Министерство образования Российской Федерации
Ростовский государственный университет

М.Г. АДИГЕЕВ

ВВЕДЕНИЕ В КРИПТОГРАФИЮ

Методические указания для студентов
механико-математического факультета

Часть 1

Основные понятия, задачи и методы криптографии

Ростов–на–Дону
2002 г.

Печатается по решению учебно-методической комиссии механико-математического факультета РГУ от

АННОТАЦИЯ

В данных методических указаниях рассмотрены базовые понятия и методы современной криптографии. При изложении основной упор был сделан на доступность материала.

Указания предназначены для студентов отделений «Прикладная математика» и «Защита информации» механико-математического факультета.

Автор: Адигеев М.Г.

ВВЕДЕНИЕ

В современном обществе все большую роль играют компьютеры, и вообще электронные средства передачи, хранения, и обработки информации.

Для того чтобы информационные технологии можно было использовать в различных областях, необходимо обеспечить их надежность и безопасность. Под безопасностью (в широком смысле) понимается способность информационной системы сохранять свою целостность и работоспособность при случайных или преднамеренных внешних воздействиях. Поэтому широкое использование информационных технологий привело к бурному развитию различных методов защиты информации, из которых основными можно, пожалуй, назвать, помехоустойчивое кодирование и криптографию.

Простейшие способы шифрования появились очень давно, однако научный подход к исследованию и разработке криптографических методов появился только в прошлом (двадцатом) веке. К настоящему времени криптография содержит множество результатов (теорем, алгоритмов), как фундаментальных, так и прикладных. Занятие криптографией невозможно без серьезной математической подготовки. Особенно необходимы знания в области дискретной математики, теории чисел, абстрактной алгебры и теории алгоритмов. Вместе с тем не следует забывать, что криптографические методы предназначены в первую очередь для практического применения, а теоретически стойкие алгоритмы могут оказаться незащищенными перед атаками, не предусмотренными математической моделью. Поэтому после анализа абстрактной математической модели всегда необходим анализ полученного алгоритма с учетом ситуации, в которой он будет использоваться на практике.

Данные методические указания рассчитаны на студентов механико-математического факультета, поэтому автор предполагает у читателей наличие определенной математической культуры. Однако, как было указано выше, в криптографии

особенно важным является четкое знание не только математики, но и «предметной области». Целью данных методических указаний является ознакомление студентов с основными понятиями, методами и задачами криптографии. При этом автор исходил из того, что читатели впервые сталкиваются с «настоящей» криптографией (детективные романы и фильмы, по понятным причинам, в расчет не берутся), поэтому основное внимание уделялось ***доступности*** материала, а не полноте и глубине изложения. Следовательно, данные методические указания ни в коем случае нельзя рассматривать как полноценный учебник или справочник, содержащий готовые к употреблению криптографические алгоритмы. Это именно *введение* в криптографию.

В первой части рассматриваются базовые криптографические понятия, схемы и алгоритмы. Детальному анализу более сложных криптографических схем и протоколов, вопросам криптоанализа и стойкости криптосистем будут посвящены последующие части методических указаний.

1 ПРЕДМЕТ КРИПТОГРАФИИ

Криптография — это наука о способах преобразования информации с целью ее защиты от незаконных пользователей. Методы решения противоположной задачи (взлом криптографической защиты) составляют предмет другой науки — *криптоанализа*. Вместе с тем, было бы неправильным разделять криптографию и криптоанализ. И криптография, и криптоанализ изучают одни и те же объекты, но с разных точек зрения. Поэтому они скорее являются двумя частями одной и той же науки (она называется «*криптология*»), а не независимыми дисциплинами. Изучать их тоже надо совместно, потому что невозможно серьезно заниматься криптографией (например, разрабатывать шифры), не изучив криптоанализ.

Таким образом, наш предмет правильнее было бы называть *криптологией*. Однако, учитывая сложившуюся традицию, всюду

в данных методических указаниях будет использоваться термин «криптография». Проблемы и методы криптоанализа сейчас будут затронуты только косвенно, подробное изложение этих методов планируется в другой части методических указаний.

1.1 Основные задачи криптографии

Криптография возникла как наука о методах шифрования, и долгое время именно шифрование (т.е. защита передаваемых или хранимых данных от несанкционированного чтения) оставалась единственной проблемой, изучаемой криптографией. Однако в последнее время, в связи с бурным развитием информационных технологий, возникло множество новых применений, напрямую не связанных с сокрытием секретной информации.

Необходимость применения криптографических методов вытекает из условий, в которых происходит хранение и обмен информацией. В современных информационных системах очень часто происходит обмен данными в коллективах, члены которых не доверяют друг другу. В качестве примеров можно привести подписание контрактов или других документов, финансовые операции, совместное принятие решений и т.п. В таких ситуациях необходимы средства, гарантирующие, что в процессе обмена или хранения информация не будет подвергнута искажениям, или не будет подменена целиком. Такую гарантию может дать только применение научно обоснованных криптографических методов.

Итак, целью применения криптографических методов является защита информационной системы от целенаправленных разрушающих воздействий (*атак*) со стороны *противника*. Способы защиты существенно зависят от ситуации: от какого рода угрозы необходимо защищаться, какими возможностями обладает противник.

Основные цели криптографии:

- Обеспечение ***конфиденциальности*** данных (предотвращение несанкционированного доступа к данным). Это одна из основных задач криптографии, для ее решения применяется *шифрование* данных, т.е. такое их преобразование, при котором прочитать их могут только законные пользователи, обладающие соответствующим *ключом*.
- Обеспечение ***целостности*** данных — гарантии того, что при передаче или хранении данные не были модифицированы пользователем, не имеющим на это права. Под модификацией понимается вставка, удаление или подмена информации, а также повторная пересылка перехваченного ранее текста.
- Обеспечение ***аутентификации***. Под аутентификацией понимается проверка подлинности субъектов (сторон при обмене данными, автора документов, и т.д.) или подлинности самой информации. Частным случаем аутентификации является *идентификация* — процедура доказательства субъектом того, что он действительно является именно тем, за кого себя выдает. Во многих случаях субъект X должен не просто доказать свои права, но сделать это так, чтобы проверяющий субъект (Y) не смог впоследствии сам использовать полученную информацию для того, чтобы выдать себя за X. Подобные доказательства называются «доказательствами с нулевым разглашением».
- Обеспечение *невозможности отказа от авторства* — предотвращение возможности отказа субъектов от совершенных ими действий (обычно — невозможности отказа от подписи под документом). Эта задача неотделима от двойственной — обеспечение невозможности приписывания авторства. Наиболее яркий пример ситуации, в которой стоит такая задача — подписание договора двумя или большим количеством лиц, не доверяющих друг другу. В такой ситуации все подписывающие стороны должны быть уверены в том, что в будущем, во-первых, ни один из подписавших не сможет отказаться от своей подписи и, во-

вторых, никто не сможет модифицировать, подменить или создать новый документ (договор) и утверждать, что именно этот документ был подписан.

Основным способом решения данной проблемы является использование *цифровой подписи*.

Помимо перечисленных основных задач можно назвать также электронное голосование, жеребьевку, разделение секрета (распределение секретной информации между несколькими субъектами таким образом, чтобы воспользоваться ей они могли только все вместе) и многое другое. Подробное описание криптографических приложений можно найти в [1–4]. В данных методических указаниях рассматривается только первая категория приложений криптографии — обеспечение конфиденциальности данных.

Замечание. Чем шифрование отличается от кодирования? Слова «*кодирование*» и «*шифрование*» часто используются как синонимы. Однако в современной прикладной математике (к которой можно отнести и криптографию) эти термины разделяются. Под *шифрованием* понимается такое преобразование текста (сообщения), в результате которого прочитать преобразованный текст может только тот, кто обладает специальным *ключом*. *Кодированием* называется любое преобразование данных из одной формы представления в другую. Таким образом, кроме шифрования, термин «кодирование» включает в себя также так называемое «помехоустойчивое кодирование» (преобразование текста, позволяющее восстанавливать его в случае сбоя при передаче или хранении), сжатие данных и т.п. В широком смысле, кодированием можно назвать также сканирование текста или изображения (информация преобразуется из визуального представления в цифровое), и даже ввод текстов с клавиатуры.

1.2 Модель криптографической системы

Простейшую модель криптографической системы можно изобразить так, как показано на рисунке (см. рис. 1). Таким образом, имеется некая информационная система, включающая двух или более *абонентов* (законных пользователей) и канал (или каналы), по которым абоненты могут обмениваться *сообщениями*. Имеется также возможность появления *противника*, т.е. незаконного пользователя. Противник может перехватывать сообщения, передаваемые абонентами друг другу.

Простейшая модель криптосистемы

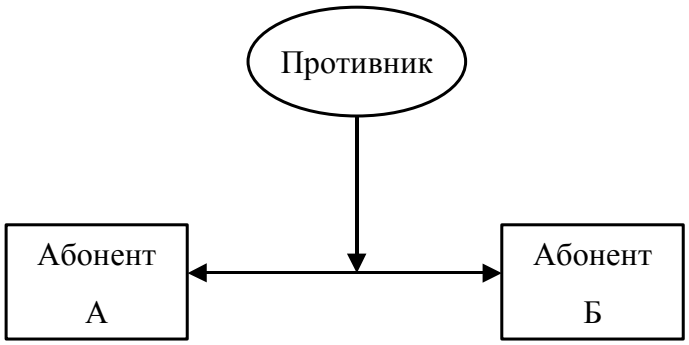

Рис. 1

Здесь необходимы следующие пояснения. Во-первых, противник может быть как *внешним* (т.е. не входить в число абонентов системы), так и *внутренним* (быть абонентом системы). В последнем случае этот абонент считается незаконным пользователем, если он пытается получить доступ к сообщениям, на которые не имеет права (например, конфиденциальные сообщения, которыми обмениваются другие абоненты).

Во-вторых, противник может перехватывать сообщения с разными целями — например, с целью разглашения перехватываемой информации (использование этой информации в своих целях или передача информации другому лицу), подмены

или имитации сообщения и т.д. Подобные цели называются *угрозами*. Для защиты от различных видов угроз необходимо применять различные криптографические методы. Рассматриваемая нами задача обеспечения конфиденциальности информации представляет собой задачу защиты от *угрозы разглашения*.

Наконец, следует иметь в виду, что описанная модель может применяться и в случаях, внешне отличных от обмена сообщениями. Например, при защите данных, хранящихся на компьютере, можно считать, что абонент А и абонент Б — одно и то же лицо, работающее с данными в разные моменты времени. В этом случае «каналом» является жесткий диск компьютера, на котором хранятся данные.

Итак, рассматривается модель, в которой противник имеет доступ к каналу передачи сообщений. Поэтому абонент, передающий сообщение (*отправитель*) должен перед отправкой преобразовать исходную информацию (*открытый текст*) в закрытый текст (который называется *шифртекстом, зашифрованным текстом или криптограммой*). Преобразование открытого текста в шифртекст называется *шифрованием* (часто используется также термин *зашифрование*). Абонент, получивший такой зашифрованный текст (*получатель*), с помощью обратного преобразования (*расшифрования*, *расшифровки*) восстанавливает исходный открытый текст.

Процедуры шифрования и расшифрования используют некоторые секретные данные, называемые *ключами*, причем в некоторых криптосистемах ключ шифрования и ключ расшифрования совпадают, а в других — различаются. Ключи известны только абонентам криптосистемы, причем для обмена данными с различными пользователями один и тот же абонент может использовать различные ключи.

Противник не знает ключ расшифрования, но может попытаться *вскрыть шифр*, т.е. либо подобрать ключ, либо преобразовать зашифрованный текст в открытый каким-либо другим способом. Методы вскрытия шифров называются

криптоанализом, а противник, применяющий эти методы — *криптоаналитиком*. Успех криптоанализа зависит как от свойств криптографической системы, так и от имеющихся у противника ресурсов (время, мощность вычислительных машин и т.п.). Способность шифра (криптосистемы) противостоять попыткам взлома (*атакам*) называется *стойкостью* шифра.

Существуют *абсолютно стойкие* системы шифрования, однако они очень не удобны и требуют больших затрат при использовании. Ни одна из широко используемых на практике систем шифрования не является абсолютно стойкой. Это означает, что если противник обладает неограниченными ресурсами и достаточно широкими возможностями для атаки (например, имеет доступ к некоторым открытым текстам и соответствующим им шифртекстам, полученным с использованием одного и того же ключа), то рано или поздно он сможет взломать шифр. Однако если выгода от использования полученной информации будет меньше, чем затраты на взлом, противник вряд ли будет этим заниматься. Поэтому при выборе алгоритма шифрования необходимо точно оценить соотношение ценности защищаемой информации, стойкости шифра и удобства его использования — иначе затраты на защиту информации могут превысить стоимость самой информации.

1.3 Формальная модель и классификация шифров

Введем формальное определение шифра и его составных частей [4]. Пусть T, C и K — конечные множества возможных открытых текстов, шифртекстов и ключей. Обычно каждое из этих множеств представляет собой множество *слов* в некотором *алфавите*, причем алфавиты открытых текстов, шифртекстов и ключей могут различаться. Для большинства современных систем шифрования открытые тексты, шифртексты и ключи представляют собой слова в алфавите {0,1}, т.е. последовательности нулей и единиц.

Процедура шифрования задает функцию E_k: $T \to C$, которая отображает множество открытых текстов во множество шифртекстов в зависимости от некоторого ключа $k \in K$. Аналогично, процедура расшифрования D_k: $C \to T$ также зависит от ключа k и отображает множество шифртекстов во множество открытых текстов. Так как получатель всегда должен иметь возможность по шифртексту восстановить исходный текст, то при любом k из K функции E_k и D_k должны удовлетворять условию: $D_k \circ E_k = I$, где I — тождественное отображение T в T.

Замечание 1. Во многих криптографических системах предполагается, что открытый текст, шифртекст и ключ — это целые числа. Такое предположение удобно для построения и обоснования алгоритмов шифрования и расшифрования, поскольку числовые функции хорошо изучены. Вместе с тем, это не ограничивает область применения таких алгоритмов, потому что любой текст, записанный с помощью букв, например, русского алфавита, всегда можно представить в виде целого числа. Обычно для этого каждый символ алфавита кодируют набором нулей и единиц (например, в соответствии с таблицей ASCII), и текст представляют в виде последовательности кодов соответствующих символов, записанных друг за другом. Получившаяся последовательность нулей и единиц является числовым представлением текста.

Замечание 2. Часто при реализации алгоритмов шифрования и расшифрования бывает удобно считать, что длина ключа, используемого для преобразования текста, равна длине самого текста или зависит от длины текста каким-то определенным образом. Очевидно, что если ключ используется для шифрования нескольких текстов, его длина не может зависеть от длины каждого конкретного текста. В

этом случае перед шифрованием текста поступают так: на основе данного *секретного ключа* фиксированной длины с помощью определенного алгоритма формируют *ключ шифрования*, имеющий необходимую длину, и именно полученный ключ шифрования используют для преобразования текста. Простейшим способом сформировать ключ шифрования нужной длины является периодическое повторение символов секретного ключа. Например, из секретного ключа $k = (k_1, k_2, \ldots, k_n)$ можно получить ключ шифрования $(k_1, k_2, \ldots, k_n, k_1, k_2, \ldots, k_n, k_1, k_2, \ldots, k_n, k_1\ldots)$ произвольной длины.

Современные системы шифрования можно разделить на два больших класса:

- Симметричные (одноключевые) системы — в них для шифрования и расшифрования текста используется один и тот же ключ k.
- Асимметричные (двухключевые) системы используют различные ключи для шифрования и расшифрования текста. Для таких систем ключ k можно представить в виде пары (k_e,k_d), где часть k_e используется для шифрования, а k_d — для расшифрования.

2 СИММЕТРИЧНЫЕ СИСТЕМЫ ШИФРОВАНИЯ

К *симметричным* системам шифрования относятся такие системы, в которых для шифрования и для расшифрования используется один и тот же ключ. Поэтому такие системы называют также *одноключевыми*.

В зависимости от типа преобразования, выполняемого над открытым текстом при шифровании, симметричные системы

шифрования можно разделить [4] на шифры *замены*, шифры *перестановки* и *композиционные* шифры.

К шифрам замены относятся преобразования, при которых фрагменты открытого текста (отдельные символы или группы символов — *блоки*) заменяются некоторыми символами или группами символов в шифртексте. Метод шифрования *гаммированием*, в принципе, также является разновидностью шифров замены. Но обычно гаммирование выделяют в отдельный тип шифрования, поскольку по многим практически важным параметрам оно отличается от «обычных» шифров замены.

Шифры перестановки для получения шифртекста лишь переставляют символы открытого текста местами.

Наиболее распространены *композиционные шифры*, представляющие собой последовательное применение нескольких процедур шифрования разных типов.

2.1 Шифры перестановки

Ключом шифра перестановки является перестановка номеров символов открытого текста. Это, в частности, означает, что длина ключа шифрования должна быть равна длине преобразуемого текста. Для того чтобы из секретного ключа получить ключ шифрования, удобный для использования в шифрах перестановки, предложен ряд методов. С помощью одного из таких методов формируются так называемые *маршрутные перестановки*. Открытый текст записывают в некоторую геометрическую фигуру (чаще всего — прямоугольник) по некоторой траектории, а затем, выписывая символы из этой фигуры по другой траектории, получают шифртекст.

Пример. Запишем фразу «это маршрутная перестановка» в прямоугольную таблицу размером 3×9, двигаясь по строкам, слева направо и пропуская пробелы (см. рис.2).

Пример маршрутной перестановки

э	т	о	м	а	р	ш	р	у
т	н	а	я	п	е	р	е	с
т	а	н	о	в	к	а		

Рис. 2

Для зашифрования текста выпишем из этой таблицы буквы, двигаясь по столбцам сверху вниз: эттт наоанмяоапврекршареус.

Из–за своей низкой стойкости, в современных криптографических системах шифры перестановки используются только как составная часть композиционных шифров.

2.2 Шифры замены

Простейшим из шифров замены является *одноалфавитная подставновка*, называемая также шифром *простой замены*. Ключом такого шифра является взаимно однозначное отображение (*подстановка*) F алфавита открытого текста (X) в алфавит шифртекста (Y): F: $X \leftrightarrow Y$. Зафиксируем нумерацию символов в алфавитах X и Y: $X = \{x_1, x_2, \ldots x_n\}$, $Y = \{y_1, y_2, \ldots y_n\}$. Тогда отображение F фактически задается перестановкой π порядка $n = |X| = |Y|$: при шифровании символ x_i открытого текста заменяется на символ $y_{\pi(i)}$ шифртекста. Эта перестановка может быть задана либо таблицей, либо с помощью формулы. При задании с помощью формулы значение $\pi(i)$ представляется в виде выражения, зависящего от i.

Пример. Типичным примером шифра замены является *шифр Цезаря*. Этот шифр реализует следующее преобразование текста, записанного с помощью латинского алфавита: каждая буква открытого текста заменяется буквой, стоящей на три позиции позже нее в алфавите (при этом алфавит считается

записанным по кругу, то есть после буквы ‘z’ идет буква ‘a’). Например, открытый текст ‘secret’ будет преобразован в ‘vhfuhw’.

Ключ для шифра Цезаря можно задать в виде следующей таблицы (см. рис. 3). В первой строке записаны буквы открытого текста, во второй — соответствующие им буквы шифртекста.

Ключ для шифра Цезаря

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

Рис. 3

Шифр Цезаря можно описать и в виде формулы. Для этого пронумеруем буквы латинского алфавита числами от 0 до 25: a = 0, b = 1, …, z = 25. Тогда правило замены можно описать следующим образом: буква с номером i заменяется на букву с номером $i+3 \pmod{26}$, где операция ‘mod 26’ означает вычисление остатка от деления на 26.

Разумеется, возможен обобщенный вариант шифра Цезаря, при котором буква с номером i заменяется на букву с номером $i+k \pmod{26}$. В этом случае ключом шифра является число k.

Еще больше обобщив этот метод, мы придем к семейству *аффинных шифров*. Для алфавита из n символов $\{a_1, a_2, \ldots, a_n\}$ аффинным шифром называется процедура, заменяющая входной символ a_i на символ a_j, где $j = k \cdot i + l \pmod{n}$.

Шифры простой замены в настоящее время не используются, поскольку их стойкость невелика. Методы взлома таких шифров основаны на анализе частотности отдельных символов и их комбинаций. Дело в том, что в любом языке различные буквы и комбинации из двух, трех или большего количества букв имеют характерные частоты повторений в текстах. Например, в текстах на русском языке чаще всего встречается буква ‘О’, затем, в порядке убывания частоты, идут

буквы ‘Е’ (считая, что ‘Е’ и ‘Ё’ — одна и та же буква), ‘А’, ‘И’, ‘Т’ и т.д. Для английского языка аналогичная последовательность самых частых букв: ‘E’, ‘T’, ‘A’, ‘I’, ‘N’. Самым частым символом в текстах является, однако, не буква, а символ пробела. В [4, Приложение 1] приведены полные таблицы частот для отдельных букв и для биграмм (пар букв).

Ясно, что при использовании шифра простой замены частота повторений зашифрованных символов в шифртексте совпадает с частотой повторений соответствующих исходных символов в открытом тексте. Это позволяет достаточно легко вскрыть такой шифр. Более тонкие характеристики (учет сочетаемости различных букв) позволяют даже автоматизировать процесс взлома.

Для того чтобы увеличить стойкость шифров замены, применяют *многоалфавитную замену*. Процедура шифрования для многоалфавитной замены включает набор подстановок $\{\pi_1, \pi_2, \ldots, \pi_m\}$ и функцию-*распределитель* $\psi(k,i)$, задающую последовательность применения подстановок π_i. При шифровании i-го символа открытого текста применяется подстановка с номером $\psi(k,i)$, где k — ключ шифрования.

Частным случаем многоалфавитной замены является шифр Виженера. Формально этот шифр можно описать следующим образом. В качестве ключа шифрования выберем набор из m целых чисел: $k = (k_1, k_2, \ldots, k_m)$. Процедуру преобразования открытого текста $t = (t_1, t_2, \ldots)$ в шифртекст $c = (c_1, c_2, \ldots)$ построим на основе обобщенного шифра Цезаря: $c_1 = t_1 + k_1 \pmod{26}$, $c_2 = t_2 + k_2 \pmod{26}$, и т.д. Когда будут использованы все m компонент ключа k, для шифрования $(m+1)$-й буквы снова возьмем k_1, и т.д. Фактически, в качестве ключа шифрования используется бесконечная последовательность, образованная периодическим повторением исходного набора: $k_1, k_2, \ldots, k_m, k_1, k_2, \ldots, k_m, k_1, k_2, \ldots$ Такую последовательность принято называть *гаммой* (подробнее см. в разделе «Гаммирование»).

Взломать шифр многоалфавитной замены немного сложнее, чем шифры простой замены, но тоже достаточно легко. Такой шифр на самом деле представляет собой одновременное применение m шифров простой замены (обобщенный шифр Цезаря), причем часть исходного текста, состоящая из букв t_i, t_{m+i}, t_{2m+i}, … шифруются с использованием «ключа» k_i (i=1, …, m).

Если известен период гаммы (т.е. число m), то к каждой такой части можно применить любой из методов взлома шифров простой замены. Если период гаммы не известен, то задача усложняется. Но и для этих случаев разработаны эффективные методы взлома (см., например, [4]). Эти методы позволяют с достаточной вероятностью определить период гаммы, после чего задача сводится к взлому шифра гаммирования с известным периодом.

Как было указано выше, основой для атак на шифры замены является анализ частот вхождений символов в шифртекст. Для того чтобы затруднить взлом шифра замены, можно попытаться скрыть частотные свойства исходного текста. Для этого надо, чтобы частоты появления разных символов в зашифрованном тексте совпадали. Такие шифры замены называются *гомофоническими*.

Простейшим вариантом гомофонического шифра является следующий. Предположим, что нам известны частоты вхождений символов в открытый текст. Пусть f_i — частота появления i–го символа в открытом текста (i — номер буквы в алфавите). Каждой букве t_i исходного алфавита (т.е. алфавита, с помощью которого записывается открытое сообщение) сопоставим подмножество F_i, содержащее f_i символов выходного алфавита (т.е. алфавита, с помощью которого записывается шифртекст), причем никакие два подмножества F_i и F_j не пересекаются. При шифровании будем заменять каждое вхождение символа t_i на случайный символ из множества F_i. Ясно, что средняя частота появления в шифртексте

любого из символов выходного алфавита одинакова, что существенно затрудняет криптоанализ.

2.3 Гаммирование

Формально гаммирование можно отнести к классу шифров многоалфавитной замены. Однако, благодаря удобству реализации и формального описания, шифры гаммирования широко используются, и обычно их выделяются в отдельный класс.

Суть метода гаммирования заключается в следующем. С помощью секретного ключа k генерируется последовательность символов $g=g_1g_2\ldots g_i\ldots$, эта последовательность называется *гаммой*. При шифровании гамма *накладывается* на открытый текст $t=t_1t_2\ldots t_i\ldots$, т.е. символы шифртекста получаются из соответствующих символов открытого текста и гаммы с помощью некоторой обратимой операции: $c_i=t_i \bullet g_i$, $i=1,2,\ldots$ В качестве обратимой операции обычно используется либо сложение по модулю количества букв в алфавите N:

$$c_i = t_i + g_i \pmod{N},$$

либо, при представлении символов открытого текста в виде двоичного кода, операция поразрядного суммирования по модулю 2 (операция ‘побитовый XOR’):

$$c_i = t_i \oplus g_i.$$

Расшифрование осуществляется применением к символам шифртекста и гаммы обратной операции: $t_i = c_i - g_i \pmod{N}$ или $t_i = c_i \oplus g_i$ (операция XOR является обратной к самой себе).

Стойкость систем шифрования, основанных на гаммировании, зависит от характеристик гаммы — ее длины и равномерности распределения вероятностей появления знаков гаммы.

Наиболее стойким является гаммирование с бесконечной равновероятной случайной гаммой, т.е. процедура шифрования, удовлетворяющая следующим трем условиям, каждое из которых является необходимым:

1) все символы гаммы полностью случайны и появляются в гамме с равными вероятностями;
2) длина гаммы равна длине открытого текста или превышает ее;
3) каждый ключ (гамма) используется для шифрования только одного текста, а потом уничтожается.

Такой шифр не может быть взломан в принципе, то есть является *абсолютно стойким* (см., например, отрывок из статьи К. Шеннона в приложении к [1]). Однако абсолютно стойкие шифры очень не удобны в использовании, и поэтому почти не применяются на практике. Обычно гамма либо получается периодическим повторением ключевой последовательности фиксированного размера, либо генерируется по некоторому правилу. Для генерации гаммы удобно использовать так называемые генераторы псевдослучайных чисел. Такие генераторы обычно основаны на рекуррентных математических формулах, использующих несколько ключевых (секретных) параметров.

Простейший генератор псевдослучайных чисел задается рекуррентной формулой:

$$g_i = a\ g_{i-1} + b \pmod{m}, \qquad (1)$$

где g_i — i-й член последовательности псевдослучайных чисел; a, b, m и g_0 — ключевые параметры. Данная последовательность состоит из целых чисел от 0 до m–1, и если элементы g_i и g_j совпадут, то последующие участки последовательности также совпадут: $g_{i+1} = g_{j+1}$, $g_{i+2} = g_{j+2}$, и т.д. Поэтому последовательность $\{g_i\}$ является периодической, и ее период не

превышает m. Для того чтобы период последовательности псевдослучайных чисел, сгенерированной по формуле (1), был максимальным (равным m), параметры формулы (1) должны удовлетворять следующим условиям (см. [5]):

- b и m —взаимно простые числа;
- a–1 делится на любой простой делитель числа m;
- a–1 кратно 4, если m кратно 4.

3 АСИММЕТРИЧНЫЕ КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ

3.1 Основы асимметричных систем

Одной из основных проблем при практическом использовании рассмотренных систем шифрования является проблема распределения секретных ключей между абонентами и проблема хранения этих ключей. Если в системе имеется N абонентов, то для обеспечения секретного обмена информацией между любыми двумя абонентами потребуется сгенерировать и распределить $N(N–1)/2$ секретных ключей, причем каждый абонент вынужден будет хранить N–1 секретный ключ для обмена информацией с остальными абонентами.

Решить проблему распределения ключей помогает использование *асимметричных* криптографических систем. В этих системах для обмена данными используются два ключа, один из которых является *секретным*, а другой — *открытым*, т.е. общедоступным. По этой причине асимметричные системы называются также *двухключевыми*.

Все асимметричные криптографические системы основаны на использовании односторонних функций с секретом.

Определение. Функция $F:X\to Y$ называется *односторонней*, если выполняются следующие два условия:

1) существует эффективный алгоритм, вычисляющий $F(x)$ для любого $x\in X$;

2) не существует эффективного алгоритма инвертирования функции F, т.е. алгоритма, позволяющего определить значение x по значению $F(x)$.

В данном определении «эффективным» называется *полиномиальный* алгоритм, т.е. алгоритм, который для получения результата для входа длины n тратит не более $P(n)$ шагов, где P — некоторый полином. Полное формальное определение сложности алгоритмов приведено, например, в [1] и [6].

В настоящее время теория алгоритмов не позволяет доказать *не существование* эффективных алгоритмов решения той или иной задачи. Поэтому, строго говоря, нам не известна ни одна односторонняя функция. Однако предложено несколько функций, которые *могут* оказаться односторонними — для этих функций в настоящее время, несмотря на интенсивные исследования, не известны эффективные алгоритмы инвертирования.

Наиболее часто используются «односторонние» функции, заимствованные из теории чисел:

- Функция $F(a, b) = ab$, то есть произведение двух чисел. Если a и b — простые числа, то по известному $c = ab$ можно однозначно определить a и b. Эта задача называется задачей *факторизации* числа. До сих пор не известен ни один полиномиальный алгоритм для решения задачи факторизации, хотя для вычисления произведения чисел (т.е. самой функции F) такие алгоритмы известны.

- Функция $F(a, n) = a^n \pmod M$, где a, n и M — целые числа. При известных a, n и M значение $b = a^n \pmod M$ может быть вычислено за полиномиальное количество шагов. Однако для обратной задачи — определить n по известным a, b и M

(задача *дискретного логарифмирования*) — полиномиальные алгоритмы, в общем случае, пока не известны.

Не любая односторонняя функция не может быть использована для шифрования. Действительно, если преобразовать открытый текст t с помощью односторонней функции: $c = F(t)$, то расшифровать полученный текст, то есть по c восстановить t, не сможет уже никто, в том числе и законный получатель. Для использования в криптографии необходимо, чтобы задача инвертирования шифрующего преобразования (т.е. вычисления t по $F(t)$) была разрешима за приемлемое время, но сделать это мог только тот, кто знает секретный ключ. Такие функции называются односторонними функциями с *секретом* (или с *потайным ходом*).

Определение. *Односторонняя функция с секретом* — это функция $F_k: X \to Y$, зависящая от параметра $k \in K$ (этот параметр называется *секретом*), для которой выполняются следующие условия:

1) при любом $k \in K$ существует эффективный алгоритм, вычисляющий $F_k(x)$ для любого $x \in X$;
2) при неизвестном k не существует эффективного алгоритма инвертирования функции F_k;
3) при известном k существует эффективный алгоритм инвертирования функции F_k.

Существование односторонних функций с секретом, как и существование односторонних функций, пока не доказано. Однако известны функции, которые могут оказаться односторонними функциями с секретом, если будет доказано, что инвертирование этих функций действительно является сложной задачей.

Рассмотрим в общем виде принцип использования односторонних функций с секретом для шифрования сообщений [2]. Каждый абонент криптосистемы выбирает некоторую одностороннюю функцию E_k с секретом k. Функции E_k всех абонентов заносятся в общедоступный справочник, но значение секрета k каждый абонент, как и следует из названия, держит в секрете. Если абонент B хочет переслать сообщение t абоненту A, он извлекает из справочника функцию E_k абонента A и с ее помощью вычисляет $c = E_k(t)$. Шифртекст c пересылается абоненту A, который по нему вычисляет исходное сообщение t, инвертировав функцию E_k с помощью секрета k. Расшифровать сообщение может только абонент A, поскольку кроме него никто не знает секрет k.

Замечание. Обычно функции шифрования E_k для разных абонентов вычисляются по одному и тому же заранее установленному алгоритму, но в зависимости от некоторого параметра p. У каждого абонента параметр p свой. Этот параметр называется *открытым* ключом данного абонента, поэтому асимметричные криптосистемы называют также криптосистемами с открытым ключом.

В качестве примера рассмотрим процедуру открытого распределения ключей и криптосистему RSA.

3.2 Процедура открытого распределения ключей

Как указывалось в начале раздела «3.1 Основы асимметричных систем», одной из наиболее сложных проблем в применении криптографических систем является распределение

секретных ключей между абонентами. При использовании классических, т.е. симметричных, систем шифрования, для распределения ключей необходимо устанавливать специальный канал, полностью защищенный от возможной атаки противника. Асимметричные криптографические системы позволяют распределять секретные ключи по открытым каналам, т.е. каналам, которые потенциально могут быть прослушаны противником. Такая процедура открытого распределения ключей была впервые опубликована в 1976 году в работе У. Диффи и М. Э. Хеллмана «Новые направления в криптографии».

В основе процедуры Диффи–Хеллмана лежит использование односторонней функции дискретного возведения в степень:

$$F(x) = g^x \pmod{p}, \qquad (2)$$

где x — целое число ($1 \le x \le p-1$), p — простое число, g — первообразный корень по модулю p.

Определение. *Первообразным корнем по модулю p* называется такое целое число g ($g < p$), для которого:

1) все степени $g^1 \pmod{p}$, $g^2 \pmod{p}$, …, $g^{p-1} \pmod{p}$ различны;
2) для любого целого числа a, такого что $1 \le a \le p-1$, найдется n, при котором $a = g^n \pmod{p}$.

Возводя число g в степени 1, 2, …, $p-1$(по модулю p), мы получим все числа от 1 до $p-1$, образующие $\mathbf{Z}_p^*$ (мультипликативную группу кольца $\mathbf{Z}_p$). Поэтому такое число g называется также *генератором* группы $\mathbf{Z}_p^*$.

Процедура Диффи–Хеллмана для открытого распределения ключей заключается в следующем. Для начала выбирается большое простое число p и число g — первообразный корень по

модулю p. Для обеспечения стойкости число p должно иметь длину, большую или равную 512 бит (здесь и далее *длиной* целого числа будем называть количество бит в двоичной записи этого числа), и разложение числа $p–1$ на множители должно содержать хотя бы один большой простой множитель (например, $p–1 = 2q$, где q — простое число). При таком выборе числа p в настоящее время не существует эффективного алгоритма для решения задачи инвертирования функции (2).

Каждый абонент в качестве своего *секретного* ключа выбирает некоторое случайное число x, по которому вычисляет свой *открытый* ключ $y = g^x \pmod p$. Все абоненты помещают свои открытые ключи в общедоступный справочник.

После этого, если два абонента, A и B, захотят обменяться секретным сообщением, они берут из общедоступного справочника открытые ключи друг друга (соответственно, y_A и y_B) и вычисляют общий секретный ключ:

1) абонент A вычисляет

$$z_A = (y_B)^{x_A} = (g^{x_B})^{x_A} \pmod p = g^{x_A x_B} \pmod p;$$

2) абонент B вычисляет

$$z_B = (y_A)^{x_B} = (g^{x_A})^{x_B} \pmod p = g^{x_A x_B} \pmod p.$$

Таким образом, после выполнения описанной процедуры у абонентов A и B есть общее число $z_A = z_B$. Это число они при обмене сообщениями могут использовать в качестве ключа для шифрования (например, методом *гаммирования*). Противник знает числа $y_A = g^{x_A} \pmod p$ и $y_B = g^{x_B} \pmod p$, но для того чтобы определить секретный ключ, ему необходимо решить задачу дискретного логарифмирования (по известным y_A и y_B вычислить x_A и x_B). Как уже отмечалось раньше, для этой задачи в настоящее время не существует эффективного алгоритма.

3.3 Криптосистема RSA

Криптосистема RSA названа так по первым буквам фамилий разработавших ее специалистов — Ривеста (R. Rivest), Шамира (A. Shamir) и Эдлмана (L. Adleman). Эта криптосистема может быть использована как для шифрования данных, так и для формирования цифровой подписи.

Опишем процедуру шифрования с помощью системы RSA, а затем дадим ее математическое обоснование. При использовании системы RSA каждый абонент формирует для себя секретный и открытый ключ следующим образом:

1) Выбирает два больших, неравных между собой, простых числа p и q, вычисляет $n = pq$ и $m = (p-1)(q-1)$.
2) Выбирает целое число e, такое, что $e<m$ и НОД(e, m) = 1 (то есть число e должно быть взаимно просто с m).
3) Вычисляет число d, удовлетворяющее условию: $ed = 1 \pmod m$.
4) Секретным ключом абонента является тройка чисел (p, q, d), открытым ключом — пара чисел (n, e).
5) Открытые ключи всех абонентов помещаются в общедоступный справочник.

Замечание. Существование числа d, удовлетворяющего условию шага 3, следует (подумайте, как?) из теоремы Евклида: для любых целых чисел a и b найдутся целые числа x и y, такие, что $ax + by$ = НОД(a,b). Расширенный алгоритм Евклида позволяет эффективно вычислить это число d. Алгоритмы для построения больших (длиной 512 и более бит) простых чисел p и q, и для нахождения числа e, удовлетворяющего условию шага 2, достаточно сложны для понимания. Их описание и обоснование можно найти, например в [1] и [7].

Функция шифрования сообщения, представленного в виде числа t ($t<n$) в системе RSA определяется формулой:

$$E(t) = t^e \pmod{n}.$$

Функция расшифрования (зависящая от секретного ключа) задается формулой:

$$D(c) = c^d \pmod{n}.$$

Длинное сообщение разбивается на блоки длиной $\log_2 n$ (чтобы каждый блок представлял собой число, меньшее n), каждый блок шифруется, и потом расшифровывается, отдельно.

Проверим, что функция $E(t)$ действительно является односторонней функцией с секретом. Свойство 1 из определения односторонней функции с секретом выполняется, поскольку для возведения в степень (в том числе и по определенному модулю) существуют эффективные алгоритмы.

Свойство 2 пока строго не доказано. Считается, что для инвертирования функции E необходимо определить число d, а для этого надо вычислить m, что невозможно без разложения числа n на простые множители p и q.

Для доказательства свойства 3 надо убедиться, что функция D действительно является обратной к функции E, то есть что для любого числа t выполняется $D(E(t)) = t$. Доказательство основано на теореме Эйлера из теории чисел: для любых взаимно простых целых чисел a и n выполняется соотношение:

$$a^{\varphi(n)} = 1 \pmod{n},$$

где $\varphi(n)$ — функция Эйлера, равная количеству целых чисел, больших 0 и меньших n, и взаимно простых с n. Для $n=pq$, где p и q — простые, $\varphi(n) = (p-1)(q-1)$, то есть $\varphi(n)$ в точности равно выбранному нами параметру m. Поскольку n равно произведению двух больших простых чисел, любое произвольно выбранное число t взаимно просто с n с вероятностью, практически равной 1.

Докажем, что функция D обратна к функции E:

$$D(E(t)) = D(t^{e} \pmod n) = (t^{e})^{d} \pmod n.$$

Числа e и d выбраны так, что выполняется условие $ed = 1 \pmod{\varphi(n)}$. Это равносильно тому, что существует целое число r, такое, что $ed = r\varphi(n) + 1$. Поэтому

$$(t^{e})^{d} \pmod n = t^{r\,\varphi(n) + 1} \pmod n = \left(t^{\varphi(n)}\right)^{r} \pmod n \cdot t \pmod n.$$

Воспользовавшись теоремой Эйлера, получим:

$$\left(t^{\varphi(n)}\right)^{r} \pmod n = 1^{r} \pmod n = 1.$$

Следовательно, мы доказали, что для любого t, меньшего n, выполняется: $D(E(t)) = t \pmod n = t$.

3.4 Особенности использования асимметричных криптосистем на практике

По эффективности и стойкости асимметричные (двухключевые) системы проигрывают симметричным (одноключевым) — при одинаковой длине ключа лучшие асимметричные процедуры шифрования работают медленнее и обеспечивают меньшую секретность, чем лучшие симметричные шифры. Поэтому обычно асимметричные системы используют для шифрования не самостоятельно, а в комплексе с симметричными системами. Например, шифрование данных с помощью симметричного (одноключевого) алгоритма A_1 и асимметричного (двухключевого) алгоритма A_2 выполняется в следующем порядке:

1) Генерируется случайный ключ k_1 для алгоритма A_1.
2) С помощью этого ключа производится шифрование данных: $c' = A_1(t, k_1)$.
3) Ключ k_1 шифруется с помощью алгоритма A_2 на открытом ключе k_p: $c'' = A_2(k_1, k_p)$.
4) Шифртекст представляет собой пару $c = (c', c'')$.

Для того чтобы расшифровать полученное сообщение (c', c''), получатель по c'' восстанавливает ключ k_1, с помощью которого затем по c' расшифровывает исходный текст. Поскольку длина ключа k_1 невелика по сравнению с длиной текста, подобная схема дает значительный выигрыш в скорости.

Замечание. Стойкость асимметричных систем основана на предположении о существовании односторонних функций. Это предположение пока не доказано, хотя и не опровергнуто. В результате развития теории алгоритмов могут быть получены эффективные методы решения задач, использующихся в асимметричных криптосистемах (например, задачи дискретного логарифмирования), и тогда такие системы перестанут быть стойкими, а все зашифрованные с их помощью документы смогут быть расшифрованы. Но даже если будет доказано отсутствие эффективных алгоритмов для решения таких задач, асимметричные системы все равно останутся только *вычислительно* стойкими, т.е. их взлом будет теоретически возможным, хотя и будет требовать больших временных и вычислительных ресурсов. Абсолютно стойкие системы шифрования есть только в классе симметричных систем.

Асимметричные системы нашли также широкое применение в криптографических протоколах, позволяя решать задачи, не сводящиеся к «классическому» шифрованию: цифровая подпись, аутентификация и т.д.

3.5 Способы увеличения стойкости шифров

Стойкость большинства из рассмотренных алгоритмов шифрования можно существенно повысить, модифицировав сам алгоритм или порядок его применения. Рассмотрим некоторые способы повышения стойкости шифров.

3.5.1 Сцепление блоков

Определение. Система шифрования называется *поточной*, если при шифровании символы исходного текста последовательно заменяются на символы шифртекста в соответствии с некоторым алгоритмом: $t_i = E_k(c_i)$. При *блочном* шифровании исходный текст разбивается на блоки, и алгоритм шифрования преобразует одновременно все символы каждого блока.

Из рассмотренных выше систем шифры замены и гаммирования относятся к поточным системам, а шифры перестановки и шифр RSA являются блочными.

Одним из существенных недостатков блочных шифров является то, что одинаковые блоки открытого сообщения они преобразуют в одинаковые блоки шифртекста. Ясно, что это понижает стойкость шифра — если к противнику попадет образец исходного текста вместе с соответствующим шифртекстом, то он сможет частично расшифровывать другие шифртексты, если в них будут встречаться такие же блоки. Одним из способов избавления от подобного недостатка является использование блочных шифров в режиме *сцепления блоков*. В этом режиме при шифровании очередного блока используются также предыдущие блоки открытого текста. Например, текущий блок открытого текста (T_i) суммируется побитово по модулю два с предыдущим блоком шифртекста (C_{i-1}), и к результату применяется алгоритм шифрования:

$$C_i = E_k(T_i \oplus C_{i-1}).$$

В качестве начального блока C_0 используется либо блок, состоящий только из нулей, либо произвольный случайный блок (в этом случае он включается в шифртекст).

Возможна также двойственная схема, при которой алгоритм шифрования применяется к предыдущему блоку

шифртекста, а затем берется побитовая сумма по модулю два с текущим блоком:

$$C_i = T_i \oplus E_k(C_{i-1}).$$

Применение описанных схем обеспечивает зависимость всех последующих блоков шифртекста от всех предыдущих блоков открытого текста. Поэтому изменение какого-то блока открытого текста приводит к изменению не только соответствующего блока шифртекста, но и всех последующих блоков шифртекста.

3.5.2 Добавление случайных данных

Еще одним эффективным способом затруднить криптоанализ шифра является добавление случайных данных к шифруемому тексту. Этот способ заключается в следующем. Перед началом шифрования текста T необходимо сгенерировать случайный блок данных R заранее определенной длины, и дописать его к тексту. Получившийся блок данных R|T, где знак ‘|’ означает конкатенацию (сцепление) двоичных наборов данных, преобразуется в шифртекст с помощью процедуры шифрования по ключу k: C = E_k(R|T). Для того чтобы расшифровать сообщение, получатель применяет к шифртексту процедуру расшифрования, получая некоторый набор данных B = D_k(C). Этот набор данных представляет собой конкатенацию R и T, и, поскольку длина блока R известна, исходный текст T однозначно восстанавливается из B.

Достоинством такого метода является то, что при шифровании одного и того же блока данных в разные моменты времени получаются различные блоки шифртекста. А это сильно затрудняет атаку на шифр.

3.5.3 Недетерминированные шифры

При оценке стойкости шифра обычно предполагается, что алгоритм шифрования известен лицу, пытающемуся взломать шифр. Это предположение основывается на том факте, что, с одной стороны, практически нереально удержать этот алгоритм в секрете, не ограничивая распространение программных средств шифрования, а с другой — оценка реальной стойкости шифра возможна только после открытого изучения алгоритма шифрования экспертами.

Очевидно, что знание алгоритма преобразования данных существенно облегчает криптоанализ. Для того чтобы этого избежать, используются *недетерминированные* (гибкие) шифры. В простейшем случае такие шифры включают в себя набор процедур F_1, F_2, … F_n и алгоритм, который по секретному ключу k формирует последовательность $a(1)$, $a(2)$, …, $a(i)$. Процедура шифрования текста T по ключу k заключается в применении к этому тексту процедур F_i в порядке, определяемом последовательностью $a(i)$:

$$C = E_k(T) = F_{a(i)}(\ldots(F_{a(2)}(F_{a(1)}(T))\ldots).$$

Таким образом, недетерминированный алгоритм шифрования состоит из известных процедур (что позволяет научно оценить стойкость шифра), но порядок применения этих процедур определяется секретным ключом и поэтому неизвестен криптоаналитику.

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

ЛИТЕРАТУРА

1. Введение в криптографию. / Под ред. В.В. Ященко. — 2-е изд., испр. — М.: МЦНМО: «ЧеРо», 1999. — 272 с.
2. Дориченко С.А., Ященко В.В. 25 этюдов о шифрах. — М.: «ТЕИС», 1994. — 69 с.
3. Молдовян А.А, Молдовян Н.А., Советов Б.Я. Криптография. — СПб.: «Лань», 2000. — 224 с.
4. Алферов А.П.,Зубов А.Ю.,Кузьмин А.С.,Черемушкин А.В. Основы криптографии: Учебное пособие. — М.: Гелиос АРВ, 2001. — 480 с.
5. Кнут Д. Искусство программирования для ЭВМ. Т.2. Получисленные алгоритмы. — М.: Мир, 1977. — 724 с.
6. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979. — 535 с.
7. A. J. Menezes, P. C. van Oorschot, S. A. Vanstone. Handbook of Applied Cryptography. — CRC Press, 1996. — 816 с.

САЙТЫ В ИНТЕРНЕТЕ

http://www.cryptography.ru/

http://algolist.manual.ru/defence/intro.php

http://algo.4u.ru

Содержание