В.А. Мухачев, В.А. Хорошко

МЕТОДЫ ПРАКТИЧЕСКОЙ КРИПТОГРАФИИ

ООО «ПолиграфКонсалтинг»

Киев 2005

ББК 32.973.2-018.2
М92

УДК 519.254

Мухачев В.А., Хорошко В.А.

Методы практической криптографии. – К.: ООО «Полиграф-Консалтинг», 2005. – 215 с.

В представленном читателям издании рассматривается круг вопросов, связанных с надежностью действующих систем криптографической защиты информации.

Обосновываются требования к параметрам ряда распространенных криптоалгоритмов и криптографическим свойствам некоторых преобразований. Приводятся методы их генерации и тестирования.

Издание предназначено для специалистов, занимающихся внедрением и эксплуатацией криптографических систем защиты информации, студентов высших учебных заведений и аспирантов.

Рецензенты: *докт. техн. наук, профессор Кузнецов Г.В.*

докт. техн. наук, профессор Скрипник Л.В.

© В.А. Мухачев

ISBN 966-8440-48-X

© В.А. Хорошко

СОДЕРЖАНИЕ

ВВЕДЕНИЕ **7**

ГЛАВА 1. ОБЩИЕ ПОЛОЖЕНИЯ И ОСНОВНЫЕ ПРИНЦИПЫ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ **11**

1.1. Общая система секретной связи (по К. Шеннону) 13

1.1.1. Основные криптографические термины *13*

1.1.2. Модель системы секретной связи К.Шеннона *14*

1.1.3. Проблема распределения ключей. Асимметричные криптосистемы *16*

1.2. Подходы к оценке надежности реальных криптосистем 18

1.2.1. Метод экспертных оценок *19*

1.2.2. Метод сведения к общей алгоритмической проблеме *21*

ГЛАВА 2. ОБЩИЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ АНАЛИЗА ОСНОВНЫХ ТИПОВ ШИФРОВ **23**

2.1. Элементарные шифры 24

2.2. Основные типы шифров 26

2.2.1. Потоковые шифры. Последовательность выбора шифрпреобразований *27*

2.2.2. Качество гаммы *28*

2.2.3. Периодичность гаммы *29*

2.2.4. Блочные шифры *31*

2.2.5. Алгоритмические проблемы, связанные со стойкостью основных типов шифров *32*

ГЛАВА 3. ТЕСТИРОВАНИЕ УЗЛОВ КРИПТОСХЕМ КАК МЕТОД КОМПРОМЕТАЦИИ ШИФРОВ **35**

3.1. Компрометация шифров 36

3.2. Задача тестирования линейной рекуррентной составляющей криптоузла 37

3.3. Задача восстановления параметров искаженной линейной рекурренты 40

3.3.1. Представление элементов рекурренты через элементы начального заполнения *40*

3.3.2. Производные соотношения *42*

3.3.3. Некоторые сведения о подходах к восстановлению параметров искаженной линейной рекурренты *43*

3.3.4. Качественная характеристика задачи восстановления параметров линейной искаженной рекурренты *45*

ГЛАВА 4. КРИПТОГРАФИЧЕСКИЕ СВОЙСТВА БУЛЕВЫХ ФУНКЦИЙ **49**

4.1. Нелинейность булевой функции 50

4.2. Критерии распространения и корреляционная иммунность 58
4.3. Устойчивые булевы отображения 62

ГЛАВА 5. ОСОБЕННОСТИ ПРИМЕНЕНИЯ АЛГОРИТМА ГОСТ 28147-89 67
5.1. Криптоэквивалентная схема алгоритма ГОСТ 28147-89 69
5.2. Влияние блока подстановки на последовательности выходов итераций 70
5.2.1. Расшифрование в режиме простой замены 72
5.2.2. Возможность ослабления шифра за счет структуры сеансового ключа 73
5.3. Замечания о режимах шифрования и имитовставки 73

ГЛАВА 6. ВЫБОР ДОЛГОВРЕМЕННОГО КЛЮЧА АЛГОРИТМА ГОСТ 28147-89 ... 77
6.1. Область сильных ключей 78
6.1.1. Достаточность условия равновероятности псевдогаммы для выбора сильного блока подстановки 79
6.1.2. Сведения о требованиях к выбору блока подстановки 81
6.2. Контроль долговременного ключа алгоритма ГОСТ 28147-89 84
6.2.1. Угроза внедрения слабых параметров 84
6.2.2. Подход к выявлению слабых долговременных ключей 86
6.2.3. Свойства теста 88
6.2.4. Тестирование долговременного ключа 90

ГЛАВА 7. ЭЛЕМЕНТЫ ТЕОРИИ СРАВНЕНИЙ 93
7.1. Некоторые сведения из арифметики 94
7.1.1. Расширенный алгоритм Эвклида 95
7.2. Модульная арифметика 96
7.2.1. Функция Эйлера и малая теорема Ферма 97
7.3. Сравнения первой степени от одного неизвестного 98
7.3.1. Китайская теорема об остатках 99
7.3.2. Степенные сравнения по простому модулю 100

ГЛАВА 8. ВЫЧИСЛЕНИЕ КВАДРАТНОГО КОРНЯ В ПРОСТОМ ПОЛЕ 103
8.1. Квадратичные вычеты по простому модулю 104
8.1.1. Символ Лежандра 105
8.1.2. Символ Якоби 107
8.2. Алгоритм нахождения квадратного корня в простом поле 109

ГЛАВА 9. КРИПТОСИСТЕМА RSA, ПРОТОКОЛ ДИФФИ-ХЭЛЛМАНА И ЦИФРОВАЯ ПОДПИСЬ ЭЛЬ-ГАМАЛЯ 115
9.1. Построение криптосистемы RSA. Идея цифровой подписи 116
9.2. Смешанные криптосистемы. Протокол Диффи-Хэллмана ключевого обмена 119
9.3. Цифровая подпись Эль-Гамаля 121
9.3.1. Криптосистема Эль-Гамаля 121
9.3.2. Механизм цифровой подписи Эль-Гамаля 122

9.3.3. Ослабление подписи Эль-Гамаля вследствие некорректной реализации схемы *123*
9.3.4. Варианты цифровой подписи типа Эль-Гамаля *125*
ГЛАВА 10. АЛГОРИТМ СИЛЬВЕРА-ПОЛЛИГА-ХЭЛЛМАНА **127**
10.1. Обозначения и постановка задачи 128
10.2. Построение корней из единицы в поле 129
10.3. Алгоритм дискретного логарифмирования 130
10.3.1. Пример вычисления дискретного логарифма *132*
10.3.2. Логарифмирование в группе единиц кольца вычетов по модулю p^r. *134*
10.4. Фальсификация подписи Эль-Гамаля в специальном случае выбора первообразного элемента и характеристики поля 137
10.4.1. Слабые параметры в подписи Эль-Гамаля *139*
ГЛАВА 11. МЕТОДЫ ФАКТОРИЗАЦИИ ПОЛЛАРДА **141**
11.1. (P-1) - метод факторизации Полларда 142
11.2. Ро - метод факторизации Полларда 144
11.2.1. Оценка вероятности выбора критической пары *146*
11.2.2. Оптимизация выбора критической пары *148*
ГЛАВА 12. НЕКОТОРЫЕ СЛУЧАИ ОСЛАБЛЕНИЯ КРИПТОСИСТЕМЫ RSA **151**
12.1. Атаки на RSA, не использующие факторизацию модуля 152
12.2. Атаки на RSA, использующие факторизацию модуля 156
12.2.1. Алгоритм факторизации Диксона *157*
ГЛАВА 13. ТЕСТ ФЕРМА ПРОВЕРКИ ЧИСЕЛ НА ПРОСТОТУ **163**
13.1. Решето Эратосфена и критерий Вильсона 164
13.2. Тест на основе малой теоремы Ферма 166
13.2.1. Основные свойства псевдопростых чисел *166*
13.2.2. Свойства чисел Кармайкла *168*
13.2.3. (n-1) - критерий Люка *169*
13.2.4. Понятие о последовательностях Люка. (n+1) - критерий Люка *171*
13.3. (P+1) – метод факторизации Вильямса 172
ГЛАВА 14. ТЕСТЫ СОЛОВЕЯ-ШТРАССЕНА И РАБИНА-МИЛЛЕРА НА ПРОСТОТУ **175**
14.1. Тест Соловея-Штрассена 176
14.1.1. Эйлеровы псевдопростые числа *177*
14.2. Тест Рабина-Миллера 178
14.2.1. Сильно псевдопростые числа *180*
ГЛАВА 15. ПОСТРОЕНИЕ БОЛЬШИХ ПРОСТЫХ ЧИСЕЛ **181**
15.1. Детерминированный тест, основанный на обобщенном критерии Люка 182
15.1.1. Теорема Поклингтона *183*

15.1.2. Обобщение критерия Люка *184*
15.2. Детерминированный тест, основанный на теореме Димитко 185
ГЛАВА 16. ВЫБОР ПАРАМЕТРОВ КРИПТОСИСТЕМЫ RSA 189
16.1. Общие требования к выбору параметров 190
16.2. Метод Гордона построения сильно простых чисел 192
16.3. Пример построения сильно простого числа 194
ГЛАВА 17. ОБЩИЕ СВЕДЕНИЯ ОБ ИНОСТРАННЫХ КРИПТОСРЕДСТВАХ 197
17.1. Аппаратные криптосредства 198
17.2. Основные принципы построения систем управления ключами 201
17.2.1. Ключевые системы потоковых шифров *202*
17.3. Блочные шифры в смешанных криптосистемах 203
17.3.1. Алгоритм RC5 ..... *203*
17.3.2. Смешанная криптосистема на основе алгоритмов RSA и IDEA *204*
ЗАКЛЮЧЕНИЕ 207
ЛИТЕРАТУРА 209

ВВЕДЕНИЕ

В современных условиях защита информации становится все более актуальной и одновременно все более сложной проблемой. Это обусловлено как массовым применением методов автоматизированной обработки данных, так и широким распространением методов и средств несанкционированного доступа к информации. Поэтому особую роль в организации противодействия потенциальным угрозам занимает подход, при котором средства защиты информации используются комплексно, каждое в соответствии со своим предназначением.

Различные методы и средства защиты информации от несанкционированного доступа, а также достаточно полное описание возможных каналов утечки, представлены в книге В.А. Хорошко и А.А. Чекатова «Методы и средства защиты информации» [25]

С принятием в Украине законов «Об электронной цифровой подписи», «Об электронных документах и электронном документообороте» перед пользователями возникла необходимость организации и внедрения систем конфиденциальной связи, на основе принятия самостоятельных решений, требующих определенного уровня подготовки в области криптографической защиты информации.

Как правило, пользователи приобретают и используют различные коммерческие криптосредства, которые разрабатываются с использованием типичных криптоузлов и методов, исходя из рекомендаций соответствующих международных стандартов.

Вследствие этого коммерческие криптосредства (при соответствующей классификации) имеют аналогичные свойства и особенности, во многих отношениях определяемые лишь спецификой выбора параметров криптоалгоритмов (в том числе, ключей). Поэтому качественные

стандартизованные криптоалгоритмы могут обладать (и действительно обладают) слабостями, зависящими от конкретных значений их параметров.

В предлагаемой книге авторы стремились ввести читателя в круг вопросов, связанных с мотивировкой выбора параметров распространенных криптоалгоритмов и процедур их генерации. Целью авторов являлось доступное и компактное изложение основных понятий и подходов, используемых в практической криптографии. В этом контексте нашу книгу можно рассматривать как раздел книги [25], являющийся развитием гл.18.

Изложение материала в книге ведется с точки зрения «дружественного криптоаналитика» и построено на описании требований к параметрам криптоалгоритмов как условий, необходимых для блокировки той или иной угрозы несанкционированного доступа к информации.

Сами условия соответствуют ситуациям, в которых существуют частные решения для алгоритмических проблем, лежащих в основе соответствующих типов криптографических преобразований, либо известен подход, ослабляющий стойкость криптоузла определенного типа.

С этой точки зрения в книге рассматриваются случаи ослабления криптосистемы RSA, возможность фальсификации цифровой подписи типа Эль-Гамаля, некоторые особенности потоковых шифров, общие требования к криптографическим свойствам булевых функций.

Особенности выбора параметров блочных шифров проиллюстрированы на примере режима простой замены ГОСТ 28147-89. Для этого шифра рассмотрены элементарные примеры слабых ключей, приведена общая методика построения подобласти стойких долговременных ключей и их тестирования.

Описанные в книге некоторые классические алгоритмы факторизации и дискретного логарифмирования также следует рассматривать как криптоаналитические атаки на соответствующие асимметричные криптосистемы. Соответственно, методы тестирования чисел на простоту,

построения случайных простых и т.н. случайных сильно простых чисел обеспечивают, в совокупности, противодействие этим атакам путем построения безопасных параметров.

Подобный подход для изложения материала, по мнению авторов, позволяет собрать воедино и рассматривать факты и методы, относящиеся к криптоалгоритмам и криптосистемам различного типа и не связанные какой-либо общей стратегией поиска слабостей.

Авторы старались сделать изложение максимально доходчивым. В книге кратко изложены основные понятия криптографии, приводятся примеры, иллюстрирующие теоретико-числовые алгоритмы и методы. По ходу изложения вводятся необходимые сведения из элементарной теории чисел.

При необходимости основные математические понятия, а также свойства объектов, нужные для понимания терминологии, используемой в криптографии, можно изучить, познакомившись с первыми главами книги [23]. Терминологию, а также краткие сведения по большинству вопросов в области защиты информации можно найти в издании [18].

Строгое и более полное изложение математических методов, лежащих в основе оценки стойкости и тестирования параметров асимметричных криптоалгоритмов, дано в [21].

Авторы надеются, что изложенный в книге материал позволит пользователям самостоятельно разбираться и сознательно использовать рекомендации стандартов в области криптографической защиты информации.

Это касается, в том числе, стандартов, обеспечивающих безопасность криптопротоколов и криптосистем на эллиптических кривых [17], несмотря на то, что указанные средства в данной книге не рассматриваются.

Авторы выражают глубокую благодарность доктору технических наук, профессору Кузнецову Георгию Витальевичу (Национальный горный университет, г.Днепропетровск), доктору технических наук, профессору Скрипнику Леониду Васильевичу (Военный институт телекоммуникаций и

информатизации НТУ Украины «КПИ», г.Киев) за рецензирование книги, которое способствовало ее значительному улучшению.

Кроме того, мы благодарны сотрудникам Службы безопасности Украины за внесенные замечания и советы, особенно Гулаку Г.Н., Кучинскому В.В. и Степанову В.Д.

Мы выражаем также особую признательность всем авторам, чьи работы помогли в расширении и углублении затронутой в нашей книге тематики.

Связаться с авторами предлагаемой читателю книги можно по адресу издательства.

Глава 1.
ОБЩИЕ ПОЛОЖЕНИЯ И ОСНОВНЫЕ ПРИНЦИПЫ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

Основное в этой главе…

Общая система секретной связи (по К. Шеннону)……...……...13

Проблема распределения ключей. Асимметричные криптосистемы……16

Подходы к оценке надежности реальных криптосистем……...……18

Внедрение и активное использование современных информационных технологий существенно повысили уязвимость информации, циркулирующей в современных информационно-телекоммуникационных системах.

Несанкционированное искажение, копирование, уничтожение информации в настоящее время затрагивает не только процессы, относящиеся к сфере государственного управления, но и интересы физических лиц.

Как следствие, возрастает ответственность за принятие точных и ответственных решений в ситуации, когда даже отдельные ошибки способны привести к тяжелым последствиям в сфере экономики, финансов, экологии.

С движением общества по пути технического прогресса не только неуклонно возрастают объемы обрабатываемой информации, но и все более проявляется фактор необходимости своевременной ее обработки.

В сложившихся условиях, очевидно, перспективным является внедрение технологий, основанных на электронном документообороте.

Использование подобных технологий приводит к необходимости использования ЭВМ для решения специфических задач, таких как обеспечение уверенности в гарантиях ответственности лиц, отдающих распоряжения, ответственности исполнителей за своевременное выполнение распоряжений и т.д. Для решения указанных задач необходимо наличие специфических алгоритмов.

Оказалось, что принципы построения соответствующих алгоритмов разработаны в криптологии – науке, изучающей методы построения и анализа систем защиты информации, основанных на математических преобразованиях данных с использованием секретных параметров. Такие системы называются криптографическими. Применение криптографических систем защиты информации позволяет обеспечить надежность электронного документооборота.

Криптология традиционно разделяется на криптографию и криптоанализ. Методы криптографии ориентированы на создание систем защиты информации.

К области криптоанализа относятся подходы к осуществлению несанкционированного доступа аналитическим путем к информации, защищенной криптографическими методами.

Криптоаналитики исследует вопросы стойкости абстрактных и реальных криптосистем. Оба направления криптологии взаимодействуют, дополняя друг друга, что позволило решить ряд указанных актуальных задач и внедрить в практику достаточно эффективные системы криптографической защиты информации.

1.1. Общая система секретной связи (по К. Шеннону)

Криптография - дисциплина, включающая принципы, средства и математические методы преобразования информации, с целью сокрытия смысла или структуры данных, а также для защиты от их несанкционированного использования или подделки. Основным методом криптографии является шифрование.

1.1.1. Основные криптографические термины

Шифрованием называется взаимно-однозначное преобразование сообщения, с целью скрытия его смысла от посторонних лиц. Исходный текст сообщения – открытый текст. Результат шифрования – шифрованный текст (шифртекст). При шифровании могут использоваться секретные параметры – ключи.

Открытый текст состоит из элементов, которые определяются шифрпреобразованием. Элемент - это наименьшая часть данных (набор битов), которая может быть зашифрована. Элементам открытого текста соответствуют элементы шифртекста.

Для передачи по каналам связи шифртекст объединяется вместе с дополнительными данными (например, адресами корреспондентов) и компонуется в последовательность, называемую криптограммой.

Лицо, в адрес которого была направлена криптограмма, производит над ней ряд определенных операций, что позволяет ему восстановить открытый текст сообщения. Эта процедура называется расшифрованием криптограммы.

Создание системы секретной связи включает в себя, кроме прочего, разработку собственно криптосистемы, под которой, в широком смысле, понимаются документы, устройства, оборудование и соответствующие методы, использование которых (в совокупности) обеспечивает средства зашифрования и расшифрования.

В узком смысле под криптосистемой понимают так называемую систему шифрования, в рамках которой рассматривают лишь методы, обеспечивающие средства зашифрования и расшифрования при обмене информацией.

Если кому-либо из посторонних удается путем анализа криптограммы получить открытый текст, то говорят, что криптограмма была дешифрована (раскрыта, решена). Заметим, что дешифрование криптограммы, вообще говоря, не обязательно требует восстановления исходного ключа.

Существуют системы, которые теоретически нельзя дешифровать. Все прочие системы, следовательно, заведомо обладают слабостями, поэтому возможность дешифрования или значительного ослабления шифра, вообще говоря, никогда нельзя исключать полностью.

Дело заключается в том, насколько всесторонне и тщательно учтены все особенности работы криптосистемы в зависимости от условий, в которых она применяется.

1.1.2. Модель системы секретной связи К.Шеннона

Широко известная модель системы секретной связи К.Шеннона была предложена в его работе «Теория связи в секретных системах», опубликованной в 1949 году [1,2].

Принято считать, что эта работа возвестила начало эры научной криптографии. В этой работе К.Шеннон, в частности, рассматривает понятие стойкости шифра в рамках соответствующей модели и вводит общее понятие практической стойкости шифров.

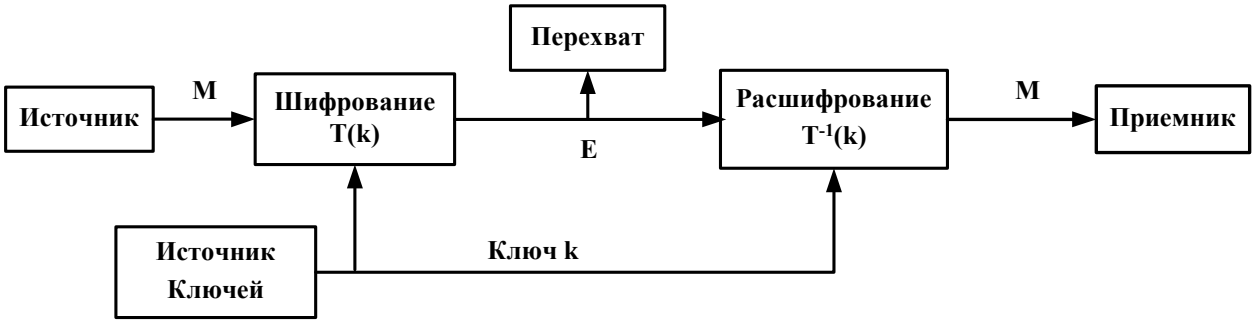

Рис.1. Модель Шеннона системы секретной связи

По Шеннону, криптографическая система есть семейство обратимых отображений множества возможных сообщений во множество криптограмм. Каждое отображение *T(k)* есть шифрование. Выбор отображения производится с помощью случайного секретного параметра *k*, называемого ключом.

Ключ также позволяет выбрать обратное преобразование (расшифрование). Реально он используется как параметр в вычислениях при шифровании и расшифровании. Ключ обязан быть доступен отправителю и получателю в необходимый момент.

Такая система секретной связи носит название симметричной криптосистемы (симметрия пользователей относительно знания секрета).

В данной системе вопрос о распределении ключей фактически выносится Шенноном за рамки модели, т.к. постулируется наличие безопасного канала распространения ключей.

Важным моментом является модель противника, которому доступен для перехвата шифртекст и которому известен алгоритм шифрования и его параметры, за исключением ключа (принцип открытости, общедоступности системы).

Практическая стойкость шифрпреобразования определяется трудоемкостью получения открытого текста аналитически, без знания ключа, с помощью наилучших существующих на сегодняшний день алгоритмов.

Таким образом, стойкость криптосистемы не выше сложности задачи определения ключа.

Заметим, что с помощью шифрования решается задача обеспечения конфиденциальности данных.

1.1.3. Проблема распределения ключей. Асимметричные криптосистемы

При практическом использовании модели Шеннона необходимость реализации защищенного канала для ключевого обмена порождает так называемую проблему безопасного распространения ключей.

Кроме того, при использовании средств шифрования в автоматизированных системах, возникает проблема подтверждения истинности той или иной информации, так называемой подписи электронных сообщений. Например, это может быть необходимо при проверке того, что ключ действительно принадлежит лицу, от имени которого он поступает в систему, поскольку существует возможность т.н. навязывания ключа.

Обе эти задачи, без использования защищенного канала связи, удалось решить в рамках модели криптосистемы с «открытым» ключом, предложенной В.Диффи и М.Хеллманом в 1976 году [3,29].

Отличие модели системы секретной связи В.Диффи и М.Хеллмана от модели К.Шеннона в том, что она является асимметричной в том смысле, что пользователи по отношению к секретному параметру неравноправны. Ключ известен полностью только получателю сообщения и представляет собой пару (*e, d*), где подключ *e* (т.н. открытый ключ) служит ключом зашифрования, а подключ *d* служит для расшифрования, при этом только *d* является секретным параметром (т.н. секретный, личный, ключ).

Ключ d применяется получателем сообщений, которые отправители должны шифровать, используя ключ e. Стойкость системы обеспечивается за счет особых свойств шифрпреобразования, которое представляет собой так называемую одностороннюю функцию с «лазейкой». Вычисление значения такой функции (от открытого текста и параметра e) должно быть несложным, в то же время ее обращение должно быть вычислительно нереализуемым без знания секретной информации, «лазейки», связанной с секретным ключом d.

Такие криптосистемы называются асимметричными или системами с открытыми ключами.

Строго говоря, не доказано, что односторонние функции существуют. Однако признано, что некоторые преобразования обладают свойствами, близкими к свойствам односторонних функций. Они широко используются в действующих системах криптографической защиты информации (еще один аргумент в пользу актуальности проблемы надежности криптосистем).

Каким образом проблема безопасного распространения ключей решается с помощью асимметричных криптосистем?

Для этого каждый желающий передать ключ для симметричной криптосистемы своему абоненту перешифровывает его ключом e этого абонента (полагается, что асимметричная система создана заранее и открытый ключ опубликован). Результат шифрования передается по открытому каналу.

Односторонняя функция гарантирует безопасность, т.к. расшифровать сообщение можно, только зная ключ d, а его знает лишь нужный абонент.

Общеизвестно, что данный механизм не является безопасным. Дело усложняется настолько, что на практике оказалось необходимым вводить в глобальном масштабе систему так называемых центров сертификации открытых ключей [4].

При дальнейшем развитии идея использования односторонних функций в криптографии позволила решить ряд проблем, связанных с защитой

информации. Так, например, большое число важных задач решается с помощью интерактивных процедур, которые называются криптографическими протоколами. Наиболее общими являются следующие задачи.

Обеспечение целостности данных. Целостность данных - свойство данных, позволяющее после передачи получить их в исходном виде, несмотря на изменения, предусмотренные протоколом.

Аутентификация абонента: проверка того, что абонент действительно является тем лицом, за которое себя выдает.

Аутентификация сообщения: проверка того, что сообщение передано без изменений от заявленного отправителя к соответствующему корреспонденту.

Реальные криптопротоколы отличаются большим разнообразием, причем в них используются как асимметричные, так и симметричные криптоалгоритмы [4,5].

Для обоснования надежности протоколов недостаточно гарантий стойкости применяемых криптопреобразований, поскольку обмен информацией может осуществляться в условиях взаимного недоверия пользователей [2].

1.2. Подходы к оценке надежности реальных криптосистем

Развитие математических методов и повышение производительности вычислительной техники может со временем привести к ослаблению используемой криптосистемы. Таким образом, возникает необходимость криптологического сопровождения криптосистем в течение всего срока их действия.

Сопровождение криптосистемы заключается в контроле за соблюдением порядка и условий ее эксплуатации, а также в принятии своевременного решения на изменение ее параметров.

Основных причин, вследствие которых следует принять тезис о потенциальной ненадежности действующих криптосистем две: отсутствие полных формальных критериев качества криптосистем, а также невозможность на практике выполнения в полном объеме требований, исходя из которых вывод о достаточном уровне защиты информации обоснован теоретически.

К тому же, поскольку в настоящее время имеет место массовое применение криптосистем, начинают проявляться маловероятные ситуации, неучтенные разработчиком. Соответственно, возникают предпосылки к нарушению защиты информации.

Еще одним источником потенциальной ненадежности криптосистем является возможность создавать криптосистемы с сознательно внесенными слабостями (лазейками).

Отсутствие формального критерия стойкости криптосистемы вынуждает использовать для определения уровня ее надежности оценку практической стойкости, т.е., согласно К.Шеннону, оценивать стойкость, исходя из параметров наилучшего известного метода дешифрования.

1.2.1. Метод экспертных оценок

Суть метода экспертных оценок заключается в использовании всего криптологического потенциала, имеющегося в распоряжении предприятия, фирмы, государства, для получения максимально объективных выводов относительно качества разрабатываемой или внедряемой криптосистемы.

Как правило, применяется многоэтапное обсуждение проектов.

На каждом этапе обсуждаются заключения экспертов и принимается решение об устранении обнаруженных недостатков, дальнейшем обсуждении, либо о завершении работы экспертов.

Окончательное решение принимается соответствующей комиссией на основе экспертных выводов. Решение комиссии утверждается руководителем уполномоченного органа.

Надежность криптосистемы зависит не только от математических свойств криптоалгоритмов. Она зависит и от системы генерации ключей, и от психологии персонала, и даже от состояния системы электроснабжения (если система неработоспособна, то она бесполезна). Таким образом, к оценке надежности криптосистемы необходимо подходить комплексно.

Комплексная оценка качества средств криптографической защиты информации предусматривает рассмотрение вопросов стойкости алгоритмов, построения ключевой системы, тестирование системы связи и т.п. Особую сложность представляют собой проблемы технической защиты информации.

Большое значение для эффективности комплексной оценки имеют полнота и корректность требований, выдвигаемых к системе пользователями.

Выбор средств криптографической защиты информации и исходные требования к их стойкости должны быть адекватными той конкретной задаче в области информационной безопасности, в которой эти средства планируется использовать.

Поэтому неправильно было бы приобретать произвольную криптосистему, а затем исследовать ее свойства. На самом деле необходимо, исходя из потребностей, проработать и выдвинуть требования относительно желаемых свойств шифра и лишь за тем оценивать качество системы или ее проекта. Фактически, оптимальная оценка потребностей в защите информации есть главная задача при принятии решения на использование криптосредств.

Особо следует подчеркнуть, что после введения криптосистемы в эксплуатацию, как правило, она становится практически недоступной для наблюдения и дальнейшего анализа, хотя именно в процессе эксплуатации недостатки выявляются наиболее полно и качественно.

1.2.2. Метод сведения к общей алгоритмической проблеме

Не существует универсальных средств защиты информации и отдельных алгоритмов, которые подходили бы для решения криптографических задач без предварительных исследований. Для каждой конкретной задачи необходима адаптация соответствующих алгоритмов.

Даже применение стандартизированных методов защиты информации требует квалифицированной проработки многих дополнительных вопросов, прежде всего, выбора конкретной конфигурации криптоалгоритма, что часто приводит к необходимости формального обоснования стойкости его параметров.

Одним из подходов к оценке стойкости и выявлению наличия известных типов слабостей для конкретной криптосистемы, является метод сведения оценки стойкости к оценке сложности некоторой общей алгоритмической проблемы.

Типичным случаем являются, например, криптосистемы с открытыми ключами, поскольку они создавались таким образом, чтобы задача их дешифрования сводилась к той или иной вычислительно нереализуемой процедуре.

Основная трудность заключается в том, что для реальных систем упомянутые алгоритмические проблемы изучены недостаточно и имеют большое количество специальных значений параметров, при которых системы становятся слабыми.

Полностью избежать подобных случаев крайне сложно, поскольку, как уже отмечалось, массовое использование криптосистем обеспечивает возможность реального возникновения этих редких ситуаций.

Разработкой криптосистем и анализом их стойкости занимаются многие специалисты в различных странах мира.

Методы, используемые при анализе криптоалгоритмов, во многом аналогичны тем, что применяются при их синтезе. Поэтому большое значение имеет взаимосвязь между подходами к дешифрованию в частных случаях и методами построения стойких параметров криптосистем.

Характерным подходом является стремление выделить некоторые наиболее критичные особенности криптосистемы, выдвинуть ряд требований к криптопреобразованиям, а после построить параметры криптопреобразований, которые наилучшим образом удовлетворяют совокупности выдвинутых условий.

Для осознания необходимых требований также очень важным является поиск частных случаев ослабления стойкости шифрпреобразований или случаев дешифрования отдельных сообщений. Соответствующие криптоаналитические приемы могут быть как очевидными, так и очень сложными.

Можно утверждать, что рекомендации международных стандартов в области криптографии созданы так, чтобы максимально учесть все известные частные случаи ослабления соответствующих криптоалгоритмов.

К недостаткам метода экспертных оценок и метода сведения к общей алгоритмической проблеме следует, соответственно, отнести: во-первых, невозможность предвидения неблагоприятных ситуаций, когда пользователем нарушаются исходные посылки, послужившие базой для оценки качества системы, во-вторых, невозможность учесть все частные случаи, когда решение сложной математической проблемы может быть эффективно реализовано.

В то же время очевидно, что сведение к общей алгоритмической проблеме является естественным подходом к оценке параметров криптоалгоритмов, позволяющим выделить частные задачи и оценивать стойкость криптоалгоритма, исходя из сложности подзадач.

Глава 2.
ОБЩИЕ АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ АНАЛИЗА ОСНОВНЫХ ТИПОВ ШИФРОВ

Основное в этой главе…

- **Элементарные шифры…..……....24**
- **Основные типы шифров……..……..26**
- **Алгоритмические проблемы, связанные со стойкостью основных типов шифров……..…32**

За многие века существования тайнописи было изобретено и практически опробовано огромное количество систем шифрования. Дошедшие до нас основные принципы шифрпреобразований явились результатом длительной эволюции и, по всей видимости, являются теми элементарными составляющими, которые являются основой для построения качественных шифрующих алгоритмов [5,7,8].

2.1. Элементарные шифры

Шифр замены (шифр подстановки) - метод шифрования, при котором каждый элемент исходного текста взаимнооднозначно заменяется одним, либо несколькими знаками некоторого алфавита. Шифр простой замены заменяет каждый знак входного алфавита на некоторый знак из того же алфавита. Результат замены не зависит от расположения знака в открытом тексте. Ключами для шифров замены являются таблицы замены.

Следующий пример представляет собой так называемый шифр двухзначной замены, в котором каждая буква входного алфавита взаимно однозначно заменяется парой цифр.

Таблица 1. Шифр двухзначной замены

	0	1	2	3	4	T O B E O R N O T T O B E 13300224301411301313300224
0	v	p	b	a	c	
1	q	n	z	t	r	
2	d	u	x	l	e	
3	o	j	s	i	f	
4	k	g	v	h	m	
5	w					

Возможно построение шифра, аналогичного шифру замены, когда в такте шифрования могут преобразовываться группы разной значности. Таким свойством обладают шифрсистемы, называемые кодами. Специфической особенностью кодов является то, что они оперируют не с произвольными

комбинациями символов, а со словами, слогами и фразами. Преимуществом кодов является сжатие информации при зашифровании, поскольку кодовые группы, как правило, короче величин, которые они заменяют, а недостатком – то, что словарный состав кода рассчитан на определенный характер переписки, т.е. специализирован.

Шифры замены часто приводятся в специальной и популярной литературе в качестве примеров слабых шифров. Необходимо отметить, что, как и для любого другого шифра, это может быть верным лишь в конкретных случаях. Например, алгоритмы DES**,** ГОСТ 28147-89 базируются на шифре замены, а алгоритм шифрования RSA реализует этот шифр непосредственно.

Шифры перестановки отличаются от шифров замены тем, что при зашифровании буква открытого текста переходит не в фиксированный знак алфавита, а в другую букву того же открытого текста, в результате чего буквы располагаются на новых местах, т.е. переставляются.

Ключи для таких шифров представляются в виде подстановок размерности до длины текста включительно. Шифры перестановки имеют много разновидностей, отличающихся в основном тем, каким способом порождаются ключи. В ручных системах, например, для этой цели используются различные варианты размещения открытых текстов в площади различной конфигурации и выписки его по закону, который содержится в секрете.

Шифры гаммирования. Широко распространенные примеры шифра данного типа основаны на т.н. операции сложения чисел по некоторому модулю.

Символы алфавита открытого текста, предварительно заменяемые на числа, складываются с элементами некоторой числовой последовательности, которая называется гаммой. Процедура зашифрования называется модульным гаммированием, а количество знаков в алфавите - модулем гаммирования.

В общем случае операция гаммирования не обязательно является модульным сложением: часто используются некоторые обратимые табличные функции.

Пример гаммирования по модулю 26.

Расположим латинский алфавит в порядке $z = 0$, $a = 1$, $b = 2,\ldots$, $x = 24$, $y = 25$, где числа означают номера букв в алфавите. Пусть открытый текст есть $d = 4$, $a = 1$, $y = 25$ и гамма равна $a = 1$, $c = 3$, $b = 2$. Сложим соответствующие номера букв. На первых двух тактах шифрования получим $e = 5$, $d = 4$.

Для комбинации $y + b$ значение номера равно 27 и никакой букве алфавита не соответствует. В этом случае запишем, вслед за первым, второй экземпляр алфавита, а нумерацию продолжим: $z = 26$, $a = 27$, …, $y = 51$.

Итак, буква с номером 27 есть буква *a*. Легко заметить, что номера одинаковых букв имеют равные «вычеты», т.е. остатки от деления на 26. Такое свойство называется сравнимостью по модулю 26.

Поскольку можно дописывать любое число алфавитов, то буквы можно складывать, умножать, вычитать. Такая арифметика называется модульной.

В общем случае сравнимость чисел *a* и *b* по модулю *n* записывается в виде $a \equiv b(\mathrm{mod}\, n)$.

Ключом шифрсистемы является гамма, длина которой, вообще говоря, равна длине открытого текста. Поскольку последовательность знаков гаммы может порождаться по некоторому алгоритму, то гамма может быть задана вспомогательным ключом, длина которого существенно меньше длины открытого текста, в частности, гамма может обладать малым периодом, а также другими особенностями.

2.2. Основные типы шифров

Обратим внимание на одно из различий между шифром простой замены и гаммирования: в шифре простой замены один и тоже элемент открытого текста

перейдет в фиксированный знак шифртекста в любом такте шифрования, а в шифре гаммирования это не так. Этот шифр преобразует элемент открытого текста в зависимости от значения гаммы (т.е. ключа) на каждом такте шифрования. Можно сказать, что упомянутый ключ задает последовательность шифрпреобразований, в отличие от шифра простой замены, где все шифрпреобразования одинаковы. Указанное различие приводит к понятиям основных типов шифров: блочных и потоковых шифров соответственно.

2.2.1. Потоковые шифры. Последовательность выбора шифрпреобразований

Рассмотрим пронумерованный список Δ всех различных шифрпреобразований, которые могли бы возникнуть в процессе шифрования сообщений с помощью данной криптосистемы.

Процесс зашифрования можно записать как последовательность номеров шифрпреобразований, выбранных на соответствующих тактах. Обозначим эту последовательность через Γ и назовем ключевым потоком. Последовательность Γ аналогична функции выбора состояний некоторого автомата. Она зависит от ключей и номеров тактов шифрования.

Свойства этой последовательности во многом отражают качество шифра и определяют его классификацию. Например, если список Δ содержит только шифрпреобразования, являющиеся сложением по модулю n, каждое с фиксированным числом c_i, $(i = 0,1, \ldots n - 1)$, то шифр является шифром гаммирования по модулю n.

Потоковым шифром называется система, в которой на каждом такте используется переменный, выбираемый с помощью элементов ключевого потока, алгоритм шифрования [5,30].

Ключевой поток определяется исходными ключевыми данными и номерами тактов шифрования, вплоть до рассматриваемого.

Потоковые шифры, очевидно, более чувствительны к нарушениям синхронизации (вставка, пропуск), чем блочные. Для некоторой компенсации данного недостатка используются потоковые шифры с обратной связью.

В этих шифрах значение элемента ключевого потока на такте t вычисляется с помощью фиксированной функции f от ключа и нескольких знаков шифртекста, полученных на m предыдущих тактах.

В криптографической литературе под потоковым шифром очень часто понимают так называемый двоичный аддитивный потоковый шифр, представляющий собой шифр гаммирования по модулю два с псевдослучайной гаммой.

Для такого шифра ключевой поток можно записать с помощью нулей и единиц и непосредственно использовать для гаммирования открытого текста.

2.2.2. Качество гаммы

Очевидно, свойства потокового шифра зависят от свойств ключевого потока (гаммы). Если в ключевом потоке выявляются закономерности, то анализ шифра упрощается.

Например, если период гаммы короткий, то в случае неравновероятного открытого текста классический шифр гаммирования становится катастрофически слабым.

Нежелательными являются не только детерминированные, но и стохастические зависимости в гамме.

В 1926 году американский инженер Вернам предложил для зашифрования каждого бита открытого текста использовать свой ключ.

В системе Вернама требуется, чтобы источник ключей вырабатывал случайную равновероятную двоичную последовательность, знаки которой независимы и чтобы очередной бит открытого текста перешифровывался очередным битом ключа (гаммы).

К.Шеннон [1] показал, что система Вернама является недешифруемой даже при полном переборе всех вариантов последовательности гаммы, вследствие отсутствия критерия открытого текста. На практике применяются последовательности гаммы, по параметрам приближающиеся к требованиям системы Вернама.

При больших затратах на проведение соответствующих организационных мер и построение громоздкой системы технической защиты информации, с помощью специфических физических процессов, внеземных источников излучений и т.п., можно организовать систему накопления случайной гаммы.

Однако чисто случайная последовательность может (в принципе) содержать некачественные участки. Поэтому необходимо разрабатывать математические алгоритмы для ее корректировки, что приводит к необходимости формальной оценки качества гаммы. Тем не менее, криптосистемы, основанные на подобном подходе, существуют.

Более практичными являются криптосистемы, порождающие гамму с очень большим периодом. Периоды колоссальной величины необходимы, но не достаточны. Значения элементов на периоде должны быть непредсказуемы, по крайней мере, со значимой вероятностью. Разработка методов построения и анализа таких последовательностей является одной из основных задач криптологии.

2.2.3. Периодичность гаммы

В настоящее время в системах криптографической защиты информации широко используются так называемые шифраторы с внутренним носителем гаммы, реализующие потоковые шифры, генерирующие последовательности с очень большими периодами и хорошими статистическими свойствами [5,30]. Эти шифраторы основаны на комбинировании т.н. регистров сдвига.

Соответствующие генераторы гаммы, в подавляющем большинстве случаев, состоят из типичных узлов, основанных на комбинациях регистров

сдвига и функциях усложнения [9]. Наиболее простым узлом является т.н. регистр сдвига с линейными обратными связями (РСЛОС), генерирующий рекуррентную последовательность вида $x_{i+0} \oplus x_{i+k} \oplus \ldots \oplus x_{i+t} = x_{i+n}$, где $\oplus$ означает сложение по модулю два.

Например, последовательность, генерируемая РСЛОС с рекуррентным законом $x_{i+0} \oplus x_{i+2} = x_{i+5}$ и начальным состоянием 11000, имеет вид:

11000110111010100001001011001111000110111010100001001 0.

Подобные последовательности являются периодическими.

При соответствующем выборе параметров РСЛОС можно достичь максимально возможных значений периода равных 2^n - 1.

В случае короткого периода гаммы *d*, выписка отрезков криптограммы длины *d* друг под другом даст совокупность колонок, для каждой из которых использовался фиксированный знак гаммы.

Если открытый текст не является равновероятным, то каждая колонка позволяет определить один знак гаммы, с точностью до инверсии. В итоге, определяется гамма на всем периоде.

Непосредственно для генерации гаммы РСЛОС не подходят, т.к. соответствующие выходные последовательности являются предсказуемыми.

На практике применяются комбинации зависимых РСЛОС, взаимно влияющих на формирование своих последовательных заполнений, а гамма формируется с помощью сложных функций от значений промежуточных вычислений.

Для предсказуемой гаммы знание небольшого ее отрезка (например, при переборе значений открытого текста) позволяет упорядочить по вероятности варианты возможного продолжения гаммы. Это позволяет резко сузить количество возможных вариантов открытого текста и продолжать последовательное вскрытие текста по смысловому критерию.

Одним из общих подходов анализа потоковых шифров является декомпозиция автомата на соответствующие узлы и анализ выходных последовательностей узлов и шифратора в целом.

2.2.4. Блочные шифры

Необходимость применения криптографической защиты информации в сетях ЭВМ, в базах данных, в системах электронных платежей привела к широкому использованию программных средств шифрования. При этом оказалось, что программная реализация потоковых шифров, в ряде случаев, уступает в быстродействии шифрам другого типа, так называемым, блочным шифрам.

Блочным шифром называется система шифрования, использующая на каждом такте постоянный, выбранный до начала шифрования, в зависимости от ключей, алгоритм [2,5,8].

Поскольку зашифрование должно быть взаимно однозначным преобразованием, то блочные шифры являются шифрами замены с очень большим алфавитом. Знаки алфавита представляются в виде двоичных блоков данных фиксированной длины. Например, алгоритм ГОСТ 28147-89 предназначен для работы с блоками длиной 64 бита. В режиме простой замены этот шифр взаимнооднозначно отображает множество мощности 2^{64} на себя.

Существуют потоковые шифры, использующие блочный шифр в качестве узла генерации гаммы. В криптографии принято рассматривать подобные шифры как режимы работы соответствующего блочного шифра.

Например, в режимах шифрования алгоритм ГОСТ 28147-89 работает как шифр гаммирования по модулю два, используя двоичную гамму, выработанную в режиме, соответствующем блочному шифру.

2.2.5. Алгоритмические проблемы, связанные со стойкостью основных типов шифров

Для блочных шифров оценка стойкости связана с оценкой качества т.н. виртуальных таблиц замены, т.е. таблиц замены, представить которые целиком на носителе невозможно из-за большого объема данных. Блочный шифр является совокупностью виртуальных таблиц замены. Ключ служит для выбора таблицы, которая является неизменной в процессе шифрования отдельного сообщения.

Очевидно, что криптографические свойства шифра простой замены существенно зависят от ключей. Поэтому в блочном шифре ключи, вообще говоря, могут быть неравноценны.

Например, виртуальные таблицы могут осуществлять нестойкие преобразования над элементами (блоками) открытого текста. Кроме того, не исключается, что выбор одной и той же виртуальной таблицы может быть осуществлен с помощью различных ключей.

Общая проблема оценки качества блочного шифра сводится к задаче определения больших областей ключей, которым соответствуют подстановки, наиболее сложные для вскрытия шифра простой замены.

В ситуации, когда виртуальная таблица замены легко описывается формулами, могут быть сформулированы общие задачи, решение которых приводит к дешифрованию соответствующих криптосистем.

К подобным задачам сводится стойкость значительного числа современных асимметричных криптоалгоритмов.

Рассмотрим, например, степенную функцию вида $g(x) = x^e \bmod n$, где $n = pq$, p, q - различные простые числа. Для обращения этой функции достаточно решить задачу разложения числа n на сомножители. Эта задача

является алгоритмической проблемой, на которой основана стойкость распространенной криптосистемы RSA.

Аналогичная ситуация имеет место с дискретной экспонентой, т.е. функцией вида $f(x) = a^x \bmod p$, где p - большое простое число. Эта функция часто используется в процедурах аутентификации.

Вследствие конечности множества вычетов по модулю p, последовательность $a^k \bmod p$, $k = 1,2\ldots$, периодична. Наименьший период называется показателем (порядком) числа a по модулю p и обозначается $ord_p a$.

Известно, что функция $f(x) = a^x \bmod p$ при больших значениях x и $ord_p a$ ведет себя как односторонняя.

Обратная функция (дискретный логарифм) вычислительно нереализуема и задача дискретного логарифмирования также является алгоритмической проблемой, на которой основана стойкость ряда криптоалгоритмов.

Что касается качественных потоковых шифров, то фактически каждый такой шифр сводится к очередной, ранее неизвестной математической задаче, которая поддается решению лишь в частных случаях.

Тем не менее, можно сформулировать общую алгоритмическую проблему, лежащую в основе стойкости шифров модульного гаммирования.

Очевидно, модульное гаммирование можно рассматривать как процедуру искажения гаммы знаками открытого текста.

При неравновероятном открытом тексте это позволяет использовать связи в гамме для составления соответствующих систем уравнений, а затем рассматривать полученные уравнения как выполняющиеся для шифртекста, но с искаженными правыми частями. Относительно искажений известно лишь распределение вероятностей.

Данный подход приводит к общей проблеме решения систем уравнений с искаженными параметрами.

Так, для шифров гаммирования, построенных на использовании комбинаций регистров сдвига, общая проблема восстановления искаженной (нелинейной) рекуррентной последовательности является алгоритмической проблемой, на которой основывается стойкость шифра.

Глава 3.
ТЕСТИРОВАНИЕ УЗЛОВ КРИПТОСХЕМ КАК МЕТОД КОМПРОМЕТАЦИИ ШИФРОВ

Основное в этой главе…

- **Компрометация шифров……..36**
- **Задача тестирования линейной рекуррентной составляющей криптоузла…….……...……..37**
- **Качественная характеристика задачи восстановления параметров искаженной линейной рекурренты…..……..45**

В процессе внедрения и эксплуатации криптосистемы используется документация, содержащая описания, инструкции и контрольные примеры. В документацию, в частности, входит описание системы шифрования, в которой описываются алгоритмы шифрования и расшифрования, иерархия ключей, их использование при шифровании-расшифровании, а также процедуры ввода открытого текста и вывода текста шифрованного.

Обычно криптоалгоритм представляется в виде графической схемы и ее описания. По традиции, графическое представление криптоалгоритма называется криптосхемой, а ее описание – описанием криптосхемы.

Криптосхемы состоят из элементов – криптоузлов, которые могут объединяться в блоки.

В описании криптосхемы дается также механизм взаимодействия узлов.

В совокупности, представленных данных должно быть достаточно для создания программной модели криптосистемы и ее тестирования на контрольных примерах.

Одной из особенностей потоковых шифров является то, что число параметров, влияющих на их стойкость, существенно больше, чем для блочных шифров. Криптосхемы потоковых шифров создаются на основе комбинирования криптоузлов со специфическими характеристиками [30]. По этой причине при их синтезе необходимо учитывать не только угрозу дешифрования для известных типов криптоатак, но и возможность т.н. компрометации шифра.

Необходимость введения этого понятия связана с тем, что в ряде ситуаций возможность дешифрования сообщений злоумышленником не исключается, но и не является неизбежной.

3.1. Компрометация шифров

Неформально, шифр называется скомпрометированным, если с достаточно малой вероятностью ошибки криптоаналитик определяет, получена ли заданная

последовательность символов в результате зашифрования конкретным шифром, либо нет [31].

Хотя суть понятия компрометации состоит в том, что неудачный выбор некоторых параметров часто позволяет идентифицировать шифр по шифрованному тексту, на практике возможность компрометации шифра оценивается как предпосылка к существованию неизвестных разработчику криптоатак, специфических для данной шифрсистемы.

Очевидно также, что при построении криптосистемы необходимо стремиться к тому, чтобы максимально затруднить возможность идентификации используемого шифра, т.к. подобный подход существенно затрудняет задачу дешифрования.

Шифр может быть скомпрометирован в той или иной мере. Подобная оценка является качественной и рассматривается в пределах от определения типа шифра до вскрытия отдельных параметров узлов и элементов ключевой системы. Задача обычно сводится к выявлению в шифртексте наличия особенностей, присущих искаженной выходной последовательности какого-либо криптоузла. Для решения подобных задач обычно используется статистический подход, комбинируемый с методами оптимизации.

3.2. Задача тестирования линейной рекуррентной составляющей криптоузла

При синтезе криптосхем потоковых шифров широко применяются криптоузлы, основанные на т.н. регистрах сдвига с обратной связью [30,32].

Регистр сдвига с обратной связью состоит из двух частей: регистра сдвига и функции обратной связи. Ограничений на область определения функции обратной связи не накладывается. Сама функция может быть задана таблично.

Двоичный регистр сдвига с обратной связью – это последовательность битовых ячеек. Их количество называется длиной регистра. Во время работы содержимое ячеек изменяется. Исходное состояние регистра называется его начальным заполнением. Содержимое ячейки называется разрядом (с соответствующим номером).

В результате одного такта работы регистра генерируется один бит. Новый бит вычисляется как функция от битов, выбираемых из ячеек регистра с заранее определенными номерами. Указанные ячейки называются ячейками обратной связи, а функция – функцией обратной связи. Номера ячеек обратной связи называются точками съема обратной связи.

В такте работы вычисляется значение функции обратной связи, затем регистр сдвигается, скажем, влево, теряя левый крайний разряд и освобождая крайнюю правую ячейку. В эту ячейку помещается значение функции обратной связи. Выходом регистра является бит, снятый с фиксированной (обычно, с крайней правой) ячейки.

Простейшим случаем подобного регистра является уже знакомый нам регистр сдвига с линейной обратной связью (РСЛОС). Функция обратной связи такого регистра является суммой по модулю два содержимого ячеек обратной связи.

В результате нескольких тактов работы РСЛОС длины n с точками съема $(0, k_1, k_2, \ldots, k_r)$ и начальным заполнением $s_0 = (a_0, a_1, \ldots, a_{n-1})$ возникает т.н. рекуррентная последовательность $R(s_0) = a_0, a_1, \ldots, a_i \ldots$, для которой выполняется линейное рекуррентное соотношение вида

$$a_i \oplus a_{k_1+i} \oplus \ldots \oplus a_{k_r+i} = a_{n+i}, \ (i=0, 1, 2\ldots).$$

Напомним, что эта последовательность является периодической. Период не превосходит числа 2^n-1 и зависит от длины и набора точек съема регистра.

Регистры, генерирующие последовательности с периодами максимальной длины, существуют и используются в криптографии. Будем считать, что рассматриваемая рекуррента обладает максимальным периодом.

Заметим, что нашу рекуррентную последовательность можно записать через все элементы текущего заполнения регистра в виде $a_{n+i} + \sum_{j=0}^{n} b_{j+i} a_{j+i} = 0 \pmod 2$, где $b_{j+i} = 1$, при $j \in \{0,\ k_1,\ k_2, \ldots,\ k_r\}$ и $b_{j+i} = 0$ в противном случае.

Таким образом, легко видеть, что каждый элемент рекуррентной последовательности выражается в виде линейной комбинации элементов начального заполнения (относительно сложения по модулю два).

Предположим, что криптосхема потокового шифра гаммирования по модулю два генерирует гамму как выход с РСЛОС, а ключом является начальное заполнение. В этом случае шифртекст является искаженной рекуррентной последовательностью и задача дешифрования сводится к восстановлению начального заполнения регистра.

Что касается компрометации шифра, то соответствующим тестом может быть решение некоторой другой задачи, например, задачи определения точек съема обратной связи. При таком подходе мы расширяем область тестируемых узлов.

Как следствие, при положительном результате тестирования уже нельзя утверждать, что тестируемый узел является РСЛОС.

Действительно, в общем случае тестирование покажет лишь наличие линейной составляющей в шифртексте или выходной последовательности узла.

Например, последнее слагаемое в последовательности

$$\widetilde{a}_{n+i} = a_i \oplus a_{k_1+i} \oplus \ldots \oplus a_{k_r+i} \oplus a_{i+1} \cdot a_{i+2} \cdot a_{i+3} \cdot a_{i+4}$$

равно нулю с вероятностью 15/16, поэтому его присутствие может быть незаметным на фоне искажений, вносимых в рекурренту битами открытого текста.

С точки зрения определения наличия линейной составляющей, линейная и нелинейная рекурренты неразличимы, но в обоих случаях несомненным является наличие в криптосхеме типового узла.

3.3. Задача восстановления параметров искаженной линейной рекурренты

Задачи восстановления точек съема обратной связи и начального заполнения РСЛОС являются реальными задачами практической криптографии. Хотя они изучались многими авторами, для больших длин регистров, в общем случае, практически приемлемого решения не найдено.

3.3.1. Представление элементов рекурренты через элементы начального заполнения

Для рассмотрения некоторой качественной характеристики различия и связи указанных задач опишем работу РСЛОС в матричных обозначениях.

Пусть $R(s)$ - рекуррентная последовательность, генерируемая РСЛОС с начальным заполнением $s_0 = (a_0, \ldots, a_{n-1})$.

Последовательность $R(s)$ удобно рассматривать в виде последовательности $s_0, s_1, \ldots$ состояний регистра, т.е. последовательности векторов.

Введем матрицу:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & \ldots & 0 & b_0 \\ 1 & 0 & \ldots & 0 & b_1 \\ \ldots & \ldots & & \ldots & \\ 0 & 0 & 1 & \ldots & b_{n-2} \\ 0 & 0 & \ldots & 1 & b_{n-1} \end{pmatrix}, \; b_i \in \{0,1\}, \; b_0 = 1.$$

Очевидно, матрица **A** обратима и

$$\mathbf{s_0 A} = (a_0, \ldots, a_{n-1}) \cdot \mathbf{A} = \left(a_1, \ldots, a_{n-1}, \oplus_{j=0}^{n-1} a_j b_j\right).$$

Таким образом, для последовательных состояний регистра выполняется соотношение $\mathbf{s_k A} = \mathbf{s_{k+1}}$, поэтому $\mathbf{s_0 A}^k = \mathbf{s_k}$, $(k = 0,1,2\ldots)$. Отсюда следует, что если $\mathbf{s_0} = \mathbf{s_0^{(1)}} \oplus \mathbf{s_0^{(2)}}$, то $R(\mathbf{s_0}) = R(\mathbf{s_0^{(1)}}) \oplus R(\mathbf{s_0^{(2)}})$ (последовательности суммируются поэлементно).

Пусть $e_i (i = 0,1,\ldots n-1)$ - строки единичной матрицы порядка *n*. Из предыдущего следует, что $R(\mathbf{s_0}) = a_0 R(\mathbf{e_0}) \oplus a_1 R(\mathbf{e}_1) \oplus \ldots \oplus a_{n-1} R(\mathbf{e}_{n-1})$.

Подписав последовательности $R(\mathbf{e_i})$ друг под другом, мы получим матрицу $\mathbf{M} = (\mathbf{h_0}, \mathbf{h}_1, \ldots)$, состоящую из *n* строк и бесконечного числа столбцов $\mathbf{h_k}$. Легко видеть, что последовательность $\mathbf{h_0}, \mathbf{h}_1, \ldots$ является рекуррентой, удовлетворяющей тому же соотношению, что и $R(\mathbf{s_0})$. Начальным заполнением является последовательность столбцов $(\mathbf{h_0}, \mathbf{h}_1, \ldots, \mathbf{h_{n-1}})$. По построению эти столбцы являются столбцами единичной матрицы, таким образом, мы всегда можем вычислить любой из векторов $\mathbf{h_k}$.

В терминах столбцов матрицы **M** элемент a_k последовательности $R(\mathbf{s_0})$ равен $a_k = \oplus_{j=0}^{n-1} a_j h_j^{(k)}$, где $h_j^{(k)}$ - координаты вектора $\mathbf{h_k}$. Если мы рассмотрим выражение $\oplus_{j=0}^{n-1} a_j h_j^{(k)}$ как функцию от $a_0, a_1, \ldots, a_{n-1}$ с коэффициентами $h_j^{(k)}$ и обозначим ее $\langle \mathbf{s}, \mathbf{h_k} \rangle$, то $a_k = \langle \mathbf{s_0}, \mathbf{h_k}, \rangle$, $(k = 0,1,2\ldots)$.

Номера координат вектора $\mathbf{h_k}$, равные единице, указывают, какие компоненты вектора $\mathbf{s_0}$ участвовали в выражении $a_k = \langle \mathbf{s_0}, \mathbf{h_k}, \rangle$.

Таким образом, мы можем сказать суммой каких элементов начального заполнения является любой элемент a_k рекуррентной последовательности $R(\mathbf{s_0})$.

3.3.2. Производные соотношения

Вспомним, что $\mathbf{s}_0\mathbf{A}^k = \mathbf{s_k}$. Строки $\mathbf{e_0}\mathbf{A}^k = \mathbf{e_0}^{(k)}$, …, $\mathbf{e_{n-1}}\mathbf{A}^k = \mathbf{e_{n-1}}^{(k)}$, образуют некоторую квадратную подматрицу $\mathbf{H_k}$ матрицы $\mathbf{M}$. Столбцами этой подматрицы являются вектора $(\mathbf{h_k},,\ldots,\mathbf{h}_{n+\mathbf{k}-1})$. С другой стороны, $\mathbf{H_k} = \mathbf{E}_n\mathbf{A^k}$, где матрица $\mathbf{E}_n$ образована строками $\mathbf{e_i}$ и поэтому является единичной.

Поскольку матрица $\mathbf{A}$ обратима, то для всех целых k $\mathbf{H_k} = \mathbf{A^k}$, т.е. $\mathbf{AH_k} = \mathbf{H}_{\mathbf{k}+1}$.

В матрице $\mathbf{M}$ последний столбец матрицы $\mathbf{H}_{\mathbf{k}+1}$ следует за последним столбцом матрицы $\mathbf{H_k}$. Поэтому соседние столбцы в матрице $\mathbf{M}$ связаны соотношением $\mathbf{Ah_k} = \mathbf{h}_{\mathbf{k}+1}$.

Воспользуемся тем, что, по условию, $R(\mathbf{s_0})$ - рекуррента максимального периода. Из теории матриц известно, что в этом случае полином $f(x) = x^n \oplus b_{n-1}x^{n-1} \oplus \ldots \oplus b_1 x \oplus b_0$, связанный с матрицей $\mathbf{A}$, обладает важными свойствами [24].

Прежде всего, он является полиномом минимальной степени, аннулирующим $\mathbf{A}$: $f(\mathbf{A}) = \mathbf{0}$. Этот полином делит любой другой полином, аннулирующий матрицу $\mathbf{A}$. Фактически, соотношение $f(\mathbf{A}) = \mathbf{0}$ является иной формой записи рекуррентного закона для столбцов матрицы $\mathbf{M}$.

Далее. Полином $f(x)$ является неприводимым полиномом, более того, он является примитивным. В контексте нашей тематики последнее означает, что

если $\mathbf{h_0} \neq \mathbf{0}$, то множество векторов $\{\mathbf{A^k h_0}\}$ $(k = 0,1,2\ldots)$ однократно содержит все ненулевые вектора соответствующего линейного пространства, т.е. пространства $\mathbf{V_2^n}$. Следовательно, матрица **M** содержит все ненулевые вектора из $\mathbf{V_2^n}$.

Заметим, что поскольку двойка и ноль сравнимы по модулю два, то в полиноме $f^2(x)$, отсутствуют удвоенные произведения. Полиному $f^2(x)$ можно сопоставить матрицу размерности 2*n*, аналогичную матрице **A**.

Поскольку $f^2(x)$ делится на $f(x)$, то можно сделать вывод, что последовательность $R(\mathbf{s_0})$ удовлетворяет еще одному рекуррентному соотношению, даже с тем же числом слагаемых, что и исходное. В общем случае, любые подобные соотношения называются производными. Они соответствуют полиномам вида $f(x)g(x)$.

Покажем, что существуют производные трехчленные соотношения. Выберем из **M** два неравных столбца $\mathbf{h_0}$ и $\mathbf{h_m}$. Их сумма является столбцом в матрице **M**: $\mathbf{h}_0 \oplus \mathbf{h_m} = \mathbf{h_d}$. Поэтому равенство сохранится при умножении слева на матрицу *A*. Таким образом, для любого начального заполнения *s*, получим: $\langle \mathbf{s}, \mathbf{h}_0 \rangle \oplus \langle \mathbf{s}, \mathbf{h_m} \rangle = \langle \mathbf{s}, \mathbf{h_d} \rangle$. Следовательно, для рекурренты *R*(*s*) выполняется трехчленное соотношение $a_i \oplus a_{m+i} = a_{d+i}$, $(i = 0,1,2\ldots)$. Заметим, что, как правило, *d* чрезвычайно велико.

3.3.3. Некоторые сведения о подходах к восстановлению параметров искаженной линейной рекурренты

Если исходный полином $f(x)$ является трехчленным, то задачи восстановления номеров точек съема и начального заполнения решаются достаточно эффективно при реальных значениях параметров рекурренты.

Основными параметрами, влияющими на эффективность решения указанных задач, являются абсолютная величина отклонения вероятности искажения рекурренты от 1/2 и количество членов в рекуррентном соотношении.

При приближении вероятности искажения к 1/2 требуется все большее количество знаков искаженной рекурренты для ее восстановления.

Для истинной рекурренты биты с номерами $0+i, k_1+i, k_2+i, \ldots, k_r+i, n+i$ в сумме дают ноль для любого i. Соотношение $a_i \oplus a_{k_1+i} \oplus \ldots \oplus a_{k_r+i} \oplus a_{i+n} = 0$, а также любое производное соотношение, которому удовлетворяет рекуррента, называется также проверкой четности или проверочным соотношением.

Для искаженной рекурренты $c_0, c_1, \ldots$ проверочное соотношение не для всех значений i равняется нулю, поскольку последовательность искажений $u_0, u_1, \ldots$ ненулевая.

Вместе с тем, для последовательности $\{c_i\} = \{a_i \oplus u_i\}$ проверочное соотношение дает значения $u_i \oplus u_{k_1+i} \oplus \ldots \oplus u_{k_r+i} \oplus u_{n+i} = r_i$, которые не зависят от рекурренты.

Это позволяет составлять системы искаженных уравнений и использовать вероятностные оценки относительно u_i.

Очевидно, чем больше членов в вероятностном соотношении, тем ближе вероятности величин r_i к 1/2. Поэтому многие методы восстановления искаженной рекурренты используют, так или иначе, проверочные соотношения с минимально возможным количеством слагаемых [33].

Например, при не очень большой вероятности искажений можно оценить вероятность искажения бита c_k, рассматривая проверочные соотношения:

$$c_k \oplus c_{k_1+k} \oplus \ldots \oplus c_{k_r+k} \oplus c_{n+k} = r_k ,$$

$$c_{k-k_1} \oplus c_k \oplus \ldots \oplus c_{k_r+k-k_1} \oplus c_{n+k-k_1} = r_{k-k_1}$$

и т.д., где «шаблон» для выбора битов сдвигается на соответствующие значения k_i влево, пока обозначение c_{n+k} в шаблоне не окажется левее бита, первоначально обозначенного как c_k.

По построению этот бит входит во все рассмотренные проверочные соотношения. Используя производные соотношения, можно получить больше значений r_i.

Исходя из вероятностей искажений, можно заранее рассчитать вероятности r_i, в предположении, что $u_k = 1$ и проверить, согласуются ли полученные значения с распределением $\{u_i\}$.

По крайней мере, часть битов рекурренты может быть восстановлена.

Заметим, что этого может оказаться достаточно для восстановления всей рекурренты. Например, *n* подряд идущих битов позволяют развернуть рекурренту на нужное число шагов назад и определить начальное заполнение.

Тем же свойством обладают любые *n* битов, расположенные на местах, где соответствующие вектора $\mathbf{h_i}$ линейно независимы.

Аналогично можно использовать проверки на четность для отбраковки вариантов при поиске точек съема обратной связи [10].

3.3.4. Качественная характеристика задачи восстановления параметров линейной искаженной рекурренты

Качественную характеристику задачи восстановления параметров линейной искаженной рекурренты можно получить, используя следующий подход.

Пусть отрезку $c_0, c_1, \ldots, c_{N-1}$ искаженной рекурренты длины *N* при истинных значениях точек съема и истинном начальном заполнении

соответствует последовательность искажений, содержащая N_- единиц и $N_+ = N - N_-$ нулей. Пусть $\mathbf{z} = (z_0, z_1, \ldots, z_{n-1})$ - искомое начальное заполнение и $N_+ \geq N_-$. Рассмотрим предполагаемый вариант набора точек съема.

Обозначим отрезок рекурренты $R(z)$, соответствующий последовательности $c_0, c_1, \ldots, c_{N-1}$ через $t_0, t_1, \ldots, t_{N-1}$. Искажения имеют вид $u_k = c_k \oplus t_k$.

Преобразуем теперь значения битов с помощью преобразования (гомоморфизма) T: $T(0) = 1$, $T(1) = -1$. Легко проверить, что сумма битов по модулю два перейдет при этом в обычное умножение: $T(x \oplus y) = T(x)T(y)$. Обозначим $T(x)$ через $\hat{x}$.

Нам не известно начальное заполнение, однако мы знаем вектора $\mathbf{h_k}$.

Поскольку $t_k = \langle \mathbf{z}, \mathbf{h_k} \rangle$, то мы можем выразить t_k через $z_0, z_1, \ldots, z_{n-1}$ в виде суммы по модулю два, скажем $g_k(\mathbf{z})$. Из свойств преобразования T следует, что $\hat{u}_k = \hat{c}_k \hat{g}_k(z)$, где $\hat{g}_k(z)$ - произведение переменных из $z_0, z_1, \ldots, z_{n-1}$, номера которых известны, а $\hat{c}_k = \pm 1$.

Таким образом, мы можем образовать полином

$$P(z_0, z_1, \ldots, z_{n-1}) = \sum_{k=0}^{N-1} \hat{c}_k \hat{g}_k(z_0, z_1, \ldots, z_{n-1}).$$

Рассмотрим уравнение $P(\mathbf{z}) = B$, где B - целое число от нуля до N, а переменные удовлетворяют ограничениям $z_i = \pm 1$. Предположим, мы в состоянии при любом B ответить на вопрос, существует решение или нет, а также найти решение, если оно существует. Тогда, решая уравнение при каждом значении B от нуля до N, мы можем найти максимальное B, при котором решение существует, а также указать соответствующее решение.

Однако $P(z)=\sum_{k=0}^{N-1}\hat{u}_k(z)$, т.е. $B=P(z)=N_+-N_-$, откуда мы получаем оптимальное для данного набора точек съема значение N_-, т.к. N_++N_- известно и равно *N*.

Таким образом, определение точек съема обратной связи РСЛОС сводится к вопросу разрешимости уравнения $P(\mathbf{z})=B$, $z_i=\pm 1$, а нахождение начального заполнения регистра – к поиску самих решений.

Глава 4.
КРИПТОГРАФИЧЕСКИЕ СВОЙСТВА БУЛЕВЫХ ФУНКЦИЙ

Основное в этой главе…

- **Нелинейность булевой функции ……..……..50**
- **Критерии распространения и корреляционная иммунность..…..58**
- **Устойчивые булевы отображения……..…..62**

Анализ стойкости криптосхем, основанных на комбинировании регистров сдвига [34, 35, 36], естественным образом приводит к необходимости исследования математических свойств булевых отображений, под которыми понимаются отображения $f : GF_2^n \to GF_2^m$ $(n \geq m)$ конечномерных векторных пространств над полем GF_2 из двух элементов.

Интуитивно очевидна необходимость использования в криптографических приложениях отображений, преобразующих последовательности своих аргументов наиболее сложным, хаотическим образом.

На практике данный подход приводит к двум взаимосвязанным задачам: определить, какие формальные свойства отображений определяют его желаемую сложность (качество), а также указать эффективный алгоритм псевдослучайного выбора отображения, обладающего требуемыми свойствами.

Поскольку подобные отображения могут использоваться для построения секретных параметров, желательно, чтобы количество отображений заданного типа было достаточно велико. Это важно и с точки зрения случайности параметров, так как на практике функция с требуемыми свойствами псевдослучайно выбирается из класса отображений, как правило, более широкого, чем класс искомых функций.

4.1. Нелинейность булевой функции

Булевой функцией называется булево отображение вида $f: GF_2^n \to GF_2$.

Таким образом, область определения булевой функции состоит из $(0,1)$-векторов размерности *n*, количество которых равно 2^n. Операции с координатами векторов выполняются по модулю два. Булева функция принимает значения 0 или 1.

Булево отображение $f: GF_2^n \to GF_2^m$ можно представить в координатном виде как систему булевых функций $f = (f_1, f_2, \ldots f_m)$, где f_i $(i = 1, \ldots, m)$ - т.н. координатные функции.

Булевы функции часто представляются в виде многочленов (от нескольких переменных) над полем GF_2, а также задаются таблично. Существуют и другие формы представления булевых функций, удобные для исследования их свойств.

Таблица, представляющая булеву функцию $f(x)$, состоит из строк вида $x_1, x_2, \ldots x_n, f(x_1, x_2, \ldots, x_n)$, причем наборы аргументов $x_1, x_2, \ldots, x_n$ лексикографически упорядочены. Крайний правый столбец таблицы называется вектором значений функции $f(x)$. Количество булевых функций равно количеству векторов значений, т.е. 2^{2^n}.

Заметим, что в упорядоченном списке аргументов любые $k \le n$ фиксированных столбцов содержат все *k*-мерные двоичные наборы. Полная совокупность таких наборов встречается в указанных столбцах 2^{n-k} раз.

Обозначим количество единиц в векторе значений через $wt(f)$. Величина $wt(f)$ называется весом булевой функции.

При равновероятном и независимом выборе аргументов булевой функции f, вероятности ее значений, равных единице и нулю, соответственно равны $P(1) = wt(f)/2^n$, $P(0) = 1 - wt(f)/2^n$.

Как меру различия между булевыми функциями $f(x)$ и $g(x)$ от n переменных удобно использовать количество покоординатных несовпадений в векторах их значений. Данная мера $dist(f, g) = wt(f \oplus g)$ называется расстоянием Хэмминга между функциями f и g.

Расстоянием Хэмминга от функции $f(x)$ до заданного множества функций G называется значение $dist(f, G) = \min_{g \in G} dist(f, g)$.

Исследование свойств таблиц, представляющих булевы функции, часто оказывается целесообразным при анализе их криптографических свойств, т.к.

комбинаторные связи между подмножествами аргументов и значениями функций становятся более очевидными.

Широко известным свойством булевых отображений, важным для криптографических приложений, является равновероятность (сбалансированность, уравновешенность).

Это свойство заключается в том, что все элементы области значений имеют прообразы и эти прообразы имеют одинаковую мощность, т.е. для любого $v \in GF_2^m \ \#\left\{u \in GF_2^n : f(u) = v\right\} = 2^{n-m}$.

Вектор значений равновероятной булевой функций, таким образом, содержит одинаковое число нулей и единиц.

Свойством равновероятности обладают т.н. линейные функции вида $u(x) = a_1 x_1 \oplus a_2 x_2 \oplus \ldots \oplus a_n x_n$, где $a = (a_1, a_2, \ldots, a_n)$ - вектор коэффициентов, $a_i \in GF_2^n$.

По аналогии со скалярным произведением будем обозначать линейные функции через $\langle a, x \rangle$. Очевидно, равновероятными являются также аффинные функции вида $l_{ab} = l(a, b; x) = \langle a, x \rangle \oplus b$, где $b \in \{0,1\}$. Множество аффинных булевых функций от *n* переменных обозначим через A_n.

Практика показывает, что криптографические преобразования, обладающие свойствами близкими к свойствам линейных функций, во многих случаях приводят к существенному снижению стойкости шифров. По этой причине в криптографии важное значение имеют функции, свойства которых исключают слабости, присущие функциям, близким к линейным.

Таким образом, желательным качеством функции является ее нелинейность, понимаемая в широком смысле: как отрицание линейности.

Конкретная трактовка нелинейности, реально определяющая тот или иной класс функций, может быть различной.

Нелинейностью булевой функции f от n переменных называется параметр N_f - расстояние от f до множества аффинных функций, т.е. значение $N_f = \min_l dist(f,l),\ l \in A_n$.

Определение нелинейности расширяется на булевы отображения $f = (f_1, f_2, \ldots f_m)$, исходя из свойств совокупности всех нетривиальных линейных комбинаций $l(c,f) = c_1 f_1 \oplus c_2 f_2 \oplus \ldots \oplus c_m f_m$ координатных функций, с коэффициентами из GF_2.

В этом случае $N_f = \min_c N_{l(c,f)},\ c \in GF_2^n,\ c \neq 0$.

Рассмотрим наряду с функцией $f(x)$ функцию $\tilde{f}(x) = f(\mathbf{M}x + c)$, где $\mathbf{M}$ - невырожденная матрица порядка n, а $x, c \in GF_2^n$.

Преобразование (замена переменных) вида $y = \mathbf{M}x + c$ называется аффинным преобразованием пространства GF_2^n. Данное преобразование является взаимно однозначным, следовательно, если исходный вектор значений не изменять, то в табличном представлении функции $f(x)$ возникает перестановка векторов аргументов. Очевидно, эта перестановка не зависит от функции f.

Таким образом, $wt(f) = wt(\tilde{f})$. Кроме того, легко показать, что при указанной замене переменных любая аффинная функция переходит в аффинную.

Следовательно, при аффинном преобразовании координат области определения нелинейность булевой функции не изменяется.

Для изучения свойств нелинейности булевой функции $f(x)$ большое значение имеет преобразование Уолша-Адамара.

При преобразовании Уолша-Адамара элементы 0 и 1 поля GF_2 рассматриваются как обычные целые числа. В результате преобразования булевой функции $f(x)$ получается функция $W_f(a)$, $a \in GF_2^n$, связанная с вектором значений функции $h(f,a) = f(x) \oplus \langle a,x \rangle$. В этом векторе единицы соответствуют местам несовпадений правой части функции $f(x)$ с правой частью линейной функции $\langle a,x \rangle$.

Значением $W_f(a)$ является разность между числом нулей и числом единиц в векторе $h(f,a)$: $W_f(a) = \sum_x (-1)^{f(x) \oplus \langle a,x \rangle}$, где $x \in GF_2^n$ пробегает все множество аргументов.

Поскольку возведение минус единицы в целые степени, отличающиеся на четное слагаемое, дает одинаковый результат, то $W_f(a)$ можно выразить через обычное скалярное произведение и вещественное сложение: $W_f(a) = \sum_x (-1)^{f(x)+(a,x)}$.

Если зафиксировать *f*, то, очевидно, можно считать, что аргументом функции $W_f(a)$ является линейная функция, заданная вектором $a \in GF_2^n$ и представлять преобразование Уолша-Адамара в виде соответствующей таблицы с целочисленным вектором значений.

В этом векторе элемент $W_f(a)$ однозначно определяет расстояние Хэмминга между функциями $f(x)$ и $l_a = \langle a,x \rangle$, а также количество аргументов, на которых они совпадают.

Действительно, если через n_+ и n_- обозначить соответственно количество нулей и количество единиц в векторе значений функции $f \oplus l_a$, то

$n_+ - n_- = W_f(a)$ и $n_+ + n_- = 2^n$, откуда $n_+ = 2^{n-1} + \frac{1}{2}W_f(a)$ и

$dist(f, l_a) = n_- = 2^{n-1} - \frac{1}{2}W_f(a)$.

Сопоставим вектору значений булевой функции g вектор $\hat{g} = T(g)$, в котором $\hat{g}(x) = (-1)^{g(x)}$. Очевидно, выражение $W_f(a) = \sum_x (-1)^{f(x)+(a,x)}$ можно рассматривать как скалярное произведение векторов $\hat{f}$ и $\hat{l}_a$.

Если при вычислении этого выражения вместо l_a использовать аффинную функцию $l_a \oplus 1$, то, поскольку $T(l_a \oplus 1) = -\hat{l}_a$, получим величину $-W_f(a)$. Поэтому выражение $n_+ = 2^{n-1} + \frac{1}{2}|W_f(a)|$ равно количеству нулей в векторе значений функции $f \oplus l_a$, либо функции $f \oplus l_a \oplus 1$, при $W_f(a) \geq 0$, $W_f(a) < 0$ соответственно.

Таким образом, чем больше абсолютное значение $|W_f(a)|$ коэффициента Уолша-Адамара, тем меньше расстояние от функции *f* до множества аффинных функций.

Следовательно, $N_f = \min_l dist(f, l) = 2^{n-1} - \frac{1}{2}\max_a |W_f(a)|$, $l \in A_n$, $a \in GF_2^n$.

Говорят, что функция $l_{ab} \in A_n$ является аффинным статистическим аналогом функции f, если вероятность их совпадения при случайном и равновероятном выборе аргументов $p(l, f) > 1/2$.

Очевидно, вероятности $p(l,f)$ можно вычислить через коэффициенты $W_f(a)$, исходя из соотношения $n_+ = 2^{n-1} + \frac{1}{2}W_f(a)$, т.е. $p(l,f) = \frac{1}{2} + \frac{1}{2^n}\left(\frac{1}{2}W_f(a)\right)$.

Последовательность $\left\{\frac{1}{2}W_f(a)\right\}$, где аргументы $a \in GF_2^n$ лексикографически упорядочены, называется статистической структурой функции *f*.

Исходя из значения $\sum_a W_f^2(a)$, можно оценить величину N_f следующим образом.

Заметим, прежде всего, что

$$\sum_a W_f^2(a) = \sum_a \left[\sum_x (-1)^{f(x)}(-1)^{(x,a)}\right]^2 = \sum_a \left[2^n + \sum_{\substack{y,z \\ y\neq z}} (-1)^{f(y)+f(z)}(-1)^{(y,a)+(z,a)}\right] =$$

$$= \sum_a \left[2^n + \sum_{\substack{y,z \\ y\neq z}} (-1)^{f(y)+f(z)}(-1)^{(a,y\oplus z)}\right] = 2^{2n} + \sum_{\substack{y,z \\ y\neq z}} (-1)^{f(y)+f(z)} \sum_a (-1)^{(a,y\oplus z)} = 2^{2n}.$$

Здесь внутренняя сумма $\sum_a (-1)^{(a,y\oplus z)}$ обращается в ноль при $y \oplus z \neq 0$.

Действительно, показатель $(a, y \oplus z)$ равен количеству единиц в векторе *a* на местах, соответствующих номерам единичных компонент вектора $y \oplus z$.

Поскольку в сумме участвуют все 2^n векторов $a \in GF_2^n$, то величина $(a, y \oplus z)$ принимает четные и нечетные значения равное число раз.

Далее. Предположим, что $\max_a \left|W_f(a)\right| < 2^{n/2}$. Тогда $\sum_a W_f^2(a) < 2^{2n}$, что невозможно. Таким образом, для любой булевой функции f, $\max_a \left|W_f(a)\right| \geq 2^{n/2}$ и, следовательно, $N_f = 2^{n-1} - \frac{1}{2}\max_a \left|W_f(a)\right| \leq 2^{n-1} - 2^{n/2-1}$.

В случае, когда $\max_a \left|W_f(a)\right| = 2^{n/2}$, значение N_f максимально. При этом, поскольку N_f является целым числом, n - необходимо четное.

Оказывается, существуют булевы функции, у которых все коэффициенты Уолша-Адамара равны $\pm 2^{n/2}$. Такие функции называются совершенными нелинейными функциями, бент-функциями или максимально нелинейными функциями.

Разница в названиях связана с тем, что соответствующие понятия вводились для различных классов функций, удовлетворяющих некоторым специфическим условиям. Позже выяснилось, что эти условия в случае булевых функций эквивалентны.

Можно показать, что множество $\left\{\left|W_f(a)\right|\right\}$ абсолютных величин коэффициентов Уолша-Адамара булевой функции f не изменяется при аффинной замене переменных.

Следовательно, класс максимально нелинейных функций является инвариантным относительно аффинных преобразований области определения их аргументов.

Следует отметить, что максимально нелинейные функции не являются равновероятными. Поэтому в криптографических приложениях важны методы построения равновероятных булевых функций, с нелинейностью близкой к максимальной: $N_{\max} = 2^{n-1} - 2^{n/2-1}, n = 2k$.

Известно, что возможно построение равновероятных булевых функций с нелинейностью, удовлетворяющей неравенствам $2^{n-1} - 2^{n/2} \le N_g < N_{\max}$ и $2^{n-1} - 2^{(n-1)/2} \le N_g < N_{\max}$ для четного и нечетного *n* соответственно.

4.2. Критерии распространения и корреляционная иммунность

Криптографическая практика показывает, что булевы функции, используемые для построения генераторов гаммы, кроме равновероятности и высокой нелинейности должны обладать рядом более тонких свойств, ослабляющих связи между выходами функций и последовательностями соответствующих аргументов [11].

Для формулировки соответствующих определений используются понятия подфункции булевой функции и ее производной.

Назовем производной булевой функции $f(x)$ по направлению $u \in GF_2^n$, $u \neq 0$, функцию вида $D_u f(x) = f(x) \oplus f(x \oplus u)$. Введем также обозначение $D_{u,t} f(x) = D_u f(x)$, при ограничении $1 \le wt(u) \le t$.

Подфункцией булевой функции $f(x_1, x_2, \ldots, x_n)$, полученной фиксацией *k* переменных на местах $1 \le i_1 < i_2 < \ldots < i_k \le n$ с помощью вектора значений $a = (a_1, a_2, \ldots a_k)$, называется функция $f_I^a(x)$, определяемая строками таблицы $x_1, x_2, \ldots x_n, f(x_1, x_2, \ldots, x_n)$, для которых $x_{i_j} = a_j$, $j = 1, \ldots, k$.

В каждой такой строке аргументами функции $f_I^a(x)$ являются аргументы $f(x)$ с номерами, не принадлежащими множеству $I = \{i_1, i_2, \ldots, i_k\}$, а значение $f_I^a(x)$ равно $f(x_1, x_2, \ldots, x_n)$.

Изучение свойств булевых функций в связи с задачей конструирования преобразований, используемых при построении блочных шифров, привело к идее строгого лавинного критерия (Strict Avalanche Criteria, *SAC*).

Подобные критерии, в той или иной мере, отражают свойство непредсказуемости поведения функции f при модификации (неизвестных) аргументов. Например, требуется, чтобы минимальное отклонение от истинного значения аргумента приводило к максимально быстрому, лавинному, распространению ошибок в рекуррентной последовательности $x_{n+1} = f(x_1, x_2, \ldots, x_n)$.

Булева функция $f(x)$ удовлетворяет строгому лавинному критерию *SAC*, если $D_{u,1} f(x)$ - равновероятная функция для любого $u \in GF_2^n$.

Обобщением *SAC* является *SAC*(*k*) - строгий лавинный критерий порядка *k*, $1 \le k < n$. Функция $f(x)$ удовлетворяет *SAC*(*k*), если любая ее подфункция $f_I^a(x)$, полученная фиксацией *k* переменных, удовлетворяет обычному критерию *SAC*.

Заметим, что в самом *SAC* переменные не фиксируются, поэтому для критерия *SAC* часто используется обозначение *SAC*(0).

В критерии *SAC*(*k*), по определению, допускается модификация произвольного, но только одного из аргументов.

Модификация нескольких аргументов рассматривается в т.н. критерии распространения *PC*(*l*) степени *l* (Propagation Criterion).

Функция $f(x)$ удовлетворяет критерию распространения *PC*(*l*) степени *l*, если любая ее производная $D_{u,l} f(x)$, $u \in GF_2^n$, равновероятна.

В случае, если известно, что $D_{u,l} f(x)$ равновероятна для конкретного вектора *u*, говорят, что $f(x)$ удовлетворяет критерию распространения $PC(l)$ степени *l* относительно вектора *u*.

Общий случай приводит к формулировке критерия распространения степени *l*, порядка *k*.

Функция $f(x)$ удовлетворяет критерию распространения $PC(l,k)$ степени l, порядка k, $1<l\le n$, $1\le k<n$, если для любой ее подфункции $f_I^a(x)$, полученной фиксацией k переменных, все производные $D_{u,l}f_I^a(x)$, $u\in GF_2^n$, равновероятны.

Для таких функций, в частности, нецелесообразно при известной последовательности $\vec{u}_i=\vec{x}_{i+1}\oplus\vec{x}_i$, $\vec{u}_i,\vec{x}_i\in GF_2^n$, искать в короткой гамме вида $\gamma_i=f(\vec{x}_i)$ номера тактов, для которых биты гаммы зависимы. Причем это верно, даже если k битов аргумента $\vec{x}_1$, а значит k битов для всех $\vec{x}_i$, известны.

В самом деле, если функция удовлетворяет $PC(l,k)$, то при фиксации соответствующих k переменных зависимые биты гаммы γ_i, $\gamma_j=f\left(x_i\oplus\left(x_j\oplus x_i\right)\right)$ могут появиться лишь в том случае, когда $wt\left(\vec{x}_i\oplus\vec{x}_j\right)>l$. Очевидно, при заданном $\vec{u}$, вероятность повторения векторов вида $\vec{u}=\vec{x}_i\oplus\vec{x}_j$, содержащих l единиц на фиксированных местах, уменьшается с возрастанием l. Поэтому, чем больше l, тем длиннее требуется отрезок гаммы для появления зависимых битов.

Формулировки критериев распространения показывают, что подфункции булевых функций, приемлемых для соответствующих криптографических приложений, должны обладать специальными свойствами.

Со свойствами подфункций связано также понятие корреляционной иммунности порядка m, которое отражает отсутствие статистической зависимости между значениями функции $f(x)$ и подмножествами мощности m переменных, входящих в состав полного аргумента $x=x_1,\ldots,x_n$.

Функция $f(x)$ удовлетворяет критерию корреляционной иммунности порядка m, если для любой совокупности $I=\{i_1,i_2,\ldots,i_m\}$ номеров m

переменных $1 \le i_1 < i_2 < \ldots < i_m \le n$ и для любых наборов $a = (a_1, a_2, \ldots a_m) \in GF_2^m$ выполняется равенство $wt\left(f_I^a\right) = wt(f)/2^m$.

Рассмотрим случай $m = n$. Таблица, определяющая каждую подфункцию, состоит в этом случае из одной строки.

Свободных переменных нет, элемент вектора значений может равняться нулю или единице. Таким образом, подфункция $f_I^a = const$ и $wt\left(f_I^a\right) \in \{0,1\}$. Если критерий выполняется, $wt(f) \in \left\{0, 2^n\right\}$, т.е. функция $f(x)$ постоянна.

Оказывается, что только две функции достигают корреляционного иммунитета степени $m = n-1$: $f(x_1, \ldots, x_n) = x_1 \oplus \ldots \oplus x_n$ и $g(x_1, \ldots, x_n) = 1 \oplus x_1 \oplus \ldots \oplus x_n$.

Заметим, кроме того, что не все корреляционно иммунные функции, отличные от констант, являются равновероятными.

В случае, когда функция $f(x)$ равновероятна, $wt(f) = 2^{n-1}$. Для каждой пары *a,I* таблица, определяющая подфункцию, состоит из 2^{n-m} строк, количество переменных равно *n-m*. В этом случае критерий корреляционной иммунности приобретает вид $wt\left(f_I^a\right) = 2^{n-m-1}$, т.е. соответствующая подфункция должна быть равновероятной.

Критерий корреляционной иммунности эквивалентен следующему определению в вероятностных терминах.

Функция $f(x_1, \ldots, x_n)$ является корреляционно иммунной порядка *m*, если для любой совокупности $I = \{i_1, i_2, \ldots, i_m\}$ номеров *m* переменных $1 \le i_1 < i_2 < \ldots < i_m \le n$, $1 \le m \le n$, при любых значениях $a = (a_1, a_2, \ldots a_m) \in GF_2^m$ выполняется соотношение

$$c(m, a, I) = P\left(x_{i_j} = a_j \Big/ f(x) = 0, j = 1, \ldots m\right) = \frac{1}{2^m}.$$

Из определения следует, что для такой функции никакое подмножество *m* переменных не обладает особенностями, позволяющими сузить множество их возможных значений, исходя из распределения $c(m,a,I)$.

Оказывается, что функция $f(x_1,\ldots,x_n)$ является корреляционно иммунной порядка *m* тогда и только тогда, когда ее коэффициенты Уолша-Адамара удовлетворяют условию $\forall a \in GF_2^n:\ 1 \le wt(a) \le m,\ W_f(a)=0$. Следовательно, свойство корреляционной иммунности инвариантно при аффинных преобразованиях координат.

Для построения булевых функций, удовлетворяющих тем или иным критериям нелинейности, существуют различные методы.

Например, в [37] показано, что если функции $f_1(x_1,x_2,\ldots x_n)$ и $f_2(x_1,x_2,\ldots x_n)$ - корреляционно иммунные порядка *k*, то функция $f(u,x_1,x_2,\ldots x_n)=(u\oplus 1)f_1(x_1,x_2,\ldots x_n)\oplus uf_2(x_1,x_2,\ldots x_n)$ от $n+1$ переменной также корреляционно иммунная порядка *k*.

В качестве другого примера можно привести функцию *g* от $n=2k+1$ переменных вида $g(x_1,x_2,\ldots x_{2k+1})=x_1\oplus f(x_1\oplus x_2,\ldots,x_1\oplus x_{2k+1})$, где $f(x)$ - максимально нелинейная функция от *n* переменных [11].

Заданная таким образом функция *g* удовлетворяет критерию распространения $PC(2k)$ и обладает высокой нелинейностью: $N_g \ge 2^{2k}-2^k$.

В качестве обобщения корреляционно иммунных функций можно указать т.н. устойчивые булевы отображения.

4.3. Устойчивые булевы отображения

Отображение $f\colon GF_2^n \to GF_2^k$, $f=(f_1,f_2,\ldots f_k)$, называется (n,k,d)-устойчивым, если для любых наборов $I=\{i_1,i_2,\ldots,i_d\}$, $1\le i_1<i_2<\ldots<i_d\le n$,

$a=(a_1,a_2,\dots a_d)\in GF_2^d$ производное отображение $f_I^a\colon GF_2^{n-d}\to GF_2^k$ вида $f_I^a(x)=\left(f_{1I}^a(x),\dots,f_{kI}^a(x)\right)$ является равновероятным.

Устойчивое отображение $f=(f_1,f_2,\dots f_k)$ называется линейным, если линейны все его функции-компоненты.

Согласно определению равновероятности булевого отображения, область значений вектор-функции $f_I^a(x)$ совпадает с GF_2^k и мощности прообразов элементов из GF_2^k одинаковы. Поскольку мощность области определения $f_I^a(x)$ равна 2^{n-d}, то мощность каждого прообраза равна 2^{n-d-k}, следовательно, необходимо $n-k\ge d$.

Можно показать, что отображение $f\colon GF_2^n\to GF_2^k$, $f=(f_1,f_2,\dots f_k)$, $1\le k\le n$ является (n,k,d)-устойчивым тогда и только тогда, когда для любого набора $b=(b_1,b_2,\dots b_k)\in GF_2^k$ функция $\varphi=b_1f_1\oplus b_2f_2\oplus\dots\oplus b_kf_k$ является $(n,1,d)$-устойчивой, т.е. равновероятной, корреляционно иммунной функцией порядка *d*.

Для построения устойчивых отображений существует ряд подходов, позволяющих строить их, например, из устойчивых отображений меньшей размерности [38, 39].

Кроме того, практическое значение имеет подход, позволяющий строить устойчивые отображения из линейных устойчивых отображений, основанный на следующем утверждении [40].

Пусть $f\colon GF_2^n\to GF_2^k$ является (n,k,d)-устойчивым, а $g\colon GF_2^k\to GF_2^k$ - отображение, осуществляющее перестановку своих агрументов, то отображение $h=gf$ также (n,k,d)-устойчиво. Кроме того, нелинейность *h* удовлетворяет условию $N_h\ge 2^{n-k}N_g$.

Приведем примеры (n,k,d)-устойчивых линейных отображений [11].

1. $k = n-1, d = 1,\ f = (f_1, f_2, \ldots f_{n-1}), f_i(x) = x_i \oplus x_{i+1},\ i = 1, \ldots, n-1.$

2. $n = 3h, k = 2, d = 2h-1,$

$$f = (f_1, f_2), f_1(x) = x_1 \oplus \ldots \oplus x_{2h}, f_2(x) = x_{2h+1} \oplus \ldots \oplus x_{3h}.$$

В заключение, отметим значение инвариантности ряда свойств булевых функций относительно множества *G* аффинных преобразований координат $x = g(x')$.

Поскольку эти преобразования образуют группу, то на множестве булевых функций можно ввести т.н. отношение *G*-эквивалентности.

Функции $f(x)$ и $h(x)$ эквивалентны, если одна из другой получается аффинным преобразованием координат: $\exists g \in G:\ h(x') = f(g(x'))$.

При этом булевы функции разбиваются на непересекающиеся классы, каждый из которых содержит все функции, эквивалентные некоторой одной функции, называемой образующей класса эквивалентности.

Любые свойства образующей класса эквивалентности, инвариантные относительно аффинных преобразований координат, переносятся на любую функцию из этого класса. Поэтому при изучении совокупности функций ряд параметров можно получить, рассматривая лишь образующие классов.

Для примера рассмотрим множество *B* равновероятных булевых функций от четырех переменных.

Это множество разбивается на четыре класса эквивалентности [12] с образующими:

$$f^{(1)}(x) = x_1,\ f^{(2)}(x) = x_1 x_2 \oplus x_3,$$

$$f^{(3)}(x) = x_1 x_2 x_3 \oplus x_4,$$

$$f^{(4)}(x) = x_1 x_2 x_3 \oplus x_1 x_4 \oplus x_2.$$

Табличный вид образующих представлен в следующей таблице.

Таблица 2. Образующие классов аффинной эквивалентности функций

x_1	x_2	x_3	x_4	$f^{(1)}$	$f^{(2)}$	$f^{(3)}$	$f^{(4)}$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0
0	0	1	1	0	1	1	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	1	1
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	0	1	0
1	1	1	1	1	0	0	1

Покажем, что для каждой функции $f \in B$ существует аффинный статистический аналог, совпадающий с f с вероятностью не менее 3/4.

Значения преобразования вида $(-1)^{\langle a,x\rangle}$ линейных функций $l_a = \langle a,x\rangle$, для $a = 1,2,4,8$, представлены ниже (тетрады аргументов и коэффициентов функции записаны в виде шестнадцатиричных чисел).

Таблица 3. Последовательности значений функций $(-1)^{\langle a,x\rangle}$, $a = 1,2,4,8$.

a	Значения аргумента $x = 0,1,\ldots,E,F$
1=0001	+1,-1,+1,-1,+1,-1,+1,-1,+1,-1,+1,-1,+1,-1,+1,-1
2=0010	+1,+1,-1,-1,+1,+1,-1,-1,+1,+1,-1,-1,+1,+1,-1,-1
4=0100	+1,+1,+1,+1,-1,-1,-1,-1,+1,+1,+1,+1,-1,-1,-1,-1
8=1000	+1,+1,+1,+1, +1,+1,+1,+1,-1,-1,-1,-1,-1,-1,-1,-1

При вычислении статистической структуры получаем большие коэффициенты: $W_{f^{(1)}}(8) = 16$, $W_{f^{(2)}}(2) = 8$, $W_{f^{(3)}}(1) = 12$, $W_{f^{(4)}}(4) = 8$.

Следовательно, исходя из минимального значения $W_f(a)=8$, вероятность совпадения значений любой образующей со значениями соответствующей аффинной функции не ниже $n_+/16=3/4$. Значит, тем же свойством обладают все функции $f \in B$.

Рассмотренные примеры понадобятся нам при рассмотрении подхода к построению долговременных ключей алгоритма ГОСТ 28147-89.

Глава 5.
ОСОБЕННОСТИ ПРИМЕНЕНИЯ АЛГОРИТМА ГОСТ 28147-89

Основное в этой главе…

- **Криптоэквивалентная схема алгоритма ГОСТ 28147-89……....69**
- **Влияние блока подстановки на последовательности выходов итераций……..……70**

Алгоритм криптографического преобразования, установленный ГОСТ 28147-89, используется для зашифрования данных в двух режимах, а также для выработки имитовставки [13].

При зашифровании данных ГОСТ сводится к шифру блочного гаммирования с длиной блока в 64 бита. Гамма накладывается поразрядно по модулю два.

Основная задача каждого из режимов гаммирования - формирование 64-х битовых блоков для входа в блочный шифр - основной режим работы ГОСТ, называемый режимом простой замены (далее -*GA*).

Для шифрования каждой криптограммы вырабатывается 64-х битовый несекретный псевдослучайный блок *S*, называемый синхропосылкой. Этот блок служит параметром в ходе некоторых итеративных преобразований.

Выход из блочного шифра является собственно блоком гаммы.

Ключи необходимы для работы ГОСТ именно в этом режиме.

Имеется два типа ключей: долговременный - *K* и сеансовый – *X,* размером 512 и 256 битов соответственно. Ключ *K* реализует потетрадную замену 32-разрядных подблоков в 32-х разрядные и состоит из 8 узлов: $K = (K_1, \ldots K_8)$. В стандарте долговременный ключ называется блоком подстановки *K*.

Узел K_i является таблицей замены для *i*-ой (слева) тетрады, т.е. состоит из 16 тетрад. Ключ *X* состоит из конкатенации восьми 32-х разрядных подключей: $X = X_0, \ldots, X_7$, каждый из которых в соответствующий момент суммируется с некоторым подблоком по модулю 2^{32} (операция +).

Зашифрование блока в режиме простой замены представляет собой реализацию т.н. блочного шифра типа Фейстеля и состоит из 32 циклов. На каждом цикле происходит преобразование 64-х битового блока в 64-х битовый.

Результатом зашифрования является результат работы (выход) тридцать второго цикла, подвергнутый дополнительному преобразованию (перестановка подблоков, т.е. половинок блоков, местами).

Порядок перекомпоновки битов открытого текста при их вводе-выводе не зависит от ключей, описан в стандарте и для дальнейшего изложения несущественен.

5.1. Криптоэквивалентная схема алгоритма ГОСТ 28147-89

Криптоэквивалентным образом процесс простой замены блока S на блок T можно представить в виде последовательности тридцати четырех подблоков размером 32 бита каждый:

$$u = (U_{-2}, U_{-1}, U_0, U_1, U_2, \ldots, U_{30}, U_{31}), U_{-2} \| U_{-1} = S, U_{31} \| U_{30} = T .$$

Здесь $U_{i-1} \| U_i$ - результат работы цикла номер i. Дополнительное преобразование меняет порядок подблоков в выходном блоке цикла номер 31.

Элементы последовательности u связаны цепной зависимостью вида $U_{i-2} \oplus \widetilde{U}_{i-1} = U_i \ (i = 0,1,\ldots,31)$, где $\oplus$ - поразрядное сложение подблоков по модулю два.

Запись $\widetilde{U}_{i-1}$ означает, что подблок U_{i-1} модифицируется с помощью преобразования, зависящего от i. Точнее, $\widetilde{U}_{i-1} = F_i(U_{i-1})$, где F_i зависит от параметров, используемых в цикле с номером i.

Каждая из функций F_i использует ключ K и некоторый подключ сеансового ключа $X_{t(i)}$, где $t(i)$ - последовательность выбора подключей: $t(i) = (0,1\ldots7, 0,1\ldots7, 0,1\ldots7, 7,6\ldots1,0)$.

Преобразование $\widetilde{U}_{i-1} = F_i(U_{i-1})$ заключается в вычислении подблока $U^*_{i-1} = U_{i-1} + X_{t(i)} \bmod 2^{32}$, замене каждой тетрады подблока U^*_{i-1} с помощью

соответствующего узла K_j и циклическом сдвиге полученного подблока на 11 разрядов влево.

Обратим внимание, что, таким образом, $\widetilde{U}_{i-1}$ является циклически сдвинутым выходом с блока подстановки K.

Расшифрование происходит в соответствии с алгоритмом зашифрования, однако выбор подключей производится в обратном порядке, по отношению к $t(i)$. Подчеркнем, что это свойство не зависит от количества циклов и последовательности $t(i)$, используемых в преобразовании GA.

Два последних наблюдения позволяют сделать некоторые выводы о связи между виртуальными таблицами замены рассматриваемого блочного шифра и ключами.

5.2. Влияние блока подстановки на последовательности выходов итераций

Влияние долговременного ключа K на выходы циклов (последовательность u) можно проследить, исходя из того, что блок $U_{30}||U_{31}$ поразрядно отличается от входного блока $U_{-2}||U_{-1}$, на слагаемое $L||R$, где подблоки L и R являются линейными комбинациями вида $\widetilde{V}_1 \oplus \widetilde{V}_2 \oplus \ldots \oplus \widetilde{V}_h$.

Действительно, по определению, $U_{i-2} \oplus \widetilde{U}_{i-1} = U_i \ \ (i = 0,1,\ldots,31)$.

Таким образом, из $U_{-1} \oplus \widetilde{U}_0 = U_1$ и $U_1 \oplus \widetilde{U}_2 = U_3$ следует $U_{-1} \oplus \widetilde{U}_0 \oplus \widetilde{U}_2 = U_3$.

По индукции: $U_{2k+1} = U_{-1} \oplus \widetilde{U}_0 \oplus \widetilde{U}_2 \ldots \oplus \widetilde{U}_{2k}$, $(k = 0,1,\ldots,15)$.

Аналогично, $U_{2k} = U_{-2} \oplus \widetilde{U}_{-1} \oplus \widetilde{U}_1 \ldots \oplus \widetilde{U}_{2k-1}$, $(k = 0,1,\ldots,15)$.

Итак, левый и правый подблоки открытого текста, фактически, шестнадцать раз гаммируются поразрядным сложением выходами с блока подстановки в нечетных и четных циклах соответственно. Следовательно,

определенную информацию о свойствах «псевдогаммы» L и R можно получить, исходя из свойств K, без учета остальных параметров.

Рассмотрим долговременный ключ K как таблицу, колонки которой являются правыми частями булевых функций от четырех переменных.

Очевидно, при преобладании, скажем, нулей в колонке, при случайном равновероятном и независимом выборе аргументов, на соответствующем месте выходного блока вероятность нуля также будет завышена. В данном случае появление нуля или единицы являются событиями, соответствующими схеме Бернулли с постоянными вероятностями.

Наименьшее (на два бита) возможное преобладание нулей в колонке соответствует распределению вероятностей единицы (p) и нуля (q), при котором $q - p = 1/8$.

Действительно, пусть количество нулей и единиц в колонке равно z и e соответственно.

Очевидно, сумма нулей и единиц равна $z + e = 16$, а их разность равна преобладанию: $z - e = h$. Поэтому h - минимальное четное число большее нуля, т.е. $h = 2$, откуда $q - p = (z - e)/16 = 1/8$.

Легко получить формулу для вычисления вероятности нуля в сумме из k битов при заданном преобладании $q - p = \delta$: $P(0) = 1/2 + 1/2(\delta)^k$.

Если $h = 2$, то после суммирования шестнадцати подблоков псевдогаммы бит на выходе GA совпадает с битом на входе с вероятностью $1/2 + 1/2(1/8)^{16} = 1/2 + (1/2)^{49}$. Следовательно, возможна генерация ослабленной гаммы.

Например, если каждый узел замены K_i содержит лишь одинаковые тетрады, скажем m_i, то блок $L||R$ равен нулю, как поразрядная сумма шестнадцати одинаковых блоков $m_1||\ldots||m_8$. В этом случае U_{-2} и U_{-1} лишь меняются местами и все сеансовые ключи являются криптоэквивалентными.

Таким образом, качество псевдогаммы полностью зависит от свойств булевых функций, составляющих блок подстановки *K*. В частности, видно, по какой причине функции, для которых $p-q=0$, предпочтительнее остальных.

5.2.1. Расшифрование в режиме простой замены

Процесс расшифрования в режиме простой замены является типичным для ряда блочных шифров. Остановимся на нем подробнее, т.к. его свойства мы существенно используем в дальнейшем.

Вспомним, что зашифрование в режиме простой замены $T=GA(S)$ представляется в виде: $u=(U_{-2},U_{-1},U_0,U_1,U_2,\ldots,U_{30},U_{31})$, $U_{-2}||U_{-1}=S$, $U_{31}||U_{30}=T$.

При расшифровании используется обратный порядок подключей, а роль $U_{-2}||U_{-1}$ играет блок *T*.

Суть механизма расшифрования состоит в повторении вычисления значений цикловой функции на циклах 31, 30,..., 1, 0, т.е. подблоков вида $\widetilde{U}_{i-1}$.

Если U_i поразрядно сложить с $\widetilde{U}_{i-1}$, то из соотношения $U_{i-2}\oplus\widetilde{U}_{i-1}=U_i$ следует, что результат будет равен U_{i-2}. Таким образом, исходя из $U_{31}||U_{30}=T$, $i=31$, мы получим U_{29}. Аналогично, исходя из $U_{30}||U_{29}$, получим U_{28} и т.д.

При этом именно дополнительное преобразование выходного блока сводит расшифрование к изменению порядка следования подключей.

Действительно, расшифрование представляется в виде $\hat{u}=(U_{31},U_{30},U_{29},U_{28},\ldots,U_0,U_{-1},U_{-2})$, $U_{31}||U_{30}=T$, а $U_{-2}||U_{-1}=S$ (после дополнительного преобразования результата последнего цикла алгоритма).

Подчеркнем, что никакие свойства блока подстановки K на осуществимость процесса расшифрования не влияют.

5.2.2. Возможность ослабления шифра за счет структуры сеансового ключа

Заметим, что выбор K с равновероятным распределением битов в колонках, строго говоря, не обеспечивает качественную псевдогамму $L||R$, т.к. для каждого K существуют блок B и сеансовый ключ X, такой, что $GA(B) = B$.

Действительно, рассмотрим четырехцикловой алгоритм $G\widetilde{A}$ с последовательностью $t(i)=(0,1,2,3)$ и пусть $G\widetilde{A}^{-1}(T)=B$. Если выход после четвертого цикла $G\widetilde{A}(B)$ имеет вид $T=U||U$, то дополнительное преобразование его не меняет. Заметим, что для $G\widetilde{A}^{-1}$ $t(i)=(3,2,1,0)$. Поэтому для восьмициклового $G\widetilde{A}$ с последовательностью $t(i)=(0,1,2,3,3,2,1,0)$ получим $GA(B)=B$.

То же будет верно для тридцати двух циклов, при $t(i)=(0,1,2,3,3,2,1,0,\ldots,0,1,2,3,3,2,1,0)$, что при стандартном значении $t(i)$ эквивалентно использованию сеансового ключа вида $X=X_0,X_1,X_2,X_3,X_3,X_2,X_1,X_0$.

Таким образом, из сказанного ранее следует, что для алгоритма ГОСТ в режиме простой замены виртуальные таблицы замены могут отличаться по качеству в зависимости от ключей.

Наибольшее влияние на качество виртуальных таблиц оказывает долговременный ключ – блок подстановки K. Некорректный выбор ключа K может привести к ослаблению шифра, независимо от сеансовых ключей.

5.3. Замечания о режимах шифрования и имитовставки

Рассмотрим процесс формирования 64-х битовых блоков для входа в блочный шифр при режимах шифрования.

В режиме гаммирования для тактов шифрования $k = 1,2,\ldots$ последовательность 64-битовых блоков гаммы имеет вид: $\gamma_k = GA\left(f\left(\sigma_{k-1}\right)\right)$, где $\sigma_k = f\left(\sigma_{k-1}\right)$, блок σ_0 выражается через синхропосылку: $\sigma_0 = GA\left(S\right)$, а функция $f\left(\sigma\right)$ использует две несекретные константы c_1, c_2, т.е. $f\left(\sigma\right) = f\left(\sigma, c_1, c_2\right)$.

Для блока σ, состоящего из двух подблоков $\sigma = s_1 || s_2$, блок $f\left(\sigma\right)$ состоит из подблоков вида: $s_1 \circ c_1$, $s_2 + c_2$.

Здесь сложение производится по модулю 2^{32}, а $s_1 \circ c_1 = s_1 + c_1 \bmod\left(2^{32} - 1\right)$, за исключением случая $s_1 + c_1 = 2^{32} - 1$, когда результат принимается равным 2^{32}-1.

В режиме гаммирования с обратной связью очередной блок гаммы для такта $k = 1, 2,\ldots$ получается в результате зашифрования в режиме простой замены блока шифрованного текста, полученного на такте с номером k-1. Под блоком шифртекста для k=0 понимается синхропосылка S.

Легко видеть, что неравновероятность гаммы, вызванная слабым долговременным ключом, проявляется в режиме гаммирования в той же мере, что и в режиме простой замены.

Также очевидно, что блок подстановки, у которого каждый узел замены K_i содержит лишь одинаковые тетрады, полностью ослабляет режим гаммирования с обратной связью. Действительно, в этом случае гамма отличается от соответствующего блока на входе в режим простой замены лишь перестановкой полублоков. При этом сам входной блок известен, т.к. является предыдущим блоком шифртекста.

Итак, слабые для режима простой замены ключи снижают стойкость основных режимов шифрования.

Необходимо также учитывать, что выбор блока подстановки наугад не является наилучшим способом, т.к. можно убедиться, что в этом случае вероятность получения равного числа нулей и единиц в одном столбце ключа *K* менее 0,2.

Следовательно, построение долговременных ключей для алгоритма ГОСТ необходимо проводить специальным образом.

Отметим, что кроме режимов шифрования, в стандарте ГОСТ 28147-89 предусмотрен т.н. режим выработки имитовставки. Этот режим служит для защиты от несанкционированной модификации открытого текста после его зашифрования.

Обеспечение имитозащиты открытого текста является необходимым, поскольку для криптограмм модульного гаммирования открытый текст можно модифицировать, не изменяя гаммы.

Действительно, пусть $c = t + \gamma(n)$ - знак шифртекста при модульном гаммировании знака открытого *t* текста гаммой γ. Пусть знак *t* известен и его необходимо заменить на знак *s*.

Очевидно, $c + (s - t) = t + (s - t) + \gamma(n)$, поэтому, если в криптограмме заменить знак *c* на знак $c + (s - t)$, открытый текст новой криптограммы после расшифрования будет соответствующим образом изменен.

Глава 6.
ВЫБОР ДОЛГОВРЕМЕННОГО КЛЮЧА АЛГОРИТМА ГОСТ 28147-89

Основное в этой главе…

- **Область сильных ключей…….....78**
- **Сведения о требованиях к выбору блока подстановки……81**
- **Контроль долговременного ключа алгоритма ГОСТ 28147- 89…..…..84**
- **Подход к выявлению слабых долговременных ключей…….86**

Итак, мы убедились, что стойкость криптографического алгоритма ГОСТ 28147-89 существенным образом определяется выбором блока подстановки *K*.

Точнее, мы пришли к выводу, что при неудачном выборе *K* сеансовые ключи могут не обеспечивать надлежащий уровень безопасности информации. В таких случаях соответствующие ключи (*X, K*) называются слабыми.

В противоположность слабым ключам, ключи, обеспечивающие обусловленный уровень безопасности информации называются сильными ключами. Хотя знание структуры ключевого пространства является желательным, как правило (и это естественно) проанализировать шифр до такой степени, чтобы стало возможным описание множества слабых ключей, не удается.

Аналогично, не удается полностью описать и множество сильных ключей, но очень часто возникает ситуация, когда можно выявить часть множества сильных ключей достаточную для приложений.

6.1. Область сильных ключей

Подмножество множества сильных ключей, достаточно большое, чтобы противостоять методу полного перебора, называется областью сильных ключей.

Одним из подходов, позволяющим указать область сильных долговременных ключей для алгоритма ГОСТ, является подход, при котором таблица замены *K* рассматривается как шифр подобный гаммированию по модулю два: $K(h) = h \oplus F(h)$. Здесь отображение $F(h)$ зависит от открытого текста и определяется как $K(h) \oplus h = F(h)$.

Назовем $F(h)$ псевдогаммой, поскольку гамма, используемая в шифре гаммирования, от открытого текста не зависит.

6.1.1. Достаточность условия равновероятности псевдогаммы для выбора сильного блока подстановки

Исходя из рассмотренного ранее влияния блока подстановки на выходные последовательности итераций, мы приходим к выводу, что чем больше псевдогамма обладает свойствами равновероятной последовательности, тем больше эти свойства усиливаются в процессе итераций циклов, что приводит к выработке качественной гаммы для режимов шифрования алгоритма ГОСТ.

С другой стороны, известные нам примеры слабых ключей показывают, что неравновероятная псевдогамма ослабляет шифр, независимо от сеансовых ключей.

Таким образом, мы приходим к выводу, что максимальное соответствие свойств псевдогаммы свойствам равновероятной двоичной последовательности с независимыми, равновероятно распределенными элементами, характеризует сильный долговременный ключ.

Для построения области сильных долговременных ключей на этой основе, необходимо сформулировать, чем характеризуется соответствие свойств псевдогаммы свойствам равновероятной двоичной последовательности, а также показать, что количество ключей, удовлетворяющих выдвигаемым требованиям, достаточно велико.

Таким образом, необходимо исследовать свойства конкретного класса булевых отображений.

Как нам уже известно, существует ряд критериев, позволяющих выделить классы булевых функций, преобразующих последовательности аргументов в последовательности, близкие к случайным. Такие булевы функции будем называть криптографически сильными.

Оказывается, сильные долговременные ключи для алгоритма ГОСТ можно строить, используя подобные критерии.

Поясним связь псевдогаммы, образуемой блоком подстановки K, с булевыми функциями от четырех переменных.

Отображение $K(h)$ можно рассматривать как $(0,1)$-матрицу размера 16x32, а тетраду $x=(x_1, x_2, x_3, x_4)$, заменяемую с помощью узла замены K_i - как 0,1-вектор $x=(x_1, x_2, x_3, x_4)$ размерности четыре. Итак, $K_i(x)=y=(y_1, y_2, y_3, y_4)$.

Однако набор битов (y_1, y_2, y_3, y_4) можно получить, выбирая их не одновременно, а последовательно, что можно рассматривать как выбор значений четырех булевых функций (для каждого узла замены - своих) $y_1 = f_{1i}(x), y_2 = f_{2i}(x), y_3 = f_{3i}(x), y_4 = f_{4i}(x)$.

Таким образом, будем рассматривать каждую колонку матрицы K как правую часть «координатной» булевой функции $y_j = f_{ji}(x)$, записанной в табличном виде. Правые части функций $y_j = f_{ji}(x), j = 1,\ldots,4$, входят в состав узла K_i, $i=1,\ldots,8$.

Соответствующая тетрада псевдогаммы равна $x \oplus y$, поэтому любой бит блока псевдогаммы представляется в виде $\gamma_i = x_i \oplus g_i(x)$, причем x_i входит в состав тетрады x. Следовательно, биты псевдогаммы можно переобозначить: $\gamma_i = \gamma_i(x), \quad i = 1,2,\ldots 32$. Аналогично можно переобозначить координатные функции (колонки) матрицы K: $y_i = f_i(x)$.

Вообще говоря, для построения криптографически сильных функций могут разрабатываться все новые критерии и выдвигаться все новые и новые требования. Оказывается, в нашем конкретном случае из-за небольшого числа переменных некоторые критерии вырождаются, а некоторые – являются неприемлемыми, т.к. недопустимо сужают область сильных ключей.

Последнее обстоятельство, конечно, обусловлено параметрами криптоалгоритма. Мы, очевидно, не можем улучшить эти параметры, т.к. рассматриваем задачу, в которой необходимо найти область сильных ключей, при условии, что криптоалгоритм задан.

6.1.2. Сведения о требованиях к выбору блока подстановки

Чтобы качество псевдогаммы γ_i и количество порождаемых ключей были удовлетворительными, как показал соответствующий анализ, достаточно обеспечить небольшое число условий, налагаемых на т.н. координатные функции $y_i = f_i(x)$ [12].

Кроме того, при решении данной задачи ответы на некоторые существенные вопросы были получены с помощью ЭВМ, поэтому дальнейшее изложение проведем, в основном, на концептуальном уровне.

Прежде всего, заметим, что если какой-либо узел замены не является взаимно однозначным отображением (подстановкой), то распределение выходных блоков на выходе *K* не будет равновероятным, вследствие сужения множества значений узла.

Отсюда следует первое требование: все узлы замены должны быть подстановками.

Если узел является подстановкой, то все его координатные функции равновероятны, поскольку любая колонка в списке всех шестнадцати аргументов содержит равное количество нулей и единиц.

Необходимо также, чтобы блок подстановки не оказался слишком простым преобразованием, например, позволяющим связать различные циклы. Наиболее простыми преобразованиями, как известно, считаются аффинные.

В нашем случае – это преобразования вида $K(x) = A(x) \oplus b$, где *x, b* - двоичные вектора размерности 32, *A* - невырожденная матрица.

Из сказанного, однако, не следует, что для подобных ключей известен метод дешифрования, просто нерационально пытаться обосновать стойкость шифра при аффинных ключах, если без них можно обойтись.

Если $K(x) = A(x) \oplus b$, то аффинными являются все узлы K_i. Для того, чтобы узел K_i не был аффинным, достаточно, чтобы его координатные функции также не были аффинными. Иными словами, для этого координатные функции не должны принадлежать множеству булевых функций от четырех переменных вида $a(x_1, x_2, x_3, x_4) = a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_4x_4 \oplus c$.

Однако, для неаффинной, равновероятной координатной функции может существовать близкий аффинный статаналог. Ранее мы убедились, что для любой равновероятной булевой функции от четырех переменных существует аффинный статаналог, совпадающий с ней с вероятностью не меньшей 3/4.

Поскольку мы ограничены в возможностях, то выдвинем **второе требование** в следующей форме: для любой координатной функции $y = f(x)$ допускается наличие аффинных статаналогов с вероятностью совпадения не более 3/4.

Дальнейшие требования к координатным функциям направлены на исключение зависимостей между входом и выходом узлов замены (следовательно, между входом и выходом всего блока K).

Одно из требований связано с порядком m корреляционной иммунности координатных функций.

Напомним, что корреляционный коэффициент $c(m, a, I)$ равен вероятности того, что m различных координат векторов-решений уравнения $f(x) = 0$ с номерами, принадлежащими списку I, принимают значения $(a_1, \ldots, a_m) = a$. Для корреляционной иммунности требуется, чтобы все корреляционные коэффициенты были равны.

Третье требование: корреляционные коэффициенты порядков 1 и 2 координатных функций могут принимать значения вида $c(1,a,I)\in\{1/2,1/2\pm1/8\}$ и $c(2,a,I)\in\{1/4,1/4\pm1/8\}$.

Можно убедиться, что разрешенное отклонение (1/8) от корреляционной иммунности является минимальным. Необходимость отклонения объясняется (с помощью ЭВМ) тем, что иначе область ключей получается слишком малой.

Определение. Коэффициентом размножения ошибки булевой функции $f(x)$ от n переменных по переменной x_i называется величина $K_i^f = {}^{1}\!/_{2^n}\sum_x\left(f(\ldots x_i\ldots)\oplus f(\ldots\bar{x}_i\ldots)\right)$. Здесь $\bar{x}_i = x_i\oplus 1$, а суммирование в правой части производится по всем аргументам функции.

Коэффициент размножения ошибки показывает, как ведет себя в среднем значение функции при изменении соответствующего (неизвестного) бита аргумента.

Значение криптографически сильной функции должно вести себя непредсказуемо. Следовательно, оптимальное значение $K_i^f = 1/2, \forall i$. Очевидно, в этом случае функция удовлетворяет строгому лавинному критерию (*SAC*).

Расчеты на ЭВМ показали, что в нашем случае это значение можно обеспечить лишь для слишком малого количества ключей, т.е. соответствующий критерий необходимо ослабить.

В итоге, возникает ч**етвертое требование:** коэффициент размножения ошибки координатных функций должен удовлетворять условию $K_i^f \in[1/4,3/4]$.

Используя ЭВМ, удалось выяснить, что количество булевых функций от четырех переменных, удовлетворяющих всем четырем условиям, равно 2328.

Число подстановок (т.е. узлов K_i), которое можно из них построить, равно $2^{11} \cdot 3^2 \cdot 864047$. В среднем одна такая подстановка приходится на одну из 1313,7 подстановок шестнадцатого порядка.

Таким образом, мощность области сильных ключей (комбинаций восьми подстановок) – величина порядка 10^{80}.

6.2. Контроль долговременного ключа алгоритма ГОСТ 28147-89

При создании средств шифрования необходимо не только обспечить стойкость шифра к методам криптоанализа, но и учитывать возможность доступа к открытому тексту неаналитическим путем, а также путем внедрения лазеек.

Например, при угрозе хищения ключей, как известно, необходимо предусмотреть систему управления ключами безопасную в отношении их хранения. Следует учитывать также возможность подмены долговременных ключей, если не исключено, что стойкость криптоалгоритма связана со спецификой их выбора. С целью защиты от подобных угроз используются средства защиты от физического доступа к компонентам шифраппаратуры, перешифрование, дефрагментация ключей и т.п.

Генерация ключей может проводиться с участием третьей стороны с использованием сертификатов ключей.

6.2.1. Угроза внедрения слабых параметров

В некоторых работах по криптографии [41] учитывается возможность сознательной генерации третьей стороной т.н. слабых параметров, которые приводят к снижению стойкости соответствующих средств криптографической защиты информации.

Отметим следующие моменты, благоприятные с точки зрения практического внедрения подобных слабых параметров.

Во-первых, применяемая в криптосредствах защита долговременных ключей от несанкционированного доступа, как правило, эффективна в одинаковой степени как против злоумышленника, так и против законного пользователя.

Иными словами, законный пользователь обычно не в состоянии ни просмотреть соответствующий ключ визуально, ни протестировать его с точки зрения слабости, особенно при аппаратной реализации криптоалгоритма.

По этой причине факт генерации третьей стороной слабых ключей может оказаться незамеченным.

Во-вторых, результаты криптоаналитических исследований, публикуемые в открытой литературе, содержат как конкретные типы слабых параметров, так и методики их использования для снижения стойкости стандартных криптоалгоритмов. Вообще говоря, злоумышленник может воспользоваться ими непосредственно.

Не исключено, что в ходе исследований стойкости алгоритма ГОСТ могут быть выявлены более сложные, чем рассмотренные нами, типы слабых долговременных ключей, представляющие определенную опасность для шифрпереписки в случае их применения.

Таким образом, если возможность внедрения слабого долговременного ключа алгоритма ГОСТ допускается, то возникает необходимость его (одноразового) контроля со стороны пользователя, невзирая на применяемые средства защиты ключей от непосредственного доступа.

Подчеркнем, что при выполнении требований соответствующих стандартов использование алгоритма ГОСТ является безопасным, поскольку долговременные ключи вырабатываются компетентными организациями по

методикам, направленным не на исключение слабых ключей, а напротив, на генерацию сильных долговременных ключей.

6.2.2. Подход к выявлению слабых долговременных ключей

Одно из общеизвестных требований для сильных ключей состоит в том, чтобы узлы замены являлись подстановками. С учетом этого требования существует принципиальная возможность построения процедуры, выявляющей некачественные ключи методом «черного ящика», т.е. с помощью многократного тестирования шифрсредства.

Для ее изложения будем использовать уже известное нам криптоэквивалентное представление преобразования данных в режиме простой замены $T = GA(S)$:

$$u = (U_{-2}, U_{-1}, U_0, U_1, U_2, \ldots, U_{30}, U_{31}),\ U_{-2} \| U_{-1} = S,\ U_{31} \| U_{30} = T,$$

$$U_{i-2} \oplus \widetilde{U}_{i-1} = U_i,\ i = 0 \ldots 31.$$

Пусть в цикле i используется долговременный ключ K и некоторый подключ сеансового ключа $X_{t(i)}$, где $t(i)$ - последовательность выбора подключей: $t(i) = (0,1\ldots7, 0,1\ldots7, 0,1\ldots7, 7,6\ldots1,0)$. Как обычно, $\widetilde{U}_{i-1}$ является значением цикловой функции в цикле с номером i. Пусть также задан блок открытого текста вида $A=U||U$, т.е. $U_{-2}=U_{-1}=U$.

Покажем, как выбрать сеансовый ключ $X = X(K, U)$, такой, что блок A в результате зашифрования в режиме простой замены не изменяется: $A = GA(A)$.

Для этого из каждого узла замены K_j, j = 1,…,8, с произвольного места i_j, выберем тетраду h_j и построим конкатенацию тетрад вида $h=h_1||\ldots||h_8$. Будем считать, что числа i_j также являются тетрадами (представлены в шестнадцатиричной системе счисления).

Более точно, построим конкатенацию тетрад, позиции выбора которых из блока подстановки *K* заданы в виде вектора $I=(i_1,\ldots,i_8)$.

Вектор $I=(i_1,\ldots,i_8)$ назовем позицией соответствующего подблока *h* в *K*. Построим искомый ключ в виде $X = X(K, U) = X(h, U)$.

Проследим процесс шифрования, продвигаясь по последовательности *u*. Выберем подключ X_1 из условия $U+X_1\pmod{2^{32}}=I$, где под значением *I* понимается конкатенация его компонент, т.е. тетрад i_j.

В этом случае $\widetilde{U}_{-1}=h^1$, где показатель указывает, что к подблоку *h* применен циклический сдвиг на 11 разрядов влево. Из соотношения $U_{i-2}\oplus\widetilde{U}_{i-1}=U_i$ получим $U_0=U\oplus h^1$.

Выберем теперь подключ X_2 из условия $U_0+X_2\pmod{2^{32}}=$ $=(U\oplus h^1)+X_2\pmod{2^{32}}=I$, тогда $\widetilde{U}_0=h^1$ и $U_1=U\oplus h^1$, т.е. $U_1=U_0$. Положим теперь $X_3=X_2$, получим $\widetilde{U}_1=\widetilde{U}_0=h^1$.

Поскольку $U_0\oplus\widetilde{U}_1=U_2$ и $U_0=U\oplus h^1$, то $U_2=U$. То есть, $U_2=U_{-1}$.

Наконец, положим $X_4=X_1$. Тогда $\widetilde{U}_2=\widetilde{U}_{-1}=h^1$, откуда $U_3=U_1\oplus\widetilde{U}_2=U$.

Таким образом, шесть первых членов последовательности *u* имеют вид

$$(U_{-2},U_{-1},U_0,U_1,U_2,U_3)=(U,U,U\oplus h^1,U\oplus h^1,U,U).$$

Следовательно, для четырехциклового алгоритма GA_4 с ключами *K*, $\widetilde{X}=X_1\|X_2\|X_2\|X_1$ и порядком выбора подключей 1,2,3,4, выполняется равенство $A=GA_4(A)$. Поскольку выходной блок имеет вид $U||U$, то дополнительное преобразование (перестановка последних двух подблоков в последовательности *u*) несущественно. Кроме того, для ключа $\widetilde{X}$ прямой и обратный порядок следования подключей совпадают.

Значит, для ключа $X = X_1 \| X_2 \| X_2 \| X_1 \| X_1 \| X_2 \| X_2 \| X_1$ стандартного алгоритма ГОСТ выбор подключей в соответствии с последовательностью (0,1…7, 0,1…7, 0,1…7, 7,6…1,0) совпадет с восьмикратным выбором построенных нами подключей X_1, X_2 в порядке X_1, X_2, X_2, X_1.

Иными словами, $GA(A)$ совпадает с восьмикратным применением GA_4 к блоку A. Следовательно, $A = GA(A)$.

6.2.3. Свойства теста

Соотношение $A = GA(A)$ назовем тестом. Используем его для определения долговременного ключа K методом тестирования «черного ящика», на вход которого подаются блоки вида $A = U||U$ и соответствующие ключи вида $X = X(K,U) = X(h,U) = X_1 \| X_2 \| X_2 \| X_1 \| X_1 \| X_2 \| X_2 \| X_1$ при различных позициях подблока h.

Предположим, что блок $A = U||U$ и позиция I подблока h фиксированы, а неизвестное значение h определяется перебором. При каждом варианте h вычисляется ключ $X = X(h,U)$ и проверяется соотношение $A = GA(A)$.

Напомним, что истинное значение h, как было показано, проходит тест.

Предположим, что все узлы замены являются подстановками. Рассмотрим ситуацию, когда тест проходит вариант, являющийся ложным, обозначив истинные, т.е. не зависящие от наших предположений относительно h, значения рассматриваемых далее подблоков через z (с индексами). В этом случае $(U_{-2}, U_{-1}, U_0, U_1, U_2, U_3) = (U, U, U \oplus z_1^1, U \oplus z_2^1, U \oplus z_1^1 \oplus z_3^1, U \oplus z_2^1 \oplus z_4^1)$, где z_k^1 $(k = 1,2,3,4)$ - выходы цикловой функции на соответствующих тактах с подключами X_k.

Покажем, что $z_1 \neq z_2$. Очевидно, z_1 - истинное значение подблока, позиция которого равна *I*, причем $z_1^1 \neq h^1$, поскольку все K_i - подстановки.

Вход в блок подстановки *K* при переходе от третьего к четвертому члену последовательности *u*, имеет вид $\left(U + z_1^1\right) + X_2 \neq \left(U + h^1\right) + X_2 = I$, иными словами, позиция подблока z_2 отличается от позиции подблока z_1. Отсюда, поскольку все K_i - подстановки, следует, что $z_1 \neq z_2$.

Покажем теперь, что если подблок проходит тест, то для произвольных *K* выполняются равенства $z_1^1 = z_2^1 = z_3^1 = z_4^1$.

Действительно, для рассматриваемого отрезка последовательности *u* соотношение $U_{-1} \oplus \widetilde{U}_0 = U_1$ дает $\widetilde{U}_0 = z_2^1$. Аналогично, $\widetilde{U}_1 = z_3^1$.

Поскольку блок проходит тест, то $U_2 = U_3 = U$. Таким образом, $U = U \oplus z_1^1 \oplus z_3^1$, $U = U \oplus z_2^1 \oplus z_4^1$, т.е. $z_1^1 = z_3^1$, $z_2^1 = z_4^1$.

Однако z_4^1 - просто другая запись подблока $\widetilde{U}_2$ (вспомните процесс перехода от U_2 к U_3 в последовательности *u*). Поскольку $U_2 = U_3 = U$, то $z_4^1 = \widetilde{U}_2 = \widetilde{U}$ и, кроме того, вход в цикловую функцию равен *U*. Для вычисления $\widetilde{U}_2 = \widetilde{U}$ использовался подключ $X_4 = X_1$. Поэтому позиция подблока z_4 (она совпадает с входом в блок подстановки *K*) равна $U + X_1 \left(\mathrm{mod}\, 2^{32}\right) = I$.

Это влечет равенство $z_4 = z_1$. Но мы получили ранее, что $z_1^1 = z_3^1$, $z_2^1 = z_4^1$, откуда следует, что $z_1^1 = z_2^1 = z_3^1 = z_4^1$.

Поскольку для обратимых *K*, при ложных вариантах *h*, $z_1 \neq z_2$, то имеет место следующее утверждение.

Пусть $\exists K^{-1}$. Тогда для любой позиции I соответствующий подблок h принадлежит K тогда и только тогда, когда h проходит тест.

Рассмотрим теперь случай, когда отображение K необратимо. В этом случае существуют две позиции, в которых расположены одинаковые подблоки.

Истинное значение $h = z_1$ по-прежнему проходит тест, причем $z_1^1 = z_2^1 = z_3^1 = z_4^1$.

Напомним, что каково бы ни было предполагаемое значение h, мы таким образом строим ключ X_1, что «в черном ящике» при первом обращении к блоку подстановки K выбирается подблок, позиция которого равна I, т.е. подблок z_1. Поэтому и для истинного, и для ложного подблока, проходящего тест, в цепочке $z_1^1 = z_2^1 = z_3^1 = z_4^1$ фигурируют одни и те же величины. Покажем, что при ложном значении $h \neq z_1$, позиции подблоков z_1 и z_2 различаются, хотя сами подблоки равны.

Действительно, подключ X_2 выбирается из соотношения $I = \left(U \oplus h^1\right) + X_2 \left(\mathrm{mod}\, 2^{32}\right)$. Вспомним, что $\widetilde{U}_0 = z_2^1$. Это значит, что в соответствующем цикле из K был выбран подблок z_2. Соответствующая позиция, очевидно, равна $J = \left(U \oplus z_1^1\right) + X_2 \left(\mathrm{mod}\, 2^{32}\right)$. Поскольку $h \neq z_1$, то $I \neq J$.

Следствие. Если ложный блок проходит тест, то отображение K необратимо.

6.2.4. Тестирование долговременного ключа

Подход для тестирования долговременного ключа состоит в следующем. Пусть необходимо определить 128 тетрад, входящих в узлы замены и

расположенных в матрице K. Будем определять тетрады последовательно, зафиксировав предварительно U.

Выберем позицию $I=(i_1,\ldots,i_8)$ искомого подблока h. Пусть, например, $I=(i_1,\ldots,i_8)=(0,\ldots,0)$, что соответствует первой строке матрицы K.

Определим состав восьми тетрад, находящихся в позиции $I=(i_1,\ldots,i_8)$ матрицы K перебором.

Вычисляем соответствующий сеансовый ключ $X=(h,U)$. Подаем на вход «черного ящика» X и комбинацию $A = U||U$. Если на выходе получаем ту же комбинацию, то располагаем тетрады подблока $h = h_1||\ldots||h_8$ в соответствующих местах формируемого долговременного ключа. Очевидно, для этого достаточно 2^{32} циклов шифрования ГОСТ.

Остальные 120 тетрад можно определять последовательно, каждую в 16 вариантах. Например, для определения оставшихся тетрад узла замены K_1 можно зафиксировать $i_2,\ldots, i_8$ и выбирать позиции, отличающиеся лишь значениями i_1. Это соответствует выбору h с переменной левой тетрадой. Всего для восстановления обратимого блока подстановки K потребуется $2^{32}+1920$ циклов шифрования ГОСТ.

Качество блока подстановки K оцениваем, исходя из следующих соображений.

Если блок подстановки K восстановить не удалось, то он необратим, следовательно, не является корректным. Если восстановленный блок подстановки $\widetilde{K}$ оказывается необратимым, то он также не является корректным.

При необратимом блоке K, в принципе, можно восстановить ложный обратимый вариант долговременного ключа. Таким образом, необходимо проверить истинность восстановленного $\widetilde{K}$.

Для этого моделируем процесс зашифрования нескольких открытых текстов достаточной длины на реальных сеансовых ключах и долговременном ключе $\widetilde{K}$. Зашифровываем те же тексты на тех же сеансовых ключах с помощью «черного ящика». Если результаты не совпадают, то ключ K - не корректный.

Если результаты совпадают, то $\widetilde{K} = K$ и необходимо протестировать ключ на принадлежность к области сильных ключей. Принципиально это возможно, например, при обращении в компетентные организации.

Глава 7.
ЭЛЕМЕНТЫ ТЕОРИИ СРАВНЕНИЙ

Основное в этой главе…

- **Расширенный алгоритм Эвклида....95**
- **Функция Эйлера и малая теорема Ферма……97**
- **Сравнения первой степени от одного неизвестного……98**
- **Степенные сравнения по простому модулю……...100**

Внедрение средств криптографической защиты информации, основанных на асимметричных криптоалгоритмах, потребовало существенного расширения диапазона средств теории чисел, используемых в криптографии.

Тем не менее, терминология, используемая в рекомендациях стандартов, а также при описании криптоалгоритмов и результатов их анализа, основана, как правило, на ряде простых понятий теории чисел, которые мы кратко рассмотрим ниже. Более полно, см. [14,15,16].

7.1. Некоторые сведения из арифметики

Числа 1,2,3,… называются натуральными. Число 0, а также числа вида $\pm a$, где a натуральное число, называются целыми числами. Отношение двух целых чисел называется рациональной дробью и является формальной записью результата деления одного числа на другое. Деление на ноль не определено.

Простым числом называется натуральное число, у которого есть в точности два неравных натуральных делителя.

Основная теорема арифметики: каждое натуральное число m единственным, с точностью до порядка сомножителей, образом представляется в виде произведения степеней простых чисел: $m = p_1^a p_2^b \dots p_s^t$.

Наибольшим общим делителем двух целых чисел a и b называется наибольшее целое число, которое делит как a, так и b. Обозначения: (a,b) или НОД(a,b).

Наименьшим общим кратным целых чисел a и b называется число НОК$(a,b) = \frac{ab}{(a,b)}$, делящееся, очевидно, как на a, так и на b.

Числа a и b называются взаимно простыми, если $d = (a,b) = 1$.

Имеет место следующее утверждение. Пусть $d = (a,b)$. Тогда существуют целые числа x, y, являющиеся решением уравнения $xa + yb = d$.

7.1.1. Расширенный алгоритм Эвклида

Решение x, y, d уравнения $xa + yb = d$, $a>b$, $d = (a,b)$, можно найти с помощью т.н. расширенного алгоритма Эвклида. Очевидно, общий случай сводится к решению подобных уравнений, при положительных *a* и *b*.

Рассмотрим схему расширенного алгоритма Эвклида на примере числе 15 и 25. Мы будем находить остатки и (неполные) частные от деления двух чисел, т.е. пользоваться равенствами вида $A = kB + r$, где все числа целые и $0 \le r < B$.

Поскольку должно выполняться неравенство $a > b$, то изменим обозначения: $r_0 = 25$, $r_1 = 15$.

Выпишем последовательность строк:

$$r_0 = 25 \quad x_0 = 1 \quad y_0 = 0$$

$$r_1 = 15 \quad x_1 = 0 \quad y_1 = 1 \quad (d_2 = 1, r_2 = 10).$$

Пояснение. Делим r_0 на r_1 с остатком. Получаем: $r_0 = d_2 r_1 + r_2$ т.е. $25 = 1 \cdot 15 + 10$, откуда $d_2 = 1$, $r_2 = 10$. Проверяем: $r_2 = 0$? Нет – работаем дальше. Вычисляем x_2, y_2: $x_2 = x_0 - x_1 d_2 = 1$ $y_2 = y_0 - y_1 d_2 = -1$. Формируем очередную строку: r_2, x_2, y_2.

Исходными данными для шага 2 будут строки r_1, x_1, y_1 (из предыдущего шага) и r_2, x_2, y_2. С этими строками действуем аналогично.

Если очередной остаток от деления равен нулю, выписываем решение из данных предыдущего шага (см. ниже).

$$r_2 = 10 \quad x_2 = 1 \quad y_2 = -1 \quad (d_3 = 1, r_3 = 5)$$

$$r_3 = 5 \quad x_3 = -1 \quad y_3 = 2 \quad (d_4 = 2, r_4 = 0).$$

При формировании очередной строки остаток $r_4 = 0$. Выписываем решение из данных предыдущего шага:

$$\text{НОД}(25,15) = r_3 = 5,\ x = x_3 = -1\ y = y_3 = 2,$$
$$xr_0 + yr_1 = -25 + 30 = 5 = \text{НОД}(25,15).$$

7.2. Модульная арифметика

Каждое целое число *a* можно разделить с остатком на натуральное число m: $a = km + r\ \ 0 \le r < m$.

Остаток от деления числа на *m* называется вычетом (в данном случае - вычетом числа *a* по модулю *m*). Операция, сопоставляющая числу *a* его вычет по модулю *m*, называется приведением *a* по модулю *m*.

Определение. Два целых числа *a* и b сравнимы по модулю *m*, если их разность делится на *m*. Аналогия – тело, движущееся по окружности, периодически попадает в одну и ту же точку окружности, хотя проходит разный путь. Длины путей «сравнимы по модулю длины окружности».

Отношение сравнимости записывается в виде: $a \equiv b(\mathrm{mod}\, m)$, $a \equiv b \bmod m$, $a \equiv b(m)$. Вместо знака сравнения часто используется знак равенства. Кроме того, если это не вызывает недоразумений, указание модуля может отсутствовать.

В соответствии с данным определением числа, имеющие одинаковые остатки от деления на m, сравнимы по модулю *m*.

Если не оговорено противное, то стандартные значения вычетов по модулю *m*, принадлежат множеству $0,1,\ldots m-1$ (т.н. система наименьших неотрицательных вычетов).

Вычет суммы по модулю *m* равен сумме вычетов, приведенной при необходимости еще раз по модулю *m*. Аналогичным свойством обладает вычет

произведения. Таким образом, нахождение вычетов больших чисел можно свести к работе с числами, по абсолютной величине не превосходящими квадрата модуля.

Алгоритм Эвклида показывает, что для взаимно простых чисел *a* и *m* всегда существует число b такое, что $ab \equiv 1 \bmod m$.

Такое число называется обратным к *a* по модулю *m* и обозначается a^{-1}.

Рассмотрим степени числа *a* по модулю *m*, где *a* и *m* взаимно просты.

Пусть *m*=11. Вычеты степеней числа 2 для показателей 0,1,2,…,10 таковы: 1,2,4,8,5,10,9,7,3,6,1. Аналогично, те же степени числа 3 сравнимы соответственно с числами 1,3,9,5,4,1,3,9,5,4,1. В каждом случае имеется периодичность.

Пусть $(a,m)=1$. Наименьшее натуральное число δ, такое, что $a^{\delta} \equiv 1 \bmod m$ по модулю *m*, называется порядком (показателем) числа *a* по модулю *m*. Порядок числа *a* по модулю *m* обозначается $ord_m a$.

7.2.1. Функция Эйлера и малая теорема Ферма

Порядки чисел по модулю *m* различны. Существуют числа, являющиеся порядками одновременно для всех вычетов, взаимно простых с *m*. Одно из таких чисел равно значению т.н. функции Эйлера $\varphi(m)$, определяемой как количество чисел в последовательности $1,\ldots m-1,m$, взаимно простых с *m*.

Из этого свойства немедленно вытекает, что порядок каждого вычета делит $\varphi(m)$. Также очевидно, что при простом p, $\varphi(p)=p-1$.

Функция Эйлера является мультипликативной: если $(a,b)=1$, то $\varphi(ab)=\varphi(a)\varphi(b)$ и $\varphi(1)=1$.

Пусть $m = p_1^a p_2^b \ldots p_s^t$, тогда $\varphi(m)= p_1^{a-1} p_2^{b-1} \ldots p_s^{t-1}(p_1-1)\ldots(p_s-1)$.

Теорема Эйлера. Если $(a,m)=1$, то $a^{\varphi(m)} \equiv 1 \bmod m$.

Из теоремы Эйлера следует малая теорема Ферма: $a^{p-1} \equiv 1 \bmod p$, где p - простое, $(a,p)=1$.

Определение. Элемент (вычет) γ называется первообразным корнем по модулю *m*, если его порядок по модулю *m* равен $\varphi(m)$. При $m = p$, где p - простое, первообразные корни всегда существуют.

Пусть $p-1=\prod_i p_i^{a_i}$. Можно показать, что элемент γ является первообразным корнем по модулю p тогда и только тогда, когда $\forall i$ $\gamma^{(p-1)/p_i} \neq 1 \bmod p$.

Таким образом, любое число *a*, взаимно простое с p, сравнимо по модулю p с числом вида γ^h.

Множество вычетов с модульными операциями по простому модулю p является т.н. простым полем Галуа из p элементов. Обозначение: $GF(p)$.

При составном модуле *m* соответствующее множество вычетов полем не является, т.к. не все элементы обратимы, однако оно образует коммутативное кольцо с единицей Z/mZ. Естественно, кольцо Z/pZ - конечное поле из p элементов.

7.3. Сравнения первой степени от одного неизвестного

Сравнения вида $ax \equiv b \bmod m$ могут иметь несколько решений, иметь единственное решение или не иметь решений вовсе. Если $(a,m)=1$, то решение единственно: $x \equiv a^{-1}b$.

Решения существуют тогда и только тогда, когда $d=(a,m)$ делит b.

В этом случае, кроме исходного сравнения, разрешимо сравнение вида $\frac{a}{d}x \equiv \frac{b}{d} \bmod \frac{m}{d}$ с единственным решением x_o. Очевидно, все решения исходного сравнения в диапазоне $0 \le x \le m-1$ являются числами вида $x_o + j\frac{m}{d} \bmod m$, $j = 0,1,\ldots,d-1$. В частности, если *m* простое, то сравнение $ax \equiv b \bmod m$ при $a \neq 0$ всегда однозначно разрешимо.

Кроме того, можно показать, что для многочлена от одной переменной с коэффициентами из $GF(p)$ количество корней, лежащих в $GF(p)$, не превосходит степени многочлена.

7.3.1. Китайская теорема об остатках

Следующее утверждение называется **китайской теоремой об остатках**.

Пусть числа $m_1, m_2, \ldots, m_k$ попарно взаимно просты и $M = m_1 m_2 \ldots m_k$. Тогда среди наименьших неотрицательных вычетов по модулю *M* существует единственное решение системы сравнений $x \equiv c_i \bmod m_i$, $i = 1,\ldots,k$.

При этом, $x \equiv \sum_{i=1}^{k} c_i M_i N_i \bmod M$, где $M_i = m_1 m_2 \ldots m_{i-1} m_{i+1} \ldots m_k$, $N_i = M_i^{-1} \bmod m_i$.

Действительно, в указанном выражении для x одно слагаемое сравнимо с c_i по модулю m_i, а все прочие сравнимы с нулем.

Заметим, что коэффициенты $M_i N_i \bmod M$ можно вычислить заранее и решать несколько систем, подставляя их правые части в линейную форму.

7.3.2. Степенные сравнения по простому модулю

Напомним, что поле является множеством с двумя внутренними коммутативными операциями, подчиненными ассоциативному и дистрибутивному законам. Операции называются сложением и умножением.

По отношению к каждой из операций в поле существует нейтральный элемент (ноль и единица). Уравнение вида $ax = b$ при $a \neq 0$ имеет единственное решение. Уравнение $a + x = b$ всегда однозначно разрешимо.

Привычным примером бесконечного поля является множество рациональных дробей. Простое поле $GF(p)$ является конечным множеством.

Очевидно, что в поле $GF(p)$ результат сложения любого элемента самого с собой p раз равен нулю. Таким образом, умножение любого элемента $GF(p)$ на p дает ноль. Число p называется характеристикой поля $GF(p)$.

Оказывается, количество элементов конечного поля всегда является степенью простого числа вида $q = p^a$.

Если a>1, то поле $GF(p^a)$ содержит простое поле $GF(p)$ и называется его расширением. Характеристика поля $GF(p^a)$ равна p. Важно помнить, что операции в этом поле не являются операциями по модулю $q = p^a$.

Известно, что в каждом конечном поле существует т.н. первообразный элемент (генератор поля). Степени g, $g^2 = g \cdot g, \ldots$ первообразного элемента g представляют все ненулевые элементы поля. Поэтому $g^{q-1} = 1$. Поскольку $g^{h(q-1)} = 1$, то любой элемент поля $b \neq 0$ удовлетворяет соотношению $b^{q-1} = 1$.

Операции в полях «учитывают» свойства характеристики поля автоматически. Поэтому обычные сравнения по модулю p являются равенствами в соответствующем простом поле.

Важной задачей, имеющей приложения в криптографии, является решение степенных сравнений вида $x^k \equiv a \bmod p$.

Пусть g - первообразный элемент простого поля. Тогда существуют числа y и b, такие что $x = g^y$, $a = g^b$, поэтому $g^{ky} = g^b$. Число b называется дискретным логарифмом числа a по модулю p при основании g. Поскольку $ord_p g = p-1$, то $ky \equiv b \bmod (p-1)$. Свойства последнего сравнения полностью характеризуют разрешимость исходного. Любое его решение y приводит к решению $x = g^y$ сравнения $x^k \equiv a \bmod p$.

Очевидно, при больших значениях переменных решению задачи препятствует необходимость явного представления числа a в виде $a = g^b$.

В общем случае, задача нахождения b является вычислительно нереализуемой (проблема дискретного логарифмирования). Дискретные логарифмы часто обозначаются как ind или $D\lg$.

Известен следующий результат: пусть p - простое, $(a, p) = 1$, $h = ord_p a$, тогда сравнение $x^h \equiv 1 \bmod p$ имеет $\varphi(h)$ решений.

Следствие. При $h = \varphi(p)$ получаем, что число первообразных корней равно $\varphi(p-1)$.

Покажем, что разрешимость сравнения $x^n \equiv a(p)$ эквивалентна выполнению условия $a^{(p-1)/d} \equiv 1 \bmod p$, где $d = (n, p-1)$.

Переход к индексам показывает, что $n\, indx \equiv inda \bmod (p-1)$. Необходимое и достаточное условие разрешимости последнего сравнения заключается в том, чтобы $\operatorname{ind} a$ делился на $d = (n, p-1)$, т.е. $\operatorname{ind} a \equiv 0 \bmod d$.

Умножая модуль и обе части последнего сравнения на $\frac{p-1}{d}$, получим $\frac{p-1}{d} inda \equiv 0 \bmod(p-1)$, откуда следует, что условие $\operatorname{ind} a \equiv 0 \bmod d$ эквивалентно условию $a^{(p-1)/d} \equiv 1 \bmod p$.

Следствие. Разрешимость сравнения $x^2 \equiv a(\bmod p)$ эквивалентна выполнению условия $a^{(p-1)/2} \equiv 1 \bmod p$.

Рассмотрим пример на решение степенных сравнений.

Пример. Решить сравнение $39x^{21} \equiv 53(73)$.

Таблица дискретных логарифмов считается известной. (Конкретные значения рассчитаны при $p = 73$ и $g = 5$).

Логарифмируя обе части сравнения, получаем:

$$\operatorname{ind}39 + 2\operatorname{ind}x \equiv \operatorname{ind}53(72)\ \ 2\operatorname{ind}x \equiv \operatorname{ind}53 - \operatorname{ind}39(72)$$

$$2\operatorname{ind}x \equiv 53 - 65(72)\ \ 2\operatorname{ind}x \equiv -12(72).$$

Поскольку наибольший общий делитель модуля и коэффициента при неизвестном равен 3 (больше единицы) и делит правую часть, то сравнение разрешимо, однако имеет несколько решений.

Деля коэффициенты и модуль на 3, получаем $7\operatorname{ind}x \equiv -4(24)$. Умножая на 7 обе части сравнения, находим $\operatorname{ind}x \equiv -4(24) = 20(24)$. Поэтому $\operatorname{ind}x \equiv 20 + j24 \bmod 72$, где $j = 0,1,2$. То есть $\operatorname{ind}x \equiv 20,44,68 \bmod 72$. Поэтому $x \equiv g^{20}, g^{44}, g^{68} \bmod 73$.

Подставляя $g = 5$, окончательно получаем: $x \equiv 18,71,57 \bmod 73$.

Глава 8.
ВЫЧИСЛЕНИЕ КВАДРАТНОГО КОРНЯ В ПРОСТОМ ПОЛЕ

Основное в этой главе…

- **Квадратичные вычеты по простому модулю……..104**
- **Алгоритм нахождения квадратного корня в простом поле……..…109**

8.1. Квадратичные вычеты по простому модулю

Квадратичным сравнением называется сравнения вида $x^2 \equiv a(n)$, где x - неизвестный вычет.

Целое число a называется квадратичным вычетом по модулю n, если сравнение $x^2 \equiv a(n)$ разрешимо. Если сравнение разрешимо, то для составного модуля количество решений, как правило, больше двух.

Поиск решений квадратичного сравнения часто используется в криптографии.

Оказывается, что в общем случае не только данная задача, но даже вопрос о разрешимости квадратичного сравнения по составному модулю, факторизация которого неизвестна, является нерешенной проблемой.

В то же время для модулей, являющихся простыми числами, задача легко поддается анализу. Это позволяет исследовать задачу и в общем случае, поскольку, если $x^2 \equiv a(n)$, то a является квадратичным вычетом по модулю любого простого делителя числа n.

Пусть p - нечетное простое число. Очевидно, при $a = 0(p)$ имеется единственное решение сравнения $x^2 \equiv a(p)$: $x = 0(p)$.

Все ненулевые вычеты по модулю p находятся среди чисел $\pm 1, \pm 2, \ldots, \pm \frac{p-1}{2}$, следовательно, их квадраты составляют список $1^2, 2^2, \ldots, \left(\frac{p-1}{2}\right)^2$. Сравнение $x^2 \equiv a(p)$ имеет решение, если a принадлежит этому списку.

Далее, сравнение $x^2 \equiv k^2(p)$ имеет два очевидных решения: $\pm k$. Кроме того, количество решений не может превышать степени многочлена в левой части, т.е. двух.

Чтобы убедиться, что решений в точности два, достаточно показать, что $k \neq -k(p)$.Очевидно, если это не так, то $2k = 0(p)$, что верно только для $k = 0(p)$.

Заметим теперь, что в нашем списке квадратичных вычетов все вычеты попарно несравнимы. Действительно, если, например, $a \equiv k^2 \equiv l^2(p)$ и $1 \le k < l \le \frac{p-1}{2}$, то сравнение $x^2 \equiv a(p)$ имело бы четыре решения: $\pm k$, $\pm l$, что невозможно. Таким образом, количество ненулевых квадратичных вычетов равно $\frac{p-1}{2}$. Следовательно, количество квадратичных невычетов также равно $\frac{p-1}{2}$.

8.1.1. Символ Лежандра

Пусть g - примитивный элемент поля $GF(p)$. Тогда a - квадратичный вычет в том и только том случае, когда в представлении $a = g^j(p)$ число j - четно.

Действительно, если $x^2 \equiv a(p)$, то дискретное логарифмирование дает разрешимое сравнение вида $2y = j(p-1)$, где модуль четен.

Существуют алгоритмы для определения, является ли данное число квадратичным вычетом по простому модулю или нет. Один из алгоритмов связан с вычислением значения т.н. символа Лежандра, который для нечетного простого p определяется следующим образом.

$$\left(\frac{a}{p}\right) = \begin{cases} 0, a \equiv 0 \bmod p \\ 1, \exists x: \; x^2 \equiv a \bmod p \; a \neq 0 \bmod p \\ -1, \neg\exists x: \; x^2 \equiv a \bmod p \; a \neq 0 \bmod p \end{cases}$$

Значение $\left(\frac{a}{p}\right)$ называется квадратичным характером числа a по модулю p.

Основные свойства символа Лежандра.

$$a_1 = a(p) \Rightarrow \left(\frac{a_1}{p}\right) = \left(\frac{a}{p}\right);$$

Критерий Эйлера: $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \bmod p$;

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right);$$

$$(a, p) = 1 \Rightarrow \left(\frac{a^2 b}{p}\right) = \left(\frac{b}{p}\right);$$

$$\left(\frac{1}{p}\right) = 1, \ \left(\frac{-1}{p}\right) = (-1)^{(p-1)/2};$$

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}.$$

Кроме того, имеет место квадратичный закон взаимности Гаусса: для любых простых нечетных чисел p и q выполняется равенство $\left(\frac{p}{q}\right) = (-1)^{\frac{(p-1)(q-1)}{4}}\left(\frac{q}{p}\right)$.

Символ Лежандра $\left(\frac{a}{p}\right)$ можно вычислить с помощью следующей последовательности действий:

1) если $a < 0$, то выделяем сомножитель $\left(\frac{-1}{p}\right)$;

2) приводим a по модулю p;

3) раскладываем a в произведение степеней простых чисел, используя мультипликативность символа Лежандра: $\left(\frac{a}{p}\right)=\left(\frac{p_1}{p}\right)^{a_1}\cdots\left(\frac{p_k}{p}\right)^{a_t}$, затем удаляем сомножители, являющиеся (ненулевыми) квадратами;

4) выделяем двойки, например, если $p_1=2$, вычисляем $\left(\frac{2}{p}\right)=(-1)^{\frac{p^2-1}{8}}$;

5) для каждого нечетного сомножителя p_i применяем квадратичный закон взаимности (уменьшаем величины участвующих в вычислениях чисел);

6) при необходимости переходим к п.1.

Пример.

$$\left(\frac{126}{53}\right)=\left(\frac{20}{53}\right)=\left(\frac{2}{53}\right)^2\left(\frac{5}{53}\right)=(-1)^{26\cdot 2}\left(\frac{53}{5}\right)=\left(\frac{-2}{5}\right)=(-1)^2(-1)^3=-1$$

Вычисление символа Лежандра состоит в использовании его свойств для снижения величин участвующих в вычислениях чисел и достаточно удобно при работе вручную. При использовании ЭВМ обычно применяется критерий Эйлера.

Для определения квадратичного характера числа по составному модулю n с помощью символа Лежандра, необходимо знать простые делители n. Для больших чисел это нереально.

Существует алгоритм, вычисляющий т.н. называемый символ Якоби, позволяющий, по крайней мере, решить вопрос, является ли число x квадратичным невычетом по заданному нечетному модулю без его факторизации.

8.1.2. Символ Якоби

Пусть n нечетно и имеет следующее каноническое разложение $n=\prod_{i=1}^{k}p_i^{a_i}$.

Символ Якоби числа x по модулю n, при $(x,n)=1$, определяется как произведение значений символов Лежандра $\left(\frac{x}{n}\right)=\left(\frac{x}{p_1}\right)^{a_1}\dots\left(\frac{x}{p_k}\right)^{a_t}$. Он обладает практически всеми теми же свойствами, что и символ Лежандра, но по значению символа Якоби, равному единице, нельзя утверждать, что соответствующий вычет – квадратичный.

Для квадратичного вычета, тем не менее, символ Якоби равен единице. Следовательно, если $\left(\frac{x}{n}\right)=-1$, то x - квадратичный невычет по модулю n.

Пусть x, x_1, x_2 - целые, n_1, n_2 - нечетные числа, большие единицы.

Свойства символа Якоби.

$$x_1 = x_2 (n) \Rightarrow \left(\frac{x_1}{n}\right)=\left(\frac{x_2}{n}\right);$$

$$\left(\frac{x_1 x_2}{n}\right)=\left(\frac{x_1}{n}\right)\left(\frac{x_2}{n}\right);$$

$$(x_2,n)=1 \Rightarrow \left(\frac{{x_2}^2 x_1}{n}\right)=\left(\frac{x_1}{n}\right);$$

$$\left(\frac{1}{n}\right)=1,\ \left(\frac{-1}{n}\right)=(-1)^{(n-1)/2};$$

$$\left(\frac{x}{n_1 n_2}\right)=\left(\frac{x}{n_1}\right)\left(\frac{x}{n_2}\right);$$

$$\left(\frac{2}{n}\right)=(-1)^{\frac{n^2-1}{8}}.$$

Кроме того, имеет место квадратичный закон взаимности Гаусса: для любых нечетных чисел $m>1$ и $n>1$ выполняется равенство $\left(\frac{m}{n}\right)=(-1)^{\frac{(m-1)(n-1)}{4}}\left(\frac{n}{m}\right)$.

Вычисление символа Якоби состоит в использовании его свойств для снижения величин участвующих в вычислениях чисел.

Пример. Вычислим символ Якоби $\left(\frac{12}{35}\right)$.

$$\left(\frac{12}{35}\right)=\left(\frac{2^2\cdot 3}{35}\right)=\left(\frac{35}{3}\right)(-1)^{\frac{(35-1)(3-1)}{4}}=-\left(\frac{2}{3}\right)=-(-1)^{\frac{3\cdot 3-1}{8}}=1$$

8.2. Алгоритм нахождения квадратного корня в простом поле

Данный алгоритм предназначен для решения относительно y сравнения вида $x=y^2(p)$ по простому модулю $p>2$ [8,15].

Перед тем как приступить к вычислениям, необходимо убедиться в разрешимости сравнения, т.е. в том, что $\left(\frac{x}{p}\right)=1$.

Далее. Алгоритм разбивается на 3 случая, в зависимости от представления p в виде $p=4k+3$, $p=8k+5$, $p=8k+1$ (можно убедиться, что любое нечетное простое число представляется одним из указанных способов).

В алгоритме существенно используется критерий Эйлера, который для разрешимого сравнения дает: $\left(\frac{x}{p}\right)\equiv x^{(p-1)/2}\equiv 1 \bmod p$.

Случай $p=4k+3$. Имеем $1=\left(\frac{x}{p}\right)\equiv x^{(p-1)/2}\equiv x^{2k+1} \bmod p$. Умножим на x левую и правую часть сравнения, получим: $x\equiv x^{2k+2} \bmod p$. Поскольку показатель справа четный, то можно непосредственно извлечь квадратный корень из правой части, следовательно, одно из решений $y\equiv x^{k+1} \bmod p$. Поскольку решений не может быть более двух, то окончательный ответ: $y\equiv \pm x^{k+1} \bmod p$.

Случай $p = 8k + 5$. Поскольку $1 = \left(\frac{x}{p}\right) \equiv x^{(p-1)/2} \equiv x^{4k+2} \bmod p$, то $x^{2k+1} \equiv \pm 1 \bmod p$.

Таким образом, верно одно из двух соотношений $x^{2k+2} \equiv \pm x \bmod p$. Поскольку x и k известны, то с помощью вычислений можно проверить, какое из соотношений выполняется. В итоге, возможны следующие два подслучая.

Если верно $x^{2k+2} \equiv x \bmod p$, то, очевидно, $y \equiv \pm x^{k+1} \bmod p$.

Иначе, $x^{2k+2} \equiv - x \bmod p$.

Воспользуемся тем, что если обе части последнего сравнения умножить на число в известной четной степени, то квадратный корень из его левой части легко записать явно. Мы подберем указанный множитель так, чтобы, кроме того, изменился знак у правой части сравнения.

Таким множителем может быть число $2^{4k+2} = 2^{(p-1)/2} = \left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = -1(p)$, поскольку $p^2 - 1 = (8k)^2 + 80k + 24$.

Следовательно, $y \equiv \pm 2^{2k+1} \cdot x^{k+1} \bmod p$

Последний случай, $p = 8k + 1$, наиболее сложный. Прежде всего, для работы алгоритма необходимо наличие (любого) квадратичного невычета по модулю p.

Чтобы его найти, приходится выбирать наугад число, скажем *a*, и проверять соотношение $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \equiv -1 \bmod p$. Предположим, некоторый квадратичный невычет *a* нам известен.

Уточним представление числа p: $p = 8k + 1 = 2^l h + 1$, где h - нечетно и, очевидно, $l \geq 3$.

Основная идея алгоритма – построить соотношение вида $x^h \cdot a^{2m} = 1 \bmod p$.

В случае успеха достаточно умножить обе части сравнения на x и извлечь корень из обеих частей (учитывая, что число $h+1$ четно).

Исходя из сравнения $x^{p-1} = 1(p)$, мы будем строить соотношения, в которых показатель при x будет снижаться вдвое, пока не станет равным h.

Деление показателей на двойки соответствует последовательному извлечению квадратных корней, в нашем случае – квадратных корней из единицы. Будем снижать показатели так, чтобы правая часть некоторого сравнения все время была равна единице.

На каждом шаге может получиться лишь один из корней: 1 или (-1). При этом у нас будет достаточно данных, чтобы выяснить, какой случай реально имеет место. Изменять знак у (-1) мы будем с помощью умножения частей сравнения на степени числа a, причем так, чтобы показатель степени в произведении таких дополнительных множителей всегда оставался четным.

Перейдем к алгоритму.

Очевидно, $1 = x^{p-1} = x^{2^l h}(p)$. Более того, т.к. x квадратичный вычет, то $\left(\frac{x}{p}\right) \equiv x^{(p-1)/2} \equiv 1(p)$ и мы можем снизить показатель в два раза: $x^{2^{l-1} h} = 1(p)$.

Если мы второй раз разделим показатель на два, то нам известно лишь, что в правой части получится ± 1. Однако все необходимые числа даны и мы можем выбрать истинный вариант, проверяя варианты сравнения непосредственно.

Если в правой части получается 1, то двигаемся дальше. Заметим, что (-1) в правой части может получиться не ранее чем на втором шаге, т.е. при показателе $2^{l-2} h$ и ниже. Предположим, что в правой части при указанном показателе получилась (-1): $x^{2^{l-2} h} = -1(p)$.

Поскольку a - квадратичный невычет, то $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \equiv a^{2^{l-1}h} \equiv -1(p)$ и, кроме того, показатель при a больше показателя при x. (Последнее означает, что когда показатель при x станет равным h, показатель при a все еще будет четным).

Умножим обе части сравнения $x^{2^{l-2}h} = -1(p)$ на $a^{2^{l-1}h} \equiv -1(p)$, получим $x^{2^{l-2}h} a^{2^{l-1}h} = 1(p)$.

Из левой части сравнения легко выражается квадратный корень, т.е. $x^{2^{l-3}h} a^{2^{l-2}h} = \pm 1(p)$.

Как и ранее, легко уточнить, какой из вариантов имеет место на самом деле, при необходимости умножить сравнение на $a^{2^{l-1}h} \equiv -1(p)$ и т.д., пока показатель при x не окажется равным h. После последнего шага показатель при a будет известным и четным, как сумма четных чисел.

В итоге, получим сравнение вида $x^h \cdot a^{2m} = 1 \bmod p$, откуда, умножив обе части на x, получим: $y = \pm x^{(h+1)/2} a^m \bmod p$.

Пример. Решить сравнение $y^2 = 8(41)$.

Выясним, прежде всего, является ли сравнение разрешимым. Это действительно так, поскольку $\left(\frac{8}{41}\right) = \left(\frac{2}{41}\right)^3 = \left(\frac{2}{41}\right) = (-1)^{\frac{41 \cdot 41 - 1}{8}} = (-1)^{5 \cdot 42} = 1$.

Далее выясним, какой из трех случаев представления p имеет место. Очевидно, имеет место случай $p = 8k + 1$. Записав $p = 2^3 \cdot 5 + 1$, получим $h = 5$, $l = 3$.

Найдем квадратичный невычет по модулю 41. Можно выбрать $a=3$, поскольку $\left(\frac{3}{41}\right)=3^{\frac{41-1}{2}}=3^4=-1(41)$. Приступим к извлечению корня, учитывая, что $3^{20}=-1(41)$.

По теореме Эйлера, имеем $8^{p-1}=8^{8\cdot5}=1(41)$. Можно извлечь квадратный корень, разделив показатель на два. При этом значение корня равно 1, т.к. 8 - квадратичный вычет и, следовательно, $8^{4\cdot5}=8^{\frac{p-1}{2}}=1(41)$. Показатель можно разделить на два снова, но необходимо выяснить, какому числу (1 или –1) равно значение квадратного корня. Итак, $8^{2\cdot5}=\pm1(41)$. Поскольку $8^5=9(41)$, то $8^{2\cdot5}=81=-1(41)$.

Нам необходимо добиться значения 1 в правой части сравнения. Умножим обе части на 3^{20}, получим $8^{2\cdot5}3^{20}=1(41)$. Как и следовало ожидать, показатель при тройке четный.

Нам остался один шаг, чтобы снизить показатель при восьмерке до $h=5$. Делим показатели на два и вычисляем значение левой части: $8^5 3^{10}=\pm1(41)$, $8^5 3^8 3^2=9\left(3^4\right)^2 3^2=9(-1)^2 9=-1(41)$, т.е. $8^5 3^{10}=-1(41)$.

Нам вновь необходимо умножить обе части сравнения на 3^{20}, что дает $8^5 3^{10} 3^{20}=1(41)$.

Показатель при восьмерке равен $h=5$. Переходим к выписке результата. Умножив обе части сравнения на 8, получим $8^6 3^{30}=8(41)$, что позволяет записать квадратный корень из 8 в виде $y=\pm8^3 3^{15}=\pm7(41)$. Проверка: $(\pm7)^2=49=8(41)$.

Задачу извлечения квадратного корня в кольце вычетов для случая составного модуля n, факторизация которого известна, можно решить с помощью

китайской теоремы об остатках. Для этого достаточно знать решения по модулям делителей числа n вида p^a, которые можно находить, исходя из решений по модулю p. Соответствующий метод для более общего случая полиномиальных сравнений описан, например, в [14].

Глава 9.
КРИПТОСИСТЕМА RSA, ПРОТОКОЛ ДИФФИ-ХЭЛЛМАНА И ЦИФРОВАЯ ПОДПИСЬ ЭЛЬ-ГАМАЛЯ

Основное в этой главе…

- **Построение криптосистемы RSA. Идея цифровой подписи…….....116**
- **Смешанные криптосистемы. Протокол Диффи-Хэллмана ключевого обмена……..……....119**
- **Цифровая подпись Эль-Гамаля……121**
- **Ослабление подписи Эль-Гамаля вследствие некорректной реализации схемы……...123**

Широко известная криптографическая система RSA, предложенная в 1978 году, является асимметричной криптосистемой, основанной на односторонней функции с лазейкой, в качестве которой выбрана степенная функция в кольце вычетов целых чисел по составному (двупростому) модулю $n = pq$. На основе криптосистемы RSA можно построить механизм цифровой подписи [42]. Стойкость системы основана на сложности задачи факторизации больших двупростых чисел.

Криптосистема RSA на каждом такте шифрования преобразует двоичный блок открытого текста m длины $size(n) = \lfloor \log_2 n \rfloor + 1$, рассматриваемый как целое число, с помощью возведения в степень по модулю n: $c = m^e(n)$. Показатель степени и модуль являются элементами открытого ключа.

Лазейка обеспечивается за счет секретного ключа d, построенного таким образом, что для всех m выполняется соотношение $c^d = m^{ed} = m(n)$.

9.1. Построение криптосистемы RSA. Идея цифровой подписи

Построение криптосистемы обеспечивает получатель сообщений.

Сначала случайным образом выбираются два различных больших простых числа p и q. Выбранные простые числа должны удовлетворять некоторым дополнительным условиям.

Затем вычисляется модуль $n = pq$, функция Эйлера от модуля $\varphi(n) = (p-1)(q-1)$, а также выбирается случайное число e, взаимно простое с $\varphi(n)$.

Секретный ключ строится с помощью расширенного алгоритма Эвклида, как число d, удовлетворяющее сравнению $de \equiv 1(\varphi(n))$. После этого все данные, кроме e, d, n, включая данные промежуточных вычислений, уничтожаются. Пара e, n объявляется в качестве открытого ключа.

Расшифрование обеспечивается тем, что для $(m,n)=1$ из теоремы Эйлера следует $c^d = m^{1+k\varphi} \equiv m(n)$ и, кроме того, для других значений *m* соотношение $m^{ed} \equiv m(n)$ также имеет место, вследствие свойств модуля вида $n = pq$.

Свойства параметров криптосистемы RSA позволяют показать принципиальную возможность построения схемы электронной цифровой подписи.

Цифровой подписью (ЦП) называется результат специального криптографического преобразования, осуществленного над электронным документом его владельцем.

Цель преобразования – доказать неоспоримость текста документа и факта преобразования данных конкретным лицом.

Основной метод – проверка факта использования секретного параметра (ключа) подписи, без знания ключа проверяющим.

Подпись представляет собой блок данных. Подписанное сообщение представляет собой исходное сообщение, передаваемое совместно с ЦП.

Владелец секретного ключа d криптосистемы RSA мог бы в качестве подписанного сообщения представить пару $(m,m^d)=(m,c)$. Действительно, преобразование m^d может осуществить только он. Поскольку m имеется в сообщении в исходном виде, любой абонент в состоянии проверить соотношение $m = c^e$, которое будет выполняться лишь в том случае, когда действительно $c = m^d$.

Однако подобный подход не обеспечивает стойкость подписи при передаче случайных данных. Действительно, выберем число a и построим сообщение $m = a^e(n)$. Тогда, очевидно, $(m,m^d)=(m,a)$, т.е. подпись построена без знания секретного ключа.

По этой причине лучше вместо (m, m^d) использовать пару $(m, h(m)^d)$, где $h(m)$ - т.н. хэш-функция сообщения *m*: несекретное, заранее обусловленное преобразование. Значение $z = h(m)$ называется хэш-кодом сообщения *m*. Проверка подписи (m, c) начинается с вычисления $h(m)$, затем результат сравнивается с $c^e(m)$.

Реальные хэш-функции представляют собой сложные алгоритмы, рекомендованные в соответствующих стандартах [17,18]. Методика их обоснования выходит за рамки данной книги.

Определение хэш-функции. Хэш-функция $z = h(m)$ - преобразование битовой строки произвольной длины в битовую строку (блок) фиксированной длины (обычно, 160-512 битов), обладающее следующими свойствами.

1. Восстановление m по z, исходя из соотношения $z = h(m)$, вычислительно нереализуемо.

2. Исходя из m и z, вычислительно нереализуемо определение второго прообраза для z, т.е. такого сообщения $m_1 \neq m$, что $z = h(m) = h(m_1)$.

На практике, как правило, используются хэш-функции, удовлетворяющие более жесткому, чем последнее, условию: требуется вычислительная нереализуемость нахождения произвольной коллизии, т.е. пары сообщений x, y, таких, что $h(x) = h(y)$.

Рассмотрим пример построения (учебной) криптосистемы RSA.

1. $p = 3$, $q = 11$, $n = 33$, $\varphi(n) = 20$.

2. $e = 7$ $(e, \varphi(n)) = 1$.

3. $d = 3$ $ed = 1(\varphi)$.

Зашифруем сообщение из трех блоков: 3,1,2.

$$RSA(3) = 3^7 = 2187 = 9(33),\ RSA(1) = 1^7 = 1(33),$$

$$RSA(2) = 2^7 = 128 = 29(33).$$

Для расшифрования возведем блоки в степень $d = 3$ по модулю 33:

$$9^3 = 729 = 3(33)\ ,\ 1^3 = 1(33)\ ,\ 29^3 = 24389 = 2(33).$$

Секретный ключ для учебной системы легко найти перебором. На практике это невыполнимо, т.к. реальный размер модуля (длина битового представления) $size(n)$ находится в диапазоне от 512 до 4096 битов.

9.2. Смешанные криптосистемы. Протокол Диффи-Хэллмана ключевого обмена

В настоящее время в системах связи общего назначения широко распространены т.н. смешанные криптосистемы. В таких системах конфиденциальность сообщений обеспечивается за счет шифрования с помощью симметричной криптосистемы, рассылка ключей для которой осуществляется с помощью асимметричных криптоалгоритмов.

Здесь возникает новый тип задач, связанный с т.н. недоверием корреспондентов друг к другу.

Имеется большое количество стандартизованных протоколов безопасного ключевого обмена, применимых в различных ситуациях. Обычно в ходе таких протоколов обеспечивается аутентификация абонентов и построение общего секретного набора данных для дальнейшей генерации ключей.

Наиболее ранний протокол обмена ключами при взаимном недоверии участников обмена предложен Диффи и Хэллманом [29]. В этом протоколе используется показательная функция в простом поле Галуа *GF(p)*, обратной к которой является дискретный логарифм.

Пусть абонент А является инициатором обмена. Он намерен выработать общий с абонентом В секретный ключ для симметричной криптосистемы.

При этом обоим корреспондентам известен первообразный элемент (точнее, элемент большого порядка) g поля $GF(p)$ и простое число p.

Протокол решает задачу построения общего секретного блока данных вида $g^{xy}(p)$, где x, y - случайные вычеты по модулю $p-1$.

Абонент А случайно выбирает *x*, вычисляет значение $g^{x}(p)$ и отправляет это значение абоненту В. Абонент В действует аналогично: выбирает y, вычисляет значение $g^{y}(p)$ и отправляет это значение абоненту А. Каждый из абонентов в состоянии теперь вычислить значение общего секретного блока $g^{xy}(p)$. Получить это значение, исходя из перехвата, оказывается, невозможно, вследствие свойств дискретного логарифма, а подходов, неэквивалентных дискретному логарифмированию, не найдено.

Пример системы экспоненциального ключевого обмена Диффи-Хэллмана.

1. $p=13$, $g=7$, поскольку $g^{(p-1)/2}=117649=-1(p)$, $g^{(p-1)/3}=9\neq 1(p)$.

2. Абонент А генерирует псевдослучайное число, например, $x=8$ и передает абоненту В значение $g^{x}=7^{8}=3(13)$.

3. В аналогично генерирует $y=5$ и отправляет А значение $g^{y}=7^{5}=11(13)$.

4. А вычисляет общий секретный параметр $k=\left(g^{y}\right)^{x}=11^{8}=9(13)$.

5. В вычисляет общий секретный параметр $k=\left(g^{x}\right)^{y}=3^{5}=9(13)$.

9.3. Цифровая подпись Эль-Гамаля

Одним из примеров использования сложности задачи дискретного логарифмирования в криптографии является т.н. схема Эль-Гамаля, которую можно использовать как для цифровой подписи, так и для шифрования [43].

9.3.1. Криптосистема Эль-Гамаля

Криптосистема Эль-Гамаля является асимметричной. Для построения пары ключей выбирается большое простое число p и два псевдослучайных числа, меньших $p-1$.

Одно из них, g, должно быть элементом большого порядка по модулю p, скажем, первообразным корнем. Второе число, x, выбирается в качестве секретного ключа. Полагается, что сообщения – вычеты по модулю p.

Открытым ключом является тройка чисел $p, g, y = g^x (p)$.

Заметим, что дискретное возведение в степень ведет себя как односторонняя функция, но не как односторонняя функция с лазейкой.

Поэтому для каждого сообщения формируются дополнительные данные, играющие роль лазейки для конкретного сеанса шифрования.

Для зашифрования сообщения m выбирается псевдослучайное число k (рандомизатор, разовый ключ) с условием НОД$(k, p-1)=1$. Рандомизаторы не должны повторяться и должны содержаться в секрете.

Затем вычисляются числа $a = g^k (p)$ - лазейка и $b = y^k m (p)$ - шифртекст. Криптограммой является пара блоков данных a, b.

Для расшифрования достаточно получить сомножитель y^k, что можно сделать с помощью секретного ключа, вычислив значение $a^x = g^{kx} (p)$.

Действительно, $y^k = g^{xk} (p)$, поэтому $m = a^{-x} b (p)$.

9.3.2. Механизм цифровой подписи Эль-Гамаля

Цифровая подпись Эль-Гамаля состоит из пары блоков r, s.

В механизме этой подписи используются те же параметры что и в криптосистеме Эль-Гамаля. Преобразование, естественно, производится лицом, обладающим секретным ключом. Кроме того, используется хэш-функция сообщения $h(m)$.

Лицо, подписывающее документ, должно для каждого подписываемого сообщения m выбрать рандомизатор k и вычислить «предподпись» $r = g^k(p)$.

Затем необходимо использовать секретный ключ как один из коэффициентов сравнения, из которого определяется блок подписи s.

Сравнение имеет вид $h = h(m) = xr + ks \bmod(p-1)$. Здесь модулем сравнения выбрано число $p-1$, поскольку обе части сравнения в проверочном соотношении будут участвовать в показателях.

Заметим, что число r может равняться $p-1$, поскольку оно определяется из сравнения по модулю p.

Подпись r, s считается действительной, если $g^h = g^{xr+ks}(p)$.

Поскольку $y = g^x(p)$ и $r = g^k(p)$, то окончательный вид проверочного соотношения следующий: $g^h = y^r r^s(p)$. Таким образом, для проверки подписи достаточно знания открытого ключа.

Очевидно, необходимость выбора рандомизатора требует применения криптографически стойких датчиков (генераторов) псевдослучайных чисел.

Выходные последовательности таких генераторов являются непредсказуемыми. Кроме того, структура выходной последовательности не позволяет определить начальное состояние и, возможно, другие параметры соответствующего датчика.

Определение и свойства криптографически стойкого псевдослучайного генератора можно найти в [2].

В Государственном Стандарте Украины ДСТУ 4145-2002 [19] для построения рандомизаторов цифровой подписи применяется датчик псевдослучайных чисел, схема которого использует криптоалгоритм ГОСТ 28147-89.

9.3.3. Ослабление подписи Эль-Гамаля вследствие некорректной реализации схемы

Схемы цифровой подписи типа Эль-Гамаля стандартизованы и широко используются на практике. Следует отметить, что малейшее отклонение от рекомендаций стандартов может привести к ослаблению подписи.

Приведем примеры. Пусть k известно, тогда из соотношения $h = xr + ks \bmod(p-1)$ находим секретный ключ x и можем подписывать сообщения вместо владельца подписи.

Далее. При повторении рандомизатора получим систему сравнений относительно x и k вида $h_1 = xr_1 + ks_1 \bmod(p-1)$, $h_2 = xr_2 + ks_2 \bmod(p-1)$, что позволяет определить ключ даже при переменной хэш-функции.

Третий пример – несоблюдение ограничения на размер параметра: $r < p$.

Допустим, что проверяющая сторона не учитывает возможность возникновения ситуации, когда $r > p$.

Наблюдая за сетью связи, можно получить подписанное сообщение m и выделить блоки некоторой истинной подписи r, s, связанные со значением хэш-кода $h = h(m)$.

Пусть необходимо подделать подпись для сообщения m_1 с хэш-кодом. $h_1 = h(m_1)$. Предположим, хэш-коды двух рассматриваемых сообщений отличаются некоторым множителем u, который можно найти из сравнения $h_1 = uh \bmod(p-1)$. (Если это сравнение не имеет решения, то перехватим в сети следующее подписанное сообщение и т.д.)

Для истинной подписи выполняется проверочное соотношение $g^h = y^r r^s (p)$. Попытаемся подобрать такую подпись r_1, s_1, чтобы аналогичное соотношение выполнялось для r_1, s_1, h_1.

Рассмотрим $g^{h_1} = g^{uh} = \left[y^r r^s\right]^u (p)$, т.е. $g^{h_1} = y^{ru} r^{su} (p)$.

Очевидно, можно принять $s_1 = su \bmod(p-1)$. Однако, если принять $r_1 = ru \bmod(p-1)$, то проверочное соотношение выполняться не будет, так как r_1 должно участвовать в основании и показателе степени одновременно, что не имеет места. Оказывается, выход из положения существует, если допускается возможность значений $r_1 > p$.

Действительно, r_1 в показателе и основании степени приводятся по различным модулям, т.е. должны одновременно выполняться сравнения $r_1 = ru \bmod(p-1)$ и $r_1 = r \bmod(p)$.

Модули сравнений взаимно просты, следовательно, по китайской теореме об остатках, существует решение $r_1 = r \bmod(p(p-1))$, однако $r_1 > p$.

Таким образом, для r_1, s_1 проверочное соотношение выполняется, т.е. подпись может быть подделана, если не контролируются ограничения на длины параметров.

9.3.4. Варианты цифровой подписи типа Эль-Гамаля

Построение различных вариантов подписей типа Эль-Гамаля можно осуществить, исходя из сравнения вида $Ak + Bx + C = 0 \bmod q$, связывающего секретный ключ x и рандомизатор k с блоками цифровой подписи r, s. Здесь $A, B, C \neq 0(q)$, где $q = p - 1$ (для краткости обозначений).

Из $Ak + Bx + C = 0 \bmod q$ следует, что $A = -k^{-1}Bx - k^{-1}C \bmod q$.

С очевидными переобозначениями, можно записать: $A = bx + c \bmod q$.

Следовательно, $g^A = \left[g^x\right]^b g^c (\bmod p)$, т.е. через g^x и другие параметры можно проверить некоторое соотношение, не раскрывая значения x.

В классической подписи Эль-Гамаля для сообщения m, $r = g^k(p)$, а блок s определяется из сравнения $Ak + Bx + C = 0 \bmod q$, при $(A, B, C) = (s, r, -h(m))$. Действительно, в этом случае, $s = k^{-1}(h - xr) \bmod(q)$, т.е. $h = xr + ks \bmod(p-1)$.

Рассмотрим пример на построение цифровой подписи Эль-Гамаля.

Выберем $p = 17$. Поскольку $3^{(p-1)/2} = 6561 = -1(p)$, то $g = 3$ - первообразный элемент.

Примем $x = 5$. Тогда открытый ключ $y = g^x(p) = 3^5 = 5(17)$. Пусть $h(m) = 9$.

Выберем рандомизатор $k = 7$. Вычислим первую часть подписи $r = g^k(p) = 3^7 = 11(17)$.

Затем – вторую часть подписи, по формуле $s = k^{-1}(h - xr)\mathrm{mod}(q)$, где $k^{-1} = 7(16)$. Получим $s = 14(16)$. Подпись равна $11{,}14$.

Проверка подписи. Вычисляем хэш-код сообщения h (допустим, что $h = 9$) и проверяем соотношение $g^h = y^r r^s(p)$.

Получим: $g^h = 3^9 = 14(17)$, $y^r = 5^{11} = 11(17)$, $r^s = 11^{14} = 9(17)$, $14 = 99(17)$, то есть в данном случае подпись верна.

Глава 10.
АЛГОРИТМ СИЛЬВЕРА-ПОЛЛИГА-ХЭЛЛМАНА

Основное в этой главе…

- **Алгоритм дискретного логарифмирования ……..……...130**
- **Логарифмирование в группе единиц кольца вычетов по модулю p^r……..134**
- **Фальсификация подписи Эль-Гамаля в специальном случае выбора первообразного элемента и характеристики поля……..……..137**

Известно, что дискретная экспонента, т.е. функция вида $f(x)=a^x \bmod p$, где p - простое, ведет себя при больших значениях x и $ord_p a$ как односторонняя. Соответствующая обратная функция (дискретный логарифм) вычислительно нереализуема.

Аналогичная ситуация может возникнуть и в общем случае коммутативной конечной группы.

Групповая операция в мультипликативной записи называется умножением. Возведение в степень x определяется как произведение элемента $b \in G$ на себя x раз.

Пусть G конечная коммутативная группа, $b \in G$, и $y = b^x$. В этом случае x называется дискретным логарифмом элемента y по основанию b.

Задача дискретного логарифмирования является алгоритмической проблемой, на которой основана стойкость многих практически используемых криптоалгоритмов. Поэтому методы дискретного логарифмирования, применимые в частных случаях, важны с точки зрения выбора параметров криптосистем.

Алгоритм Сильвера-Поллига-Хэллмана позволяет решить задачу дискретного логарифмирования в конечном поле F_q, состоящем из $q = P^\alpha$ элементов, при условии, что число $q-1$ - «гладкое», т.е. все простые делители числа $q-1$ невелики [15,44,45].

Последнее означает, что имеющиеся ресурсы памяти и вычислительной мощности позволяют провести необходимые вычисления в реальное время.

Заметим, что вычисление больших степеней элементов в конечном поле можно выполнить относительно быстро.

10.1. Обозначения и постановка задачи

Обозначим множество ненулевых элементов поля F_q через F_q^*.

Задача дискретного логарифмирования, которую мы рассмотрим, формулируется следующим образом.

Пусть задано поле F_q и первообразный элемент поля b.

При заданном $y \in F_q^*$, найти целое x: $0 \le x < q-1$, такое, что выполняется равенство $y = b^x$.

Пример. Пусть $q = 17$, $b = 3$. Элемент b - первообразный корень поля вычетов по модулю 17. Очевидно, если q является простым числом, то уравнение $y = b^x$ в поле является сравнением вида $y \equiv b^x \bmod q$.

Число $q-1$ является гладким, т.к. $q-1 = 2^4$, т.е. его единственный простой делитель $p = 2$ мал.

Пусть каноническое разложение $q-1$ имеет вид: $q-1 = \prod_p p^a$. Обозначим количество различных простых в этом разложении через n.

Пусть *m*- максимальное простое число в разложении $q-1$. Для удобства записи введем также дополнительное обозначение $u = q-1$.

Мы постоянно будем пользоваться тем фактом, что любой ненулевой элемент поля $g \in F_q^*$ удовлетворяет условию $g^{q-1} = 1$, так что $g^u = 1$.

10.2. Построение корней из единицы в поле

Построим таблицу R размером $n \times m$, строкам которой присвоим в качестве меток простые числа p из разложения $q-1$. Заполним сначала ее нулями.

В строку с меткой p поместим элементы поля $r_{p,j} = b^{ju/p}$, $(j = 0, \ldots, p-1)$.

Запись показателя в виде дроби корректна, поскольку один из сомножителей числителя, а именно u, делится на p, т.е. дробь является обычным целым числом.

Очевидно, $r_{p,j}^{p} = b^{ju} = 1$, следовательно, элементы $r_{p,j}$ - корни степени p из единицы. Аналогичным образом заполним все строки таблицы R.

Дальнейшая работа сводится к вычислениям, в результате которых появляются элементы, являющиеся корнями степени p из единицы в поле.

Для каждого такого элемента будет необходимо определить его позицию j (номер колонки) в строке таблицы R с меткой p.

Поскольку в каждой строке элементы различны, то для данного числа соответствующую ему позицию мы определим однозначно.

Для этого, конечно, мы должны быстро просмотреть строку таблицы R, что возможно, поскольку число $q-1$ - гладкое.

10.3. Алгоритм дискретного логарифмирования

Предположим, что x представлен в p-ичной системе счисления. Тогда его вычет по модулю p^a имеет вид $x = x_0 + x_1 p + \ldots + x_{a-1} p^{a-1} \bmod p^a$, $0 \le x_i \le p-1$.

Обозначим $y_0 = y$. Для определения чисел x_k, $k = 0, \ldots, a-1$, предложим следующую процедуру, которую обсудим позже.

Прежде всего, определяем x_0 как позицию $y_0^{u/p}$ в строке с номером p таблицы R.

Для $k>0$ коэффициент x_k определяем как позицию элемента $y_k^{u/p^{k+1}}$ в строке с меткой p таблицы R. При этом, $y_k = y_0/b^{h(k)}$, где $h(k) = x_0 + x_1 p + \ldots + x_{k-1} p^{k-1}$.

Повторив процедуру для каждого p, делящего $q-1$, мы получаем значения $x \bmod p^a$, а затем с помощью китайской теоремы об остатках восстанавливаем $x \bmod (q-1)$.

Обоснуем процедуру определения x_k.

Вычислим $y_0^{u/p}$. Очевидно, $y_0^{u/p}$ - корень степени p из единицы, причем $y_0^{u/p} = y^{u/p} = b^{xu/p} = b^{x_0 u/p + (x-x_0)u/p}$, где $x - x_0 = x_1 p + \ldots + x_{a-1} p^{a-1}$.

Кроме того, число $x_0 u/p$ является целым, т.к. u делится на p.

В выражении $(x-x_0)u/p$ оба сомножителя делятся на p. Разделив на p первый сомножитель, получаем, что $(x-x_0)u/p = ku$, т.е. $y_0^{u/p} = b^{x_0 u/p}$. Сравнив с обозначением $r_{p,j} = b^{ju/p}$, получаем, что $y_0^{u/p} = r_{p,j}$ при $j = x_0$.

Это позволяет определить x_0, как позицию $y_0^{u/p}$ в строке таблицы R с меткой p.

Уничтожим теперь x_0 в показателе степени b^x, разделив y_0 на b^{x_0}. Обозначим результат через y_1: $y_1 = y_0/b^{x_0 p^0}$ и вычислим $y_1^{u/p^2} = b^{u(x-x_0)/p^2}$.

По условию, число u/p^2 - целое (при $a \geq 2$). В выражении $x - x_0$ все члены, кроме $x_1 p$, делятся на p^2. Поэтому в показателе $u(x-x_0)/p^2$ все члены, кроме $u x_1 p/p^2$, кратны u и на значение $b^{u(x-x_0)/p^2}$ не влияют.

Поскольку u делится на p, то число $ux_1 p/p^2 = x_1 u/p$ - также целое, откуда $y_1^{u/p^2} = b^{x_1 u/p} = r_{p,j}$ при $j = x_1$. Таким образом, x_1 равен позиции y_1^{u/p^2} в строке с меткой p таблицы R.

Для определения x_2 уничтожим x_1 в показателе степени b^{x-x_0}, разделив y_1 на $b^{x_1 p^1}$. Получим: $y_2 = y_0 / b^{x_0 p^0 + x_1 p^1}$, т.е. $y_2 = b^d$, где $d = x_2 p^2 + \ldots + x_{a-1} p^{a-1}$.

Вычисляем $y_2^{u/p^3} = b^{x_2 u/p} = r_{p,j}$, при $j = x_2$, что позволяет определить x_2 по таблице R и т.д., пока не определим x_{a-1}.

10.3.1. Пример вычисления дискретного логарифма

В поле F_{37}, при $b = 2$, найти дискретный логарифм элемента 28.

Решение. Задача сводится к решению в поле F_{37} уравнения $28 = 2^x$.

Число $q = 37$ является первой степенью простого числа, поэтому операции в поле совпадают с операциями в поле вычетов по модулю 37, в частности, деление есть умножение на обратный элемент.

Далее, $u = q - 1 = 36 = 2^2 \cdot 3^2$, следовательно, имеем два простых делителя: 2 и 3.

Составим таблицу R. Начнем со строки с меткой $p = 2$. Вычислим $r_{2,j} = b^{j(q-1)/2}$ для $j = 0,1$: $r_{2,0} = 1$, $r_{2,1} = 2^{(q-1)/2} \equiv -1(37)$.

Элементами строки с меткой 3 являются числа: $r_{3,j} = b^{j(q-1)/3}$, $j = 0,1,2$, т.е. $r_{3,0} = 1$, $r_{3,1} = 2^{36/3} \equiv 26(37)$, $r_{3,2} = 2^{2 \cdot 36/3} \equiv 2^{24} \equiv 10(37)$. Таблица R имеет следующий вид.

Таблица 4. Корни из единицы степеней 2 и 3

R=

	0	1	2
2	1	-1	0
3	1	26	10

Найдем вычет $x = x_0 + x_1 p + \ldots + x_{a-1} p^{a-1} \bmod p^a$, при $p = 2$, $a = 2$. Число шагов равно $a = 2$. Итак, необходимо определить x_0, x_1. Найдем x_0. Вычислим $y_0^{u/p} = 28^{18} \equiv 1(37)$. Позиция единицы в строке 2 таблицы R равна 0, следовательно, $x_0 = 0$.

Далее будем вычислять элементы y_k и возводить их в степени u/p^{k+1}.

Вычислим y_1, уничтожив член с x_0 в показателе числа b^x: $y_1 = y_0 / b^{x_0 p^0}$. Поскольку $x_0 = 0$, то $y_1 = y_0$. Возводим y_1 в степень u/p^2, где $p^2 = 4$: $y_1^{u/4} = 28^{36/4} = -1(37)$.

Позиция числа (-1) в строке 2 таблицы R равна 1, следовательно, $x_1 = 1$. Итак $x = x_0 + x_1 p = 2 \bmod 4$.

Найдем вычет $x \bmod p^a$, при $p = 3$, $a = 2$. Число шагов равно $a = 2$. $y_0^{u/p} = 28^{12} \equiv 26(37)$. Позиция числа 26 в строке 3 таблицы R равна 1, следовательно, $x_0 = 1$, поэтому $y_1 = y_0 / b^{x_0 p^0} = 14$.

Возводим y_1 в степень u/p^2, где $p^2 = 9$: $y_1^{u/9} = 14^{36/9} = 10(37)$, следовательно $x_1 = 2$. Итак, $x = 7 \bmod 9$.

Решаем систему сравнений $x = 2 \bmod 4$, $x = 7 \bmod 9$: $9^{-1}(4) = 1$, $4^{-1}(9) = 7$, $x \equiv 2 \cdot 9 \cdot (9^{-1} \bmod 4) + 7 \cdot 4 \cdot (4^{-1} \bmod 9) = 214$, т.е. $x \equiv 34 \bmod 36$.

10.3.2. Логарифмирование в группе единиц кольца вычетов по модулю p^r.

Существование частных методов дискретного логарифмирования в конечном поле приводит к необходимости построения больших простых чисел со специальными свойствами. В качестве подхода, позволяющего снизить величины генерируемых простых чисел, можно, в принципе, рассмотреть использование в протоколе Диффи-Хэллмана операцию приведения по модулю p^r, $p > 2$, $r > 1$. В этом случае легко обеспечить большие размеры и разнообразие генерируемых ключей.

Известно [14,гл.6], что в кольце Z/p^rZ вычетов по модулю $m = p^r$ существует первообразный элемент γ, степени которого представляют все вычеты, взаимно простые с модулем. Эти вычеты образуют в Z/p^rZ мультипликативную группу $U(p^r)$ из $\varphi(p^r)$ элементов (т.н. группу единиц).

Можно показать [20,гл.4], что если γ_p - первообразный корень в поле $GF(p)$, то одно из чисел $\gamma = \gamma_p + kp$, где $k \in \{0,1\}$, удовлетворяет условию $(\gamma_p + kp)^{p-1} \neq 1 (\bmod p^2)$ и является первообразным корнем по любому модулю p^α, $\alpha > 1$. Заметим, что пары чисел *a*, *p*, для которых выполняется соотношение $a^{p-1} = 1 \bmod p^2$, встречаются редко. Поэтому будем считать, что $\gamma = \gamma_p$.

Таким образом, любой из $\varphi(p^r) = p^{r-1}(p-1)$ вычетов $b \in U(p^r)$ можно представить в виде $b = \gamma^x \bmod p^r$.

Соответственно, задача дискретного логарифмирования в группе единиц кольца Z/p^rZ состоит в определении вычета x по $\bmod\varphi(p^r)$, исходя из b и γ.

К сожалению, использование модуля p^r не дает преимуществ, поскольку при наличии эффективного алгоритма логарифмирования в простом поле $GF(p)$ логарифмирование в группе единиц кольца вычетов по модулю p^r, $r>1$, практически всегда можно осуществить, воспользовавшись свойствами так называемого обобщенного отношения Ферма $L_m(a)$ [21,46].

Областью определения отображения $L_m(a)$ является группа $U(m)$ вычетов по модулю m, взаимно простых с модулем. По теореме Эйлера, существует λ: $a^{\varphi(m)}-1=\lambda m$. Значение $L_m(a)$ определяется как вычет числа λ по модулю m: $L_m(a)=\dfrac{a^{\varphi(m)}-1}{m}(\bmod m)$.

Легко убедиться, что отношение Ферма обладает следующими замечательными свойствами.

$$L_m(ab)=L_m(a)+L_m(b)(\bmod m),$$

$$L_m(a+mc)=L_m(a)+\varphi(m)ca^{-1}(\bmod m),$$

где $a,b\in U(m)$, $c\in Z/mZ$.

Далее будем считать, что $m=p^r$, $r>1$. Заметим, что в этом случае, $L_m(a+mc)=L_m(a)+(p^r-p^{r-1})ca^{-1}(\bmod m)=L_m(a)(\bmod p^{r-1})$. Таким образом, если $a\equiv b(\bmod m)$, то $L_m(a)\equiv L_m(b)(\bmod p^{r-1})$.

Если $L_m(\gamma)=0(\bmod p^{r-1})$, то из определения $L_m(\gamma)$ следует, что $\gamma^{\varphi(m)}=1(\bmod p^{2r-1})$.

Однако, γ - первообразный корень по любому модулю p^{α}, $\alpha > 1$, поэтому $\varphi\left(p^{2r-1}\right) \le \varphi(m)$, откуда $2r-2 \le r-1$, т.е. $r \le 1$, что противоречит условию.

Если $L_m(\gamma) = 0 \left(\mathrm{mod}\, m\right)$, то $L_m(\gamma) = 0\left(\mathrm{mod}\, p^{r-1}\right)$ и $r \le 1$.

Аналогично, при $L_m(\gamma) = pt\left(\mathrm{mod}\, p^{r-1}\right)$, получаем $\gamma^{\varphi(m)} = 1\left(\mathrm{mod}\, p^{r+1}\right)$, что невозможно, т.к. $\varphi\left(p^{r+1}\right) > \varphi(m)$.

Таким образом, элемент $L_m(\gamma)$ взаимно прост с p и, следовательно, обратим по модулю p^{r-1}.

Рассмотрим задачу логарифмирования в группе единиц кольца Z/mZ, полагая, что эффективный алгоритм логарифмирования в кольце Z/pZ известен.

Из соотношения $b = \gamma^x \bmod p^r$ следует, что $b = \gamma^x \bmod p$, т.е. нам известно значение x по модулю $p-1$. Если мы найдем $x\left(\mathrm{mod}\, p^{r-1}\right)$, то значение x по модулю $\varphi(m) = p^{r-1}(p-1)$ можно вычислить по китайской теореме об остатках.

Очевидно, что значение $x\left(\mathrm{mod}\, p^{r-1}\right)$ легко определить из сравнения $L_m(b) = xL_m(\gamma)\left(\mathrm{mod}\, p^{r-1}\right)$.

Таким образом, нам необходимо вычислять значения $h = L_m(a)\left(\mathrm{mod}\, p^{r-1}\right)$.

Непосредственно вычислить $L_m(a) \bmod p^{r-1}$ трудно, т.к. величина $a^{\varphi(m)}-1$ быстро возрастает с ростом $m = p^r$.

Очевидно, однако, что $\frac{a^{\varphi(m)}-1}{p^r} = h + kp^{r-1}$, где $h < p^{r-1}$. Поэтому $a^{\varphi(m)} - 1 = hp^r + kp^{2r-1}$, причем $hp^r < p^{2r-1}$.

Следовательно, наименьший неотрицательный вычет $H = a^{\varphi(m)} - 1 \left(\bmod p^{2r-1}\right)$ совпадает с числом hp^r. Это позволяет проводить вычисления по $\bmod p^{2r-1}$ и получать h в виде $\frac{H}{p^r}$.

10.4. Фальсификация подписи Эль-Гамаля в специальном случае выбора первообразного элемента и характеристики поля

В качестве предварительного замечания рассмотрим вариант алгоритма Сильвера-Поллига-Хэллмана в ситуации, когда $F_q = GF(p)$, где число $p-1$ не является гладким, но для некоторого делителя w числа $p-1$ гладким является число $u = (p-1)/w$.

Пусть g - первообразный элемент поля $GF(p)$. Рассмотрим задачу определения z из сравнения $g^{wz} \equiv y^w \left(\bmod p\right)$. Обозначим $g^w = b$.

Поскольку существует m: $y = g^m\ (p)$, то $y^w = g^{mw} = b^m\ (p)$. Поэтому исходная задача сводится к задаче $\tilde{y} \equiv b^z \equiv b^m (p)$, которая является разрешимой.

Заметим, что порядок элемента b равен u. Следовательно, наша задача является задачей дискретного логарифмирования в мультипликативной

подгруппе поля $GF(p)$, элементы которой выражаются в виде степеней элемента b.

По условию, порядок группы u является гладким числом. Это позволяет создать таблицу R для простых делителей числа u и найти z с помощью алгоритма Сильвера-Поллига-Хэллмана.

Таким образом, если $u=(p-1)/w$ - гладкое число, то определение z из сравнения $g^{wz} \equiv y^w \pmod p$ вычислительно реализуемо.

Как следствие докажем следующий факт.

Теорема [41]. Пусть дана схема цифровой подписи Эль-Гамаля с параметрами p, g, x, $y \equiv g^x \pmod p$ и $p-1=uw$, где u - гладкое число. Пусть, кроме g, задан примитивный элемент поля β вида $\beta = cw$ и известно число t, такое, что $\beta^t = g \pmod p$.

Тогда для любого значения хэш-функции от сообщения h можно построить корректную подпись (r,s) без знания секретного ключа *x*.

Действительно, найдем z: $y^w \equiv g^{wz} \pmod p$. Для заданного значения h предложим подпись (r,s), где $r=\beta$, $s = t(h-cwz) \bmod (p-1)$. Подставим данные в проверочное соотношение для подписи, которое имеет вид $g^h = y^r r^s \pmod p$.

Имеем: $y^r r^s = y^\beta \beta^s = y^{cw} \beta^{t(h-cwz)} \pmod p$. Заменяя y^w на g^{wz}, а β^t на g, получим: $g^{cwz} g^{h-cwz} = g^h \ (p)$, т.е. подпись корректна.

Данное следствие позволяет сделать вывод о существовании частных случаев, когда подпись Эль-Гамаля может быть фальсифицирована.

10.4.1. Слабые параметры в подписи Эль-Гамаля

Пусть $p = 4k + 1$. Допустим, что параметры схемы подписи можно выбрать так, чтобы примитивный элемент g совпадал с гладким числом u: $g = u = (p-1)/w$.

В обозначениях предыдущего следствия, положим: $\beta = (p-1)/g = w$, $t = (p-3)/2$. Иными словами, $p - 1 = g\beta$, $u = g$, $w = \beta$, $c = 1$.

При таком выборе, $\beta = -g^{-1} \bmod p$. Элемент g^{-1} - первообразный, т.к. при $ord_p g^{-1} = m < p-1$, получим $g^{-m+p-1} = 1 \bmod p$, что неверно. Следовательно, если $n = ord_p \beta < p-1$, то n - нечетное число, делящее $p-1$.

Однако, n не делит $p-1$, т.к. $\beta^{(p-1)/2} \equiv \left(-g^{-1}\right)^{2k} = g^{-(p-1)/2} = -1 \bmod p$. Следовательно, элемент β первообразный. Кроме того, $\beta^t = \beta^{(p-1)/2}\beta^{-1} \equiv -\beta^{-1} \equiv g \bmod p$.

Таким образом, для схемы цифровой подписи с указанными параметрами подпись может быть подделана. Покажем, что такие схемы существуют.

Пусть теперь $p = 4k + 1$, где k – простое и $p \geq 29$. Тогда можно выбрать $g = 2$.

Действительно, $2^{(p-1)/k} \neq 1(p)$. Далее, по критерию Эйлера, $2^{(p-1)/2} = \left(\frac{2}{p}\right) \bmod p$. Символ Лежандра $\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8} \bmod p$. Показатель в правой части нечетен: $(p^2-1)/8 = 2k^2 + k$, т.е. $2^{(p-1)/2} = -1 \bmod p$.

Таким образом, слабую подпись Эль-Гамаля можно получить, используя, например, следующие параметры: $p = 4k + 1$, где k – большое простое число,

$g=2,\quad p-1=g\beta,\quad u=g,\quad w=\beta,\quad c=1,\quad t=(p-3)/2,$
$(r,s)=(\beta,t(h-wz)).$

Здесь z удовлетворяет сравнению $g^{wz}\equiv y^{w}(\mathrm{mod}\ p)$, где y - открытый ключ цифровой подписи, а u используется для определения z.

Глава 11.
МЕТОДЫ ФАКТОРИЗАЦИИ ПОЛЛАРДА

Основное в этой главе…

- **(P-1) - метод факторизации Полларда…..……...142**
- **Po - метод факторизации Полларда……..……144**

При анализе криптосистемы RSA легко обнаружить, что сложность задачи факторизации модуля $N = pq$, на которой основана стойкость системы, зависит от соотношений между сомножителями, например, от величины разности $q - p$.

На сложность факторизации влияют, кроме того, индивидуальные особенности каждого из сомножителей, выраженные как теоретико-числовые свойства некоторых функций от p и q [47].

11.1. (P-1) - метод факторизации Полларда

Рассмотрим метод разложения числа *n* на сомножители, известный как $(p-1)$-метод факторизции Полларда. Метод является частным, т.е. он применим не для каждого n.

Фактически, $(p-1)$-метод Полларда является оптимизацией следующего подхода.

Пусть *p* - нетривиальный делитель числа *n*. Перебираем пары чисел a, t. На каждом шаге вычисляем наибольший общий делитель двух чисел $d = (a^t - 1, n)$. Если $a^t = 1 \bmod p$, но $a^t \neq 1 \bmod n$, то *d* является нетривиальным делителем *n*.

Пусть дано нечетное натуральное число $n = pq$, где *p* - его наименьший простой делитель и $(p,q) = 1$. Рассмотрим задачу факторизации *n* при условии, что число $p-1$ является гладким, т.е., $p - 1 = \prod_i p_i^{m_i}$, $\forall i \; p_i \le B$, где *B* - граница, определяемая вычислительными возможностями при реализации алгоритма.

Допустим, для некоторого *a*, $(a,n) = 1$, нам удалось локализовать значение $\mathrm{ord}_p a$, которое, очевидно, делит $p-1$.

Например, удалось найти число v, удовлетворяющее следующим условиям:

1) v делится на $\mathrm{ord}_p a$;

2) $a^v = 1 \bmod p$, но $a^v \neq 1 \bmod n$.

В этом случае, поскольку $a^v - 1$ не делится на n, мы можем определить $d = \left(a^v - 1, n\right)$, где $d < n$, $d = \lambda p$.

Все сводится к тому, как построить число v и воспользоваться его свойствами, при неизвестном p.

Сначала, в зависимости от вычислительных возможностей, выберем параметр k и построим число $v = v(k)$ так, чтобы заведомо $v(k)$ делилось на $p-1$.

Например, $v(k) = k!$, либо $v(k) = \mathrm{НОК}(1,2,\ldots k)$, где $k \approx \lfloor \sqrt{n} \rfloor$, поскольку $p < \sqrt{n}$. Данный этап является неформальным и наиболее сложным.

Не исключено, что следует выбирать $v(k) = m(k)^h$ как степень произведения всех, либо части простых чисел, меньших B, поскольку $p-1$ может делиться на высокие степени малых простых чисел. По возможности, h следует выбирать, начиная с минимального значения.

Если $p-1$ делит $v(k)$, то порядки $ord_p a$ всех чисел a, взаимно простых с p, также делят $v(k)$. Следовательно $a^v = 1 \bmod p$. Поскольку необходимо, чтобы $a^v \neq 1 \bmod n$, то очевидно, что v не должно делиться на $\varphi(n)$.

При этом все же нельзя утверждать, что $a^v \neq 1 \bmod n$, поскольку возможен случай, когда $ord_n a$ делит $v(k)$. Поэтому, при $\left(a^v - 1, n\right) = n$

необходимо псевдослучайно выбрать новое значение *a* и повторить вычисление $\left(a^v - 1, n\right)$.

Пример.

Найти нетривиальный делитель d числа $n = 2431 = p_1 p_2 p_3 = 11 \cdot 13 \cdot 17$.

Выберем относительно небольшое значение *k*, скажем, $k = \left\lfloor \sqrt[4]{n} \right\rfloor = 7$.

Заметим, что если выбрать $v(k) = (2 \cdot 3 \cdot 5 \cdot 7)^4$, то мы не сможем решить задачу. Действительно, все $\varphi(p_i)$, $i = 1,2,3$ делят $v(k)$, следовательно, для всех a, взаимно простых с n, $\left(a^v - 1, n\right) = n$.

Пусть $v(k) = (2 \cdot 3 \cdot 5 \cdot 7)^2 = 210^2$. В данном случае $\varphi(17)$ не делит $v(k)$. Конечно, заранее это не может быть известно.

Выберем *a*: $(a, n) = 1$, например, $a = 3$.

Вычислим $a^v - 1 = 3^{210 \cdot 210} - 1 \bmod n$.

Получим:

$$a^v - 1 \equiv 3^{180} - 1 \equiv 1287 \bmod 2431, \; d = \text{НОД}(1287, 2431) = 143.$$

11.2. Ро - метод факторизации Полларда

Следующий метод, который мы рассмотрим, носит название ро-метода Полларда. Название связано с тем, что метод осуществляет случайный поиск, используя свойства цикличности некоторых последовательностей.

Эти последовательности не являются чисто периодическими, а имеют так называемые подходы к циклам, что графически выглядит как греческая буква ρ (ро).

Данный метод эффективнее, чем метод полного перебора делителей n. Кроме того, идея метода применима и в других ситуациях, например, для логарифмирования в группах точек эллиптических кривых.

В ро-методе прежде всего выбирается некоторое отображение $f : Z/nZ \to Z/nZ$ кольца вычетов по модулю *n* в себя. Необходимо, чтобы значения этого отображения можно было бы вычислить достаточно быстро. Например, это может быть полином с целыми коэффициентами, скажем, $f(x) = x^2 + x + 1 \bmod n$.

Затем псевдослучайно выбирается число x_0 (начальное значение) и рассматриваются элементы последовательности итераций: $x_{j+1} = f(x_j)$, $j = 0,1,2,\ldots$, то есть $x_1 = f(x_0)$, $x_2 = f(x_1)$, $x_3 = f(f(f(x_0)))$ и т.д.

Очевидно, данная последовательность циклическая не только как последовательность вычетов по модулю n, но и как последовательность вычетов по некоторому простому делителю $p < \sqrt{n}$ числа *n*.

Будем рассматривать пары элементов последовательности.

Среди этих пар найдутся такие, скажем x_j, x_k, что $x_j \neq x_k (n)$, но $x_j = x_k (p)$. Если такая пара, назовем ее критической, попадется среди рассматриваемого множества пар, то можно найти собственный делитель *r* числа *n* вида $r = (x_j - x_k, n) = \lambda p$.

Пример. Найти нетривиальный делитель числа $n = 2431$.

Выберем $f(x) = x^2 + x + 1$, $x_0 = \lfloor \sqrt{2431} \rfloor = 50$. Вычисляем рекурренту и проверяем пары: $x_0 = 50\,(2431)$, $x_1 = 50^2 + 50 + 1 = 120(2431)$, $x_2 = 2366\,(2431)$, $x_3 = 1730\,(2431)$, $x_4 = 2070(2431)$ и так далее.

При этом, $x_4 - x_1 = 1950(2431)$ имеет с n нетривиальный общий делитель: $(2431,1950) = 13$.

Желательно максимально сократить количество пар, подлежащих проверке. С этой точки зрения выбор полинома в качестве отображения f удобен, поскольку из $x = y(\mathrm{mod}\, p)$ следует $f(x) = f(y)(\mathrm{mod}\, p)$.

Очевидно, расположение критических пар определяется свойствами рекурренты, приведенной по (неизвестному) модулю r. В состав рекурренты входят r различных элементов.

11.2.1. Оценка вероятности выбора критической пары

Для оценки количества членов рекуррентной последовательности, необходимого для появления критической пары, используется допущение, что отрезки длины l последовательности, порожденной выбранным многочленом, в среднем, ведут себя как результаты случайной выборки объема l из r элементов.

На самом деле это не так, поскольку многочлен нельзя рассматривать как случайное отображение.

Практика, тем не менее, показывает, что полученная оценка достаточно хорошо согласуется с реальными данными [15,16].

При указанном предположении выясним, сколько требуется в среднем произвести итераций, чтобы появилась хотя бы одна критическая пара для фиксированного делителя r числа n.

Рассмотрим последовательности, полученные для каждого начального значения x_0, применением каждого отображения $f: Z/nZ \to Z/nZ$.

Теорема [15, гл.5]. Пусть $\lambda > 0$ и $l = 1 + \lfloor \sqrt{2\lambda r} \rfloor$. Пусть S - множество, состоящее из r элементов и f - случайно и равновероятно выбранное отображение $f : S \to S$. Пусть $x_0 \in S$ и $x_{j+1} = f(x_j),\ j = 0,1,2,\ldots$.

Тогда доля всех сочетаний (f, x_0), для которых все элементы $x_0, x_1, \ldots, x_{l-1}$ попарно различны, менее $e^{-\lambda}$.

Действительно, каждая функция f отображает последовательность аргументов $u = (0,1,\ldots,r-1)$ в уникальную последовательность $\tilde{f}(u) = (x_0, x_1, \ldots, x_{r-1})$ длины r, т.е. всего существует r^r различных функций.

Каждой функции соответствует r рекуррентных последовательностей вида $(f, x_0), \ldots, (f, x_{r-1})$. Следовательно, общее число всех последовательностей указанного вида равно r^{r+1}.

Пусть в последовательности (f, x_0) элементы $x_0, x_1, \ldots, x_{l-1}$ попарно различны. Значение x_0 можно выбрать r способами. Если $x_1 \neq f(x_0) \bmod r$, то число вариантов x_1 равно $r-1$ и т.д. Поэтому число последовательностей $x_0, x_1, \ldots, x_{l-1}$, для которых все элементы попарно различны, равно $r(r-1)\ldots(r-l)$.

Следовательно, количество последовательностей $x_0, x_1, \ldots, x_{r-1}$ длины r, таких, что первые l элементов $x_0, x_1, \ldots, x_{l-1}$ попарно различны, равно

$$h = r^{r-l} \prod_{j=0}^{l} (r - j).$$

Вероятность появления указанных последовательностей равна

$$h / r^{r+1} = r^{-l-1} \prod_{j=0}^{l} (r - j) = \prod_{j=1}^{l} (1 - j/r).$$

Оценим логарифм этой вероятности, воспользовавшись тем, что $\ln(1-x)<-x$, при $0<x<1$.

Поскольку сумма l первых натуральных чисел равна $l(l+1)/2$, получим:

$$\ln\prod_{j=1}^{l}(1-j/r)<\sum_{j=1}^{l}(-j/r)=-l(l+1)/2r<-l^2/2r<-\left(\sqrt{2\lambda r}\right)^2/2r=-\lambda .$$

Итак, для появления критической пары с вероятностью не ниже $1-e^{-\lambda}$, необходимо вычислить $l=1+\lfloor\sqrt{2\lambda r}\rfloor$ элементов последовательности $\{x_j\}$.

Легко видеть, что поиск критической пары путем сравнения всех элементов последовательности $x_0,x_1,\ldots,x_{l-1}$ требует вычисления НОД(x_j-x_k,n) для всех $j<k$, что слишком трудоемко.

11.2.2. Оптимизация выбора критической пары

Оказывается, при поиске критической пары можно выиграть во времени, если работать с более длинной последовательностью $x_0,x_1,\ldots,x_{t-1}$, но для каждого вновь формируемого элемента вычислять НОД(x_j-x_k) только один раз, сравнивая его с элементами x_j, обладающими некоторыми специфическими номерами.

Заметим, что если пара x_j,x_k критическая, то $f(\ldots f(x_j))=f(\ldots f(x_k))\bmod p$ для любого числа итераций. Иными словами, пары x_{j+s},x_{k+s} необходимо проверять не более одного раза. С другой стороны, нам не обязательно искать критические пары с минимальными номерами, чтобы решить задачу.

Предлагаемый способ опирается на поиск критической пары, которая не обязательно является первой, поэтому необходимая длина рекурренты увеличивается. Оказывается, она увеличивается не более, чем в четыре раза.

Пусть (j_0, k_0) - минимальные индексы критической пары, существующей в нашей рекурренте. Найдем индексы другой критической пары (j, k), где $k - j = k_0 - j_0$.

Будем ориентироваться на длину представления k_0 в двоичной системе счисления. Пусть представление k_0 в двоичной системе счисления занимает $h+1$ бит. Это значит, что $2^h \le k_0 < 2^{h+1}$.

Выберем в качестве j максимальное $(h+1)$-битовое число, т.е. $j = 2^{h+1} - 1$. Тогда $k = j + (k_0 - j_0)$. Поскольку $k_0 > j_0$, то $2^{h+1} \le k < 2^{h+2}$. Следовательно, $k < 4 \cdot 2^h \le 4k_0 \le 4l$.

Таким образом, мы можем искать критические пары, испытывая все значения $k > j$ в каждом из диапазонов вида $2^m \le k < 2^{m+1}$, при фиксированном, определяемом диапазоном, значении $j = j(m) = 2^m - 1$. Эти пары имеют индексы вида $(2^m - 1, 2^m \le k < 2^{m+1})$.

Пример. Разложим число $n = 4087$, используя $f(x) = x^2 + x + 1$ и начальное значение $x_0 = 2$.

$$x_1 = f(2) = 7(n), \qquad (x_1 - x_0, n) = (7 - 2, n) = 1;$$

диапазон двухбитовых значений k: $m = 1$, $j = 1$:

$$x_2 = f(7) = 57(n), \qquad (x_2 - x_1, n) = (57 - 7, n) = 1;$$

$$x_3 = f(57) = 3307(n), \quad (x_3 - x_1, n) = (3307 - 7, n) = 1;$$

диапазон трехбитовых значений k: $m = 2$ $j = 3$:

$$x_4 = f(3307) = 2745(n), \; (x_4 - x_3, n) = (2745 - 3307, n) = 1;$$

$$x_5 = f(2745) = 1343(n),\ (x_5 - x_3, n) = (1343 - 3307, n) = 1;$$

$$x_6 = f(1343) = 2626(n),\ (x_6 - x_3, n) = (2626 - 3307, n) = 1;$$

$$x_7 = f(2626) = 3734(n),\ (x_7 - x_3, n) = (3734 - 3307, n) = 61;$$

$$4087 = 61 \cdot 67.$$

Глава 12.
НЕКОТОРЫЕ СЛУЧАИ ОСЛАБЛЕНИЯ КРИПТОСИСТЕМЫ RSA

Основное в этой главе…

- **Атаки на RSA, не использующие факторизацию модуля……...152**
- **Атаки на RSA, использующие факторизацию модуля……..156**
- **Алгоритм факторизации Диксона……..157**

Криптосистема *RSA* основана на функции возведения в степень в кольце вычетов по двупростому модулю $n = pq$. Эта функция часто применяется в смешанных криптосистемах и широко используется в криптопротоколах, рекомендованных различными стандартами.

Зависимость сложности обращения степенной функции от ее параметров и наличие частных случаев снижения стойкости криптосистемы *RSA* приводит к выводу о необходимости разработки специальных алгоритмов для генерации простых чисел p и q.

12.1. Атаки на RSA, не использующие факторизацию модуля

Известным свойством криптосистемы *RSA* является зависимость ее стойкости от свойств сомножителей модуля $n = pq$. При необоснованном выборе этих сомножителей возможно полное дешифрование криптосистемы.

Важной особенностью криптосистемы *RSA* является возможность чтения отдельных сообщений или подделки цифровой подписи независимо от свойств сомножителей модуля, например, за счет неудачного выбора открытого ключа.

Существенно также и то, что ослабить стойкость системы можно без раскрытия секретного ключа, используя ее для зашифрования посторонних сообщений с целью использования возникающих в шифртексте специфических математических соотношений.

Случаи снижения стойкости системы *RSA,* без использования свойств сомножителей модуля, облегчающих его непосредственную факторизацию, сводятся к зависимостям между открытыми текстами сообщений и к особенностям выбранных ключей [22].

Подобные слабости не связаны с качеством криптосистемы *RSA* как таковой, но отражают то обстоятельство, что стойкость криптосистемы может изменяться, в зависимости от особенностей ее проектирования и эксплуатации.

Рассмотрим ряд примеров. Будем исходить из того, что некоторый абонент А использует криптосистему *RSA* с параметрами (e,d,n), $n = pq$.

Пример 1. Пусть в сети используется общее значение модуля для нескольких абонентов и перехвачены криптограммы вида $c_1 = m^{e_1}(n)$, $c_2 = m^{e_2}(n)$, где экспоненты взаимно просты.

Вычислив с помощью расширенного алгоритма Эвклида значения r, s: $re_1 + se_2 = 1$, получаем возможность определить открытый текст: $c_1^r c_2^s = m(n)$.

Заметим, что общее значение модуля для двух абонентов позволяет каждому из них читать сообщения, предназначенные для другого (см. ниже).

Пример 2. Очевидно, что если c - шифртекст, то расшифрованное сообщение имеет вид: $m = c^d(n)$. Выберем псевдослучайно число *x* вида $x = r^e(n)$ и вычислим произведение $y = xc(n)$, т.е. замаскируем шифртекст. Если по просьбе абонента В абонент А подпишет сообщение y: (y, y^d), то абонент В получит значение $y^d = x^d c^d = rm(n)$, откуда можно найти m. Очевидно, использование хэш-функции не позволит воспользоваться данной лазейкой.

Пример 3. Малое значение экспоненты, например, $e = 3$, не позволяет непосредственно зашифровать блок *x*, представляемый числом в три раза меньшей длины, чем модуль.

Действительно, тогда $c = x^3 < n$, т.е. приведение по модулю не происходит. Откуда $x = \sqrt[3]{c}$ - обычное число.

Пример 4. (Атака Франклина). При зашифровании сообщений с известной по модулю *n* разностью на стойкость криптосистемы может повлиять величина открытого ключа [48].

Пусть $e = 3$, m_1, m_2 - сообщения, $m_2 = m_1 + \Delta(\mathrm{mod}\, n)$ и соответствующие шифртексты равны $a = m_1^e(n), b = m_2^e(n)$. Пусть также известно значение $\Delta \neq 0(n)$.

То же, другими словами, означает, что два многочлена $g(x) = x^3 - a$ и $f(x) = (x + \Delta)^3 - b$ имеют общий корень $x = m_1(n)$. Отсюда следует, что m_1 является корнем наибольшего общего делителя указанных многочленов.

Можно показать, что условие взаимной однозначности шифра позволяет легко определить корень полинома НОД$(f(x), g(x))$ в случае, когда его степень больше единицы.

С большой вероятностью выполняются также условия обратимости по модулю *n* некоторых элементов (они будут очевидны по ходу рассуждения), при которых искомый наибольший общий делитель имеет вид $ux + v$ и его корень может быть вычислен.

Напомним, что $d(x) = \text{НОД}(f(x), g(x))$, где степень $f(x)$ не ниже степени $g(x)$, можно найти с помощью алгоритма Эвклида: $r_1(x) = f(x) \mathrm{mod}\, g(x)$, $r_2(x) = g(x) \mathrm{mod}\, r_1(x)$ $r_3(x) = r_1(x) \mathrm{mod}\, r_2(x)$ и т.д., пока остаток от деления не станет равным нулю. В этом случае предыдущий остаток есть $d(x)$.

Вычислим НОД$(f(x), g(x))$ по модулю *n* явно, с неопределенными коэффициентами.

Имеем: $g(x) = x^3 - a$, $f(x) = (x + \Delta)^3 - b = x^3 + 3x^2\Delta + 3x\Delta^2 + \Delta^3 - b$.

Деля $f(x)$ на $g(x)$, получим, что старший член частного равен 1. Поэтому остаток от деления равен $r_1(x) = f(x) - 1 \cdot g(x)$, где $r_1(x) = 3x^2\Delta + 3x\Delta^2 + \Delta^3 - b + a$.

Ясно, что $d(x)=\text{НОД}(r_1(x),g(x))$.

Делим $g(x)$ на $r_1(x)$. Старший член частного равен $\frac{1}{3\Delta}x$.

В итоге, $g(x)=\frac{1}{3\Delta}xr_1(x)+s_2(x)$, где $s_2(x)=-x^2\Delta-\frac{1}{3\Delta}(\Delta^3+a-b)x-a$.

Следовательно, $d(x)=\text{НОД}(r_1(x),s_2(x))$. Поэтому далее делим $r_1(x)$ на $s_2(x)$ и получаем старший член частного равный (–3).

Вычисляем остаток от деления $r_1(x)$ на $s_2(x)$, который имеет вид $d(x)=\left[3\Delta^2-\frac{1}{\Delta}(\Delta^3+a-b)\right]x+\Delta^3-2a-b$.

Откуда получаем корень полинома $d(x)$: $m_1=x=\frac{\Delta(2a+b-\Delta^3)}{2\Delta^3-a+b}$.

Условия корректности рассуждения очевидны: $\text{НОД}(\Delta,n)=1$, $\text{НОД}(2\Delta^3-a+b,n)=1$.

Пример 5. Необходимость выбора несовпадающих модулей при построении криптосистемы RSA для различных пользователей доверенным лицом. Уточним, что в этом случае числа p,q пользователям не известны.

Пусть пользователи A и B используют соответственно криптосистемы с параметрами (e_A,d_A,n) и (e_B,d_B,n). Очевидно, пользователь B, например, в состоянии вычислить значение $e_Bd_B-1=k\varphi(n)$, хотя сомножители в правой части ему неизвестны.

По построению криптосистемы $(e_A,\varphi(n))=1$, следовательно, наибольший общий делитель чисел e_A и e_Bd_B-1 совпадает с $d=(e_A,k)$.

Поэтому числа e_A и $\frac{e_B d_B - 1}{d}$ взаимно просты. Кроме того, $\frac{e_B d_B - 1}{d} = \frac{k}{d}\varphi(n)$, т.е. левая часть делится на $\varphi(n)$.

С помощью алгоритма Эвклида пользователь B может найти числа C и D, такие, что $\frac{e_B d_B - 1}{d} C + e_A D = 1$, откуда $e_A D = 1(\varphi(n))$. Таким образом, расшифрование сообщений можно производить с помощью ключа D.

12.2. Атаки на RSA, использующие факторизацию модуля

Задача разложения числа $n = pq$ на сомножители является нерешенной проблемой. По этой причине не полностью ясны условия, которым должны соответствовать сомножители p и q.

С другой стороны, имеется ряд частных случаев и соответствующих алгоритмов, когда такое разложение возможно. Упомянутые алгоритмы могут быть различной степени сложности: от элементарных, до использующих тонкие теоретико-числовые свойства.

Рассмотрим некоторые элементарные случаи.

Поскольку произведение $n = pq$ известно, то любое другое нетривиальное соотношение между p и q может привести к системе двух уравнений, позволяющей получить разложение n.

Пусть скажем, $q - p < c$. Если разности, вплоть до c, можно перебрать, то можно решить систему и найти сомножители. Отсюда следует, что p и q не должны быть близки.

Более того, даже относительная величина разности не может быть произвольной.

Действительно, пусть $p<q$ и разность $q-p<2n^{1/4}$. Из этого условия и тождества $4n+(q-p)^2=(q+p)^2$ следует, что $q+p$ превышает корень квадратный из $4n$, но не более чем на единицу. Действительно, $(q-p)^2<4\sqrt{n}$, поэтому $2\sqrt{n}<q+p<\sqrt{4n+4\sqrt{n}}<2\sqrt{n}+1$.

Таким образом, $q+p=\lfloor 2\sqrt{n}\rfloor+1$, следовательно, p и q легко определить.

Следующий пример. Пусть порядки чисел по (неизвестному) модулю p невелики. Тогда, испытывая числа a и h перебором, можно оказаться в ситуации, когда $a^h=1(p)$ и в то же время $a^h\neq 1(pq)$.

В этом случае НОД $(a^h-1,pq)=p$. Поэтому порядки чисел по модулю p должны быть большими. Но порядки вычетов должны делить функцию Эйлера от модуля: $p-1=0(h)$, таким образом, приходим к требованию, что числа $p-1$ и $q-1$ должны обладать большими простыми делителями.

12.2.1. Алгоритм факторизации Диксона

Рассмотрим теперь более общий пример атаки на RSA, непосредственно направленной на факторизацию модуля [15,16,21].

В целом ряде алгоритмов факторизации нечетного числа n используется идея Лежандра, состоящая в поиске пар чисел x,y, удовлетворяющих соотношению $x^2=y^2(n)$.

Наибольший общий делитель $(x-y)$ и n найти легко. Если он равен единице или совпадает с n, то необходимо искать следующее соотношение, иначе, упомянутый делитель равен некоторому делителю числа n.

Будем говорить, что для чисел x, y выполняется нетривиальное соотношение $x^2 = y^2 (n)$, если $x \neq \pm y (n)$. Для таких чисел выражение $(x - y)(x + y)$ делится на *n*, причем каждый из указанных сомножителей, скажем $(x - y)$, имеет с *n* нетривиальный общий делитель.

В алгоритме факторизации Диксона для поиска нетривиальных соотношений используется заранее выбранная совокупность относительно небольших простых чисел, не превышающих некоторой границы *M*.

Множество указанных чисел называется факторной базой (или фактор-базой): $B = \{p_1, \ldots p_h\}$, $\max p_i \leq M$. Граница выбирается, исходя из наличия вычислительных ресурсов. Часто в факторную базу вводят число –1.

Это делается для того, чтобы уменьшить абсолютные величины чисел, участвующих в вычислениях. Мы также будем считать, что –1 входит в *B*.

Будем говорить, что натуральное число b является B - числом, если после приведения по модулю *n* число b^2 разлагается в произведение степеней простых из факторной базы, иными словами, если $b^2 \bmod n = \prod_{p \in B} p^{u_p(b)}$.

Заметим, что когда $b^2 < n$, то все показатели в правой части последнего равенства четные, т.е. отличий от случая разложения квадрата натурального числа нет.

Если же b^2 велико, то после приведения по модулю *n* разложение его вычета может содержать как четные, так и нечетные степени простых чисел.

Так что, с одной стороны, $b^2 \bmod n$ сравнимо с произведением квадратов, а с другой стороны – его можно записать иначе: в виде произведения как четных так и нечетных степеней простых чисел.

Это возможно потому, что разложение вычетов на степени простых сомножителей не является однозначным, вследствие приведения по модулю n.

Подобная ситуация в натуральных числах не имеет места, в силу основной теоремы арифметики.

Далее. Если мы найдем несколько *B*-чисел, то перемножив соответствующие части сравнений вида $b^2 \bmod n = \prod_{p \in B} p^{u_p(b)}$, получим в его левой части квадратичный вычет, корень из которого нам известен.

Метод поиска второго корня, дающего нетривиальное соотношение, в общем случае не найден.

Однако, зная показатели в правой части для каждого b^2, мы в состоянии обнаружить случай, когда все показатели при простых сомножителях в произведении указанных правых частей окажутся четными. Тогда мы сможем найти второй корень, уменьшая показатели вдвое.

Таким образом, алгоритм Диксона направлен на увеличение вероятности получения нетривиальных соотношений.

Схему алгоритма рассмотрим на примере.

Пусть необходимо факторизовать число $n = 1829$. В качестве факторной базы *B* выберем множество простых чисел, меньших 15, к которому добавим (-1): $B = \{-1,2,3,5,7,11,13\}$.

Начнем с поиска *B*-чисел. Такие числа b_i мы будем искать среди чисел вида $\left[\sqrt{1829k}\right]$ и $\left[\sqrt{1829k}\right]+1$, $k = 1,2\ldots$, пока не наберем их достаточно для возникновения нужной ситуации.

Числа b_i характеризуются условием разложения по элементам нашей базы: $b_i^2 \bmod n = \prod_j p_j^{u_{ij}}$.

Итак, для $k = 1$ рассмотрим два числа: $\left[\sqrt{1829k}\right] = 42$ и 43.

Поскольку $p_0 = 2302 + 59 \cdot 47 = 5075$, то 42 является B-числом. Действительно, $42^2 = (-1)^1 5^1 13^1 (1829)$. Аналогично, $43^2 = 2^2 5^1 (1829)$.

Заметим, что каждому *B*-числу b удобно поставить в соответствие вектор $a(b)$ показателей разложения $b^2 \bmod n$ по элементам базы.

Отсутствующим в разложении элементам базы, как обычно, соответствует показатель равный нулю.

Таким образом, $a(42)=(1001001)$, $a(43)=(0201000)$.

Кроме вектора $a(b)$, введем вектор $e(b)=a(b)\bmod 2$, поскольку в ходе алгоритма возникает необходимость вычислений сумм векторов с точностью до четности компонент, что соответствует операциям над векторами по модулю два. Таким образом, $e(42)=(1001001)$, $e(43)=(0001000)$.

Нашей целью является получить столько *B*-чисел, чтобы из них можно было выбрать подмножество, для которого сумма векторов вида $a(b)$ состояла из четных координат.

Это эквивалентно линейной зависимости соответствующих векторов вида $e(b)$ над $GF(2)$. Теперь ясно, что необходимое количество *B*-чисел не превышает увеличенной на 1 размерности векторов $e(b)$.

Можно убедиться, что числа $42,43,61,74,85,86$ являются B-числами.

Выпишем таблицу соответствующих векторов вида $a(b)$.

Таблица 5. Разложение квадратов B-чисел по фактор-базе

	-1	2	3	5	7	11	13
42	1	0	0	1	0	0	1
43	0	2	0	1	0	0	0
61	0	0	2	0	1	0	0
74	1	0	0	0	0	1	0
85	1	0	0	0	1	0	1
86	0	4	0	1	0	0	0

Очевидно, $e(43) \oplus e(86) = (0000000)$, т.е. в этом случае наличие нетривиального соотношения не исключено. Перемножим квадраты соответствующих *B*-чисел по модулю *n*.

Приравняем результат к произведению разложений квадратов соответствующих чисел, полученных на основе векторов $a(b)$ из таблицы: $43^2 \cdot 86^2 = 2^6 5^2 (\mathrm{mod} 1829)$.

Мы можем извлечь корни из обеих частей и записать $(43 \cdot 86) = \pm 40 (\mathrm{mod} 1829)$, но левая часть равна $43 \cdot 86 = 40 (1829)$ и мы получили тривиальное соотношение. Необходимо продолжить поиск линейно зависимых векторов вида $e(b)$.

Заметим, что $e(42) \oplus e(43) \oplus e(61) \oplus e(85) = (0000000)$.

Выпишем сравнение $42^2 \cdot 43^2 \cdot 61^2 \cdot 85^2 = 2^2 3^2 5^2 7^2 13^2 (\mathrm{mod} 1829)$, что дает соотношение $1459^2 = 901^2 (\mathrm{mod} 1829)$ и $1459 \neq \pm 901 (\mathrm{mod} 1829)$. Поэтому число НОД $(1459 + 901, 1829) = 59$ является делителем 1829.

Глава 13.
ТЕСТ ФЕРМА ПРОВЕРКИ ЧИСЕЛ НА ПРОСТОТУ

Основное в этой главе…

- **Тест на основе малой теоремы Ферма……..….166**
- **Основные свойства псевдопростых чисел…….…..…166**
- **Понятие о последовательностях Люка. (n+1) – критерий Люка..…..171**
- **(P+1) – метод факторизации Вильямса…….…..….172**

При построении асимметричных криптосистем, а также при модификации их параметров в ходе эксплуатации, возникает необходимость построения сверхбольших псевдослучайных простых чисел, обладающих некоторыми специфическими свойствами.

Соответствующие вычислительные процедуры включают в себя алгоритмы, реализующие этап проверки чисел на простоту. В литературе и криптографической практике подобные алгоритмы носят название тестов.

В основе тестов лежат т.н. критерии простоты [15,16]. Существует два типа критериев простоты: детерминированные и вероятностные. Детерминированные тесты позволяют доказать, что тестируемое число – простое. Практически применимые детерминированные тесты способны дать положительный ответ не для каждого простого числа, поскольку используют лишь достаточные условия простоты.

Детерминированные тесты более полезны, когда необходимо построить большое простое число, а не проверить простоту, скажем, некоторого единственного числа.

В отличие от детерминированных, вероятностные тесты можно эффективно использовать для тестирования отдельных чисел, однако их результаты, с некоторой вероятностью, могут быть неверными. К счастью, ценой количества повторений теста с модифицированными исходными данными вероятность ошибки можно сделать как угодно малой.

13.1. Решето Эратосфена и критерий Вильсона

Широко известен метод последовательного построения списка всех простых чисел, не превосходящих заданного числа N. Этот метод называется решетом Эратосфена.

В списке L всех чисел от 2 до N вычеркиваем числа кратные 2, большие двойки. Из оставшихся чисел выбираем наименьшее (оно равно 3), а из L вычеркиваем числа, кратные 3, большие тройки и так далее. Если очередное,

выбираемое из L, наименьшее число $a > \lfloor \sqrt{N} \rfloor$, то работу прекращаем. Оставшиеся в L числа составляют искомый список простых чисел.

Данный метод основан на том, что все кратные указанного (простого) числа a и кратные превосходящих его чисел к моменту выбора a уже вычеркнуты. Действительно, если $ab \le N$, то $b \le \lfloor \sqrt{N} \rfloor$, следовательно, все кратные числа b были вычеркнуты ранее, как кратные некоторого его простого делителя.

Уместно задать вопрос, существуют ли критерии простоты, являющиеся необходимыми и достаточными?

Примером такого критерия является т.н. критерий Вильсона: число n - простое тогда и только тогда, если $(n-1)! = -1(n)$.

Для $n = 2,3$ утверждение очевидно. Если $n = p > 3$ - простое, то для каждого взаимно простого с p натурального числа a существует обратный по модулю p, причем, если $a \ne \pm 1(p)$, то $a^{-1} \ne a(p)$. В число $(p-1)!$ входят сомножителями все ненулевые вычеты по модулю p. Поэтому произведение всех вычетов $a \ne \pm 1(p)$ дает единицу, а пара вычетов $a = \pm 1(p)$ дает в произведении –1.

Обратно, если $n = ab$, $1 < a < n$, то a делит $(n-1)!$, поэтому вычета, обратного к $(n-1)! \bmod n$, не существует. Значит, $(n-1)!$ не может быть сравнимо с (-1) по модулю n, что противоречит условию.

13.2. Тест на основе малой теоремы Ферма

При построении вероятностных критериев простоты возникает ряд типичных вопросов, которые удобно рассмотреть на примере, в качестве которого выберем тест на основе малой теоремы Ферма. Как известно, эта теорема утверждает, что если n - простое, то для всех a, взаимно простых с *n*, выполняется условие (сравнение Ферма): $a^{n-1} = 1(n)$.

Таким образом, если сравнение Ферма не выполнено, хотя бы для одного числа в интервале $\{2,\ldots,n-1\}$, то *n* - составное.

Тест на основе малой теоремы Ферма заключается в следующем.

Псевдослучайно выбираем вычет $a \in \{2,\ldots,n-1\}$, проверяем условие $(a,n)=1$. Если это условие не выполнено, значит, *n* - составное. Проверяем сравнение Ферма. Если оно не выполняется, то число *n* - составное. Иначе, повторяем тест для другого значения *a*.

Очевидно, основной вопрос состоит в том, чтобы оценить, в какой мере тест является эффективным. Прежде всего, следует выяснить, существуют ли составные числа *n*, для которых при некотором *a* выполняется условие малой теоремы Ферма. Оказывается это так, поскольку существуют контрпримеры: $2^{340} = \left(2^{10}\right)^{34} = 1(341)$, $7^{24} = 1(25)$.

13.2.1. Основные свойства псевдопростых чисел

Назовем составное число *n* псевдопростым по основанию *a*, если пара (a,n) удовлетворяет сравнению Ферма $a^{n-1} = 1(n)$.

Оказывается, можно показать, что для любого $a > 1$ существует бесконечно много псевдопростых чисел по основанию *a*.

В частности, если пара $(2,n)$ удовлетворяет указанному сравнению, то ему удовлетворяет также и пара $(2,2^n-1)$. Действительно, по условию, $2^{n-1}-1$ делится на *n*, поэтому $2(2^{n-1}-1)=2tn$.

Следовательно, $2^{2tn}-1=(2^n-1)(2^{(2t-1)n}+\ldots+1)=0 \bmod(2^n-1)$.

Но $2tn=(2^n-1)-1$, а число 2^n-1 не является простым, поскольку n - составное. Поэтому число 2^n-1 является псевдопростым по основанию два.

Основные свойства псевдопростых чисел таковы.

Теорема [16]. Пусть *n* - нечетное составное число. Тогда:

а) *n* псевдопростое по основанию *a* в том и только том случае, когда $(a,n)=1$ и $ord_n a \mid n-1$;

б) если n псевдопростое по основаниям *a* и b, то *n* псевдопростое по основаниям ab и $ab^{-1}(\bmod n)$;

в) множество $F_n=\{a\in Z_n : a^{n-1}=1(\bmod n)\}$ образует мультипликативную подгруппу в мультипликативной группе Z_n^* обратимых элементов кольца вычетов по модулю n;

г) если n не является псевдопростым по основанию *a*, хотя бы для одного числа *a*, то $|F_n|\le(1/2)|Z_n^*|$. Заметим, что $|Z_n^*|=\varphi(n)\le n-1$. Таким образом, если тест Ферма выявляет составное *n* при одном основании *a*, то существует не менее $(n-1)/2$ оснований с аналогичным свойством.

Действительно, свойство а) следует из определения $ord_n a$. Свойства б) и в) следуют из правила почленного перемножения частей сравнений и проверки сравнения Ферма для основания b^{-1}.

Докажем свойство г). Пусть $F_n = \{a_1, \ldots, a_s\}$ и a - основание, по которому n не является псевдопростым. Тогда для любого i пара (aa_i, n) не удовлетворяет сравнению Ферма. Поэтому количество оснований, для которых n не является псевдопростым, не меньше, чем количество элементов в F_n. Следовательно, если $|F_n| > (1/2)|Z_n^*|$, то общее число элементов в Z_n^* окажется больше n, что невозможно.

Таким образом, можно заключить, что если существует (не известное нам) основание, по которому n не является псевдопростым, то при повторении теста Ферма k раз вероятность k-кратного выбора оснований из множества F_n не превосходит $(1/2)^k$. В этом случае вероятность ошибки теста стремится к нулю с увеличением k.

Проблема может возникнуть лишь в том случае, когда n является псевдопростым для всех (ненулевых) оснований. Оказывается, такие числа существуют. Они называются числами Кармайкла. Например, числом Кармайкла является число $n = 561 = 3 \cdot 11 \cdot 17$.

13.2.2. Свойства чисел Кармайкла

Свойства чисел Кармайкла описываются следующей теоремой.

Теорема (Кармайкл). Пусть n - нечетное составное число. Тогда

а) если $p^2 | n$, $p > 1$, то n не является числом Кармайкла;

б) если $n = p_1 p_2 \ldots p_k$, $p_i \neq p_j$ для $(i \neq j)$, то n - число Кармайкла в том и только том случае, когда $\forall i \ (p_i - 1) | (n - 1)$;

в) если $n = p_1 p_2 \ldots p_k$, $p_i \neq p_j$ $(i \neq j)$ - число Кармайкла, то $k \geq 3$.

Числа Кармайкла являются достаточно редкими. В пределах до 100000 существует лишь 16 чисел Кармайкла: 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, 46657 52633, 62745, 63973, 75361.

13.2.3. (n-1) - критерий Люка

Из предыдущего можно сделать вывод, что в тесте Ферма на простоту простые числа и числа Кармайкла неразличимы.

Тем не менее, сравнение Ферма можно использовать для построения тестов, достаточных для доказательства простоты конкретного числа, путем введения дополнительных условий в соответствующие критерии.

Достаточно эффективным критерием является т.н. $(n-1)$- критерий Люка, для применения которого должна быть известны все простые делители числа $(n-1)$. Основой данного критерия является следующая теорема.

Теорема (Люка). Натуральное число *n* является простым в том и только том случае, когда существует такое натуральное число *a*, что для любого собственного делителя $q \,||(n-1)$ числа $n-1$ выполняются следующие условия: $a^{n-1} = 1(n)$, $a^{(n-1)/q} \neq 1(n)$.

Доказательство. Пусть *n* - простое и $n-1 = \prod_{i=1}^{k} q_i^{m_i}$. Тогда существует элемент *a*, порядок которого максимален, т.е. равен $(n-1)$. Поэтому оба условия теоремы выполняются для всех $q \,||(n-1)$.

Обратно, предположим, что для некоторого элемента *a* выполнены условия теоремы и порядок числа *a* по модулю *n* равен *m*, где $m \neq n-1$. В этом случае $m|(n-1)$ и $(n-1) = km$, где $k > 1$.

Пусть $q \mid k$, тогда $a^{(n-1)/q} = 1(n)$. Это возможно лишь при $q = k = 1$, т.е. при $m = n-1$. Таким образом, порядок числа *a* - максимально возможный, следовательно, он совпадает с функцией Эйлера от n и $\varphi(n) = n-1$.

По определению функции Эйлера, это выполнимо только в случае, когда *n* - простое.

Замечание. Легко видеть, что условия теоремы Люка эквивалентны существованию натурального числа b, такого, что $ord_n b = n-1$.

Покажем, что в критерии Люка можно выбирать число *a* для каждого *q* независимо, т.е. в условии теоремы можно заменить *a* на $a(q)$.

Теорема (Сэлфридж). Натуральное число *n* является простым в том и только том случае, когда для любого собственного делителя $q \,||\, (n-1)$ числа $n-1$ существует такое натуральное число $a = a(q)$, что выполняются следующие условия: $a^{n-1} = 1(n)$, $a^{(n-1)/q} \neq 1(n)$.

Доказательство. Очевидно, из условий теоремы Люка следует выполнение условий теоремы Сэлфриджа. Необходимо доказать, что из условий теоремы Сэлфриджа следуют условия теоремы Люка. Мы сделаем это, убедившись, что в условиях теоремы Сэлфриджа существует натуральное число b, такое, что $ord_n b = n-1$.

Действительно, пусть $n-1 = \prod_{i=1}^{k} q_i^{m_i}$. По условию, для каждого собственного простого делителя $q \,||\, (n-1)$ найдется $a = a(q)$, такое, что $m = ord_n a(q)$ делит $(n-1)$, но $ord_n a(q)$ не делит число $(n-1)/q$. Пусть $(n-1) = md$. Тогда d не делится на q, иначе, число $(n-1)/q$ было бы

кратно m и выполнялось равенство $a^{(n-1)/q} = 1(n)$, что невозможно. Следовательно, для $q_i = q$, получим: $q_i^{m_i} | m$. Пусть $Q = Q(q) = q_i^{m_i}$.

Рассмотрим число вида $b(q) = a^{m/Q}(n)$. Порядок этого числа по модулю n равен $Q(q)$. Таким образом, для разных q порядки чисел $b(q)$ взаимно просты и равны $q_i^{m_i}$.

Следовательно, порядок числа $b = \prod_q b(q)$ равен $\prod_{i=1}^{k} q_i^{m_i} = n-1$.

13.2.4. Понятие о последовательностях Люка. (n+1) - критерий Люка

Тестирование с помощью теста Ферма можно рассматривать как поиск и анализ членов последовательностей вида $f_{n-1}(a) = a^{n-1} - 1$, делящихся на n.

Аналогичный подход может быть реализован для других видов последовательностей, в частности, рекуррентных. Рассмотрим общую идею $(n+1)$- критерия Люка, в котором для тестирования чисел на простоту используется знание простых делителей числа $(n+1)$.

Пусть $n > 1$ - нечетное число, с которым целые числа p и q связаны условием: $D = p^2 - 4q$ - квадратичный невычет по модулю n.

Построим две рекуррентные последовательности вычетов по модулю n, которые называются последовательностями Люка, ассоциированными с p и q:

$$U_0 = 0,\ U_1 = 1,\ U_{k+2} = pU_{k+1} - qU_k;$$

$$V_0 = 2,\ V_1 = p,\ V_{k+2} = pV_{k+1} - qV_k,\ k \geq 0.$$

Существуют различные соотношения, позволяющие быстро вычислять элементы последовательностей Люка с большими номерами.

Например, $U_{2k} = U_k V_k,\ V_{2k} = V_k^2 - 2q^k$.

Оказывается, для этих последовательностей можно сформулировать утверждения, аналогичные малой теореме Ферма.

Например, последовательность $\{U_k\}$ обладает следующим свойством: если n - простое, то $U_{n+1}(p,q)=0(n)$. Аналогия с условием Ферма $f_{n-1}(a)=0(n)$ очевидна.

Так называемый $(n+1)$- критерий Люка проверки чисел на простоту формулируется следующим образом.

Теорема. Пусть $n>1$ – нечетное число. Если для любого простого делителя r числа $n+1$ существуют такие простые числа p и q, что $D=p^2-4q$ - квадратичный невычет по модулю n и такие, что $U_{n+1}(p,q)=0(n)$, $U_{(n+1)/r}(p,q)\neq 0(n)$, то n– простое.

Данная теорема играет роль, аналогичную $(n-1)$- критерию Люка.

13.3. (P+1) – метод факторизации Вильямса

Пусть n - составное число и p - его неизвестный простой делитель. На основе последовательностей Люка можно построить метод факторизации числа n, аналогичный $(p-1)$ - методу Полларда, если известно, что число $p+1$ является гладким [49].

Это связано с рядом свойств последовательности Люка $U_1(P,Q),U_2(P,Q),\ldots$, ассоциированной с числами P и Q, где $D=P^2-4Q\neq 0(n)$.

Прежде всего, последовательность Люка сохраняет свойство делимости: если $a|b$, то $U_a|U_b$. Более того, следующие свойства показывают, что делители числа U_b ограничивают возможные значения индекса b.

Допустим, $M \mid U_b$. Не исключено, что число M делит также один или несколько элементов последовательности Люка с номерами, меньшими b.

Пусть $\omega = \omega(M)$ - наименьший такой номер. Назовем ω рангом появления M в последовательности Люка. Оказывается, если $M \mid U_b$, то $\omega \mid b$.

Далее. Если p - простое, причем НОД$(p, 2DQ) = 1$, то $p \mid U_{\Phi(p)}$, где $\Phi(p)$ выражается через символ Лежандра: $\Phi(p) = p - \left(\frac{D}{p}\right)$.

В случае, когда D является квадратичным невычетом, $\Phi(p) = p + 1$ и $\omega(p) \mid (p+1)$. Иными словами, делители $(p+1)$ могут оказаться индексами элементов последовательности Люка, которые делятся на p.

Таким образом, как и в $(p-1)$ - методе Полларда, полагая число $(p+1)$ гладким по отношению к некоторой фактор-базе, можно пытаться строить числа k, делящиеся на $\omega(p)$, например, кратные $(p+1)$.

В этом случае $p \mid U_k$, следовательно, при $U_k \neq 0(n)$ можно найти нетривиальный делитель n_1 числа n в виде $n_1 = \text{НОД}(U_k, n)$.

Если $U_k = 0(n)$, то необходимо повторить вычисления для новых значений P, Q, для которых значение $\left(\frac{D}{p}\right) = -1$.

Глава 14.
ТЕСТЫ СОЛОВЕЯ-ШТРАССЕНА И РАБИНА-МИЛЛЕРА ПРОВЕРКИ ЧИСЕЛ НА ПРОСТОТУ

Основное в этой главе…

- **Тест Соловея-Штрассена……..…176**
- **Тест Рабина-Миллера……178**

При тестировании чисел на простоту с помощью вероятностного теста, основанного на малой теореме Ферма, может возникнуть ситуация, когда вероятность ошибки не снижается с количеством повторений теста. В подобном случае упомянутая вероятность равна единице и в результате тестирования может быть принято неверное решение. В этой связи разработаны и применяются на практике вероятностные тесты, свободные от указанного недостатка.

Примерами таких тестов являются тест Соловея-Штрассена [50] и тест Рабина-Миллера [51,52] проверки чисел на простоту. См. также [8,15,16].

14.1. Тест Соловея-Штрассена

На основании следующей теоремы в тесте Соловея-Штрассена используется критерий Эйлера для определения значения символа Лежандра (квадратичного характера числа по простому модулю).

В самом тесте, естественно, вычисляется символ Якоби по модулю n.

Теорема [16]. Нечетное целое число *n* является простым тогда и только тогда, когда для всех чисел *a*: $1 \le a \le n-1$ выполняется сравнение вида $a^{\frac{n-1}{2}} = \left(\frac{a}{n}\right) \bmod n$.

Назовем указанное сравнение соотношением Эйлера.

Доказательство. Если *n* простое, то данное сравнение, очевидно, выполнено в силу свойств символа Лежандра.

Пусть теперь $a^{\frac{n-1}{2}} = \left(\frac{a}{n}\right) \bmod n$, $1 \le a \le n-1$, но n составное.

В любом случае, $a^{n-1} = \left(a^{\frac{n-1}{2}}\right)^2 = \left(\frac{a}{n}\right)^2 = 1 \bmod n$, поэтому n - число Кармайкла, т.е. $n = p_1 p_2 \dots p_k$, $k > 2$, $p_i \ne p_j$ для $i \ne j$ и $\forall i \ \varphi(p_i) | (n-1)$.

Выберем элемент b, не являющийся квадратичным вычетом по модулю p_1. По китайской теореме об остатках, существует число *a*, удовлетворяющее системе сравнений: $a = b(p_1)$, $a = 1(p_2)$, …, $a = 1(p_k)$.

По определению символа Якоби, $\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right) \cdots \left(\frac{a}{p_k}\right) = \left(\frac{a}{p_1}\right) = \left(\frac{b}{p_1}\right) = -1$,

следовательно, по условию, $a^{\frac{n-1}{2}} = \left(\frac{a}{n}\right) \bmod n = -1 \bmod n$.

Последнее сравнение выполняется по любому делителю *n*, следовательно, $a^{\frac{n-1}{2}} = -1 \bmod p_2$. Однако это не так, поскольку $(n-1)/2$ делится на $\varphi(p_2)$ - противоречие.

14.1.1. Эйлеровы псевдопростые числа

Составное число *n*, удовлетворяющее соотношению Эйлера, называется эйлеровым псевдопростым по основанию *a*.

Из доказанного следует, что составных чисел, которые были бы эйлеровыми псевдопростыми по любому основанию, не существует.

Следовательно, мы можем предложить следующий тест, аналогичный тесту Ферма.

Псевдослучайно выбираем вычет $a \in \{2, \ldots, n-1\}$, проверяем условие $(a, n) = 1$. Если условие не выполнено, значит, *n* - составное. Проверяем соотношение Эйлера. Если оно не выполняется, то число *n* - составное. Иначе, повторяем тест для другого значения *a*.

Если мы могли бы проверить соотношение Эйлера для всех *a*, то мы смогли бы точно определить, является ли число *n* простым. Но для больших *n* это невозможно. Поэтому необходимо оценить, как ведет себя вероятность ошибки при увеличении числа k повторений теста.

Это можно сделать, исходя из утверждения, аналогичного тому, которое мы рассматривали при анализе свойств псевдопростых чисел. Интересно, что эйлеровы псевдопростые являются псевдопростыми числами.

Теорема [16]. Пусть n - нечетное составное число. Тогда:

а) если n - эйлерово псевдопростое по основанию a, то оно псевдопростое по основанию a;

б) если n - эйлерово псевдопростое по основаниям a и b, то n - эйлерово псевдопростое по основаниям ab и $ab^{-1}(\mathrm{mod}\, n)$;

в) множество $E_n = \left\{ a \in Z_n : a^{(n-1)/2} = \left(\frac{a}{n}\right)(\mathrm{mod}\, n) \right\}$ является подгруппой группы $F_n = \left\{ a \in Z_n : a^{n-1} = 1(\mathrm{mod}\, n) \right\}$;

г) если n не является эйлеровым псевдопростым по основанию a, хотя бы для одного числа a, то $|E_n| \le (1/2)|Z_n^*|$.

Действительно, свойства а), б) и в) следуют из мультипликативности символа Якоби и того факта, что его значение равно ± 1.

Докажем г): $|E_n| \le |F_n| \le (1/2)|Z_n^*|$.

Таким образом, при повторении теста Соловея-Штрассена k раз, вероятность неотбраковки составного числа $\le (1/2)^k$.

14.2. Тест Рабина-Миллера

Еще более эффективным тестом является тест Рабина-Миллера, в котором используется критерий, в конечном счете основанный на факте, что для простого модуля квадратными корнями из единицы являются лишь числа ± 1, а для составного нечетного модуля $n = uv$, $(u, v) = 1$, число таких корней больше двух.

Пусть n - нечетное натуральное число. Тогда можно записать $n-1=2^s t$, где t - нечетное и $s \geq 1$.

Если число n - простое, то $a^{n-1}=1(n)$, при $(a,n)=1$. Поэтому квадратные корни из единицы имеют вид: $a^{(n-1)/2}=\pm 1(n)$, где показатель равен $2^{s-1}t$.

Это означает, что в ряду вычетов по модулю n чисел a^t, a^{2t},… $a^{2^{s-1}t}$ либо появится $-1(n)$, либо все они сравнимы с единицей, т.е. $a^t=1(n)$.

Если n - составное, то возможны и другие случаи.

Основанный на данном замечании тест Рабина-Миллера заключается в следующем:

1) псевдослучайно выбираем вычет $a \in \{2,\ldots,n-1\}$, проверяем условие $(a,n)=1$. Если условие не выполнено, значит, n - составное и работа закончена;

2) вычисляем $a^t(\mathrm{mod}\, n)$. Если $a^t=\pm 1(\mathrm{mod}\, n)$, то не исключено, что число n - простое и необходимо перейти на начало, чтобы повторить тест для другого основания;

3) вычисляем последовательно вычеты чисел a^{2t},… $a^{2^{s-1}t}$ по модулю n, пока не появится (-1), либо не исчерпается список;

4) если (-1) обнаружена в списке, то не исключено, что число n - простое и необходимо перейти на начало, чтобы повторить тест для другого основания;

5) если ни одно число из списка не сравнимо с (-1), то число n - составное и необходимо закончить работу.

Как и для ранее рассмотренных тестов, существуют составные числа n, которые для соответствующих оснований a проходят тест.

14.2.1. Сильно псевдопростые числа

Назовем число $n = 2^s t + 1$, где $s \geq 1$, t - нечетно, сильно псевдопростым по основанию $a > 1$, если выполняется одно из двух условий: $a^t = \pm 1(n)$, либо в последовательности $a^{2t}, \ldots a^{2^{s-1}t}$ существует число, сравнимое с –1 по модулю *n*.

Оказывается, можно показать, что для любого $a > 1$ существует бесконечно много сильно псевдопростых чисел *n* по основанию *a*. Пример: $a = 7$, $n = 25$; *a*=5, *n*=781.

Можно доказать следующие основные свойства сильно псевдопростых чисел [15]:

1) число *n* сильно псевдопростое по основанию *a* является эйлеровым псевдопростым по тому же основанию;

2) если нечетное составное число *n* является сильно псевдопростым по основанию *a*, то общее количество оснований, по которому это число является сильно псевдопростым, не превышает $(n-1)/4$.

Таким образом, при повторении теста Рабина-Миллера k раз вероятность неотбраковки составного числа $\leq (1/4)^k$.

Оказывается, количество повторений теста, достаточное для практических приложений, можно ограничить величиной $2\log_2^2 n$.

Интересно отметить, что простоту небольших простых чисел можно проверять, используя несколько заранее указанных оснований.

Примеры [16]: если *n*<1373653 - сильно псевдопростое по основаниям 2 и 3, то *n* - простое; если *n*<341550071728321 - сильно псевдопростое по основаниям 2, 3, 5, 7, 11, 13, 17, то *n* - простое.

Глава 15.
ПОСТРОЕНИЕ БОЛЬШИХ ПРОСТЫХ ЧИСЕЛ

Основное в этой главе…

- **Теорема Поклингтона……..……..183**
- **Обобщение критерия Люка……..184**
- **Детерминированный тест, основанный на теореме Димитко…….……185**

В отличие от вероятностных тестов, рассмотрим теперь такие способы проверки чисел на простоту, при применении которых можно утверждать, что проверяемые числа действительно являются простыми. Эти способы применяются для построения псевдослучайных простых чисел [16].

Общая схема в данном случае такова: выбирается псевдослучайное число, которое тестируется на простоту. Сам тест состоит в построении членов некоторой последовательности, которые связаны с тестируемым числом и имеют особенности, если тестируемое число – простое.

Поскольку аналогичные особенности могут возникать и при тестировании некоторых составных чисел, то требуется ряд дополнительных условий, которые являются достаточными, чтобы доказать простоту тестируемого числа. Подобные процедуры называются алгоритмами построения доказуемо простых чисел.

Заметим, что указанный подход предполагает, что при выборе большого случайного числа существует возможность организовать в его окрестности успешный поиск простых чисел. Данная возможность обеспечивается, например, т.н. теоремой Дирихле о простых числах в арифметической прогрессии.

Эта теорема утверждает, что в арифметической прогрессии вида $an+b$, $n=0,1,2\ldots$, где $(a,b)=1$, существует бесконечно много простых чисел.

15.1. Детерминированный тест, основанный на обобщенном критерии Люка

Основой для перехода от вероятностных тестов к ряду детерминированных тестов служит $(n-1)$-критерий Люка.

Напомним, что в соответствии с этим критерием натуральное число *n* является простым в том и только в том случае, если существует натуральное число *a*, обеспечивающее выполнение следующих условий для любого

нетривиального делителя q числа $n-1$: $a^{n-1}=1(n)$, $\forall q \,||\, (n-1)$ $a^{(n-1)/q} \neq 1(n)$.

Ранее мы показали, что в критерии Люка можно заменить a на $a(q)$, однако требование полной факторизации числа $n-1$ осталось.

Рассмотрим теперь ряд математических утверждений, которые можно применить для обоснования более оптимальных детерминированных тестов на простоту, основанных на знании не всех, а части простых делителей числа $n-1$.

Прежде всего, нам понадобится т.н. теорема Поклингтона о виде простых делителей некоторых целых чисел.

15.1.1. Теорема Поклингтона

Теорема (Поклингтон). Пусть $n=q^k R+1>1$, где q - простое, не делящее R и p - произвольный простой делитель n. Если существует целое a такое, что $a^{n-1}=1(n)$ и НОД$\left(a^{(n-1)/q}-1, n\right)=1$, то каждый простой делитель p числа n имеет вид $p=q^k r+1$, при некотором $r=r(p)$.

Замечание. Условие НОД$\left(a^{(n-1)/q}-1, n\right)=1$ эквивалентно условию $\forall p \,||\, n$ $a^{(n-1)/q} \neq 1(p)$. Последнее выражение удобнее при доказательстве теоремы, но, в отличие от первого, не может быть проверено без знания p.

Доказательство. Пусть p - простой делитель n. Тогда $a^{n-1}=1(p)$ и $a^{(n-1)/q} \neq 1(p)$. Если m - порядок числа a по модулю p, то $n-1=md$, где d - целое. Допустим, q делит d. В этом случае $(n-1)/q=m(d/q)$, где (d/q) - целое, следовательно, $a^{(n-1)/q}=1(p)$, что невозможно. Поскольку $n-1=md=q^k R$, то m делится на q^k. Однако m обязано делить число $p-1$. Следовательно, $p=q^k r+1$ при некотором $r=r(p)$.

15.1.2. Обобщение критерия Люка

Теорема Поклингтона позволяет обобщить критерий Люка на ситуацию, когда число n-1 факторизовано не полностью, скажем, n-1=FR, где простые сомножители F известны.

Теорема. Пусть $n = FR + 1 > 1$, где 0<R<F. Если для любого простого делителя q числа F существует целое $a = a(q)$, такое, что $a^{n-1} = 1(n)$ и НОД$\left(a^{(n-1)/q} - 1, n\right) = 1$, то число n - простое.

Проведем доказательство от противного. Пусть n составное, тогда существует его простой делитель p, такой, что $p \le \sqrt{n}$. Зафиксируем $q \,||\, F$.

Пусть k максимальное число с условием $q^k \,|\, F$, $k \ge 1$. Аналогично, пусть q^h - максимальная степень, делящая FR.

Из условия теоремы следует, что $a^{n-1} = 1(p)$ и $a^{(n-1)/q} \ne 1(p)$. Пусть m - порядок числа a по модулю p, и $n - 1 = q^h R_1$. Как и при доказательстве теоремы Поклингтона, получим, что число q^h делит m. Следовательно, число $Q(q) = q^k$ делит m.

Рассмотрим число $b = b(q)$ вида $b = a^{m/Q}(p)$. Очевидно, $Q = Q(q)$ - порядок числа b по модулю p. Кроме того, для делителей $q_1 \ne q_2$ факторизованной части F числа $Q(q_1)$ и $Q(q_2)$ взаимно просты.

Построим для каждого $q \,||\, F$ число $b(q)$. Произведение всех таких чисел обозначим через B. Поскольку порядок числа B по модулю p равен наименьшему общему кратному чисел $Q(q)$, то он равен F. Но порядок любого элемента по модулю p делит $p - 1$, поэтому $F \le p - 1$.

Следовательно, $p^2 \geq (F+1)^2 > (R+1)(F+1) > FR+1 = n$, т.е. $p > \sqrt{n}$, что невозможно. Теорема доказана.

Если $R > 1$, то, поскольку $n = FR+1 > 1$, неравенство $1 < R < F$ выполняется при $F > \sqrt{n}$. Поэтому предварительную частичную факторизацию числа $n-1$ следует проводить лишь до тех пор, пока $F \leq \sqrt{n}$.

15.2. Детерминированный тест, основанный на теореме Димитко

Рассмотрим еще один детерминированный тест, основой которого является т.н. теорема Димитко. В этой теореме вместо условия НОД$\left(a^{(n-1)/q}-1, n\right)=1$, используемого в обобщенном критерии Люка, используется условие $a^{(n-1)/q} \neq 1(n)$.

Лемма. Пусть $n = q^k R + 1 > 1$, где q - простое, не делящее R. Если существует целое *a* такое, что $a^{n-1} = 1(n)$ и $a^{(n-1)/q} \neq 1(n)$, то существует простой делитель p числа *n* вида $p = q^k r + 1$, при некотором $r = r(p)$.

Доказательство. Пусть $n = \prod_{i=1}^{m} p_i^{m_i}$. Поскольку $a^{(n-1)/q} \neq 1(n)$, то не все делители *n* делят $a^{(n-1)/q} - 1$. Поэтому, в силу китайской теоремы об остатках, существует i: $a^{n-1} = 1\left(p_i^{m_i}\right)$, но $a^{(n-1)/q} \neq 1\left(p_i^{m_i}\right)$. Следовательно, порядок t элемента *a* по модулю $p_i^{m_i}$ делит $n-1$ и не делит $(n-1)/q$. Поэтому $q^k \mid t$.

Известно, что группа классов вычетов, обратимых по модулю $p_i^{m_i}$, имеет первообразный элемент (циклична). Ее порядок равен $\varphi\left(p_i^{m_i}\right) = p_i^{m_i-1}(p_i-1)$.

Этот порядок должен делиться на t и, следовательно, на q^k. Поскольку q делит $n-1$, а p_i делит *n*, то q и p_i взаимно просты. Значит, $q^k \mid (p_i - 1)$, т.е. $p_i = p = q^k r + 1$.

Теорема (Димитко). Пусть $n = qR + 1 > 1$, где q - простое, R - четное и $R < 4(q+1)$. Если существует целое *a* такое, что $a^{n-1} = 1(n)$ и $a^{(n-1)/q} \neq 1(n)$, то число *n* - простое.

Доказательство. Пусть *n* - составное, $n = \prod_{i=1}^{m} p_i^{a_i}$. По условию, $n = q^h R_1 + 1$, где $h \geq 1$ и q не делит R_1. Лемма показывает, что существует i, такое, что $q \mid (p_i - 1)$. Таким образом, $n = 1(q)$ и $p_i = 1(q)$. Запишем *n* в виде $n = p_i u$. Очевидно, $u = 1(q)$, поэтому, $u = 1 + Kq$ и $p_i = 1 + Sq$.

Заметим, что $K, S \geq 2$. Действительно, *n* - нечетно, поскольку R - четное число. Если $K = 0$, то n - простое, что по допущению неверно. Если $K = 1$, то n - четно, т.к. u четно.

Если $S = 0$, то $p_i = 1$, что противоречит допущению. Если $S = 1$, то p_i - четно, что также невозможно. Итак, $u \geq 2q + 1$, $p_i \geq 2q + 1$, следовательно, $n = p_i u \geq (2q+1)^2 = 4q(q+1) + 1 > qR + 1 = n$ - противоречие.

Теорема Димитко позволяет строить большие доказуемо простые числа.

Пусть дано некоторое число q, о котором известно, что оно простое. Выберем R - четное число, разрядность которого равна разрядности q. В этом случае, очевидно, $R < 4(q+1)$.

Построим $n = qR + 1$. Если существует целое *a*, такое, что $a^{n-1} = 1(n)$ и $a^{(n-1)/q} \neq 1(n)$, то число *n* - простое по теореме Димитко. Разрядность *n* в два раза больше разрядности числа q.

Если мы в качестве q примем построенное число *n*, то можем построить следующее простое число, вдвое большей разрядности, чем *n* и т.д.

Данный подход лежит в основе алгоритма генерации простых чисел в межгосударственном стандарте на цифровую подпись ГОСТ 34.310-95, широко используемом в Украине.

Глава 16.
ВЫБОР ПАРАМЕТРОВ КРИПТОСИСТЕМЫ RSA

Основное в этой главе…

- **Общие требования к выбору параметров……....190**
- **Метод Гордона построения сильно простых чисел……192**

Ранее нами были рассмотрены различные частные случаи ослабления криптосистемы *RSA* с параметрами (e,d,n), $n=pq$, которые возникают вследствие ее неправильного использования, либо при специфических (слабых) значениях параметров.

Данная криптосистема широко распространена, так что вопрос о правильном выборе ее параметров является достаточно актуальным.

16.1. Общие требования к выбору параметров

Корректность параметров связана с оценкой стойкости системы и может быть определена лишь с точки зрения практической стойкости. Следовательно, корректные параметры должны быть построены так, чтобы минимизировать ущерб от известных подходов к ослаблению криптосистемы.

Исходя из этого соображения, рассмотрим наиболее общие требования к выбору чисел *p, q, e, d* [16].

Прежде всего, следует учитывать, что слабость одного из параметров практически не компенсируется усилением свойств других параметров.

Очевидно, число $n=pq$ должно быть большим. Числа *p, q* не должны содержаться в списках известных больших простых чисел, не должны быть слишком близки друг к другу, либо существенно различаться по величине. Они не должны быть построенными по детерминированным алгоритмам с небольшим числом известных вариантов начальных параметров или содержать закономерности в двоичной записи. В общем, *p* и *q* не должны отличаться от типичных представителей случайных простых чисел.

Аналогичными свойствами должны обладать параметры *e* и *d*.

Например, если секретный ключ *d* содержит в двоичной записи небольшое количество единиц, то номера мест этих единиц легко определить перебором.

Можно доказать, что при известном *d* существует возможность факторизации модуля.

Известно, что для чтения сообщений, зашифрованных криптосистемой *RSA,* достаточно знания некоторого кратного функции Эйлера от модуля, т.к. в этом случае можно вычислить ключ, криптоэквивалентный ключу *d*.

Заметим также, что при наличии $ord_n a$ легко получить *a* из сравнения $a^e = c(n)$. Достаточно возвести *c* в степень *h*, удовлетворяющую соотношению $eh = 1(ord_n a)$

Далее. Для любого *a*, взаимно простого с *n*, $ord_p a$ делит *p*-1, а $ord_q a$ делит *q*-1. Поэтому $ord_n a$ делит *G*=НОК$(p-1, q-1)$. Следовательно, для построения криптосистемы, вместо определения *d* из сравнения $ed = 1(\varphi(n))$, можно воспользоваться решением сравнения $ed_1 = 1(G)$.

Пусть g=НОД$(p-1, q-1)$. Тогда $Gg = \varphi(n)$. Очевидно, из соотношения $ed = 1(\varphi(n))$ следует $ed = 1(G)$, поэтому $d = d_1(G)$ и $d \neq d_1(\varphi(n))$. Этим условиям удовлетворяют ключи d_1, d_1+G, d_1+2G, …, $d_1+(g-1)G$, криптоэквивалентные, таким образом, ключу *d*. Следовательно, чем больше НОД$(p-1, q-1)$, тем больше криптоэквивалентных ключей, тем хуже для системы.

Очевидно, в наилучшем случае НОД$(p-1, q-1) = 2$, при этом $p = 2s+1$, $q = 2t+1$, где $(s, t) = 1$.

Чтобы исключить возможность применения $(p \pm 1)$-методов факторизации, необходимо потребовать, чтобы числа $p_1 = (p-1)/2$, $p_2 = (p+1)/2$, $q_1 = (q-1)/2$, $q_2 = (q+1)/2$ не разлагались в произведение степеней небольших простых чисел, т.е. чтобы они содержали в разложении большое простое число.

Требования, сформулированые Р. Ривестом в наиболее сильной форме, заключаются в том, чтобы числа p_1, p_2, q_1, q_2 были простыми, причем в разложении как $p_1 - 1$, так и $q_1 - 1$ содержалось большое простое число.

Заметим, кстати, что неизвестно, является ли множество простых чисел вида $p_1 = (p-1)/2$ бесконечным.

Для практики достаточно, чтобы существовал достаточно большой простой делитель числа $p-1$. Очевидно, такой делитель имеет вид $r = (p-1)/(2j)$.

Таким образом, мы должны выделить некоторый специфический класс простых чисел.

16.2. Метод Гордона построения сильно простых чисел

Определение. Простое число p называется сильно простым, если выполняются условия:

$$p = 1 (\mathrm{mod}\, r),\ p = -1 (\mathrm{mod}\, s),\ r = 1 (\mathrm{mod}\, t),$$

где *r, s, t* - большие простые числа.

Поскольку числа *p, r, s, t* - нечетные, то они представляются в виде $p = 1 + 2jr$, $p = -1 + 2ks$, $r = 1 + 2lt$. Кроме того, для наших целей, чем меньше числа j, k, l, тем лучше.

Существует т.н. метод Гордона для построения больших сильно простых чисел. Заметим, что при реализации метода машинный алгоритм требует отслеживания ряда особых ситуаций. Это связано с тем, что в методе неоднократно используются вероятностные процедуры построения промежуточных данных, а также применяется тестирование чисел на простоту.

Данные процедуры работают тем лучше, чем больше разрядность чисел, используемых в вычислениях.

Тем не менее, всегда необходимо предусматривать соответствующие действия в случае, если промежуточные данные на соответствующем шаге построить не удалось. Кроме того, как правило, необходимо контролировать разрядность промежуточных данных и результатов, поскольку для криптосистемы *RSA* требуется построение сильно простых чисел заданной разрядности.

Суть метода Гордона построения сильно простого числа p заключается в следующем [16,53].

1. Строим случайное простое число *s*, исходя из заранее выбранной для него разрядности. Для этого выбираем псевдослучайно число *x* разрядности *s* и с помощью метода пробных делений оставляем в промежутке $[x, x+\log_2 x]$ числа, не имеющие малых делителей. Среди оставшихся чисел с помощью тестов на простоту определяем простое число *s*.

2. Строим случайное простое число *t* аналогично построению числа *s*.

3. С помощью метода пробных делений и тестов на простоту, аналогично пункту 1, строим простое число $r = 1 + 2lt$, перебирая *l* в промежутке $[1, \log_2 t]$.

4. Вычисляем $u = u(r,s) = \left(s^{r-1} - r^{s-1}\right) \bmod rs$. Это удобно сделать с помощью китайской теоремы об остатках, т.к. $u = 1(r)$ и $u = -1(s)$.

Для дальнейшего заметим, что искомое число *p* должно удовлетворять тем же условиям: $u = p = 1(\bmod r)$, $u = p = -1(\bmod s)$.

5. Если число *u* - нечетное, то присваиваем $p_0 = u$, иначе, полагаем $p_0 = u + rs$.

6. Строим p - ближайшее простое число, сравнимое с нечетным числом p_0 по модулю rs, т.е. тестируем на простоту числа вида $p = p_0 + 2krs$, k=0, 1, …, пока не найдется простое число (либо сработают ограничения реализации).

Алгоритм основан на следующей теореме.

Теорема (Гордон). Если r, s - нечетные простые числа, то простое число p удовлетворяет условиям $p = -1 (\mathrm{mod}\, s)$, $p = 1 (\mathrm{mod}\, r)$, тогда и только тогда, когда оно представимо в виде $p = p_0 + 2krs$, где p_0 - нечетное число из пары, $u, u + rs$.

Действительно, пусть $p = p_0 + 2krs$, тогда $p = u = -1 (\mathrm{mod}\, s)$, $p = u = 1 (\mathrm{mod}\, r)$ и p - искомое простое число.

Покажем, что простое число p', не представимое в виде $p = p_0 + 2krs$, не удовлетворяет условию теоремы.

Действительно, предположим обратное. По условию, $p' = -1 (\mathrm{mod}\, s)$, $p' = 1 (\mathrm{mod}\, r)$, т.е. $p' = p (\mathrm{mod}\, s)$, $p' = p (\mathrm{mod}\, r)$. Поэтому $p' - p$ делится на rs и, естественно, на двойку, т.е. $p' = p (\mathrm{mod}\, 2rs) = p_0 (\mathrm{mod}\, 2rs)$, что противоречит допущению.

16.3. Пример построения сильно простого числа

1. Строим исходное случайное простое число s размером, скажем, в 6 битов. Выбираем псевдослучайно шестибитовое число: x = 46. В промежутке $[46, 46 + 5]$ определяем простое число s = 47.

2. Строим случайное простое число t, аналогично построению числа s. Пусть x = 25. В промежутке $[25, 25 + 4]$ определяем простое число t = 29.

3. Строим простое число $r = 1 + 2lt$, перебирая l в промежутке $[1, 4]$. Получаем r = 59.

4. Вычисляем $u(r,s)$, решая с помощью китайской теоремы об остатках систему: $u = 1(r)$, $u = -1(s)$.

Используя расширенный алгоритм Эвклида, получим соотношение $4 \cdot 59 - 5 \cdot 47 = 1$, откуда: $59^{-1} \bmod 47 = 4$, $47^{-1} \bmod 59 = -5$.

Следовательно,

$$u(r,s) = 1 \cdot 47 \cdot (47^{-1} \bmod 59) + (-1) \cdot 59 \cdot (59^{-1} \bmod 47) = -471 = 2302(2773).$$

5. Число $u(r,s)$ - четное, поэтому полагаем $p_0 = 2302 + 59 \cdot 47 = 5075$.

6. Строим простое число, сравнимое с p_0 по модулю 2773, тестируя на простоту числа вида $p = p_0 + 2 \cdot 2773k$. При $k = 0,1,2,3$ получаем соответственно: 5075, 10621, 11167, 21713. Лишь последнее число является простым.

Глава 17.
ОБЩИЕ СВЕДЕНИЯ ОБ ИНОСТРАННЫХ КРИПТОСРЕДСТВАХ

Основное в этой главе…

- **Основные принципы построения систем управления ключами……..…201**
- **Блочные шифры в смешанных криптосистемах……..……203**

Использование в Украине систем криптографической защиты информации, разработанных за рубежом, является следствием широкого распространения вычислительной техники и операционных систем иностранного производства. Наиболее часто используются программно реализованные средства криптографической защиты информации (КЗИ).

Краткую характеристику большинства коммерческих криптоалгоритмов можно найти в [18].

В целом, рынок иностранных криптосредств очень широк: от криптосистем индивидуального использования до криптосредств военного предназначения. Порядок приобретения и использования криптосредств регулируется национальным законодательством и международными соглашениями.

Средства КЗИ реализуются аппаратным, аппаратно-программным и программным способами.

17.1. Аппаратные криптосредства

Наиболее надежными криптосистемами являются системы, основанные на аппаратных средствах КЗИ. Аппаратно-программные и программные средства, с точки зрения криптографии, преимуществ перед аппаратными средствами КЗИ не имеют.

Аппаратные средства позволяют:

- реализовать лишь необходимые функции аппаратуры;
- максимально повысить скорость обработки данных;
- обеспечить надлежащую защиту от побочных электромагнитных излучений информации;
- реализовать требования по прочности изделия;
- обеспечить меры по защите от доступа к узлам аппаратуры, ключам и постоянной информации, хранящимся в электронных модулях.

- использовать модульный принцип компоновки криптосредств, позволяющий легко устранять неисправности, производить смену ключей, модификацию криптосхем, обеспечивать совместимость с различными системами связи;

- изготавливать отдельные экземпляры аппаратуры по индивидуальным заказам.

В аппаратной реализации для шифрования используются как блочные, так и потоковые шифры. Аппаратные средства шифрования называются шифраторами.

На рынке достаточно распространены шифраторы, предназначенные для организации смешанных криптосистем, использующих для шифрования гаммирование по модулю два, а для распространения ключей и организации связи – асимметричное шифрование и криптопротоколы.

В зависимости от принятой системы передачи информации, существуют шифраторы предварительного шифрования и канальные шифраторы.

При предварительном шифровании сообщение зашифровывается целиком. Передача его абоненту производится либо сразу после зашифрования, либо может быть отложена на некоторый срок.

Канальные шифраторы используются для организации постоянной работы системы передачи информации в защищенном режиме. Канал является защищенным, даже если в нем отсутствует информация. Данные шифруются и отправляются абоненту поэлементно, по мере появления.

Шифраторы, кроме того, разделяются на два типа по способу генерации гаммы.

Шифраторы с внутренним носителем шифра генерируют гамму в процессе шифрования сами, в зависимости от ключей.

Шифраторы, использующие последовательность гаммы, заготовленную заранее другими средствами, называются смесителями. Подобная гамма иногда называется внешней. Ключи смесителя указывают, какой конкретно участок внешней гаммы следует использовать в данный момент.

Одной из особенностей шифраторов иностранного производства является наличие многих сервисных и вспомогательных функций.

Эти функции могут быть использованы для тестирования шифратора перед началом работы, для обучения персонала, для перехода к работе в аварийном режиме, изменения конфигурации внешнего оборудования, например, для задания типа модема и т.д.

Шифратор позволяет выбрать форматирование документа при его распечатке у абонента, задать различные кодировки шифртекста для согласования с системами связи, обеспечить режим передачи криптограммы (циркулярный, групповой) и многое другое.

Как следствие, для передачи криптограммы в сети связи она должна быть соответствующим образом оформлена (скомпонована). Кроме шифртекста, в криптограмму обычно входит различная служебная информация, которая составляет так называемый заголовок.

Наличие тех или иных служебных данных обусловлено сервисными функциями, организацией шифрованной связи и используемыми системами шифрования.

В частности, служебные части криптограммы могут содержать данные, управляющие работой шифратора при вхождении в связь, установкой режимов и несекретную служебную информацию об используемых ключах.

17.2. Основные принципы построения систем управления ключами

Криптосистемы основаны на использовании ключей. Совокупность всех действующих в системе ключей называется ключевой информацией.

В каждой криптосистеме большое количество ключей должно быть сформировано и распределено между абонентами. При проведении соответствующих работ несанкционированный доступ к секретным ключам должен быть исключен.

Процесс обработки и передачи информации, включающий генерацию, хранение и распределение ключей, называется управлением ключами [5].

Генерация ключей в реальных системах производится с использованием специальных аппаратных и программных методов, для которых необходимо наличие т.н. случайного фактора. Например, имеются генераторы на основе белого радиошума, либо программные датчики псевдослучайных последовательностей.

Хранение ключей состоит в организации их учета, хранения и уничтожения (удаления). Обычно хранение происходит в базах данных. Секретные ключи никогда не хранятся в явном виде на носителе, который может быть считан или скопирован.

Ключевая информация должна храниться в перешифрованном виде. Ключи для зашифрования ключевой информации называются мастер-ключами.

Мастер-ключи могут использоваться также в процессе выработки элементов ключевой информации.

Важным условием безопасности системы является периодическое обновление как обычных ключей, так и мастер-ключей. Обновление ключей связано с третьей составляющей системы управления ключами – их распределением.

Распределение ключей, по сути, состоит в организации безопасного канала доставки элементов ключевой информации назначенным абонентам сети секретной связи. Распределение ключей должно выполняться скрытно, а также оперативно и своевременно. При этом необходимо исключить факторы, способные привести к ослаблению криптосистемы, например, любую связь между старыми и новыми ключами, в частности, повторение ключей.

17.2.1. Ключевые системы потоковых шифров

Множество ключевых элементов, порядок их использования и закон формирования ключей из ключевых элементов составляют ключевую систему шифра.

В потоковых шифрах в настоящее время распространены трех и двухуровневые ключевые системы. Для трехуровневой системы имеются три вида ключей: сетевой, долговременный и сеансовый. Для двухуровневой – сетевой ключ отсутствует.

Сетевой ключ является ключевым элементом, срок действия которого может быть неограниченным. Он заносится при изготовлении конкретной партии устройств для сети связи.

Долговременный ключ - это ключ, действующий в течение длительного промежутка времени и сменяемый, как правило, периодически.

Сеансовый ключ действует значительно более короткий интервал времени, чем долговременный. Обычно один сеансовый ключ используется на один сеанс связи, т.е. на группу сообщений.

Если свой сеансовый ключ вырабатывается на каждое сообщение, то он называется разовым ключом.

В настоящее время, чаще всего, сеансовые и разовые ключи совпадают. Суммарная длина ключевых элементов в потоковых шифрах составляет порядка 128-512 и более битов.

В смешанных криптосистемах, основанных на потоковых шифрах, распределение ключей упрощается, поскольку канал доставки сеансовых ключей защищен асимметричной криптосистемой.

17.3. Блочные шифры в смешанных криптосистемах

Кроме потоковых шифров, в смешанных криптосистемах для шифрования данных используются и блочные шифры. Количество блочных шифров, описанных в открытой литературе, исчисляется десятками. Они отличаются как по длине блоков, так и сложности реализации.

В современных блочных шифрах длина блока, число циклов, длина ключа часто являются параметрами.

17.3.1. Алгоритм RC5

Изящным криптоалгоритмом, используемым в некоторых смешанных криптосистемах, является алгоритм *RC5* [4, 54]. Алгоритм состоит из трех компонент: алгоритма расширения ключа, алгоритма зашифрования и алгоритма расшифрования.

Алгоритм *RC5* (автор алгоритма Р. Ривест) предполагает выполнение операций с данными длиной в одно слово, скажем, размером в w битов. Количество итераций алгоритма задается параметром $r \geq 16$. Длина ключа – переменная.

Алгоритм расширения ключа инициализирует некоторую таблицу S, состоящую из $t = 2(r+1)$ слов, с помощью секретного ключа пользователя.

Суть процедур зашифрования-расшифрования состоит в следующем.

Пусть входной блок состоит из двух w–битовых слов: $A||B$.

Сложение и вычитание производится по модулю 2^w, знак $\oplus$ означает поразрядное сложение слов по модулю два.

Операция $A<<B$ ($A>>B$) является циклическим сдвигом слова A влево (вправо) на число разрядов, записанное в слове B (фактически, на величину $B \bmod 2^w$).

Слова ключа при зашифровании блока $A||B$ используются парами.

Сначала A и B соответственно модифицируются первым и вторым словом ключа: $A = A + S[0]$, $B = B + S[1]$. Полученный в результате модификации блок подвергается преобразованию, состоящему из r итераций. При этом используются последовательно выбираемые r пар слов ключа, начиная со второй пары.

Алгоритм, записанный на псевдокоде, выглядит следующим образом.

for $i = 1$ to r do

$\{ A = ((A \oplus B) << B) + S[2i];$

$B = ((B \oplus A) << A) + S[2i+1];\}$

Результат - выходной блок в словах A и B.

Алгоритм расшифрования легко получается из зашифрования:

for $i = r$ downto 1 do

$\{ B = ((B - S[2i+1]) >> A) \oplus A;$

$A = ((A - S[2i]) >> B) \oplus B;\}$

$B = B - S[1];$

$A = A - S[0].$

Очевидна простота и легкость реализации блока шифрования-расшифрования данного алгоритма.

17.3.2. Смешанная криптосистема на основе алгоритмов RSA и IDEA

Симметричный блочный шифр IDEA (International Data Encryption Algorithm) был разработан швейцарскими криптологами С.Лэем и Д. Мэсси [55].

Длина ключа – 128 битов. Длина блока – 64 бита.

Шифр построен на основе концепции использования «несовместимых» алгебраических операций на парах 16-битовых подблоков.

Каждая из операций может рассматриваться как закон композиции в соответствующих алгебраических группах с одинаковым числом элементов.

В алгоритме используются три различные операции с 16-ти битовыми подблоками: побитовое сложение по модулю 2 двух 16-битовых подблоков; сложение целых чисел по модулю 2^{16}, где 16-битовые подблоки задают двоичное представление соответствующего целого числа; а также умножение целых чисел по модулю $2^{16}+1$, где 16-битовый подблок есть двоичное представление соответствующего целого числа, за исключением подблока из всех нулей, интерпретируемого как 2^{16}.

Признано, что алгоритм является стойким и трудным для анализа.

Шифр IDEA как одна из возможностей применяется для шифрования данных в системе защиты электронной почты PGP (Pretty Good Privacy) [4, гл.5].

Соответствующая криптосистема является смешанной.

Для зашифрования сообщений используются разовые симметричные ключи. Для перешифровки разового ключа как альтернатива может использоваться RSA. Зашифрованный разовый ключ включается в сообщение.

Кроме того, криптосистема RSA используется для перешифровки данных в ходе аутентификации сообщений, которая состоит в следующем:

- отправитель формирует сообщение (открытый текст);
- с помощью стандартной функции хэширования (SHA-1) получает хэш-код сообщения;
- хэш-код зашифровывается алгоритмом RSA с помощью личного ключа получателя, результат добавляется в начало сообщения;
- все сообщение зашифровывается симметричным шифром;

- получатель, после снятия симметричного шифра, расшифровывает хэш-код с помощью открытого ключа отправителя. Затем он повторяет вычисление хэш-кода открытого текста и сравнивает его с результатом расшифрования соответствующих данных из сообщения. Если хэш-коды совпадают, сообщение считается подлинным.

Интересно отметить, что, несмотря на стойкость асимметричной и симметричной компонент данной криптосистемы, в одной из ранних версий PGP была выявлена слабость, проявляющаяся с вероятностью, достаточной для реального дешифрования сообщений.

Как оказалось, в ряде случаев возникала возможность факторизации модуля вследствие неудачной генерации псевдослучайных простых чисел, используемых при формировании параметров криптосистемы RSA [56].

ЗАКЛЮЧЕНИЕ

Сегодня мы являемся свидетелями революции, пожалуй, самой важной в истории человечества. Появляются новые средства общения людей и, похоже, тем средствам, которым мы обязаны прежним революциям – печатному станку, телефону, телевидению, вычислительной машине, далеко до них по влиянию на общественную жизнь, промышленность, науку и экономику.

Жизнеспособность общества все в большей мере определяется уровнем развития информационной среды. Информация играет все более весомую роль в функционировании государственных и общественных институтов, в жизни каждого человека. В современных условиях сформировался новый вид трудовой деятельности, связанный с получением, распространением и хранением информации. Промышленное общество трансформируется в информационное.

Информатизация ведет к созданию единого мирового информационного пространства, к унификации информационных технологий различных стран.

Новые технологии сулят грандиозные перспективы. Вместе с тем, катастрофически возрастает цена потерь в случае нештатного функционирования или снижения надежности систем обработки и передачи информации. Следует признать, что в настоящих условиях все более зримо проявляется зависимость экономики от качества систем защиты информации, в том числе криптографических.

Вместе с тем, практика показывает, что развитие криптографического потенциала страны можно осуществить только собственными силами. Иными словами, в области криптографической защиты информации нельзя рассчитывать на существенную помощь извне.

Как Вы имели возможность убедиться, действующие криптосистемы следует рассматривать как потенциально ненадежные. Поэтому они должны разрабатываться специалистами-криптологами и проектироваться с использованием официальных стандартов.

Необходимо подчеркнуть, что обеспечению надежности отдельной криптосистемы способствует стратегия взаимодействия пользователей, направленная на согласованные действия по обеспечению надлежащего уровня криптологического обеспечения систем связи.

Важность указанного подхода объясняется тем, что при эксплуатации криптографических систем защиты информации возникают факторы, касающиеся каждого пользователя.

Например, очевидно, что коммерческие криптосистемы одного и того же производителя могут применяться различными абонентами.

Следовательно, необходимо сохранять в секрете обнаруженные в ходе эксплуатации эффективные подходы, а также недоработки, снижающие стойкость действующей криптосистемы, по крайней мере, пока не завершится ее замена или модификация. В данном случае наиболее важно, чтобы недостатки системы не были обнаружены злоумышленником.

Кроме того, каждому пользователю в ходе внедрения и эксплуатации криптосистемы следует учитывать необходимость ее соответствия современным стандартам, независимо, является ли унификация на данный момент действительно актуальной. Перечень примеров можно легко продолжить.

Мы все имеем возможность убедиться в том, что высокий уровень информатизации, успешное применение высоких информационных технологий в производстве и потреблении, достигается, прежде всего, в экономически развитых странах с достаточным образовательным уровнем населения и высоким научно-техническим потенциалом. Ситуация в отношении технологий, связанных с криптографическими методами защиты информации, полностью аналогична.

Таким образом, решить задачу построения эффективных систем криптографической защиты информации можно лишь на основе опережающего овладения технологиями, соответствующими современному этапу развития криптографии и криптоанализа.

ЛИТЕРАТУРА

1. *К.Шеннон* «Теория связи в секретных системах» // Работы по теории информации и кибернетике, - М.: Иностранная литература, 1963. - 442с.
2. Введение в криптографию / *Под общ. ред. В.В.Ященко*. 3-е изд., доп. - М.: МЦНМО: «ЧеРо», 2000. – 236с.
3. *Саломаа А.* Криптография с открытым ключом. - М.: Мир, 1996. – 204с.
4. *В. Столингс* Основы защиты сетей. Приложения и стандарты. - М.: Издательский дом «Вильямс», 2002. – 432с.
5. *Б. Шнайер* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. - М.: Изд-во «ТРИУМФ», 2002. - 816с.
6. *Аграновский А.В, Хади Р.А*. Практическая криптография: алгоритмы и их программирование. –М.: СОЛОН-Пресс, 2002. -256с.
7. *М.Н. Аршинов, Л.Е. Садовский.* Коды и Математика (рассказы о кодировании). - М.: «Наука», 1983. - 144с.
8. *Вербіцький О.В.* Вступ до криптографії. Львів: Видавництво науково-технічної літератури, 1998. - 248 с.
9. *Холоша А.А.* Некоторые вопросы генерации псевдослучайных последовательностей с помошью конечных автоматов. // Збірник наукових праць інституту проблем моделювання в енергетиці. - Вип. 1. - Львів: Світ, 1998. - С. 74-90.
10. *Мухачев В.А.* Об одном подходе к определению параметров аппроксимирующей трехчленной линейной рекуррентной последовательности. // Збірник наукових праць інституту проблем моделювання в енергетиці. - Вип. 1. - Львів: Світ, 1998. - С. 120-129.
11. *О.А.Логачев, А.А. Сальников, В.В. Ященко* «Криптографические свойства дискретных функций». Из материалов конференции «Московский университет и развитие криптографии в России». (МГУ, 17-18 октября 2002г., http://www/cryptography.ru/db/msg.html?mid=1169258).

12. *Холоша А.А.* Об одном подходе к анализу качества блока подстановки битовых векторов. // Збірник наукових праць ІПМЕ НАН України. Вип.2. Львів: Світ, 1998. – С.59-74.
13. ГОСТ 28147-89. Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. - М.: Изд-во стандартов, 1989. - 26 с.
14. *Виноградов И. М.* Основы теории чисел. —9-е изд., перераб. —М.: Наука, 1981. – 124с.
15. *Коблиц Н.* Курс теории чисел и криптографии. — М.: Научное издательство ТВП, 2001. – 106с.
16. *Черемушкин А.В.* Лекции по арифметическим алгоритмам в криптографии. – М.: МЦНМО, 2002.
17. *А.В. Бессалов, А.Б. Телиженко.* Криптосистемы на эллиптических кривых: Учеб. пособие. – К.: ІВЦ „Видавництво «Політехника»”, 2004. – 224с.
18. *Богуш В.М., Кривуца В.Г., Кудін А.М.* Інформаційна базпека: Термінологічний навчальний довідник /*за ред. Кривуци В.Г.* – К.:ООО Д.В.К., 2004. – 508с.
19. *ДСТУ 4145-2002.* Криптографічний захист інформації. Цифровий підпис, що грунтується на еліптичних кривих. – К.: Держ. Ком. України з питань технічного регулювання та споживчої політики, 2003. –31 с.
20. *К. Айерлэнд, М. Роузен.* Классическое введение в современную теорию чисел. М.: Мир, 1987. -144с.
21. *О.Н. Василенко.* Теоретико-числовые алгоритмы в криптографии. М.: МЦНМО, 2003. -328с.
22. *Мухачев В. А.* Особые случаи снижения стойкости асимметричных криптоалгоритмов. // Моделювання в енергетиці та інформаційні технології. Збірник наукових праць інституту проблем моделювання в енергетиці НАН України. вип.3. - Київ, 1999. - С.104-109.

23. *Кузнецов Г.В., Фомичов В.В., Сушко С.О., Фомичова Л.Я.* Математичі основи криптографі: навчальний посібник. – Дніпропетровськ: Національний гірничий університет, 2004. –ч.1. -391 с.

24. *Ф.Р. Гантмахер* «Теория Матриц». - М.: «Наука», 1983. - 576с.

25. *Хорошко В.А., Чекатков А.А.* Методы и средства защиты информации /Под редакцией Ю.С. Ковтанюка – К.: Изд. Юниор, 2003. – 504с.

26. *Мухачов В. А.* Зведення задачі відновлення спотвореної рекуренти до диофантового рівняння з обмеженнями. // Моделювання в енергетиці та інформаційні технологіїї. Збірник наукових праць інституту проблем моделювання в енергетиці НАН України. вип.2. - Київ, 1999. - С.49-54.

27. *Мухачев В.А.* Об использовании алгоритма ГОСТ 28147-89 для генерации псевдослучайных последовательностей. // Збірник наукових п.2. - Львів: Світ, 1998. - С.74-79.

28. *Мухачев В. А.* Выявление слабых долговременных ключей в шифрсредствах, использующих алгоритм ГОСТ 28147. // Збірник наукових праць інституту проблем моделювання в енергетиці НАН України. вип.15. - Київ, 2002. - С.33-40.

29. *Diffie W., Hellman M.E.* New Directions in Cryptography //IEEE Trans. Inform. Theory. – 1976. –Vol. IT-22. –P.644-654.

30. *R.A. Ruppel* Analysis and Design of Stream Ciphers. Berlin: Springer-Verlag, 1986. –250p.

31. *U.M. Maurer* Provable Security in Cryptography: Diss. ETH № 9260. -1990. - 120 p.

32. *J. Golic* On the Security of Nonlinear Filter Generators // Fast software Encryption – Third International Workshop, Cambridge, February 1996. Berlin: Springer-Verlag, 1996, p.p.173-188.

33. *W.Meier and O.Staffelbach* Fast correlation attack on certain stream ciphers. //Journal of Cryptology. Vol. 1, no.3. 1989, p.p.159-176.

34. *J. Golic* On the Security of shift register based keystream generators // Fast software Encryption, Cambridge Security Workshop, December 1993. Berlin: Springer-Verlag, 1994, p.p.90-100.

35. *D.Gollman, W.G. Chambers* Clock-controlled shift registers: a review. // IEEE J. Selected Areas Commun., vol.7, p.p.525-533, May 1989.

36. *J. Golic, O'Connor* Embedding and probabilistic correlation attack on clock-controlled shift registers // Advances in Cryptology – Eurocrypt '94. - Berlin: Springer-Verlag, 1994, p.p.90-100.

37. *P.Camion, C. Carlet, P. Charpin, N. Sendrier* On correlation-immune functions. Lecture Notes in Computer Science vol. 576; Advances in Cryptology: CRYPTO '91 Proc. - Berlin: Springer-Verlag, 1991, p.p.90-100.

38. *J. Bierbrauer, K. Gopalakrishnan and D.R. Stinson* Bounds on resilient functions and orthogonal arrays. Advances in Cryptology: Proc. CRYPTO '94. Vol.839, LNCS, Berlin: Springer-Verlag, 1994, p.p.247-256.

39. *D.R. Stinson* Resilient functions and large sets of orthogonal arrays. Congressus numerantium, vol.92, 1993, p.p.105-110.

40. X.-M. Zhang and Y.Zheng On nonlinear resilient functions. IEEE Transactions on Information Theory, September 1997.

41. *Daniel Bleihenbacher* Generating ElGamal Signature Without knowing the Secret Key // Eurocrypt 96 Proceedings, LNCS 1070, Springer – Verlag, Berlin, 1996, p.p. 10-14.

42. *Rivest R.L., Shamir A., Adleman L.* A Method for Obtaining Digital Signatures and Public-Key Cryptosystems //Communs. ACM. - Feb. 1978. - Vol.21, No.2 - P.120-126.

43. *El Gamal T.* A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms // Advances in Cryptology –CRYPTO '84. – Berlin etc.: Springer-Verlag, 1985. – P.10-18. – (LNCS; 196).

44. Pohlig S., Hellman M.E. An Improved. Algorithm for Computing Logarithms and Its Cryptographic Significance //IEEE Trans. Inform. Theory. – 1978. –Vol. IT-24. –P.106-110.

45. Odlyzko A.M. Discrete logarithms in finite Fields and Their Cryptographic Significance // Advances in Cryptology –EUROCRYPT '84. – Berlin etc.: Springer-Verlag, 1985. – P.224-314. – (LNCS; 209).

46. *Satoh T., Araki K.* Fermat quotients and polynomial time discrete log algorithm for anomalous elliptic curves // Comment. Math. Univ. Sancti Pauli. 1998. V. 47. P. 81-92.

47. *Pollard J.M.* Theorems of Factorisations and Primality Testing //Proc. Cambr. Phil. Soc. – 1974. –Vol. 76. –P.521-528.

48. *Coopersmith D., Franklin M., Patarin J., Reiter M.* Low-Exponent RSA with Related Messages // Advances in Cryptology –EUROCRYPT '96. – Berlin etc.: Springer-Verlag, 1996. – P.1-9. – (LNCS; 1070).

49. *Williams H.C.* A *p+1* Method of Factoring //Math. Comput. -1982. –Vol. 39, No. 159. - P. 225-234.

50. *Solovey R., Strassen V.* A fast Monte-Carlo Test For Primality // SIAM J. Comput. –1977. -V. 6(1). - P.84-85.

51. *Miller G.L.* Riemann's hypothesis and tests for primality //J. Comput. System Sci. –1976. –V.13. –P.300-317.

52. *Rabin M.* Probabilistic algorithms for testing primality // Journal of Number Theory. –1980. –V.12 –P.128-138.

53. *Gordon J.* Strong Primes are Easy to Find // Advances in Cryptology – EUROCRYPT '84. – Berlin etc.: Springer-Verlag, 1985. – P.216-223. – (LNCS; 209).

54. *Rivest R.L.*The RC5 Encryption Algorithm // Fast Software Encryption 1994. Proc. 2nd Intern. Workshop. – Berlin etc.: Springer-Verlag, 1995. – P.86-96. – (LNCS; 1008).

55. *Lay X.*, Massy J.L. A Proposal for a New Block Encryption Standard // Advances in Cryptology – EUROCRYPT ’90. – Berlin etc.: Springer-Verlag, 1991. – P.389-404. – (LNCS; 473).

56. *Sakai Y., Sakyrai K., Ishizuka H*. On Weak RSA-Keys Produced from Pretty Good Privacy //Information and Communication Security – ICICS ’97. Berlin etc.: Springer-Verlag, 1997. – P.314-324. – (LNCS; 1334).

НАУЧНОЕ ИЗДАНИЕ

Владислав Андреевич Мухачев

Владимир Алексеевич Хорошко

МЕТОДЫ ПРАКТИЧЕСКОЙ КРИПТОГРАФИИ

Редактор *Орленко В.С.*

Техн. редактор *Чирков Д.В.*

Корректор *Мухачева Т.Н.*

Подписано к печати 26.08.2005. Формат 70x100 1/16
Бумага офсетная. Гарнитура «Таймс». Печать офсетная.
Усл. изд. л. 15.35. Уч. – изд. л. 13.43.
Тираж 500 экз. Заказ № 5-91

Отпечатано с готового оригинал-макета ООО «ПолиграфКонсалтинг»
03150, г. Киев, ул. Тельмана 5