GF(p) определяется величиной $0\left[\exp(\ln p)^{1/2} \cdot (\ln((\ln p)^{1/2}\right]$. Метод решета числового поля (number

field sieve) более эффективен, его сложность $0\left[\exp(\ln p)^{1/3}\cdot(\ln((\ln p)^{2/3}\right]$. Более подробно об этом изложено в работах [17, 23, 32].

Другие сведения из теории чисел и алгебры, необходимые для изложения, будут даны далее в соответствующих разделах.

Раздел 2. ОСНОВНЫЕ ПОНЯТИЯ И МЕТОДЫ СОВРЕМЕННОЙ КРИПТОЛОГИИ

2.1. Криптографические протоколы

Протокол – это точно определенная последовательность действий, посредством которых две или более стороны совместно решают (выполняют) некоторую задачу. Хотя это определение недостаточно строгое и смысл понятия будет ясен из дальнейших примеров, следует обратить внимание на некоторые моменты, отмеченные в [42].

Во-первых, раз это последовательность действий, то действия имеют очередность. Ни одно действие не выполняется, пока не закончится предыдущее. Точное определение каждого действия предполагает, что не должно быть двусмысленности в действиях и из каждой ситуации должен быть определенный выход. Во-вторых, одно действующее лицо недостаточно для протокола, хотя один человек может выполнять серию последовательных шагов для выполнения какой-нибудь задачи. Кроме того, предполагается, что все участвующие в протоколе стороны должны заранее знать всю последовательность действий, а также должны быть согласны следовать этой последовательности действий. В-третьих, наконец, важно, что стороны именно выполняют некоторую задачу, а не просто что-то делают.

Криптографический протокол – это протокол, в котором используются криптографические функции или примитивы (алгоритмы) и который гарантирует, что данные функции (алгоритмы) действительно обеспечивают безопасность (секретность, подлинность, целостность...) информации.

Существуют примеры, когда цели безопасности не достигаются не в результате нестойкого используемого криптографического алгоритма, а из-за наличия слабых мест в протоколе. Поэтому разработка системы безопасности информации опирается на две довольно независимые области исследований: во-первых, разработку стойких алгоритмов и, во-вторых, разработку надежных протоколов. Описание криптографического протокола должно

включать спецификацию требуемых характеристик криптографических алгоритмов.

Важно отметить, что цель криптографического протокола в общем случае не только в обеспечении секретности от стороннего нарушителя, но и, например, защиты от взаимного обмана. Причины для недоверия есть всегда, особенно в компьютерных сетях со многими пользователями. Конечно, надо верить, что большинство действуют честно, но наивно было бы полагать, что все или всегда. Поэтому протокол должен быть таким, чтобы исключить или обнаружить попытки обмана участниками взаимодействия.

Протоколы с арбитром (Arbitrated Protocols)

Арбитр (arbitrator) – это не заинтересованная, пользующаяся всеобщим доверием сторона, непосредственно участвующая в протоколе. Незаинтересованность означает, что арбитр не имеет своих личных предпочтений, как завершить протокол, и не имеет зависимости ни от одной из сторон. Доверительность означает, что все участники протокола признают, что любые утверждения или действия арбитра истинны и корректны.

Простым примером протокола с арбитром может служить спортивная игра футбол. Другим хорошим примером арбитра является нотариус (электронный нотариус). Но при перенесении понятия арбитра на компьютерный уровень возникает ряд проблем. Компьютерная сеть должна дополнительно нести стоимость обеспечения работы арбитра. Кто-то должен платить за это. Кроме того, существуют издержки времени в любом протоколе с арбитром, так как арбитр должен контролировать каждое действие сторон. Арбитры являются узким местом в любом сложном протоколе, а увеличение числа арбитров ведет к увеличению временных и материальных затрат. Особенно важно то, что арбитр представляет собой самую уязвимую точку для злоумышленника.

Протоколы с третейским судьей (Adjudicated Protocols)

Эти протоколы не требуют больших затрат на арбитров, так как арбитр – третейский судья появляется только в исключительных случаях, когда возникают спорные ситуации между сторонами. Третейский судья также является незаинтересованной и пользующейся доверием стороной, но, в отличие от арбитра, он не участвует напрямую в протоколе. Участие третейского судьи не всегда необходимо. Существующие компьютерные протоколы исходят из того, что участвующие стороны действуют честно по правилам, но если кто-нибудь обманывает, то существуют данные, по которым незаинтересованная третья сторона может определить, что кто-то обманывает. В хороших протоколах с третейским судьей он может также определить, кто именно обманывает.

Самодостаточные протоколы (Self Enforcing protocols). Это наилучший тип протоколов, когда сам протокол гарантирует соблюдение правил. Такой протокол построен так, что не может быть никакой спорной ситуации. И если одна из сторон попытается обмануть, то другая сторона немедленно определит это. К сожалению, не всегда удается построить такой протокол для каждой ситуации.

Атаки на протоколы могут быть направлены как против криптографических алгоритмов, используемых в протоколах, как против средств, использованных для реализации алгоритмов (например, против генератора ключей), так и непосредственно против самих протоколов. Такие атаки делятся на **пассивные и активные**. Во время пассивной атаки злоумышленник только наблюдает за действиями сторон в протоколе и старается извлечь из этого полезную ему информацию, не вмешиваясь и не нарушая протокола. (Это соответствует атаке на криптосистему только по шифрованному тексту.) При активной атаке на протокол злоумышленник старается видоизменить протокол в собственных интересах. Он может попытаться ввести новые сообщения в протокол, удалить присутствующие,

заменить одно сообщение на другое, вывести из строя канал связи или изменить хранимую в компьютерах информацию.

Атакующий не обязательно может быть сторонней стороной, он может быть и законным участником протокола (участником, пользователем системы). Кроме того, таких злоумышленников может быть несколько, они могут даже действовать совместно в сговоре. Ситуации могут возникать самые разнообразные, и защититься от них – исключительно сложная задача.

2.2. Однонаправленная функция

Понятие это, как сказано в [3], первым ввел Нидхэм (Needham R.M.) в работе о защите входа в вычислительные системы.

Функция f(x) называется **однонаправленной** (one-way function – иногда переводят как односторонняя функция), если для всех x из ее области определения легко вычислить y=f(x), но почти для всех y из ее области значений, нахождение любого x, для которого y=f(x), вычислительно неосуществимо.

Заметим, что в определении фраза «почти для всех y» необходима, так как можно легко вычислить и запомнить согласно первой части определения целую таблицу значений $y_1 = f(x_1)$, $y_2 = f(x_2), \ldots, y_n = f(x_n)$, и если окажется, что выбранное значение y принадлежит этой таблице, то соответствующий x может быть легко определен. Фраза «вычислительно неосуществимо» означает с точки зрения теории сложности, что неосуществимо за полиномиальное время, которое характеризует «легко вычислимые задачи».

Может быть дано более строгое определение однонаправленной функции, но это лишь затруднит первое знакомство с такой функцией.

До сих пор строго не доказано, что однонаправленные функции существуют. Показано, что проблема существования однонаправленной функции эквивалентна известной нерешенной проблеме совпадения классов задач P и NP. Тем не менее, было предложено много функций, которые похожи на односторонние. У. Диффи в качестве односторонней функции

предложил функцию на основе задачи об укладке ранца. Д. Кнут высказал идею о том, что умножение двух простых чисел не составляет труда, а вот разложение полученного числа – достаточно трудная задача. В частности, возведение в квадрат просто, а извлечение квадратного корня сложно.

В качестве возможной односторонней функции Дж. Гилл предложил показательную функцию f(x) $= a^x \mod m$, где a (основание) принадлежит интервалу (l,m-1), а умножение ведется по модулю числа m. Значение $a^x \mod m$ вычисляется достаточно эффективно, даже при больших числах x за $0[\log_2 n]$ операций, с помощью схемы Горнера «слева-направо» или «справа-налево».

Если двоичное разложение числа x имеет вид $x = \sum_{i=0}^{k-1} x_i 2^i$ =(x_(k-1),…, x_0), то

$$a^x \bmod m \begin{cases} (((a^{x_{k-1}})\cdot a^{x_{k-2}})^2 \cdot a^{x_0} \bmod m \\ a^{x_0} (a^2)^{x_1} \cdot \ldots \cdot (a^{2^{k-1}})^{k-1} \bmod m \end{cases}$$

(вариант «слева-направо»)/(вариант «справа-налево»).

Операция, обратная к этой операции, известна как операция **вычисления дискретного логарифма (DLP – Discrete Logarithm Problem)**: по данным a и y из алгебраической группы G найти такое целое m, что a*a*…*a (m раз) = y. До настоящего времени не найдено достаточно эффективных алгоритмов решения этой задачи для некоторых групп. Более подробно об этом сказано в [50].

С помощью показательной функции Диффи и Хеллман разработали так называемую **схему открытого распределения ключей**.

Диффи и Хеллман в работе [3] предложили для построения однонаправленных функций использовать симметричные криптосистемы с одним ключом, стойкие к методам вскрытия по известному открытому и шифрованному текстам (known plaintext attack). Если зафиксировать какой-нибудь открытый текст M_0 и рассмотреть отображение $f : \tilde{k} \to \tilde{c}$ пространства

ключей в пространство шифртекстов, определяемое в виде f(x) = $E_x[M_0]$ = c, то нахождение $x \in K$ по f(x) эквивалентно задаче нахождения ключа по известному открытому и соответствующему шифрованному текстам.

Для шифрования информации однонаправленная функция не применима.

Понятие однонаправленной функции оказалось также связанным с понятием широко используемых в криптографии **генераторов псевдослучайных последовательностей** [23], а также с понятием так называемой **функции хэширования информации** [17]. Кроме того, это понятие послужило толчком к определению однонаправленной функции с секретом (trap-door one-way function).

2.3 Открытое распределение ключей. Схема Диффи-Хеллмана

Одной из проблем криптографии, решенной в работе Диффи и Хеллмана [3], стала проблема распределения (рассылки) ключей для секретной связи. Использование криптосистем с секретным ключом предполагает заблаговременные до сеансов связи договоренности между абонентами о сеансовых секретных ключах или их предварительную пересылку по защищенному каналу связи. Разработанные в [4] принципы так называемого **открытого распределения ключей (ОРК)** и **открытого шифрования (ОШ)** и явились «новыми направлениями в криптографии», давшими начало криптографии с открытым ключом. Несмотря на то, что идеи ОРК и ОШ были сформулированы одновременно, удалось сразу предложить конкретную реализацию только для схемы ОРК, на основе показательной односторонней функции. Эта схема получила название **экспоненциального ключевого обмена Диффи-Хеллмана**.

В настоящее время употребимы и другие названия приведенного взаимодействия участников A и B. В терминологии справочника [23] это есть так называемый **протокол согласования ключей** (key agreement protocol), при котором общий секретный ключ получается как функция от информации,

созданной и переданной каждым из законных участников связи так, что никто из них не может предугадать результирующее значение ключа.

На CRYPTO`94 Hughes E. предложил **протокол транспортировки ключей** (при котором один из участников создает ключ и безопасно передает его другим законным участникам связи), похожий на протокол Диффи-Хеллмана. Протоколы согласования ключей и протоколы транспортировки ключей являются разновидностями **протоколов установки ключей** (key establishment protocol), при которых общий секретный ключ становится доступным всем законным участникам взаимодействия для последующего использования.

Рассмотрим схему в ее оригинальном изложении. В протоколе обмена секретными ключами предполагается, что все пользователи знают некоторое большое простое число p и примитивный элемент α ($1<\alpha<p$) конечного поля GF(p) (то есть элемент, степени которого a^t дают все ненулевые элементы {1,2,...,p-1} простого поля). Такие элементы всегда существуют, их число равно $\varphi(\varphi(p))$, где φ – функция Эйлера. Для выработки общей секретной информации k пользователи A и B должны сделать следующее.

1. Пользователи A и B независимо выбирают случайные числа K_a и K_b из интервала {1,...,p-1}, называемые **секретными ключами пользователей**.

2. Вычисляют с использованием известных p и α величины

$y_a = \alpha^{k_a} (\mathrm{mod}\, p)$ и $y_b = \alpha^{k_b} (\mathrm{mod}\, p)$,

которые являются **открытыми ключами пользователей**.

3. Обмениваются ключами Y_a и Y_b по открытому каналу связи (с подтверждением их авторства, чтобы избежать замены их кем-то другим).

4. По полученным ключам Y_a и Y_b каждый из пользователей независимо вычисляет секретный параметр K, который и будет их общим сеансовым секретным ключом.

$$A: y_B^{k_A} (\mathrm{mod}\, p) = \left[\alpha^{k_B} \right]^{k_A} \mathrm{mod}\, p = \alpha^{k_B k_A} \mathrm{mod}\, p = k$$

$$B: y_A^{k_B} (\mathrm{mod}\, p) = \left[\alpha^{k_A} \right]^{k_B} \mathrm{mod}\, p = \alpha^{k_A k_B} \mathrm{mod}\, p = k$$

Еще раз следует подчеркнуть, что в схеме не устраняется необходимость аутентификации открытых ключей в п. 3, то есть подтверждения того факта, что Y_a и Y_b действительно выработаны пользователями А и В.

Безопасность (секретность) изложенной схемы зависит от (не превышает) сложности вычисления дискретных логарифмов в мультипликативной группе конечного поля GF(p). Пока не найдено удовлетворительных быстрых алгоритмов нахождения К из $\alpha, \alpha^{K_A}, \alpha^{K_B}$.

В настоящее время наиболее интересны с точки зрения реализации следующие алгебраические группы, где сложно решается задача вычисления дискретных логарифмов:

- мультипликативная группа конечного поля: $GF(p)^*$, p – простое число > 2 ;

- мультипликативная группа конечного поля характеристики 2: $GF(2^n)^*$;

- группа точек эллиптической кривой над конечным полем F: EC(F).

Существуют **общие методы решения задачи дискретного логарифмирования для произвольной группы:** метод Шенкса (Daniel Shanks) или «baby steps, giant steps»; ro-метод Полларда (рандомизированный алгоритм); λ - метод Полларда.

Для некоторых групп задача решается просто. Например, для аддитивной группы кольца вычетов целых чисел (Z_m, +) по модулю m задача сводится к решению сравнения a x=b (mod m), что очень просто.

Для мультипликативной группы целых чисел по модулю простого числа p сложность задачи долго определялась сложностью **м**етода Шенкса и ro-метода Полларда: O($\sqrt{q}$), где q – наибольший простой делитель числа (p-1), хотя метод Полларда почти не использует память. В 1993 г. Van

Oorschot и Wiener предложили способ распараллеливания метода Полларда на t процессоров. Это дало сложность в t раз меньшую.

Обычно число q выбирают сравнимым с числом p, например, берут p=2q+1.

Использование дополнительных свойств группы привело к новым методам.

Методы исчисления индексов (Index calculus methods).

Обычно состоят из трех этапов. Двух предварительных вычислительных этапов: генерации соотношений, решение уравнений – и оперативного рабочего этапа: вычисление доступных логарифмов с использованием результатов предыдущих этапов. Они имеют субэкспоненциальную сложность. $L_p[1/2, c]$, где c – константа > 0.

$L_p[t,c] = \exp((c+0(1))((\ln p)^t(\ln\ln p)^{1-t})$

Метод линейного решета (Lenear sieve method) и **Gaussian integer method** достигли c=1.

Улучшить оценку сложности позволило ускорение предварительных вычислительных этапов. Решение разреженных систем линейных уравнений за $O(n^2)$ шагов (метод Wiedemann, Berlekamp-Massey algorithm). (См. Odliyzko, Eurocrypt`84.)

В 1988 г. Поллард нашел новый подход для факторизации целых чисел. Х. Ленстра (H. Lenstra) развил его для чисел специального вида. Далее рядом авторов он был распространен на произвольные числа со сложностью L_N [1/3, c], где $c=(64/9)^{1/3}$ = **1,9229**… . Далее Копперсмит улучшил оценку до **c=1,9018**… (см. Modifications to the Number Field Sieve(NFS), J.Cryptology 6(1993),169-180).

В 1992 г. Гордон Д. (Gordon) адаптировал метод NFS для решения задачи логарифмирования при простом p со сложностью $L_p[1/3, c]$, где **c=2,0800**. В 1993 г. О. Широкауер (Shirokaurer) снизил оценку с до **c=1,9229**. Алгоритм Широкауера был реализован Вебером, и в работе [Weber D. Eurocrypt`95, 95–105] описано логарифмирование по модулю p порядка

10^40, а в работе 1996 г. [Schirokauer O., Weber D., Denny T., ANTS II, 337–362] p порядка 10^65. Жу и Лерсье [Joux A., Lercier R.] в апреле 2001 г. провели логарифмирование по модулю 10^120. Это рекорд для модуля не специального вида (2003).

В 2003 г. Матюхин Д.В. довел константу c до **c = 1,9018** (см. Дискретная математика и ее приложения, 13(2003), №1, 17–50)). Это на сегодня лучшая оценка трудоемкости.

Для группы точек эллиптической кривой над конечным полем задача логарифмирования (ECDLP – дискретное логарифмирование на эллиптической кривой) пока решается ро-методом и λ - методом Полларда. Для группы из N элементов, ро-метод Полларда дает сложность O($\sqrt{(N/2)}$), а λ- метод Полларда дает O($\sqrt{N}$). Методы Полларда работают в любой абелевой группе. Оба метода эффективно распараллеливаются. До сих пор не предложено субъэкспоненциальных алгоритмов для произвольных кривых, что и делает эти группы привлекательными для использования (см. пособие далее).

Однако существуют субъэкспоненциальные алгоритмы для специфических кривых: суперсингулярных ($t^2 = 0, q, 2q, 3q$, или $4q$) и аномальных (кривая над Z_p с p точками).

В отчете проекта **NESSIE (2003)** после анализа сложности основных теоретико-числовых задач, предложена таблица эквивалентных по стойкости размеров ключей.

Размер симметричного ключа	56	64	80	112	128	160
Размер модуля (pq)	512	768	1536	4096	6000	10000
Размер характеристики поля эллиптической кривой	112	128	160	224	256	320

Таблица 5. Размеры эквивалентных по стойкости ключей

Заметим, что в российском стандарте цифровой подписи ГОСТ 34.10-2001 минимальной характеристики простого поля $p > 2^{255}$, в то время как размер симметричного ключа в ГОСТ 28147-89 равен 256.

В результате исследований стали различать задачи Диффи-Хеллмана.

Вычислительная задача Диффи-Хеллмана (The computational DH problem – CDH): дано – циклическая группа G = (g), g, g^a, g^b; найти g^{ab}.

Задача принятия решения Диффи-Хеллмана (The decisional HH problem – DDH): дано – G = (g), g, g^a, g^b, g^c; определить, верно равенство $g^{ab} = g^c$ или не верно.

The Gap DH problem: дано – G = (g), g, g^a, g^b и оракул, который корректно решает задачу принятия решений Диффи-Хеллмана. Найти: g^{ab}.

Очевидно, что если задача DDH является трудно решаемой в группе (g), то тогда трудно решаемы задачи CDH и DLP.

t-сильная задача Диффи-Хеллмана (t-strong Diffie-Hellman(t-sDH) problem): дано – G = (g), g, g^{a^i}) для i = 1,2,…,t; найти: $g^{a^{t+1}}$.

Задача SDH обощается на так называемые билинейные группы.

t-слабая задача Диффи-Хеллмана (t-weak Diffie-Hellman(t-wDH) problem: дано – G = (g), g, g^{a^i}) для i = 1,2,…,t; найти: $g^{1/a}$.

Существенные продвижения в снижении сложности задачи логарифмирования могут быть достигнуты за счет применения квантовых компьютеров и разработанных для них специальных алгоритмов [Shor P., 1997], что существенно повлияет на стойкость связанных с этими группами криптосистем.

2.4. Односторонняя функция с секретом

Диффи и Хеллман развили понятие однонаправленной (односторонней) функции, позволившее приспособить его для целей шифрования и давшее начало всей новой области криптографии – криптографии с открытым ключом [3].

Однонаправленной (односторонней) функцией с секретом (или с лазейкой, с потайной дверью – a trapdoor one-way function) называется зависящая от параметра k функция $f_k(x)$, такая, что при известном параметре k известны полиномиальные алгоритмы E_k и D_k, позволяющие легко вычислять соответственно значение $f_k(x)$ для всех x из области определения и прообраз $f_k^{-1}(y)$ для всех y из области значений, однако почти для всех k и почти для всех y из области значений функции f_k нахождение $f_k^{-1}(y)$ вычислительно неосуществимо без знания k (не существует полиномиального алгоритма), даже при известном алгоритме E_k.

Один из первых предложенных примеров таких функций основан на степенной функции f(x) = x^m mod n, вычисление которой при известных m и n производится одним из методов быстрого экспоненцирования не более чем за $0(\log_2 n)^3$ операций умножения. Преобразование, обратное к преобразованию возведения в степень x^m mod n, называется **вычислением корня m-й степени по модулю n**. В настоящее время эффективный алгоритм вычисления такого x, что x^m mod n = y при известных числах m, n и y требует знания разложения числа n по степеням простых чисел. Таким образом, эта информация может служить как секретный параметр k.

Именно эта односторонняя функция с секретом используется в криптосистеме с открытым ключом RSA [5]. Другие широко известные примеры односторонних функций с секретом легли в основу криптосистемы на основе задачи о рюкзаке (knapsack function) [11], криптосистемы Мак Элайса [25] и другие. Выбор систем и функций для изучения должен определяться их распространенностью применения и объемом необходимого для их объяснения нового материала.

2.5. Открытое шифрование, криптосистема с открытым ключом

На основе введенного понятия односторонней функции с секретом Диффи и Хеллман предложили новый принцип так называемого открытого шифрования, который вместе с открытым распределением ключей составил новое направление в криптографии, когда для организации секретной связи не требуется снабжения абонентов секретной информацией (ключами).

Действительно, если получатель секретного сообщения выберет одностороннюю функцию с секретом $f_x(x)$ и сообщит по открытому каналу отправителю эффективный алгоритм E_k ее вычисления, то отправитель может вычислить значение функции $f_k(M) = C$ от сообщения M и передать это значение получателю также в открытом виде. Только знающий секретный параметр k знает алгоритм D_k вычисления обратной функции $f_k^{-1}(C)$ и может определить M.

В криптографии вместо понятия функции чаще употребляется понятие криптосистемы, а термин «секретный параметр» из определения функции с секретом будет означать секретный ключ.

Итак, **криптосистема с открытым ключом** представляет собой систему, включающую следующие компоненты:

- пространство открытых текстов M;
- пространство шифрованных текстов C;
- пространство секретных ключей K;
- множество преобразований зашифрования $m \in M \left\{E_k, k \in K\right\}$,

  $E_k : M \to C$, где $m \in M, c \in C$;
- множество преобразований расшифрования $\left\{D_k \in K\right\}$.

При этом преобразования E_k и D_k должны удовлетворять следующим свойствам.

1. Для каждого $k \in K$ преобразование D_k является обратным к преобразованию E_k, то есть $D_k\left[E_k(M)\right] = M$ при всех $m \in M$.

2. По каждому выбранному $k \in K$ легко найти пару обратимых преобразований E_k и D_k.

3. Для всех $k \in K, m \in M, c \in C$ величины $E_k(M)$ и $D_k(C)$ легко вычисляются (в полиномиальное время).

4. Почти для всех $k \in K$ D_k вычислительно невозможно из E_k вывести какое-либо легко вычисляемое преобразование, эквивалентное преобразованию D_k.

Свойство 4 отличает эти системы от криптосистем с секретным ключом, рассмотренных ранее, в которых преобразования E_k и D_k также зависят от параметра к, но знание преобразования E_k дает знание k и D_k или наоборот, знание D_k позволяет определить k и E_k, так что преобразования E_k и D_k либо оба известны, либо оба неизвестны. Это свойство 4 позволяет не засекречивать преобразование E_k, а сделать его открытым, общедоступным.

Так как злоумышленник имеет доступ к преобразованию зашифрования E_k, то он может всегда выбрать любой открытый текст и получить соответствующий ему шифрованный текст. Поэтому криптосистемы с открытым ключом должны быть всегда стойки к методам по выбранному открытому тексту (chosen-plaintext attacks).

Также видно, что если вероятных открытых текстов сравнительно мало и существует возможность применения метода полного их перебора (или энтропия сообщения слишком мала), то система небезопасна. Этот недостаток может быть устранен добавлением к сообщению строки из случайных бит, чтобы один и тот же открытый текст за счет этого добавления зашифровывался в разные шифрованные тексты. (Этот прием рандомизации применяется и в криптосистемах с секретным ключом.)

В ряде учебников по криптографии [7] секретное преобразование D_k называют **секретным ключом** (private key), а открытое преобразования E_k – **открытым ключом** (public key), а сами системы называют также **двухключевыми криптосистемами** (two key cryptosystem). Другим

синонимом этих систем является понятие **асимметричной криптосистемы** (asymmetric cryptosystem), в то время как обычные криптосистемы с секретным ключом называются **симметричными** (symmetric cryptosystem).

Криптосистемы с открытым ключом обеспечивают только практическую стойкость.

Интересно, что авторы идеи и понятия о системах открытого шифрования не смогли сразу предложить такую систему для реализации (в отличие от схемы открытого распределения ключей). Первыми такими системами являются ранцевая криптосистема Меркля-Хеллмана [11] и криптосистема Райвеста-Шамира-Адлемана (RSA) [5].

2.6. Криптосистема RSA

Название криптосистемы образовано от первых букв фамилий предложивших ее авторов (Rivest H., Shamir A., Adleman L.) [5].

Это одна из первых и широко сейчас применяемых криптосистем с открытым ключом. Она построена на основе степенной односторонней функции с секретом, рассмотренной ранее.

Рассмотрим сначала **блочную экспоненциальную систему шифрования** (к которым принадлежит и RSA), когда каждый блок M открытого текста, представленный как целое число от 0 до (n-1), преобразуется в блок $C \in [0, n-1]$ шифрованного текста вычислением

$C = E_{(e,n)}(M) = M^e \bmod n$, где (e, n) – **ключ преобразования зашифрования**. При расшифровании блок открытого текста M восстанавливается также экспоненцированием, но с другой степенью d в качестве ключа расшифрования:

$M = D_{(d,n)}(C) = C^d \bmod n$.

Зашифрование и расшифрование могут быть выполнены с использованием быстрых алгоритмов экспоненцирования не более чем за $0[\log_2 n]$ операций. Выясним, при каких значениях параметров e, d и n такие преобразования возможны.

Теорема. Если числа e, d и n удовлетворяют соотношению $ed \equiv 1(\mathrm{mod}\,\varphi(n))$ ($\varphi(n)$ – функция Эйлера), а $M \in 1,\overline{n-1}$ взаимно просто с n, то $\left[M^e \bmod n\right]^d \bmod n = M$. (Это и означает, что D(E(M)) = M.)

Действительно, $\left[M^e \bmod n\right]^d \bmod n = M^{ed} \bmod n$. Условие $ed \equiv 1(\mathrm{mod}\,\varphi(n))$ означает, что $ed = 1 + t \cdot \varphi(n)$ для некоторого целого t. Таким образом,

$M^{ed} \bmod n = (M \cdot 1) \bmod n = M$, где $M^{t\cdot\varphi(n)} \bmod n = \left[M^{\cdot\varphi(n)} \bmod n\right] \bmod n = 1^t \bmod n = 1$.

Значит, $M^{ed} \bmod n = (M \cdot 1) \bmod n = M$. Здесь мы воспользовались **теоремой Эйлера**, согласно которой для каждого числа M, взаимно простого с модулем n, выполнено соотношение $M^{\varphi(n)} \equiv 1(\mathrm{mod}\,n)$.

При наличии $\varphi(n)$ (или при известном разложении n по степеням простых чисел) можно легко найти пару чисел e и d, удовлетворяющих соотношению $ed \equiv 1(\mathrm{mod}\,\varphi(n))$. Для этого выбирается число e, взаимно простое с $\varphi(n)$. Взаимная простота нужна для разрешимости сравнения $e \cdot x \equiv 1(mo\varphi(n))$ относительно неизвестного x. Число d находится с помощью расширенного алгоритма Евклида, определяющего числа d и t, удовлетворяющие соотношению ed+t $\varphi(n)$ = 1, которое и означает, что $e \cdot d \equiv 1(\mathrm{mod}\,\varphi(n))$. Сложность алгоритма Евклида не превосходит $0(\log n)^3$ операций.

Все сказанное справедливо для произвольного модуля n. В этом виде система может использоваться как симметричная, когда оба ключа e и d являются секретными ключами.

Рассмотрим теперь классическую систему RSA. В системе RSA модуль n есть произведение двух больших простых чисел p и q, $n = p \cdot q$. Поэтому $\varphi(n)$=(p-1)(q-1) (в общем случае напомним, что $\varphi(n) = n\left[1 - 1/p_1\right] \ldots \left[1 - 1/p_r\right]$ для $n = p_1^{\alpha_1} \ldots p_r^{\varphi_r}$).

Без знания простых сомножителей p и q значение функции $\varphi(n)$ определить очень трудно, и если сделать открытыми числа e и n, а число d держать в секрете, то нахождение числа M из C сводится к трудной задаче извлечения корня степени e из числа C по модулю m. Это и означает, что

предложенная система может быть использована как **система шифрования с открытым ключом**, когда преобразование зашифрования открыто, а преобразование расшифрования держится в секрете. В этом случае открытым ключом является пара (e,n), а закрытым – (d,n).

Напомним, что по теореме Эйлера числа M и n =pq должны быть взаимно простыми. Именно в этом предположении была показана справедливость преобразований зашифрования – расшифрования. Но оказывается, что вероятность этого близка к единице при больших значениях p и q. Действительно, число чисел M, не взаимно простых с n, равно $n-\varphi(n)=pq-(p-1)(q-1)=p+q+1$, а их доля в интервале [0,1] близка к нулю в силу соотношения $\frac{n-\varphi(n)}{n}=\frac{p+q-1}{n}<\frac{1}{p}+\frac{1}{q}$, при больших значениях p и q.

Можно показать, что при модуле, равном произведению двух простых чисел n=pq, соотношение «зашифрования-расшифрования» также верно для любого M, а не только для M, взаимно простого с n.

Для полноты картины можно отметить, что есть небольшая вероятность того, что шифрование числа M возведением его в степень e по модулю n не изменяет открытый текст M. Так, в [12] показано, что для каждого значения e существует, по крайней мере, девять значений M, таких, что $M^e \equiv M \bmod n$. Так как e взаимно просто с $\cdot\ \varphi(n)=(p-1)(q-1)$, то e является нечетным числом, и поэтому два сравнения $x^e \equiv x \bmod p$ и $x^e \equiv x \bmod q$ имеют три решения. x=0, 1, -1. Комбинируя их в условиях китайской теоремы об остатках, получаем искомые 9 сообщений M.

Например, для $n=47\cdot 59=2773$ это: 0, 1, 2772, 2537, 2303, 235, 2538, 471, 236.

В криптосистеме RSA вместо функции Эйлера $\varphi(n)$ можно рассмотреть функцию Кармайкла (Carmichael) – $\lambda(n)$, определяемую для произвольного целого числа следующим образом:

$$\lambda(m)=\begin{cases}2^{t-1}\,если\{n=2^t, t<3\}\\ 2^{t-2}\,если\{n=2^t, t\geq 3\}\\ H.O.K.[\lambda(2^{a_0}),....,\lambda(p_r{}^{a_r})]если\{n=2^{a_0}p_1^{a_1}...p_r^{a_r}, 2+p_i..\}\end{cases}.$$

Значение функции Эйлера совпадает с порядком мультипликативной группы целых чисел по модулю n, а значение функции Кармайкла совпадает с экспонентой этой группы.

В **теореме Кармайкла**, обобщающей теорему Эйлера, утверждается, что для любого целого числа M, взаимно простого с числом n, верно сравнение $M^{\lambda(n)}\equiv 1(\mathrm{mod}\, n)$.

Далее в приведенных рассуждениях $\varphi(n)$ можно заменить на $\lambda(n)$. При n = pq имеем $\lambda(n)$ = НОК [p-1, q-1] = $\varphi(n)$/нод (p-1, q-1), где в знаменателе стоит наибольший общий делитель чисел (p-1) и (q-1), который не меньше 2 в силу четности p-1 и q-1.

Как уже было сказано, описанная блочная криптосистема может использоваться и как обычная криптосистема с секретным ключом. Так, в **криптосистеме Полига-Хеллмана (Pohllg-Hellman**) в качестве n выбирается большое простое число p. Оно может быть и несекретным. Числа e и d являются секретными ключами $(\varphi(p)=p-1)$. Стойкость этой криптосистемы зависит от сложности вычисления дискретных логарифмов в конечном поле GF(p), так как при атаке с известным открытым текстом криптоаналитик решает задачу нахождения числа $e=\log_m C$, где C и M – шифрованный и открытый тексты соответственно.

Рассмотрим теперь вопрос эффективной реализации криптосистемы.

Воспользуемся для этого китайской теоремой об остатках, позволяющей сводить вычисление по большому модулю n=pq к вычислениям по меньшим модулям p и q, и малой теоремой Ферма. Покажем это на примере вычисления $M=C^d \bmod n$. Введем обозначения $a=C^d \bmod p$, $b=C^d \bmod p$. Разделив d на (p-1) и (q-1) с остатком, получим d = k(p-l)+ r =J(q-l)+s или r = d mod (p-1), s = d mod(q-1). Далее находим

$$a = \left[C^{p-1}\right]^{k} C^{r} \bmod p = C^{r} \bmod p = \left[C \bmod\right]^{d \bmod(p-1)} \bmod p,$$

$$b = \left[C^{q-1}\right]^{j} C^{s} \bmod q = C^{s} \bmod q = \left[C \bmod\right]^{d \bmod(q-1)} \bmod q.$$

Для восстановления M из вычисленных таким образом чисел a и b найдем и, 0<u<p, из условия $u \cdot q \equiv 1 (\bmod p)$. Тогда, как показано в [12], текст M находится одним из двух способов.

$$M = \begin{cases} (((a - (b \bmod p))u) \bmod p)q + b < npu > a \geq b \bmod p \\ (((a + (b \bmod p))u) \bmod p)q + b < npu > a < b \bmod p \end{cases}$$

Легко видеть, что сложность операции расшифрования составляет $0[\log_2 n]^3$ операций.

Стойкость системы RSA

Существует несколько постановок вопроса о вскрытии RSA. Под задачей RSA понимают задачу нахождения текста M по имеющимся числам C, n, а под гибкой задачей RSA – нахождение по C и n пары чисел M и e, таких, что $C = E_{(e,n)}(M) = M^{e} \bmod n$.

Существуе много атак как на сам алгоритм RSA, так и на протоколы с использованием алгоритма RSA. Рассмотрим сначала атаки на алгоритм.

1. Сразу заметим, что если число значений открытого текста мало, то его можно найти простым перебором и сравнением с шифртекстом C.

Чтобы избежать этого, нужно воспользоваться конструкцией RSA-OAEP. Схема оптимального расширения для асимметричного шифрования (Optimal Asymmetric Encryption Padding – OAEP) – преобразует строку M открытого текста и случайную равновероятную строку r с помощью двух функций хэширования G и H в строку вида ((M||0^{k_1} xor G(r))|| (r xor H(M||0^{k_1})). Далее эта строка шифруется по схеме RSA. Схема предложена Белларе и Роджуэем на конференции Eurocrypt`1994. Альтернативой ей является RSA-KEM, рекомендованная проектом NESSIE в 2003 г. Схема RSA-OAEP поддерживается стандартами ANSI X9.44, IEEE P1363, SET.

2. Стойкость можно оценить сверху сложностью разложения большого числа n на множители p и q с последующим определением $\varphi(n)$ и d. Именно

потребности криптографии дали толчок в развитии методов факторизации целых чисел. Обзор по этим методам можно найти в [50]. Прогресс методов факторизации существенным образом влияет на выбор ключевых параметров p и q.

Метод Ферма основан на соотношении $(p+q/2)^2=n+ (p-q/2)^2$, которое позволяет перебирать (p-q) и искать полный квадрат $(p+q/2)^2$

В 1988 г. Поллард нашел новый подход (the Number Field Sieve) для факторизации целых чисел, который далее рядом авторов был распространен на произвольные числа со сложностью $L_N[1/3, c]$, где $c=(64/9)^{1/3} = 1,9229…$, где $L_N[t,c] = \exp((c+0(1))((\log n)^t)(\log\log n)^{(1-t)})$.

Далее Копперсмит в 1993 г. улучшил оценку для c=1,9018…

В трудах конференции CRYPTO`00 приведена формула года Y, когда будет факторизовано целое число из D десятичных знаков
$Y = 1928,6 + 13,24\ D^{1/3}$. Эта формула получена на основе достигнутых значений с учетом развития как вычислительной техники, так и методов факторизации. Например, при D=231(768 бит) получаем Y = 2010; при D=309(1024 бит) – Y = 2018.

До сих пор в общем случае не показана эквивалентность задачи факторизации и задачи RSA. В 2006г. G. Leander, A. Rupp показали, что любой эффективный общий (generic) алгоритм, учитывающий структуру кольца Z_N, который решает гибкую (flexible) задачу RSA с малой открытой экспонентой, может быть преобразован в эффективный алгоритм факторизации модуля N=pq.

Таким образом, задача RSA с малой экспонентой **e** трудно решаема в классе указанных алгоритмов при условии, что такой является задача факторизации.

Задачу факторизации модуля n=pq легко решить, зная функцию Эйлера от n из соотношения $p^2 – (n - \varphi(n)+ 1)p+n=0$. То есть задача нахождения функции Эйлера эквивалентна задаче факторизации.

Известно, что нахождение d эквивалентно задаче факторизации. В работе [38] можно найти доказательство того факта, что если для системы RSA удалось найти такое число d, что $M^{ed} \equiv M \bmod n$ для всех M взаимно простых с модулем n, тогда это дает возможность разложить число n на множители. Изложенный там алгоритм является примером так называемого вероятностного алгоритма. Далее Coron J. и May A. на CRYPTO 2004 представили детерминированный алгоритм решения.

3. Атака с помощью нахождения дискретного логарифма.

Противник, зная открытые значения e и n может выбрать свой открытый текст M_0 , получить $C_0 = (M_0)^e$ и по этим данным пытаться найти d из соотношения $M_0 = (C_0)^d$, то есть d= $\log_{C_0}(M_0)$. То есть задача RSA не сложнее задачи дискретного логарифмирования.

4. То, что определить открытый текст в системе RSA можно без разложения числа n на множители, показывает следующий метод итерации преобразования зашифрования, предложенный в [12]. В этом методе криптоаналитик пытается найти целое число j, такое, что

$C^{e^j} \bmod n = C$, где $C = M^e \bmod n$ – шифртекст, полученный из M на ключе (n,e). Если такое число j найдено, то M получается следующим образом:

$C^{e^{j-1}} \bmod n = M$, так как справедливо соотношение $\left[C^{e^{j-1}} \bmod n\right]^e \bmod n = C$.

В [12] приводится пример для p = 983, q = 563, e = 49, M = 123456.

Тогда $C = M^{49} \bmod n = 1603$, $C_1 = C^{49} \bmod n, ..., C_8 = C_7^{49} \bmod n = 85978$, $C_9 = C_8^{49} \bmod n = 123456 = M$, $C_{10} = C_9^{49} \bmod n = 1603 = C$ (число j делит порядок числа e по модулю $\varphi(n)$). В примере j = 10.

Приведенную идею «бесключевого чтения» можно обобщить и стараться найти многочлен p(e) вида p(e) = e·Q(e), такой, что $C^{p(e)} = C(\bmod n)$. Тогда открытый текст M находится вычислением $M = C^{Q(e)} \bmod n$.

Из приведенных данных следуют некоторые требования для выбора параметров. a). abs(p-q)> 2^{128}. b). Все числа (p-1), (q-1), (p+1), (q+1) должны

иметь большие простые делители, чтобы противостоять известным методам факторизации и логарифмирования. c). Мультипликативный порядок e по модулю НОК[p-1, q-1] должен быть большим.

4. Атака Винера (M.Wiener) в случае малой секретной экспоненты d.

В 1990 г. Винер показал, что если $d < (1/3)n^{1/4}$ и $q<p<2q$, то секретную экспоненту d можно найти с использованием аппарата цепных дробей, зная e/n. Идеи Винера были далее развиты в работе Майтры и Саркара [ePrint Archive 2008/228]. Они показали, что возможно решение задачи RSA, если $d=n^s$ и $s<3/4-s-r$, где $2q-p= n^s$ и r малое значение. А также, когда $d<0,5\ n^s$ и $e=O(n^{3/2-2s})$ для $s <= 0,5$.

5. В 1999 г. Boneh и Durfee увеличили предложенную Винером границу безопасности для секретного ключа d до $n^{0,292}$).

6. Выше был приведен эффективный метод для расшифрования/подписи по схеме RSA, если для секретной экспоненты d малы величины d mod(p-1) и d mod(q-1). Такие d называются **малыми CRT-экспонентами**.

На конференции CRYPTO 2007 было показано (Jochemsz E., May A.), что если d mod(p-1) и d mod(q-1) меньше $n^{0,073}$, то n можно разложить на множители за полиномиальное время.

7. Атака на основе времени работы алгоритма (Timing attack). В 1995 г. Кочер (Kocher, CRYPTO`96) предложил способ определения секретной экспоненты d на основе определения времени расшифрования $M_i = C_i^d \bmod n$ в случае аппаратной реализации алгоритма. Например, ясно, что это время тем больше, чем больше единиц в двоичном разложении числа d, когда применяется быстрое экспоненцирование по схеме Горнера «справа-налево» или «слева-направо».

8. Патарин (Patarin J.) предложил метод дешифрования в случае малой экспоненты e и известной разности двух шфруемых текстов M_1 и $M_2= M_1+h$. В этом случае M_1 является корнем многочлена НОД $(X^e-C_1, (x+h)^e-C_2)$, который с большой вероятностью линеен при малой e.

9. Атаке может быть подвержен и протокол, использующий стойкую схему RSA. Например, при циркулярной связи, когда двум разным абонентам передается один и тот же текст M, зашифрованный на разных e_1 и e_2, но с одним значением n. В этом случае $M=C_1^r C_2^s$, где $(e_1)r+(e_2)s=1$, если нод(e_1, e_2)=1.

2.7. Цифровая подпись

Одним из основных применений криптосистем с открытым ключом является их использование при создании так называемой цифровой подписи (digital signature). Впервые идея цифровой подписи была высказана все в той же известной работе Диффи и Хеллмана [3].

Для ее изложения потребуем, чтобы преобразования зашифрования E_k и расшифрования D_k из определения криптосистемы с открытым ключом действовали также соответственно на пространствах шифрованных и открытых текстов.

$$E_k : \mathcal{C} \to \mathcal{M}$$
$$D_k : \mathcal{M} \to \mathcal{C},$$

и к тому же преобразование E_k было бы обратным преобразованием к D_k, то есть выполнялось бы соотношение

$E_k[D_k(M)] = M$ для любого открытого текста $M \in \mathcal{M}$.

Теперь если некий пользователь A желает послать сообщение M пользователю B с подтверждением своего авторства этого сообщения, то он может воспользоваться своим секретным преобразованием $D_k = D_{A,K}$, вычислить величину из $C = D_{A,K}(M)$ и послать это значение (именно оно и будет цифровой подписью) пользователю B. То есть в этом случае преобразование $D_{A,K}$ используется как шифрующее преобразование текста M. В этом смысле система цифровой подписи обратна системе открытого шифрования.

Пользователь B, а также любой другой пользователь, знающий открытое преобразование $E_{A,K}$ пользователя A, может убедиться в авторстве сообщения M вычислением этого сообщения из соотношения $M = E_{A,k}(C) = E_{A,k}\left[D_{A,k}(M)\right]$ и проверкой, является ли полученное значение осмысленным открытым текстом. Здесь преобразование $E_{A,k}$ действует как преобразование расшифрования.

Авторство пользователя A основано на том, что только он знает секретное преобразование $D_{A,k}$. Злоумышленник, желающий подменить сообщение M на другое осмысленное сообщение M', должен решить задачу нахождения такого значения C', что

$M' = E_{A,k}(C')$.

В силу же односторонней природы преобразования $E_{A,k}$ сделать это вычислительно невозможно.

В действительности, для сокращения времени подписывания и размера подписи (а также и для других целей) в процессе подписи используется общеизвестная функция H, действующая на пространстве открытых текстов и отображающая любое сообщение M в сообщения H(M) фиксированного малого размера, которое далее и преобразуется в подпись $D_{A,k}\left[H(M)\right] = H(M)$. Пользователь B, получив сообщение M и подпись к нему $D_{A,k}\left[H(M)\right]$, может также вычислить H(M) и проверить выполнимость соотношения

$E_{A,k}\left[D_{A,k}\left[H(M)\right]\right] = H(M)$.

Функция H называется функцией хэширования (hash function). Более подробно она будет рассмотрена отдельно.

Теперь будет понятно следующее определение [7] схемы (системы) цифровой подписи, которая включает следующие компоненты.

1. Пространство открытых сообщений $M = \{m\}$, к которым применяется алгоритм цифровой подписи.

2. Пространство секретных параметров $K = \{k\}$, которые выбираются пользователем.

3. Алгоритм G генерации за полиномиальное время пары $[E_k, D_k]$ – открытого и секретного ключей по выбранному параметру k. (Здесь, как и в ряде учебников, отождествляются открытое и секретное преобразования E_k и D_k с открытым и секретным ключами.)

4. Алгоритм подписи Q, который вырабатывает значение $Q[M, E_k]$ (с использованием секретного ключа D_k), называемое **цифровой подписью** к сообщению M.

5. Алгоритм проверки подписи, который проверяет правильность подписи $Q[M, D_k]$ к сообщению M с использованием открытого ключа E_k.

Для криптографа важно определить, какими возможностями может располагать злоумышленник при атаке на схему цифровой подписи и оценивать ее стойкость исходя из этого.

По возможности доступа к информации о схеме выделяют три класса методов атаки.

1. Методы атаки только по открытому ключу (key-only attack). В этом случае предполагается, что злоумышленник знает только открытый ключ схемы подписи и имеет возможность проверки правильности подписей сообщений, которые у него окажутся.

2. Методы атаки по открытому ключу и сообщениям (message attacks). В этом случае злоумышленник знает открытый ключ подписывающего и может наблюдать случайные пары (сообщение/подпись).

3. Методы атаки по выбранным сообщениям (chosen-message attack). В этом случае злоумышленник может получить подписи к сообщениям, выбранным им лично (как в случае с нотариусом). Выбор этих сообщений может зависеть от ранее полученных подписей.

Различают несколько уровней по степени раскрытия схемы подписей.

1. Эпизодическое подделывание (Existential forgery). Злоумышленник подделывает подпись одного сообщения, не обязательно им выбранного. Сообщение это может быть или случайным, или не имеющим смысла.

2. Выборочное подделывание (Selective forgery). Злоумышленник подделывает подпись некоторого сообщения, по своему выбору.

3. Универсальное подделывание (Universal forgery). Злоумышленник может подделывать подписи к любому сообщению, но без знания секретного ключа подписи. Он может найти функционально эквивалентный алгоритм подписи.

4. Полное раскрытие схемы подписи (Total break).

Злоумышленник может вычислить секретный ключ подписывающего.

2.8. Схема цифровой подписи Эль Гамаля

В 1985 г. Эль Гамаль (El Gamal) предложил схему цифровой подписи, основанную на сложности решения задачи дискретного логарифмирования [El Gamal T.A., A Public Key cryptosystems and a Signature Scheme based on Discrete Logarithms, IEEE Tr. Inform. Theory, v. 31, № 4, 1985]. В той же работе содержится и описание криптосистемы открытого шифрования.

Приведем сначала классическое описание схемы, чтобы потом увидеть, в каком направлении происходили ее модификации.

Для функционирования схемы выбирается большое простое число p и примитивный корень $\lambda \in \overline{2, p-1}$ по модулю p. Числа эти несекретные и должны быть известны как подписывающему сообщение (пользователь A), так и проверяющим подпись под этим сообщением (пользователь B).

Секретная информация подписывающего пользователя A состоит из двух частей:

1. $x_A \in \overline{2, p-1}$ – **долговременный секретный ключ подписи**, выбирается случайно из указанного интервала и хранится в секрете.

2. $k_A \in \overline{1, p-1}$ – **разовый секретный ключ подписи**, конкретного сообщения, НОД $[k_a, p-1] = 1$.

Открытая информация подписывающего тоже состоит из двух частей.

1. $Y_A = \alpha^{X_A} \bmod p$ – **открытый ключ подписи**, вычисляется и сообщается всем проверяющим подпись пользователя A.

2. $r = \alpha^{k_A} \bmod p$ – составляет правую из двух частей (r,s) подписи.

Подписываемое сообщение должно быть представлено числом M из интервала [0,p-1]. (Далее в качестве M будет также рассматриваться значение функции хэширования H от сообщения.)

Процедура подписи сообщения M следующая.

1. Пользователь A выбирает случайное число k_A из интервала $1, \overline{p-1}$ таким образом, чтобы выполнялось условие НОД $[K_a, p-1] = 1$.

2. Вычисляет $K_a^{-1} \bmod(p-1)$, то есть число, удовлетворяющее сравнению $K_a \cdot x \equiv 1 \bmod(p-1)$. Именно для разрешимости этого сравнения при выборе k_A наложено ограничение его временной простоты C(p-l).

3. Вычисляет первую часть подписи $r = \alpha^{k_A} \bmod p$.

4. Вычисляет вторую часть подписи по формуле

$S = K_A^{-1}\left[M - r \cdot x_A\right] \bmod(p-1)$.

На этом процедура выработки подписи (r,s) к сообщению M заканчивается, и эти данные сообщаются всем проверяющим подпись.

Процедура проверки данных $\overline{M}$, $\tilde{r}$, $\tilde{s}$ такова (черта над буквами поставлена потому, что поступившие на проверку данные могут не совпадать с оригинальными).

1. По полученным данным $\overline{M}, \tilde{r}, \tilde{s}$ и имеющемуся у проверяющего открытому ключу подписывающего вычисляются величины $\alpha^{\overline{M}} \bmod p$ и $y_A^{\bar{r}} \cdot r^{\bar{s}} \bmod p$.

2. В случае выполнения равенства $\alpha^{\overline{M}} \bmod p = y_A^{\bar{r}} \cdot r^{\bar{s}} \bmod p$ считается, что $\overline{M} = M$, $\bar{r} = r$, $\bar{s} = s$ и подпись верная.

Для объяснения схемы подписи рассмотрим проверяемое сравнение $\alpha^{\overline{M}} \bmod p \equiv y_A^{\bar{r}} \cdot r^{\bar{s}} \bmod p$, решением которого и являются числа r и s. После подстановки значений Y_a и r оно примет вид

$$\alpha^M \equiv \alpha^{x_A \cdot r} \cdot \alpha^{k_A \cdot s} \bmod p .$$

По свойствам сравнений последнее сравнение эквивалентно сравнению по модулю (p-l) вида

$$M \equiv x_A \cdot r + k_A \cdot s (\bmod (p-1)).$$

Именно из этого сравнения подписывающий вычисляет величину $s \equiv [M - x_A \cdot r] k^{-1} \bmod (p-1)$.

Стойкость метода зависит во многом от сложности вычисления дискретных алгоритмов в GF(p). Так, с этой задачей злоумышленник сталкивается при определении секретного ключа x_A из наблюдаемых данных M, r и s, так как из известного ему соотношения $\alpha^M \equiv y_A^r \cdot r^s (\bmod p)$ следует $y^r \equiv \alpha^M \cdot \left[r^s\right]^{-1}$ или $\left[a^r\right]^{X_A} \equiv \alpha^M \cdot \left[r^s\right]^{-1} \bmod p$. В случае если ему удастся найти x_A, то это будет полное вскрытие схемы цифровой подписи и злоумышленник может выдавать себя за пользователя А. Отсюда следует, что нужно очень осторожно выбирать простое число p, чтобы не выбрать такое, для которого разработаны противником или известны быстрые методы вычисления дискретных логарифмов. (Например, в силу метода Полига-Хеллмана, когда (p-l) имеет только «маленькие» делители [35].)

В случае если злоумышленник получил в свое распоряжение N наборов данных $[M_i, r_i, s_i]$, $(i = \overline{1, N})$, то он сталкивается с решением системы из N сравнений $M_i \equiv x_A \cdot r_i + k_{A,i} \cdot s_i \bmod (p-1)$ от N+1 неизвестного x_A, $k_{A,1}, \ldots, k_{A,N}$, которая является неопределенной и имеет экспоненциальное число решений. Но в случае повторения какого-нибудь разового ключа k_A долговременный ключ x_A может быть определен из этой системы однозначно.

Может оказаться более простым, чем нахождение дискретных логарифмов, способ нахождения злоумышленником пары (r,s) из соотношения $\alpha^M \equiv y_A^r \cdot r^s \bmod p$, но пока такой способ не известен.

В [12] показано, что схема Эль Гамаля допускает возможность выработки ложных сообщений и подписей к ним, удовлетворяющих соотношению $\alpha^M \equiv y_A^r \cdot r^s \bmod p$ (Existential forgery), но получаемые при этом сообщения являются неосмысленными. Показано, что если злоумышленник имеет истинную тройку $[M,r,s]$, то он может получить подпись $(\bar{r},\bar{s})$ из соотношений

$$r \equiv r^A \alpha^B y^C \bmod p\ ,\ \bar{s} \equiv s\bar{r}/(Ar-(s) \bmod (p-1)$$

к сообщению вида

$$M \equiv r(A_m - B_s)/(A_r - C_s) \bmod (p-1)\ ,$$

где A, B, C выбираются произвольно, но так, чтобы существовал обратный эксперимент к $(A_r - C_s)$. Правильные подписи могут быть получены и без знания истинных данных $[M,r,s]$, что следует при A = 0. В этом случае $[M,r,s]$ имеют вид

$$\bar{r} = \alpha^B y^C \bmod p\ ,\bar{s} = -\bar{r}/c \bmod (p-1)\ ,\ \overline{M} = -\bar{r}B/c \bmod (p-1)\ .$$

Желание устранить способ подделки приводит к тому, что схема Эль Гамаля используется только с функцией хэширования открытого сообщения.

Квантовые компьютеры, квантовые атаки

Большую угрозу стойкости схем типа Эль Гамаля или RSA представляет разработка и создание так называемых квантовых компьютеров. Для квантовых компьютеров уже разработаны методы нахождения порядка элементов для ряда известных конечных групп. Один из таких полиномиальных алгоритмов предложил в 1997 г. П. Шор. Идея алгоритма в создании регистра на квантовых элементах, разбитого на две части. В одной части записываются числа a = 0, 1, 2, 3, …, а во второй части – значение показательной функции от этого числа $f(a)\text{-}x^a$. Первая часть регистра преобразуется с использованием быстрого преобразования Фурье. Наблюдая

за обеими частями регистра, находят порядок элемента x. Применение алгоритма Шора сдерживается отсутствием квантового компьютера с достаточно большим размером регистра. В 2007 г. анонсирован регистр с 28 разрядами, против необходимых сегодня 1000–2000. В настоящее время идет интенсивное изучение групп, где алгоритм Шора не имеет полиномиальную (экспоненциальную) сложность [см., например, ePrint 2006/246].

2.9. Управление ключами

Управление ключами (key management) представляет собой процесс, посредством которого ключевой материал, используемый в симметричных или асимметричных криптосистемах, предоставляется в распоряжение пользователей и подвергается обработке определенными процедурами безопасности, вплоть до его уничтожения. Управление ключами связано со следующими процедурами:

- генерация ключей (key generation);
- распределение ключей (key distribution);
- хранение ключей (key storage);
- уничтожение ключей (key deiition);
- регистрация пользователей (user registration).

Рассмотрим последовательно некоторые из них.

Генерация. Стойкость криптографических алгоритмов, описание которых согласно принципу Керкхоффса может быть известно и злоумышленнику, основывается на безопасности ключей. Поэтому, как подчеркивается в [42], если вы используете слабый алгоритм для генерации ключей, то и вся система становится небезопасной. Противнику достаточно анализировать не саму криптосистему, а алгоритм генерации ключей.

Первое, от чего стоит предостеречь, это сокращение возможных значений ключей путем выбора в качестве них, например, осмоленных, легко запоминающихся слов и выражений, сокращения алфавита знаков ключей и т.д. Это приводит к возможности полного перебора ключей с помощью

современной вычислительной техники, производительность которой постоянно растет (см. www.top500.org). Конечно, все зависит от модели предполагаемого злоумышленника и необходимого уровня стойкости системы. Можно использовать метод преобразования (key crunching) легко запоминающихся фраз в псевдослучайный ключ с помощью каких-нибудь процедур (хэш-функций).

Хорошие ключи – это случайные равновероятные строки в некотором алфавите, поэтому для их получения используются различные генераторы случайных и псевдослучайных последовательностей. Это целая тема в криптографии, требующая отдельного рассмотрения [23, 40, 42]. В качестве примера можно привести метод генерации ключей, содержащийся в американском стандарте **ANSI X.9.17.**

Если обозначить через $E_k(M)$ преобразование зашифрования блока открытого текста M на ключе k (в стандарте $E_k(M) = DES_k(M)$, то генерация ключей $R_i(i = 0,1,...)$ определяется по следующему правилу:

$R_i = E_k\left[E_k(T) XOR V_i\right]$, i = 0, 1,…,

$V_{i+1} = E\left[E_k(T) XOR R_i\right] V_{i+1} = E\left[E_k(T) XOR R_i\right]$, i = 0, 1,…,

где V_0 – начальный секретный вектор, а T_i – время выработки i-го ключа в двоичном представлении.

Изложенный способ генерации ключей подходит для симметричных криптосистем. Генерация ключей асимметричных криптосистем сложнее, так как они должны удовлетворять ряду дополнительных требований.

Следует также учитывать, что некоторые криптоалгоритмы могут иметь «слабые» ключи, которые менее безопасны, чем другие. Например, алгоритм DES имеет 16 таких ключей. Поэтому нужно при генерации ключей предусмотреть контроль за появлением слабых ключей.

Распределение ключей. Это центральная проблема при управлении ключами.

В случае симметричных криптосистем эта проблема решается следующими способами. Наиболее известным способом является создание секретного канала передачи ключей удаленным пользователям с помощью курьерской службы. Недостатками такого способа являются его высокая цена, невысокая скорость. Особенно это проявляется при большом числе пользователей. Например, число различных парных ключей для секретной связи N пользователей равно N(N-1)/2. К тому же передачу секретной информации с помощью курьера трудно назвать полностью безопасной. Всегда остаются сомнения, не были ли ключи украдены (списаны) или подменены во время доставки. Можно использовать при этом некоторые аппаратные средства для исключения доступа курьера к данным.

Используется иногда способ, когда ключ разбивается на несколько частей (или образуется из нескольких частей), которые доставляются пользователям по различным каналам: курьером, по телефону, по почте и другим образом. Противнику для получения ключа необходимо контролировать все эти каналы.

Существует целый ряд способов распределения ключей, использующих **центр(ы) доверия** (или **центр распределения ключей** – ЦРК). Это дает возможность сократить число необходимых ключей, так как не каждому из пользователей сети надо секретно взаимодействовать с каждым, что приводит к необходимости установления секретной связи между каждым пользователем и центром. Однако центр доверия становится основной мишенью для злоумышленника, так как имеет доступ ко всем ключам пользователей системы. Иногда на практике трудно найти третью, незаинтересованную, сторону, пользующуюся доверием всех пользователей. Недостатком включения центров доверия является также то, что возникают колоссальные нагрузки на эти центры по генерации ключей и взаимодействию со многими пользователями одновременно, что снижает надежность и оперативность секретной связи.

Приведем пример протокола распределения ключей с использованием симметричной криптосистемы и центра распределения ключей из работы [56].

1. ЦРК вырабатывает и передает по защищенному каналу долговременные ключи каждому пользователю системы для связи с ЦРК.

2. Когда пользователь A желает установить секретную связь с пользователем B, он посылает запрос в ЦРК с требованием секретного сеансового ключа для связи с пользователем B.

3. ЦРК вырабатывает сеансовый ключ K_{AB}, зашифровывает его два раза на долговременных ключах K_A и K_B вместе с идентифицирующей пользователей информацией и случайно выбранными добавками, чтобы исключить совпадения обоих открытых текстов и рассылает соответствующие шифртексты пользователям A и B.

4. Пользователи A и B расшифровывают поступившие шифртексты и получают сеансовый ключ.

Для полноты приведем **трехэтапный протокол Шамира** с коммутативным преобразованием E_k зашифрования, для которого

$E_{k_1}\left[E_{k_2}(M)\right] = E_{k_2}\left[E_{k_1}(M)\right]$ при любом открытом тексте M.

При этом предполагается, что канал связи не позволяет осуществить злоумышленнику имитацию или подмену сообщения. В качестве шифрсистемы для реализации этого протокола Месси (Massey), и независимо Омура (Omura D.), предложил систему, где преобразование зашифрования и расшифрования имело вид

$$M = E_d(C) = C^d (\mathrm{mod}\, p),$$

где p – простое число, e и d – секретные ключи зашифрования и расшифрования, положительные целые числа, такие, что $e,d \in 0,\overline{p-1}$, НОД(e, p-1)=1, $de \equiv \mathrm{mod}(p-1)$.

Итак, если пользователь *A* желает послать секретную информацию *M* (ключ или открытое сообщение) пользователю *B*, то необходимо сделать следующее:

1. Пользователь *A* посылает пользователю *B* сообщение

C_1 = M^(e_A) mod p.

Это сообщение ничего не дает для *B*, так как он не знает ключ расшифрования d_A пользователя *A*. Для нахождения d_A злоумышленник, наблюдающий C_1, должен в лучшем для него случае определить e_A, решая задачу нахождения дискретного логарифма в предположении, что ему известен открытый текст *M.* Отсюда дополнительные требования к числу *p-1* – оно, например, должно содержать большой простой делитель.

2. Пользователь *B* вычисляет $C_2 = C_1^{e_B} = M^{e_A e_B}$ и посылает C_2 пользователю *A*. На этом этапе злоумышленник может попытаться найти e_B как $e_B = \log_{C_1} c_2$.

3. Наконец, пользователь А вычисляет $C_3 = C_2^{d_A} = M^{e_A e_B e_D} = \left[M^{e_A d_A}\right]^{e_B} = M^{e_B}$ (по теореме Ферма) и посылает C_3 пользователю $B\left[d_A = \log_{c_2} C_3\right]$.

4. Пользователь *B* находит секретную информацию *M* с помощью возведения $c_3^{d_B} = M^{e_B e_D} = M$. Заметим, что, получив сообщение *M,* пользователь *B* также может попытаться найти секретный ключ e_A из $e_B = \log_M C_1$, но это для него вычислительно трудная задача. Аналогично *A* не может найти $e_B = \log_M C_3$.

Если канал связи не обладает приведенными выше свойствами, обеспечивающими подлинность передаваемых сообщений, то злоумышленник *G* может выдать себя за пользователя *B* и послать вместо C_2 сообщение $\overline{C_2} = C_1^{e_G}$ и определить далее сообщение M из сообщения $\overline{C_3} = C_2^{d_A} = M^{e_G}$.

Распределение ключей с помощью криптосистем с открытым ключом кажется несколько простым в силу возможности использования незащищенных каналов и отсутствия необходимости в секретности при

хранении и передаче открытых ключей E_A и E_B пользователей A и B. Но распределение открытых ключей также требует подтверждения их подлинности (аутентификации). Для этого также может использоваться центр доверия, где пользователи регистрируют свои открытые ключи, но сам центр при этом не становится активным участником протокола, и нагрузка на него невысока.

Протокол обмена сеансовых ключей между A и B может быть следующим.

1. Пользователь B получает от центра доверия или непосредственно от пользователя B его открытый ключ E_B, генерирует сеансовый ключ K_{AB}, зашифровывает его с использованием E_B и посылает по открытому каналу $E_B[K_{AB}]$ пользователю B.

2. Пользователь B расшифровывает полученное от A сообщение $E_B[K_{AB}]$ с помощью своего секретного ключа D_B и получает сеансовый ключ K_{AB}.

Ключ K_{AB} может быть ключом для симметричной криптосистемы, скорость работы которой выше, чем у криптосистем с открытым ключом **(гибридные системы).**

В этом протоколе открытый ключ должен быть защищен от подмены, иначе злоумышленник может заменить его на свой ключ $E_G[K_{AB}]$, и пользователь A передает сеансовый ключ K_{AB} в сообщении ему **(man-in-the-middle attack).** В работе [23] предложен протокол, не позволяющий противнику, контролирующему открытый канал между A и B, получить доступ к ключу K_{AB}.

1. Пользователи A и B обмениваются своими открытыми ключами E_A и E_B.

2. Пользователь A зашифровывает свое сообщение K_{AB} на открытом ключе E_B, $E_B[K_{AB}]$. Посылает половину зашифрованного сообщения $E_B[K_{AB}]$ (1) пользователю B.

3. Пользователь B зашифровывает свое сообщение K_{BA} на ключе E_A и посылает половину $E_A[K_{BA}]$ (1) пользователю A.

4. A посылает другую половину $E_B[K_{AB}]$ (2) пользователю B.

5. Пользователь B расшифровывает обе части $E_B[K_{AB}]$ (1) и $E_B[K_{BA}]$ (2) на своем секретном ключе D_B. В случае получения осмысленного значения K_{AB} он посылает другую половину своего зашифрованного сообщения $E_A[K_{BA}]$ (2) пользователю A.

6. A расшифровывает обе части $E_A[K_{BA}]$ (1) и $E_A[K_{BA}]$ (2) на своем секретном ключе D_A.

Пользователи A и B не могут прочитать направленные им сообщения соответственно до шага 5 и 6. Злоумышленник может подставить свои открытые ключи на шаге 1. Но теперь, получив половину сообщения $E_G[K_{AB}]$ (1) на шаге 2, он не может в силу этого получить K_{AB} с помощью своего секретного ключа D_G и перешифровать его с помощью открытого ключа E_B. Он вынужден создавать новое сообщение $\overline{K}_{AB}$, шифровать его с помощью E_B и посылать половину его B. Аналогично обстоит для противника дело с половиной сообщения от B. Наконец, он получает вторые половины реальных сообщений на шаге 4 и 5, но это слишком поздно, чтобы заменить новые сообщения на те, которые он желает.

Ранее был рассмотрен протокол ключевого обмена Диффи и Хеллмана, требующий также подтверждения подлинности передаваемых сообщений. Для него также справедливо все сказанное выше по поводу возможной подмены злоумышленником этих сообщений и противодействия этому.

В целом ряде работ [17, 23, 39] содержатся протоколы обмена ключами с подтверждением подлинности отправителя.

2.10. Криптографические хэш-функции. Аутентификация

Термин «хэш-функция» (или функция хэширования) возник в теории сложности вычислений, где он обозначал функцию, которая сжимает строку

чисел произвольного размера в строку чисел фиксированного размера. Это понятие использовалось в алгоритмах поиска данных по значениям хэш-функции от них. Рассмотрим это понятие применительно к криптографии.

Криптографические хэш-функции подразделяются на два класса: с ключом и без ключа. Значение хэш-функции с ключом может вычислить лишь тот, кто знает некоторый секретный параметр – ключ. Часто в литературе они называются **кодами аутентификации сообщений** (MAC – Message Authentication Code), видимо, по широко известному примеру такой функции, описанному в стандарте FIPS PUB 113-1985. Российским аналогом этого американского стандарта может служить режим имитовставки в стандарте ГОСТ 28147-89.

Хэш-функцией с ключом (зависящей от ключа) называется функция, имеющая следующие свойства.

1. Описание функции H(k,x) должно быть открыто, а секретная информация должна содержаться только в выборе ключа k (правило Керкхоффса).

2. Аргумент x функции H(k,x) может быть строкой чисел произвольной длины, а значение функции должно быть строкой чисел фиксированной длины.

3. При любых данных k и x вычисление H(k,x) должно быть быстрым (за полиномиальное время).

4. По любому данному x должно быть трудно угадать значение H(k,x) с вероятностью большей, чем $(1/2)^n$, где n – число бит в выходной строке. Должно быть трудно определить ключ k даже по большому числу известных пар $\{x_i, H(k,x_i)\}$ при выбранных криптоаналитиком входах (adaptive chosen text attack) или вычислить по этой информации H(k,x') для любого $x' \neq x_i$.

В стандарте FIPS PUB 113-1985 хэш-функция с ключом построена на основе блочного стандарта шифрования DES с ключом из 56 бит. В качестве аргумента функции выступает открытый текст и $M_1, M_2, ..., M_n$, разбитый на 64 битовых блока, которые преобразуются по правилу (CFB)

$C_i = DES_k(M_i XOR C_{i-1})$, $i = 1, n$; $C_0 = \theta$.

В качестве значения функции берется необходимое число бит (от 16 до 64, кратное 8) из последнего блока C_n. Точно так же может использоваться и любая другая криптографическая стойкая блочная система.

Аналогично преобразуется открытый текст по ГОСТ 28147-89 в режиме выработки имитовставки. При этом преобразование 64-битовых блоков происходит с помощью 16 циклов алгоритма зашифрования в режиме простой замены.

В работе [103] предложена хэш-функция на основе поточной криптосистемы, двоичная гамма которой управляет поступлением битов открытого текста как входного воздействия на два неавтономных регистра сдвига. Значение функции определяется конечным состоянием регистров сдвига. Это очень быстрый способ вычисления хэш-функции.

Хэш-функция без ключа иногда называется кодом определения манипуляции (MDC – Manipulation Detection Code). Такие функции в свою очередь подразделяются на два подкласса: слабые (weak) односторонние хэш-функции и сильные (strong) односторонние хэш-функции. Дадим их определение.

Однонаправленной слабой хэш-функцией называется функция H(x), удовлетворяющая следующим условиям.

1. Аргумент x функции может быть строкой чисел произвольного размера.

2. Значение функции H(x) представляет собой строку чисел фиксированного размера.

3. Значение функции H(x) легко вычисляется (в полиномиальное время).

4. Почти для всех y вычислительно невозможно найти такое x, что H(x) = y .

5. Для любого фиксированного x вычислительно невозможно найти другое x ≠ x', такое, что H(x') = H(x). Такое событие называется **коллизией (collision).**

Свойства (3) и (4) утверждают, что H является однонаправленной функцией. Последнее свойство (5) сильнее, чем (4). Заметим также, что по свойствам (1) и (2) коллизии должны существовать, однако свойство (5) требует, чтобы найти их было вычислительно невозможно.

Однонаправленной сильной хэш-функцией называется функция H(x), удовлетворяющая свойствам (1)–(4) предыдущего определения, а свойство (5) имеет вид:

5`. Вычислительно невозможно найти любую пару аргументов функции $x \neq x'$, такую, что H(x') = H(x).

Оба определения предполагают, что описание функции H(x) открыто (правило Керкхоффса).

Хотя последние два определения похожи друг на друга, но с вычислительной точки зрения они разные. Для пояснения этого приведем пример из работы [12]. Пусть H является хэш-функцией с n-битовым выходом. И пусть криптоаналитик желает найти для фиксированного сообщения M другое сообщение M', такое, что H(M) = H(M'). Предполагая, что $m = 2^{-n}$ возможных выходов функции H появляются случайно при опробовании входа, то любой кандидат м' имеет вероятность только 2^{-n} дать тот же выход H(M). Если опробовано N кандидатов, то вероятность того, что хотя бы один вариант даст равенство H(M) = H(M'), есть $1-(2-2^{-n})^N \approx 2^{-n}N$.

При n = 80, $N = 2^{60} = 10^{18}$, коллизия найдется с вероятностью 10^{16}. То есть H(x) безопасная хэш-функция в смысле свойства (5). Опробование N вариантов входа функции H можно сравнить со случайным бросанием дробинок в 2^n = m ящиков, соответствующих возможным значениям функции H. Общее число способов размещения N дробинок в m ящиков равно $(2^n)^N = m^N$ (каждая из m дробинок может попасть в любой из N ящиков). Чтобы ни одна пара дробинок не попала в один ящик, первая может быть помещена в любой ящик, вторая может попасть в любой из (m-1) оставшихся ящиков, третья – в любой из (m-2) ящиков, … и так далее. Вероятность отсутствия коллизий есть

m(m-1)... $(m-N+1)/m^N$, а вероятность, по крайней мере, одной коллизии есть

$$P(m,N)=1\left[(m-1)...(m-N+1)/m^N\right]=1-\left[(1-1/m)...\left[1-\frac{(N-1)}{m}\right]\right].$$

Можно показать, что при m >N > (2 с m)$^{1/2}$ вероятность

P (m,N) > $1-e^{\frac{N(N-1)}{2m}}\approx 1-e^{-c}$. Откуда при $N>\left[2(\ln 2)m\right]^{1/2}=1,7m^{1/n}$ вероятность p(m,N)>1/2.

При n = 80 вычисление $N=1,7\cdot 2^{40}<2\cdot 10^{12}$ вариантов хода даст, по крайней мере, одну коллизию с вероятностью > 1/2 и H не является безопасной в смысле свойства (5').

Приведенный пример ценен еще и тем, что дает метод атаки на функцию хэширования, атаки, связанной с **задачей о днях рождения (birthday attack),** то есть вычислением вероятности того, что два члена из группы людей имеют один и тот же день рождения.

Применение криптографических хэш-функций исключительно разнообразно.

Одно из основных применений они находят в реализациях схем цифровой подписи. Впервые идея такого применения была высказана в работе Девиса и Прайса (Davies, Price). Это значительно ускоряет процесс подписи документа, когда его данные сжимаются в короткий отрезок с помощью односторонней хэш-функции (как правило, без ключа) и только к этому отрезку применяется преобразование цифровой подписи. К тому же, сам процесс подписи и проверки подписи значительно медленнее, чем процесс симметричного шифрования или хэширования. Размеры входа алгоритма цифровой подписи и выхода алгоритма вычисления функции хэширования должны быть, естественно, согласованы. Кроме того, сами эти алгоритмы также должны быть проанализированы совместно, чтобы не ослабить друг друга, как это было со схемой цифровой подписи RSA и алгоритмом хэширования, также использующим модульное экспоненцирование. Так, к существующим стандартам цифровой подписи

DSS, ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001 разработаны соответствующие стандарты на функцию хэширования SHS (Secure Hash Standard) и ГОСТ Р 34.11-94. В отличие от схем цифровой подписи, эти функции хэширования SHS и ГОСТ отличаются друг от друга существенным образом, поэтому будут рассмотрены далее отдельно.

Важно отметить и применение хэш-функций в парольных системах и при защите данных своего компьютера от искажений. Об этом существует многочисленная литература [23, 39, 45].

Обеспечение секретности предполагает много затрат, в том числе и времени, на зашифрование и расшифрование, поэтому в первую очередь рассмотрим **аутентификацию без шифрования.**

Если используется хэш-функция с ключом (MAC), то простейший способ защиты подлинности сообщения – это вычисление значений этой функции и добавление его к сообщению. Подлинность информации здесь зависит теперь от секретности и подлинности ключа аутентификации и может быть проверена каждым, кто знает этот ключ. Значения хэш-функций на практике часто передаются по медленным, но обеспечивающим аутентичность и/или секретность каналам, таким как телефонная линия или почтовая связь (с узнаванием голоса или с обычной рукописной подписью). Участники связи должны полностью доверять друг другу. Даже если ключи не попадут к какому-нибудь стороннему злоумышленнику, все равно есть возможность кому-то из законных участников отказаться от посланного сообщения или заявить, что не получал другого сообщения. Лучший способ преодолеть это – использование цифровой подписи.

Преимущество использования для аутентификации хэш-функций без ключа в том, что нет необходимости в управлении ключами, но должен быть канал, защищающий подлинность значений хэш-функций. Так же, как и выше, все участники связи должны доверять друг другу.

Есть отличие в выборе слабой или сильной хэш-функции в зависимости от модели предполагаемого противника. Если это один из законных

пользователей, то он может попытаться найти пару сообщений M ≠ M' и H(M) = H(M'), послать сначала M и H(M), но позднее извлекать преимущества из знания м'. Поэтому, чтобы избежать этой атаки, хэш-функция должна быть сильной. Если злоумышленник не является законным пользователем системы связи и он может только наблюдать значения M и H(M), то его задача найти M' такое, что H(M) = H(M'). Чтобы противостоять этой атаке, достаточно слабой хэш-функции. Разработка слабых хэш-функций проще, да и значений таких хэш-функций может быть меньше.

В случае использования MAC с обеспечением секретности сообщений должны быть две независимые процедуры управления ключами шифрования и аутентификации. Заманчиво использовать один и тот же ключ дважды, но этого делать нельзя в целях безопасности, а подчас и невозможно из-за разного времени действия ключей, времени хранения после использования и др. вопросов.

Наиболее простой способ обеспечения подлинности и секретности – это вычисление $MAC_{K_A}(M)$, добавление его к открытому тексту и зашифрование всей информации:

$E_{K_Ш}(MAC_{K_A}(M))$.

Можно не зашифровывать $MAC_{K_A}(M)$. Иногда вычисляют MAC от зашифрованного сообщения $MAC_{K_A}(E_K(M))$, что дает возможность проверять целостность сообщения без знания открытого текста и ключа шифрования. Но лучше все же защищать подлинность открытого текста, а не шифрованного.

Наиболее простой способ защиты подлинности сообщения с обеспечением секретности и применением хэш-функции без ключа – это вычисление MDC от сообщения M и шифрование его значения вместе с открытым текстом: $E_E(M, MDC(M))$. Но можно $MDC(M)$ и не шифровать в силу того, то хэш-функция должна по определению быть однонаправленной функцией.

Раздел 3. СТАНДАРТИЗАЦИЯ КРИПТОГРАФИЧЕСКИХ МЕТОДОВ

3.1. Организации, разрабатывающие стандарты по криптологии

В мире существует достаточно много организаций по стандартизации методов защиты информации или их использования в информационных технологиях.

Разработанные ими стандарты находятся в большой взаимосвязи друг с другом, образуя целые системы стандартов разного уровня развития. Разработка стандартов является сложным процессом, требующим больших затрат времени, средств и усилий большого числа экспертов. Кроме того, стандарты необходимо периодически пересматривать в свете достижений науки и техники, методов криптоанализа, изменений в потребностях практики.

США

ANSI (American National Standards Institute) – Американский национальный институт стандартов. Занимается вопросами по электронному обмену данных (EDI - Electronic Date Interchange), банковскому делу, безопасности взаимосвязи открытых систем (OSI – Open systems interconnection). Все стандарты этой организации имеют номера, начинающиеся с буквы X.

IEEE (Institute of Electrical and Electronic Engineers) – Институт инженеров по электронике и радиоэлектронике. Занимается безопасностью локальных сетей связи (LAN – Loca/AreaMetwork) и другими вопросами.

NIST (National Institute for Standards and Technology) – Национальный институт по стандартам и вычислительной технике. До 1988 г. он назывался NBS (National Bureau of Standards).

Национальное бюро стандартов. Это подразделение Министерства торговли США. В 1987 г. Конгресс принял Закон о защите ЭВМ (computer security Act), согласно которому ответственность за стандарты, касающиеся

ЭВМ и соответствующих систем связи, возлагались на NBS-NIST. Институт выпустил целый ряд стандартов (Federal Information Processing Standards – FIPS), включая известный DES (Data Encryption Standard).

US/Pod (Department of Defense) – Министерство обороны США также занимается разработкой своих стандартов.

NCSC (US National Computer Security Center) – Национальный институт компьютерной безопасности США (отделение NSA). Под эгидой этой организации выпущены стандарты по оценке безопасности компьютерных систем:

NSA (US National Security Agency) – Агентство национальной безопасности США (АНБ). Создано в 1940 г. в соответствии с секретной директивой президента Трумэна, по которой в обязанности АНБ вменялись задачи по «разведке в области средств связи» (т.е. перехват, дешифрование криптограмм иностранных государств) и «обеспечению безопасности национальных приемопередающих систем». АНБ находится на передовой научных поисков и разработок в области электронно-вычислительной техники, криптологии. АНБ субсидирует многочисленные исследования по интересующей его тематике, проводит активную политику в разработке и принятии стандартов. Этому способствует и меморандум о взаимопонимании (Memorandum of Understanding) между NSA и NIST. Именно NSA влияло на выбор параметров для DES.

ABA (American Bankers Assotiation) – Американская ассоциация банкиров, разрабатывает стандарты использования криптографических методов применительно к банковской практике.

Европа

CEM/CEHELEC (European Comrltte for Standardization) – Европейский комитет по стандартизации (Аналог ISO и IES).

ETSI (European Telecommunications Standard Institute) – Европейский институт стандартов в телекоммуникациях.

ITAEGV (Information Technology Advisory Expert Group for Information security). Разрабатывает требования для будущих стандартов.

EUOS (European Workshop for Open Systems) – Европейский симпозиум по открытым системам.

ECMA (European Computer Manufactures Association) – Европейская ассоциация производителей компьютерной техники. Работает над безопасностью систем баз данных, безопасностью открытых систем.

UM/EDIFACT Security Joint Working Group (United Nations/Economic and Social Council) – объединенная рабочая группа по безопасности электронного обмена данными (EDI -Electronic Date Interchange).

CCITT (International Telegraphy and Telephone Consultative Committee) – Международный консультативный комитет по телеграфии и телефонии. Занимается вопросами обмена данными, является частью международного союза телекоммуникации (International Telecommunication Union – ITU) в ООН. Стандарты, разработанные комитетом, носят рекомендательный характер и пересматриваются каждые 4 года. Вопросы безопасности изучаются группой SG VII (Data Networks).

IEC (International Electronical Commission) – Международная электротехническая комиссия.

ISO (International Standards Organization) – Международная организация по стандартизации. Работа внутри ISO ведется техническими комитетами (Technical committee – TC), которые учреждают подкомитеты (SC). Подкомитеты в свою очередь разбиты на рабочие группы (Working Groups – WG), некоторые из которых – специальные рабочие группы (Special working group – swg) – находятся на уровне подкомитетов. Документ проходит следующие стадии разработки.

- NP (Hew Work item proposal) – Предложение по новой теме работы;
- WD (Working Draft) – Проект рабочей группы;
- CD (Committee Draft) **–** Проект комитета;
- DIS (Draft International Standard) – Проект международного стандарта;

- ISC (International Standard) – Международный Стандарт.

Прохождение документа является длительным процессом. Стадия, на которой находится проект стандарта, отражается в его заголовке.

ISO и IES объединяют свою работу в рамках Объединенного технического комитета YTC1 – «Информационная технология». Этот комитет подразделяется на несколько подкомитетов (SC) и рабочих групп (WG), среди которых можно выделить SC610SI (Lower Layers), SC17 (Identification Cards), SC18 (Text and Office Systems), SC21 (OSI Architecture, Management, and Upper Layers), SC22 (Languages) и SC27 (Security Techniques).

Работа в SC27 сосредоточена в рабочих группах, среди которых: UG1 (Requirements, Security Services and Guidelines), VG2 (Techniques and Mechanisms), MG3 (Security Evaluation Criteria).

Особо можно выделить технический комитет ISO/TC68 – «Банковское дело и связанные с этим финансовые услуги» (Banking and Related Financial Services).

Стандарты Интернет (RFC) также имеют в своем составе большое число криптографических стандартов. Усилиями российских разработчиков прошли процедуру стандартизации и российские криптографические алгоритмы.

3.2. Первые варианты стандартов цифровой подписи США и России

Американский стандарт цифровой подписи DSS (Digital Signature Standard – FIPS 186) и российский его аналог ГОСТ 50 34.10-94 приняты в 1994 г. и являются фактически модернизациями схемы цифровой подписи Эль Гамаля. Рассмотрим поэтапно внесенные изменения.

Хотя к настоящему времени были приняты и используются другие стандарты (FIPS 186-3, ГОСТ Р 34.10-2001), но актуальность их не потеряна. Во-первых, надо проверять сейчас подписи документов, ранее полученные по этим стандартам. Во-вторых, эти стандарты лежат в основе новых стандартов, а отличие заключается лишь в используемых алгебраических группах.

Быстродействие процедуры подписи и проверки подписи в схеме Эль Гамаля определяется числом операций возведения в степень, которая является наиболее трудоемкой. В классическом варианте схемы необходимо выполнить соответственно одно и три возведения в степень по модулю простого числа p. Схему можно модернизировать так, что при проверке вместо трех возведений будут выполняться два, используя при этом данное проверяющему число $r = \alpha^k \bmod p$, полученное при подписи.

Для этого в сравнении $M = xr + ks \bmod(p-1)$ нужно поменять один знак и рассмотреть сравнение $M \equiv -x \cdot r + ks \bmod(p-1)$, из которого следует, что при подписывании величина S будет вычисляться по формуле

$$s \equiv k^{-1}(M + x \cdot r) \bmod(p-1),$$

а сравнение будет определять проверочное соотношение

$$r \equiv \alpha^{Ms^{-1}} \cdot y^{rs^{-1}} \bmod p,$$

для вычисления правой части которого требуются только два возведения.

Однако теперь появилось новое дополнительное требование: необходимо существование обратного элемента по модулю (p-1), т.е. чтобы S и (p-1) были бы взаимно просты. Конечно, это можно учитывать в процедуре подписи и выбирать k так, чтобы, помимо требования (k, p-1) = 1, выполнялось $(k^{-1}(M + x \cdot r), p-1) = 1$. Но это приводило бы к дополнительным пробам и увеличивало, хотя и незначительно, время подписи.

Как уже отмечалось выше, для усложнения методов логарифмирования и увеличения стойкости схемы число (p-1) должно иметь хотя бы один большой простой делитель. Обозначим его через q, а сравнение $M \equiv -x \cdot r + ks$ рассмотрим именно по модулю q. Тогда требование существования обратных элементов k и s отпадает, так как они всегда существуют по модулю простого числа q (если $k \neq 0,\ s \neq 0$). При этом величины k и x можно рассматривать по модулю q, а не p.

Существенным в схеме Эль Гамаля является то, что элементы данных (M.r.s), передаваемых проверяющему, связаны линейно, то есть соотношением вида

$$c_1 \cdot M + c_2 \cdot r + c_3 s \equiv 0 \bmod (p-1),$$

с набором коэффициентов (c_1, c_2, c_3), являющихся перестановкой элементов $(\pm x, \pm k, \pm 1)$. Знаки значения не имеют. Возможны следующие шесть случаев.

$$1. M + x \cdot r - ks \equiv 0 \bmod (p-1)$$
$$2. x \cdot M + r - ks \equiv 0 \bmod (p-1)$$
$$3. M - k \cdot r + x \cdot s \equiv 0 \bmod (p-1)$$
$$4. k \cdot M - r - x \cdot s \equiv 0 \bmod (p-1)$$
$$5. k \cdot M - x \cdot r - s \equiv 0 \bmod (p-1)$$
$$6. x \cdot M - k \cdot r + s \equiv 0 \bmod (p-1)$$

Знаки выбраны так, чтобы в проверочных соотношениях не было отрицательных степеней, и они соответственно имели бы вид:

$$1. r \equiv \alpha^{MS^{-1}} \cdot y^{rS^{-1}} \bmod p$$
$$2. r \equiv \alpha^{rS-1} \cdot y^{MS^{-1}} \bmod p$$
$$3. r \equiv \alpha^{Mr^{-1}} \cdot y^{Sr^{-1}} \bmod p$$
$$4. r \equiv \alpha^{SM^{-1}} \cdot y^{rM^{-1}} \bmod p$$
$$5. r \equiv \alpha^{SM^{-1}} \cdot y^{rM^{-1}} \bmod p$$
$$6. r \equiv \alpha^{Sr^{-1}} \cdot y^{Mr^{-1}} \bmod p$$

Отсюда видно, как изменятся процедура подписи и проверки при вычислении r и s одним из приведенных шести способов. Так, при $c_3 = \pm k$ (1 и 2 случай) или $c_3 = \pm x$ (3 и 4 случай) при подписи приходится вычислять соответственно $k^{-1} \bmod (p-1)$ или $x^{-1} \bmod (p-1)$. В случае $c_3 = \pm 1$ (5 и 6 случай) этого делать не надо. Вычисление $x^{-1} \bmod (p-1)$ можно также отнести к предварительной работе, когда вырабатывается ключ подписи x. Соответственно, при проверке надо вычислять $s^{-1} \bmod (p-1)$ (1 и 2 случай), $r^{-1} \bmod (p-1)$ (3 и 6 случай) и $M^{-1} \bmod (p-1)$ (4 и 5 случай).

Стандарты США и России соответствуют 1 и 5 из приведенных случаев соответственно.

Помимо необходимости вычислять $k^{-1} \bmod (p-1)$, $x^{-1} \bmod (p-1)$, $r^{-1} \bmod (p-1)$ и $M^{-1} \bmod (p-1)$, необходимо само существование этих обратимых элементов, что гарантируется их взаимной простотой с модулем (p-l) . Проверка взаимной простоты также требует дополнительного времени на подписывание, поэтому естественно гарантировать существование этих обратных элементов за счет выбора в качестве модуля простого числа q, являющегося делителем числа (p-l). В качестве q надо взять наибольший простой делитель числа (p-l), который должен существовать по требованию усложнения методов логарифмирования и увеличения стойкости схемы. При этом произойдет и уменьшение показателя степени, в которую надо возводить числа при проверке подписи, а также уменьшится размер самой подписи, что важно для экономии памяти.

Наконец, в качестве α надо взять элемент g порядка q, степени которого $\{g^i\}$ пробегают все множество чисел {1,...,q-1}, и все проверочное соотношение привести по модулю q, так как левая его часть нам тоже нужна в виде $r = (g^k \bmod p) \bmod q$.

Для сравнения американского и российского стандартов цифровой подписи выпишем их параллельно друг другу.

FIPS PUB 186	**ГОСТ Р 50 34.10-94**
$p : 2^{L-1} < p < 2^L,\ 512 \le L \le 1024,\ 64 \mid L$	$p : 2^{510} < p < 2^{512},$ или $g : 1 < g < p-1$
$q : \mid p - 12^{159} < q < 2^{160}$	$q : \mid p - 12^{254} < q < 2^{256}$
$g : g = h^{(p-1)/q} \bmod p,\ 1 < h < p-1$ g имеет порядок q	$g : 1 < g < p-1$ g имеет порядок q
$x : 0 < x < q$	$x : 0 < x < q$
$y : y = g^x \bmod p$	$y : y = g^x \bmod p$
$k : 0 < k < q$	$k : 0 < k < q$

Процедура	подписи
$r = (g^k \bmod p) \bmod q$	$r = (g^k \bmod p) \bmod q$
$s = k^{-1}(H(M) + xr) \bmod q$ (предварительно найти $k^{-1} \bmod q$)	$s = (x \cdot r + k \cdot H(M)) \bmod q$

Процедура проверки подписи по $\overline{M}, \bar{r}, \bar{s}$ $\overline{M}, \bar{r}, \bar{s}$	
$W = (\bar{S})^{-1} \bmod q$	$W = H(\overline{M})^{-1} \bmod q = \overline{H}(M)^{q-2} \bmod q$
$u1 = (H(\overline{M})^{-1} w) \bmod q$	$u1 = \bar{s} \cdot w \bmod q$
$u2 = (\bar{r} \cdot w) \bmod q$	$u2 = -r \cdot w \bmod q = (q - r) v \bmod q$
$v = (g^{u1} \cdot y^{u2} \bmod p) \bmod q =$ $= (g^{H(\overline{M})\bar{s}^{-1}} \bmod p) \bmod q$	$v = (g^{u1} g^{u2} \bmod p) \bmod q =$ $= (g^{\bar{s}H(\overline{M})^{-1}} \bmod p) \bmod q$
$r = v(?)$	$r = v(?)$

Таблица 6. Сравнение американского и российского стандартов цифровой подписи

Для полного соответствия предыдущим рассуждениям надо в них вместо сообщения *M* иметь в виду значение *H(M)* функции хэширования *H*. Кроме того, в приведенной таблице опущены для сокращения изложения проверки неравенство $0 < \bar{s} < q$, $0 < \bar{r} < q$, а также замена *H(M)* на 1 в случае $H(M) \equiv 0 (\bmod q)$ в российском стандарте, что необходимо для существования обратного элемента к *H(M)* по модулю *q*.

Если проверка неравенств не производится (например, из-за ошибки программиста при реализации проверки), то это позволяет подделать подпись методом из работы [Bleichenbacher D., Eurocrypt 96].

Приведенный материал дает возможность показать этапы становления этих стандартов из классической цифровой подписи Эль Гамаля.

3.3 Развитие американских стандартов хэш-функции

Первый американский стандарт хэш-функции (Secure Hash Standard) принят в 1993 г. Это был первый стандарт, позже были приняты и другие. Стандарт был разработан на основе алгоритма хэш-функции MD4, предложенного Райвестом [23]. Стандарт содержит описание односторонней функции хэширования без ключа (MDC), преобразующий двоичное сообщение длины меньшей 2^{64} в строку длины 160 бит, называемую **хэш-кодом** или **дайджестом сообщения** (message digest). Размер хэш-кода согласован с размером входа алгоритма цифровой подписи DSS (FIPS PUB 186-94).

Приведем основные используемые обозначения и понятия. **Словом** (word) называется 32-битовая строка, которая может представлять собой и целое число из множества $0, 2^{32}-1$. Наряду с этим рассматриваются и шестнадцатеричные цифры (hex digit) – элементы множества {0,1,...,9,A,...,F}, которые представляют 4-битовые строки (7 = 0111, A = 1010). Наконец, **блок** (block) – это 512-битовая строка =16 слов.

Символ «+» обозначает сложение слов, как чисел по модулю 2^{32}, символы «$\wedge$», «$\vee$», «$\oplus$», «-» обозначают соответственно побитовое умножение, логическое сложение, сложение по модулю два, логическое отрицание. Через $S^n(x)$ $S^n(x)$ (или x«n) обозначен оператор циклического сдвига слова x на n позиций влево.

Пусть $(m_0, m_1, ..., m_{b-1})$ – преобразуемое хэш-функцией сообщение, представленное в битовой форме. Значение функции получается последовательным поблочным преобразованием входных данных.

Сначала производится расширение сообщения (message palling) так, что его длина становится кратна 512. Расширенное сообщение имеет вид:

$(m_0, m_1, ..., m_{b-1}, 1, 0^{a-1}, 1_1, ..., 1_{64}); m_i, 1_i \in \overline{0,1}$.

Последние 64 бита последнего блока содержат значение длины исходного сообщения (отсюда ограничение на длину 2^{64}).

Размер (a-l) нулевой строки 0^{a-1} равен минимальному целому числу, при котором 512 делит b + a + 64.

Обозначим через $(M_1, M_2, ..., M_n)$ разбиение этого расширенного сообщения на блоки по 512 бит. Для получения значения функции хэширования используются два регистра $H = (H_0, H_1, ..., H_4)$ и Q из 5 разрядов каждый. Разряды содержат 32-битовые слова.

Если обозначить через $\overline{H}(i) = (H_0(i), ..., H_4(i))$ – состояние первого регистра H в момент времени i, то оно будет меняться по следующему рекуррентному правилу:

$\overline{H}(i) = \overline{H}(i-1) + \overline{k}\left[\overline{H}(i-1), M_i\right],\ i = \overline{1,n}$.

Начальное состояние имеет вид:

$\overline{H}(0) = 67452301\quad \overline{H}_1(0) = SFCDAB89\quad \overline{H}_2(0) = 98BADCFE$

$\overline{H}_3(0) = 10325476\quad \overline{H}_4(0) = C3D2E1F10$.

Последовательное соединение (конкатенация) заполнений ячеек в конечном состоянии $\overline{H}(n)$ и является значением хэш-функции. В последнем соотношении «+» обозначает поразрядное сложение по модулю 2^{32} состояния $\overline{H}(i-1)$ и вектор-функции $\overline{k}(\overline{H}(i-1), M_i)$, описание которой будет дано далее. Таким образом, получение дайджеста сообщения можно реализовать следующей схемой неавтономного регистра сдвига:

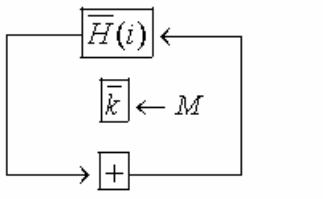

Рис. 6

Наконец, опишем функцию $\overline{k}$. Для получения ее значений используется 5-разрядный регистр Q и 16-разрядный регистр W, каждая ячейка памяти которого также содержит слово. Значение функции $\overline{k}$ представляет собой состояние регистра Q после его изменения по следующему рекуррентному

правилу за 80 тактов работы:

$$\overline{Q}(t+1) = \left[A(t), B(t), C(t), D(t), E(t)\right] =$$
$$\left[\begin{array}{l} S^5 A(t) + f_t\left[B(t), C(t), D(t)\right] + \\ E(t) + W(t) + K(t), A(t), S^{30}\left[B(t)\right], C(t), D(t)t \end{array}\right] = \overline{0,79} \ .$$

Здесь

$$K_t \begin{cases} 5A827999(t = \overline{0,19}) \\ 6ED9EBA1(t = \overline{20,39}) \\ 8F1BBCDC(t = \overline{40,59}) \\ CA62C1D6(t = \overline{60,79}) \end{cases} \qquad f(B,C,D)_t \begin{cases} BC \oplus \overline{B}D(t = \overline{0,19}; t = \overline{40,59}) \\ B \oplus C \oplus D(t = \overline{20,39}; t = \overline{60,79}) \end{cases},$$

а W(t) получается по линейному рекуррентному соотношению из блока сообщения $M_i = \left[W(0), ..., W(15)\right]$ следующим образом:

$$W(t) = W(t-3) \oplus W(t-8) \oplus W(t-14) \oplus W(t-16), t = \overline{16,79} \ .$$

Так, получение значения функции к можно реализовать следующей схемой:

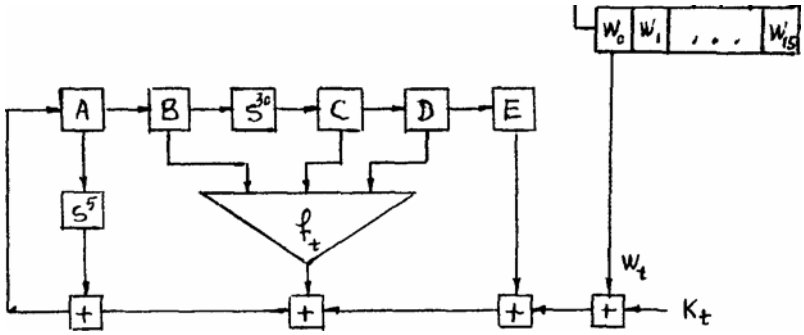

Рис. 7. Схема реализации функции хэширования SHA-1

Этим заканчивается описание хэш-функции SHS. В заключение заметим, что схема, реализующая вектор-функцию $\overline{k}$, может быть использована в качестве блочной криптосистемы с секретным ключом.

К настоящему моменту принят целый ряд американских стандартов хэш-функции с разными размерами хэш-кода от 224 до 512.

SHA-256

Алгоритм SHA-256 работает наподобие алгоритма SHA-1. Функция сжатия алгоритма работает на 512-bit блоке сообщения и 256-bit промежуточном хэш-значении (переменная связи). На самом деле функция сжатия представляет собой алгоритм 256-битного блочного шифра, который шифрует промежуточное значение хэш-функции, используя для этого блок сообщения в качестве ключа.

$\oplus$	bitwise XOR
$\wedge$	bitwise AND
$\vee$	bitwise OR
$\neg$	bitwise complement
$+$	mod 2^{32} addition
R^n	right shift by n bits
S^n	right rotation by n bits

Обозначения

Все эти операторы действуют на 32-битные слова.

Начальное значение $H^{(0)}$ является следующей последовательностью 32-bit слов (которые получаются путем взятия дробных частей квадратных корней первых восьми простых чисел):

$$H_1^{(0)} = \texttt{6a09e667}$$
$$H_2^{(0)} = \texttt{bb67ae85}$$
$$H_3^{(0)} = \texttt{3c6ef372}$$
$$H_4^{(0)} = \texttt{a54ff53a}$$
$$H_5^{(0)} = \texttt{510e527f}$$
$$H_6^{(0)} = \texttt{9b05688c}$$
$$H_7^{(0)} = \texttt{1f83d9ab}$$
$$H_8^{(0)} = \texttt{5be0cd19}$$

Предварительная обработка

Вычисление значения хэш-кода сообщения начинается с подготовки сообщения:

1) Сообщение увеличивается таким образом, чтобы его длина была кратна 448 по модулю 512. Добавление осуществляется всегда, даже если сообщение уже имеет нужную длину. Таким образом, число добавляемых битов находится в диапазоне от 1 до 512. Добавление состоит из единицы, за которой следует необходимое количество нулей. К сообщению добавляется блок из 64 битов. Этот блок трактуется как беззнаковое 64-битное целое и содержит длину исходного сообщения до добавления. Для примера (8bit-ASCII) сообщение «abc» имеет длину 8*3=24, тогда, дополняя его единичным битом, а затем 448-(24+1)=423 нулевыми битами, его длина становится дополненной до 512-битного сообщения

$$\texttt{01100001 01100010 01100011 1}\ \underbrace{00\cdots0}_{423}\ \underbrace{00\cdots011000}_{64}.$$

2) Разбиваем сообщение на N 512-битовых блоков $M^{(1)}$, $M^{(2)}$,…, $M^{(N)}$. Первый 32-битовый блок сообщения обозначим как $M^{(1)}{}_0$, следующий 32-bit блок через $M^{(1)}{}_1$ и так далее включительно до $M^{(1)}{}_{15}$ блока. Далее мы используем представление big-endian, так чтобы крайний левый бит являлся старшим.

Основной цикл

Вычисление хэш-кода продолжается следующим образом:

For i=1 to N (где N=число блоков увеличенного сообщения)

{

Устанавливаем регистры a,b,c,d,e,f,g,h с $(i-1)^{\text{st}}$ промежуточным значением хэш-функции (оно равно первоначальному значению хэш-кода, если i=1) •

$$a \leftarrow H_1^{(i-1)}$$
$$b \leftarrow H_2^{(i-1)}$$
$$\vdots$$
$$h \leftarrow H_8^{(i-1)}$$

Применяем функцию сжатия алгоритма SHA-256 для обновления регистров a,..,h •

For j=0 to 63

{

Основой алгоритма является модуль, состоящий из 64 циклических обработок каждого блока M{i}.

Вычисляем *Ch(e,f,g),Maj(a,b,c),*$\sum_0(a)$, $\sum_1(e)$,W_j (смотри описание ниже)

$$T_1 \leftarrow h + \Sigma_1(e) + Ch(e,f,g) + K_j + W_j$$
$$T_2 \leftarrow \Sigma_0(a) + Maj(a,b,c)$$
$$h \leftarrow g$$
$$g \leftarrow f$$
$$f \leftarrow e$$
$$e \leftarrow d + T_1$$
$$d \leftarrow c$$
$$c \leftarrow b$$
$$b \leftarrow a$$
$$a \leftarrow T_1 + T_2$$

}

Вычисляем i-ое промежуточное хэш-значение $H^{(i)}$ •

$$H_1^{(i)} \leftarrow a + H_1^{(i-1)}$$
$$H_2^{(i)} \leftarrow b + H_2^{(i-1)}$$
$$\vdots$$
$$H_8^{(i)} \leftarrow h + H_8^{(i-1)}$$

}

$H^{(N)} = (H_1^{(N)}, H_2^{(N)}, \ldots, H_8^{(N)})$ является хэш-кодом M-ого блока.

Описание

Алгоритм SHA-256 использует шесть логических функций, каждая из которых выполняется над 32-битными словами (обозначенные как x, y, z).

Результатом каждой функции является 32-битное слово. Каждая функция определяется следующим образом:

$$
\begin{aligned}
Ch(x,y,z) &= (x \wedge y) \oplus (\neg x \wedge z) \\
Maj(x,y,z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\
\Sigma_0(x) &= S^2(x) \oplus S^{13}(x) \oplus S^{22}(x) \\
\Sigma_1(x) &= S^6(x) \oplus S^{11}(x) \oplus S^{25}(x) \\
\sigma_0(x) &= S^7(x) \oplus S^{18}(x) \oplus R^3(x) \\
\sigma_1(x) &= S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x)
\end{aligned}
$$

Расширенные блоки сообщения $W_0, W_1, \ldots, W_{63}$ вычисляются, как следует ниже, через расписание сообщения алгоритма SHA-256:

$W_j = M_j^{(i)}$ for $j = 0, 1, \ldots, 15$, and
For $j = 16$ to 63
{

$$W_j \leftarrow \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$$

}

Последовательность постоянных слов (констант) $K_0, \ldots, K_{63}$, используемых в SHA-256 в шестнадцатеричном виде, является первыми 32-мя битами дробной части кубических корней первых 64-х простых чисел.

Схема работы

Функция сжатия алгоритма SHA-256 приведена ниже:

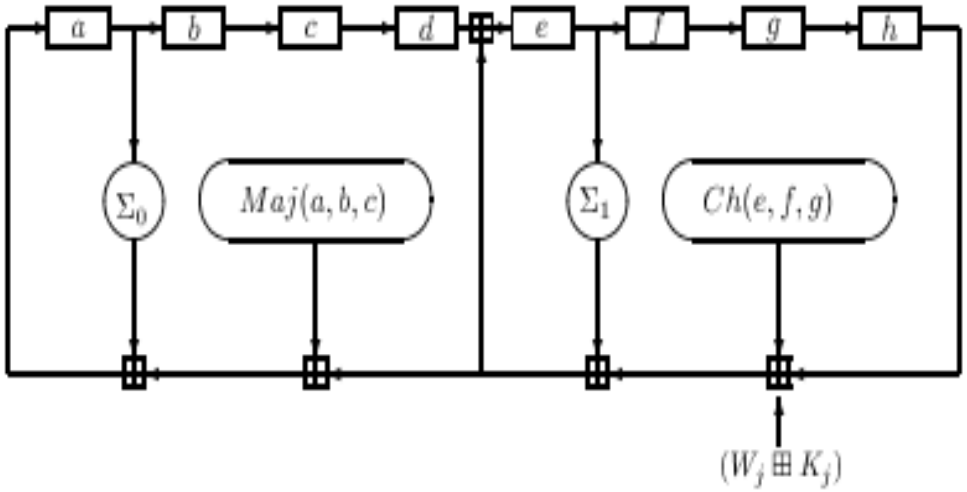

Рис. 8. j^{th} **внутренний шаг функции сжатия С алгоритма SHA-256, где символ** $\boxplus$ **означает сложение по** 2^{32}

Усложнение сообщения может быть описано следующим образом:

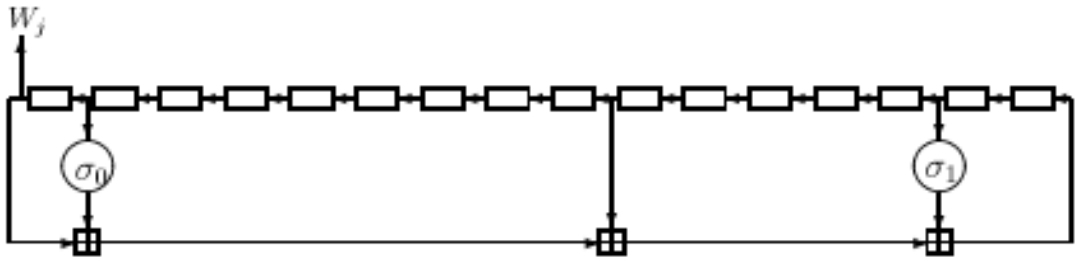

Рис. 9. Регистр для усложнения сообщения в SHA-256

Регистры здесь загружены с соответствующими 32-bit блоками $W_0, W_1, \ldots, W_{15}$.

SHA-512

Алгоритм SHA-512 является вариантом SHA-256, который оперирует на восьми 64-битных словах. Сообщение в начале хешируется следующим образом:

1) его длина дополняется таким образом, чтобы длина результата являлась кратной 1024-битам, затем
2) сообщение разбивается на 1024-битные блоки $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$

Блоки сообщения обрабатываются по одному: начиная с фиксированного начального значения $H^{(0)}$ последовательно вычисляем $H^{(i)} = H^{(i-1)} + C_{M^{(i)}}(H^{(i-1)})$, где C является функцией сжатия SHA-512, а + означает сложение по $\mod 2^{64}$. $H^{(N)}$ является значением хэш-функции для M-ого блока.

Описание алгоритма SHA-512

Функция сжатия алгоритма SHA-512 работает на 1024-битном блоке сообщения и 512-битном промежуточном хэш-значении. На самом деле функция сжатия представляет собой алгоритм 512-битного блочного шифра, который шифрует промежуточное значение хэш-функции, используя для этого блок сообщения в качестве ключа.

Начальное хэш-значение $H^{(0)}$ является следующей последовательностью 64-битных слов (которые получаются путем взятия дробной части квадратных корней первых восьми простых чисел).

Предварительная обработка

Вычисление значения хэш-функции сообщения начинается с предварительной подготовки сообщения (т.е. добавление определенных битов до целого числа блоков и последующее разбиение на блоки выполняется аналогично тому, как это делалось в SHA-1 с учетом длины блока каждой хэш-функции):

1. Дополняем сообщение обычным способом: предполагаем, что длина сообщения M в битах есть L. Конец сообщения дополняем единичным битом и затем k-тыми нулевыми битами, где k является наименьшим неотрицательным решением уравнения L+1+k =896 mod 1024. К полученному добавляем 128-битный блок, который равен числу L в двоичном представлении. Для примера (8bit- ASCII) сообщение «abc» имеет длину 8*3=24, тогда, дополняя его единичным битом, затем 896-(24+1)=871 нулевыми битами, получим, что его длина станет добавленной до 1024-битного сообщения

$$01100001\ 01100010\ 01100011\ 1\ \underbrace{00\cdots0}_{871}\ \underbrace{00\cdots0011000}_{128}.$$

Длина добавленного сообщения должна быть кратной 1024 битам.

2. Разбиваем сообщение на N 1024-битные блоки $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$.

Первый 64-битный блок сообщения обозначим как $M_0^{(i)}$, следующий 64-битный блок через $M_1^{(i)}$ и так далее включительно до $M_{15}^{(i)}$ блока. Далее мы используем представление big-endian, так чтобы в каждом 64-битном слове крайний левый бит являлся старшим.

Основной цикл

Вычисление хэша продолжается следующим образом:

For i=1 to N (где N = число блоков, добавленных в сообщение)

{

•Устанавливаем регистры a,b,c,d,e,f,g,h с $(i-1)^{\text{st}}$ промежуточным значением хэш-функции (оно равно первоначальному значению хэша в тех случаях, когда i=1) •

$$a \leftarrow H_1^{(i-1)}$$
$$b \leftarrow H_2^{(i-1)}$$
$$\vdots$$
$$h \leftarrow H_8^{(i-1)}$$

Применяем функцию сжатия алгоритма SHA-512 для обновления регистров a,..,h •

For j=0 to 79

{

Основой алгоритма является модуль, состоящий из 80 циклических обработок каждого блока M:

Вычисляем $Ch(e,f,g), Maj(a,b,c), \sum_0(a), \sum_1(e), W_j$ (см. описание ниже)

$$
\begin{aligned}
T_1 &\leftarrow h + \Sigma_1(e) + Ch(e,f,g) + K_j + W_j \\
T_2 &\leftarrow \Sigma_0(a) + Maj(a,b,c) \\
h &\leftarrow g \\
g &\leftarrow f \\
f &\leftarrow e \\
e &\leftarrow d + T_1 \\
d &\leftarrow c \\
c &\leftarrow b \\
b &\leftarrow a \\
a &\leftarrow T_1 + T_2
\end{aligned}
,
$$

где Kj-восемьдесят 64-битных констант, каждая из которых является первыми 64-мя битами дробной части кубических корней первых восьмидесяти простых чисел

}

Вычисляем i-ое промежуточное хэш-значение $H^{(i)}$ •

$$
\begin{aligned}
H_1^{(i)} &\leftarrow a + H_1^{(i-1)} \\
H_2^{(i)} &\leftarrow b + H_2^{(i-1)} \\
&\vdots \\
H_8^{(i)} &\leftarrow h + H_8^{(i-1)}
\end{aligned}
$$

}

$H^{(N)}=(H^{(N)}_1, H^{(N)}_2,\ldots, H^{(N)}_8)$ является хэш-кодом M-ого блока.

Описание

Алгоритм SHA-512 использует шесть логических функций, каждая из которых выполняется над 64-битными словами, обозначенными как x,y,z. Результатом каждой функции является 64-битное слово. Каждая функция определяется следующим образом:

$$
\begin{aligned}
Ch(x,y,z) &= (x \wedge y) \oplus (\neg x \wedge z) \\
Maj(x,y,z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\
\Sigma_0(x) &= S^{28}(x) \oplus S^{34}(x) \oplus S^{39}(x) \\
\Sigma_1(x) &= S^{14}(x) \oplus S^{18}(x) \oplus S^{41}(x) \\
\sigma_0(x) &= S^{1}(x) \oplus S^{8}(x) \oplus R^{7}(x) \\
\sigma_1(x) &= S^{19}(x) \oplus S^{61}(x) \oplus R^{6}(x)
\end{aligned}
$$

Расширенные блоки сообщения W_0, W_1,…, W_{79} вычисляются, как следует ниже, через расписание сообщения алгоритма SHA-512:

$W_j = M_j^{(i)}$ for $j = 0, 1, \ldots, 15$, and
For $j = 16$ to 79
{

$$W_j \leftarrow \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$$

}

Последовательность постоянных слов (констант) K_0, K_1,…, K_{79}, используемых в SHA-512 в шестнадцатеричном виде, является первыми 64-мя битами дробной части кубических корней первых 80-ти простых чисел.

Схема работы

Функция сжатия алгоритма SHA-512 приведена ниже:

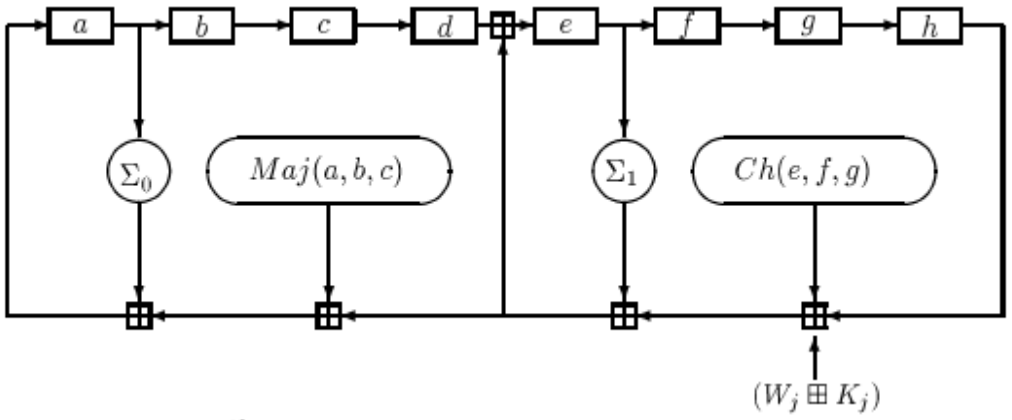

Рис. 10. j^{th} **внутренний шаг функции сжатия алгоритма SHA-512, где символ** $\boxplus$ **означает сложение по** $\bmod 2^{64}$

Расписание сообщения может быть описано следующим образом:

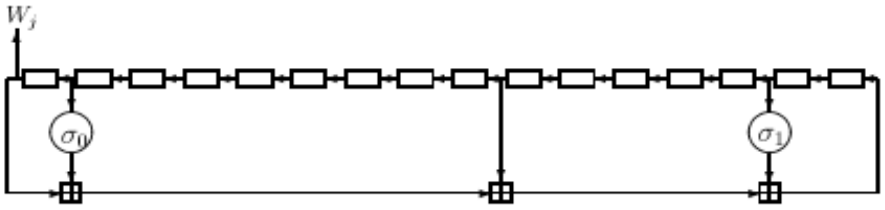

Рис. 11. Регистр для усложнения сообщения в SHA-512

Регистры здесь загружены соответствующими блоками $W_0, W_1,\ldots, W_{15}$.

SHA-384

SHA-384 работает так же, как и SHA-512, за исключением двух следующих особенностей:

1. начальное значение хэш-функции $H^{(0)}$ базируется на дробной части квадратных корней первых шестнадцати простых чисел;

2. 384-битный хэш-код получается из левых 384 битов окончательного хэш-кода H(N): $H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)}$.

3.4. Российский стандарт хэш-функции ГОСТ Р 34.11-94

Стандарт разработан в 1994 г. Описанная в нем хэш-функция h отображает любую последовательность двоичных символов в 256-битовый вектор в зависимости от стартового вектора хэширования H (тоже вектор размера 256) и таблиц замены $П_1, П_2,\ldots, П_8$, используемого алгоритма шифрования по ГОСТ 28147-89 в режиме простой замены. Поэтому в зависимости от протокола использования эта функция хэширования может как зависеть от ключа, так и не зависеть, если указанные ключевые параметры зафиксировать и сделать открытыми (число возможных вариантов $(16^{16})^8$). Размер выхода согласуется с входом алгоритма стандарта цифровой подписи ГОСТ Р 34.10-94 и ГОСТ Р 34.10-2001.

Процедура вычисления функции итеративная, обработка последовательности M начинается с конца по 256 бит (за исключением возможно меньшего начального блока).

Процедура начинается с присвоения начальных значений параметрам:

M – часть необработанной процедурой последовательности M,

H – текущее значение хэш-функции (в начальный момент это стартовый вектор),

E – текущее значение контрольной суммы $< 2^{256}$,

L – текущее значение длины обработанной последовательности $< 2^{256}$.

Этап 1. Присвоение начальных значений $M, N, \sum := 0^{256}, L := 0^{256}$.

Этап 2.

2.1. Проверить условие для длины |M| > 256. Да --> этап 3.

2.2. L = L + |H|

2.3. M' = $0^{256-|M|} \| M$ (символ $\|$ обозначает присоединение)

2.4. $\sum := \sum + M'$

2.5. H = к (M',H) (функция к будет описана позже)

2.6. H = к (L,H)

2.7. H = κ ($\sum$,H)

2.8. Конец работы алгоритма. Значение H на шаге 2,7 является значением функции хэширования.

Этап 3.

3.1. Вычислить суффикс M_s слова м длины 256 бит $(M = M_p \| M_s)$.

3.2. $H := k(M_s, H)$

3.3. L : = L + 256

3.4. $\sum := \sum \oplus M_s$

3.5. $M := M_p$

3.6. Перейти к этапу 2.

Фигурирующие в описании параметры могут рассматриваться и как двоичные векторы, и как целые числа, имеющие двоичное представление в виде этих соответствующих векторов, в зависимости от операции, которая к ним применяется.

Перейдем теперь к описанию пошаговой функции хэширования, преобразующей два 256-битных вектора в один такого же размера (256 бит). Ее реализация также может быть разбита на 3 последовательных этапа.

Этап 1. Генерация ключей-слов длины 256 – для ГОСТ 28147-89.

Этап 2. Зашифрование 64-битных локов слова H на ключах $k_i (i = \overline{1,4})$, полученных на этапе 1, с помощью ГОСТ 28147-89 в режиме простой замены $E_{k_1}(H) = S_i$.

Этап 3. Перемешивание выхода этапа к. 2 с помощью регистров сдвига.

Пусть $X = (b_{256},....,b_1) = x_4 \| x_3 \| x_2 \| x_1 \| = \eta_{16} \| ... \| \eta_1 = \xi_{32} \| ... \| \xi_1 |$ – представление двоичного вектора X длины 256 в виде конкатенации разного числа векторов одинаковой длины.

Обозначим через

$A(x) = x_1 \oplus x_2 \| x_n \| x_3 \| x_2$, "$\oplus$" – сложение векторов по модулю 2.

$P : \xi = (\xi_{32} \| ... \| \xi_1) \to (\xi_{\phi(32)}) \| ... \| \xi_{\phi(1)})$,

где $\phi(i + 1 + 4(k-1)) = 8i + k, i = \overline{0,3}, k = \overline{1,8}$

Для генерации ключей этапа к.1 необходимо использовать исходные данные H, M - 256-битные слова и слова C_2, C_3, C_4, такие, что

$C_2 = C_4 = 0^{256}, C_3 = 1^8 0^8 1^{16} 0^{24}.....$

1. i=1, U:=H, V:=M

2. W:=U $\oplus$ V, $K_1 = P(W)$

3. i=i+1

4. Проверить i=5. Да —> п.7, нет —> п.5.

5. $U = A(U) \oplus C_i, V := A(A()V), W := U \oplus V, K_i = P(W)$

6. Перейти к п.3.

7. Конец.

На этапе 3 исходные данные $H = h_4 \| h_3 \| n_2 \| h_1$ с помощью набора ключей с этапа 1 преобразуются с использованием ГОСТ 28147-89

$$S = S_4 \| S_3 \| S_2 \| S_1 = E_{K_4}(h_4) \| E_{K_3}(h_3) \| E_{K_2}(h_2) \| E_{K_1}(h_1 \| \ ,$$

где $E_k(h)$ – обозначает преобразование зашифрования в режиме простой замены.

Пусть Ψ преобразование вида

$$\Psi : n_{16} \| ... \| \eta_1 \to \eta_1 \oplus \eta_2 \oplus \eta_3 \oplus \eta_4 \oplus \eta_{13} \oplus \eta_{16} \| \eta_{16} \| ... \| \eta_2 .$$

Тогда значение пошаговой функции хэширования к после последнего этапа к.3 равно

$$k(M, H) = \Psi^{61}\left[H \oplus \Psi(M \oplus \Psi^{12}(S)) \right],$$

где Ψ^i – i-тая степень преобразования Ψ.

Этим завершается описание всего алгоритма вычисления функции хэширования.

3.5 Использование эллиптических кривых в криптологии

Эллиптические кривые над различными полями оказались в последнее десятилетие в центре внимания по нескольким причинам, одной из которых является их применение в криптографии.

Криптосистемы на основе эллиптических кривых впервые были предложены в работах Миллера и Коблица [38, CRYPTO`85]. Многие современные стандарты схем цифровой подписи также основаны на группе точек эллиптических кривых [например, ГОСТ 34.10-2001, FIPS 186-2]. Генераторы псевдослучайных последовательностей, широко используемые при получении криптографических ключей, также могут быть построены над группой точек эллиптических кривых [http://citeseer.ist.psu.edu/449444.html]. В некоторых алгоритмах проверки простоты и факторизации целых чисел также используются эллиптические кривые [50].

Рассмотрим вначале основные понятия и определения теории эллиптических кривых.

Аффинное и проективное пространства

Пусть F – некоторое поле и $A^n(F)$ – множество наборов $a=(a_1,...a_n)$ с a_i из F. Его можно рассматривать как векторное пространство при обычном способе определения сложения и умножения на скаляры. А можно его рассматривать как множество и называть его **n-мерным аффинным пространством над F**. Элементы $a=(a_1,...a_n)$ с a_i из F этого пространства будем называть **аффинными точками**_или просто **точками**. Точка этого пространства (0,…,0) называется **началом координат**.

P^n(F), **n-мерное проективное пространство над F**, несколько более сложное понятие. Для этого рассмотрим $A^{(n+1)}$(F), обозначая его точки через $(a_0,a_1,...,a_n)$.

Определим отношение эквивалентности на этом (n+1)-мерном аффинном пространстве с выброшенным началом координат следующим образом. Точка $(a_0,a_1,...,a_n)$ **эквивалентна** точке $(b_0,b_1,...,b_n)$, если существует такой элемент d из F*, что $a_0=d\cdot b_0, a_1=d\cdot b_1,...,a_n=d\cdot b_n$ (F* – обратимые элементы поля F). Классы эквивалентности называются **точками пространства**_P^n(F) или **проективными точками**. Во многих работах, например в [K04], для обозначения классов эквивалентности

[a] =[$(a_0,a_1,...,a_n)$] используется обозначение $(a_0:a_1:...:a_n)$.

Пространство P^n(F) содержит больше точек, чем $A^n(F)$. Можно утверждать следующее. Пусть H – множество классов [a] из P^n(F) с нулевой координатой a_0. Определим отображение ф точки [a] из (P^n(F)\ H) в $A^n(F)$ следующим образом

ф([a]) = $(a_1 \setminus a_0,...,a_n \setminus a_0)$.

Тогда это отображение ф является взаимно однозначным.

Множество H называется **бесконечно удаленной гиперплоскостью**. Множество H обладает структурой пространства $P^{(n-1)}(F)$. Таким образом,

P^n(F) состоит из двух частей (P^n(F)\ H) и H, первая есть копия пространства $A^n(F)$, и его точки называются **конечными точками**, а другая – копия пространства $P^{(n-1)}(F)$, и его точки называются **бесконечно удаленными точками**. Например, P^0(F) состоит из одной точки [a] (a – не нулевая) и начала координат [0].

Для определения множеств, называемых **гиперповерхностями**, привлечем многочлены. Пусть $F[x_1, x_2, ..., x_n]$ – кольцо многочленов от n переменных над полем F. Если f(x) – элемент этого кольца, то он имеет вид

$$f(x) = \sum_{(i_1,...,i_n)} a_{i_1,...,i_n} \cdot x_1^{i_1} \cdot ... \cdot x_n^{i_n},$$

где сумма берется по всем наборам неотрицательных целых чисел $(i_1,\ldots,i_n)$, для которых не равны нулю коэффициенты $a_{i_1,...,i_n}$.

Многочлен вида $a_{i_1,...,i_n} x_1^{i_1} \cdot ... \cdot x_n^{i_n}$ называется **одночленом**. По определению общая **степень одночлена** равна сумме $(i_1+i_2+\ldots+i_n)$, а **степень по переменной** x_m равна i_m. **Степень многочлена f(x)** есть максимум степеней одночленов, которые входят в f(x) с ненулевыми коэффициентами. **Степень по x_m многочлена f(x)** есть максимум степеней по x_m одночленов, которые входят в f(x) с ненулевыми коэффициентами. Если обозначить эти степени через deg f(x) и $\deg_m$ (f(x)), то

(a) deg(f(x) g(x)) = deg f(x) + deg g(x);

(b) $\deg_m$ (f(x)) g(x)) = $\deg_m$ (f(x)) + $\deg_m$(g(x)).

Если все одночлены, входящие в f(x), имеют степень L, то f(x) называется **однородным многочленом степени L.** Однородный многочлен иногда называется **формой**. Форма степени 1 называется **линейной формой**, степени 2 называется **квадратичной формой**, а форма степени 3 – **кубической формой** и так далее.

Каждый однородный многочлен $f(x_1, \ldots, x_n)$ степени L удовлетворяет условию $f(d\,x_1, \ldots, d\,x_n) = (d^L)\, f(x_1, \ldots, x_n)$ для всех $x_1, \ldots, x_n$, d из поля F.

Оказывается, для бесконечного поля многочлен, удовлетворяющий этому условию, является однородным степени L (см. [Кнэпп Э. Эллиптические кривые. – М.: Факториал Пресс, 2004. – 448 с. С. 37]).

Пусть К – некоторое поле, содержащее поле F. Если f(x) из $F[x_1, x_2, ..., x_n]$ и а из $A^n(K)$, то можно подставить a^i вместо x^i и вычислить f(a). Это показывает, что f(x) определяет функцию из $A^n(K)$ в K, которая переводит а в f(a). Точка а из $A^n(K)$, для которой f(a)=0, называется нулем функции f(x) [Айерленд К., Роузен М. Классическое введение в современную теорию чисел, М.: Мир, 1987.- 416 с.].

Для произвольного ненулевого многочлена f(x) определим множество $H_f(K)$ = {а из $A^n(K)$: f(a)=0}, называемое **гиперповерхностью в аффинном пространстве $A^n(K)$, определяемой многочленом f**. Если поле К конечно, то конечно и число точек в $H_f(K)$.

Определим теперь **проективную гиперповерхность**. Пусть h(x) из $F[x_0, x_1, x_2, ..., x_n]$ – ненулевой **однородный** многочлен степени L. Как и выше, К есть поле, содержащее поле F. Для d из K* имеем $h(dx) = (d^L)\, h(x)$. Отсюда следует, что если а из $A^{n+1}(K)$ и h(a)=0, то h(da)=0. То есть любой другой элемент из класса эквивалентности [a] также является нулем многочлена h(x). Таким образом, можно положить $H`_h(K)$ = {[a] из $P^n(K)$: h(a)=0}. Это множество называется **гиперповерхностью в проективном пространстве P^n(K) , определенной однородным многочленом h**.

В более общем виде, если f_1, …, f_m – многочлены в $F[x_1, x_2, ..., x_n]$, положим

V = {$(a_1, ...a_n)$: a_i из F, i=1,…,n, $f_j((a_1, ...a_n))$=0, j=1,…,m}, которое называется **алгебраическим множеством, определенным над полем F.** В работе [Рид М. Алгебраическая геометрия для всех, - М.: Мир, 1991.- 151 с.] это множество называется **(алгебраическим) многообразием**.

Аналогично, множество общих проективных нулей (т.е. корней из $P^n(K)$) конечного набора однородных многочленов из $F[x_0, x_1, x_2, ..., x_n]$ называется **проективным алгебраическим множеством**.

Определим **проективное замыкание аффинной гиперповерхности**. Пусть f(x) из $F[x_1, x_2, ..., x_n]$, и определим $f`(y) = f`(y_0, y_1 \ldots, y_n)$ посредством формулы $f`(y) = y_0^{\deg f} f(y_1/y_0, \ldots, y_n/y_0))$.

Можно утверждать следующее: f `(y) – однородный многочлен степени deg f. Кроме того, $f`(1, y_1, \ldots, y_n) = f(y_1, \ldots, y_n)$.

Рассмотрим гиперповерхность H_f (K) в $A^n(K)$. Многочлен f `(y) однороден по переменным $y_0$, …, $y_n$, а потому f` определяет гиперповерхность $H`_{f`}(K)$ ={[a] из $P^n(K)$: f`(a)=0} в $P^n(K)$. Эта гиперповерхность называется **проективным замыканием аффинной гиперповерхности** $H_f(K)$ в $P^n(K)$.

Пусть отображение λ: $A^n(K)$ в $P^n(K)$ определено равенством

$\lambda\ (a_1, \ldots, a_n) = [(1, a_1, \ldots, a_n)]$.

Отображение λ взаимно однозначно с $\mathrm{Im}\lambda$, где $\mathrm{Im}\lambda$ – область значений отображения (т.е. образ) и, кроме того, образ H_f (K) при λ содержится в $H`_{f`}(K)$, так как, очевидно,

$f`([(1, a_1, \ldots, a_n)]) = f(a_1, \ldots, a_n) = 0$, для всех a из H_f (K).

Вообще говоря, гиперповерхность $H`_{f`}(K)$ имеет больше точек, чем H_f (K), поскольку в ней еще имеется пересечение с бесконечно удаленной гиперплоскостью.

Основываясь на предыдущих определениях, перейдем к эллиптическим кривым. Далее мы будем фактически рассматривать случай n=2. А именно аффинное пространство $A^2(F)$ и проективное пространство $P^2(F)$. Согласно определениям приведенных выше работ пространство $P^2(F)$ называется **проективной плоскостью (projective plane)**. Поменяем для простоты обозначения и точки $P^2(F)$ будем обозначать через (x :y :z). Иногда, где не

может возникнуть путаница, будем обозначать точку проективной плоскости $P^2(F)$ с однородными координатами (x : y : z) и через (x , y , z).

Более наглядно проективную плоскость можно представить над полем действительных чисел R. В этом случае точками проективной плоскости $P^2(R)$ являются прямые в $A^3(R)$ (или R^3), проходящие через начало координат, или классы эквивалентности (x : y : z) на $R^3\backslash\{0\}$, где (x, y, z) эквивалентно (dx, dy, dz), если d из $R\backslash\{0\}$. Приведенные выше рассуждения показывают, что $P^2(R)$ содержит экземпляр $A^2(R)$, и такое стандартное вложение задается отображением λ точки (x, y) из $A^2(R)$ вида: (x, y) → (x : y : 1). Дополнение к образу в $P^2(R)$ состоит в точности из тех точек, у которых z=0 (ранее обозначалось через H). Эти точки называются **бесконечно удаленными**.

В свою очередь $(P^2(R)\backslash H)$ отображается взаимно однозначно на $A^2(R)$ (где H – множество точек (x : y : z) из $P^2(R)$ с нулевым значением z) с помощью отображения вида

$$\phi((x : y : z)) = (x/z, y/z).$$

Пусть K – некоторое поле и f(x,y,z) из K[x,y,z] – однородный многочлен степени d. Полезно ввести геометрическую терминологию. Говорят, что уравнение f(x,y,z) = 0 определяет **кривую степени d над полем К**. Поле К называется ее **полем определения**.

В работе [Кнэпп Э. Эллиптические кривые. – М.: Факториал Пресс, 2004. – 448 с.] **проективной плоской кривой над полем К** называется именно однородный многочлен степени d. Многочлен f нельзя рассматривать как функцию, определенную на проективной плоскости $P^2(K)$. Тем не менее, можно говорить о множестве корней многочлена f как о подмножестве $P^2(K)$. В самом деле, если f(x,y,z) = 0 для некоторых однородных координат точки (x:y:z) проективной плоскости, то это же равенство выполняется для любых других однородных координат этой точки в силу однородности многочлена f. Множество $f(K) = \{(x:y:z) : x,y,z \in K, f(x,y,z) = 0\}$ называется **множеством K-точек кривой f**. В этих обозначениях **соответствующей**

аффинной кривой называется кривая, задаваемая многочленом вида F(x,y) = f(x,y,1).

В тех случаях, когда d = 1,2 или 3, проективная кривая называется соответственно **прямой**, **кривой второго порядка** и **кубической кривой**. В работе [Рид М. Алгебраическая геометрия для всех, М.: Мир, 1991., с. 18, 20] плоская кривая, задаваемая однородным квадратным уравнением

$$Ax^2 + Bxy + Cy^2 + Dxz + Eyz + Fz^2 = 0,$$

называется **коникой**.

Если L – поле, содержащее K, то можно рассматривать корни многочлена f в проективном пространстве $P^2(L)$. В введенной выше терминологии это есть $H`_f(L)$. Гиперповерхность в проективном 2-пространстве и называется **кривой**.

Точка a из $H`_f(L)$ называется **неособой**, если она не является решением системы уравнений df/dx = 0, df/dy = 0, df/dz = 0.

Прямой в $P^2(K)$ над полем K будем далее называть гиперповерхность, определяемую в $P^2(K)$ ненулевым многочленом Ax+By+Cz с коэффициентами из K (правда, в русском переводе в [Кнэпп Э. Эллиптические кривые] прямой называется и ненулевой многочлен ax+by+cz, и уравнение ax+by+cz=0).

Точки из $P^2(R)$, для которых z=0, соответствуют классам (x : y : 0). Эти точки образуют «бесконечно удаленную прямую». По определению прямая L на проективной плоскости P^2 задается уравнением aX+bY+cZ=0 и проходит через точку (X:Y:0) тогда и только тогда, когда aX+bY =0. В аффинных координатах та же прямая задается уравнением ax+by+c=0, так что все прямые с одним и тем же отношением a:b проходят через одну точку на бесконечности.

В таком случае прямая, задаваемая уравнением

(df/dx(a)) x + (df/dy(a)) y + (df/dz(a)) z = 0,

называется **касательной к кривой f в точке a**.

Кривая, определяемая уравнением f(x,y,z) = 0, называется **неособой кривой**, если все точки в $H_f(L)$ неособые для всех расширений L поля K.

Можно утверждать следующее.

(a) Две различные прямые в $P^2(K)$ пересекаются ровно в одной точке.

(b) Для любых двух различных точек (x:y:z) и (x`:y`:z`) проективной плоскости $P^2(K)$ существует ровно одна прямая ax+by+cz=0, содержащая эти точки.

Утверждение согласуется с нашими представлениями из аналитической геометрии в 3-мерном пространстве над полем действительных чисел R. Прямой в $P^2(R)$ соответствует плоскость в R^3, проходящая через начало координат, а точке в $P^2(R)$ соответствует прямая, проходящая через начало координат. Две такие различные плоскости в R^3 пересекаются по прямой, проходящей через начало координат. Она соответствует точке в пространстве $P^2(R)$. Также в R^3 через две прямые, проходящие через начало координат, можно провести одну плоскость, что соответствует утверждению (b).

Заметим, что в $A^2(K)$ прямые не обязательно пересекаются в одной точке.

Можно утверждать следующее. Если поле L алгебраически замкнуто, то прямая в пространстве $P^2(L)$ пересекает кривую степени d, определяемую однородным многочленом f(x,y,z), в d точках.

Чтобы показать это, запишем x=x/z, y=y/z и обозначим f*(x,y) = f(x,y,1). Мы рассмотрим в данный момент аффинное пространство $A^2(L)$ и аффинную часть кривой. Для нахождения точек пересечения кривой, задаваемой уравнением f*(x, y) = 0, с прямой y = mx+b (получается из ax+by+cz=0) подставляем значение y в f* и находим корни уравнения. Если f имеет степень d, то последнее уравнение будет иметь в общем случае степень d, а так как L алгебраически замкнуто, будет в наличии d корней с учетом кратности. Исключениями являются лишь пересечения на бесконечности, когда f*(x, mx+b) будет иметь степень меньшую d.

Если a из $H`_F(L)$, то касательная прямая к f в точке a пересекается с кривой f=0 с кратностью 2 или больше. Если кратность больше чем 2, то a называется **точкой перегиба**.

Если многочлен f(x,y,z) определен над K, то его корень в $P^2(K)$ называется рациональной точкой над K.

Будем говорить, что неособый однородный кубический многочлен f(x,y,z) из K[x,y,z] определяет эллиптическую кривую над K, если имеется по крайней мере одна рациональная точка [Айерленд К., Роузен М.].

Группы преобразований занимают в геометрии центральное место. **Аффинная замена координат** в K^2 имеет вид Ф v +B , где v=(x,y) из K^2, Ф – обратимая 2X2 – матрица из группы обратимых матриц GL(2,K), а B – вектор сдвига. Известно, что любая невырожденная коника $Ax^2+Bxy+Cy^2+Dxz+Eyz+Fz^2=0$ приводится аффинным преобразованием к одной из 12 форм. [Рид М., с. 18]

Рассмотрим группу GL(3,K) всех обратимых матриц третьего порядка над полем K. Так как для любой матрицы Ф из GL(3,K) будет Ф(d v) = dФv для любого v из $A^3(K)$, то Ф задает взаимно однозначное отображение $P^2(K)$ на $P^2(K)$, называемое **проективным преобразованием проективной плоскости $P^2(K)$, соответствующим матрице Ф** [Кнэпп Э., Рид М.].

Матрицы Ф1 и Ф2 задают одно и то же преобразование пространства $P^2(K)$ тогда и только тогда, когда одна из них пропорциональна другой, то есть Ф1 = d Ф2, где d из K*.

Кубическая кривая общего вида задается однородным многочленом

$$F(x,y,z) = C_{yyy}\,y^3+C_{xyy}\,x\,y^2+C_{xxy}\,x^2\,y+C_{yyz}\,y^2\,z+C_{xyz}\,x\,y\,z+C_{yzz}\,y\,z^2+C_{xxx}\,x^3 + \quad +C_{xxz}\,x^2\,z+C_{xzz}\,x\,z^2+C_{zzz}\,z^3.$$

Пусть f – такая кубическая кривая над полем K, что точка (x_0,y_0,z_0) является ее точкой перегиба. Тогда существует такое проективное преобразование Ф, что

$$f^{\phi} = y^2z+a_1\,xyz+a_3\,yz-x^3-a_2\,x^2z-a_4\,x\,z^2-a_6z^3.$$

Доказательство этого в [Кнэпп Э, с. 59].

Кубическую кривую такого вида называют **кривой, заданной в длинной форме Вейерштрасса.** Соответствующее уравнение в аффинных координатах (при z=1) записывается в виде

$$y^2 + a_1 xy + a_3 = x^3 + a_2 x^2 + a_4 x + a_6.$$

(Обозначения в этих записях стандартные.)

Эллиптической кривой над полем К называется неособая (гладкая) кривая, заданная в длинной форме Вейерштрасса

$$y^2 + a_1 xy + a_3 = x^3 + a_2 x^2 + a_4 x + a_6,$$

при условии, что все коэффициенты лежат в поле К [Соловьев Ю.П., Садовничий В.А., Шавгулидзе Е.Т., Белокуров В.В. Эллиптические кривые и современные алгоритмы теории чисел. – Москва-Ижевск: инст. компьют. исследований, 2003. – 192 с.].

Через E(K) обозначается множество, состоящее из точек (x, y) из $A^2(K)$, удовлетворяющих этому уравнению, и «бесконечно удаленной» точки O.

Следует еще раз обратить внимание на встречающиеся различные определения эллиптической кривой. Например, во многих работах и документах [ГОСТ 34.10-2001, FIPS 186-2] **эллиптической кривой** называется именно множество точек E(K), удовлетворяющих соотношению $y^2 + a_1 xy + a_3 = x^3 + a_2 x^2 + a_4 x + a_6$ или $y^2 = x^3 + a x + b$, со специально выбранными коэффициентами для гладкости, причем «бесконечно удаленную» точку O не всегда включают в множество E(K) [ГОСТ 34.10-2001].

Условие гладкости кривой означает, что в множестве E(K), где K – алгебраическое замыкание поля K, не существует точек, в которых одновременно обращались бы в ноль частные производные $d f(x,y)/dx$ и $d f(x,y)/d y$, где $f(x,y)= y^2 + a_1 xy + a_3 y - x^3 - a_2 x^2 - a_4 x - a_6$.

Иными словами, система уравнений

$d\ f(x,y)/dx = a_1 y - 3x^2 - 2a_2 x - a_4 = 0$

$d\ f(x,y)/d\ y = 2y + a_1 x + a_3 = 0$

не имеет решений в E(K). Условие гладкости необходимо для задания на точках кривой структуры абелевой группы. Проверка гладкости кривой достаточно проста.

Если характеристика поля char K не равна 2, то линейной заменой переменных y → $y - (a_1 x + a_3)/2$ кривая приводится к виду

$$y^2 = x^3 + a_2 x^2 + a_4 x + a_6.$$

Конечно, в последнем выражении значения коэффициентов a_2, a_4, a_6 другие, чем в исходном.

Если характеристика поля char K не равна 2 и 3, то линейной заменой переменных x →x – 1/3 a2 эллиптическая кривая примет вид

$$y^2 = x^3 + a\,x + b,$$

где a, b из поля K. Этот вид задания эллиптической кривой называется **короткой формой Вейерштрасса.**

Условие гладкости кривой в этом случае означает, что кубический многочлен $x^3 + a\,x + b$ не имеет кратных корней. Это выполняется тогда и только тогда, когда дискриминант этого многочлена, равный $-(4a^3 + 27b^2)$, отличен от нуля.

Как правило, в многочисленных криптографических приложениях эллиптические кривые задаются именно в этой короткой форме Вейерштрасса [ГОСТ 34.10-2001, FIPS 186-2], а в качестве поля выбираются конечные поля вида F_p и F_{2^n}.

Сложение точек эллиптической кривой

Сначала заметим, что на некоторых плоских кривых существуют естественные законы сложения, такие, что относительно этой операции точки кривых образуют алгебраическую группу. Простейшими примерами таких кривых являются прямая и окружность [Прасолов В.В., Соловьев Ю.П. Эллиптические функции и алгебраические уравнения. – М.: Факториал,

1997.-288с.]. Например, суммой двух точек (r cos a, r sin a) и (r cos b, r sin b) окружности $x^2+y^2=r^2$ можно считать точку (r cos (a+b), r sin (a+b)).

Нагляднее всего сложение точек эллиптической кривой продемонстрировать для поля действительных чисел R, хотя многие рассуждения будут верны и для произвольного поля.

Определим операцию сложения точек P и Q на кривой E, отправляясь от графического изображения кривой [38]. Проведем через точки P и Q прямую. В общем случае эта прямая пересечет кривую еще в третьей точке. Отразим эту точку относительно оси Ox и назовем полученную точку суммой P + Q точек P и Q. Не всегда прямая, проходящая через две точки, пересекает кривую E в третьей точке, например, этого не происходит с вертикальной прямой. Этот случай рассмотрим далее подробно.

Заметим, что определить сумму P + Q точек P и Q можно не только таким образом, но именно такое определение наделяет множество точек эллиптической кривой структурой абелевой группы. Проиллюстрируем описание операции на рисунке.

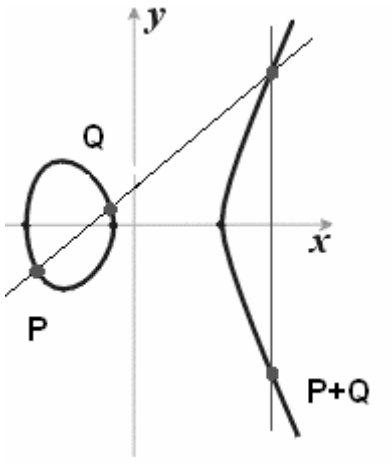

Рис.12. Общий вид эллиптической кривой над полем действительных чисел

Покажем, что относительно этой операции точки кривой образуют абелеву группу.

Очевидно, что так определенная операция коммутативна, так как для вычисления Q + P используется та же самая прямая, что и для P + Q.

Займемся существованием нейтрального по сложению элемента – нуля. Это такая точка O на кривой, что P + O = P для любой точки P кривой. Мы хотим найти нечто такое, что если провести прямую через P и это нечто, пересечь получившуюся прямую с кривой и потом отразить точку пересечения от оси Ox, то вновь получится точка P. Обозначим через P` точку, симметричную P относительно оси Ox. Из сказанного вытекает, что прямая должна проходить через точки P и P`, то есть должна быть вертикальной. Следовательно, если имеется точка O, для которой P + O = P для всех P, то эта точка должна лежать и на кривой, и на любой вертикальной прямой. Разумеется, в плоскости (xOy) такой точки нет. Поэтому добавим ее к плоскости и кривой и назовем ее **бесконечно удаленной точкой** или **точкой в бесконечности**. Каким требованиям она должна удовлетворять? Любая вертикальная прямая стремится к бесконечности сверху и снизу. Потребуем, чтобы все эти точки в бесконечности были одной и той же точкой O, то есть будем считать, что точка O есть точка пересечения всех вертикальных прямых. Для того чтобы точно понять, что это значит, рассмотрим проективное дополнение аффинной плоскости и кривой на ней. Рассмотрим проективную плоскость $P^2(K)$ с однородными координатами (X:Y:Z), причем будем считать, что координаты на исходной аффинной плоскости имеют вид

$$x = X/Z,\ y = Y/Z.$$

Тогда уравнение, соответствующее эллиптической кривой $y^2 = x^3 + a\,x + b$, примет вид $Y^2\,Z = X^3 + a\,X\,Z^2 + b\,Z^3$.

При ненулевой Z мы видим, что $y^2 = x^3 + a\,x + b$ выполняется тогда и только тогда, когда $(Y/Z)^2 = (X/Z)^3 + a\,(X/Z) + b$.

Бесконечные точки нашей кривой – это классы точек (X : Y : 0), для которых выполнено соотношение $Y^2\,0 = X^3 + a\,X\,(0)^2 + b\,0$, то есть X=0.

Существует только один такой класс эквивалентности (0 : 1 : 0). Это и есть искомая бесконечно удаленная точка O.

Требования, которые мы предъявили к точке O, корректно определяют ее как нулевую точку относительно операции сложения точек на эллиптической кривой. Действительно, в силу нашего соглашения, вертикальная прямая, проходящая через точку P, проходит через P и O. Поэтому точка P` пересечения этой прямой с эллиптической кривой удовлетворяет соотношению P + P` = O, то есть является обратной по сложению к точке P. В то же время P` – это точка, симметричная к P относительно оси Ox. Значит, любая точка P имеет обратную точку – P = P`.

Для вычисления точки 2P = P + P нужно провести не секущую, а касательную в этой точке. Точки вида n P теперь находятся по индукции: 3P= 2P+P, … .

Для того чтобы окончательно убедиться, что точки эллиптической кривой относительно описанной операции сложения образуют абелеву группу, осталось проверить ассоциативность для этой операции, то есть выполнение равенства (P+Q)+S=P+(Q+S) для всех точек кривой E(K), включая бесконечно удаленную точку O.

Геометрически доказать это трудно, но можно показать с помощью приводимых ниже формул вычисления координат точки, являющейся суммой двух произвольных точек. При этом надо будет рассмотреть несколько случаев, в зависимости от того, P равно Q или нет, и от того, S равно (P+Q) или нет, что делает доказательство достаточно трудоемким (http://math.rice.edu/~friedl/papers/AAELLIPTIC.PDF).

Формулы для вычисления координат суммы точек эллиптической кривой.

Пусть $P = (x_1, y_1)$, $Q = (x_2, y_2)$, $P + Q = (x_3, y_3)$. Уравнение прямой на аффинной плоскости, соединяющей точки P и Q, имеет вид y=к x+d, где

$k=(y_2 - y_1)/(x_2 - x_1)$, $d = y_1 - k\,x_1 = (x_2\,y_1 - x_1\,y_2)/(x_2 - x_1)$.

Подставив y=к x+d в уравнение эллиптической кривой $y^2 = x^3 + a x + b$, получим уравнение третьей степени $(к\ x+d)^2 = x^3 + a x + b$, то есть

$$x^3 - k^2 x^2 + (a - 2kd) x + b - d^2 = 0.$$

Мы знаем два его корня x1 и x2, так как точки $(x_1, k x_1 + d)$ и $(x_2, k x_2 + d)$ лежат на кривой. Отсюда по теореме Виета

$x_3 = k^2 - x_1 - x_2$,

$y_3 = -k x_3 - d$ (знак минус взят в силу определения операции сложения точек). Подставляя в эти формулы значения k и d, получим

$x_3 = ((y_2 - y_1)^2 / (x_2 - x_1)^2) - x_1 - x_2$,

$y_3 = -y1 + ((y_2 - y_1)/(x_2 - x_1))(x_1 - x_3)$.

При $x_1 = x_2$ эти формулы не имеют смысла. Это соответствует случаю, когда P и Q лежат на вертикальной прямой и их сумма равна точке в бесконечности.

Случай, когда P = Q, разбирается аналогично. В этом случае уравнение секущей y = к x+d нужно заменить уравнением касательной и действовать по прежней схеме; здесь k есть значение производной в точке P и равно $k = ((3 x_1^2 + a)/2 y_1)$, и поэтому мы получаем координаты удвоенной точки P вида

$x_3 = ((3x_1^2 + a)/2 y_1)^2 - 2x_1$,

$y_3 = - y_1 + ((3x_1^2 + a)/2y_1)(x_1 - x_3)$.

Ясно, что эти формулы верны для любого поля, характеристика которого отлична от 2 и 3. Для указанных случаев будут справедливы приводимые ниже формулы.

Если $a_1 = a_3 = 0$ в длинной форме Вейерштрасса, а a_2 не обязательно нуль (случай, включающий char K = 3), то

$x_3 = ((y_2 - y_1)/(x_2 - x_1))^2 - a_2 - x_1 - x_2$,

$y_3 = -y_1 + ((y_2 - y_1)/(x_2 - x_1))(x_1 - x_2)$,

когда складываются различные точки, и

$$x_3 = (3x_1^2 + 2\,a_2\,x_1 + a_4/2y_1)^2 - a_2 - 2x_1,$$

$$y_3 = -\,y_1 + (3x_1^2 + 2\,a_2\,x_1 + a_4/2y_1)\,(x_1 - x_3),$$

когда точка удваивается.

Если char K = 2 и в длинной форме Вейерштрасса $a_3 = a_4 = 0$, $a_1 = 1$, то

$$x_3 = ((y_1+y_2)/(x_1 + x_2))^2 + (y_1+y_2)/(x_1+x_2) + x_1+x_2+a_2,$$

$$y_3 = ((y_1+y_2)/(x_1 + x_2))\,(x_1+x_3) + x_3 + y_3,$$

когда складываются различные точки, и

$$x_3 = x_1^2 + a_6/x_1^2,$$

$$y_3 = x_1^2 + (x_1 + y_1/x_1)\,x_3 + x_3$$

когда точка удваивается.

Если char K =2 и в длинной форме Вейерштрасса $a_1 = a_2 = 0$, но a_3 не равно 0, то

$$x_3 = ((y_1+y_2)/(x_1 + x_2))^2 = x_1 + x_2,$$

$$y_3 = (y_1+y_2)/(x_1 + x_2)\,(x_1+x_3) + y_1 + a_3,$$

когда складываются различные точки, и

$$x_3 = (x_1^4 + a_4^2)/a_3^2,\ y_3 = (x_1^2 + a_4/a_3)(x_1+x_3) + y_1 + a_3,$$

когда точка удваивается.

Отметим еще раз важность рассмотрения именно гладких (или неособых) кривых. Уравнение неособой кубической кривой можно записать в виде

$$y^2 = (x - e_1)(x - e_2)(x - e_3),$$

где числа e_1, e_2, e_3 попарно различны. В качестве примера можно рассмотреть кривую, задаваемую уравнением

$$y^2 = x^3 - x = (x + 1)\,x\,(x - 1).$$

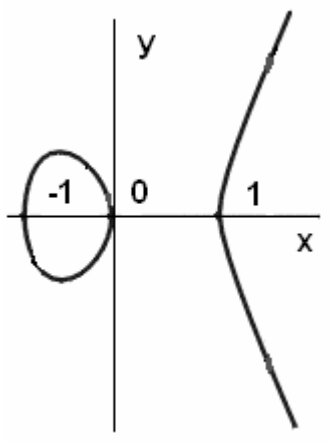

Рис. 13. Кривая $y^2 = x^3 - x$

Пусть $e_1 < e_2 < e_3$. При слиянии корней e_1 и e_2 получается кривая вида $y^2 = x^2 (x-1)$. При слиянии корней e_2 и e_3 получается кривая вида $y^2 = x^2 (x+1)$.

При слиянии всех трех корней получается кривая вида $y^2 = x^3$.

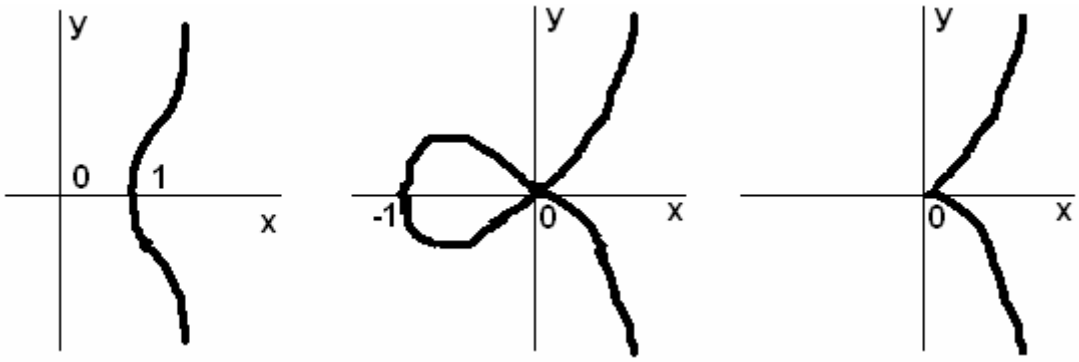

Рис. 14. Кривая $y^2 = x^3$

Для всех этих трех кривых начало координат является особой точкой. Для особых точек определить операцию сложения не удается. Вот почему для наделения точек кривой структурой абелевой группы необходимо рассматривать неособые кривые.

Существуют и отличные от формы Вейерштрасса представления эллиптических кривых (наиболее часто используются также формы Лежандра, Монтгомери). **Форма Лежандра** для эллиптической кривой над алгебраически замкнутым полем позволяет выразить кривую с помощью

всего одного параметра. Если характеристика поля не равна 2, то уравнение, задающее кривую, можно представить в виде

$$y^2 = x^3 + a_2 x^2 + a_4 x + a_6 = (x - e_1)(x - e_2)(x - e_3),$$

где e_1, e_2, e_3 принадлежат полю в силу его замкнутости. Сделаем замены $x = x_1 (e_2 - e_1) + e_1$ и $y = y_1 (e_2 - e_1)^{0.5}$. Получим

$$y_1^2 (e_2 - e_1) = x_1 (e_2 - e_1)(x_1 (e_2 - e_1) - (e_2 - e_1))(x_1 (e_2 - e_1) + e_1 - e_3),$$

или при разных e_2 и e_1 получим

$$y_1^2 = x_1 (x_1 - 1)(x_1 - L), \quad \text{где } L = (e_3 - e_1)/(e_2 - e_1).$$

Значение параметра L в представлении не однозначно и в зависимости от перестановки e_1, e_2, e_3 в приведенных заменах может принимать одно из 6 значений L, 1/L, 1 –L, 1/(1 – L), L/(1-L), (L-1)/L.

Уравнение неособой кубической кривой, как уже отмечалось, можно записать в виде $y^2 = (x - e_1)(x - e_2)(x - e_3)$, где числа e_1, e_2, e_3 попарно различны. В случае поля действительных чисел такая кривая изображена на следующем рисунке:

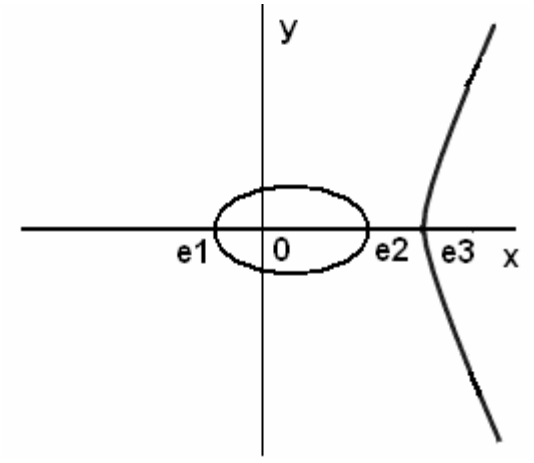

Рис. 15. Кривая $y^2 = (x - e_1)(x - e_2)(x - e_3)$,

j-инвариант эллиптической кривой

Пусть уравнение $y^2 = x^3 + a x + b$ определяет некоторую эллиптическую кривую над полем K, характеристики отличной от 2 или 3.

Если положить $x = x_1/d^2$ и $y = y_1/d^3$ при некотором обратимом d (из алгебраического замыкания поля К), то получим аналогичное соотношение в форме Вейерштрасса

$y_1^2 = x_1^3 + a_1 x_1 + b_1$, где $a_1 = a\ (d^4)$, $b_1 = b\ (d^6)$.

Заметим, что если бы мы сделали такую замену в длинной форме Вейерштрасса, то новые коэффициенты имели бы вид $a_i (d^i)$, что объясняет нумерацию коэффициентов в этой форме.

Определим **j-инвариант** кривой как $j = 1728\ (4a^3/(4a^3 + 27b^2))$.

Знаменатель в этом выражении совпадает с дискриминантом многочлена $x^3 + a\ x + b$, взятым со знаком минус, поэтому не может быть нулевым в силу выбора коэффициентов a и b для гладкости кривой. Оказывается, что проведенная замена не меняет j-инвариант.

Можно утверждать следующее.

Пусть $y_1^2 = x_1^3 + a_1 x_1 + b_1$ и $y_2^2 = x_2^3 + a_2 x_2 + b_2$ будут определять эллиптические кривые с j-инвариантами j_1 и j_2 соответственно. Если $j_1 = j_2$, то существует d, не равное 0 в алгебраическом замыкании поля К, такое, что $a_2 = a_1\ (d^4)$ и $b_2 = b_1\ (d^6)$. Преобразование $x_2 = x_1\ (d^2)$, $y_2 = y_1\ (d^3)$ преобразует одно соотношение в другое.

Чтобы показать это, сначала предположим, что a_1 не равно 0. Это равносильно тому, что $j_1 = j_2$ не равен 0, поэтому и a_2 не равно 0. Выберем d так, $a_2 = a_1\ (d^4)$, что возможно в силу замкнутости. Тогда в силу равенства $j_1 = j_2$ получаем

$$4\ a_2^3/(4a_2^3 + 27\ b_2^2) = 4\ a_1^3/(4a_1^3 + 27b_1^2) = 4\ d^{-12}\ a_2^3/(4\ d^{-12}\ a_2^3 + 27\ b_1^2) =$$

$$= 4\ a_2^3/(4a_2^3 + 27d^{12}\ b_1^2),$$ что означает, что $b_2^2 = (b_1\ d^6)^2$.

Поэтому $b_2 = \pm(b_1\ d^6)$. Если $b_2 = + b_1\ d^6$, то все доказано. Если $b_2 = - b_1 d^6$, то заменим d на i d (где $i^2 = - 1$). Это дает $a_2 = a_1\ (d^4)$ и $b_2 = b_1\ d^6$.

Если a_1 равно 0, то и $a_2 = 0$. Так как дискриминант не равен 0, то b_1 и b_2 не равны 0. Выбираем d таким, что $b_2 = b_1\ (d^6)$. Ч.т.д.

Заметим, что если поле К не алгебраически замкнуто, то возможно существование кривых с одинаковым инвариантом, но которые не могут быть преобразованы друг в друга, используя рациональные функции с коэффициентами из К.

Наконец, заметим, что число j является j-инвариантом для кривой вида $Y^2 = x^3 + (3j/(1728 - j))\, x + (2j/1728 - j)$, если j не равен 0 или 1728. То есть как коэффициенты эллиптической кривой определяют j-инвариант, так и j-инвариант определяет коэффициенты эллиптической кривой.

При построении криптосистем широко используются **эллиптические кривые над конечными полями**. В основном рассматривается **простое поле** F_p и поле F_{2^n}, но исследования ведутся и для произвольного конечного поля F_q, где $q=p^n$ (p – простое число).

В силу конечности поля F_q число пар (x,y), удовлетворяющих уравнению $y^2 = x^3 + a\,x + b$ над этим полем, тоже конечно, и группа точек эллиптической кривой является конечной абелевой группой.

Легко видеть, что порядок этой абелевой группы не может превышать число 2q+1. Если задать кривую в форме Вейерштрасса $y^2 = x^3 + a\,x + b$, то для каждого x из F_q число решений сравнения $y^2 = a \pmod q$ не может превышать двух, и 1 добавляет точка в бесконечности.

Кстати, в криптографических протоколах, когда надо хранить значения точек эллиптической кривой, хранится только одна координата x и знак координаты y.

Напомним основную **теорему о строении конечных абелевых групп**.

Любая конечная абелева группа либо является примарной циклической группой, либо раскладывается в прямую сумму примарных циклических подгрупп: $G = (g_1) + (g_2) + \ldots + (g_t)$, где порядки элементов $\mathrm{ord}\, g_i = p_i^{k_i}$, где $p_1, \ldots, p_t$ – не обязательно различные простые числа. При этом приведенное разложение, в котором слагаемые упорядочены так, что

$(p_i >= p_{i+1})\ \&\ ((p_i = p_{i+1} \rightarrow (k_i >= k_{i+1}),\ i = 1, \ldots, (t-1),$

называется **каноническим разложением конечной абелевой группы**, а вектор $(p_1^{k_1}, \ldots, p_t^{k_t})$ – **типом этого разложения**.

Существование приведенного разложения равносильно тому, что существует изоморфизм: $G = Z\, p_1^{k_1} + \ldots + Z\, p_t^{k_t}$, который также называется **каноническим разложением группы G.**

В [38, 197 с.] показано, что тип абелевой группы точек эллиптической кривой в форме $y^2 = x^3 - x$ над полем F_{71} равен (4,2,9).

Иногда теорему о разложении конечных абелевых групп формулируют в другом виде, когда разложение приводится в так называемой **форме Смита**.

Любая конечная абелева группа изоморфна прямой сумме групп $Z_{n1} + \ldots + Z_{ns}$, где n_i делит n_{i+1} для i=1, …, (s-1). Числа n_i однозначно определяются группой.

Эта форма записи и называется **формой Смита**. Форму Смита можно преобразовать в форму представления по степеням простых чисел и обратно. Для перехода к форме Смита надо выбрать наибольшие из степеней каждого простого числа и перемножить их, а затем повторить это пока ничего не останется. Для перехода от формы Смита надо просто разложить порядки циклических групп. Например, $Z_4+Z_4+Z_3+Z_9$ изоморфна $Z_{12}+Z_{36}$. Это разные формы представления одной и той же группы.

Теорема.

Пусть $E(F_q)$ – группа точек эллиптической кривой над конечным полем F_q. Тогда она изоморфна или Z_n, или прямой сумме $Z_{n1}+ Z_{n2}$, для некоторых целых n>=1 или n_1, n_2 >=1 и n_1 делит n_2.

Рассмотрим несколько примеров.

Пример 1.

Пусть $E(F_5)$ – группа точек эллиптической кривой, задаваемой уравнением $y^2 = x^3 + x + 1$ над простым конечным полем F_5. При небольшом размере поля легко посчитать число элементов в этой группе.

x	X^3+x+1	y	Точки
0	1	+1, -1	(0,1), (0,4)
1	3	-	-
2	1	+1, -1	(2,1), (2,4)
3	1	+1, -1	(3,1), (3,4)
4	4	+2, -2	(4,2), (4,3)

Табл. 7. Точки кривой $y^2 = x^3 + x + 1$ над полем F_5

Поэтому порядок группы равен 9, группа является циклической с порождающим элементом (0, 1). Из таблицы видно, что уравнению удовлетворяют 8 точек и, добавляя точку в бесконечности O, получаем, что порядок группы равен 9.

Для доказательства цикличности воспользуемся критерием [Глухов М.М., Елизаров В.П., Нечаев А.А., Алгебра. - М.: Гелиос АРВ, 2003.]:

G-циклическая $\Leftrightarrow \exists\ g \in G$: ord $g=|G|$

Рассмотрим элемент (0,1) и по формулам сложения точек эллиптической кривой докажем критерий:

1. (0,1)+O=(0,1)
2. (0,1)+(0,1)=(4,2)
3. (4,2)+(0,1)=(2,1)
4. (2,1)+(0,1)=(3,4)
5. (3,4)+(0,1)=(3,1).
6. (3,1)+(0,1)=(2,4)
7. (2,4)+(0,1)=(4,3)
8. (4,3)+(0,1)=(0,4)
9. (0,4)+(0,1)=O

Порядок элемента (0,1) равен 9 по определению (ord $g \in G = \min n \in N : gn=e$).

Пример 2.

Пусть $E(F_7)$ – группа точек эллиптической кривой, задаваемой уравнением $y^2 = x^3 + 2$ над простым конечным полем F_7. Для этого примера группа $E(F_7)$ содержит следующие элементы:

$E(F_7)$ = {O, (0,3), (0,4), (3,1), (3,6), (5,1), (5,6), (6,1), (6,6)}

Все эти точки P удовлетворяют соотношению 3P = O, а группа $E(F_7)$ изоморфна прямой сумме $Z_3 + Z_3$.

Первое утверждение проверяется непосредственной проверкой по формулам сложения точек эллиптической кривой.

Составим **таблицу Кели** для группы $E(F_7)$:

+	**O**	**(0,3)**	**(0,4)**	**(3,1)**	**(3,6)**	**(5,1)**	**(5,6)**	**(6,1)**	**(6,6)**
O	O	(0,3)	(0,4)	(3,1)	(3,6)	(5,1)	(5,6)	(6,1)	(6,6)
(0,3)	(0,3)	(0,4)	O	(6,1)	(5,6)	(3,1)	(6,6)	(5,1)	(3,6)
(0,4)	(0,4)	O	(0,3)	(5,1)	(6,6)	(6,1)	(3,6)	(3,1)	(5,6)
(3,1)	(3,1)	(6,1)	(5,1)	(3,6)	O	(6,6)	(0,3)	(5,6)	(0,4)
(3,6)	(3,6)	(5,6)	(6,6)	O	(3,1)	(0,4)	(6,1)	(0,3)	(5,1)
(5,1)	(5,1)	(3,1)	(6,1)	(6,6)	(0,4)	(5,6)	O	(3,6)	(0,3)
(5,6)	(5,6)	(6,6)	(3,6)	(0,3)	(6,1)	O	(5,1)	(0,4)	(3,1)
(6,1)	(6,1)	(5,1)	(3,1)	(5,6)	(0,3)	(3,6)	(0,4)	(6,6)	O
(6,6)	(6,6)	(3,6)	(5,6)	(0,4)	(5,1)	(0,3)	(3,1)	O	(6,1)

Таблица 8. Таблица Кели для группы $E(F_7)$

Представим группу $E(F_7)$ в виде прямой суммы двух подгрупп, воспользовавшись основной теоремой о строении конечной абелевой группы и теоремой о выделении прямого слагаемого в примарной абелевой группе; получаем $E(F_7)$=<g>+<f> (группа не циклическая, т.к. ее порядок равен 9, а порядки всех элементов равны 3), в частности, из таблицы следует $E(F_7)$=<(0,3)>+<(5,1)>, т.к. порядки образующих элементов равны 3, то и порядки подгрупп <g> и <f> будут равны 3. А по критерию об изоморфизме циклических групп (Две циклические группы изоморфны $\Leftrightarrow$ их порядки равны) каждая из подгрупп изоморфна Z_3, из чего следует:

$E(F_7)= Z_3 + Z_3$. QED.

Пример 3.

Пусть $E(F_{23})$ – группа точек эллиптической кривой, задаваемой уравнением $y^2 = x^3 + x$ над простым конечным полем F_{23}. Имеется 23 точки, удовлетворяющие этому уравнению над данным полем:

E(F_23) = {O,
(0,0),(1,5),(1,18),(9,5),(9,18),(11,10),(11,13),(13,5),(13,18),(15,3),(15,20),
(16,8),(16,15),(17,10)
(17,13),(18,10),(18,13),(19,1),(19,22),(20,4),(20,19),(21,6),(21,17)}

Графически точки кривой можно изобразить следующим образом:

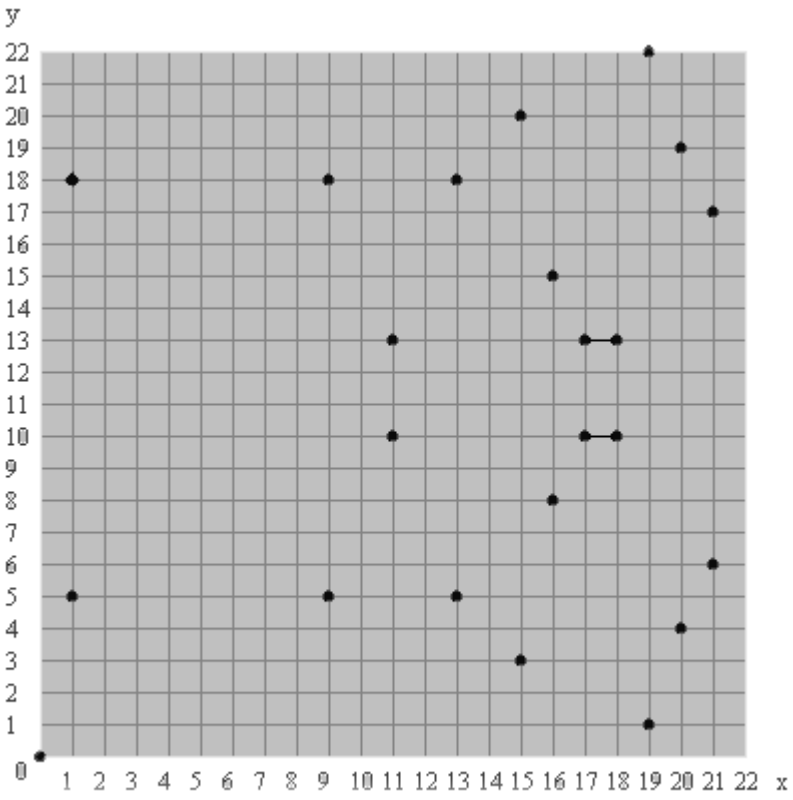

Рис. 16. Кривая $y^2 = x^3 + x$ над полем F_{23}

На рисунке видна симметрия точек.

Во многих приложениях на основе аппарата эллиптических кривых используются кривые над полем вида F_2^m ,поскольку в этом случае точки на кривой и ее коэффициенты можно представить в виде двоичных векторов.

При перечислении точек кривой в форме Вейерштрасса $y^2 = x^3 + a\,x + b$ над конечным полем F_p мы перебирали все возможные значения x, а затем находили y как квадратные корни из $(x^3 + ax + b)$, если они существуют. Рассмотрим вопрос существования и нахождения решений сравнения $y^2 = a \pmod p$ подробнее.

Целое число a, взаимно простое с простым числом p, называется **квадратичным вычетом по модулю p**, если сравнение $y^2 = a \pmod p$ имеет решение, и **квадратичным невычетом по модулю p** в противном случае.

Известно, что среди чисел {1,…, (p – 1)} содержится (p-1)/2 квадратичных вычетов и (p-1)/2 квадратичных невычетов по модулю p.

Теорема (критерий Эйлера)

Целое число a, взаимно простое с p, является квадратичным вычетом или невычетом по простому модулю p>2 в том и только том случае, когда соответственно выполняется условие:

a) $a^{(p-1)/2} = 1 \pmod p$ или b) $a^{(p-1)/2} = -1 \pmod p$.

При этом в случае a) сравнение $y^2 = a \pmod p$ имеет точно 2 решения.

Нахождение самих решений сравнения $y^2 = a \pmod p$ в случае его разрешимости достаточно просто при $p=3 \pmod 4$ и $p=5 \pmod 8$. Есть и вероятностные методы нахождения решений [38, Глухов М.М., Алгебра и аналитическая геометрия: Учебное пособие. – М.: Гелиос АРВ, 2005. – 392 с. – С. 257].

Выяснение того, является ли число a квадратичным вычетом или нет, с помощью критерия Эйлера затруднено необходимостью возведения чисел в очень большую степень. Поэтому для выяснения квадратичной вычетности был разработан метод с использованием символа Лежандра.

Символ Лежандра (обозначается как (a/p), a – числитель, p – знаменатель символа) для нечетных простых чисел p определяется как 1, если a – квадратичный вычет по модулю p, и как –1, если a квадратичный невычет.

С использованием этого понятия критерий Эйлера можно переформулировать так: целое число a является квадратичным вычетом по нечетному простому модулю p тогда и только тогда, когда

$(a/p)=a^{(p-1)/2} \pmod p$.

В работе [38] вводится понятие **квадратичного характера** ξ, обобщающего символ Лежандра. Это отображение, переводящее элемент конечного поля F_q в 1 или –1, в зависимости от того, является или нет этот элемент квадратом некоторого элемента из этого поля. В случае, когда q=p, где p – простое число, ξ (a) = (a/p), где (a/p) – символ Лежандра. В любом случае число решений уравнения $y^2 = a$ над F_q равно $1+\xi(a)$. Тогда число точек N эллиптической кривой $E(F_q)$ равно величине

$$N = 1+ \sum (1+\xi (x^3+ax+b)) = q + 1 +\sum (\xi (x^3+ax+b)),$$

где суммирование берется по всем x из F_q.

Вычисление суммы очень похоже на случайное блуждание, когда мы подбрасываем монету q раз, продвигаясь на шаг вперед, если выпал герб, и назад – если решетка. В теории вероятностей подсчитывается, что после q бросаний случайное положение оказывается на расстоянии порядка $\sqrt{q}$ от исходного положения.

Теорема (Hasse)

Число точек N эллиптической кривой над полем F_q ($q=p^n$) удовлетворяет неравенству $|N- (q+1)| < 2\sqrt{q}$.

Если обозначить через N(a, b, p) число точек эллиптической кривой в форме Вейерштрасса над полем F_p, равное (p+1–t) при некотором t, то для числа $N(a, b, p^j)$ – числа точек кривой с теми же коэффициентами a,b, но над полем F_{p^j}, то его можно найти [50, 109 с.] из соотношения

$N(a, b, p^j) = p^j + 1 – t_j$, где t_j удовлетворяет рекуррентному соотношению 2-го порядка $t\_(j+1) = (t_1)(t_j) – p (t_{j-1})$, $j>=1$, $t_1 = t$, $t_0 = 2$.

Поэтому важно уметь находить число точек кривой над конечным простым полем. Это имеет важное значение как в криптографии [ГОСТ 34.10-2001], так и в алгоритмах проверки простоты чисел [50, 123 с.].

На практике часто необходимо знать не оценку, а точное число точек эллиптической кривой. Для подсчета этого числа к настоящему времени разработано несколько алгоритмов, а их реализацию можно найти в Интернете (например, на странице http://www.informatik.tu-darmstadt.de/TI/LiDIA/Welcome.html).

Сначала Шуф (Schoof) в 1985 г. предложил полиномиальный алгоритм для подсчета числа точек эллиптической кривой для нечетных q. Вскоре Коблиц (Koblitz) распространил его на случай поля F_{2^n} [50]. Алгоритм Шуфа эффективен для небольших значений q<100. Далее методы подсчета числа точек кривых были усовершенствованы Аткиным, Элкисом, Мюллером и другими для больших значений q.

В отечественном стандарте [ГОСТ 34.10-2001] требуется, чтобы порядок группы точек эллиптической кривой делился на большое простое число q из интервала $2^{254} < q < 2^{256}$. В стандарте [FIPS 186-2] используются кривые, порядок которых имеет вид q f, где f равен 1,2 или 4, а q также большое простое число.

Пусть P – точка из группы $E(F_q)$ точек эллиптической кривой. **Порядком точки P** называется наименьшее положительное целое число k, такое, что kP = (P+ … +P) = O (суммирем k точек P).

По теореме Лагранжа порядок точки всегда делит порядок группы $E(F_q) = N$. Даже если порядок группы неизвестен, то в силу теоремы Хассе для него верно соотношение $q+1 - 2q^{0.5} < N < q+1 + 2q^{0.5}$ и можно попробовать искать в этом интервале. Размер его $4\,q^{0.5}$. Есть алгоритмы, позволяющие сократить число проб до $4\,q^{0.25}$ (см., например, [50, «Baby Step-Giant Step»]).

Заметим, что во многих криптосистемах надо вычислять кратную точку kP , поэтому делать это желательно как можно быстрее.

Дискретное логарифмирование в группе точек эллиптической кривой

Задача логарифмирования может быть поставлена для любой группы (G,*) с операцией *, в том числе и для группы точек эллиптической кривой. В общем случае в задаче требуется найти целое число x по элементам a и b = a^x, где a^x обозначает (a*a*…*a), то есть результат многократного применение операции * к элементу a из группы G. Ясно, что задача имеет решение, если в принадлежит подгруппе G, порожденной элементом a.

Сложность решения задачи логарифмирования зависит от конкретной группы. Например, для аддитивной группы (Z_m, +) кольца вычетов целых чисел по модулю m эта задача сводится к решению линейного сравнения первой степени вида ax = b(mod m) и не представляет трудности. Гораздо сложнее решение этой задачи в мультипликативной группе кольца Z_p, где p – большое простое число[B03]. В настоящее время размер этого простого числа должен составлять порядка 1000 бит, чтобы эта задача была трудно решаема и ее можно было использовать при построении стойких криптосистем. Но ввиду больших размеров используемых чисел реализация таких систем требует больших объемов машинной памяти.

В случае группы точек эллиптической кривой (E(F_p), +) с введенной операцией сложения точек задача принимает вид нахождения целого числа x из соотношения x P = Q для точек P и Q эллиптической кривой. Временная сложность наилучшего из всех известных алгоритмов решения этой задачи имеет порядок O($p^{0.5}$). С такой сложностью решать задачу дискретного логарифма можно в любой конечной группе [50], и до сих пор не удалось существенно использовать специфику самой группы точек кривой. Поэтому использование группы точек эллиптической кривой при построении криптосистем позволило уменьшить параметры криптосистем при сохранении их стойкости.

Цифровые подписи

ECDSA: Elliptic Curve Digital Signature Algorithm

Алгоритм Цифровой Подписи на основе Эллиптической Кривой (ECDSA) является аналогом Алгоритма Цифровой Подписи (DSA), но для группы точек эллиптической кривой над конечным полем. Схема описана в ряде стандартов – ANSI X9.62, FIPS 186-2, IEEE 1363-2000 и ИСО/IEC 15946-2.

В последующем обозначим хеш-функцию – H, и она имеет не более n значений.

Генерация подписи ECDSA

ВХОД: Параметры подписи D = (q, FR, S, a, b, P, n, h), секретный ключ d, сообщение m.

ВЫХОД: Подпись (r, s).

1. Выбор k из [l,n-1].
2. Вычислить $kP = (x_1, y_1)$ и преобразовать x_1 в целое x_i.
3. Вычислить $r = x_i \bmod n$. Если r = 0, тогда шаг 1.
4. Вычислить e = H(m).
5. Вычислить $s = k^{-1} (e + dr) \bmod n$. Если s = 0, тогда шаг 1.
6. Возврат (r, s).

Проверка подписи ECDSA

ВХОД: Параметры подписи D = (q, FR, S,a,b, P,n,h), открытый ключ Q, сообщение m, подпись (r, s).

ВЫХОД: Принятие или отвержение подписи.

1. Проверить, что r и s – целые в интервале [1, n - 1]. Если они не удовлетворяют этому условию, тогда признание некорректности подписи.
2. Вычислить e = H(m).
3. Вычислить $w = s_1 \bmod n$.
4. Вычислить u1 = ew mod n и u_2 = rw mod n.
5. Вычислить $X = u_1 P + M_2<2$.

6. Если X = O (точка в бесконечности), тогда признание некорректности подписи.

7. Преобразовать x-координату x_1 из X в целое x_1; вычислить $v = x_1 \bmod n$.

8. Если v = r, тогда признание корректности подписи; иначе – признание некорректности подписи.

Доказательство того, что проверка подписи верна

Если подпись (r, s) в сообщении m на самом деле была сгенерирована законным владельцем, тогда $s = k^{-1} (e + dr) \pmod n$. Перестроенное дает

$k = s^{-1} (e + dr) = s^{-1} e + s^{-1} rd = we + wrd = u_1 + ud \pmod n$.

Таким образом, $X = u1P + u_2 Q = (M_1 + u_2 d)P = kP$, и так v = r, как и требовалось.

Схема Нуберга-Рюппеля (Nyberg-Rueppel) цифровой подписи с восстановлением сообщения и с использованием группы точек эллиптической кривой

В схемах **цифровой подписи с восстановлением сообщения** не используется текст сообщения при проверке подписи, а, напротив, выдается подписанное сообщение при положительном результате проверки. При этом сообщение перед генерацией подписи должно быть преобразовано так, чтобы в него была включена избыточная информация.

Напомним схему Нуберга-Рюппеля для мультипликативной группы $Z^*_{p.}$

Пространством сообщений здесь является $M = Z^*_p$, где p – простое число, в качестве пространства Ms подготовленных к подписанию сообщений используется декартово произведение Ms = Zp x Zq, где q – простое число, являющееся делителем числа (p - 1). Пусть R – функция избыточности из пространства сообщений M в пространство подписанных сообщений Ms. Например, значение R(m) может иметь вид (m, m(mod q). Через R^{-1} обозначается функция из образа функции R в ее область определения: R^{-1} : Im(R) ->M, например, R^{-1}(m,m(mod q) = m.

Алгоритм генерации ключа тот же, что и в алгоритме DSA.

Алгоритм генерации подписи на сообщении m следующий:

а) Вычислить $m^* = R(m)$.

б) Выбрать случайное секретное целое число к, $1 < к < q- 1$, и вычислить $r = a^{~к} (mod p)$.

в) Вычислить $e = m^*r (mod\ p)$.

г) Вычислить $s = (ae + к) (mod\ q)$.

д) Объявить пару (e, s) как подпись на сообщении m.

Алгоритм проверки подписи следующий:

а) Получить открытый ключ подписавшего.

б) Проверить, что $0<e<p$ и $0<s<p$, если нет, то отклонить подпись.

г) Вычислить $v = a^s y^{~e} (mod\ p)$ и $m^* = ve (mod\ p)$.

д) Проверить, что $m^* \varepsilon Mr$, если нет, то отклонить подпись.

е) Восстановить сообщение $m = R^{~l}(m^*)$.

Доказательство корректности алгоритма весьма коротко:

По алгоритму генерации подписи $v = a^s y^{~e} = a^{s~ae} = a^k (mod\ p)$.

Таким образом, $ve = a^k m^* a^{-k} = m^* (mod\ p)$, что и требуется.

Обобщим Схему Нуберга-Рюппеля (Nyberg-Rueppel) цифровой подписи на группу точек эллиптической кривой.

Пусть $e = h(m)$ – значение хеш-функции h для документа m. Пусть E – точка из группы точек эллиптической кривой, P – базовая точка открытого ключа, n – порядок этой точки P, s – секретный ключ подписывающего документ участника. **Открытым ключом** последнего является точка $Q = sP$.

Алгоритм генерации подписи следующий:

а) Выбрать случайное целое число k и вычислить точку $R = kP$.

б) Используя первую x-координату точки R как целое число, вычислить

$c = x + e (mod\ n)$, (*)

$d = к - sc (mod\ n)$. (**)

Пара (c,d) является подписью для документа m, такого, что $h(m) = e$.

Для проверки, что h(m) является корректным хеш-значением, выполняется следующий алгоритм:

а) Вычислить R' = dP + cQ. (***)

б) Используя первую x-компоненту R', вычислить e' =c-x'{ mod n). (****)

в) Если полученное значение совпадает с хеш-значением h(m), то последнее удостоверяется.

Вместо вычисления c по (*) вычислим cP = xP + eP (*****) над кривой E. Далее вместо (**) можно использовать уравнение

dP = kP- s(cP). (******)

Теперь можно использовать выражение (*), подставляя в него вычисленные точки cP и dP. При этом (***) умножается на s – секретный ключ подписывающего сообщение участника. Но публично известна только точка Q – замаскированная версия секретного ключа.

Это означает, что после прибавления к (****) второго члена из выражения (*), а именно cQ = c(sP), получается исходная точка R эллиптической кривой.

Если теперь x-компоненту этой точки вычесть из значения c, входящего в подпись, то восстановится хеш-значение e = h(m). Если это восстановленное значение совпадает с хэш-значением h(m'), вычисленным по полученному сообщению m', то можно считать, что последнее мог подписать только обладатель секретного ключа s и что ни сообщение, ни его хеш-значение не было изменено криптоаналитиком или вследствие ошибок при передаче или хранении.

Протокол согласования ключей (key agreement) с использованием эллиптических кривых

Классический протокол Диффи-Хеллмана можно адаптировать применительно к группе точек эллиптической кривой следующим образом.

Допустим, что E – эллиптическая кривая и Q – предварительно согласованная и опубликованная точка этой кривой. A выбирает, сохраняя в секрете, случайное число $к_a$, вычисляет координаты точки k_AQ и пересылает их Б. Аналогично Б выбирает $к_в$, вычисляет и пересылает A точку k_bQ.

Общим ключом является точка $P = к_а к_в O$. Участник А вычисляет ее, умножая на свой секретный ключ $к_а$ сообщение, поступившее от Б, а Б вычисляет ее, умножая сообщение, поступившее от А, на свой секретный ключ $к_в$. Ввиду того, что группа точек эллиптической кривой абелева, результат не зависит от порядка вычисления и, следовательно, А и Б имеют одинаковые точки:

$к_а\{к_в Q) = k_B(k_A Q) = k_A k_B Q$.

Теперь А и Б имеют одинаковые копии искомой секретной точки эллиптической кривой. Координаты этой точки они преобразуют обусловленным заранее способом в секретный ключ. Например, для этого можно взять одну из ее координат и отобразить ее с использованием некоторого отображения из Fp в множество натуральных чисел.

Проблема, стоящая перед посторонним наблюдателем, имеющим намерение узнать секретный ключ, заключается в вычислении $к_а к_в$ по известным Q, $k_a Q$-, $k_b Q$, но при неизвестных $к_а$, $к_в$. Это проблема Диффи-Хеллмана для эллиптических кривых.

Гипотеза об эквивалентности проблем дискретного логарифмирования и проблемы Диффи-Хеллмана для эллиптических кривых не доказана.

Шифрование с использованием эллиптических кривых

ECIES

Интегрированная схема шифрования на основе эллиптических кривых (The Elliptic Curve Integrated Encryption Scheme (ECIES)) была предложена Белларом и Рогэвеем как модифицированный вариант схемы шифрования Эль-Гамаля с открытым ключом. Данная схема стандартизована в ANSI X9.63 и ISO/IEC 15946-3, а также в черновом стандарте IEEE P1363a.

В ECIES схема Диффи-Хеллмана с разделяемым секретом используется для получения двух симметричных ключей k_1 и k_2. Ключ k_1 используется для шифрования открытого текста, используя симметричный шифр, в то время как ключ k_2 – для подтверждения подлинности получившегося

шифрованного текста. Используются следующие криптографические примитивы:

1. KDF (key derivation function) – функция выработки ключа, которая получается с помощью хэш-функции H. Если необходим ключ длиной l бит, тогда KDF(S) определена как конкатенация хэш-значений H(S,i), где i – счетчик, инкрементирующийся после каждого вычисления хэш-функции до тех пор, пока не сгенерированы все l бит хэш-значений.

2. ENC – функция шифрования для схемы шифрования с симметричным ключом, такой как, например, AES. DEC – функция расшифрования.

3. MAC – алгоритм кода аутентификации сообщения, например, HMAC.

ECIES зашифрование

Входные данные: $D = (q, FR, S, a, b, P, n, h)$, открытый ключ Q, открытый текст M.

Выходные данные: шифртекст (R, C, t).

1. Выбираем $k \in_R [1, n-1]$.
2. Вычисляем $R = KP$ и $Z = hkQ$. Если $Z = \infty$, то переходим к шагу 1.
3. $(k_1, k_2) \leftarrow KDF(x_Z, R)$, где x_Z – x-координата Z.
4. Вычисляем $C = Enc_{k1}(M)$ и $t = MAC_{k2}(C)$.
5. Возвращаем (R, C, t).

ECIES расшифрование

Входные данные: $D = (q, FR, S, a, b, P, n, h)$, секретный ключ d, шифртекст (R, C, t).

Выходные данные: открытый текст M или непринятие шифртекста.

1. Проводим встроенную проверку на открытом ключе величины R. Если проверка возвращает ошибку, то возвращаем («непринятие шифртекста»).

2. Вычисляем $Z = hkR$. Если $Z = \infty$, то возвращаем («непринятие шифртекста»).
3. $(k_1, k_2) \leftarrow KDF(x_Z, R)$, где x_Z – x-координата Z.
4. Вычисляем $t' = MAC_{k2}(C)$. Если $t' \neq t$, то возвращаем («непринятие шифртекста»).
5. Вычисляем $M = Dec_{k1}(C)$.
6. Возвращаем (M).

ECIES схема является безопасной, основываясь на предположениях, что схема симметричного шифрования и схема MAC являются безопасными, а также потому, что определенные нестандартизированные (но рациональные) варианты вычислительных проблем Диффи-Хеллмана являются трудноразрешимыми. Эти проблемы Диффи-Хеллмана включают в себя KDF-функцию выработки ключа.

PSEC

Схема вероятностного безопасного шифрования на основе эллиптических кривых (Probably Secure Encryption Curve scheme (PSEC)) была предложена Фуджисаки и Окамото. Данная схема получена в результате объединения PSEC-KEM, механизма инкапсуляции ключей, и DEM1, механизма инкапсуляции данных, которые описаны в ISO 18033-2.

Используются следующие криптографические примитивы:

1. KDF (key derivation function) – функция выработки ключа, которая получается с помощью хэш-функции H.

2. ENC – функция шифрования для схемы шифрования с симметричным ключом, такой как, например, AES.

DEC – функция расшифрования.

3. MAC – алгоритм кода аутентификации сообщения, например, HMAC.

PSEC зашифрование

Входные данные: $D = (q, FR, S, a, b, P, n, h)$, открытый ключ Q, открытый текст M.

Выходные данные: шифртекст (R, C, s, t).

1. Выбираем $r \in_R \{0,1\}^l$, где l – длина в битах числа n.
2. $(k', k_1, k_2) \leftarrow KDF(r)$, где l + 128 – длина в битах числа k'.
3. Вычисляем $k = k' \bmod n$.
4. Вычисляем $R = KP$ и $Z = kQ$.
5. Вычисляем $s = r \oplus KDF(R, Z)$.
6. Вычисляем $C = Enc_{k1}(M)$ и $t = MAC_{k2}(C)$.
7. Возвращаем (R, C, s, t).

PSEC расшифрование

Входные данные: $D = (q, FR, S, a, b, P, n, h)$, секретный ключ d, шифртекст (R, C, s, t).

Выходные данные: открытый текст M или непринятие шифртекста.

1. Вычисляем $Z = dR$.
2. Вычисляем $r = s \oplus KDF(R, Z)$.
3. $(k', k_1, k_2) \leftarrow KDF(r)$, где l + 128 – длина в битах числа k'.
4. Вычисляем $k = k' \bmod n$.
5. Вычисляем $R' = kP$.
6. Если $R' \neq R$, то возвращаем («непринятие шифртекста»)
7. Вычисляем $t' = MAC_{k2}(C)$. Если $t' \neq t$, то возвращаем («непринятие шифртекста»).
8. Вычисляем $M = Dec_{k1}(C)$.
9. Возвращаем (M).

PSEC схема является безопасной, основываясь на предположениях, что схема симметричного шифрования и схема MAC являются безопасными, вычислительная задача Диффи-Хеллмана является труднообрабатываемой и функция выработки ключей является случайной функцией.

Российский стандарт цифровой подписи на основе группы точек эллиптических кривых – ГОСТ Р 34.10-2001

Общепризнанная схема (модель) цифровой подписи (ИСО/МЭК 14888) охватывает три процесса:

- генерация параметров и ключей (подписи и проверки);
- формирование подписи;
- проверка подписи.

Российский стандарт не определяет процесс генерации ключей (подписи и проверки). Также в ГОСТе не рассматривается генерация параметров схемы цифровой подписи. В стандарте приводятся две основные процедуры: формирование подписи и проверка подписи.

Схема реализована с использованием операции сложения в группе точек эллиптической кривой, определенной над конечным простым полем, а также с использованием хэш-функции. Алгоритм хэширования определен в ГОСТ Р 34.11.

Размер цифровой подписи равен 512 битам, как в ГОСТ Р 34.10-94.

Параметрами схемы цифровой подписи являются:

- простое число $p > 2^{255}$, при этом $p^i \neq 1 \bmod q$ для всех целых i, меньших 32;
- эллиптическая кривая E, задаваемая своими коэффициентами;
- целое число m – порядок группы точек эллиптической кривой $E(m \neq p)$;

Простое число q – порядок циклической подгруппы группы точек кривой E, для которого выполняются условия:

$m = nq,\ n \in Z, n \geq 1$

$2^{254} < q < 2^{256}$;

- точка $P \neq O$ кривой с координатами (x_P, y_P), удовлетворяющая равенству qP=O.

Ключ формирования подписи – целое число d, удовлетворяющее неравенству 0<d<q.

Ключ проверки подписи – точка эллиптической кривой Q с координатами (x_Q, y_Q), удовлетворяющая равенству qP = Q.

Последовательность формирования цифровой подписи сообщения

Исходными данными этого процесса являются ключ подписи d и подписываемое сообщение M, а выходным результатом – цифровая подпись ζ.

1. Вычислить хэш-код сообщения $M : \bar{h} = h(M)$.

2. Вычислить целое число α, двоичным представлением которого является вектор $\bar{h}$, определить число e $\equiv \alpha$ **(**mod q**).** Если **e** = 0, то определить e = 1.

3. Сгенерировать случайное (псевдослучайное) целое число k, удовлетворяющее условию 0 < k < q.

4. Вычислить точку эллиптической кривой C = kP (P – точка эллиптической кривой порядка q) и определить $r \equiv x_c \pmod q$,

где x_c – x-координата точки C.

Если *r* = 0, то вернуться к шагу 3.

5. Вычислить значение

$$s \equiv (rd + ke) \pmod q.$$

Если ***s*** = 0, то вернуться к шагу 3.

6. Вычислить двоичные векторы $\bar{r}$ и $\bar{s}$, соответствующие *r* и *s*, и определить цифровую подпись $\zeta = (\bar{r} \| \bar{s})$ как конкатенацию двух двоичных векторов.

Последовательность проверки цифровой подписи:

Исходными данными этого процесса являются подписанное сообщение M, цифровая подпись ζ и ключ проверки Q, а результатом – свидетельство о достоверности или ошибочности данной подписи.

1. По полученной подписи ζ вычислить целые числа r и s. Если выполнены неравенства $0 < r < q$, $0 < s < q$, то перейти к следующему шагу. В противном случае подпись отвергается.

2. Вычислить хэш-код полученного сообщения M

$$\bar{h} = h(M).$$

3. Вычислить целое число α, двоичным представлением которого является вектор h, и определить

$$e \equiv \alpha \pmod{q}.$$

Если e = 0, то определить e = 1.

4. Вычислить значение $v \equiv e^{-1} \pmod{q}$.

5. Вычислить значения

$$z_1 \equiv sv \pmod{q}, \quad z_2 \equiv -rv \pmod{q}.$$

6. Вычислить точку эллиптической кривой $C = z_1P + z_2Q$ и определить

$$R \equiv x_c \pmod{q}$$

где x_c – x-координата точки **C**.

7. Если выполнено равенство R = r, то подпись принимается, в противном случае подпись неверна.

Перечислим некоторые особенности рассмотренного стандарта:

1. В ГОСТе не оговорен двоичный вид всех используемых математических объектов, различная служебная информация (дата, версия системы и пр.).

2. На усмотрение разработчика конкретной системы ЭЦП в ГОСТе оставлены несколько параметров.

Приведем полный список требуемых параметров ЭЦП:

Параметр	Описание	Длина (бит)	Вид
p	простое число	>255	постоянный
a,b	коэффициенты эллиптической кривой E, заданной уравнением $y^2=x^3+ax+b \bmod p$	256	постоянный
x_P,y_P	координаты точки P эллиптической кривой E	256	постоянный
q	порядок точки P в группе точек кривой E	256	постоянный
V_0	вектор начального заполнения хэш-функции	256	переменный
S	К1, … К8 – узлы замены в ГОСТ 28147-89	512	постоянный

Таблица 9. Список параметров цифровой подписи

Из перечисленных выше данных можно сделать вывод о том, что, хотя все разработчики криптографических систем должны следовать одному и тому же ГОСТу, сами электронные подписи получаются несовместимыми: одна криптографическая система не может проверить подпись, выработанную другой системой.

Российский стандарт цифровой подписи стал и стандартом Интернет – RFC 4357 (http://www.ietf.org/rfc/rfc4357.txt), разработаны и другие важные стандарты в этой области RFC 4490, RFC 4491, RFC 2246, RFC 3280, RFC 2560, RFC 3161.

ЗАКЛЮЧЕНИЕ

В учебном пособии наглядно продемонстрировано, как развивались и усложнялись методы и средства криптографии, как для решения ее задач привлекался все более и более сложный математический аппарат, как сама криптография влияет на развитие математики, вычислительной техники.

Обеспечение безопасности информации криптографическими методами должно постоянно контролироваться и совершенствоваться с использованием всех новейших достижений криптоанализа.

Можно отметить еще раз, что число приложений криптографии постоянно растет по мере развития информационных технологий и их проникновением во все большее число сфер деятельности и жизни общества.

ЛИТЕРАТУРА

Литература расположена в хронологическом порядке, включает в основном книги по криптографии и ее применению. Большинство книг доступно на русском языке.

1. Шеннон К. Теория связи в секретных системах. Работы по теории информации и кибернетике. – М.: ИЛ, 1963.

2. Kahn D. The Codebreakers, the story of secret writing. – NY.: Mc Millan, 1967.

3. Diffie W., Hellman M.F. New direction in cryptography. IEEE IT-22, 1976.

4. Mercle R. Secure communication over insecure channels. Comm. ACM, v. 21, № 4, 1978.

5. Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and public key cryptosystems. Comm. ACM, v/ 21, № 2, 1978.

6. Konheim A. Cryptography, a primer. J. Wiley & sons. Inc., 1981. – 432 p.

7. Denning D. Cryptography and data security. Addison – Weslay Publishing Company. 1982. – 400 p.

8. Kranakis E. Primality and Cryptography. J. Wiley & Sons. 1985.

9. Brassard G. Modern Cryptology. Springer Verlag. 1988. – 110 p.

10. ТИИЭР т. 76, № 5. Защита информации. Малый тематический выпуск. – М.: Мир, 1988.

11. Саломаа А. Криптография с открытым ключом. – М.: Мир, 1996.

12. Simmons G. (Ed.). Contemporary cryptology: the science of information integrity. 1992. – 640 p.

13. Спесивцев А.В., Вегнер В.А. и др. Защита информации в персональных ЭВМ. – М.: Радио и связь, 1992. – 191 с.

14. Jackson K., Hruska J. (Ed.) Computer Security Reference Book. Butterworth-Heinemann Ltd., 1992. – 932 p.

15. Фролов Г. Тайны тайнописи. – М.: Информсервис, 1992.

16. Мафтик С. Механизмы защиты в сетях ЭВМ. – М.: Мир, 1993.

17. Stinson D. Cryptography: theory and practice. CRC Press. 1995.

18. Hoffman L. (Ed.) Building in big Brother: the cryptography policy debate. Springer-Verlag NewYork. Inc., 1995. – 560 pp.

19. Варфоломеев А.А., Пеленицин М.Б. Методы криптографии и их применение в банковских технологиях. – М.: МИФИ, 1995. – 116 стр.

20. Варфоломеев А.А., Гаврилкевич М.В., Устюжанин Д.Д., Фомичев В.М. Методические указания к выполнению лабораторного практикума «Информационная безопасность. Криптографические методы защиты информации», ч. 1, ч. 2. – М.: МИФИ, 1995.

21. Горохов П.К. Информацинная безопасность. Англо-русский словарь. – М.: Радио и связь, 1995. – 224 с.

22. Garfinkel S. PGP: Pretly Good Privacy. – O'Reilly and Associates, Inc., 1995.

23. Menezes A., Van Oorschot P., Vanstone S. Handbook of Applied cryptography. CRC Press, 1996. – 816 стр. http://www.cacr.math.uwaterloo.ca/hac/

24. Варфоломеев А.А., Домнина О.С., Пеленицин М.Б. Управление ключами в системах криптографической защиты банковской информации. – М.: МИФИ, 1996.

25. Анохин М.И., Варновский Н.П., Сидельников В.М., Ященко В.В. Криптография в банковском деле. – М.: МИФИ, 1997. – 274 с.

26. Герасименко В.А., Малюк А.А. Основы защиты информации. – М.: МИФИ, 1997. – 538 с.

27. Варфоломеев А.А., Жуков А.Е. и др. Блочные криптосистемы. Основные свойства и методы анализа стойкости. – М.: МИФИ, 1998. – 198 с.

28. Введение в криптографию. Под общ. Ред. В.В. Ященко. – М.: МЦНМО, «ЧеРо», 1998. – 272 с.

29. Варфоломеев А.А., Жуков А.Е., Пудовкина М.А. Поточные криптосистемы. Основные свойства и методы анализа стойкости. – М.: МИФИ, 2000. – 268 с.

30. Петров А.А. Компьютерная безопасность. Криптографические методы защиты. – М.:ДМК, 2000. – 448 с.

31. Варфоломеев А.А., Запечников С.В., Маркелов В.В., Пеленицын М.Б. Интелектуальные карты и криптографические особенности их применения в банковском деле. – М.: МИФИ, 2000. – 188 с.

32. Ростовцев А.Г. Алгебраические основы криптографии. – СПб: Мир и Семья, 2000.

33. Ростовцев А.Г., Маховенко Е.А. Введение в криптографию с открытым ключом. – СПб.: Мир и Семья, 2000.

34. Анин Б. Защита компьютерной информации. – СПб.: БХВ-Санкт-Петербург, 2000. – 384 с.

35. Петров А.А. Компьютерная безопасность. Криптографические методы защиты. – М.:ДМК, 2000. – 448 с.

36. Голдовский И. Безопасность платежей в Интернете. – СПб: Питер, 2001. – 240 с.

37. Малюк А.А., Пазизин С.В., Погожин Н.С. Введение в защиту информации в автоматизированных системах – М.: Горячая линия – Телеком, 2001. – 148 с.

38. Коблиц Н. Курс теории чисел и криптография. – М.: ТВП, 2001.

39. Столлингс В. Криптография и защита сетей: принципы и практика, 2-е изд. – М.: Вильямс, 2001. – 672 с.

40. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии: Учебное пособие. – М.: Гелиос АРБ, 2001. – 480 с.

41. Харин Ю.С., Агиевич С.В. Компьютерный практикум по математическим методам защиты информации. – Мн.: БГУ, 2001. – 190 с.

42. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Триумф, 2002. – 816 с.

43. Бернет С., Пэйн С. Криптография. Официальное руководство RSA Security. – М.: Бином-Пресс, 2002. – 384 с.

44. Пярин В.А., Кузьмин А.С., Смирнов С.Н. Безопасность электронного бизнеса. – М.: Гелиос АРВ, 2002. – 432 с.

45. Смит Р. Аутентификация: от паролей до открытых ключей. – М.: Вильямс, 2002. – 432 с.

46. Чмора А.Л. Современная прикладная криптография. 2-е изд., стер. – М.: Гелиос АРВ, 2002. – 256 с.

47. Мамаев М., Петренко С. Технологии защиты информации в Интернете. Специальный справочник. – СПб.: Питер, 2002. – 848 с.

48. Снытников А.А. Лицензирование и сертификация в области защиты информации. – М.: Гелиос АРВ, 2003. – 192 с.

49. Шнайер Б. Секреты и ложь. Безопасность данных в цифровом мире. – СПб, 2003.

50. Василенко О.Н. Теоретико-числовые алгоритмы в криптографии. – М.: МЦНМО, 2003. – 328 с.

51. Масленников М.Е., Практическая криптография. – СПб.: БХВ-Петербург, 2003. – 464 с.

52. Фомичев В.М., Дискретная математика и криптология. – М.: ДИАЛОГ-МИФИ, 2003.

53. Болотов А.А., Гашков С.Б., Фролов А.Б. Алгоритмические основы эллиптической криптографии, 2003. – 526 с.

54. Харин Ю.С., Берник В.И., Матвеев Г.В., Агиевич С.В. Математические и компьютерные основы криптологии. – Мн.: Новое знание, 2003. – 382 с.

55. Вельшенбах М., Криптография на Си и Си++ в действии. – М.: Триумф, 2004.

56. Максим М., Полино Д. Безопасность беспроводных сетей. – М.: Компания АйТи, ДМК Пресс, 2004. – 288 с.

57. Зубов А.Ю. Криптографические методы защиты информации. Совершенные шифры. – М.: Гелиос АРВ, 2005.

58. Молдовян А.А., Молдовян Н.А. Введение в криптографию с открытым ключом. – СПб.: BHV-Санкт-Петербург, 2005. – 288 с.

59. Фергюссон Н., Шнайер Б. Практическая криптография. – Издательский дом «Вильямс», 2005. – 424 с.

60. Венбо Мао. Современная криптография: теория и практика: Пер. с англ. – М.: Вильямс, 2005. – 768 с.

61. Смарт Н. Криптография. – М.: Техносфера, 2005. – 528 с.

62. Земор Ж. Курс криптографии. – М.-Ижевск: НИЦ «Регулярная и хаотическая динамика»; Институт компьютерных исследований, 2006. – 256 с.

63. Тилборг Ван Х.К.А. Основы криптологии: Профессиональное руководство и интерактивный учебник. – М.; Мир, 2006. – 471 с.

64. Словарь криптографических терминов / Под ред. Б.А. Погорелова и В.Н. Сачкова. – М.: МЦНМО, 2006. – 94 с.

65. Болотов А.А., Гашков С.Б., Фролов А.Б. Протоколы криптографии на эллиптических кривых: Элементарное введение в эллиптическую криптографию. 2006. – 280 с.

66. Handbook of elliptic and hyperelliptic curve cryptography, Taylor & Francis Group, Scientific editors Henri Cohen & Gerard Frey, 2006. – 808 p.

67. Mangard S., Oswald E., Popp T. Power Analysis attacks, Revealing the Secrets of Smart Cards, Springer, 2007. – 337 p.

68. Торстейнсон П., Ганеш Г.А. Криптография и безопасность в технологии.NET. – М.: БИНОМ, 2007. – 479 с.

69. Бабаш А.В., Шанкин Г.П. Криптография. – М.: СОЛОН-ПРЕСС, 2007. – 512 с.

70.Запечников С.В. Криптографические протоколы и их применение в финансовой и коммерческой деятельности. – М.: Горячая линия – Телеком, 2007. – 320 с.

71. Зубов А.Ю. Математика кодов аутентификации. – М.: Гелиос АРВ, 2007. – 480с.

72. Song Y. Yan, Cryptanalytic Attacks on RSA. Springer. 2008. – 270 p.

ОПИСАНИЕ КУРСА И ПРОГРАММА

1.1. Цели и задачи курса

Основная цель курса:

обеспечить комплексность и полноту подготовки бакалавриата по направлениям «Информационные технологии», «Прикладная математика и информатика», «Математика. Прикладная математика», «Автоматизация и управление» путем формирования у студентов знаний и навыков в области современной криптографии и ее приложениям для обеспечения безопасности новых информационных технологий.

Задачи курса:

ознакомить студентов с основными проблемами в области создания и анализа средств криптографической защиты информации, а также с методами и средствами их решения; обеспечить необходимыми сведениями и навыками для последующего обучения по направлениям «Информационные технологии», «Прикладная математика и информатика», «Математика. Прикладная математика», "Автоматизация и управление".

1.2. Профессиональные знания, умения и навыки, приобретаемые в результате изучения курса

Знания

Студент должен знать:

- основные понятия и задачи криптографии;
- типовые методы криптографического анализа и оценивания криптографической стойкости;

- классификацию угроз криптографическим средствам защиты информации;

- проблемы и методы управления ключевым материалом криптосистем;

- принятые отечественные, зарубежные и международные стандарты криптографических примитивов и протоколов и рекомендации по их использованию;

- тенденции развития и основные направления исследований в области криптографии;

- о развитии современных вычислительных и технических средств, реализующих криптографические средства защиты;

- о системах международных и отечественных стандартов, и организациях по стандартизации в области криптографической защиты информации;

- о вопросах лицензирования и сертификации криптографических средств;

- методы и средства криптографической защиты информации и контроля широко используемых информационных технологий;

Навыки и умения

Студент должен уметь:

- применять криптографические средства защиты информации в различных областях деятельности;

- разрабатывать новые криптографические протоколы, удовлетворяющие различным требованиям информационной безопасности и способы выбора систем защиты.

Студент должен иметь навыки:

- оценки качества криптографических стандартов и протоколов;

- управления ключевым материалом криптографических стандартов.

1.3. Инновационность курса по содержанию, методике преподавания, литературе, организации учебного процесса.

Инновационность курса по содержанию и литературе определяется в первую очередь тем, что он основан на изучении последних самых современных направлениях в криптографии, которые выделены на основе анализа тематики конференций и семинаров по криптографии. К этим направлениям относятся

доказательства с нулевым разглашением знаний;

пороговая криптография;

безопасное извлечение информации;

схемы электронных платежей;

безопасные многосторонние вычисления;

криптосистемы, основанные на эллиптических кривых;

криптосистемы, основанные на задачах о целочисленных решётках;

схемы электронного голосования;

схемы битовых обязательств;

забывающая передача;

электронная дактилоскопия, электронные водяные знаки;

электронные аукционы;

справедливый обмен секретами;

доказательная безопасность;

применение криптологии в электронной коммерции;

квантовая криптография;

проблемы анонимности и неотслеживаемости;

криптосистемы на основе отображений спаривания;

криптосистемы с открытым ключом без сертификатов;

другие разделы.

Конечно, должное место уделяется и традиционным направлениям, где в последнее время были получены важные практические результаты:

теория однонаправленных функций,

псевдослучайные функции и псевдослучайные генераторы;

схемы шифрования с секретным ключом;

схемы шифрования с открытым ключом;

схемы электронной цифровой подписи;

хэш-функции;

схемы идентификации и аутентификации;

протоколы распределения ключей;

методы управления криптографическими ключами;

проблемы факторизации и дискретного логарифмирования,

другие трудноразрешимые проблемы;

булевы функции и S-блоки;

эффективная реализация криптографических механизмов;

Большое внимание в курсе уделено именно прикладным вопросам криптографии, без глубокого изучения теоретических, математических результатов, влияющих на стойкость криптосистем. Это позволяет экономить учебное время и перераспределить его в пользу практики, а также увеличить число изучаемых тем. В курсе больше рассматривается влияние теоретических результатов на практический выбор параметров криптосистем, обеспечивающих их высокую криптостойкость, чем методы и строгость доказательства этих результатов.

В последнее время за рубежом и у нас вышли достаточно хорошие учебные пособия по криптографии. Список их можно найти в отчете. Но все они по разному подходят к выбору и изложению материала. Поэтому выбор разделов и их согласование является достаточно творческой задачей для преподавателя. К сожалению, не все нужные книги переведены на русский язык к настоящему времени. (Например, Stinson D. Cryptography: theory and

practice. CRC Press, 1995; Menezes A., Van Oorschot P., Vanstone S. Handbook of Applied cryptography. CRC Press, 1996.- 816 стр.) Поэтому, при составлении курса пришлось перевести с английского языка большое количество материала.

Инновационность курса определяется и методикой корректировки изучаемых тем по запросам специалистов для работы в зарубежных фирмах и высших учебных заведениях. Такие запросы можно найти на сайте Международной ассоциации криптографических исследований (www.iacr.org). Такой подход должен повысить востребованность подготавливаемых в ВУЗе специалистов на рынке труда.

Инновационность курса в методике преподавания и организации учебного процесса связана с методикой выполнения рефератов и курсовых работ. Суть ее заключается в случайном разбиении студентов на рабочие группы по 3-4 человека, с назначением руководителя группы. Каждой группе назначается одна из актуальных тем курса, полученная в результате анализа последних публикаций по криптографии. Этим достигается выполнение принципа коллективизма при работе над темой, что имеет место в большинстве организаций и предприятий. Случайный выбор членов рабочий группы тоже позволит каждому проявить и развить коммуникабельность, умение работать в коллективе. Кроме того, предполагается работа над каждой темой двух случайных групп. Этим обеспечивается принцип конкуренции между группами. Руководитель группы должен отразить участие каждого члена группы в пояснительной записке к отчету по теме.

1.4. Структура курса.

Общая трудоемкость – 136 часов за 2 семестра (2 кредита за семестр)

Виды учебных работ	**Объем работ, час.**		
	Всего	**7 сем.**	**8сем.**
Выделено на дисциплину	136	51	85
Аудиторная работа:	85	34	51
- лекции	68	34	34
- семинары	-	-	-
- лабораторные занятия	17	-	17
Внеаудиторная работа:	51	17	34
- курсовая работа	-	-	-
- курсовой проект	-	-	КП
- самостоятельное изучение разделов	51	17	34
Виды отчетности по дисциплине:			
- зачет	-	-.	-
- экзамен	-	экз.	экз.

Темы лекций.

Тема 1. Основные понятия и определения криптографии.

Тема 2. Этапы развития криптографии. Роль математики в развитии методов защиты информации. Новые направления в криптографии.

Тема 3. Криптографические примитивы и криптографические протоколы по защите информации.

Тема 4. Двухсторонние и многосторонние протоколы. Типы предполагаемых противников. Формальные методы оценки качества криптографических протоколов.

Тема 5. Шифры. Примеры. Стойкость шифра. Классификация методов дешифрования. Шифрующие автоматы. Типовые узлы. Регистры сдвига с обратной связью. Линейные последовательностные машины.

Тема 6. Блочные и поточные криптосистемы и их классификация. Описание DES - AES, ГОСТ 28147-89, RC4 и др. Режимы использования и их сравнение (ECB,CBC, OFB, …).

Тема 7. Криптографические свойства функций.

Тема 8. Теория информации и криптография. Совершенная секретность по Шеннону.

Тема 9. Теория сложности вычислений и криптография. Используемые в криптографии задачи теории сложности и их оценка.

Тема 10. Основные понятия криптографии с открытым ключом. Сравнение криптосистем с открытым и секретным ключом.

Тема 11. Однонаправленные (односторонние) функции по Нидхэму. Однонаправленные функции, основанные на сложности задачи дискретного логарифмирования. Применения в современных технологиях.

Тема 12. Однонаправленные (односторонние) функции с секретом и их применение для цели шифрования информации. Схемы RSA, Рабина, Эль Гамаля, МакЭлайса, Меркля – Хеллмана.

Тема 13. Понятия о цифровой подписи на основе однонаправленной функции с секретом. Классификация атак на схемы цифровой подписи.

Тема 14. Некоторые методы быстрой модульной арифметики и их применение для ускорения криптографических алгоритмов.

Тема 15. Сравнение стандартов цифровой подписи США (FIPS PUB 186) и России (ГОСТ Р 34.10-94). Стандарт цифровой подписи ГОСТ Р 34.10-2001 на основе эллиптических кривых.

Тема 16. Схемы подписи Фиата-Шамира, Файге-Фиата-Шамира и др. Схема Шнорра.

Тема 17. Подпись вслепую (blind signature) и ее применения.

Тема 18. Схемы конфиденциальной подписи (undeniable signature) и их применение. Протоколы проверки и отвержения как примеры протоколов доказательств с нулевым разглашением. Схемы Шаума.

Тема 19. Схемы подписи, в которых подделка подписи может быть доказана.

Тема 20 Схемы мультиподписи (multisignature scheme).

Тема 21. Групповая подпись (group signature scheme).

Тема 22. Подпись по доверенности (proxy signature).

Тема 23. Функции хэширования.

Тема 24. Американский стандарт функции хэширования (SHS) и его изменения. Российский стандарт функции хэширования (ГОСТ Р 34.11-94).

Тема 25. Управление ключами. Доказуемо безопасные генераторы ключей. Некоторые способы сокращения объемов хранимых ключей.

Тема 26. Протоколы распределения криптографических ключей.

Тема 27. Криптографическая инфраструктура на основе механизма открытых ключей(PKI). Модели криптографической инфраструктуры.

Тема 28. Протоколы, основанные на идентификационной информации (ID-based cryptosystems).

Тема 29. Протоколы с разделением секрета. Пороговые схемы.

Тема 30. Криптосистемы и протоколы на эллиптических кривых.

Тема 31. Протоколы идентификации и аутентификации.

Тема 32. Протоколы честного обмена секретами.

Тема 33. Интерактивные схемы доказательств

Тема 34. Протоколы электронного тайного голосования .

Тема 35. Понятие о протоколах электронных платежей

Тема 36. Вопросы стандартизации и патентования

Система контроля знаний:

Форма итогового контроля знаний по дисциплине – экзамен.

ПРОГРАММА КУРСА УМК

2.1. Аннотированное содержание курса

Изучение данного курса обеспечивает студента сведениями о современном состоянии в области прикладной криптографии. Курс существенно расширяет и углубляет знания, полученные студентами при изучении дисциплины «Основы информационной безопасности». Материал курса основан на последних достижениях зарубежных и отечественных специалистов – криптографов как в классических областях применения криптографии, так и в новых, связанных с новыми информационными технологиями.

Существенное место в курсе уделено и стандартным методам и рекомендациям криптографической защиты информации, позволяющим существенно ускорить разработку и внедрение новых систем.

В читаемой дисциплине излагаются:

- блочные и поточные криптосистемы и их классификация;
- криптографические свойства функций;
- криптографические применения теории сложности вычислений;
- основные понятия криптографии с открытым ключом;
- разновидности протоколов электронной цифровой подписи;
- функции хэширования;
- проблемы управления ключами;
- криптосистемы и протоколы на эллиптических кривых;
- протоколы идентификации и аутентификации;
- протоколы честного обмена секретами;
- интерактивные схемы доказательств с нулевым разглашением;

- протоколы электронного тайного голосования;

- протоколы электронных платежей;

- вопросы стандартизации и патентования;

- необходимые сведения из алгебры и теории чисел.

Для самостоятельной работы студентов, подготовки рефератов и курсовых проектов выбираются темы из перечня тем известных отечественных и международных конференций и семинаров по криптографии.

Лабораторные работы выполняются студентами по заданиям, подобранных преподавателем на основе следующих разработок:

Харин Ю.С., Агиевич С.В. Компьютерный практикум по математическим методам защиты информации. – Мн.: БГУ, 2001. – 190 с.

Варфоломеев А.А., Гаврилкевич М.В., Устюжанин Д.Д., Фомичев В.М. Методические указания к выполнению лабораторного практикума “Информационная безопасность. Криптографические методы защиты информации” Часть 1(-44с.), Часть 2(-40 с.). М., МИФИ, 1995.

2.2. Список обязательной и дополнительной литературы для преподавателей.

Литература к каждому разделу учебного тематического плана будет указана после каждого раздела с перечислением конкретных страниц источника. Нумерация источников будет соответствовать приведенной ниже нумерации.

Обязательная:

1. Шеннон К. Теория связи в секретных системах. Работы по теории информации и кибернетике. – М.: ИЛ, 1963.

2. Diffie W., Hellman M.F. New direction in cryptography. IEEE IT-22. 1976.

3. Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and public key cryptosystems. Comm. ACM, v/ 21, №2, 1978.

4. Konheim A. Cryptography, a primer. J. Wiley & sons. Inc., 1981.- 432 pp.

5. Denning D. Cryptography and data security. Addison- Weslay Publishing Company. 1982,- 400 pp.

6. Kranakis E. Primality and Cryptography. J. Wiley & sons, 1985.

7. Koblitz N. A Course in Number theory and cryptography. Springer – Verlag NewYork. Inc., 1987.- 208 pp.

Русский перевод: Коблиц Н. Курс теории чисел и криптография: - М.: ТВП, 2001.-432 с.

8. Brassard G. Modern Cryptology. Springer Verlag. 1988,- 110 pp.

9. Salomaa A. Public – Key Cryptography. Springer -Verlag, 1990.

Русский перевод: Саломаа А. Криптография с открытым ключом. – М.: Мир, 1996.

10. Simmons G. (Ed.). Contemporary cryptology: the science of information integrity. 1992.- 640 pp.

11. Спесивцев А.В., Вегнер В.А. и др. Защита информации в персональных ЭВМ.- М.: Радио и связь, 1992.- 191 стр.

12. Rhee Man Joung. Cryptography and Secure Communications, MC Graw – Hill Book Co., 1994.

13. Stinson D. Cryptography: theory and practice. CRC Press, 1995.

14. Hoffman L. (Ed.) Building in big Brother: the cryptography policy debate. Springer-Verlag NewYork. Inc. , 1995.- 560 pp.

15. Wayner P. Digital Cash, AP Professional, 1995.

16. Варфоломеев А.А., Пеленицин М.Б. Методы криптографии и их применение в банковских технологиях. – М.: МИФИ, 1995.- 116 стр.

17. Варфоломеев А.А., Гаврилкевич М.В., Устюжанин Д.Д., Фомичев В.М. Методические указания к выполнению лабораторного практикума “Информационная безопасность. Криптографические методы защиты информации”, ч.1 ,ч.2. – М.: МИФИ, 1995. – 44 с., - 38с.

18. Menezes A., Van Oorschot P., Vanstone S. Handbook of Applied cryptography. CRC Press, 1996.- 816 стр.

19. Schneier B. Applied cryptography, second edition: protocols, algoritums, and source code in C. J. Wiley & sons, Inc. 1996.- 758 pp.

Русский перевод: Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си.- М.: Триумф, 2002. – 816 с.

20. Анохин М.И., Варновский Н.П., Сидельников В.М., Ященко В.В. Криптография в банковском деле. М.: МИФИ. 1997.- 274 стр.

21. Варфоломеев А.А., Жуков А.Е. и др. Блочные криптосистемы. Основные свойства и методы анализа стойкости. М.: МИФИ, 1998.- 198 стр.
Введение в криптографию. Под общ. Ред. В.В. Ященко.- М.: МЦНМО, «ЧеРо», 1998.- 272 с.

22. Stallings W. Network and Internetwork Security: principles and practice, Second Edition, Prentice-Hall, Inc., 1999.- 459 pp.

Русский перевод: Столлингс В. Криптография и защита сетей: принципы и практика, 2-е изд. – М.: Вильямс, 2001. – 672 с.

23. Петров А.А. Компьютерная безопасность. Криптографические методы защиты. – М.:ДМК, 2000. – 448 с.

24. Ростовцев А.Г. Алгебраические основы криптографии. – СПб: Мир и Семья, 2000.

25. Ростовцев А.Г., Маховенко Е.А. Введение в криптографию с открытым ключом. – СПб.: Мир и Семья, 2000.

26. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии: Учебное пособие. – М.: Гелиос АРБ, 2001. –480 с.

27. Burnet S., Paine S. RSA Security`s Official Guide to Cryptography.- NY.: The McGraw-Hill Companies, 2001.

Русский перевод: Бернет С., Пэйн С. Криптография. Официальное руководство RSA Security.- М.: Бином-Пресс, 2002. –384 с.

28. Харин Ю.С., Агиевич С.В. Компьютерный практикум по математическим методам защиты информации. – Мн.:БГУ, 2001. – 190 с.

29. Пярин В.А., Кузьмин А.С., Смирнов С.Н. Безопасность электронного бизнеса. – М.: Гелиос АРБ, 2002. – 432 с.

30. Smith R. Authenticaton: From Passwords to Public Keys. – NY: Addison-Wesley Publishing Company, Inc., 2002.

Русский перевод: Смит Р. Аутентификация: от паролей до открытых ключей. –М.: Вильямс, 2002. – 432 с.

31. Чмора А.Л. Современная прикладная криптография. 2-е изд., стер. – М.: Гелиос АРБ, 2002. – 256 с.

32. Василенко О.Н. Теоретико-числовые алгоритмы в криптографии. – М.: МЦНМО, 2003. – 328 с.

33. Масленников М.Е., Практическая криптография, -СПб.: БХВ-Петербург, 2003.-464 с.

34. Фомичев В.М., Дискретная математика и криптология. - М.: ДИАЛОГ - МИФИ, 2003.

35. Болотов А.А., Гашков С.Б., Фролов А.Б. Алгоритмические основы эллиптической криптографии, 2003.-526 стр.

36. Вельшенбах М., Криптография на Си и Си++ в действии. -М.: Триумф, 2004.

37. Зубов А.Ю. Криптографические методы защиты информации. Совершенные шифры. – М.: Гелиос АРВ, 2005.

38. Фергюссон Н., Шнайер Б. Практическая криптография. – Издательский дом «Вильямс», 2005.-424 с.

39. Венбо Мао. Современная криптография: теория и практика.: Пер. с англ.- М.: Вильямс, 2005. 768 с.

40. Смарт Н. Криптография. М.: Техносфера, 2005.- 528 с.

41. Земор Ж. Курс криптографии.- М.-Ижевск: НИЦ"Регулярная и хаотическая динамика"; Институт компьютерных исследований, 2006.-256.

42. Тилборг Ван Х.К.А. Основы криптологии. Профессиональное руководство и интерактивный учебник. – М.; Мир, 2006, 471 с.

43. Словарь криптографических терминов/ Под ред. Б.А. Погорелова и В.Н. Сачкова. – М.: МЦНМО, 2006.- 94 с.

44. Болотов А.А., Гашков С.Б., Фролов А.Б. Протоколы криптографии на эллиптических кривых: Элементарное введение в эллиптическую криптографию. 2006. - 280 с.

45. Handbook of elliptic and hyperelliptic curve cryptography, Taylor & Francis Group, Scientific editors Henri Cohen & Gerard Frey, 2006. – 808 стр.

46. Зубов А.Ю. Математика кодов аутентификации. – М.: Гелиос АРБ, 2007.- 480с.

Дополнительная литература

47. Hoffman L. Modern methods for computer security and privacy. Prentice-Hall,Inc.,1977.

Русский перевод: Хоффман Л.Д.Современные методы защиты информации. - М.: Сов. радио, 1980.

48. ТИИЭР т.76, №5. Защита информации. Малый тематический выпуск. – М.: Мир, 1988.

49. Seberry J., Pieprcyk J. Cryptography: An Introduction to computer security, Prentce Hall, Inc., 1989.

50. Russel D., G.T.Gangemi Sr. Computer Security Basics. –O`Reilli & Associates, Inc., 1991. – 448 pp.

51. Jackson K., Hruska J. (Ed.) Computer Security Reference Book. Butterworth-Heinemann Ltd., 1992. - 932 pp.

52. Muftic S. Security Mechanisms for Computer Networks. Halsted Press.

Русский перевод: Мафтик С. Механизмы защиты в сетях ЭВМ. – М.: Мир, 1993.

53. Расторгуев С.П. Программные методы защиты информации в компьютерах и сетях. –М.: Яхтсмен, 1993.- 188 с.

54. Горохов П.К. Информацинная безопасность. Англо-русский словарь. – М.: Радио и связь, 1995. 224 с.

55. Garfinkel S. PGP: Pretly Good Privacy. – O'Reilly and Associates, Inc., 1995.

56. Kaufman C., Pelman R., Speciner M. Network Security – PRIVATE Communication in a PUBLIC World, Prentice-Hall, Inc.,1995.

57. Герасименко В.А., Малюк А.А. Основы защиты информации. – М.: МИФИ, 1997 г., - 538 с., учебник (рекомендован Минобразованием России в качестве учебника для студентов ВУЗов).

58. Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. – М.:Радио и связь, 1999. – 328 с.

59. Варфоломеев А.А., Запечников С.В., Маркелов В.В., Пеленицын М.Б. Интелектуальные карты и криптографические особенности их применения в банковском деле. М.: МИФИ. 2000.- 188 стр.

60. Зегжда Д., Ивашко А.М. Основы безопасности информационных систем. – М.: Горячая линия – Телекомю 2000. – 452 с.

61. Устинов Г.Н. Основы информационной безопасности систем и сетей передачи данных. Учебное пособие. М.: СИНТЕГ, 2000, -248 с.

62. Голдовский И. Безопасность платежей в Интернете. – СПб : Питер, 2001. – 240 с.

63. Малюк А.А., Пазизин С.В., Погожин Н.С. Введение в защиту информации в автоматизированных системах – М.: Горячая линия-телеком, 2001г.,-148 с.

64. Безопасность сети на основе Windows 2000. Учебный курс MCSE. – М.: ИТД Русская Редакция. 2001. –912 с.

65. Мамаев М., Петренко С. Технологии защиты информации в Интернете. Специальный справочник. – СПб.: Питер, 2002. – 848 с.

66. Снытников А.А. Лицензирование и сертификация в области защиты информации. – М.: Гелиос АРВ, 2003.- 192 с.

67. Новиков А.А., Устинов Г.Н. Уязвимость и информационная безопасность телекоммуникационных технологий: Учеб. пособие. – М.: Радио и связь. 2003.- 296 с.

68. Конеев И., Беляев А. Информационная безопасность предприятия. - СПб.: БХВ-Петербург, 2003.-752с.(Часть 5. Криптография, 209-364 стр.)

69. Шнайер Б. Секреты и ложь. Безопасность данных в цифровом мире. - СПб: 2003.
http://lib.aldebaran.ru/author/shnaier_bryus/shnaier_bryus_sekrety_i_lozh_bezopasnost_dannyh_v_cifrovom_mire/

70. Скляров Д.В., Искусство защиты и взлома информации.- СПб.: БХВ-Петербург, 2004.-288 с.

71. Максим М., Полино Д. Безопасность беспроводных сетей. – М.: Компания АйТи, ДМК Пресс, 2004. – 288 с.(пер. книги 2002 изд.)

72. Белов Е.Б., Лось В.П., Мещеряков Р.В., Шелупанов А.А. Основы информационной безопасности. Учебное пособие для вузов, М.: Горячая линия – Телеком, 2006.- 544 с.

73. Mangard S., Oswald E., Popp T., Power Analysis attacks, Revealing the Secrets of Smart Cards, Springer, 2007, - 337 стр.

74. Сердюк В.А. Новое в защите от взлома корпоративных систем. М.: Техносфера, 2007.- 360 с.(2.1.1. Средства криптографической защиты информации, 66-70 стр.)

2.3. Список обязательной и дополнительной литературы для студентов.

Обязательная:

16. Варфоломеев А.А., Пеленицин М.Б. Методы криптографии и их применение в банковских технологиях. – М.: МИФИ, 1995.- 116 стр.

19. Schneier B. Applied cryptography, second edition: protocols, algoritums, and source code in C. J. Wiley & sons, Inc. 1996.- 758 pp.

Русский перевод: Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си.- М.: Триумф, 2002. – 816 с.

38. Фергюссон Н., Шнайер Б. Практическая криптография. – Издательский дом «Вильямс», 2005.-424 с.

39. Венбо Мао. Современная криптография: теория и практика.: Пер. с англ.- М.: Вильямс, 2005. 768 с.

40. Смарт Н. Криптография. М.: Техносфера, 2005.- 528 с.

42. Тилборг Ван Х.К.А. Основы криптологии. Профессиональное руководство и интерактивный учебник. – М.; Мир, 2006, 471 с.

Дополнительная:

26. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии: Учебное пособие. – М.: Гелиос АРВ, 2001. –480 с.

31. Чмора А.Л. Современная прикладная криптография. 2-е изд., стер. – М.: Гелиос АРБ, 2002. – 256 с.

34. Фомичев В.М., Дискретная математика и криптология. - М.: ДИАЛОГ - МИФИ, 2003.

41. Земор Ж. Курс криптографии.- М.-Ижевск: НИЦ”Регулярная и хаотическая динамика”; Институт компьютерных исследований, 2006.-256.

2.4. Темы рефератов, курсовых работ, эссе

Темы для самостоятельной работы студентов при подготовке рефератов и курсовых работ определяются на основе тематики известных признанных научных конференций и семинаров, рассматривающих вопросы современной криптографии. Например, среди мероприятий 2006 года можно отметить следующие:

- SASC 2006 - Stream Ciphers Revisited, February 2-3, Leuven, Belgium.
- Workshop on Mathematical Techniques in Cryptology (WMTC-2005), February 10-12, Mathura, INDIA.
- RSA Conference 2006, Cryptographers' Track, February 13-17, San Jose, CA, USA.
- German Workshop "Applied Cryptography", February 20-23, Magdeburg, Germany.
- Workshop "Kryptographie in Theorie und Praxis", February 20-23, Magdeburg, Germany.
- 10th International Conference on Financial Cryptography and Data Security (FC06), February 27-March 2, Anguilla, British West Indies.
- The third Theory of Cryptography Conference (TCC'06), March 5-7, New York, United States.
- International Workshop on Boolean Functions : Cryptography and Applications, March 13-15, Rouen, FRANCE.
- Fast Software Encryption 2006, March 15-17, Graz, Austria.
- SHARCS'06 - Special-purpose Hardware for Attacking Cryptographic Systems, April 3-4, Cologne, Germany.
- 5th Annual PKI R&D Workshop, April 4-6, Gaithersburg, USA.
- 7th Smart Card Research and Advanced Application Conference (CARDIS'2006), April 19-21, Tarragona, Catalonia, Spain.
- Ninth International Workshop on Practice and Theory in Public Key Cryptography (PKC 2006), April 24-26, New York, USA.

- Workshop on Gröbner Bases in Cryptography, Coding Theory, and Algebraic Combina, May 1-6, Linz, Austria.
- Applied Cryptography and Information Security 06, May 8-11, Glasgow, UK.
- Workshop on Cryptology and Information Hiding (M15), May 22-26, Ragusa (Sicily), Italy.
- International Workshop on Post-Quantum Cryptography, May 23-26, Leuven, Belgium.
- Quo Vadis Cryptology 4: The Future of Financial and Critical Data Security, May 26-26, Warsaw, Poland.
- Eurocrypt 2006, May 28-June 1, St. Petersburg, Russia.
- 4th International Conference on Applied Cryptography and Network Security, June 6-9, Singapore, Singapore.
- Third European PKI workshop: theory and practice, June 19-20, Turin, Italy.
- Summer School on Computational Number Theory and Applications to Cryptography, June 19-July 7, Laramie, USA.
- Yet Another Conference in Cryptography, June 19-23, Porquerolles, France.
- Workshop on Recent Advances in Stream Ciphers and Hash Functions, June 26-30, Brisbane, Australia.
- Workshop On Trustworthy Elections, June 29-30, Cambridge, UK.
- Workshop on Mathematical Cryptology, June 29-30, Santander, Spain.
- IAVoSS Workshop On Trustworthy Elections, June 29-30, Cambridge, UK.
- Information Hiding 2006, July 10-12, Washington, DC, USA.
- 2nd Workshop on Cryptography for Ad hoc Networks (satellite of ICALP 06), July 16-16, San Servolo, Venice, Italy.
- crypt@b-it 2006, summer school on cryptography, July 17-21, Bonn, Germany.
- Algorithmic Number Theory Symposium VII (ANTS VII), July 23-28, Berlin, Germany.

- 15th USENIX Security Symposium, July 31-August 4, Vancouver, CANADA.
- Workshop on Models for Cryptographic Protocols, July 31-August 1, Århus, Denmark.
- International Conference on Security and Cryptography, August 7-10, Setubal, Portugal.
- 13th Annual Workshop on Selected Areas in Cryptography, August 17-18, Montreal, Canada.
- Crypto 2006, August 20-24, Santa Barbara (CA), USA.
- The Second Cryptographic Hash Workshop, August 24-25, Santa Barbara, USA.
- Second Cryptographic Hash Workshop, August 24-25, Santa Barbara, USA.
- Fall 2006 Thematic Program in Cryptography, September 1-1, Toronto, Canada.
- Security and Cryptography for Networks, September 6-8, Maiori, Italy.
- IX Spanish Meeting on Cryptology and Security of the Information, September 7-9, Barcelona, Spain.
- Securing Cyberspace: Application and Foundations of Cryptography and Computer Se, September 11-11, Los Angeles, USA.
- Tenth Workshop on Elliptic Curve Cryptography, September 18-20, Toronto, Canada.
- IAVoSS Workshop on Frontiers in Electronic Elections, September 19-19, Hamburg, Germany.
- Training on Smart Card & RFID Security - Smart University 06, September 19-20, Sophia-Antipolis, French Riviera, France.
- International Conference on SEQUENCES AND THEIR APPLICATIONS 2006, September 24-28, Beijing, China.
- International Conference on Cryptology in Vietnam 2006, September 25-28, Hanoi, Vietnam.

- Workshop on Codes and Lattices in Cryptography, September 25-27, Darmstadt, Germany.
- IJSN Special Issue on Cryptography in Networks, October 1-1, ., ..
- 20th Midwestern Conference on Combinatorics, Cryptography and Computing, October 5-7, Wichita, Kansas, U.S.A..
- IPA Cryptography Forum 2006, October 5-5, Tokyo, Japan.
- IPAM Workshop: Number Theory and Cryptography--Open Problems, October 9-13, Los Angeles, CA, USA.
- Workshop on Fault Diagnosis and Tolerance in Cryptography 2006, October 10-10, Yokohama, Japan.
- Mathematics and Security of Information Technologies, October 25-28, Moscow, Russia.
- International School on Zero Knowledge: Foundations and Applications, October 28-November 3, Bertinoro, Italy.
- IPAM Workshop: Foundations of Secure Multi-Party Computation and Zero-Knowledge, November 13-17, Los Angeles, CA, USA.
- Fifth International Workshop on Applied PKC (IWAP'06), November 27-29, Dalian, China.
- The 2nd SKLOIS Conference on Information Security and Cryptology, November 29-December 1, Beijing, China.
- The 9th Annual International Conference on Information Security and Cryptology, November 30-December 1, Busan, Korea.
- Asiacrypt 2006, December 3-7, Shanghai, China.
- IPAM Workshop: Special Purpose Hardware for Cryptography: Attacks and Applicatio, December 4-8, Los Angeles, CA, USA.
- The 5th International Conference on Cryptology and Network Security (CANS 2006), December 8-10, Suzhou, China.
- 7th International Conference on Cryptology in India (Indocrypt-2006), December 11-13, Kolkata, India.

- NATIONAL CRYPTOLOGY SYMPOSIUM II, December 15-17, ANKARA, TURKEY.

Труды большинства этих конференций и семинаров доступны для образовательных учреждений в сети Интернет на сайте http://www.springerlink.com/ . Для подготовки рефератов и тем курсовых работ в помощь студентам авторами подготовлена база с публикациями по темам.

Наименование раздела базы	Кол-во файлов в разделе	Размер раздела в МБ(айтах)
Auctions – Электронные аукционы	96	31,9
Block ciphers-Блочные шифры	63	18,5
Broadcast encryption- Широковещательное шифрование	8	1,76
Commitment Залоговые схемы	12	2,69
Complexity- Сложность вычислений	10	1,44
Conferences- Конференции по криптографии	123	7,33
Crypto courses- Курсы по криптографии	90	14,2
Crypto Policy- Политика в криптографии	14	1,87
Digital Signatures- Цифровые подписи	98	23,7
E_Commerce- Электронная коммерция и криптография	57	19,9
E_Lotteries- Электронные лотереи	18	3,64
Efficient Implementation- эффективная реализация	14	2,54

Electronic Payment Systems- Электронные платежные системы	82	26,0
Elliptic Curve Cryptography – Криптография на эллиптических кривых	201	47,3
Fair Exchange – Честный обмен секретами	13	2,49
Fingerprinting – Электронная дактилоскопия	24	4,16
Foundations of Cryptography – Основания криптографии	68	20,8
GSM security- Безопасность в мобильных сетях GSM	16	2,67
Handbook of Applied Cryptography Справочник по прикладной криптографии	21	6,13
Hard problems – Труднорешаемые задачи для криптографии	32	7,98
Hash functions –Функции хеширования	36	9,81
Lattices and Cryptography – Алгебраические решетки и криптография	39	9,37
Multiparty computations – Многосторонние безопасные вычисления	23	9,25
Oblivious Transfer – Забывающая передача	9	1,70
PIR- Private Information Retrieval – Безопасное ознакомление с информацией	32	10,0
PKC-Public Key Cryptography – Криптография с открытым ключом	133	34,2
PRNG – Псевдослучайные генераторы	102	23,6
Protocols – Криптографические протоколы	156	42,5
Quantum Cryptography – Квантовая криптография	18	3,83

Stream ciphers – Поточные шифры	61	11,9
Threshold cryptography – Пороговая криптография	26	7,05
Timestamping – Расстановка временных меток	6	1,22
Tracing Schemes -	15	3,14
Voting – Электронное голосование	30	6,48
Watermarking – электронные водяные знаки	39	21,3
Zero-Knowledge – нулевое разглашение секретов	23	9,30

Примеры тем рефератов, курсовых работ.

1. Криптографические средства защиты информации в стандарте GSM и их стойкость.
2. Исследование алгоритма поточного шифрования RC4.
3. Особенности применения цифровой подписи вслепую в протоколах электронного тайного голосования.
4. Новые американские стандарты режимов шифрования с аутентификацией.
5. Схемы криптосистем на основе парных отображений.
6. Методы эффективной реализации схем электронной цифровой подписи на основе группы точек эллиптических кривых.
7. Возможности преобразования отечественного стандарта цифровой подписи в схему цифровой подписи вслепую.
8. Сравнение криптографических средств различных протоколов мобильных платежей.
9. Исследование свойств подстановок на двоичных векторах при малых размерностях и их применение при построении узлов алгоритмов шифрования.
10. Решение проблемы повторной траты криптографическими методами в схемах электронных платежей.

2.5. Учебный тематический план курса «Современная прикладная криптография»

Введение.

Основные понятия и определения. Криптография или криптология. Информационная безопасность и криптография. Различные аспекты безопасности информации (секретность, целостность, аутентичность, неотказуемость, неотслеживаемость, …) и методы криптографии, обеспечивающие их выполнение при хранении и передаче информации в телекоммуникационных системах. Политика различных организаций в области защиты информации. Криптография и криптоанализ (дешифрование). Этапы развития криптографии. Роль математики в развитии методов защиты информации. Новые направления в криптографии. Криптографические примитивы и криптографические протоколы по защите информации. Классификация примитивов с открытым ключом. Протоколы с арбитром, с судьей, самодостаточные (self-enforcing) протоколы. Двухсторонние и многосторонние протоколы. Типы предполагаемых противников. Формальные методы оценки качества криптографических протоколов.

Литература к разделу: 19(15-36, 125-149, 661-688), 38(26-39, 268-281)

Понятие о шифрах.

Шифры перестановки и замены. Примеры. Стойкость шифра. Классификация методов дешифрования по информации, известной криптоаналитику. Ручные шифры. Электронные и механические реализации шифров. Модель предполагаемого нарушителя. Правила Керкхоффса. Понятие о криптографических протоколах.

Конечные автоматы.Эквивалентность конечных автоматов и их состояний. Шифрующие автоматы. Регистры сдвига с обратной связью над различными

алгебраическими структурами. Линейные последовательностные машины. Линейные рекуррентные последовательности.

Литература к разделу: 19(221-229), 38(40-57), 39(245-260), 42(19-34)

Блочные и поточные криптосистемы и их классификация.

Определения. Примеры. Описание DES - AES, ГОСТ 28147-89, RC4 и др. Режимы использования и их сравнение (ECB,CBC, OFB, …). Некоторые методы криптоанализа.

Литература к разделу: 19(221-246, 303-414, 445-480), 38(61-176), 39(261-289), 42(68-78)

Криптографические свойства функций

Равновероятность (равновесность). Свойство лавинного эффекта. Свойство строго лавинного эффекта порядка m. Совершенные нелинейные булевы функции. Множество булевых функций, обладающих линейными структурами. Понятие корреляционной независимости. Бент - функции. Свойство размазывания.

Строение и свойства S-блоков.

Литература к разделу: 19(324-325, 393-395), 34(125-154)

Теория информации и криптография.

Энтропия, условная энтропия. Совершенная секретность по Шеннону. Примеры. Шифр одноразового блокнота (Шифр Вернама). Практическая стойкость шифров. Рабочая характеристика. Расстояние единственности для не совершенно секретных шифров. Метод Хеллмана оценки расстояния единственности. Понятие об управлении ключами криптосистем.

Литература к разделу: 19(268-271), 39(93-116), 42(79-91)

Теория сложности вычислений и криптография.

Краткое введение в теорию сложности. Вычислительные машины, задачи, алгоритмы и сложность. Временная, емкостная, асимптотическая сложность. Сложность в худшем случае, средняя сложность. Модели вычислений. Решаемые, трудные и алгоритмически неразрешимые задачи. Классификация задач по сложности. Классы P, NP, NP-полные, … . Полиномиальная сводимость. Теорема Кука.

Используемые в криптографии задачи теории сложности: задача о коммивояжере, задача о рюкзаке, задача о выполнимости, задача о факторизации больших целых чисел, задача о дискретном логарифмировании, и др. Современные оценки сложности решения этих задач.

Литература к разделу: 19(272-276), 39(118-174)

Основные понятия криптографии с открытым ключом.

Предпосылки появления криптографии с открытым ключом. Сравнение криптосистем с открытым и секретным ключом.

Однонаправленные (односторонние) функции по Нидхэму. Связь с NP-полными задачами. Примеры однонаправленных функций на основе блочных шифров. Применение в протоколах аутентификации. Программный продукт S/KEY фирмы Bellcore. Однонаправленные функции, основанные на сложности задачи дискретного логарифмирования. Схема открытого распределения ключей Диффи и Хеллмана. Шарады Меркля.

Однонаправленные (односторонние) функции с секретом и их применение для цели шифрования информации. Понятия о цифровой подписи на основе однонаправленной функции с секретом. Классификация атак на схемы цифровой подписи.

Схемы шифрования с открытым ключом. Основные принципы. Схемы RSA и Рабина и их применение. Схемы открытого шифрования Эль Гамаля, МакЭлайса, Меркля – Хеллмана и др. Атаки, выбор безопасных параметров.

Некоторые методы быстрой модульной арифметики и их применение для ускорения криптографических алгоритмов. Алгоритм Монтгомери и его модификации. Подходы к конструированию криптосистем: теоретико-информационный, теоретико - сложностной и теоретико-системный.

Литература к разделу: 19(46-52, 515-540), 38(247-267), 39(290-345, 355-378, 554-598)

Цифровая подпись.

Основные понятия. Типы атак на схемы подписи. Схема Лампорта одноразовой подписи. Схемы цифровой подписи RSA и Рабина. Схема цифровой подписи Эль Гамаля и ее модификации. Способы ускорения процедур подписи и проверки. Сравнение стандартов цифровой подписи США (FIPS PUB 186) и России (ГОСТ Р 34.10-94). Новый стандарт цифровой подписи ГОСТ Р 34.10-2001 на основе эллиптических кривых. Методы генерации секретных параметров для стандартов цифровой подписи. Схемы подписи Фиата-Шамира, Файге-Фиата-Шамира и др. Схема Шнорра.

Литература к разделу: 19(52-62, 541-562), 39(600-631)

Разновидности протоколов электронной цифровой подписи.

Подпись вслепую (blind signature) и ее применения. Схемы конфиденциальной подписи (undeniable signature) и их применение. Протоколы проверки и отвержения как примеры протоколов доказательств с нулевым разглашением. Схемы Шаума. Схемы подписи, в которых подделка подписи может быть доказана. Схемы мультиподписи (multisignature scheme). Групповая подпись (group signature scheme). Схемы подписи с восстановлением сообщения (message recovery). Подпись по доверенности (proxy signature). Подписи с обнаружением подделки (fail-stop digital signature). Подписи, подтверждаемые доверенным лицом (designated confirmer signature). Ring signature.

Литература к разделу: 19(103-107, 136-139)

Функции хэширования.

Классификация. Слабые и сильные функции хэширования. Функции хэширования без ключа (MDC) и с ключом (MAC). Атаки на функции хэширования. Принципы построения. Слабости функций хэширования Ривеста: MD2, MD4, MD5. Американский стандарт функции хэширования (SHS) и его изменения. Российский стандарт функции хэширования (ГОСТ Р 34.11-94). Применение функции хэширования в схемах цифровой подписи и при построении криптосистем.

Литература к разделу: 19(481-514), 38(104-117), 39(346-353), 42(278-306)

Управление ключами.

Классификация ключей по типу алгоритма и использованию. Генерация и хранение ключей. Требования к генераторам псевдослучайных последовательностей. Тестирование генераторов. Доказуемо безопасные генераторы ключей. Некоторые способы сокращения объемов хранимых ключей.

Протоколы распределения криптографических ключей. Виды протоколов распределения. Протоколы типа Диффи-Хеллмана. Протоколы выработки и распределения сеансовых ключей. Kerberos. Особенности реализации в ОС Windows 2000.

Криптографическая инфраструктура на основе механизма открытых ключей(PKI). Модели криптографической инфраструктуры. Стандарт X.509, SPKI – Simple Public Key Infrastructure, PGP – Pretty Good Privacy.

Протоколы, основанные на идентификационной информации (ID-based cryptosystems). Протоколы для конференц- связи. Протокол Ингемарссона-Танга-Вонга. Иерархические схемы распределения ключей. Протоколы с разделения секрета. Пороговые схемы.

Депонирование ключей (Key Escrow).

Литература к разделу: 19(63-66, 199-220, 415-426, 573-586), 38(177-208, 321-388)

Криптосистемы и протоколы на эллиптических кривых.

Представление открытого текста. Аналог схемы открытого распределения ключей Диффи- Хеллмана, аналоги схем шифрования с открытым ключом Месси-Омуры, Эль-Гамаля и др. Примеры реализации.

Литература к разделу: 39(206-214), 42(208-229)

Протоколы идентификации и аутентификации.

Слабая и сильная аутентификация. Методы аутентификации на основе криптосистем с секретным или открытым ключом. Протоколы Файге-Фиата-Шамира, GQ протокол идентификации (Guillou-Quisquater). Протокол Шнорра. Протоколы, основанные на идентификационной информации (identity-based). Атаки на протоколы идентификации.

Литература к разделу: 19(72-86, 563-572), 38(118-129), 39(379-477, 633-672)

Протоколы честного обмена секретами

Честный обмен секретами. Двусторонние и многосторонние протоколы. Асинхронные протоколы честного обмена. Честный обмен с доверенной или с почти доверенной стороной. Протоколы без доверенной стороны. Одновременное подписание контракта. Применение протоколов честного обмена в платежных системах. Заказная электронная почта.

Литература к разделу: 19(143-150)

Интерактивные схемы доказательств

Интерактивные схемы доказательств с нулевым разглашением. Доказательство знания. Доказательство идентичности. Практические применения теории доказательств с нулевым разглашением.

Литература к разделу: 19(133-136), 39(675-723), 42(307-313)

Протоколы электронного тайного голосования .

Требования к идеальному протоколу. Использование схем подписи вслепую. Голосование с одной и двумя Центральными Избирательными Комиссиями (ЦИК). Использование Центрального Управления Регистрации (ЦУР). Голосование без ЦИК, схема Меррита (Merritt M.). Классификация протоколов голосования. Протоколы с перемешиванием и протоколы с разделением. Протоколы секретного многостороннего вычисления.

Литература к разделу: 18(151-160)

Понятие о протоколах электронных платежей

Общие требования к платежным системам. Неотслеживаемость. Анонимность. Централизованные и автономные системы. Схемы Шаума, Якоби, Брандса, Шнорра. Идентификация повторной траты “электронных денег”. Переводимые монеты. Примеры. Платежи в Интернет. Протоколы SSL, SET,3D Secure, SEPP, STT. Микроплатежи. Протокол iKP, DigiCash, PayCash и др.

Платежные системы в мобильной коммерции.PayBox, GiSMo, и др.

Классификация, характеристика и примеры протоколов электронной коммерции. Методы обеспечения честности и неотказуемости участников криптографического протокола. Методы конструирования и анализа робастных протоколов. Доказательность действий участников протокола.

Безопасность протоколов электронных игр, лотерей, аукционов.

Литература к разделу: 19()166-176

Вопросы стандартизации и патентования

Стандарты Интернет и RFCs. Политика различных организаций в области защиты информации.

Литература к разделу: 19(366-367, 633, 651, 664, 675), 38(389-403)

Необходимые сведения из алгебры и теории чисел

Алгебраические структуры с одной и двумя бинарными операциями. Теория делимости в кольце целых чисел и многочленов. Наибольший общий делитель и наименьшее общее кратное. Алгоритм Евклида. Расширенный алгоритм Евклида. Теорема Чезаро. Простые числа. Попарно взаимно простые числа. Китайская теорема об остатках. Функция Эйлера и ее свойства. Теорема Эйлера-Ферма. Теорема Кармайкла. Псевдопростые числа. Вероятностные тесты на простоту целых чисел.

Сведение сравнений n-ой степени по произвольному модулю к системе сравнений по попарно взаимно простым модулям, к сравнениям по примарному и простому модулю. Сравнения первой и второй степени. Символы Лежандра и Якоби. Критерий Эйлера. Метод Берлекемпа решения сравнений второй степени по простому модулю. Теорема эквивалентности Рабина.

Основные факты об эллиптических кривых над полями. Эллиптические кривые и факторизация больших целых чисел. Дискретный логарифм на эллиптической кривой над полем.

Литература к разделу: 19(277-302), 38(210-230, 390-403), 39(175-205, 215-244)