Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра защиты информации

В.Ф. Голиков, А.В. Курилович

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

Учебно-методическое пособие
для студентов специальностей
«Сети телекоммуникаций» и «Защита информации
в телекоммуникациях»
всех форм обучения

В 2-х частях

Часть 1

Минск 2006

УДК 621.391.25 (075.8)
ББК 32.811 я 73
Г 60

Р е ц е н з е н т:
зав. каф. информационно-измерительной техники и технологий БНТУ,
д-р физ.-мат. наук, проф. И.Е. Зуйков

Голиков В.Ф.

Г 60 Криптографическая защита информации в телекоммуникационных системах: Учеб.-метод. пособие для студ. спец. «Сети телекоммуникаций» и «Защита информации в телекоммуникациях» всех форм обуч.: В 2 ч. Ч.1/ В.Ф. Голиков, А.В. Курилович. – Мн.: БГУИР, 2006. – 55 с.: ил.

ISBN 985-444-988-2 (ч. 1)

Рассмотрены основные методы защиты информации в телекоммуникационных системах: правовые, организационные и технические. Особое внимание уделено техническим методам и средствам защиты информации от несанкционированного доступа, среди которых наиболее эффективными являются криптографические методы.

Пособие предназначено для студентов высших учебных заведений, обучающихся по специальности «Защита информации», а также будет полезно студентам других специальностей, изучающим курс «Основы защиты информации».

УДК 621.391.25 (075.8)
ББК 32.811 я 73

ISBN 985-444-988-2 (ч.1)
ISBN 985-444-987-4

СОДЕРЖАНИЕ

1. ЗАЩИТА ИНФОРМАЦИИ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

1.1. Цели защиты информации. Основные понятия

Телекоммуникационные системы (ТКС), как и любая информационная система, работают в условиях воздействия многочисленных угроз безопасности информации, циркулирующей в них.

Согласно, целями защиты информации являются: предотвращение утечки, хищения, утраты, искажения, подделки, несанкционированных действий по уничтожению, модификации, копированию, блокированию документированной информации и иных форм незаконного вмешательства в информационные системы.

Рассмотрим основные понятия и термины науки о защите информации. Под информацией будем понимать сведения о лицах, предметах, фактах, событиях, явлениях и процессах. Информация может существовать в виде документа (бумажного), в виде физических полей и сигналов (электромагнитных, акустических, тепловых и т.д.), в виде биологических полей (память человека).

В дальнейшем будем рассматривать информацию в документированной (на бумаге, дискете и т.д.) форме, в форме физических полей (радиосигналы, акустические сигналы). Среду, в которой информация либо создается, либо передается, обрабатывается, хранится, будем называть информационным объектом.

Под безопасностью информационной системы (ИС) будем понимать ее защищенность от случайного или преднамеренного вмешательства в нормальный процесс его функционирования.

Природа воздействия на ИС может быть двух видов:

- непреднамеренной (стихийные бедствия, отказы, ошибки персонала и т.д.);
- преднамеренной (действия злоумышленников).

Все воздействия могут привести к последствиям (ущербу) трех видов: нарушению конфиденциальности; нарушению целостности; нарушению доступности.

Нарушение конфиденциальности – нарушение свойства информации быть известной только определенным субъектам.

Нарушение целостности – несанкционированное изменение, искажение, уничтожение информации.

Нарушение доступности (отказ в обслуживании) – нарушается доступ к информации, нарушается работоспособность системы, доступ в которую получил злоумышленник.

В отличие от разрешенного (санкционированного) доступа к информации в результате преднамеренных действий злоумышленник получает несанкционированный доступ (НСД). Суть НСД состоит в получении нарушителем доступа в систему в нарушении установленных

правил. Под угрозой информационной безопасности системы будем понимать возможные воздействия на нее, приводящие к ущербу. Некоторое свойство системы, делающее возможным возникновение и реализацию угрозы, будем называть уязвимостью. Действие злоумышленника, заключающееся в поиске и использовании той или иной уязвимости, будем называть атакой. Целью защиты ИС является противодействие угрозам безопасности.

Защищенная ИС – это система со средствами защиты, которые успешно и эффективно противостоят угрозам безопасности. Совокупность норм, правил, рекомендаций, регламентирующих работу средств защиты ИС от заданного множества угроз безопасности, называется политикой безопасности.

1.2. Телекоммуникационные системы как объекты защиты информации. Потенциальные угрозы безопасности информации в ТКС

Современные ТКС предназначены для обмена сообщениями и базируются на элементах компьютерных сетей. Для таких систем взаимодействие процессов регламентируется международным стандартом «Эталонная модель открытых систем».

Как и для других информационных систем, угрозы для ТКС можно классифицировать как угрозы конфиденциальности информации, целостности и доступности. Для ТКС, как систем с распределенной структурой, имеющей территориально разнесенные элементы, соединенные каналами связи все множество угроз условно можно разделить на следующие группы:

1. Несанкционированный доступ к информационным ресурсам сети (процессам, данным).
   1.1. Доступ несанкционированных пользователей.
   1.2. Несанкционированное расширение прав санкционированных пользователей.
   1.3. Несанкционированное соединение.
2. Вторжение в линии связи.
   2.1. Пассивное вторжение в линии связи.
   2.2. Активное вторжение в линии связи.

Несанкционированный доступ к информационным ресурсам сети нелегальных пользователей осуществляется с целью чтения, модификации, копирования, уничтожения данных или программ.

Несанкционированное расширение прав санкционированных пользователей очень активная угроза для ТКС. ТКС, как объект коллективного использования, доступна для большого количества пользователей для выполнения некого стандартного набора действий: создание, хранение, передача данных и сообщений. Поэтому всегда существует реальная угроза, что пользователь легально войдя в систему,

может попытаться производить в ней неразрешенные ему действия, пытаясь расширить свои полномочия, используя для этого слабости программно-аппаратных средств ТКС или с помощью специально внедренного программного продукта.

Несанкционированное соединение осуществляется нарушителем, выдавшим себя за санкционированного абонента, с другим санкционированным абонентом с целью получения от него важной информации или передачи ему ложной информации.

При пассивном вторжении нарушитель только наблюдает за сообщениями, передаваемыми по линиям связи, не нарушая их передачу. При этом нарушитель делает попытку читать передаваемые сообщения и анализировать трафик сообщений, передаваемых по каналам связи.

Анализ трафика может позволить ему выявить идентификаторы объектов сети, адреса, длину сообщений, время их отправления, частоту сеанса связи и т.д.

При активном вторжении нарушитель осуществляет различные манипуляции над сообщениями во время соединения. Сообщения могут быть модифицированы, уничтожены, задержаны, скопированы, изменен порядок их следования, введены в сеть ложные сообщения

1.3. Механизмы защиты ТКС

Для защиты от НСД используются процедуры управления доступом. С каждым субъектом (пользователем, процессом) связывается некоторая информация, однозначно, идентифицирующая его. Это может быть число, строка символов, алгоритм. Эта информация получила название идентификатор. Идентификаторы всех санкционированных субъектов регистрируются в сети. Поэтому при входе в сеть проверяется идентификатор. Если результат положителен, то далее проверяется подлинность субъекта (аутентификация). В результате этой процедуры субъект должен подтвердить, что он является тем, за кого себя выдает. Это происходит путем предъявления некоторой секретной информации (пароля), которую знает только санкционированный субъект. После процесса аутентификации субъекту предоставляются регламентированные полномочия.

При разнесенном взаимодействии двух субъектов через каналы передачи данных перечисленных мер может оказаться недостаточно, поэтому возникает проблема подтверждения подлинности соединения абонентов, она заключается в том, что:

- получатель должен быть уверен в истинности источника данных и истинности самих данных;
- отправитель должен быть уверен в доставке данных получателю и в истинности доставленных данных.

В первом случае используется технология электронной цифровой подписи (ЭЦП). Во втором, отправитель должен получить “расписку” в получении

(уведомление). Эта процедура реализуется также с помощью ЭЦП. При этом для доказательства отправителю истинности переданного сообщения, уведомление о получении включает модификацию ЭЦП полученного сообщения. При этом в силу свойств ЭЦП отправитель не может отрицать ни факта отправления сообщения, ни его содержания, а получатель не может отрицать ни факта получения сообщения, ни истинности его содержания.

При отправке сообщений по каналам электронной почты получатель не является активным в момент пересылки. Это означает, что:

- взаимоподтверждение подлинности невозможно;
- ЭЦП проверяется в более позднее время, когда получатель забирает сообщение;
- получение уведомления отправителя о доставке сообщения также откладывается.

Поэтому защита электронной почты требует специальных протоколов подтверждения подлинности отсроченных процедур.

Специфические проблемы, связанные с подтверждение передачи сообщений, возникают при организации телеконференций. Здесь для обеспечения взаимодействия участников также должны использоваться специальные протоколы.

Для всего рассмотренного предполагалось, что взаимодействуют взаимодоверяющие субъекты в недружественном сетевом окружении. То есть, считалось, что источник угроз по отношению к соединению находится вне его. Более сложно обеспечить защиту при передаче сообщений между недружелюбными субъектами, когда никто никому не доверяет. Это заставляет входить в соединение, обмениваясь минимумом (или одинаковой по важности) идентификационной информацией. Процедуры защиты, используемые при этом, получили название - подписание контракта, жеребьевки.

Для противодействия пассивным вторжениям в линии связи должна использоваться двухуровневая защита линий связи, при которой защищается как сам трафик, так и тексты сообщений.

Защита от активных вторжений сводится, по существу, к действиям, позволяющим устанавливать факт атаки на линию связи. И работа сети при активном воздействии может быть временно нарушена.

Из рассмотренных необходимых механизмов защиты ТКС следует, что система защиты ТКС должна решать следующие задачи:

- идентифицировать субъекты и подтверждать их подлинность;
- предоставлять полномочия и осуществлять контроль доступа;
- защищать данные при хранении и передаче по каналам связи;
- быть управляемой со стороны уполномоченного субъекта и недоступной для модификации для всех остальных.

В настоящее время все эти задачи решаются применением криптографии, электронной цифровой подписи, протоколов идентификации и аутентификации, межсетевых экранов. Коротко рассмотрим возможности и особенности перечисленных процедур.

Криптография (шифрование).

В ТКС для защиты информации в каналах связи используются два вида шифрования: линейное (канальное) и абонентское.

Линейное шифрование производится на выходе узла связи, расшифрование - на входе узла связи. При этом используются поточные шифры и между узлами поддерживается сплошной поток битов шифрованного текста. Шифруется и заголовок, и информационная часть пакетов. Все это затрудняет анализ потоков сообщений, так как адреса источника и получателя зашифрованы, непрерывность трафика обеспечивается передачей пустых (незначащих) сообщений. При линейном шифровании трафик уязвим только в узлах связи, так как там информация обрабатывается в открытом виде, поэтому узлы связи должны быть хорошо защищены.

При абонентском шифровании каждое сообщение шифруется в его источнике и расшифровывается только получателем, при этом используется ключ, известный только этой паре абонентов. Сообщение, зашифрованное абонентом, может еще подвергаться и линейному шифрованию. Таким образом, абонентское шифрование обеспечивает конфиденциальность передаваемого сообщения, линейное - конфиденциальность, защиту трафика от анализа.

Следует отметить, что при абонентском шифровании за счет использования особых режимов, попутно, решается и задача контроля целостности сообщения и автоматически подлинности сообщения и его источника, если используется симметричная криптосистема.

Электронная цифровая подпись.

Использование ЭЦП целесообразно для подтверждения подлинности передаваемого сообщения, его целостности, а также подлинности отправителя и получателя, для сообщений, содержание которых не содержит конфиденциальной информации, при большом количестве абонентов, участвующих в обмене сообщениями.

Идентификация и аутентификация субъектов для предоставления им доступа к ресурсам ТКС осуществляется на основании специальных протоколов.

2. ОСНОВЫ ПОСТРОЕНИЯ КРИПТОСИСТЕМ

Криптография – это наука о методах, алгоритмах, программных и аппаратных средствах преобразования информации в целях сокрытия ее содержания, предотвращения видоизменения или несанкционированного использования.

Исторически сложилось так, что криптография длительное время использовалась, исключительно, как средство обеспечения конфиденциальности сообщений. Областью применения криптографии была область защиты государственной тайны: в военной, дипломатической и разведывательной сферах. Поэтому, естественно, криптография находилась в руках спецслужб. Всякие упоминания об этой науке в открытой печати были запрещены, хотя работы велись полным ходом, огромное число специалистов трудилось в этой области: математики, инженеры, разведчики. Криптографическая империя СССР противостояла аналогичной империи США. Часть специалистов трудилась над созданием стойких криптоалгоритмов, другая часть - над раскрытием чужих криптосистем.

В конце 20-го века обстановка в сфере использования криптографии коренным образом меняется. Здесь несколько причин. Главная - бурное развитие вычислительной техники, появление на этой базе информационных технологий. Доступность информационных технологий широкому кругу коммерческих компаний и частным лицам породила потребность, во-первых, обеспечивать конфиденциальность той информации, которая циркулирует в ТКС, во-вторых, обеспечивать ряд функций, таких как аутентификация субъектов системы, целостность сообщений, истинность документов и т.д. Оказалось, что все это можно обеспечить, используя принципы криптографии.

Криптография, обслуживающая задачи управления, бизнеса, телекоммуникаций, получила название открытой. Открытые криптотехнологии (ЭЦП, идентификация и аутентификация, защита от НСД) становятся коммерческими продуктами и распространяются без особых ограничений. Платежные системы: банковские, индивидуальные на основе пластиковых карт, локальные и корпоративные компьютерные сети - вот далеко неполный перечень применения криптографических технологий.

Наряду с решением задач обеспечения конфиденциальности, целостности и доступности информации существует задача анализа стойкости используемых криптопреобразований. Эта задача решается наукой, называемой криптоанализ. Криптография и криптоанализ составляют науку - криптологию.

2.1. Общие принципы криптографической защиты информации

Обобщенная схема криптографической системы, обеспечивающей шифрование передаваемой информации, имеет вид (рис.2.1).

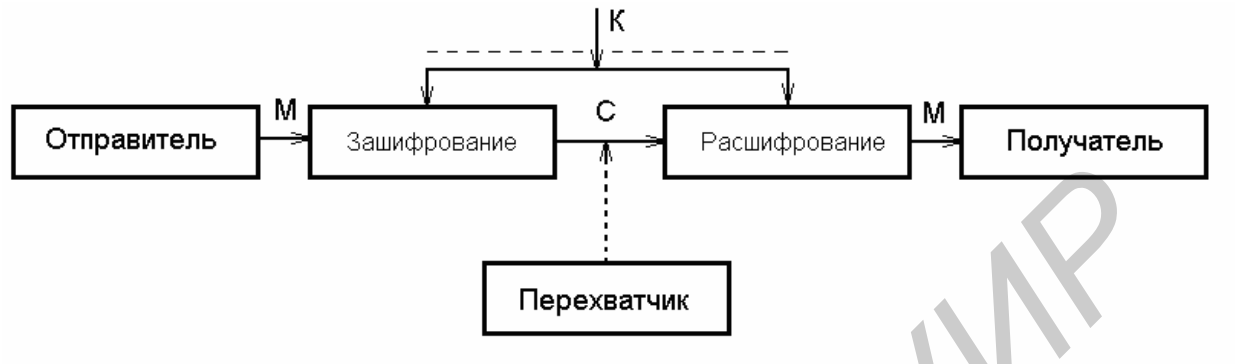

Рис. 2.1. Обобщенная схема симметричной криптографической системы

Отправитель генерирует открытый текст исходного сообщения M, которое должно передаваться по открытому каналу. Отправитель шифрует текст с помощью обратимого преобразования E и ключа K: E_K и получает шифротекст $C = E_K(M)$, который отправляет получателю. Получатель, приняв шифротекст C, расшифровывает его с помощью обратного преобразования $D = E_K^{-1}$ и получает исходное сообщение в виде открытого текста $M : M = D(C) = E_K^{-1}(E_K(M))$.

Преобразование E_K выбирается из семейства криптографических преобразований, называемых *криптоалгоритмами*. Параметр, с помощью которого выбирается конкретное преобразование, называется *криптографическим ключом* K. Система, в которой осуществляется шифрование и расшифрование сообщений, называется *криптосистемой.*

Формально криптосистема – это однопараметрическое семейство $(E_K)_{K \in \overline{K}}$ обратимых преобразований $E_K : \overline{M} \to \overline{C}$ из пространства $\overline{M}$ сообщений открытого текста в пространство $\overline{C}$ шифрованных текстов. Параметр K (ключ) выбирается из конечного множества $\overline{K}$, называемого пространством ключей. Криптосистема может иметь разные варианты реализации: набор инструкций; аппаратные или программные средства; аппаратно-программные средства.

Вообще говоря, преобразование шифрования может быть симметричным или асимметричным относительно преобразования расшифрования. Поэтому различают два класса криптосистем: симметричные криптосистемы и асимметричные криптосистемы. Иногда их называют: одноключевые (с секретным ключом) и двухключевые (с открытым ключом). Схема симметричной криптосистемы с одним секретным ключом K была показана на рис.2.1. Обобщенная схема асимметричной криптосистемы с двумя разными ключами K_1 и K_2 показана на рис.2.2.

В этой криптосистеме один из ключей является открытым K_1, а другой K_2 – секретным. Для этой криптосистемы $C = E_{K_1}(M)$, а $M = D_{K_2}(C) = E_{K_2}^{-1}\left(E_{K_1}(M)\right)$.

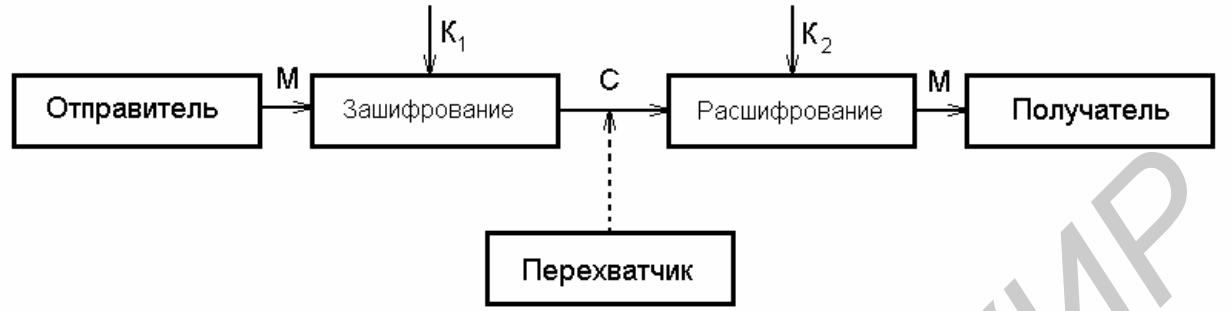

Рис. 2.2. Обобщенная схема асимметричной криптографической системы

В симметричной криптосистеме секретный ключ надо передавать отправителю и получателю по защищенному каналу распространения ключей, например, спецсвязью. В асимметричной криптосистеме передают по незащищенному каналу только открытый ключ, а секретный ключ сохраняют в месте его генерации.

Злоумышленник при атаке на криптосистему может не только считывать шифротексты, передаваемые по каналу связи, но и пытаться их изменить по своему усмотрению.

Любая попытка со стороны злоумышленника расшифровать шифротекст C для получения открытого текста M или зашифровать свой собственный текст M^1 для получения правдоподобного шифротекста C^1, не имея подлинного ключа, называется криптоатакой.

Свойство криптосистемы, противостоять криптоатаке называется криптостойкостью. Оно измеряется в затратах злоумышленника, которые он несет, вскрывая криптосистему. Например, криптостойкость может выражаться в количестве машинного времени, затраченного на вскрытие криптосистемы.

Фундаментальное правило криптоанализа, впервые сформулированное голландцем А.Керкхоффом еще в XIX веке заключается в том, что стойкость шифра (криптосистемы) должна определяться только секретностью ключа. Иными словами, правило Керкхоффа состоит в том, что весь алгоритм шифрования, кроме значения секретного ключа, известен криптоаналитику противника. Это обусловлено тем, что криптосистема, реализующая семейство криптографических преобразований, обычно рассматривается как открытая система. Такой подход отражает очень важный принцип технологии защиты информации: защищенность системы не должна зависеть от секретности чего-либо такого, что невозможно быстро изменить в случае утечки секретной информации. Обычно криптосистема представляет собой совокупность аппаратных и программных средств, которую можно изменить только при значительных затратах времени и средств, тогда как ключ

является легко изменяемым объектом. Именно поэтому стойкость криптосистемы определяется только секретностью ключа.

Другое почти общепринятое допущение в криптоанализе состоит в том, что криптоаналитик имеет в своем распоряжении шифротексты сообщений.

Существует четыре основных типа криптоаналитических атак. Конечно, все они формулируются в предположении, что криптоаналитику известны применяемый алгоритм шифрования и шифротексты сообщений. Перечислим эти криптоаналитические атаки.

1. Криптоаналитическая атака при наличии только известного шифртекста. Криптоаналитик имеет только шифротексты $C_1, C_2, ..., C_i$, нескольких сообщений, причем все они зашифрованы с использованием одного и того же алгоритма шифрования E_K. Работа криптоаналитика заключается в том, чтобы раскрыть исходные тексты $M_1, M_2, ..., M_i$, по возможности большинства сообщений или, еще лучше, вычислить ключ K, использованный для шифрования этих сообщений, с тем, чтобы расшифровать и другие сообщения, зашифрованные этим ключом.

2. Криптоаналитическая атака при наличии известного открытого текста. Криптоаналитик имеет доступ не только к шифротекстам $C_1, C_2, ..., C_i$, нескольких сообщений, но также к открытым текстам $M_1, M_2, ..., M_i$ этих сообщений. Его работа заключается в нахождении ключа K, используемого при шифровании этих сообщений, или алгоритма расшифровывания D_K любых новых сообщении, зашифрованных тем же самым ключом.

3. Криптоаналитическая атака при возможности выбора открытого текста. Криптоаналитик не только имеет доступ к шифротекстам $C_1, C_2, ..., C_i$, и связанным с ними открытым текстам $M_1, M_2, ..., M_i$, нескольких сообщений, но и может по желанию выбирать открытые тексты, которые затем получает в зашифрованном виде. Такой криптоанализ получается более мощным по сравнению с криптоанализом с известным открытым текстом, потому что криптоаналитик может выбрать для шифрования такие блоки открытого текста, которые дадут больше информации о ключе. Работа криптоаналитика состоит в поиске ключа K, использованного для шифрования сообщений, или алгоритма расшифрования D_K новых сообщений, зашифрованных тем же ключом.

4. Криптоаналитическая атака с адаптивным выбором открытого текста. Это - особый вариант атаки с выбором открытого текста. Криптоаналитик может не только выбирать открытый текст, который затем шифруется, но и изменять свой выбор в зависимости от результатов предыдущего шифрования. При криптоанализе с простым выбором открытого текста криптоаналитик обычно может выбирать несколько крупных блоков открытого текста для их шифрования, при криптоанализе с адаптивным выбором открытого текста он имеет возможность выбрать сначала более мелкий пробный блок открытого текста, затем выбрать следующий блок в зависимости от результатов первого выбора, и т.д. Эта атака

предоставляет криптоаналитику еще больше возможностей, чем предыдущие типы атак.

Кроме перечисленных криптоаналитических атак существует силовая атака (перебор всех возможных значений ключа). С появлением мощных компьютеров и сетей этот вид атаки становится очень актуальным. Он может сочетаться с перечисленными ранее аналитическими атаками. В связи с этим ключ криптосистемы должен обладать определенными свойствами: если рассматривать его как совокупность двоичных знаков, то это должна быть случайная равномерно распределенная последовательность длины, которая делала бы перебор всех возможных значений ключа, практически невозможным.

2.2. Блочные и поточные шифры

Применение функции шифрования ко всему сообщению в целом реализуется очень редко. Практически все применяемые криптографические методы связаны с разбиением сообщения на большое число фрагментов (или знаков) фиксированного размера, каждый из которых шифруется отдельно. Такой подход существенно упрощает задачу шифрования, так как сообщения обычно имеют различную длину.

Можно выделить следующие характерные признаки методов шифрования данных:

- выполнение операций с отдельными битами или блоками.
- зависимость или независимость функции шифрования от результатов шифрования предыдущих частей сообщения.
- зависимость или независимость шифрования отдельных знаков от их положения в тексте.

В соответствии с этим различают три основных способа шифрования:

- поточные шифры;
- блочные шифры;
- блочные шифры с обратной связью.

Поточное шифрование

Поточное шифрование состоит в том, что каждый бит открытого текста и соответствующий бит ключа преобразовываются по определенному алгоритму (например, складываются по модулю 2). К достоинствам поточных шифров относятся высокая скорость шифрования, относительная простота реализации и отсутствие размножения ошибок. Недостатком является необходимость использовать для каждого сообщения другой ключ. Это обусловлено тем, что если два различных сообщения шифруются на одном и том же ключе, то эти сообщения легко могут быть расшифрованы ($C_1 = M_1 \oplus K$, $C_2 = M_2 \oplus K$, $C_1 \oplus C_2 = M_2 \oplus M_2$, считая M_2 ключом можно вычислить M_1, т.к. M_2 не обладает свойствами ключа). Поэтому часто

используют дополнительный, случайно выбираемый ключ сообщения, который передается в начале сообщения и применяется для модификации ключа шифрования. В результате разные сообщения будут шифроваться с помощью различных последовательностей. Это требует передачи информации синхронизации перед заголовком сообщения, которая должна быть принята до расшифрования любого сообщения.

Поточные шифры широко применяются для шифрования преобразованных в цифровую форму речевых сигналов и цифровых данных, требующих оперативной доставки потребителю информации.

Блочное шифрование

При блочном шифровании открытый текст сначала разбивается на равные по длине блоки, затем применяется зависящая от ключа функция шифрования для преобразования блока открытого текста длиной m бит в блок шифротекста такой же длины. При блочном шифровании каждый бит блока шифротекста зависит от значений всех битов соответствующего блока открытого текста, и никакие два блока открытого текста не могут быть представлены одним и тем же блоком шифротекста. При этом небольшие изменения в шифротексте вызывают большие и непредсказуемые изменения в соответствующем открытом тексте, и наоборот. Вместе с тем применение блочного шифра имеет серьезные недостатки. Первый из них заключается в том, что вследствие детерминированного характера шифрования при фиксированной длине блока 64 бита можно осуществить криптоанализ шифротекста "со словарем" в ограниченной форме. Это обусловлено тем, что идентичные блоки открытого текста длиной 64 бита в исходном сообщении представляются идентичными блоками шифротекста, что позволяет криптоаналитику сделать определенные выводы о содержании сообщения. Другой потенциальный недостаток этого шифра связан с размножением ошибок. Результатом изменения только одного бита в принятом блоке шифротекста будет неправильное расшифровывание всего блока. Это, в свою очередь, приведет к появлению искаженных битов (от 1 до 64) в восстановленном блоке исходного текста.

Из-за отмеченных недостатков блочные шифры редко применяются в указанном режиме для шифрования длинных сообщений. Однако в финансовых учреждениях, где сообщения часто состоят из одного или двух блоков, блочные шифры широко используют в режиме прямого шифрования. Такое применение обычно связано с возможностью частой смены ключа шифрования, поэтому вероятность шифрования двух идентичных блоков открытого текста на одном и том же ключе очень мала.

Криптосистема с открытым ключом также является системой блочного шифрования и должна оперировать блоками довольно большой длины. Это обусловлено тем, что криптоаналитик знает открытый ключ шифрования и мог бы заранее вычислить и составить таблицу соответствия блоков открытого текста и шифротекста. Если длина блоков мала, например 30 бит,

то число возможных блоков не слишком большое (при длине 30 бит это $2^{30} \approx 10^9$), и может быть составлена полная таблица, позволяющая моментально расшифровать любое сообщение с использованием известного открытого ключа.

Блочное шифрование с обратной связью

Как и при блочном шифровании, сообщения разбивают на ряд блоков, состоящих из m бит. Для преобразования этих блоков в блоки шифротекста, которые также состоят из m бит, используются специальные функции шифрования. Однако если в блочном шифре такая функция зависит только от ключа, то в блочных шифрах с обратной связью она зависит как от ключа, так и от одного или более предшествующих блоков шифротекста.

Практически важным шифром с обратной связью является шифр со сцеплением блоков шифротекста. В этом случае m бит предыдущего шифротекста суммируются по модулю 2 со следующими m битами открытого текста, а затем применяется алгоритм блочного шифрования под управлением ключа для получения следующего блока шифротекста. Достоинством криптосистем блочного шифрования с обратной связью является возможность применения их для обнаружения манипуляций сообщениями, производимых активными перехватчиками. При этом используется факт размножения ошибок в таких шифрах, а также способность этих систем легко генерировать код аутентификации сообщений. Поэтому системы шифрования с обратной связью используют не только для шифрования сообщений, но и для их аутентификации. Криптосистемам блочного шифрования с обратной связью свойственны некоторые недостатки. Основным из них является размножение ошибок, так как один ошибочный бит при передаче может вызвать ряд ошибок в расшифрованном тексте. Другой недостаток связано тем, что разработка и реализация систем шифрования с обратной связью часто оказываются более трудными, чем систем поточного шифрования.

На практике для шифрования длинных сообщений применяют поточные шифры или шифры с обратной связью. Выбор конкретного типа шифра зависит от назначения системы и предъявляемых к ней требований.

3. СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

3.1. Основные понятия и определения

Большинство средств защиты информации базируется на использовании криптографических шифров и процедур шифрования - расшифрования. В соответствии со стандартом ГОСТ 28147-89 под шифром понимают совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, задаваемых ключом и алгоритмом криптографического преобразования.

Ключ - это конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор только одного варианта из всех возможных для данного алгоритма. Основной характеристикой шифра является криптостойкость, которая определяет его стойкость к раскрытию методами криптоанализа. Обычно эта характеристика определяется интервалом времени, необходимым для раскрытия шифра. К шифрам, используемым для криптографической защиты информации, предъявляется ряд требований:

- достаточная криптостойкость (надежность закрытия данных);
- простота процедур шифрования и расшифрования;
- незначительная избыточность информации за счет шифрования;
- нечувствительность к небольшим ошибкам шифрования и др.

В той или иной мере этим требованиям отвечают:

- шифры перестановок;
- шифры замены;
- шифры гаммирования;
- шифры, основанные на аналитических преобразованиях шифруемых данных.

Шифрование перестановкой заключается в том, что символы шифруемого текста переставляются по определенно правилу в пределах некоторого блока этого текста. При достаточной длине блока, в пределах которого осуществляется перестановка, и сложном неповторяющемся порядке перестановки можно достигнуть приемлемой для простых практических приложений стойкости шифра.

Шифрование заменой (подстановкой) заключается в том, что символы шифруемого текста заменяются символами того же или другого алфавита в соответствии с заранее обусловленной схемой замены.

Шифрование гаммированием заключается в том, что символы шифруемого текста складываются с символами некоторой случайной последовательности, именуемой гаммой шифра. Стойкость шифрования определяется в основном длиной (периодом) неповторяющейся части гаммы шифра. Поскольку с помощью ЭВМ можно генерировать практически бесконечную гамму шифра, то данный способ является одним из основных для шифрования информации в автоматизированных системах.

Шифрование аналитическим преобразованием заключается в том, что шифруемый текст преобразуется по некоторому аналитическому правилу (формуле). Например, можно использовать правило умножения вектора на матрицу, причем умножаемая матрица является ключом шифрования (поэтому ее размер и содержание должны хранится в секрете), а символами умножаемого вектора последовательно служат символы шифруемого текста. Процессы шифрования и расшифрования осуществляются в рамках некоторой криптосистемы. Характерной особенностью симметричной криптосистемы является применение одного секретного ключа как при шифровании, так и при расшифровании сообщений.

Как открытый текст, так и шифротекст образуются из букв, входящих в конечное множество символов, называемое алфавитом. Примерами алфавитов являются конечное множество всех заглавных букв, конечное множество всех заглавных и строчных букв и цифр и т.п. В общем случае некоторый алфавит можно представить как $\Sigma = \{a_0, a_1, \ldots, a_{m-1}\}$.

Объединяя по определенному правилу буквы из алфавита Σ, можно создать новые алфавиты:

- алфавит Σ^2, содержит m^2 биграмм $a_0a_0, a_0a_1, \ldots, a_{m-1}a_{m-1}$;
- алфавит Σ^3, содержит m^3 биграмм $a_0a_0a_0, a_0a_0a_1, \ldots, a_{m-1}a_{m-1}a_{m-1}$;

В общем случае, объединяя по n букв, получаем алфавит Σ^n, содержащий m^n - n-грамм. Например, английский алфавит $\{ABCD...XYZ\}$ объемом $m = 26$ букв позволяет сгенерировать $26^2 = 676$ биграмм $AA, AB, ..., ZZ$, $26^3 = 17576$ триграмм $AAA, AAB, ...ZZY, ZZZ$.

При выполнении криптографических преобразований полезно заменить буквы алфавита целыми числами $0,1,2,...$ Это позволяет упростить выполнение необходимых алгебраических манипуляций. Например, можно установить взаимно однозначное соответствие между русским алфавитом {АБВГ…ЮЯ} и множеством целых чисел $\Sigma_{32} = \{0,1,2,...,31\}$, между английским алфавитом {ABCD…YZ} и множеством целых чисел $\Sigma_{32} = \{0,1,2,...,26\}$.

В дальнейшем будет обычно использоваться алфавит $\overline{\Sigma}_m = \{0,1,2,...,m-1\}$,
содержащий m «букв» в виде чисел. Замена букв традиционного алфавита числами позволит более четко сформулировать основные концепции и приемы криптопреобразований. В тоже время в большинстве иллюстраций будет использоваться алфавит естественного языка.

Текст с n буквами из алфавита $\overline{\Sigma}_m$ можно рассматривать как n-грамму $\overline{X} = (x_0, x_1, ..., x_{n-1})$, где $x \in \overline{Z}_m$, для некоторого целого $n = 1,2,3,...$. Через $\overline{Z}_{m,n}$ будем обозначать множество n-грамм, образованных из букв множества $\overline{Z}_m$.

Криптографическое преобразование E представляет собой совокупность преобразований $E = \left\{ E^{(n)} : 1 \le n < \infty \right\}$, $E^{(n)} : \overline{Z}_{m,n} \to \overline{Z}_{m,n}$. Преобразование $E^{\{n\}}$ определяет, как каждая n-грамма открытого текста $\overline{x} \in \overline{Z}_{m,n}$ заменяется n-граммой шифротекста $\overline{y}$, т.е. $\overline{y} = E^{(n)}\left(\overline{x}\right)$, причем $\overline{x}$, $\overline{y} \in \overline{Z}_{m,n}$, при этом обязательным является требование взаимной однозначности преобразования, $E\left(n\right)$ на множестве $\overline{Z}_{m,n}$.

Криптографическая система может трактоваться как семейство криптографических преобразований $E = \left\{ E_K : K \in \overline{k} \right\}$, помеченных параметром K, называемых ключом. Множество значений ключа образуют ключевое пространство $\overline{k}$.

3.2. Традиционные симметричные криптосистемы

Традиционные (классические) методы шифрования, отличаются симметричной функцией шифрования. К ним относятся шифры перестановки, шифры простой и сложной замены, а также некоторые их модификации и комбинации. Следует отметить, что комбинации шифров перестановок и шифров замены образуют все многообразие применяемых на практике симметричных шифров.

Шифры перестановок

Правило перестановок символов - является ключом и задается различными предметами: цилиндром (скитала, древние греки), размером таблицы, условным словом или фразой (шифрующие таблицы в эпоху Возрождения), магическим квадратом в средние века.

Шифры простой замены

В шифрах простой замены каждый символ открытого текста заменяется символом того же или другого алфавита по определенному правилу. Широко известны и исследованы шифры Цезаря. Такие шифры имеют слабость по отношению к атакам на основе подсчета частот появления букв в шифротексте. Более устойчивыми являются биграммные шифры (замена двух букв) и n граммные шифры, позволяющие маскировать частоты появления букв.

Шифры сложной замены

Шифры сложной замены называют многоалфавитными, т.к. для шифрования каждого символа исходного сообщения применяют свой шифр простой замены. Многоалфавитная подстановка последовательно и

циклически меняет используемые алфавиты. Например, в r- алфавитной подстановке символ x_0 исходного текста заменяется символом y_0 из алфавита B_0, x_1 на y_1 из B_1, x_{r-1} на y_{r-1} из B_{r-1}, x_r на y_r из B_0. Многоалфавитная подстановка маскирует естественную статистику исходного языка, т.к. конкретный символ из алфавита А может быть преобразован в несколько символов шифровальных алфавитов В. К шифры сложной замены относят шифры Гронсфельда, Вижинера, Вернама. В 20 годах были созданы первые шифровальные машины (электромеханические), реализующие шифры сложной замены. Эти машины использовались до 60-х годов.

Шифрование методом гаммирования

Под гаммированием понимают процесс наложения по определенному закону гаммы шифра на открытые данные. Гамма шифра - это псевдослучайная последовательность, выработанная по заданному алгоритму для шифрования открытых данных и расшифрования зашифрованных данных. Процесс шифрования заключается в генерации гаммы шифра и наложении полученной гаммы на исходный открытый текст обратимым образом, например с использованием операции сложения по модулю 2.

Следует отметить, что перед шифрованием открытые данные разбивают на блоки T_O одинаковой длины, обычно по 64 бита. Гамма шифра вырабатывается в виде последовательности блоков Гш аналогичной длины.

Уравнение шифрования можно записать в виде $T_ш^i = Г_ш^i \oplus T_O^i$, $i = 1,2,...,M$, где $T_ш^i$ - i-й блок шифротекста, $Г_ш^i$ - i-й блок гаммы шифра, T_O^i- i-й блок открытого текста, M - количество блоков открытого текста.

Процесс расшифрования сводится к повторной генерации гаммы шифра и наложению этой гаммы на зашифрованные данные. Уравнение расшифрования имеет вид: $T_O^i = Г_ш^i \oplus T_ш^i$

Получаемый этим методом шифротекст достаточно труден для раскрытия, поскольку теперь ключ является переменным. По сути дела гамма шифра должна изменяться случайным образом для каждого шифруемого блока. Если период гаммы превышает длину всего шифруемого текста и злоумышленнику неизвестна никакая часть исходного текста, то такой шифр можно раскрыть только прямым перебором всех вариантов ключа. В этом случае криптостойкость шифра определяется длиной ключа.

3.3. Современные симметричные криптосистемы

По мнению К. Шеннона, в практических шифрах необходимо использовать два общих принципа: рассеивание и перемешивание.

Рассеивание представляет собой распространение влияния одного знака открытого текста на много знаков шифротекста, что позволяет скрыть статистические свойства открытого текста.

Перемешивание предполагает использование таких шифрующих преобразований, которые усложняют восстановление взаимосвязи статистических свойств открытого и шифрованного текстов. Однако шифр должен не только затруднять раскрытие, но обеспечивать легкость шифрования и расшифрования при известном пользователю секретном ключе.

Распространенным способом достижения эффектов рассеивания и перемешивания является использование составного шифра, т.е. такого шифра, который может быть реализован в виде некоторой последовательности простых шифров, каждые из которых вносит свой вклад в значительное суммарное рассеивание и перемешивание.

В составных шифрах в качестве простых шифров чаще всего используются простые перестановки и подстановки. При перестановке просто перемешивают символы открытого текста, причем конкретный вид перемешивания определяется секретным ключом. При подстановке каждый символ открытого текста заменяют другим символом из того же алфавита, а конкретный вид подстановки также определяется секретным ключом. Следует заметить, что в современном блочном шифре блоки открытого текста и шифротекста представляют собой двоичные последовательности обычно длиной 64 бита. В принципе каждый блок может принимать 2^{64} значений. Поэтому подстановки выполняются в очень большом алфавите, содержащем до $2^{54} \approx 10^{19}$ символов.

При многократном чередовании простых перестановок и подстановок, управляемых достаточно длинным секретным ключом, можно получить очень стойкий шифр с хорошим рассеиванием и перемешиванием.

4. СТАНДАРТ ШИФРОВАНИЯ ДАННЫХ ГОСТ 28147-89

ГОСТ 28147-89 представляет собой 64-битовый блочный алгоритм с 256-битовым ключом. Он предназначен для аппаратной и программной реализации, удовлетворяет криптографическим требованиям и не накладывает ограничений на степень секретности защищаемой информации.

Алгоритм предусматривает четыре режима работы:

1. шифрование данных в режиме простой замены;
2. шифрование данных в режиме гаммирования;
3. шифрование данных в режиме гаммирования с обратной связью;
4. выработка имитовставки.

Основными режимами шифрования являются режимы с использованием гаммирования, однако они базируются на использовании шифрования данных в режиме простой замены.

4.1. Режим простой замены

Шифрование открытых данных в режиме простой замены

Открытые данные, подлежащие шифрованию, разбивают на 64-разрядные блоки T_i . Процедура шифрования 64-разрядного блока T_i в режиме простой замены включает 32 цикла $(j = 1,2,\ldots,32)$. В ключевое запоминающее устройство вводят 256 бит ключа K в виде восьми 32-разрядных подключей (чисел) K_i

$$K=K_7K_6K_5K_4K_3K_2K_1K_0.$$

Последовательность битов блока

$$T_O = (a_1(0), a_2(0), \ldots, a_{31}(0), a_{32}(0), b_1(0), b_2(0), \ldots, b_{31}(0), b_{32}(0))$$

разбивают на две последовательности по 32 бита: $b(0)$ и $a(0)$, где $b(0)$ - левые или старшие биты, $a(0)$ - правые или младшие биты.

Работа алгоритма в режиме простой замены изображена на рис. 4.1.

Обозначения на схеме:

N_1, N_2 32-разрядные накопители;

CM_1 32-разрядный сумматор по модулю $2^{32}(+)$;

СМ2 32-разрядный сумматор по модулю $2(\oplus)$;

R 32-разрядный регистр циклического сдвига;

КЗУ - ключевое запоминающее устройство на 256 бит, состоящее из восьми 32-разрядных накопителей $X_0, X_1, X_2, \ldots, X_7$;

S - блок подстановки, состоящий из восьми узлов замены (S-блоков замены) $S_1, S_2, S_3, \ldots, S_8$.

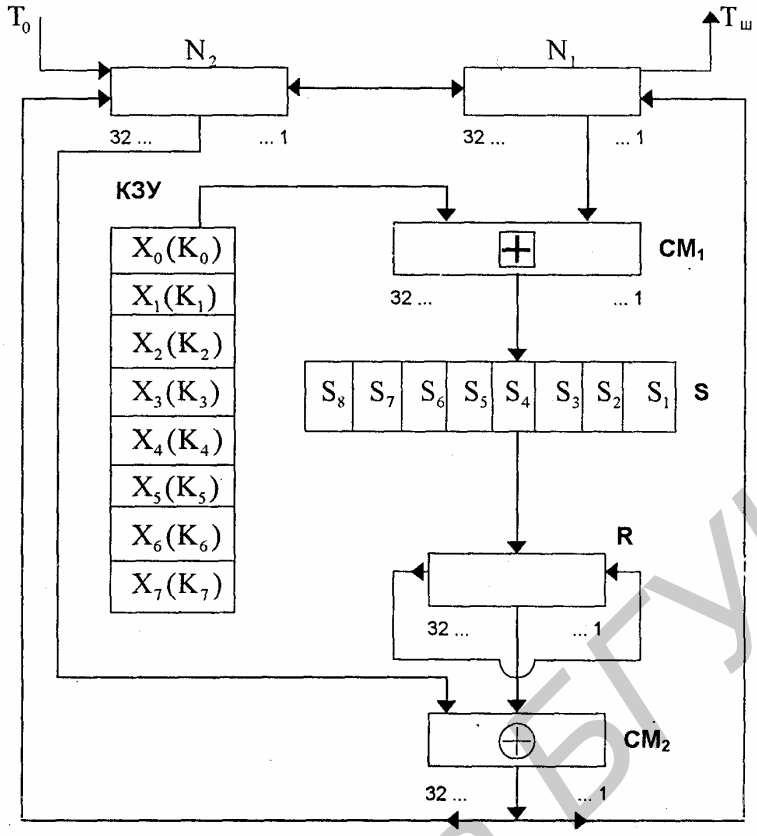

Рис. 4.1. Схема реализации режима простой замены

Эти последовательности вводят в накопители N_1 и N_2 перед началом первого цикла шифрования. В результате начальное заполнение накопителя N_1:

$$a(0) = (a_{32}(0),\ a_{31}(0),\ \ldots,\ a_2(0),\ a_1(0))$$

$$32,\quad 31,\quad \ldots,\quad 2,\quad 1 \qquad \leftarrow \textit{номер разряда}\ N_1,$$

начальное заполнение накопителя N_2:

$$b(0) = (b_{32}(0),\ b_{31}(0),\ \ldots,\ b_2(0),\ b_1(0))$$

$$32,\quad 31,\quad \ldots,\quad 2,\quad 1 \qquad \leftarrow \textit{номер разряда}\ N_2.$$

Первый цикл $(j=1)$ процедуры шифрования 64-разрядного блока открытых данных можно описать уравнениями:

$$\begin{cases} a(1) = f\left(a(0) + K_0\right) \oplus b(0), \\ b(1) = a(0). \end{cases}$$

Здесь $a(1)$ - заполнение N_1 после 1-го цикла шифрования; $b(1)$ - заполнение N_2 после 1-го цикла шифрования; f - функция шифрования.

Аргументом функции f является сумма по модулю 2^{32} числа $a(0)$ (начального заполнения накопителя N_1) и числа K_0 подключа, считываемого из накопителя X_0 КЗУ. Каждое из этих чисел равно 32 битам.

Функция f включает две операции над полученной 32-разрядной суммой ($a(0)+\mathrm{K}_0$).

Первая операция называется подстановкой (заменой) и выполняется блоком подстановки S. Блок подстановки S состоит из восьми узлов замены (S-блоков замены) $\mathrm{S}_1,\mathrm{S}_2,\ldots,\mathrm{S}_8$ с памятью 64 бит каждый. Поступающий из CM_1 на блок подстановки S 32-разрядный вектор разбивают на восемь последовательно идущих 4-разрядных векторов, каждый из которых преобразуется в четырехразрядный вектор соответствующим узлом замены. Каждый узел замены можно представить в виде таблицы-перестановки шестнадцати четырехразрядных двоичных чисел в диапазоне 0000…1111. Входной вектор указывает адрес строки в таблице, а число в этой строке является выходным вектором. Затем четырехразрядные выходные векторы последовательно объединяют в 32-разрядный вектор. Узлы замены (таблицы-перестановки) представляют собой ключевые элементы, которые являются общими для сетей ТКС и редко изменяются. Эти узлы замены должны сохраняться в секрете.

Вторая операция - циклический сдвиг влево (на 11 разрядов) 32-разрядного вектора, полученного с выхода блока подстановки S. Циклический сдвиг выполняется регистром сдвига R. Затем результат работы функции шифрования f суммируют поразрядно по модулю 2 в сумматоре CM_2 с 32-разрядным начальным заполнением $b(0)$ накопителя N_2. Затем полученный на выходе CM_2 результат (значение $a(1)$) записывают в накопитель N_1, а старое значение N_1 (значение $a(0)$) переписывают в накопитель N_2 (значение $b(1)=a(0)$). Первый цикл завершен. Последующие циклы осуществляются аналогично, при этом во втором цикле из КЗУ считывают заполнение X_1 - подключ K_1, в третьем цикле - подключ K_2 и т.д., в восьмом цикле - подключ K_7. В циклах с 9-го по 16-й, а также в циклах с 17-го по 24-й подключи из КЗУ считываются в том же порядке: $\mathrm{K}_0,\mathrm{K}_1,\mathrm{K}_2,\ldots,\mathrm{K}_6,\mathrm{K}_7$. В последних восьми циклах с 25-го по 32-й порядок считывания подключей из КЗУ обратный: $\mathrm{K}_7,\mathrm{K}_6,\ldots,\mathrm{K}_2,\mathrm{K}_1,\mathrm{K}_0$. Таким образом, при шифровании в 32 циклах осуществляется следующий порядок выборки из КЗУ подключей:

$$\mathrm{K}_0,\mathrm{K}_1,\mathrm{K}_2,\mathrm{K}_3,\mathrm{K}_4,\mathrm{K}_5,\mathrm{K}_6,\mathrm{K}_7,\mathrm{K}_0,\mathrm{K}_1,\mathrm{K}_2,\mathrm{K}_3,\mathrm{K}_4,\mathrm{K}_5,\mathrm{K}_6,\mathrm{K}_7,$$
$$\mathrm{K}_0,\mathrm{K}_1,\mathrm{K}_2,\mathrm{K}_3,\mathrm{K}_4,\mathrm{K}_5,\mathrm{K}_6,\mathrm{K}_7,\mathrm{K}_7,\mathrm{K}_6,\mathrm{K}_5,\mathrm{K}_4,\mathrm{K}_3,\mathrm{K}_2,\mathrm{K}_1,\mathrm{K}_0.$$

В 32-м цикле результат из сумматора CM_2 вводится в накопитель N_2, а в накопителе N_1 сохраняется прежнее заполнение. Полученные после 32-го цикла шифрования заполнения накопителей N_1 и N_2 являются блоком зашифрованных данных $\mathrm{T}_{\text{Ш}}$, соответствующим блоку открытых данных $\mathrm{T}_{\text{О}}$.

Уравнения шифрования в режиме простой замены имеют вид

$$\begin{cases} a(j) = f\left(a(j-1) \boxplus \mathrm{K}_{(j-1) \bmod 8}\right) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad при\ j = 1 \ldots 24,$$

$$\begin{cases} a(j) = f\left(a(j-1) \boxplus \mathrm{K}_{(32-j) \bmod 8}\right) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad при\ j = 25 \ldots 31,$$

$$\begin{cases} a(32) = a(31) \\ b(32) = f\left(a(31) \boxplus \mathrm{K}_0\right) \oplus b(31) \end{cases} \quad при\ j = 32.$$

где $a(j) = \left(a_{32}(j), a_{31}(j), \ldots, a_1(j)\right)$ - заполнение N_1 после j-го цикла шифрования; $b(j) = \left(b_{32}(j), b_{31}(j), \ldots, b_1(j)\right)$ - заполнение N_2 после j-го цикла шифрования, $j = 1 \ldots 32$.

Блок шифрованных данных $\mathrm{T}_{\text{Ш}}$ (64 разряда) выводится из накопителей N_1, N_2 в следующем порядке: из разрядов 1…32 накопителя N_1, затем из разрядов 1…32 накопителя N_2, т.е. начиная с младших разрядов:

$$\mathrm{T}_{\text{Ш}} = \left(a_1(32), a_2(32), \ldots, a_{31}(32), a_{32}(32), b_1(32), b_2(32), \ldots, b_{31}(32), b_{32}(32)\right).$$

Остальные блоки открытых данных шифруются в режиме простой замены аналогично.

Расшифрование в режиме простой замены

Криптосхема, реализующая алгоритм расшифрования в режиме простой замены, имеет тот же вид, что и при шифровании (см. рис. 1).

В КЗУ вводят 256 бит ключа, на котором осуществлялось шифрование. Зашифрованные данные, подлежащие расшифровыванию, разбиты на блоки $\mathrm{T}_{\text{Ш}}$, по 64 бита в каждом. Ввод любого блока

$$\mathrm{T}_{\text{Ш}} = \left(a_1(32), a_2(32), \ldots, a_{31}(32), a_{32}(32), b_1(32), b_2(32), \ldots, b_{31}(32), b_{32}(32)\right)$$

в накопители N_1 и N_2 производят так, чтобы начальное значение накопителя N_1 имело вид

$$\begin{array}{cccccc} \left(a_{32}(32), & a_{31}(32), & \ldots, & a_2(32), & a_1(32)\right) & \\ 32, & 31, & \ldots, & 2, & 1 & \leftarrow \textit{номер разряда}\ \mathrm{N}_1, \end{array}$$

а начальное заполнение накопителя N_2:

$$\begin{array}{cccccc} \left(b_{32}(32), & b_{31}(32), & \ldots, & b_2(32), & b_1(32)\right) & \\ 32, & 31, & \ldots, & 2, & 1 & \leftarrow \textit{номер разряда}\ \mathrm{N}_2. \end{array}$$

Расшифрование осуществляется по тому же алгоритму, что и шифрование, с тем изменением, что заполнения накопителей $\mathrm{X}_0, \mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_7$ считываются из КЗУ в циклах расшифрования в следующем порядке:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0,$$
$$K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

Уравнения расшифрования имеют вид

$$\begin{cases} a(32-j) = f\left(a(32-j+1) \square K_{j-1}\right) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad при\ j = 1 \ldots 8,$$

$$\begin{cases} a(32-j) = f\left(a(32-j+1) \square K_{(32-j) \bmod 8}\right) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad при\ j = 9 \ldots 31,$$

$$\begin{cases} a(0) = a(1) \\ b(0) = f\left(a(1) \square K_0\right) \oplus b(1) \end{cases} \quad при\ j = 32.$$

Полученные после 32 циклов работы заполнения накопителей N_1 и N_2 образуют блок открытых данных

$$T_O = \left(a_1(0), a_2(0), \ldots, a_{31}(0), a_{32}(0), b_1(0), b_2(0), \ldots, b_{31}(0), b_{32}(0)\right),$$

соответствующий блоку зашифрованных данных $T_Ш$. При этом состояние накопителя N_1:

$$\begin{array}{cccccc} \left(a_{32}(0), & a_{31}(0), & \ldots, & a_2(0), & a_1(0)\right) & \\ 32, & 31, & \ldots, & 2, & 1 & \leftarrow номер\ разряда\ N_1, \end{array}$$

состояние накопителя N_2:

$$\begin{array}{cccccc} \left(b_{32}(0), & b_{31}(0), & \ldots, & b_2(0), & b_1(0)\right) & \\ 32, & 31, & \ldots, & 2, & 1 & \leftarrow номер\ разряда\ N_2. \end{array}$$

Аналогично расшифровываются остальные блоки зашифрованных данных.

Если алгоритм шифрования в режиме простой замены 64-битового блока T_O обозначить через A, то

$$A(T_O) = A\left(a(0), b(0)\right) = \left(a(32), b(32)\right) = T_Ш.$$

Следует иметь в виду, что режим простой замены допустимо использовать для шифрования данных только в ограниченных случаях при выработке ключа и шифровании его с обеспечением имитозащиты для передачи по каналам связи или для хранения в памяти ЭВМ.

4.2. Режим гаммирования

Шифрование открытых данных в режиме гаммирования

Криптосхема, реализующая алгоритм шифрования в режиме гаммирования, показана на рис. 4.2. Открытые данные разбивают на 64-разрядные блоки

$$T_O^{(1)}, T_O^{(2)}, \ldots, T_O^{(i)}, \ldots, T_O^{(m)},$$

где $T_{О}^{(i)}$ - i-й 64-разрядный блок открытых данных, $i = 1 \ldots m$, m определяется объемом шифруемых данных.

Эти блоки поочередно шифруются в режиме гаммирования путем поразрядного сложения по модулю 2 в сумматоре $СМ_5$ с гаммой шифра $Г_{Ш}$, которая вырабатывается блоками по 64 бита, т.е.

$$Г_{Ш} = \left(Г_{Ш}^{(1)}, Г_{Ш}^{(2)}, \ldots, Г_{Ш}^{(i)}, \ldots, Г_{Ш}^{(m)}\right),$$

где $Г_{Ш}^{(i)}$ - i-й 64-разрядный блок, $i = 1 \ldots m$.

Число двоичных разрядов в блоке $T_{О}^{(m)}$ может быть меньше 64, при этом не использованная для шифрования часть гаммы шифра из блока $Г_{Ш}^{(m)}$ отбрасывается.

Уравнение шифрования данных в режиме гаммирования имеет вид

$$T_{Ш}^{(i)} = T_{О}^{(i)} \oplus Г_{Ш}^{(i)},$$

где $Г_{Ш}^{(i)} = A\left(Y_{i-1} \square C_2, Z_{i-1} \square C_1\right)$, $i = 1 \ldots m$; $T_{Ш}^{(i)}$ - i-й блок 64-разрядного блока зашифрованного текста; $A(*)$ - функция шифрования в режиме простой замены; C_1, C_2 - 32-разрядные двоичные константы; Y_i, Z_i - 32-разрядные двоичные последовательности.

Величины Y_i, Z_i определяются итерационно по мере формирования гаммы $Г_{Ш}$ следующим образом:

$$\left(Y_0, Z_0\right) = A\left(\tilde{S}\right),$$

где $\tilde{S}$ - синхропосылка (64-разрядная двоичная последовательность),

$$\left(Y_i, Z_i\right) = \left(Y_{i-1} + C_2, Z_{i-1} + C_1\right), i = 1 \ldots m.$$

Рассмотрим реализацию процедуры шифрования в режиме гаммирования. В накопители N_6 и N_5 заранее записаны 32-разрядные двоичные константы C_1 и C_2, имеющие следующие значения (в шестнадцатеричной форме):

$$C_1 = 01010104_{(16)}, C_2 = 01010101_{(16)}.$$

В КЗУ вводится 256 бит ключа; в накопители N_1 и N_2 - 64-разрядная двоичная последовательность (синхропосылка)

$$\tilde{S}=(S_1,S_2,\ldots,S_{64}).$$

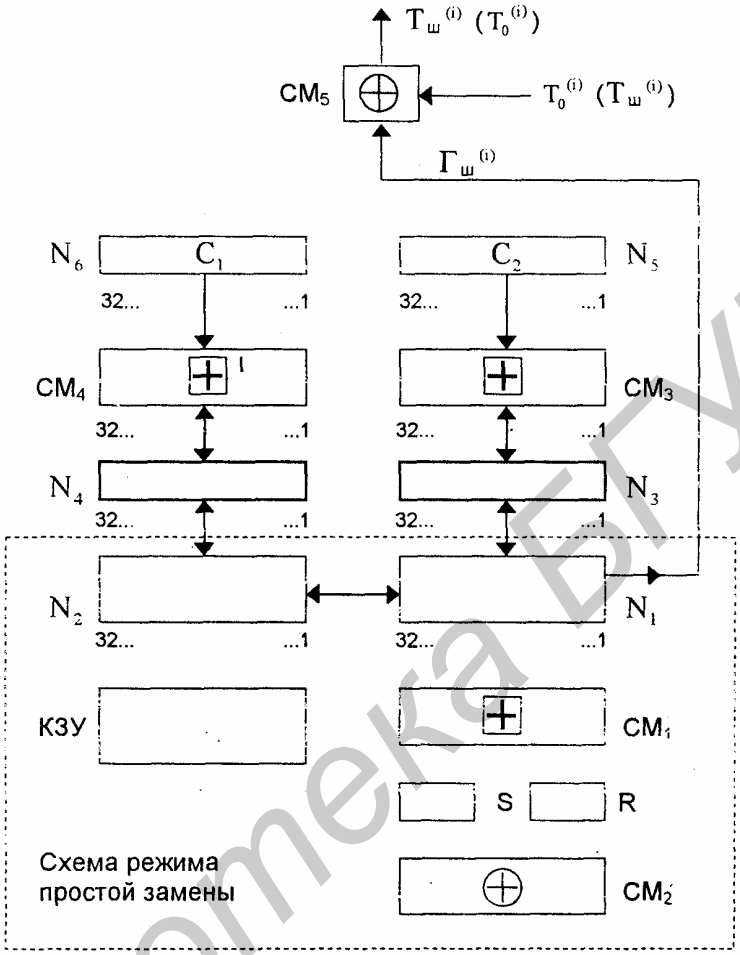

Рис. 4.2. Схема реализации режима гаммирования

Синхропосылка $\tilde{S}$ является исходным заполнением накопителей N_1 и N_2 для последовательной выработки m блоков гаммы шифра.

Исходное заполнение накопителя N_1:

$$\begin{array}{cccccc}(S_{32}, & S_{31}, & \ldots, & S_2, & S_1) & \\ 32, & 31, & \ldots, & 2, & 1 & \leftarrow \textit{номер разряда}\, N_1,\end{array}$$

состояние накопителя N_2:

$$\begin{array}{cccccc}(S_{64}, & S_{63}, & \ldots, & S_{34}, & S_{33}) & \\ 64, & 63, & \ldots, & 34, & 33 & \leftarrow \textit{номер разряда}\, N_2.\end{array}$$

Исходное заполнение N_1 и N_2 (синхропосылка $\tilde{S}$ шифруется в режиме простой замены. Результат шифрования

$$A(\tilde{S})=(Y_0,Z_0)$$

переписывается в 32-разрядные накопители N_3 и N_4 так, что заполнение N_1 переписывается в N_3, а заполнение N_2 - в N_4.

Заполнение накопителя N_4 суммируют по модулю $2^{32}-1$ в сумматоре $СМ_4$ с 32-разрядной константой C_1 из накопителя N_6. Результат записывается в N_4. Заполнение накопителя N_3 суммируется по модулю 2^{32} в сумматоре $СМ_3$ с 32-разрядной константой C_3 из накопителя N_5. Результат записывается в N_3. Заполнение N_3 переписывают в N_1, а заполнение N_4 - в N_2, при этом заполнения N_3, N_4 сохраняются. Заполнение накопителей зашифровывается в режиме простой замены.

Полученное в результате шифрования заполнение накопителей N_1 и N_2 образует первый 64-разрядный блок гаммы шифра

$$\Gamma_{Ш}^{(1)} = \left(\gamma_1^{(1)}, \gamma_2^{(1)}, ..., \gamma_{63}^{(1)}, \gamma_{64}^{(1)}\right),$$

который суммируют поразрядно по модулю 2 в сумматоре $СМ_5$ с первым 64-разрядным блоком открытых данных

$$T_{О}^{(1)} = \left(t_1^{(1)}, t_2^{(1)}, \ldots, t_{63}^{(1)}, t_{64}^{(1)}\right).$$

В результате суммирования по модулю 2 значений $\Gamma_{Ш}^{(1)}$ и $T_{О}^{(1)}$ получают первый 64-разрядный блок шифрованных данных

$$T_{Ш}^{(1)} = \Gamma_{Ш}^{(1)} \oplus T_{О}^{(1)} = \left(\tau_1^{(1)}, \tau_2^{(1)}, \ldots, \tau_{63}^{(1)}, \tau_{64}^{(1)}\right),$$

где $\tau_i^{(1)} = t_i^{(1)} \oplus \gamma_i^{(1)}$, $i = 1 \ldots 64$.

Для получения следующего 64-разрядного блока гаммы шифра $\Gamma_{Ш}^{(2)}$ заполнение N_4 суммируется по модулю $\left(2^{32}-1\right)$ в сумматоре $СМ_4$ с константой C_1 из N_6. Результат записывается в N_4. Заполнение N_3 суммируется по модулю 2^{32} в сумматоре $СМ_3$ с константой C_2 из N_5. Результат записывается в N_3. Новое заполнение N_3 переписывают в N_1, а новое заполнение N_4 - в N_2, при этом заполнения N_3 и N_2 сохраняют. Заполнения N_1, N_2 шифруют в режиме простой замены.

Полученное в результате шифрования заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $\Gamma_{Ш}^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ со вторым блоком открытых данных $T_{О}^{(2)}$

$$T_{Ш}^{(2)} = \Gamma_{Ш}^{(2)} \oplus T_{О}^{(2)}.$$

Аналогично вырабатываются блоки гаммы шифра $\Gamma_{Ш}^{(3)}, \Gamma_{Ш}^{(4)}, \ldots, \Gamma_{Ш}^{(m)}s$ и шифруются блоки открытых данных $T_{О}^{(3)}, T_{О}^{(3)}, \ldots, T_{О}^{(m)}$.

В канал связи или память ЭВМ передаются синхропосылка $\tilde{S}$ и блоки зашифрованных данных:

$$Т_{Ш}^{(1)}, Т_{Ш}^{(2)}, \ldots, Т_{Ш}^{(m)}.$$

Расшифрование в режиме гаммирования

При расшифровании криптосхема имеет тот же вид, что и при шифровании (см. рис. 2).

Уравнение расшифрования

$$Т_{О}^{(i)} = Т_{Ш}^{(i)} \oplus Г_{Ш}^{(i)} = Т_{Ш}^{(i)} \oplus A\left(Y_{i-1} \square C_2, Z_{i-1} \square C_1\right),\ i = 1 \ldots m.$$

Следует отметить, что расшифрование данных возможно только при наличии синхропосылки, которая не является секретным элементом шифра и может храниться в памяти ЭВМ или передаваться по каналам связи вместе с зашифрованными данными.

Рассмотрим реализацию процедуры расшифрования. В КЗУ вводят 256 бит ключа, с помощью которого осуществляется шифрование данных $Т_{О}^{(1)}, Т_{О}^{(2)}, \ldots, Т_{О}^{(m)}$. В накопители N_1 и N_2 вводится синхропосылка и осуществляется процесс выработки m блоков гаммы шифра $Г_{Ш}^{(1)}, Г_{Ш}^{(2)}, \ldots, Г_{Ш}^{(m)}$. Блоки шифрованных данных $Т_{Ш}^{(1)}, Т_{Ш}^{(2)}, \ldots, Т_{Ш}^{(m)}$ суммируются поразрядно по модулю 2 в сумматоре $СМ_5$ с блоками гаммы шифра $Г_{Ш}^{(1)}, Г_{Ш}^{(2)}, \ldots, Г_{Ш}^{(m)}$. В результате получаются блоки открытых данных $Т_{О}^{(1)}, Т_{О}^{(2)}, \ldots, Т_{О}^{(m)}$, при этом $Т_{О}^{(m)}$ может содержать меньше 64 разрядов.

4.3. Режим гаммирования с обратной связью

Криптосхема, реализующая алгоритм шифрования в режиме гаммирования с обратной связью, имеет вид, показанный на рис. 4.3.

Шифрование открытых данных в режиме гаммирования с обратной связью

Открытые данные, разбитые на 64-разрядные блоки $Т_{О}^{(1)}, Т_{О}^{(2)}, \ldots, Т_{О}^{(m)}$, шифруются в режиме гаммирования с обратной связью путем поразрядного сложения по модулю 2 с гаммой шифра $Г_{Ш}$, которая вырабатывается блоками по 64 бита: $Г_{Ш}^{(1)}, Г_{Ш}^{(2)}, \ldots, Г_{Ш}^{(m)}$.

Число двоичных разрядов в блоке $Т_{О}^{(m)}$ может быть меньше 64, при этом не использованная для шифрования часть гаммы шифра из блока $Г_{Ш}^{(m)}$ отбрасывается.

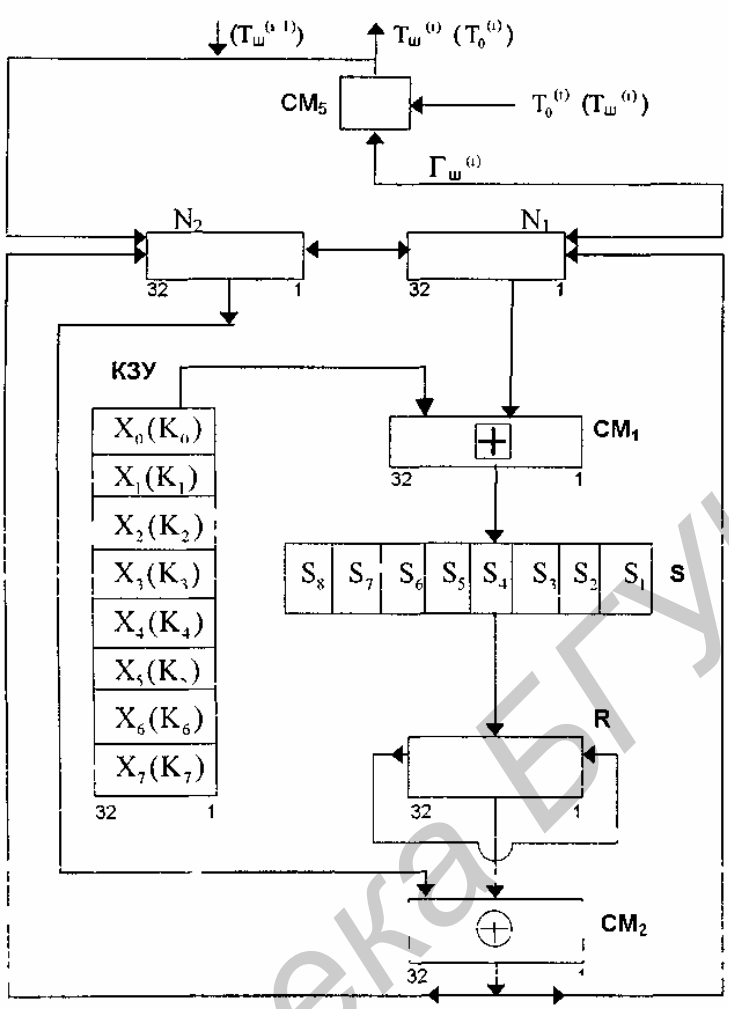

Рис. 4.3. Схема реализации режима гаммирования с обратной связью

Уравнения шифрования в режиме гаммирования с обратной связью имеют вид:

$$T_{Ш}^{(1)} = A\left(\tilde{S}\right) \oplus T_{O}^{(1)} = \Gamma_{Ш}^{(1)} \oplus T_{O}^{(1)},$$

$$T_{Ш}^{(i)} = A\left(T_{Ш}^{(i-1)}\right) \oplus T_{O}^{(i)} = \Gamma_{Ш}^{(i)} \oplus T_{O}^{(i)},\ i = 2 \ldots m.$$

Здесь $T_{Ш}^{(i)}$ - i-й 64-разрядный блок шифрованного текста; $A(*)$ - функция шифрования в режиме простой замены; m определяется объемом открытых данных.

Аргументом функции $A(*)$ на первом шаге итеративного алгоритма является 64-разрядная синхропосылка S, а на всех последующих шагах - предыдущий блок шифрованных данных $T_{Ш}^{(i-1)}$.

Процедура шифрования данных в режиме гаммирования с обратной связью реализуется следующим образом. В КЗУ вводятся 256 бит ключа, в накопители N_1 и N_2 вводится синхропосылка $\tilde{S}=(S_1, S_2, \ldots, S_{64})$ из 64 бит. Исходное заполнение накопителей N_1 и N_2 шифруется в режиме простой замены. Полученное в результате шифрования заполнение накопителей N_1 и

N_2 образует первый 64-разрядный блок гаммы шифра $\Gamma_{Ш}^{(1)} = A\left(\tilde{S}\right)$, который суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ с первым 64-разрядным блоком открытых данных

$$T_{О}^{(1)} = \left(t_1^{(1)}, t_2^{(1)}, \ldots, t_{63}^{(1)}, t_{64}^{(1)}\right).$$

В результате получают первый 64-разрядный блок шифрованных данных

$$T_{Ш}^{(1)} = \Gamma_{Ш}^{(1)} \oplus T_{О}^{(1)},$$

где $T_{Ш}^{(1)} = \left(\tau_1^{(1)}, \tau_2^{(1)}, \ldots, \tau_{63}^{(1)}, \tau_{64}^{(1)}\right)$.

Блок шифрованных данных $T_{Ш}^{(i)}$ одновременно является также исходным состоянием накопителей N_1 и N_2 для выработки второго блока гаммы шифра $\Gamma_{Ш}^{(2)}$, и поэтому по обратной связи $T_{Ш}^{(1)}$ записывается в указанные накопители N_1 и N_2.

Заполнение накопителя N_1:

$$\left(\tau_{32}^{(1)}, \quad \tau_{31}^{(1)}, \quad \ldots, \quad \tau_2^{(1)}, \quad \tau_1^{(1)}\right)$$

$$32, \quad 31, \quad \ldots, \quad 2, \quad 1 \qquad \leftarrow \textit{номер разряда}\, N_1.$$

Заполнение накопителя N_2:

$$\left(\tau_{64}^{(1)}, \quad \tau_{63}^{(1)}, \quad \ldots, \quad \tau_{34}^{(1)}, \quad \tau_{33}^{(1)}\right)$$

$$32, \quad 31, \quad \ldots, \quad 2, \quad 1 \qquad \leftarrow \textit{номер разряда}\, N_1.$$

Заполнение накопителей N_1 и N_2 шифруется в режиме простой замены. Полученное в результате шифрования заполнение накопителей N_1 и N_2 образует второй 64-разрядный блок гаммы шифра $\Gamma_{Ш}^{(2)}$, который суммируется поразрядно по модулю 2 в сумматоре СМ со вторым блоком открытых данных $T_{О}^{(2)}$:

$$T_{Ш}^{(2)} = \Gamma_{Ш}^{(2)} \oplus T_{О}^{(2)}.$$

Выработка последующих блоков гаммы шифра $\Gamma_{Ш}$ и шифрование соответствующих блоков открытых данных $T_{О}^{(i)}$ $\left(i = 3 \ldots m\right)$ производятся аналогично. Если длина последнего m-го блока открытых данных $T_{О}^{(m)}$ меньше 64 разрядов, то из $\Gamma_{Ш}^{(m)}s$ используется только соответствующее число разрядов гаммы шифра, остальные разряды отбрасываются.

В канал связи или память ЭВМ передаются синхропосылка $\tilde{S}$ и блоки шифрованных данных $T_{Ш}^{(1)}, T_{Ш}^{(2)}, \ldots, T_{Ш}^{(m)}$.

Расшифрование в режиме гаммирования с обратной связью

При расшифровании криптосхема имеет тот же вид, что и при шифровании (см. рис. 3).

Уравнения расшифровывания:

$$T_{О}^{(1)} = A\left(\tilde{S}\right) \oplus T_{Ш}^{(1)} = \Gamma_{Ш}^{(1)} \oplus T_{Ш}^{(1)},$$

$$T_{О}^{(i)} = \Gamma_{Ш}^{(i)} \oplus T_{Ш}^{(i)} = A\left(T_{Ш}^{(i-1)}\right) \oplus T_{Ш}^{(i)},\ i = 2 \ldots m.$$

Реализация процедуры расшифрования шифрованных данных в режиме гаммирования с обратной связью происходит следующим образом. В КЗУ вводят 256 бит того же ключа, на котором осуществлялось шифрование открытых блоков $T_{О}^{(1)}, T_{О}^{(2)}, \ldots, T_{О}^{(m)}$. В накопители N_1 и N_2 вводится синхропосылка $\tilde{S}$. Исходное заполнение накопителей N_1 и N_2 (синхропосылка $\tilde{S}$) шифруется в режиме простой замены. Полученное в результате шифрования заполнение N_1 и N_2 образует первый блок гаммы шифра

$$\Gamma_{Ш}^{(1)} = A\left(\tilde{S}\right),$$

который суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ с блоком шифрованных данных $T_{Ш}^{(1)}$. В результате получается первый блок открытых данных

$$T_{О}^{(2)} = \Gamma_{Ш}^{(2)} \oplus T_{Ш}^{(2)}.$$

Блок шифрованных данных $T_{Ш}^{(1)}$ является исходным заполнением накопителей N_1 и N_2 для выработки второго блока гаммы шифра $\Gamma_{Ш}^{(2)}$: $\Gamma_{Ш}^{(2)} = A\left(T_{Ш}^{(1)}\right)$. Полученное заполнение накопителей N_1 и N_2 шифруется в режиме простой замены. Образованный в результате шифрования блок $\Gamma_{Ш}^{(2)}$ суммируется поразрядно по модулю 2 в сумматоре $СМ_5$ со вторым блоком зашифрованных данных $T_{Ш}^{(2)}$. В результате получают второй блок открытых данных. Аналогично в N_1 и N_2 последовательно записывают блоки шифрованных данных $T_{Ш}^{(2)}, T_{Ш}^{(3)}, \ldots, T_{Ш}^{(m)}$, из которых в режиме простой замены вырабатываются блоки гаммы шифра $\Gamma_{Ш}^{(3)}, \Gamma_{Ш}^{(4)}, \ldots, \Gamma_{Ш}^{(m)}$. Блоки гаммы шифра суммируются поразрядно по модулю 2 в сумматоре $СМ_5$ с блоками шифрованных данных $T_{Ш}^{(3)}, T_{Ш}^{(4)}, \ldots, T_{Ш}^{(m)}$.

В результате получают блоки открытых данных $T_{О}^{(3)}, T_{О}^{(4)}, \ldots, T_{О}^{(m)}$, при этом последний блок открытых данных $T_{О}^{(m)}$ может содержать меньше 64 разрядов.

4.4. Режим выработки имитовставки

Имитовставка - это блок из P бит, который вырабатывают по определенному правилу из открытых данных с использованием ключа и затем добавляют к шифрованным данным для обеспечения их имитозащиты.

Имитозащита - это защита системы шифрованной связи от навязывания ложных данных.

В стандарте ГОСТ 28147-89 определяется процесс выработки имитовставки, который единообразен для любого из режимов шифрования данных. Имитовставка $И_P s$ вырабатывается из блоков открытых данных либо перед шифрованием всего сообщения, либо параллельно с шифрованием по блокам. Первые блоки открытых данных, которые участвуют в выработке имитовставки, могут содержать служебную информацию (например, адресную часть, время, синхропосылку) и не шифруются.

Значение параметра P (число двоичных разрядов в имитовставке) определяется криптографическими требованиями с учетом того, что вероятность навязывания ложных помех равна $1/2^P$.

Для выработки имитовставки открытые данные представляют в виде последовательности 64-разрядных блоков $T_O^{(i)} s$, $i = 1 \ldots m$. Первый блок открытых данных $T_O^{(1)} s$ подвергают преобразованию $\tilde{A}(*)$, соответствующему первым 16 циклам алгоритма шифрования в режиме простой замены. В качестве ключа для выработки имитовставки используют ключ длиной 256 бит, по которому шифруют данные.

Полученное после 16 циклов 64-разрядное число $\tilde{A}\left(T_O^{(1)}\right) s$ суммируют по модулю 2 со вторым блоком открытых данных $T_O^{(2)}$. Результат суммирования $\tilde{A}\left(T_O^{(1)}\right) \oplus T_O^{(2)} s$ снова подвергают преобразованию $\tilde{A}(*)$.

Полученное 64-разрядное число $\tilde{A}\left(\tilde{A}\left(T_O^{(1)}\right) \oplus T_O^{(2)}\right)$ суммируют по модулю 2 с третьим блоком $T_O^{(3)}$ и снова подвергают преобразованию $\tilde{A}(*)$, получая 64-разрядное число $\tilde{A}\left(\tilde{A}\left(\tilde{A}\left(T_O^{(1)}\right) \oplus T_O^{(2)}\right) \oplus T_O^{(3)}\right)$, и т.д.

Последний блок $T_O^{(m)}$ (при необходимости дополненный нулями до полного 64-разрядного блока) суммируют по модулю 2 с результатом вычислений на шаге $(m-1)$, после чего шифруют в режиме простой замены, используя преобразование $\tilde{A}(*)$.

Из полученного 64-разрядного числа выбирают отрезок $И_P$ (имитовставку) длиной P бит:

$$И_P = \left[a_{32-P+1}^m(16), a_{32-P+2}^m(16), \ldots, a_{32}^m(16) \right],$$

где $a_i^{(m)}$ - i бит 64-разрядного числа, полученного после 16-го цикла последнего преобразования $\tilde{\mathrm{A}}(*)$, $32-\mathrm{P}+1 \le i \le 31$.

Имитовставка $\mathrm{И}_\mathrm{P}$ передается по каналу связи в конце шифрованных данных, т.е.

$$\mathrm{T}_\mathrm{Ш}^{(1)}, \mathrm{T}_\mathrm{Ш}^{(2)}, \ldots, \mathrm{T}_\mathrm{Ш}^{(m)}, \mathrm{И}_\mathrm{P}.$$

Поступившие к получателю шифрованные данные $\mathrm{T}_\mathrm{Ш}^{(1)}, \mathrm{T}_\mathrm{Ш}^{(2)}, \ldots, \mathrm{T}_\mathrm{Ш}^{(m)}$ расшифровываются, и из полученных блоков открытых данных $\mathrm{T}_\mathrm{O}^{(1)}, \mathrm{T}_\mathrm{O}^{(2)}, \ldots, \mathrm{T}_\mathrm{O}^{(m)}$ аналогичным образом вырабатывается имитовставка $\mathrm{И}'_\mathrm{P}$, которая сравнивается с $\mathrm{И}_\mathrm{P}$. В случае несовпадения блок открытых данных считается ложным.

5. СТАНДАРТ ШИФРОВАНИЯ ДАННЫХ DES

5.1. Обобщенная схема алгоритма DES

Алгоритм DES использует комбинацию подстановок и перестановок. DES осуществляет шифрование 64-битовых блоков данных с помощью 64-битого ключа, в котором значащими являются 56 бит (остальные 8 бит - проверочные биты для контроля на четность). Расшифрование в DES является операцией, обратной шифрованию, и выполняется путем повторения операций шифрования в обратной последовательности.

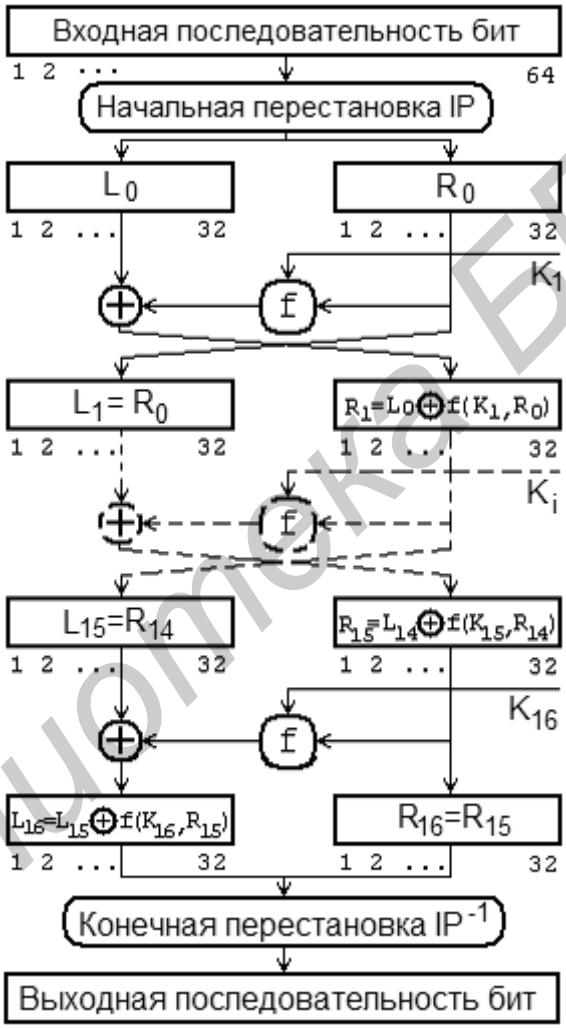

Рис. 5.1. Обобщенная схема шифрования в алгоритме DES

Обобщенная схема процесса шифрования в алгоритме DES (рис. 5.1) заключается в начальной перестановке бит 64-битого блока, шестнадцати циклах шифрования и, наконец, в конечной перестановке бит.

Следует отметить, что все приводимые таблицы являются стандартными и должны включаться в реализацию алгоритма DES в неизменном виде. Все перестановки и коды в таблицах подобраны разработ-

чиками таким образом, чтобы максимально затруднить процесс взлома шифра.

Пусть из файла исходного текста считан очередной 64-битовый блок T_O. Этот блок преобразуется с помощью матрицы начальной перестановки IP (табл. 5.1).

Таблица 5.1

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

Биты входного блока T (64 бита) переставляются в соответствии с матрицей IP: бит 58 входного блока T становится битом 1, бит 50 - битом 2 и т.д. Эту перестановку можно описать выражением $T_O = IP(T)$. Полученная последовательность бит T_O разделяется на две последовательности: L_0 - левые, или старшие, биты, R_0 - правые, или младшие, биты - каждая из которых содержит 32 бита.

Затем выполняется итеративный процесс шифрования, состоящий из 16 шагов (циклов). Пусть $T_i s$ - результат i-й итерации: $T_i = L_i R_i$, где $L_i = t_1 t_2 \ldots t_{32}$ (первые 32 бита); $R_i = t_{33} t_{34} \ldots t_{64}$ (последние 32 бита). Тогда результат i-й итерации описывается следующими формулами:

$$L_i = R_{i-1},\ i = 1,2,\ldots,16;$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i),\ i = 1,2,\ldots,16.$$

Функция f называется функцией шифрования. Ее аргументами являются последовательность R_{i-1}, получаемая на предыдущем шаге итерации, и 48-битовыйключ K_i, который является результатом преобразования 64-битого ключа шифра K. (Подробнее функция шифрования f и алгоритм получения ключа K описаны ниже.)

На последнем шаге итерации получают последовательности R_{16} и L_{16} (без перестановки местами), которые конкатенируются в 64-битую последовательность $R_{16}L_{16}$.

По окончании шифрования осуществляется восстановление позиций бит с помощью матрицы обратной перестановки IP^{-1} (табл. 5.2).

Таблица 5.2

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

Процесс расшифрования данных является инверсным по отношению к процессу шифрования. Все действия должны быть выполнены в обратном порядке. Это означает, что расшифровываемые данные сначала переставляются в соответствии с матрицей IP^{-1}, а затем над последовательностью бит $R_{16}L_{16}$ выполняются те же действия, что и в процессе шифрования, но в обратном порядке.

Итеративный процесс расшифрования может быть описан следующими формулами:

$$R_{i-1} = L_i,\ i = 1,2,\ldots,16;$$

$$L_{i-1} = R_i \oplus f\left(L_i, K_i\right),\ i = 1,2,\ldots,16.$$

Таким образом, для процесса расшифрования с переставленным входным блоком $R_{16}L_{16}$ на первой итерации используется ключ K_{16}, на второй итерации - K_{15} и т.д. На 16-й итерации используется ключ K_1. На последнем шаге итерации будут получены последовательности L_0 и R_0, которые конкатенируются в 64-битую последовательность L_0R_0. Затем в этой последовательности 64 бита переставляются в соответствии с матрицей IP. Результат такого преобразования - исходная последовательность бит (расшифрованное 64-битовое значение).

5.2. Реализация функции шифрования

Схема вычисления функции шифрования $f\left(R_{i-1}, K_i\right)$ показана на рис. 5.2.

Для вычисления значения функции f используются:

- функция E (расширение 32 бит до 48);
- функция $S_1, S_2, \ldots, S_8$ (преобразование 6-битового числа в 4-битовое);
- функция P (перестановка бит в 32-битовой последовательности).

Приведем определения этих функций.

Аргументами функции шифрования f являются R_{i-1} (32 бита) и K_i (48 бит). Результат функции $E\left(R_{i-1}\right)$ есть 48-битовое число. Функция

расширения E, выполняющая расширение 32 бит до 48 (принимает блок из 32 бит и порождает блок из 48 бит), определяется табл. 5.3.

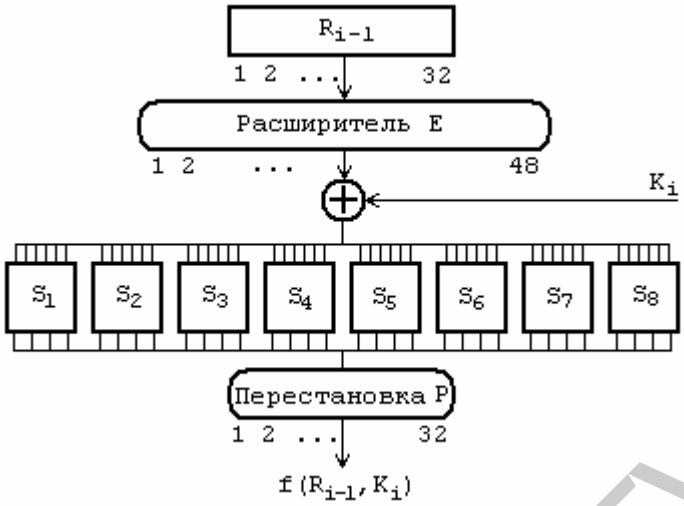

Рис. 5.2. Схема вычисления функции шифрования f

В соответствии с табл. 5.3 первые три бита $E(R_{i-1})$ - это биты 32, 1 и 2, а последние - 31, 32 и 1. Полученный результат (обозначим его $E(R_{i-1})$) складывается по модулю 2 с текущим значением ключа K_i и затем разбивается на восемь 6-битовых блоков $B_1, B_2, \ldots, B_8 = E(R_{i-1}) \oplus K_i$. Далее каждый из этих блоков используется как номер элемента в функциях - матрицах $S_1, S_2, \ldots, S_8$, содержащих 4-битовые значения (табл. 5.4).

Таблица 5.3

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

Следует отметить, что выбор элемента в матрице S осуществляется достаточно оригинальным образом. Пусть на вход матрицы S поступает 6-битовый блок $B_j = b_1 b_2 b_3 b_4 b_5 b_6$, тогда 2-битовое число $b_1 b_6$ указывает номер строки матрицы, а 4-битовое число $b_2 b_3 b_4 b_5$ - номер столбца. Например, если на вход матрицы S_1 поступает 6-битовый блок $B_1 = b_1 b_2 b_3 b_4 b_5 b_6 = 100110_{(2)}$, то 2-битовое число $b_1 b_6 = 10_{(2)} = 2_{(2)}$ указывает

строку с номером 2 матрицы S_1, а 4-битовое число $b_2b_3b_4b_5 = 0011_{(2)} = 3_{(10)}$ указывает столбец с номером 3 матрицы S_1. Это означает, что в матрице S_1 блок $B_1 = 100110$ выбирает элемент на пересечении строки с номером 2 и столбца с номером 3, т.е. элемент $8_{(10)} = 1000_{(2)}$. Совокупность 6-битовых блоков $B_1, B_2, \ldots, B_8$ обеспечивает выбор 4-битового элемента в каждой из матриц $S_1, S_2, \ldots, S_8$.

Таблица 5.4

Функции S																	
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_1	**0**	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	4	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	**0**	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	**0**	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	**0**	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	**0**	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	**0**	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	**0**	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	**0**	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

В результате получаем $S_1(B_1), S_2(B_1), \ldots, S_8(B_1)$, т.е. 32-битовый блок (поскольку матрицы S, содержат 4-битовые элементы). Этот 32-битовый блок преобразуется с помощью функции перестановки бит P (табл. 5.5).

Таблица 5.5

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

Таким образом, функция шифрования

$$f(R_{i-1}, K_i) = P(S_1(B_1), \ldots, S_8(B_1)).$$

5.3. Алгоритм вычисления ключей

Как нетрудно заметить, на каждой итерации используется новое значение ключа K_i (длиной 48 бит). Новое значение ключа K_i вычисляется из начального ключа K (рис. 5.3).

Ключ K представляет собой 64-битовый блок с 8 битами контроля по четности, расположенными в позициях 8, 16, 24, 32, 40, 48, 56, 64. Для удаления контрольных бит и подготовки ключа к работе используется функция G первоначальной подготовки ключа (табл. 5.6).

Таблица 5.6

Функция G							
C_0	57	49	41	33	25	17	9
	1	58	50	42	34	26	18
	10	2	59	51	43	35	27
	19	11	3	60	52	44	36
D_0	63	55	47	39	31	23	15
	7	62	54	46	38	30	22
	14	6	61	53	45	37	29
	21	13	5	28	20	12	4

Таблица 5.6 разделена на две части. Результат преобразования $G(K)$ разбивается на две половины C_0 и D_0, по 28 бит каждая. Первые четыре

строки матрицы G определяют, как выбираются биты последовательности C (первым битом C_0 будет бит 57 ключа шифра, затем бит 49 и т.д., а последними битами - биты 44 и 36 ключа).

Следующие четыре строки матрицы G определяют, как выбираются биты последовательности D_0 (т.е. последовательность D_0 будет состоять из бит 63, 55, 47,...,12, 4 ключа шифра).

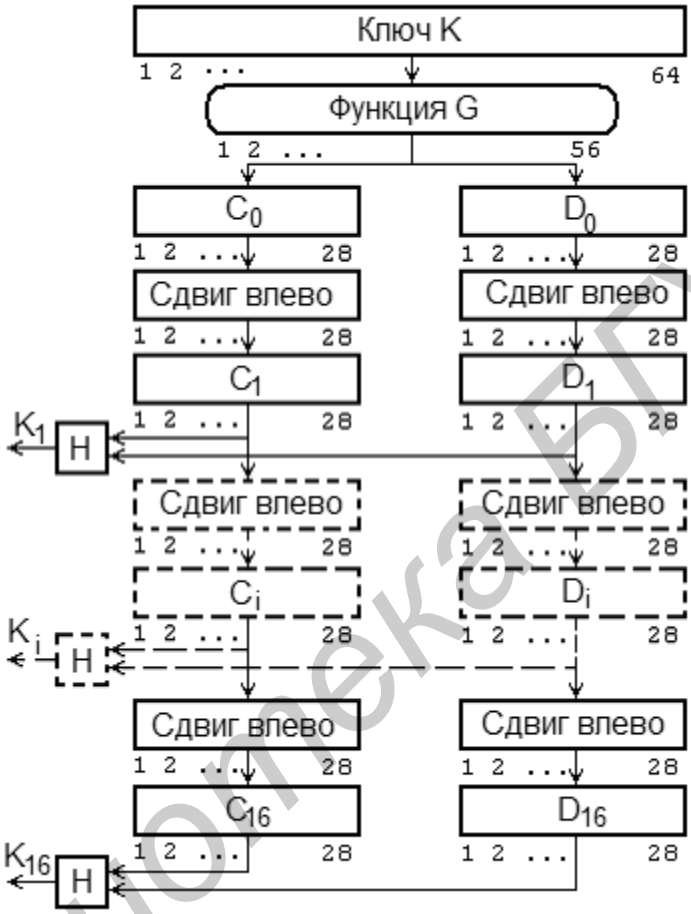

Рис. 5.3. Схема алгоритма вычисления ключей K_i

Как видно из табл. 5.6, для генерации последовательностей C_0 и D_0 не используются биты 8, 16, 24, 32, 40, 48, 56 и 64 ключа шифра. Эти биты не влияют на шифрование и могут служить для других целей (например, для контроля по четности). Таким образом, в действительности ключ шифра является 56-битовым.

После определения C_0 и D_0 рекурсивно определяются C_i и D_i, $i = 1,2,\ldots,16$. Для этого применяются операции циклического сдвига влево на один или два бита в зависимости от номера шага итерации, как показано в табл. 5.7.

Таблица 5.7

Таблица сдвигов для вычисления ключа																
Итерация	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сдвиг влево	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Операции сдвига выполняются для последовательностей C_i и D_i независимо. Например, последовательность C_3 получается посредством циклического сдвига влево на две позиции последовательности C_2, а последовательность D_3 - посредством сдвига влево на две позиции последовательности D_2, C_{16} и D_{16} получаются из C_{15} и D_{15} посредством сдвига влево на одну позицию.

Ключ K_i, определяемый на каждом шаге итерации, есть результат выбора конкретных бит из 56-битовой последовательности C_iD_i и их перестановки. Другими словами, ключ $K_i = H(C_iD_i)$, где функция H определяется матрицей, завершающей обработку ключа (табл. 5.8).

Таблица 5.8

Функция H					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	22	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Как следует из табл. 5.8, первым битом ключа K_i будет 14-й бит последовательности C_iD_i, вторым - 17-й бит, 47-м битом ключа K_i будет 29-й бит C_iD_i, а 48-м битом - 32-й бит C_iD_i.

5.4. Основные режимы работы алгоритма DES

Чтобы воспользоваться алгоритмом DES для решения разнообразных криптографических задач, разработаны четыре рабочих режима:

1. электронная кодовая книга ECB (Electronic Code Book);
2. сцепление блоков шифра CBC (Cipher Block Chaining);
3. обратная связь по шифротексту CPB (Cipher FeedBack);
4. обратная связь по выходу OFB (Output FeedBack).

Режим "Электронная кодовая книга"

Длинный файл разбивают на 64-битые отрезки (блоки) по 8 байт. Каждый из этих блоков шифруют независимо с использованием одного и того же ключа шифрования (рис. 5.4).

Основное достоинство - простота реализации. Недостаток – относительно слабая устойчивость против квалифицированных криптоаналитиков. Из-за фиксированного характера шифрования при ограниченной длине блока 64 бита возможно проведение криптоанализа "со словарем". Блок такого размера может повториться в сообщении вследствие большой избыточности в тексте на естественном языке. Это приводит к тому, что идентичные блоки открытого текста в сообщении будут представлены идентичными блоками шифротекста, что дает криптоаналитику некоторую информацию о содержании сообщения.

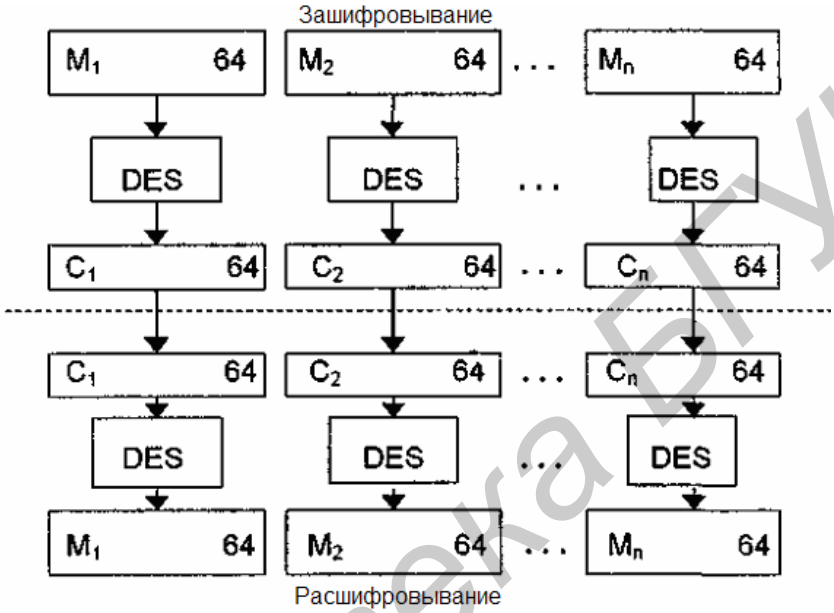

Рис. 5.4. Схема алгоритма DES в режиме электронной кодовой книги

Режим "Сцепление блоков шифра"

В этом режиме исходный файл M разбивается на 64-битые блоки: $M=M_1M_2\ldots M_n$. Первый блок M_1 складывается по модулю 2 с 64-битовым начальным вектором IV, который меняется ежедневно и держится в секрете (рис. 5.5). Полученная сумма затем шифруется с использованием ключа DES, известного и отправителю, и получателю информации. Полученный 64-битовый шифр C_1 складывается по модулю 2 со вторым блоком текста, результат шифруется и получается второй 64-битовый шифр C_2 и т.д. Процедура повторяется до тех пор, пока не будут обработаны все блоки текста.

Таким образом, для всех $i=1\ldots n$ (n - число блоков) результат шифрования C определяется следующим образом: $C_i=\mathrm{DES}(M_i\oplus C_{i-1})$, где $C_0=\mathrm{IV}$ - начальное значение шифра, равное начальному вектору (вектору инициализации).

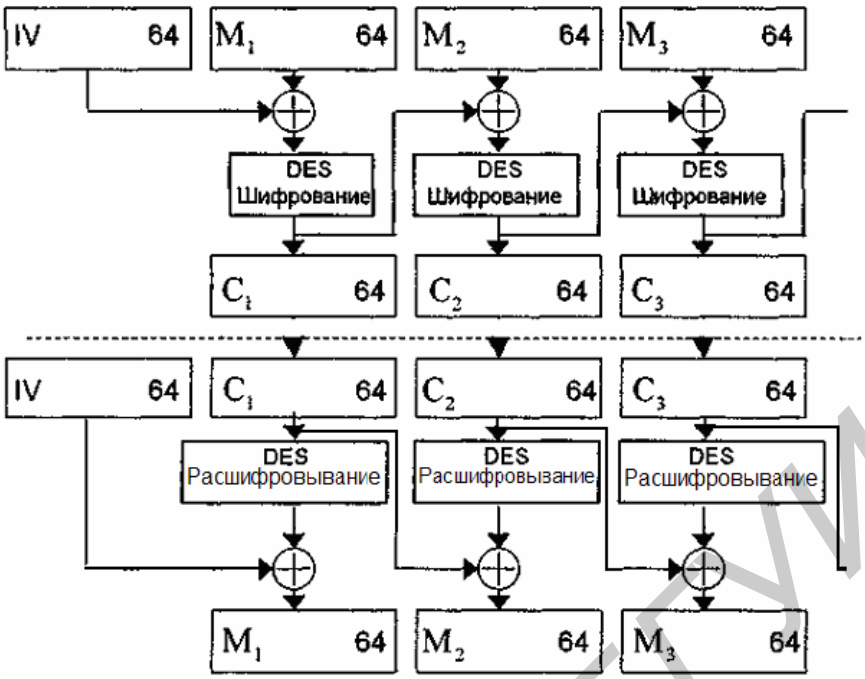

Рис. 5.5. Схема алгоритма DES в режиме сцепления блоков шифра

Очевидно, что последний 64-битовый блок шифротекста является функцией секретного ключа, начального вектора и каждого бита открытого текста независимо от его длины. Этот блок шифротекста называют кодом аутентификации сообщения (КАС).

Код КАС может быть легко проверен получателем, владеющим секретным ключом и начальным вектором, путем повторения процедуры, выполненной отправителем. Посторонний, однако, не может осуществить генерацию КАС, который воспринялся бы получателем как подлинный, чтобы добавить его к ложному сообщению, либо отделить КАС от истинного сообщения для использования его с измененным или ложным сообщением.

Достоинство данного режима в том, что он не позволяет накапливаться ошибкам при передаче.

Блок M_i является функцией только C_{i-1} и C_i. Поэтому ошибка при передаче приведет к потере только двух блоков исходного текста.

Режим "Обратная связь по шифротексту"

В этом режиме размер блока может отличаться от 64 бит (рис 5.6). Файл, подлежащий шифрованию (расшифрованию), считывается последовательными блоками длиной k бит ($k = 1 \ldots 64$).

Входной блок (64-битовый регистр сдвига) вначале содержит вектор инициализации, выровненный по правому краю. Предположим, что в результате разбиения на блоки мы получили n блоков длиной k бит каждый (остаток дописывается нулями или пробелами). Тогда для любого $i = 1 \ldots n$ блок шифротекста $C_i = M_i \oplus P_{i-1}$, где P_{i-1} обозначает k старших бит предыдущего зашифрованного блока.

Обновление сдвигового регистра осуществляется путем удаления его старших k бит и записи C_i в регистр. Восстановление зашифрованных данных также выполняется относительно просто: P_{i-1} и C_i вычисляются аналогичным образом и $M_i = C_i \oplus P_{i-1}$.

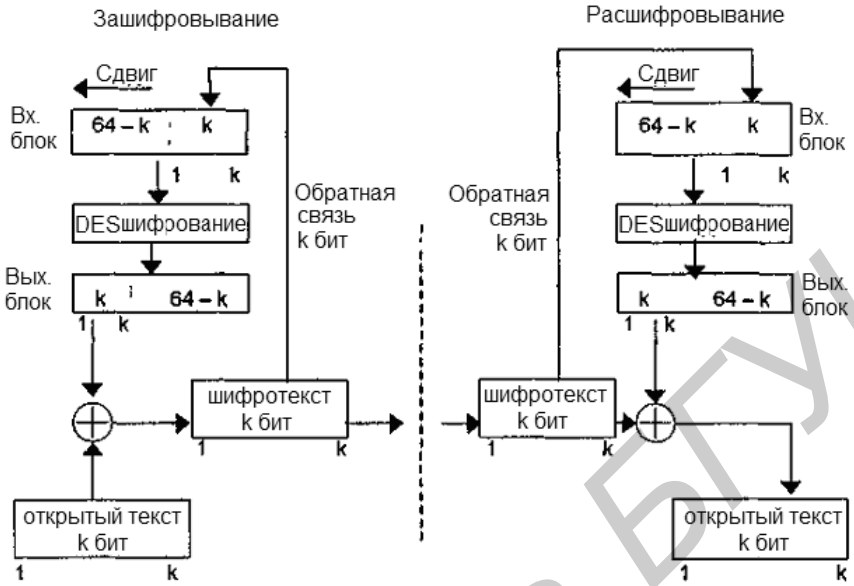

Рис. 5.6. Схема алгоритма DES в режиме обратной связи по шифротексту

Режим "Обратная связь по выходу"

Этот режим тоже использует переменный размер блока и сдвиговый регистр, инициализируемый так же, как в режиме СРВ, а именно - входной блок вначале содержит вектор инициализации IV, выровненный по правому краю (рис. 5.7). При этом для каждого сеанса шифрования данных необходимо использовать новое начальное состояние регистра, которое должно пересылаться по каналу открытым текстом.

Положим $M=M_1M_2\ldots M_n$ для всех $i=1\ldots n$ $C_i = M_i \oplus P_i$, где P_i - старшие k бит операции $DES(C_{i-1})$. Отличие от режима обратной связи по шифротексту состоит в методе обновления сдвигового регистра.

Это осуществляется путем отбрасывания старших k бит и дописывания справа P_i.

6. АСИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

6.1. Концепция криптосистемы с открытым ключом

Эффективными системами криптографической защиты данных являются асимметричные криптосистемы, называемые также криптосистемами с открытым ключом. В таких системах для шифрования данных используется один ключ, а для расшифрования другой (отсюда и название - асимметричные). Первый ключ является открытым и может быть опубликован для использования всеми пользователями системы, которые зашифровывают данные. Расшифрование данных с помощью открытого ключа невозможно. Для расшифровывания данных получатель шифрованной информации использует второй ключ, который является секретным. Разумеется, ключ расшифровывания не может быть определен из ключа шифрования.

Обобщенная схема асимметричной криптосистемы с открытым ключом показана на рис. 6.1.

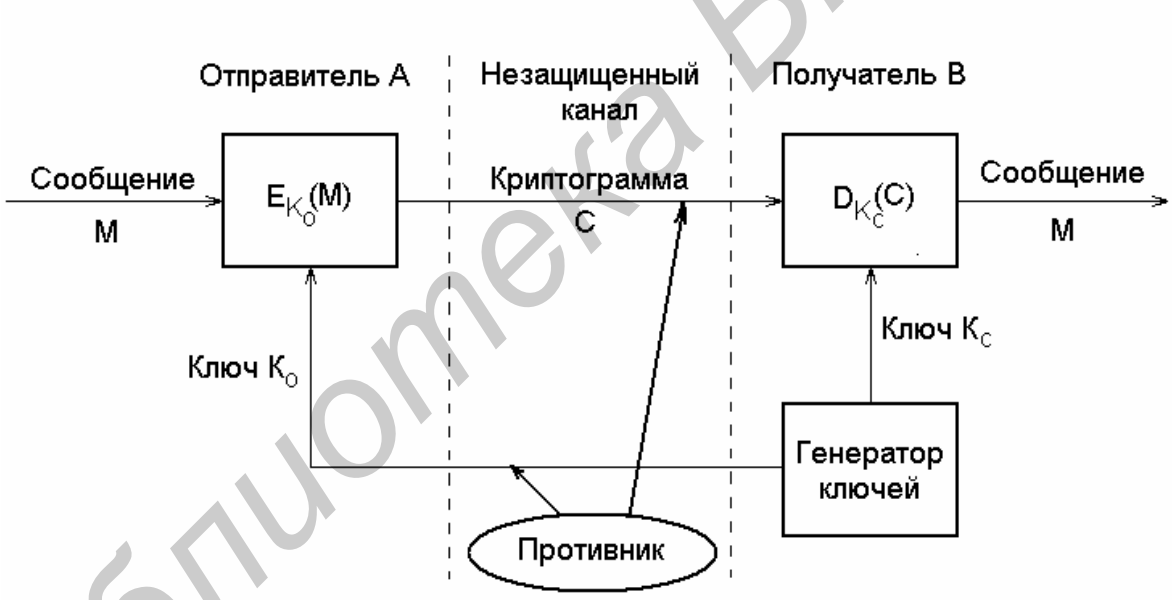

Рис. 6.1. Обобщенная схема асимметричной криптосистемы с открытым ключом

В этой криптосистеме применяют два различных ключа: K_O - открытый ключ отправителя A; K_C - секретный ключ получателя B. Генератор ключа целесообразно располагать на стороне получателя B (чтобы не пересылать секретный ключ K_C по незащищенному каналу). Значения ключей K_O, K_C – зависят от начального состояния генератора ключей.

Раскрытие секретного ключа K_C по известному ключу K_O должно быть вычислительно неразрешимой задачей.

Характерные особенности асимметричных криптосистем:

1. Открытый ключ K_O и криптограмма С могут быть отправлены по незащищенному каналу, т.е. могут быть известны противнику.
2. Алгоритмы шифрования $E_{K_O}(M) \to C$ и расшифрования $D_{K_C}(C) \to M$ являются открытыми.
3. Защита информации в асимметричной криптосистеме основана на секретности ключа K_C.

У.Диффи и М.Хелман сформулировали требования, выполнение которых обеспечивает безопасность асимметричной криптосистемы:

1. Вычисление пары ключей (K_O, K_C) получателем В на основе начального условия должно быть простым.
2. Отправитель А, зная открытый ключ K_O и сообщение М, может легко вычислить криптограмму $C = E_{K_O}(M)$.
3. Получатель В, используя секретный ключ K_C и криптограмму С, может легко восстановить исходное сообщение $M = D_{K_C}(C) = D_{K_C}\left(E_{K_O}(M)\right)$.
4. Противник, зная открытый ключ K_C, при попытке вычислить секретный ключ K_C наталкивается на непреодолимую вычислительную проблему.
5. Противник, зная пару (K_O, C), при попытке вычислить исходное сообщение М, наталкивается на непреодолимую вычислительную проблему.

6.2. Однонаправленные функции

Концепция асимметричных криптосистем с открытым ключом основана на применении однонаправленных функций. Однонаправленную функцию (ОФ) можно определить следующим образом.

Пусть X и Y - некоторые производные множества. Функция $f : X \to Y$ является однонаправленной, если для всех $x \in X$ можно легко вычислить функцию $y = f(x)$, где $y \in Y$. И в тоже время для большинства $y \in Y$ достаточно сложно получить значение $x \in X$, такое, что $f(x) = y$ (при этом полагаем, что существует по крайней мере одно такое значение x).

Основным критерием отнесения функции f к классу ОФ является отсутствие эффективных алгоритмов обратного преобразования $Y \to X$. Примеры однонаправленных функций.

1. Целочисленное произведение двух больших чисел.

Прямое преобразование - вычисление произведения двух очень больших чисел P и Q, т.е. нахождение значения $N = P \cdot Q$, является относительно несложной задачей для ЭВМ. Обратное преобразование -

разложение на множители большого целого числа, т.е. нахождение деталей P и Q большого $N = P \cdot Q$, является практически неразрешенной задачей при достаточно больших значениях N. По оценке теории чисел для разложения целого $N \approx 2^{664}$ потребуется около 10^{23} операций т.е. задача практически неразрешима для ЭВМ.

2. Модульная экспонента с фиксированным основанием и модулем.

Пусть A и N - целые числа, такие, что $1 \le A \le N$. Модульная экспонента с основанием A по модулю N представляет собой функцию $f_{A,N}(x) = A^x \bmod N$, где x - целое число, $1 \le x \le N-1$. Существуют эффективные алгоритмы, позволяющие достаточно быстро вычислить значение функции $f_{A,N}(x)$. Обратное преобразование, т.е. нахождение x из соотношения $A^x \bmod N = Y$ представляет собой трудновыполнимую задачу, т.к. для $y = A^x$ существует обратная функция $x = \log_A Y$, то часто нахождение аргумента x по известным y, A и N называют задачей дискретного логарифмирования. Следует иметь ввиду, что $y \in Z_n$, где $Z_n = \{1, 2, \ldots, N-1\}$.

По оценкам теории чисел при целых числах порядка $A \approx 2^{664}$ и $N \approx 2^{664}$ решением задачи дискретного логарифмирования (нахождение показателя степени x для известного y) потребуется 10^{26} операций. Т.е. для модульной экспоненты на 10^3 сложнее вычислять обратное преобразование, чем для целочисленного произведения. Однако до сих пор не доказано, что не существует эффективного логарифма за приемлемое время. Тем не менее модульная экспонента отнесена к однонаправленным функциям условно и широко используется на практике.

Кроме однонаправленных функций рассмотренного типа применяются однонаправленные функции с секретом (потайным ходом). Функция $f : X \to Y$ относится к классу ОФ с секретом в том случае, если она является однонаправленной и, кроме того, возможно эффективное вычисление обратной функции, если известен “потайной ход” (секретное число, строка, или другая информация о данной функции).

6.3. Элементы теории чисел

Определения

Число a называется простым, если оно не имеет других натуральных делителей, кроме 1 и a.

Например, 17, 23.

Числа a и b называются взаимно простыми, если наибольший общий множитель этих чисел $(a, b) = 1$.

Например: 8 и 9.

Модулярная арифметика

В модулярной арифметике все арифметические действия выполняются как в обычной арифметике с учетом того, что получаемые числа не могут превышать некоторой величины называемой модулем.

$(3+14)\bmod 12 = 5$

$(3+14) \equiv 5 \bmod 12$

В общем случае $a \equiv r \bmod n$. Читается a сравнимо с r по модулю n. Это справедливо, если $a = n \cdot k + r$, где $k = 0,1,2,\ldots$. Отсюда $r = a - n \cdot k$ называется вычетом числа a по модулю n, $(0 \le r < n)$. Справедливо:

$(a \pm b) \bmod n = (a \bmod n \pm b \bmod n) \bmod n$;

$(a \cdot b) \bmod n = (a \bmod n \cdot b \bmod n) \bmod n$;

$a \cdot (b \pm c) \bmod n = ((a \cdot b) \bmod n \pm (a \cdot c) \bmod n) \bmod n$.

Использование модулярной арифметики позволяет оперировать с очень большими числами, например, при возведении в степень:

$$a^8 \bmod n = \left(\left(a^2 \bmod n \right)^2 \bmod n \right)^2 \bmod n .$$

Малая теорема Ферма

Если n - простое и Í Î Ä$(a, n) = 1$, то

$$a^{n-1} \equiv 1 \bmod n .$$

Функция Эйлера

Количество положительных целых, меньших n, которые взаимно просты с n, определяется с помощью функции Эйлера $\varphi(n)$:

Модуль	n простое	n^2	n^m	$p \cdot q$ (p и q простые)
$\varphi(n)$	$n-1$	$n(n-1)$	$n^{m-1}(n-1)$	$(p-1)\cdot(q-1)$

Обобщение Эйлера малой теоремы Ферма: если Í Î Ä$(a, n) = 1$, то

$$a^{\varphi(n)} \equiv 1 \bmod n .$$

Нахождение обратных величин

Если задано уравнение $(a \cdot x) \bmod n = 1$, то величина $a^{-1} \equiv x \bmod n$ называется обратной величиной a по модулю n.

Обратная величина существует, если a и n – взаимно простые числа.

Способы нахождения обратных чисел

1. Перебором возможных значений.

Подставляя вместо x числа: $1,2,\ldots,n-1$ – добиваемся выполнения исходного уравнения.

Пример 6.1. $(5\cdot x)\bmod 7=1$, $x=\left(5^{-1}\right)\bmod 7=3$, т.к. $(5\cdot 3)\bmod 7=15\bmod 7=(15-7\cdot 2)\bmod 7=1$.

2. С помощью функции Эйлера $\varphi(n)$. $\left(a^{-1}\right)\bmod n=\left(a^{\varphi(n)-1}\right)\bmod n$.

Пример 6.2.

$$x=\left(5^{-1}\right)\bmod 7=\left(5^{6-1}\right)\bmod 7=\left(\left(5^2\right)\bmod 7\cdot\left(5^3\right)\bmod 7\right)\bmod 7=(4\cdot 6)\bmod 7=3.$$

3. С помощью алгоритма Евклида.

Алгоритм Евклида применяется для нахождения НОД чисел a и b. Однако его расширенный вариант можно использовать и для вычисления обратной величины.

Основной вариант.

Даны a и b, $(a>b)$. Алгоритм имеет итерационный характер:

$a=b\cdot q_1+r_1,\ 0<r_1<b;$

$b=r_1\cdot q_2+r_2,\ 0<r_2<r_1;$

$r_1=r_2\cdot q_3+r_3,\ 0<r_3<r_2;$

$\vdots$

$r_{k-2}=r_{k-1}\cdot q_k+r_k,\ 0<r_k<r_{k-1};$

$r_{k-1}=r_k\cdot q_{k+1},\ r_{k+1}=0;$

$(a,b)=r_k,$

где q_i, r_i - частное и остаток на i-м шаге алгоритма. На первом шаге делимое - a, делитель - b, частное - q_1, остаток - r_1. На i-м, $i>1$ шаге алгоритма: делимое - делитель $i-1$-го шага, делитель - остаток $i-1$-го шага (r_{i-1}), частное - q_i, остаток - r_i.

Пример 6.3. Пусть $a=1071$ и $b=693$. Найти НОД(a,b).

$$1071=693\cdot 1+378,\ \text{т.е. } q_1=1,\ r_1=378;$$

$$693=378\cdot 1+315,\ \text{т.е. } q_2=1,\ r_2=315;$$

$$378=315\cdot 1+63,\ \text{т.е. } q_3=1,\ r_3=63;$$

$$315=63\cdot 5,\ \text{т.е. } q_4=5,\ r_4=0.$$

То есть на четвертом шаге остаток от деления $r_4=0$, следовательно, алгоритм останавливается и НОД$(a,b)=63$.

Доказано, что при неотрицательных a и b можно найти такие целые числа: u_1, u_2, u_3, что будет выполняться

$$a\cdot u_1+b\cdot u_2=u_3=(a,b).$$

Если выбрать $b = n$ и a, n - взаимно простые числа, т.е. $(a, n) = 1$, тогда

$$a \cdot \mathrm{u}_1 + n \cdot \mathrm{u}_2 = 1,$$
$$(a \cdot \mathrm{u}_1 + n \cdot \mathrm{u}_2) \bmod n = (a \cdot \mathrm{u}_1) \bmod n = 1,$$
$$(a^{-1}) \bmod n = \mathrm{u}_1 \bmod n.$$

То есть для нахождения обратной величины необходимо вычислить $\mathrm{u}_1 \bmod n$. Эта задача решается в ходе вычисления НОД(a, n) в соответствии с алгоритмом Евклида. Дополнительно на каждом шаге вычисляются координаты двух векторов: $\vec{\mathrm{u}}=(\mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3)$, $\vec{v}(v_1, v_2, v_3)$.

Алгоритм вычисления u_1 имеет следующий вид

1. Начальные установки:

$\vec{\mathrm{u}}_0=(0,1,n)$, т.е. $\mathrm{u}_1 = 0$, $\mathrm{u}_2 = 1$, $\mathrm{u}_3 = n$. При этом $a \cdot 0 + b \cdot 1 = n$, т.е. $b = n$, $\vec{v}_0 = (1, 0, a)$, т.е. $v_1 = 1$, $v_2 = 0$, $v_3 = a$. При этом $a \cdot 1 + n \cdot 0 = a$.

2. Проверяем, выполняется ли $\mathrm{u}_3 = 1$, если да, то алгоритм заканчивается.

3. Делим n на a (u_3 на v_3) и определяем:

$q_1 = \left[\dfrac{\mathrm{u}_3}{v_3}\right]$ и значения векторов: $\vec{\mathrm{u}}_1 = \vec{v}_0$; $\vec{v}_1 = \vec{\mathrm{u}}_0 - q_1 \cdot \vec{v}_0$.

4. Вернуться к шагу 2.

На каждом шаге при расчетах используются результаты предыдущего:

$q_i = \left[\dfrac{\mathrm{u}_3}{v_3}\right]_{i-1}$, $\vec{\mathrm{u}}_i = \vec{v}_{i-1}$, $\vec{v}_i = \vec{\mathrm{u}}_{i-1} - q_i \cdot \vec{v}_{i-1}$.

При $\mathrm{u}_3 = 1$ вычисления заканчиваются $(a^{-1}) \bmod n = \mathrm{u}_1 \bmod n$, где u_1 значение u_1, полученное на последнем шаге.

Пример 6.4. Пусть $n = 23$ и $a = 5$. Найти число x, обратное числу a по модулю n, т.е. найти $5^{-1} \bmod 23$.

Используя расширенный алгоритм Евклида, выполним вычисления.

q	u_1	u_2	u_3	v_1	v_2	v_3
-	0	1	$n = 23$	1	0	$a = 5$
4	1	0	5	-4	1	3
1	-4	1	3	5	-1	2
1	5	-1	2	-9	2	1
-	-9	2	1			

При $\mathrm{u}_1 = -9$, $\mathrm{u}_2 = 2$, $\mathrm{u}_3 = 1$ выполняется уравнение $a \cdot \mathrm{u}_1 + n \cdot \mathrm{u}_2 = 1$, $a \cdot (-9) + n \cdot 2 = 5 \cdot (-9) + 23 \cdot 2 = 1$ и $a^{-1} \bmod n = 5^{-1} \bmod 23 = (-9) \bmod 23 = 14$.

Итак, $x = 5^{-1} \bmod 23 = (-9) \bmod 23 = 14$.

6.4. Криптосистема RSA

Последовательность действий абонентов криптосистемы RSA

Действия получателя криптограммы B:

1. B генерирует два произвольных больших простых числа P и Q. Эти числа должны быть примерно одинаковыми, размерностью 100-150 десятичных разрядов. Они должны быть секретными.

2. B вычисляет значение модуля $n = \mathrm{P} \cdot \mathrm{Q}$ и функции Эйлера $\varphi(n) = (\mathrm{P}\text{-}1) \cdot (\mathrm{Q}\text{-}1)$ и выбирает значение открытого ключа $\mathrm{K_O}$ с соблюдением условий: $1 < \mathrm{K_O} \le \varphi(n)$, $(\mathrm{K_O}, \varphi(n)) = 1$, т.е. $\mathrm{K_O}$ и $\varphi(n)$ должны быть взаимно простыми.

3. B вычисляет значение секретного ключа $\mathrm{K_C}$, используя расширенный алгоритм Евклида: $\mathrm{K_C} = \left(\mathrm{K_O^{-1}}\right) \bmod \varphi(n) s$.

4. B посылает A пару чисел $n, \mathrm{K_O}$ по открытому каналу.

Действия отправителя криптограммы A:

1. Разбивает исходный текст M на блоки M_i, $i = 1, 2, \ldots, m$, т.е. $\mathrm{M}=\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_m$. Величина $\mathrm{M}_i < n$.

2. Шифрует каждое число M_i по формуле $\mathrm{C}_i = \left(\mathrm{M}_i^{\mathrm{K_O}}\right) \bmod n$ и отправляет криптограмму $\mathrm{C}=\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_m$.

Получатель B, получив криптограмму, расшифровывает каждый блок секретным ключом $\mathrm{K_C}$, $\mathrm{M}_i = \left(\mathrm{C}_i^{\mathrm{K_C}}\right) \bmod n$, и восстанавливает весь текст $\mathrm{M}=\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_m$.

Реализуемость и безопасность RSA

Покажем, что при расшифровании восстанавливается исходный текст. Согласно обобщению Эйлером малой теоремы Ферма: если Í Î Ä$(a, n) = 1$, то $a^{\varphi(n)} \equiv 1 \bmod n$, или $a^{\varphi(n)+1} \equiv a \bmod n$. Открытый $\mathrm{K_O}$ и закрытый $\mathrm{K_C}$ ключи в алгоритме связаны соотношением $\mathrm{K_O} \cdot \mathrm{K_C} \equiv 1 \bmod \varphi(n)$, или $\mathrm{K_O} \cdot \mathrm{K_C} = k \cdot \varphi(n) + 1$ для некоторого целого k. Таким образом, процесс шифрования, а затем расшифрования некоторого сообщения M_i выглядит следующим образом:

$$\left(\left(\mathrm{M}_i^{\mathrm{K_O}}\right) \bmod n\right)^{\mathrm{K_C}} \bmod n = \left(\mathrm{M}_i^{\mathrm{K_O} \cdot \mathrm{K_C}}\right) \bmod n = \left(\mathrm{M}_i^{k \cdot \varphi(n)+1}\right) \bmod n = \mathrm{M}_i .$$

В процессе применения RSA злоумышленник может иметь: C_i, K_O, n – и организовать дешифрование двумя способами:

1. По C_i, K_O, n получить M_i. Для этого он решает задачу вычисления M_i из уравнения $C_i = M_i^{K_O} \bmod n$. Эта задача вычислительно трудна.

2. По n вычислить P, Q, затем найти $\varphi(n)$ и вычислить $K_C = \left(K_O^{-1}\right) \bmod \varphi(n)$ и дешифровать сообщение $M_i = C_i^{K_C} \bmod n$.

Однако задача разложения большого числа на простые множители вычислительно сложна.

Пользователи A и B должны быстро осуществлять все вычисления: вычислять K_O, шифровать и расшифровывать.

Вычисление K_O с использованием алгоритма Евклида - довольно быстрый процесс и не представляет трудности. Шифрование и расшифрование - возведение большого числа в большую степень - требует определенных затрат времени, но, с учетом наличия быстрых алгоритмов и быстродействия современных компьютеров, это приемлемая процедура.

6.5. Криптосистема Эль-Гамаля

Схема Эль-Гамаля, предложенная в 1985 г., может быть использована как для шифрования, так и для цифровых подписей. Безопасность схемы Эль-Гамаля обусловлена сложностью вычисления дискретных логарифмов в конечном поле.

Для того чтобы генерировать пару ключей (открытый ключ – секретный ключ), сначала выбирают некоторое большое простое число P и большое целое число G, причем $G < P$. Числа P и G могут быть распространены среди группы пользователей. Затем выбирают случайное целое число X, причем $X < P$. Число X является секретным ключом и должно храниться в секрете. Далее вычисляют $Y = G^X \bmod P$. Число Y является открытым ключом.

Для того чтобы зашифровать сообщение M, выбирают случайное целое число $1 < K < P - 1$ такое, что числа K и $(P-1)$ являются взаимно простыми. Затем вычисляют числа $a = G^K \bmod P$, $b = \left(Y^K \cdot M\right) \bmod p$. Пара чисел (a,b) является шифротекстом. Заметим, что длина шифротекста вдвое больше длины исходного открытого текста M.

Для того чтобы расшифровать шифротекст (a,b), вычисляют $M = \left(b / a^X\right) \bmod P$. Справедливость этого равенства следует из: $a^X \equiv G^{KX} \bmod P$, $b / a^X \equiv Y^K M / a^X \equiv G^{KX} M / G^{KX} \equiv M \bmod P$.

ЛИТЕРАТУРА

1. Закон Республики Беларусь от 6 сентября 1995 г. «Об информатизации».
2. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.
3. СТБ 1176.2-99 Информационная технология. Защита информации. Процедуры выработки и проверки электронной цифровой подписи.
4. СТБ 1176.1-99 Информационная технология. Защита информации. Функция хэширования.
5. Романец Ю. В., Тимофеев П. А., Шаньгин В. Ф. Защита информации в компьютерных системах и сетях. М.: Радио и связь, 1999.
6. Харин Ю. С., Берник В. И., Матвеев Г.В. Агиевич С.В. Основы криптологии. Мн.: Новое знание, 2003.
7. Шнайер Б. Прикладная криптография. М., 2001.
8. Зима В.М., Молдавян А.А., Молдовян Н.А. Безопасность глобальных сетевых технологий. СПб., 2001.
9. Мафтик С. Механизмы защиты в сетях ЭВМ. М.: Мир, 1993.
10. Голиков В.Ф., Лыньков Л.М., Прудник А.М., Борботько Т.В. Правовые и организационно-технические методы защиты информации: Учеб. пособие. Мн.: БГУИР, 2004.
11. Голиков В.Ф., Курилович А.В. Криптографическое кодирование информации: Метод. указания к лабораторным работам. Мн.: БГУИР, 2002.

Св. план 2006, поз. 105

Учебное издание

Голиков Владимир Федорович
Курилович Андрей Владимирович

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

Учебно-методическое пособие
для студентов специальностей
«Сети телекоммуникаций» и «Защита информации
в телекоммуникациях»
всех форм обучения

В 2-х частях

Часть 1

Ответственный за выпуск А.В. Курилович

Подписано в печать 12.06.2006. Формат 60x84 1/16. Бумага офсетная.
Гарнитура «Таймс». Печать ризографическая. Усл. печ. л. 3,37.
Уч.-изд. л. 3,2. Тираж 100 экз. Заказ 288.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0056964 от 01.04.2004. ЛП №02330/0131518 от 30.04.2004.
220013, Минск, П. Бровки, 6