Министерство образования и науки Российской Федерации
Сибирский федеральный университет

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И ЗАЩИТА ИНФОРМАЦИИ

Учебно-методическое пособие

Красноярск
СФУ
2014

УДК 004.4.056(07)
ББК 32.81я73

Составитель: А.Н. Шниперов

И741 Информационная безопасность и защита информации: учебно-методическое пособие [Текст] / сост. А.Н. Шниперов. – Красноярск: Сиб. федер. ун-т, 2012. – 73 с.

В учебно-методическом пособии содержатся методические указания к лабораторному практикуму, включающие в себя краткие теоретические сведения, описание и порядок выполнения 5 лабораторных работ по дисциплине «Информационная безопасность и защита информации».

Пособие предназначено студентов и магистрантов дневной и заочных форм обучения, обучающихся по направлениям подготовки специалистов в области ИТ-технологий. Кроме этого данное учебно-методическое пособие может быть полезно студентам других специальностей, изучающим вопросы обеспечения информационной безопасности.

УДК 004.4.056(07)
ББК 32.81я73

ОБЩИЕ УКАЗАНИЯ

В настоящих методических указаниях представлен лабораторный практикум, целью проведения которого является изучение основ криптографических методов защиты информации и простейших симметричных и ассиметричных алгоритмов шифрования, а также их уязвимостей. Кроме этого лабораторный практикум включает в себя работы, связанные с основами криптоанализа, на примере атак на классические шифры.

Все лабораторные работы включает в себя разработку алгоритмов и их реализаций с использованием **одного из языков** высокоуровневого программирования, например, *Microsoft* C#.

В указаниях к практикуму включены достаточно подробные теоретические сведения по конкретной проблематике.

Лабораторный практикум проводится каждым студентом индивидуально. Для этого студент получает одно из заданий по указанному преподавателем варианту (либо персональное задание).

При выполнении лабораторной работы студент должен предъявить файл с исходным кодом программы, реализующей поставленные в работе задачи, а также результаты работы программы. Задание считается выполненным, если оно соответствует варианту (либо персональному заданию преподавателя) и в полной мере реализует поставленную задачу.

Лабораторная работа считается выполненной после её защиты. Для защиты работы необходимо представить файл с результатами выполнения, а также отчёт, оформленный по указаниям действующего стандарта организации (СТО) «Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной и научной деятельности».

Последнюю пробную версию пакета Microsoft Visual C# всегда можно бесплатно скачать и использовать в учебных целях с официального сайта компании-разработчика Microsoft Corporation по адресу в сети интернет http://www.microsoft.com/express/Downloads/.

ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Криптография – это наука защиты информации путем ее преобразования в нечитаемый вид. Криптография является эффективным способом защиты критичной информации при ее хранении и передаче по открытым (не доверительным) каналам связи.

Одной из целей криптографии и лежащих в ее основе механизмов, является скрытие информации от неуполномоченных лиц. Однако злоумышленники, обладающие достаточным объемом времени, ресурсов и мотивации, могут взломать почти любой алгоритм и получить доступ к зашифрованной информации. Более реалистичной целью криптографии является попытка сделать взлом зашифрованной информации слишком сложной и длительной по времени задачей для злоумышленника, обладающего ограниченными ресурсами.

Первые методы шифрования появились около 4000 лет назад и основывались на графических методах. Позднее криптография была адаптирована для использования в военных, коммерческих, правительственных и других целях – там, где секреты нуждались в защите. Незадолго до появления Интернета, шифрование получило новое применение – оно стало важным инструментом для ежедневных операций. На протяжении всей истории люди и правительства работали над защитой передаваемой информации с помощью ее шифрования. В результате алгоритмы и устройства шифрования становились все более сложными, постоянно внедрялись новые методы и алгоритмы. В настоящее время шифрование стало неотъемлемой частью компьютерного мира.

Шифрование является одним из основных методом криптографии. Суть его заключается в преобразования читаемых данных, называемых открытым текстом (*plaintext*), в форму, которая выглядит случайной и нечитаемой, она называется шифротекстом (*chiphertext*). Открытый текст – это форма текста, понятная каждому человеку (документ) или компьютеру (исполняемый код). После того, как открытый текст будет преобразован в шифротекст, ни человек, ни машина не смогут правильно обработать его до осуществления процесса расшифрования. Это позволяет передавать конфиденциальную информацию через незащищенные каналы, не опасаясь несанкционированного доступа к ней.

Системы или продукты, которые предоставляют функции зашифрования и расшифрования, называются **криптосистемами** (*cryptosystem*), они могут создаваться в виде аппаратных или программных компонентов. Криптосистемы используют алгоритмы шифрования (которые определяются как простые или сложные с точки зрения процесса шифрования), ключи, а также необходимые

программные компоненты и протоколы. У большинства алгоритмов базисом являются сложные математические выражения, которые в определенной последовательности применяются к открытому тексту. Большинство методов шифрования используют секретное значение, называемое ключом (обычно ключ представляет собой длинную последовательность битов), который используется в процессе работы алгоритмом для зашифрования и расшифровывания текста.

Алгоритм – это набор правил, определяющий процесс шифрования и дешифрования. Под понятием шифра очень часто понимается именно алгоритм шифрования, что, вообще говоря, не совсем верно. Тем не менее, именно он определяет сложность и тип шифра.

Многие математические алгоритмы, используемые сегодня в компьютерных системах, являются публично доступными и широко известными – процесс шифрования не является секретом. Раз внутренние механизмы алгоритма не секретны, значит секретным должно быть что-то другое. Секретной частью известного алгоритма шифрования является **ключ**, по аналогии с замком.

В шифровании, ключ – это значение, которое состоит из длинной последовательности случайных битов. На практике, однако, ключ представляет собой псевдослучайное значение, в силу того, что алгоритм использует ключевое пространство (*keyspace*), являющееся диапазоном значений, которые могут использоваться для создания ключа. Когда алгоритму нужно сгенерировать новый ключ, он использует случайные значения из этого ключевого пространства.

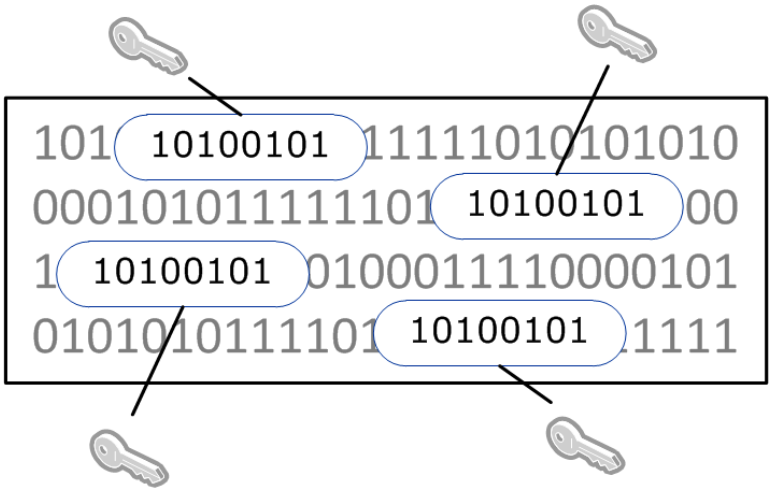

Рис. 1. Выбор ключа шифрования из ключевого пространства

Чем шире ключевое пространство, тем больше доступных значений можно использовать для создания ключа, а чем больше случайных вариантов ключей, тем сложнее злоумышленнику подобрать их. Например, если алгоритм позволяет использовать ключи длиной 2 бита, ключевое пространство для этого

алгоритма составляет всего 4 возможных значения, это максимальное количество возможных вариантов различных ключей для этого алгоритма. Это очень узкое ключевое пространство, поэтому атакующему не потребуется много времени, чтобы найти правильный ключ путём простого перебора и воспользоваться им. Поэтому алгоритмы шифрования должны использовать как можно более широкое ключевое пространство и выбирать значение для нового ключа максимально случайным образом.

Если злоумышленник перехватит сообщение, передаваемое между двумя людьми, он сможет просмотреть это сообщение, однако, если оно передается в зашифрованном виде, оно нечитаемое. Даже если злоумышленник знает алгоритм, используемый этими людьми для зашифрования и расшифровывания информации, без знания ключа эта информация будет бесполезной для злоумышленника

Стойкость шифров

Стойкость методов шифрования основывается на алгоритме, обеспечении секретности ключа, длине ключа, векторах инициализации, а также том, как все это работает вместе в рамках криптосистемы. Когда рассуждают о стойкости того или иного шифра в первую очередь подразумевают сложность вскрытия (нахождение уязвимых мест) алгоритма или ключа, даже если алгоритм не является публичным. Самый простой способ взлома криптосистемы заключается в переборе всех возможных значений ключа шифрования, чтобы найти именно то значение (ключ), которое позволяет расшифровать конкретное сообщение. Однако в силу того, что общее количество ключей может быть колоссально велико, такой способ взлома криптосистемы может занять много лет или даже десятилетий. Поэтому стойкость метода шифрования находится в прямой связи с величиной мощностей и количеством времени, необходимых для взлома криптосистемы (перебора всех возможных значений) и получения правильного ключа. В зависимости от алгоритма и длины ключа, это может быть как легкой задачей, так и практически невозможной.

Если ключ можно взломать за три часа на компьютере с современным процессором, то шифр считается абсолютно нестойким. Если ключ может быть взломан только при использовании многопроцессорной системы с тысячей процессоров за 1,2 миллионов лет, шифр считается очень стойким.

В процессе создания нового метода шифрования, основной задачей является создание условий, при которых взлом станет слишком дорогим или требующим слишком много времени. Синонимом стойкости криптосистемы является

фактор трудозатрат, который указывает на оценку необходимых ресурсов для взлома криптосистемы атакующим.

Следует отметить, что даже если алгоритм очень сложный и совершенный, другие проблемы шифрования могут ослабить его. Например, ослабить даже самый стойкий шифр может ненадлежащая защита секретного значения ключа.

Для обеспечения стойкости шифрования важно использовать алгоритмы, не имеющие недостатков, применять длинные ключи, при генерации ключей использовать весь возможный диапазон ключевого пространства, а также хорошо защищать ключ. Если хотя бы один из этих пунктов не выполняется, это может оказать негативное влияние на весь процесс.

Несмотря на то, что процесс шифрования охватывает массу составляющих, можно выделить две его основные части, которыми являются алгоритмы и ключи. Как было отмечено ранее, алгоритмы, используемые в компьютерных системах, содержат в себе сложные математические формулы, диктующие правила преобразования открытого текста в шифротекст и наоборот. Ключ является строкой случайных битов, которая используется алгоритмом для добавления случайности в процесс шифрования. Чтобы два субъекта могли взаимодействовать с использованием шифрования, они должны использовать один и тот же алгоритм и, в ряде случаев, один и тот же ключ. В некоторых технологиях шифрования получатель и отправитель используют один и тот же ключ, тогда как в других технологиях они должны использовать различные, но связанные ключи для зашифрования и расшифровывания информации.

Криптографические алгоритмы делятся на симметричные алгоритмы, которые используют симметричные ключи (также называемые секретными ключами (*secret key*)), и асимметричные алгоритмы, которые используют асимметричные ключи (называемые также открытыми (*public key*) и закрытыми ключами (*private key*)).

Симметричная криптография

Алгоритм, символическая запись которого представлена преобразованиями (1.1) называется криптосистемой с симметричным ключом, поскольку обе стороны, обменивающиеся шифрованной информацией, применяют один и тот же секретный ключ. Иногда симметричные криптосистемы используют два ключа: один для шифрования, а другой для обратного процесса. В этом случае полагается, что шифрующий ключ легко восстанавливается по расшифровыва-

ющему и наоборот. Работа симметричных шифров включает в себя два преобразования:

$$C = E_k(m) \text{ и } m = D_k(C),$$

где m – открытый текст, E – шифрующая функция, D – функция дешифрования, k – секретный ключ, C – шифротекст. Следует отметить, что как шифрующая, так и расшифровывающая функции общеизвестны, и тайна сообщения при известном шифротексте зависит только от секретности ключа k.

Число возможных ключей в симметричной криптосистеме должно быть очень велико. Это требование возникает в связи с тем, что при проектировании криптоалгоритма, необходимо учитывать самый плохой сценарий развития событий, т.е. считая, что гипотетический противник:

- обладает полнотой информации о шифрующем (расшифровывающем) алгоритме;
- имеет в своём распоряжении некоторое количество пар (открытый текст, шифротекст) ассоциированных с истинным ключом k.

Если количество возможных ключей мало, то атакующий имеет возможность взломать шифр простым перебором вариантов. Он может шифровать один из данных открытых текстов, последовательно используя разные ключи, до тех пор, пока не получит соответствующий известный шифротекст. Принято считать, что вычисления, состоящие из 2^{80} шагов, в ближайшее несколько лет будут неосуществимы.

На рис. 2 изображена упрощённая модель шифрования битовой строки, которая, несмотря на свою простоту, вполне подходит для практического применения. Идея модели состоит в применении к открытому тексту обратимой операции для получения шифротекста, а именно побитовое сложение по модулю 2 (⊕) открытого текста со «случайным потоком» битов. Получатель может восстановить текст с помощью обратной операции, сложив шифротекст с тем же самым случайным потоком.

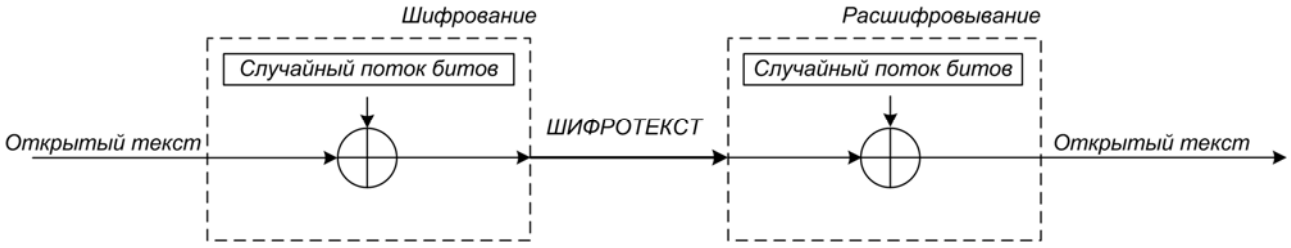

Рис. 2. Упрощенная модель, шифрующая строку битов

Такую модель легко реализовать на практике, поскольку для её реализации необходима одна из простейших компьютерных операций – исключающее

ИЛИ, т.е. сложение по модулю 2 (⊕). Следует отметить, что абсолютной стойкости шифр достигает тогда, когда каждое новое сообщение шифруется своим уникальным ключом, длина которого совпадает с длиной открытого текста.

Однако, несмотря на совершенство этого алгоритма, он не применяется на практике, поскольку порождает почти не разрешимую проблему распределения ключей. В связи с этим разрабатываются симметричные криптосистемы, в которых длинное сообщение шифруется коротким ключом, причём этот ключ можно использовать несколько раз. Естественно, такие системы далеки от абсолютно стойких, но, с другой стороны, распределение ключей для них – хотя и трудная, но вполне решаемая задача.

Асимметричная криптография

В отличие от симметричной криптографии, где для зашифрования и расшифровывания используется один и тот же секретный ключ, в ассиметричной криптографии (криптографии с открытыми ключами) для этих целей используются различные (асимметричные) ключи. При этом два отличающихся асимметричных ключа связаны между собой математически. Если сообщение зашифровано одним ключом, для его расшифровывания требуется другой ключ.

В ассиметричных криптосистемах, создается пара ключей, один из которых является закрытым, другой – открытым. Открытый ключ (*public key*) может быть известен всем, а закрытый ключ *(private key*) должен знать только его владелец (см. рис. 3). Часто открытые ключи хранятся в каталогах и базах данных адресов электронной почты, общедоступных всем желающим использовать эти ключи для зашифрования и расшифровывания данных при взаимодействии с отдельными людьми.

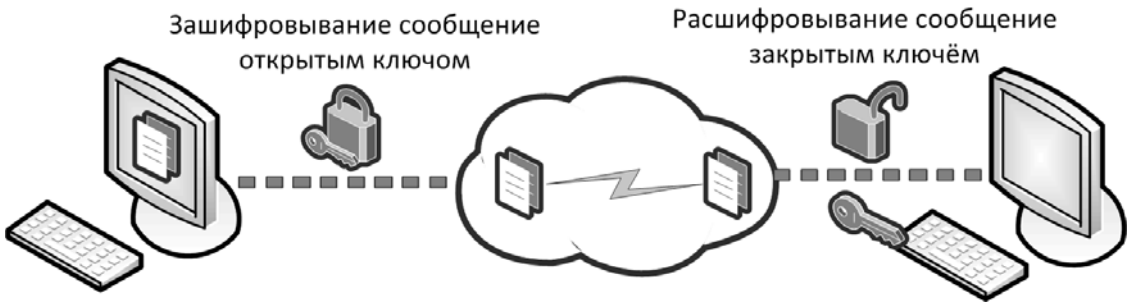

Рис. 3. Общий принцип ассиметричной криптосистемы

Открытый и закрытый ключи асимметричной криптосистемы математически связаны, однако наличие у кого-то открытого ключа другого человека не позволяет узнать соответствующий ему закрытый ключ.

Предположим, что двум людям, Бобу и Алисе, необходимо обмениваться секретными сообщениями. Если Боб зашифровал данные на своем закрытом

ключе, Алисе потребуется открытый ключ Боба, чтобы расшифровать их. При этом Алиса может не только расшифровать сообщение Боба, но и ответить Бобу зашифрованным сообщением. Для этого ей нужно зашифровать свой ответ на открытом ключе Боба, тогда Боб сможет расшифровать этот ответ с помощью своего закрытого ключа. При использовании асимметричного алгоритма, невозможно зашифровывать и расшифровывать сообщение одним и тем же ключом, эти ключи, хотя и связаны математически, они не совпадают (в отличие от симметричных алгоритмов). Боб может зашифровать данные на своем закрытом ключе, тогда Алиса сможет расшифровать их на открытом ключе Боба. Расшифровывая сообщение на открытом ключе Боба, Алиса может быть уверена, что сообщение действительно исходит от Боба, ведь сообщение может быть расшифровано на открытом ключе Боба только в том случае, если оно было зашифровано на соответствующем закрытом ключе Боба. Это обеспечивает возможность аутентификации, т.к. Боб является (гипотетически) единственным, кто имеет этот закрытый ключ. Если Алиса хочет быть уверена, что единственным, кто сможет прочитать её ответ, будет Боб, она должен зашифровать свое сообщение Бобу на его открытом ключе. Только тогда Боб сможет расшифровать это сообщение, поскольку только у него есть необходимый для этого закрытый ключ.

Если отправителю (Бобу) в большей степени важна конфиденциальность передаваемой информации, ему следует зашифровать свое сообщение на открытом ключе получателя. Это называют безопасным форматом сообщения (*secure message format*), поскольку только человек, имеющий соответствующий закрытый ключ (например, Алиса), сможет расшифровать это сообщение.

Если же отправителю (Бобу) в большей степени важна аутентификация, ему следует зашифровывать передаваемые данные на своем закрытом ключе. Это позволит получателю (Алисе) быть уверенной в том, что зашифровал данные именно тот человек, который имеет соответствующий закрытый ключ. Если отправитель шифрует данные на открытом ключе получателя, это не обеспечивает возможность аутентификации, т.к. открытый ключ доступен всем.

Оба ключа, как закрытый, так и открытый, могут использоваться как для зашифрования, таки и для расшифровывания данных. Особенно следует обратить внимание на то, что открытый и закрытый ключи могут использоваться как для зашифрования, так и для расшифровывания. При этом следует понимать, что если данные зашифрованы на закрытом ключе, они не могут быть

расшифрованы на нем же. Зашифрованные на закрытом ключе данные могут быть расшифрованы на соответствующем ему открытом ключе, и наоборот.

Асимметричные алгоритмы, как правило, значительно медленнее симметричных, т.к. они используют гораздо более сложную математику для выполнения своих функций, что требует больше вычислительных ресурсов. Однако асимметричные алгоритмы могут обеспечить аутентификацию и неотказуемость в зависимости от используемого алгоритма. Кроме того, асимметричные системы позволяют использовать более простой и управляемый процесс распространения ключей, по сравнению с симметричными системами и не имеют проблем с масштабируемостью, которые есть у симметричных систем. Причина этих различий в том, что при использовании асимметричных систем вы можете отправлять свой открытый ключ всем людям, с которыми вы хотите взаимодействовать, а не использовать для каждого из них отдельный секретный ключ.

ЛАБОРАТОРНАЯ РАБОТА №1

(Симметричная криптография. Простые шифры)

Цель работы:

- ознакомиться с основами симметричного шифрования;
- ознакомиться с простыми симметричными криптографическими шифрами на основе методов подстановок, перестановок и гаммирования;
- освоить основные этапы проектирования и реализации симметричных шифров;

1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Все симметричные шифры можно разделить на две большие группы. Первая – *поточные шифры*, где за один цикл алгоритма обрабатывается один элемент данных (бит или байт), а вторая – *блочные шифры*, в которых за один шаг обрабатывается группа элементов данных (*N* бит).

1.1. Поточные шифры

Поточные шифры, в отличие от блочных осуществляют поэлементное шифрование данных без задержки в криптосистеме, их важнейшим достоинством является высокая скорость преобразования, соизмеримая со скоростью поступления входной информации. Таким образом, обеспечивается шифрование практически в реальном времени вне зависимости от объёма и разрядности потока данных.

На рис. 4 изображена упрощённая модель поточного шифра, которая в принципе основана на схеме, представленной на рис. 2. Тем не менее, случайный поток битов теперь генерируется по короткому секретному ключу с помощью открытого алгоритма, называемого генератором ключевого потока, в котором биты шифротекста получаются по правилу: $C_i = m_i \oplus k_i$, где m_0, m_1, … – биты открытого текста, а k_0, k_1, … – биты ключевого потока.

Поскольку процесс шифрования – это сложение по модулю 2, расшифровывание является, по существу, той же самой операцией: $m_i = C_i \oplus k_i$.

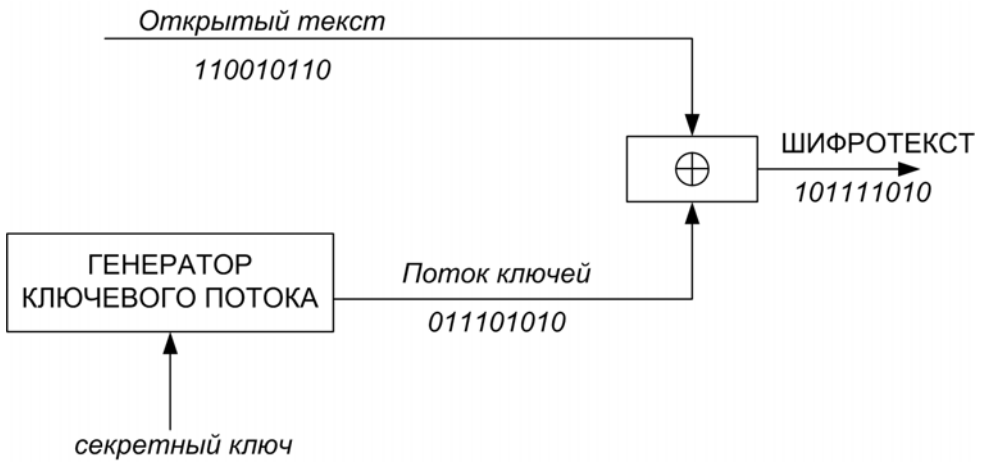

Рис. 4. Упрощенная модель поточного шифра

Как уже было отмечено, поточные криптосистемы позволяют очень быстро шифровать большой объём данных. Поэтому они отлично подходят для передачи аудио- и видео-сигналов в реальном времени. Кроме того, в этом процессе не происходит накопления ошибки. Если отдельный бит шифротекста исказился в процессе передачи вследствие слабого радиосигнала или из-за вмешательства противника, то в расшифрованном открытом тексте только один бит окажется неверным. Однако повторное использование того же ключа даёт тот же ключевой поток, что влечёт за собой зависимость между соответствующими сообщениями. Предположим, например, что сообщения m_1 и m_2 были зашифрованы одним ключом k. Тогда противник, перехватив шифровки, легко найдёт сумму по модулю 2 открытых текстов:

$$C_1 \oplus C_2 = (m_1 \oplus k) \oplus (m_2 \oplus k) = m_1 \oplus m_2 .$$

Следовательно, необходимо менять ключи либо с каждым новым сообщением, либо с очередным сеансом связи, в результате возникает проблема управления ключами и их распределения, которая решается с помощью криптосистем с открытым ключом.

Чтобы придать необходимую стойкость шифру, генератор ключевого потока производит строку битов с определенными свойствами. Как минимум, ключевой поток должен:

- иметь большой период. Поскольку ключевой поток получается в результате детерминированного процесса из основного ключа, найдётся такое число n, что $k_i = k_i + n$ для всех значений i. Число n является периодом последовательности, и для обеспечения стойкости шифра должно быть достаточно большим.
- Иметь псевдослучайные свойства. Генератор должен производить последовательность, которая кажется случайной, т.е. должна выдерживать определённое число статистических тестов на случайность.
- Обладать линейной сложностью.

1.2. Блочные шифры

Основным отличием блочного шифра от поточного является обработка в текущий момент времени целого блока байтов открытого текста. На рис. 5. изображена схема блочного шифра, где процесс шифрования и дешифрования описываются выражениями:

$$C = E_k(m) \text{ и } m = D_k(C),$$

где m – блок открытого текста, k – секретный ключ, E – функция шифрования, D – функция дешифровывания, C – блок шифротекста.

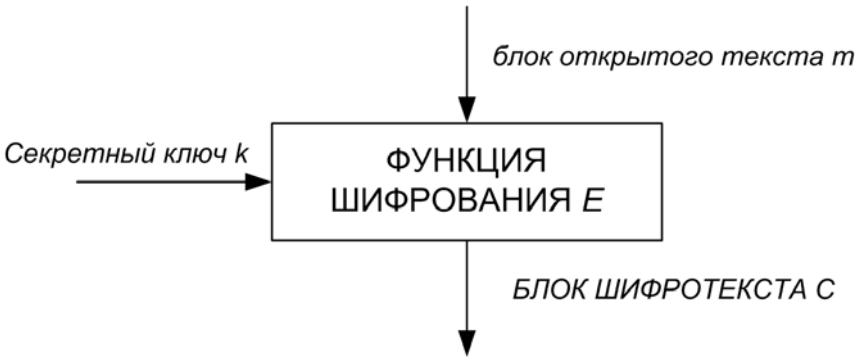

Рис. 5. Схема работы блочного шифра

Размер блока для шифрования обычно выбирается достаточно большим, размерностью 2^N. В современных блочных криптосистемах он достигает 256 и более бит.

Общий принцип шифрования заключается в следующем. Открытое сообщение делится на блоки 2^N битов. Затем эти блоки передаются на обработку математическим функциям, по одному блоку за раз. Если представить себе, что необходимо зашифровать сообщение размерностью 1400 бит с помощью блоч-

ного шифра, размерность блока которого равна 140 битам, то сообщение делится на 10 отдельных блоков по 140 бит. Каждый блок последовательно передается на вход математической функции. Этот процесс продолжается до тех пор, пока каждый блок не будет преобразован в шифротекст. После этого зашифрованное сообщение может быть передано адресату любым удобным способом. Адресат сообщения должен использовать тот же алгоритм и секретный ключ, чтобы осуществить дешифрацию. Зашифрованное сообщение делится на 10 блоков шифротекста и последовательно передаются в алгоритм в обратной последовательности до тех пор, пока не будет получен исходный открытый текст.

Следует отметить, что немоловажным фактором, определяющим эффективность блочного симметричного шифра, является способ связи между отдельными блоками шифротекста. К настоящему времени стали общепринятыми следующие режимы работы блочных алгоритмов:

- ***ECB***. Этот режим прост в обращении, но слабо защищён от возможных атак с удалениями и вставками. Ошибка, допущенная в одном из битов шифротекста, влияет на целый блок в расшифрованном тексте.
- ***CBC*** – наилучший способ эксплуатации блочного шифра, поскольку предназначен для предотвращения потерь в результате атаки с использованием удалений и вставок. В этом режиме ошибочный бит шифротекста при расшифровывании не только предотвращает в ошибочный блок, в котором содержится, но и портит один бит в следующем блоке открытого текста, что можно легко определить и интерпретировать как сигнал о предпринятой атаке.
- ***OFB***. При таком режиме блочный шифр превращается в поточный. Режим обладает тем свойством, что ошибка в один бит, просочившаяся в шифротекст, даёт только один ошибочный бит в расшифрованном тексте.
- ***CFB***. Как и в предыдущем случае, здесь блочный шифр трансформируется в поточный. Отдельная ошибка в криптограмме при этом влияет как на блок, в котором она была допущена, так и на следующий блок, как при режиме ***CBC***.

1.3. Криптостойкость симметричных алгоритмов

Обеспечение высокой криптостойкость шифра (устойчивости к взлому) является сложной научно-технической задачей, охватывающей множество необходимых и достаточных условий. Одними из необходимых условий являются свойства эффективного перемешивания (*confusion*) и рассеивания (*diffusion*) элементов шифротекста. В идеальном случае появление тех или иных элементов шифротекста должно быть равновероятным вне зависимости от открытого текста на входе шифра и секретного ключа. Например, в случае если под эле-

ментом понимается бит, то вероятность появления 1 или 0 в шифротексте должна быть $P = 0.5$. На практике, однако, добиться такого эффекта крайне сложно.

Перемешивание обычно выполняется с помощью операции ***подстановки***, тогда как рассеивание – с помощью ***перестановки***. Чтобы шифр был действительно стойким, он должен использовать оба эти метода, чтобы сделать процесс обратного (реверсивного) инжиниринга практически невозможным. На уровень перемешивания и рассеивания указывают случайность значения ключа и сложность применяемых математических функций.

В алгоритмах рассеивание может происходить как на уровне отдельных битов в блоках, так и на уровне самих блоков. Перемешивание выполняется с помощью сложных функций подстановки, чтобы злоумышленник не мог понять, каким образом заменялись исходные значения и получить оригинальный открытый текст.

Перемешивание выполняется для создания взаимосвязи между ключом и получаемым в результате шифротекстом. Эта взаимосвязь должна быть максимально сложной, чтобы невозможно было вскрыть ключ на основе анализа шифротекста. Каждое значение в шифротексте должно зависеть от нескольких частей ключа, но для наблюдателя эта связь между значениями ключа и значениями шифротекста должна выглядеть полностью случайной.

Рассеивание, с другой стороны, означает, что один бит открытого текста оказывает влияние на несколько бит шифротекста (или даже на все биты). Замена значения в открытом тексте должна приводить к замене нескольких значений в шифротексте, а не одного.

1.4. Подстановочные шифры

Подстановочным шифром называется шифр, который каждый символ открытого текста в шифротексте заменяет другим символом. Получатель инвертирует подстановку шифротекста, восстанавливая открытый текст. В классической криптографии существует четыре типа подстановочных шифров:

- *простой подстановочный шифр***,** или *моноалфавитный шифр***,** – это шифр, который каждый символ открытого текста заменяет соответствующим символом шифротекста. Простыми подстановочными шифрами являются криптограммы в газетах;
- *однозвучный подстановочный шифр*, – это шифр, который схож на простую подстановочную криптосистему за исключением того, что один символ открытого текста отображается на несколько символов шифротекста.

Например, «А» может соответствовать 5, 13, 25 или 56, «В» - 7, 19, 31 или 42 и так далее;

- *полиграммный подстановочный шифр* - это шифр, который блоки символов шифрует по группам. Например, "ABA" может соответствовать «RTQ», «ABB» может соответствовать «SLL» и так далее;
- *полиалфавитный подстановочный шифр*, – шифр, состоящий из нескольких простых подстановочных шифров. Например, могут быть использованы пять различных простых подстановочных фильтров, причём каждый символ открытого текста заменяется с использованием одного конкретного шифра.

Элементарным примером подстановочной криптосистемы является *шифр Цезаря*. Этот шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами. Применительно к современному русскому языку он состоял в следующем. Выписывался алфавит: А, Б, В, Г, Д, Е, и т.д. Затем под ним выписывался тот же алфавит, но с циклическим сдвигом на 3 буквы влево:

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

При зашифровании буква А заменялась буквой Г, Б заменялась на Д, В на Е и так далее. Так, например, слово «РИМ» превращалось в слово «УЛП». Получатель сообщения «УЛП» искал эти буквы в нижней строке и по буквам над ними восстанавливал исходное слово «РИМ». Ключом в шифре Цезаря является величина сдвига 2-й нижней строки алфавита. Преемник Юлия Цезаря - Цезарь Август использовал тот же шифр, но с ключом – сдвиг 1. Слово «РИМ» он в этом случае зашифровал бы в буквосочетание «СЙН».

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами модульной арифметики:

$$y = (x + k) \bmod n$$
$$x = (y - k) \bmod n\text{'}$$

где x – представляет собой позицию исходной буквы в алфавите (символ открытого текста), k – сдвиг, которым шифруется сообщение (ключ шифрования), n – мощность алфавита (количество букв в алфавите), а y – положение зашифрованной буквы в алфавите (символ шифрованного текста).

1.5. Перестановочные шифры

Шифр, преобразования из которого изменяют только порядок следования символов исходного текста, но не изменяют их самих, называется шифром перестановки. Рассмотрим преобразование перестановочного шифра, предназначенное для зашифрования сообщения длиной n символов с помощью таблицы

$$\begin{pmatrix} 1 & 2 & \dots & n \\ i_1 & i_2 & \dots & i_n \end{pmatrix},$$

где i_1 – номер позиции шифротекста, на которое попадает первая буква исходного сообщения при выбранном преобразовании, i_2 – номер позиции для второй буквы и т.д.

В верхней строке таблицы выписаны по порядку числа от 1 до n, а в нижней те же числа, но в произвольном порядке. Такая таблица представляет собой перестановочную матрицу степени n. Зная перестановку, задающую преобразование, можно осуществить как шифрование, так и расшифрование текста. Например, если для преобразования используется перестановка

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 4 & 2 & 5 \end{pmatrix},$$

в соответствии с ней зашифровывается слово РАДИО, то получится ДРИАО. Может легко убедиться в том, что существует $n!$ вариантов заполнения нижней строки таблицы. Таким образом, число различных преобразований шифра перестановки, предназначенного для зашифрования сообщений длины n, меньше либо равно $n!$ (заметим, что в это число входит и вариант преобразования, оставляющий все символы на своих местах!). Важно отметить, что с увеличением числа n значение $n!$ растет очень быстро. Приведем таблицу значений для первых 10 натуральных чисел:

Таблица 1. Количество возможных перестановок для первых 10 натуральных чисел

n	1	2	3	4	5	6	7	8	9	10
$n!$	1	2	6	24	120	720	5040	40320	362880	3628800

При больших значениях размерности шифруемого блока, т.е. n приближённое значение всех возможных перестановок $n!$ Можно вычислить по формуле Стирлинга:

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \text{ где } e = 2,718281828...$$

Хорошим вариантом перестановочного шифра являлся бы шифр, в котором при зашифровании сообщения длины n использовалось бы ключевое пространство (возможное количество перестановок) размерностью $n!$. Однако на практике такой шифр не удобен, так как при больших значения n необходимы перестановочные таблиц очень больших размерностей, а для их хранения в процессе шифрования нужны большие объёмы памяти компьютера. По этой причине криптографы-математики всё время ищут различные способы оптимизации перестановочных операций с позиции сокращения требуемых вычислительных ресурсов и одновременного сохранения высокой криптостойкости.

1.5.1. Маршрутная перестановка

Широкое распространение получили шифры перестановки, использующие некоторую геометрическую фигуру. Преобразования из этого шифра состоят в том, что в фигуру исходный текст вписывается по определённому направлению («маршруту»), а затем по ходу совершенно другого направления осуществляется извлечение из неё. Такой шифр называют ***маршрутной перестановкой***. Например, можно вписывать исходное сообщение в прямоугольную таблицу, выбрав такой маршрут: по горизонтали, начиная с левого верхнего угла поочередно слева направо и справа налево. Выписывать же сообщение можно по другому маршруту: по вертикали, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх.

Пример. Зашифруем, указанным способом фразу:

П Р И М Е Р М А Р Ш Р У Т Н О Й П Е Р Е С Т А Н О В К И

Для этого используем прямоугольник размером 4×7

П	Р	И	М	Е	Р	М
Н	Т	У	Р	Ш	Р	А
О	Й	П	Е	Р	Е	С
И	К	В	О	Н	А	Т

Согласно маршруту выписывания, зашифрованная фраза будет выглядеть так:

М А С Т А Е Р Р Е Ш Р Н О Е Р М И У П В К Й Т Р П Н О И

Маршруты могут быть значительно более изощрёнными, однако запутанность маршрутов усложняет использование таких шифров.

1.5.2. Поворотная перестановка

Суть данного способа шифрования заключается в следующем. Изготавливается трафарет из прямоугольного листа клетчатой бумаги размера $2m \times 2k$ клеток. В трафарете вырезано $m \times k$ клеток так, что при наложении его на лист

чистой бумаги того же размера четырьмя возможными способами его вырезы полностью покрывают всю площадь листа (см. рис. 6). Буквы сообщения последовательно вписываются в вырезы трафарета (по строкам, в каждой строке слева направо) при каждом из четырех его возможных положений в заранее установленном порядке.

Пример. Пусть в качестве ключа используется решетка 10×6, приведенная на рис. 6. Зашифруем с ее помощью текст:

Ш И Ф Р Р Е Ш Е Т К А Я В Л Я Е Т С Я Ч А С Т Н Ы М Л У Ч А Е М Ш И Ф Р А М А Р Ш Р У Т Н О Й П Е Р Е С Т А Н О В К И

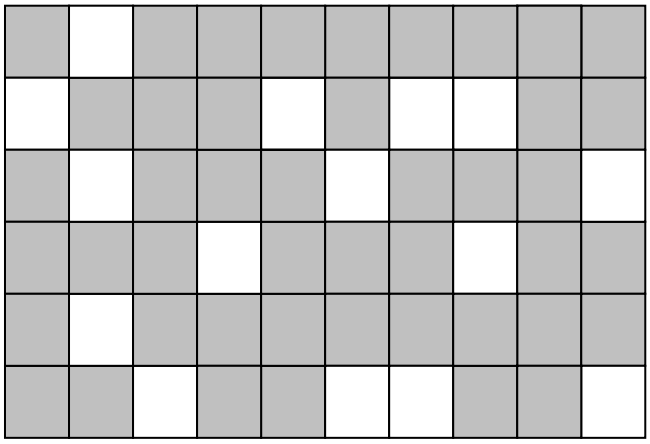

Рис. 6. Пример трафаретной решётки 10×6

Наложив решетку на лист бумаги, вписываем первые 15 (по числу вырезов) букв сообщения ШИФРРЕШЕТКАЯВЛЯ….. Сняв решетку, мы увидим текст, представленный на рис. 7. Поворачиваем решетку на 180 градусов. В окошках появятся новые, еще не заполненные клетки. Вписываем в них следующие 15 букв. Получится запись, приведенная на рис. 8.

	Ш								
И				Ф		Р	Р		
	Е				Ш				Е
			Т				К		
	А								
		Я			В	Л			Я

Рис. 7. Первое заполнение решётки

Е	Ш		Т	С			Я		
И				Ф		Р	Р	Ч	
	Е	А			Ш	С			Е
			Т				К		
	А								
		Я			В	Л			Я

Рис. 8. Второе заполнение решётки

Затем поворачиваем решетку на другую сторону и зашифровываем остаток текста аналогичным образом (рис. 9, 10).

Е	Ш	А	Т	С	Е	М	Я		Ш
И	И			Ф		Р	Р	Ч	
	Е	А	Ф		Ш	С	Р		Е
Т	А		Т	Н	М		К	Ы	А
Р	А	М	С	Ш	Л	Р	У		У
	Т	Я			В	Л		Ч	Я

Рис. 9. Третье заполнение решётки

Е	Ш	А	Т	С	Е	М	Я	Н	Ш
И	И	О	Й	Ф	П	Р	Р	Ч	Е
Р	Е	А	Ф	Е	Ш	С	Р	С	Е
Т	А	Т	Т	Н	М		К	Ы	А
Р	А	М	С	Ш	Л	Р	У	Н	У
О	Т	Я	В	К	В	Л	И	Ч	Я

Рис. 10. Четвёртое заполнение решётки

Получатель сообщения, имеющий точно такую же решетку, без труда прочтет исходный текст, наложив решетку на шифротекст по порядку четырьмя способами. Очевидно, что в данном способе шифрования, решётка представля-

ет собой ключ. Можно доказать, что общее количество ключей шифрования будет $K = 4^{mk}$. Следует отметить, что данный шифр предназначен для сообщений длины $n = 4mk$, а число всех перестановок в тексте такой длины составит $(4mk)!$, что во много раз больше числа K. Например, при размере трафарета 8×8 число возможных решеток превысит 4 миллиарда.

1.5.3. Вертикальная перестановка

Шифром вертикальной перестановки называются популярную разновидность шифра маршрутной перестановки. В нем используется прямоугольник, в котором сообщение вписывается обычным способом (по строкам слева направо). Выписываются буквы по вертикали, а столбцы при этом берутся в порядке, определяемом ключом. Пусть, например, этот ключ таков: (5, 1, 4, 7, 2, 6, 3), и с его помощью надо зашифровать сообщение:

ВОТПРИМЕРШИФРАВЕРТИКАЛЬНОЙПЕРЕСТАНОВКИ

Впишем сообщение в прямоугольник, столбцы которого пронумерованы в соответствии с ключом:

5	1	4	7	2	6	3
В	О	Т	П	Р	И	М
Е	Р	Ш	И	Ф	Р	А
В	Е	Р	Т	И	К	А
Л	Ь	Н	О	Й	П	Е
Р	Е	С	Т	А	Н	О
В	К	И	-	-	-	-

Теперь, выбирая столбцы в порядке, заданном ключом и выписывая последовательно буквы каждого из них сверху вниз, получаем такую криптограмму (зашифрованный текст):

ОРЕЬЕРФИЙА-МААЕО-ТШРНСИВЕВЛРВИРКПН-ПИТОТ-

В данном способе следует отметить, что число ключей шифрования в данном алгоритме не более $m!$, где m – число столбцов таблицы. Как правило, m гораздо меньше, чем длина текста n (сообщение укладывается в несколько строк по m букв), а значит, и $m!$ много меньше $n!$.

1.5.4. Гаммирование

Гаммирование также является широко применяемым криптографическим преобразованием. Принцип шифрования гаммированием заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и «наложении» полученной гаммы на открытые данные обратимым образом. Обычно это суммирование по модулю 2, т.е. обычного «исключающего ИЛИ» (см. рис. 11).

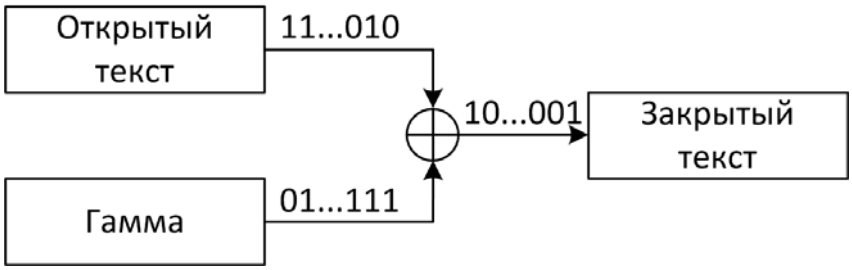

Рис. 11. Гаммирование с использование операции «исключающее ИЛИ»

Процесс дешифрования данных сводится к повторной генерации гаммы шифра при известном ключе и обращении процесса наложения такой гаммы на зашифрованные данные.

2. ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ

Согласно Вашему персональному варианту (см. табл. 2) или индивидуальному заданию преподавателя разработайте и составьте в виде блок-схемы алгоритмы шифрования и дешифрования текста. Убедитесь в правильности составления алгоритмов и затем на языке программирования составьте программу, которая реализует данные алгоритмы.

На ряде контрольных примеров (не менее 10) открытого текста, состоящего из различного количества символов, проверьте правильность работы алгоритмов шифрования и дешифрования.

Самостоятельно придумайте оригинальный способ модификации шифра с целью повышения его криптостойкости. Внесите изменения в исходный алгоритм и программу. Проверьте работоспособность алгоритма на тестовых примерах.

Докажите, что предложенный Вами способ модификации действительно повышает криптостокость.

Разработанная Вами программа должна содержать графический интерфейс пользователя.

Таблица 2. Варианты заданий в лабораторной работе (начало)

№ вар.	Задание
1	Шифр на основе «магических» квадратов размерностью $N \times N$
2	Квадрат Полибия
3	Шифр Виженера
4	Шифр маршрутной перестановки
5	Шифр поворотной решетки
6	Шифр Плейфера
7	Шифр Трисемуса

Таблица 2. Варианты заданий в лабораторной работе (продолжение)

8	Шифр Хилла
9	Шифр Атбаш
10	Шифр вертикальной перестановки.
11	Шифр Мирабо
12	Шифр графа Гронфельда

3. СОДЕРЖАНИЕ ОТЧЁТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

После выполнения лабораторной работы и проверки адекватности полученных результатов, необходимо подготовить отчёт, включающий в себя:

- титульный лист;
- задание к лабораторной работе (согласно вашему варианту или индивидуальному заданию, выданному преподавателем);
- подробное описание алгоритма шифрования (структура, режимы работы, криптостойкость и т.п.);
- блок-схемы алгоритмов шифрования и дешифрования, оформленные в соответствии с ГОСТ 19.701-90;
- листинг составленной программы, реализующей алгоритмы (шрифт `Courier New`, 6 кегель);
- контрольные примеры работы программы, с различными исходными текстами, ключами и/или другими параметрами (не менее 5);
- подробное описание предложенной Вами модификации алгоритма, включая доказательства, изложенные в любой форме, повышения криптостойкости алгоритмов;
- модифицированные алгоритмы шифрования и дешифрования, разработанные самостоятельно, оформленные в соответствии с ГОСТ 19.701-90 **с выделением** отличающихся от оригинала блоков;
- листинг составленной программы, реализующей модифицированные алгоритмы (шрифт `Courier New`, 6 кегель);
- контрольные примеры работы программы, с различными исходными текстами, ключами и/или другими параметрами (не менее 5);
- заключение, содержащее выводы о полученных **лично Вами** результатах, в ходе выполнения лабораторной работы (приобретённые знания, навыки, умения и т.п.).

Отчёт должен быть подготовлен в текстовом редакторе, согласно действующему стандарту организации на оформление студенческих работ, вклю-