1. **Основные понятия и определения криптографической защиты информации**

**Криптография** — это наука о методах защиты информации путём шифрования исходных данных.

Для обеспечения безопасности данных необходимо поддерживать три основные функции:

* защиту конфиденциальности передаваемых или хранимых в памяти данных;
* подтверждение целостности и подлинности данных;
* аутентификацию абонентов при входе в систему и при установлении соединения;

Для реализации указанных функций используются криптографические технологии шифрования, цифровой подписи и аутентификации.

**Конфиденциальность** обеспечивается с помощью алгоритмов и методов симметричного и асимметричного шифрования, а также путем взаимной аутентификации абонентов

Целостность и подлинность передаваемых данных обычно достигается с помощью различных вариантов технологии электронной подписи, основанных на односторонних функциях и асимметричных методах шифрования.

**Аутентификация** разрешает устанавливать соединения только между легальными пользователями и предотвращает доступ к средствам сети нежелательных лиц. Абонентам, доказавшим свою легальность (аутентичность), предоставляются разрешенные виды сетевого обслуживания.

Обеспечение конфиденциальности, целостности и подлинности передаваемых и сохраняемых данных осуществляется прежде всего правильным использованием криптографических способов и средств защиты информации.

Основой большинства криптографических средств защиты информации является **шифрование данных.**

Под **шифром** понимают совокупность процедур и правил криптографических преобразований, используемых для зашифровывания и расшифровывания информации по ключу шифрования.

Под **зашифровыванием информации**понимается процесс преобразования открытой информации (исходный текст) в зашифрованный текст (шифртекст).

Процесс восстановления исходного текста по криптограмме с использованием ключа шифрования называют **расшифровыванием** (дешифрованием).

Обобщенная схема криптосистемы шифрования показана на рис. 5.1. Исходный текст передаваемого сообщения (или хранимой информации) М зашифровывается с помощью криптографического преобразования Ек с получением в результате шифртекста С:

C=Eki(Nf), где — параметр функции Е, называемый ключом шифрования.



Шифртекст С, называемый также *криптограммой*, содержит исходную информацию *М* в полном объеме, однако последовательность знаков в нем внешне представляется случайной и не позволяет восстановить исходную информацию без знания ключа шифрования *кх*.

**Ключ шифрования** является тем элементом, с помощью которого можно варьировать результат криптографического преобразования. Данный элемент может принадлежать конкретному пользователю или группе пользователей и являться для них уникальным. Зашифрованная с использованием конкретного ключа информация может быть расшифрована только его владельцем (или владельцами).

Обратное преобразование информации выглядит следующим образом:

*M’ = Dki(C)*

Функция *D* является обратной к функции *Е* и производит расшифровывание шифртекста. Она также имеет дополнительный параметр в виде ключа *к2*. Ключ расшифровывания *к2* должен однозначно соответствовать ключу *к1* в этом случае полученное в результате расшифровывания сообщение *М’* будет эквивалентно *М*. При отсутствии верного ключа *к2* получить исходное сообщение *М’* = *Мс* помощью функции *D* невозможно.

Преобразование шифрования может быть симметричным или асимметричным относительно преобразования расшифровывания. Соответственно различают два класса криптосистем:

* симметричные криптосистемы (с единым ключом);
* асимметричные криптосистемы (с двумя ключами).

1. **Шифрование**

**Шифрование** — это преобразование открытой информации в зашифрованную и наоборот, а две составные части этого процесса называют соответственно зашифрованием и расшифрованием.

Для того, чтобы зашифрованное сообщение было возможным расшифровать, необходимо единовременное исполнение двух условий:

1. Функция расшифрования и функция зашифрования должны соответствовать друг другу
2. Ключ расшифрования аналогично обязан соответствовать ключу зашифрования.

**Ключ** — это уникальный код, при помощи которого можно изменять результаты работы алгоритма шифрования: одна и та же исходная информация при использовании различных ключей будет зашифрован по-разному.

**Криптостойкость** — основная характеристика алгоритмов шифрования и указывает прежде всего на степень сложности получения исходного текста из зашифрованного без соответствующего ключа.

**Симметричное шифровани**е

История алгоритмов симметричного шифрования (шифрования с закрытым ключом) начинается с римского императора Гая Юлия Цезаря — в I веке до н.э. римлянин уже пользовался этим алгоритмом, впоследствии ставшего известным как "криптографическая система Цезаря".

Сегодня наиболее распространенным алгоритмом симметричного шифрования является стандарт DES, ГОСТ 28147–89, (AES).

**Асимметричное шифровани**е

В настоящее время среди алгоритмов ассиметричного шифрования широкое распространение получил алгоритм RSA.

1. **Аутентификация**

**Аутентификация** — проверка принадлежности субъекту доступа по предъявленному им идентификатору (пароль, ключ и т.д.); подтверждение подлинности.

Методы аутентификации:

* парольные (PIN коде и т.д.) — уникальная последовательность символов, которую пользователь должен знать.
* "ключ" — в случае электронных систем это электронный ключ, который хранится на носителе (смарт-карты, электронные таблетки iButton, USB-токены и т. д.)
* биометрические (отпечаток пальца, рисунок радужной оболочки глаза, форма лица, параметры голоса и т. д.)
* криптографические

Протоколы аутентификации:

**PAP — аутентификация по имени и паролю пользователя**. Протокол PAP ненадежен при использовании в сетях, т.к. пароли можно перехватить.

Алгоритм PAP:

* клиент посылает имя и пароль серверу
* сервер сверяет присланный пароль с паролем в своем хранилище

Преимущества:

1. простота

Недостатки:

1. подбор паролей
2. просмотр паролей в системе
3. перехват паролей при передачи
4. пароль можно «подсмотреть» при вводе
5. человеческий фактор – человек не может запомнить сложные пароли (записывает), диктует открытым способом (по телефону) и т.д.
6. каждый раз нужно набирать на клавиатуре
7. нужна предварительная регистрация пользователя в системе

**Аутентификация по одноразовым паролям (One-time password)**

Различные подходы к созданию одноразовых паролей:

* использующие математические алгоритмы для создания нового пароля на основе предыдущих (пароли фактически составляют цепочку, и должны быть использованы в определённом порядке).
* основанные на временной синхронизации между сервером и клиентом, обеспечивающей пароль (пароли действительны в течение короткого периода времени)
* использующие математический алгоритм, где новый пароль основан на запросе (например. случайное число, выбираемое сервером или части входящего сообщения) и/или счётчике.

Одноразовые пароли клиент может получать:

1. на бумаге
2. в токене
3. пересылкой (по СМС)

**Многофакторная аутентификация**

Иногда используются сразу несколько методов аутентификации.

Например: электронный ключ и логин.

При использовании SIM-карт в мобильных телефонах. Субъект вставляет свою SIM-карту в телефон и при включении вводит свой PIN-код (пароль).

В случае банковской карты. Субъект вставляет свою банковскую карту в банкомат и вводит свой PIN-код (пароль).

1. **Хэширование**

**Хеширование** — преобразование по детерминированному алгоритму входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования также называются **хеш-функциями** или **функциями свёртки**, а их результаты называют **хешем**, **хеш-кодом** или **сводкой сообщения**.

Хеширование применяется для построения достаточно уникальных идентификаторов для наборов данных, для хранения паролей в системах защиты (в этом случае доступ к области памяти, где находятся пароли, не позволяет восстановить сам пароль), при выработке электронной подписи (на практике часто подписывается не само сообщение, а его хеш-образ).

Криптостойкой хеш-функция может быть только в том случае, если выполняются главные требования:

* стойкость к восстановлению хешируемых данных
* стойкость к коллизиям, то есть образованию из двух разных массивов данных двух одинаковых значений хеша.

1. **Шифры замены и перестановки**

**Шифрование перестановкой** заключается в том, что символы открытого текста переставляются по определенному правилу в пределах некоторого блока этого текста. Рассмотрим перестановку, предназначенную для шифрования сообщения длиной **n** символов. Его можно представить с помощью таблицы

1

Зная перестановку, задающую преобразование, можно осуществить как шифрование, так и расшифрование текста. В этом случае сама таблица перестановки служит ключом шифрования.

Число различных преобразований шифра перестановки, предназначенного для шифрования сообщений длины n, меньше либо равно n!.

С увеличением числа n значение n! растет очень быстро. Для использования на практике такой шифр не удобен, так как при больших значениях n приходится работать с длинными таблицами.

Примеры: Шифр перестановки "скитала" Шифрующие таблицы Шифрование с помощью магических квадратов

**Шифрами замены** называются такие шифры, преобразования из которых приводят к замене каждого символа открытого текста на другие символы — шифрообозначения, причем порядок следования шифрообозначений совпадает с порядком следования соответствующих им символов открытого сообщения.

В своей простейшей форме шифр замены может быть задан таблицей подстановки, устанавливающей соответствие между буквами двух алфавитов An и Bn:

1

где 1- i-тая буква алфавита открытого текста, 1- шифрообозначение ai (соответствующая ai буква алфавита шифртекста).

В качестве примера преобразования, которое может содержаться в шифре замены, приведем такое правило. Каждая буква исходного сообщения заменяется на ее порядковый номер в алфавите. В этом случае исходный буквенный текст преобразуется в числовой.

Алфавиты An и Bn не обязательно должны быть различными. В практической криптографии очень часто применяются шифры, в которых алфавиты An и Bn совпадают.

В **шифре простой замены** каждый символ исходного текста заменяется символами того же алфавита одинаково на всем протяжении текста.

В **шифрах сложной замены** для шифрования каждого символа открытого текста применяют свой шифр простой замены. Для реализации шифров сложной замены последовательно и циклически меняют используемые таблицы подстановки.

Примеры: Система шифрования Цезаря Система шифрования Вижинера

1. **Блочные шифры**

**Блочными** называются шифры, в которых логической единицей шифрования является некоторый блок открытого текста, после преобразований которого получается блок шифрованного текста такой же длины. Обычно используются блоки размером 64 бита.

Большинство сетевых приложений, в которых применяется схема традиционного шифрования, использует блочные шифры.

Чтобы шифрование было обратимым (т.е. чтобы обеспечивалась возможность дешифрования), каждый из таких блоков должен преобразовываться в свой уникальный блок шифрованного текста. Такие преобразования называются обратимыми, или несингулярными.

1. **Режимы шифрования**

**Режим шифрования** — метод применения блочного шифра (алгоритма), позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных. При этом для шифрования одного блока могут использоваться данные другого блока.

Обычно режимы шифрования используются для изменения процесса шифрования так, чтобы результат шифрования каждого блока был уникальным вне зависимости от шифруемых данных и не позволял сделать какие-либо выводы об их структуре. Это обусловлено, прежде всего, тем, что блочные шифры шифруют данные блоками фиксированного размера, и поэтому существует потенциальная возможность утечки информации о повторяющихся частях данных, шифруемых на одном и том же ключе.

Основные режимы:

* Electronic Codebook (ECB)
* Cipher Block Chaining (CBC)
* Propagating Cipher Block Chaining (РСВС)
* Cipher Feedback (CFB)
* Output Feedback (OFB)
* Counter mode (CTR)
* Random Delta (RD)
* Galois/Counter Mode (GCM) и AEAD

1. **Система RSA**

Шаг первый. Подготовка ключей

Сгенерировать публичный и приватный ключ.

Выбираю два простых числа. Пусть это будет **p=3** и **q=7**.

Вычисляем модуль-произведение наших **p** и **q: n=p×q=3×7=21.**

Вычисляем функцию Эйлера: **φ=(p-1)×(q-1)=2×6=12.**

Выбираем число **e**, отвечающее следующим критериям:

1. оно должно быть простое
2. оно должно быть меньше **φ** — остаются варианты: 3, 5, 7, 11
3. оно должно быть взаимно простое с **φ**; остаются варианты 5, 7, 11. Выберем **e=5**. Это, так называемая, открытая экспонента.

Теперь пара чисел **{e, n}** — это мой открытый ключ. Я отправляю его вам, чтобы вы зашифровали своё сообщение. Но для меня это ещё не всё. Я должен получить закрытый ключ.

Мне нужно вычислить число **d**, обратное **е** по модулю **φ**. То есть остаток от деления по модулю **φ** произведения **d×e** должен быть равен 1. Запишем это в обозначениях, принятых во многих языках программирования: **(d×е)%φ=1**. Или **(d×5)%12=1**. **d** может быть равно 17. Можете проверить (**17×5-12×7=1**). Итак, **d=17**. Пара **{d, n}** — это секретный ключ, его я оставляю у себя. Его нельзя сообщать никому. Только обладатель секретного ключа может расшифровать то, что было зашифровано открытым ключом.

Теперь пришла ваша очередь шифровать ваше сообщение. Допустим, ваше сообщение — это число 19. Обозначим его **P=19**. Кроме него у вас уже есть мой открытый ключ: **{e, n} = {5, 21}.** Шифрование выполняется по следующему алгоритму:

Возводите ваше сообщение в степень **e** по модулю **n**. То есть, вычисляете 19 в степени 5 (2476099) и берёте остаток от деления на 21. Получается 10 — это ваши закодированные данные.

Строго говоря, вам вовсе незачем вычислять огромное число «19 в степени 5». При каждом умножении достаточно вычислять не полное произведение, а только остаток от деления на 21. Но это уже детали реализации вычислений, давайте не будем в них углубляться.

Полученные данные **E=10**, вы отправляете мне.

Здесь надо заметить, что сообщение **P=19** не должно быть больше **n=21**. иначе ничего не получится.

Я получил ваши данные (**E=10**), и у меня имеется закрытый ключ **{d, n} = {17, 21**}.

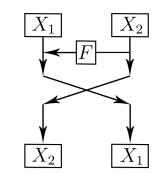
Обратите внимание на то, что открытый ключ не может расшифровать сообщение. А закрытый ключ я никому не говорил. В этом вся прелесть асимметричного шифрования.

Начинаем раскодировать:

Я делаю операцию, очень похожую на вашу, но вместо **e** использую **d**. Возвожу **E** в степень **d**: получаю 10 в степени 17 (позвольте, я не буду писать единичку с семнадцатью нулями). Вычисляю остаток от деления на 21 и получаю 19 — ваше сообщение.

Заметьте, никто, кроме меня (даже вы!) не может расшифровать ваше сообщение (**E=10**), так как ни у кого нет закрытого ключа.

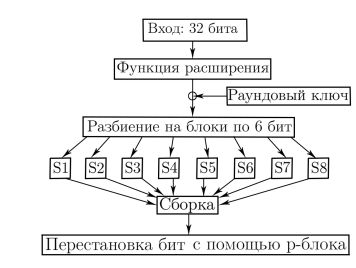
1. **Шифр DES**



Здесь применяются так называемые **ячейки Фейстеля** — это способы организовать блочный алгоритм шифрования таким образом, что криптографу нет нужды заботиться о восстановимости шифра.

Суть алгоритма состоит в следующем: текст разделяется на две половины. Затем к правой части применяется функция Фейстеля. Далее результат применения складывается побитово по модулю 2 с левой частью, после чего подставляется в правую часть следующей итерации, а правая часть подставляется в левую (без изменения).

После одной итерации в ячейке Фейстеля меняется только половина текста. Эти итерации (раунды) повторяются множество раз. Благодаря этому обеспечивается эффект надежного шифра.



На вход подается 32 бита, затем расширяется до 48 бит. Далее для каждой ячейки Фейстеля побитово прибавляется раундовый ключ (размером 48 бит). Потом разбиваем то, что получилось, на блоки по 6 бит, после чего каждый из восьми блоков подается на соответствующий s-блок. Там выполняется нелинейная замена, и потом применяется перестановка бит.

1. **Шифр ГОСТ 28147-89**

**ГОСТ 28147-89** является отечественным блочным шифром. То есть открытый текст разбивается на блоки (в данном случае 64 бита), и каждый блок преобразовывается отдельно. В основу алгоритма положена сеть Фейстеля.

1. Каждый блок разбивается на два «подблока» (левый и правый, соотвественно).
2. Исходное заполнение правого блока записывается в левый блок на выходе.
3. Над правым блоком производится криптографическое преобразование с применением ключевых данных.
4. Левый (исходный) и правый (преобразованный) блоки складываются по модулю 2 в сумматоре по модулю 2.
5. Так повторяется несколько раз.