МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ

О.А. ГУЛЯЄВА

ЗАХИСТ ІНФОРМАЦІЇ

(конспект лекцій)

Дніпропетровськ
2015

Содержание

Тема 1. Введение в информационную безопасность

70-годы XX века ознаменовались появлением ПК, и это кардинально изменило нашу жизнь. Упростился доступ к информации.

1. Глобальная сеть Интернет создавалась как открытая система, предназначенная для свободного обмена информации. Работая во Всемирной сети, следует помнить, что абсолютно все действия фиксируются и протоколируются специальными программными средствами. Таким образом, к обмену информацией следует подходить как к обычной переписке с использованием почтовых открыток.

Информация свободно циркулирует в обе стороны, но в общем случае она доступна всем участникам информационного процесса. Даже с точки зрения этических норм общения переписку лучше скрывать.

2. Сегодня Интернет является не только средством общения и универсальной справочной системой — в нём циркулируют договорные и финансовые обязательства, необходимость защиты которых как от просмотра, так и от фальсификации очевидна.

3. Начиная с 1999г. Интернет становится мощным средством обеспечения розничного торгового оборота, а это требует защиты данных кредитных карт и других электронных платёжных средств.

Таким образом, информация приобрела самостоятельную комерческую ценность и стала товаром. Её производят, транспортируют, продают, покупают, а значит — воруют и подделывают. **«Кто владеет информацией, тот владеет миром».**

Число компьютерных преступлений растёт, и у компьютерного преступника шансов быть пойманным гораздо меньше, чем у грабителя. Но с другой стороны, ошибок персонала еще больше. Около 60 процентов всех потерь, наносят ошибки людей. Как выразился один эксперт, "мы теряем из-за ошибок больше денег, чем могли бы украсть". В конечном счёте, всё это приводит к подрыву экономики.

Информация в современном мире – это ценный ресурс, а значит, её нужно защищать.

Информационной безопасностью называют меры по защите информации

- ***от несанкционированного доступа,***
- ***от разрушения,***

- ***от модификации***
- ***от задержек в доступе***.

Каждый раз, решая вопрос о защите информации, мы должны оценить важность информации, чтобы затраты на защиту информации не превышали ценность самой информации.

С другой стороны, противник поступает так же: если затраты на взлом защиты превышают выгоду от использования полученной информации, то он вряд ли будет этим заниматься.

Четыре уровня защиты информации

1. Предотвращение *– обеспечивается контроль над информацией, т.е. такие мероприятия, которые не допустят компьютерных преступлений. Доступ к информации только авторизованных пользователей. Авторизация – процесс предоставления определённому лицу прав на выполнение некоторых действий.*

2. Обнаружение *- обеспечивается раннее обнаружение преступлений и злоупотреблений, даже если механизмы защиты не сработали.*

3. ***Ограничение*** *- уменьшается размер потерь, если преступление все-таки произошло несмотря на меры по его предотвращению и обнаружению.*

4. Восстановление - обеспечивается эффективное восстановление информации.

Мероприятия по обеспечению информационной безопасности

1. Контролировать доступ к информации на компьютере. Не оставлять на рабочем столе важные документы. Архивировать с использованием пароля или криптографировать конфиденциальные документы.

***2.*Идентификация и аутентификация пользователей**

Идентификация позволяет субъекту назвать себя (сообщить свое имя).

У любого зарегистрированного в системе пользователя есть свой ID (идентификатор) —уникальный признак, который позволяет отличать его от других объектов. Идентификатор предоставляет субъекту определённый уровень полномочий.

Примеры идентификаторов:

- пароль, личный идентификационный номер —это *то, что знает только пользователь*;
- магнитная карта,— это *то, чем владеет пользователь;*

- голос, отпечаток пальца, радужная оболочка глаза — *это уникальные характеристики пользователя.*

Аутентификация — процедура проверки подлинности идентификации (проверка подлинности субъекта или информации).

Идентификацию и *аутентификацию* можно считать основой программно-технических средств безопасности, поскольку сервисы рассчитаны на обслуживание именованных субъектов.

Большинство систем безопасности функционируют следующим образом:

- в базе данных системы хранится пароль пользователя, или оцифрованные отпечаток пальца, радужная оболочка глаза, голос;
- человек, собирающийся получить доступ к системе, вводит информацию о себе:
    - своё имя и подтверждает его паролем;
    - отсканированный отпечаток пальца;
    - фразу, с помощью микрофона;
    - либо посмотрев в окуляр анализатора радужной оболочки глаза.
- поступившие данные сравниваются с образцом, хранящимся в базе данных;
- средства защиты применяются в том случае, когда подлинность субъекта не подтвердилась.

Парольная аутентификация

Главное достоинство ***парольной аутентификации*** — простота и привычность. Пароли давно встроены в операционные системы и иные сервисы. При правильном использовании пароли могут обеспечить приемлемый для многих организаций уровень безопасности. Тем не менее, по совокупности характеристик их следует признать самым слабым средством проверки подлинности.

Меры, позволяющие значительно повысить надежность парольной защиты:

- не делитесь своим паролем ни с кем, держите его в памяти;
- выбирайте пароль трудно угадываемым: используйте строчные и прописные буквы, цифры, длинные пароли — более безопасны;
- позвольте компьютеру самому сгенерировать ваш пароль;
- управляйте сроком действия паролей, их периодически меняйте;
- ограничьте число неудачных попыток входа в систему;
- обеспечьте неотображаемость пароля на экране компьютера при его вводе.

3. Процедуры авторизации

Разработайте процедуры авторизации, которые определяют, кто из пользователей должен иметь доступ к той или иной информации.

4. Предосторожности при работе

- отключайте не используемые терминалы, закрывайте комнаты, где находятся терминалы
- разворачивайте экраны компьютеров так, чтобы они не были видны со стороны двери, окон;
- программируйте терминал отключаться после определенного периода не использования.

Тема 2. Основы криптографии

Если данные передаются через открытые системы (а Internet относится к таким), то исключить доступ к ним посторонних лиц невозможно. Поэтому должны быть системы защиты.

Существует 3 возможности тайной передачи информации:

1. Создать абсолютно надёжный канал связи между абонентами.

Реализовать это между удалёнными абонентами для многократной передачи информации не реально.

2. Использовать общедоступный канал связи, но передавать по нему информацию в преобразованном виде, что бы восстановить её мог только адресат. Этим занимается *криптография*.

3. Использовать общедоступный канал связи, но скрыть сам факт передачи информации. Этим занимается *стеганография*.

Предмет криптографии

Существуют различные методы защиты информации, из которых основным является криптография.

Криптография — это наука о способах преобразования информации с целью ее защиты от незаконных пользователей. Т.е. это наука о шифровании.

Шифрование — это такое преобразование текста, в результате которого прочитать его сможет только тот, кто обладает специальным ключом.

Но существуют и противники, которые хотят сломать шифр. Поэтому обязательным этапом при создании шифра является изучение его стойкости к различным атакам. Этим занимается противоположная наука — криптоанализ.

Криптоанализ — это наука, изучающая стойкость шифров и методы их взлома (другими словами, это наука о раскрытии зашифрованных сообщений без доступа к ключу).

И криптография, и криптоанализ изучают одни и те же объекты, но с разных точек зрения . Поэтому они являются двумя частями одной и той же науки «*криптологии*».

Криптология — это наука, изучающая способы преобразования информации с целью ее защиты, а также стойкость шифров и методы их взлома.

Часто шифрование путают с кодированием.

Кодированием называется любое преобразование данных из одной формы представления в другую. К кодированию относится:

- преобразование информации, вводимой с клавиатуры (преобразование в 2-ую систему);
- шифрование;
- помехоустойчивое кодирование (преобразование текста, позволяющее восстанавливать его в случае сбоя при передаче или хранении);
- сжатие данных;
- сканирование текста или изображения, при котором информация преобразуется из визуального представления в цифровое.

Простейшие способы шифрования появились ещё в древности, но научный подход к разработке криптографических методов появился только в прошлом XX веке.

Основателем теории информации признан Клод Шеннон (амер. учёный). Его статья «Математическая теория связи» (1948 г.) послужила началом обширных исследований в теории передачи информации и придала криптографии статус науки. Современная криптография находится на стыке математики и информатики.

История криптографии

Термин *криптография* (тайнопись) ввел Джон Валлис (англ. математик, 17 век).

Еще в V-IV вв. до н. э. греки применяли специальное шифрующее устройство. По описанию Плутарха, оно состояло из двух палок одинаковой длины и толщины. Эти палки называли *скиталами*.

Допустим, два правителя хотят осуществить между собой переписку. Одна палка у одного правителя, другая — у другого. Когда правителям нужно было сообщить какую - нибудь важную тайну, они вырезали длинную и узкую, вроде ремня, полосу папируса, наматывали ее на свою скиталу, не оставляя на ней никакого промежутка, так чтобы вся поверхность палки была охвачена этой полосой.

Затем, писали на папирусе все, что нужно, а, написав, снимали полосу и без палки отправляли адресату. Так как буквы на ней разбросаны в беспорядке, то прочитать написанное можно только при помощи соответствующей скиталы, намотав на нее без пропусков полосу папируса.

Аристотелю принадлежит способ дешифрования этого шифра. Надо изготовить длинный конус и, начиная с основания, обертывать его лентой с шифрованным сообщением, постепенно сдвигая ее к вершине. В какой - то момент начнут просматриваться куски сообщения. Так можно определить диаметр скиталы.

В Древней Греции (II в. до н. э.) был известен шифр, называемый *квадрат Полибия*.

Это устройство представляло собой квадрат 5 х 5, столбцы и строки которого нумеровали цифрами от 1 до 5. В каждую клетку этого квадрата записывалась одна буква. В греческом алфавите 24 буквы, поэтому одна клетка оставалась пустой, в латинском – 26 букв, в одну клетку помещали две буквы ***i*** и j (см. рис. 1).

а)

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I,J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

б)

	A	B	C	D	E
A	A	B	C	D	E
B	F	G	H	I,J	K
C	L	M	N	O	P
D	Q	R	S	T	U
E	V	W	X	Y	Z

Рис.1. Квадрат Полибия с латинским алфавитом

Шифрование сводилось к замене буквы парой цифр, сообщение превращалось в последовательность пар цифр.

Пример использования квадрата Полибия с латинским алфавитом (рис.1.а).

шифротекст 13 34 22 24 44 34 15 42 22 34 43 45 32

В шифротексте 1-я цифра — № строки, а 2-я цифра — № столбца.

"Cogito, ergo sum" – лат, "Я мыслю, следовательно, существую").

Иначе (рис 1.б):

AC CD DD BD CD AE DB BB CD DE CB — шифротекст

Здесь квадрат Полибия реализован без ключа.

Идею квадрата Полибия проиллюстрируем таблицей с русскими буквами. Число букв в русском алфавте 33 (32 без ё), поэтому размер таблицы выбран иным — прямоугольник 8х4.

а)

	1	2	3	4	5	6	7	8
1	А	Б	В	Г	Д	Е	Ж	З
2	И	Й	К	Л	М	Н	О	П
3	Р	С	Т	У	Ф	Х	Ц	Ч
4	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я

б)

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

Зашифруем слово КРИПТОГРАФИЯ:

Таблица а) 23 31 21 28 33 27 14 31 11 35 21 48

Таблица б) ЕФ МШ ЕШ ЕА МФ ЕК ТР МШ ТШ МО ЕШ АА

Расшифровка в случае а) без ключа, в случае б) нужно знать 2 слова.

В 1 в. н. э. Ю. Цезарь во время войны с галлами, переписываясь со своими друзьями в Риме, заменял в сообщении первую букву латинского алфавита (A) на четвертую (D), вторую (B) – на пятую (E), наконец, последнюю – на третью:

Лат. алфавит – 26 букв.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Пример. Донесение Ю. Цезаря Сенату об одержанной им победе над Понтийским царем выглядело так:

Шифротекст YHQL YLGL YLFL

Расшифровка Veni, vidi, vici – лат. "Пришел, увидел, победил".

Император Август (1 в. н. э.) в своей переписке заменял первую букву на вторую, вторую – на третью и т. д., наконец, последнюю – на первую:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

B C D E F G H I J K L M N O P Q R S T U V W X Y Z A

Пример. Любимое изречение императора Августа выглядело так:

GFTUJOB MFOUF ("Festina lente" – лат. "Торопись медленно").

Квадрат Полибия, шифр Цезаря входят в класс шифров, называемых *подстановка* или *простая замена*, т.е. это такой шифр, в котором каждой букве алфавита ставится в соответствие буква, цифра, символ или группа символов.

Основные цели и задачи криптографии

Цель криптографических методов — защита информационной системы от разрушающих воздействий (*атак*) со стороны *противника*.

1. Классической модель системы секретной связи: есть два полностью доверяющих друг-другу участника, которым необходимо передавать между собой информацию, не предназначенную для третьих лиц. Такая информация называется конфиденциальной или секретной.

Отсюда возникает ***1-я задача*** — *обеспечения конфиденциальности* — предотвращение несанкционированного доступа к данным, *т.е. защита секретной информации от противника.* Для ее решения применяется *шифрование* данных.

2. При обмене информацией между участниками часто возникает ситуация, когда информация не является конфиденциальной, но важен факт, что бы сообщения поступали в неискаженном виде, т.е. должны быть гарантии, что никто не сумеет подделать сообщение.

2-я задача —*о*беспечение ***целостности*** данных — гарантии того, что при передаче или хранении данные не будут изменены несанкционированными пользователями. Основным способом решения данной проблемы является использование ***цифровой подписи***.

3. Подписывается договор между двумя или большим количеством лиц, не доверяющих друг другу. В такой ситуации все стороны должны быть уверены в том, что в будущем, во-первых, ни один из подписавших не сможет отказаться от своей подписи и, во-вторых, никто не сможет подменить или создать новый документ (договор) и утверждать, что именно этот документ был подписан.

Отсюда возникает **3-я задача** — обеспечение ***аутентификации***. А так же обеспечение ***невозможности отказа от авторства*** (невозможности отказа от подписи под документом) и ***невозможности приписывания авторства.***

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

Помимо перечисленных основных задач можно назвать также разделение секрета (распределение секретной информации между несколькими субъектами таким образом, чтобы воспользоваться ей они могли только все вместе) и многое другое.

Основные базовые методы преобразавания информации, которыми располагает криптография:

1. шифрование симметричное и несимметричное;
2. вычисление хэш-функций;
3. генерация последовательности псевдослучайных чисел;
4. генерация электронной цифровой подписи.

Тема 3. Модель криптографической системы

Простейшая модель криптографической системы.

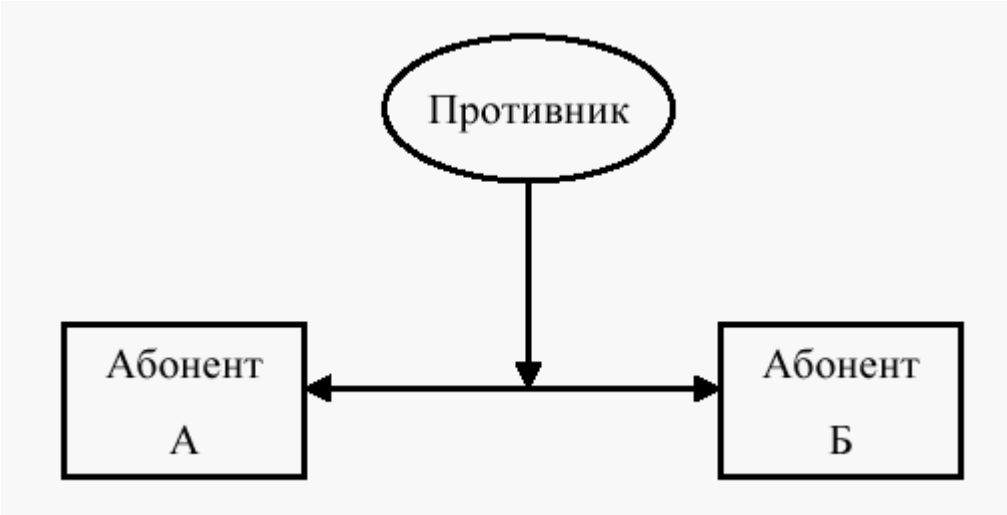

Рис . 1 Простейшая модель криптосистемы

Имеется некая информационная система, включающая двух или более *абонентов* и канал (или каналы), по которым абоненты могут обмениваться *сообщениями*. *Абонент* — законный пользователь системы. *Противник* — незаконный пользователь.

Противник может быть внешним — это не абонент системы. *Противник может быть внутренним — это абонент системы с ограниченным правом доступа.*

Противник может перехватывать сообщения с разными целями, например

- с целью разглашения перехватываемой информации
- с целью подмены сообщения и т.д.

Подобные цели называются *угрозами*. *Угроза* — это возможность перехвата информации.

Описанная модель может применяться и при защите данных, хранящихся на компьютере. В этом случае «каналом» является жесткий диск, на котором хранятся данные.

Итак, рассматривается модель, в которой противник имеет доступ к каналу передачи сообщений. Поэтому абонент, передающий сообщение (*отправитель*) должен преобразовать исходную информацию.

Открытый текст — это исходный текст документа.

Шифротекст — это криптограмма, полученная в результате преобразования открытого текста с помощью специального ключа.

Преобразование открытого текста в шифротекст называется *шифрованием.*

Расшифрование — это восстановление открытого текста из криптограммы с помощью ключа. *Ключи* — некоторые секретные данные.

Противник не знает ключ, но может попытаться *вскрыть шифр*, т.е. либо подобрать ключ, либо преобразовать зашифрованный текст в открытый каким – либо другим способом.

Термины *расшифрование* — для абонентов системы, *дешифрование* —для противника.

Способность шифра (криптосистемы) противостоять попыткам взлома (*атакам*) называется *стойкостью* шифра.

Существуют *абсолютно стойкие* системы шифрования, однако они очень не удобны и требуют больших затрат при использовании.

Реально, на практике ни одна из широко используемых систем шифрования не является абсолютно стойкой. Это означает, что если противник обладает неограниченными ресурсами например, имеет доступ к некоторым открытым текстам и соответствующим им шифротекстам, полученным с использованием одного и того же ключа, то рано или поздно он сможет взломать шифр. *Однако если выгода от использования полученной информации будет меньше, чем затраты на взлом, противник вряд ли будет этим заниматься.*

С другой стороны, при выборе алгоритма шифрования так же необходимо оценить соотношение с одной стороны ценность защищаемой информации, а другой — стойкость шифра и удобства его использования, иначе затраты на защиту информации могут превысить стоимость самой информации.

Формальная модель и классификация шифров

Пусть T — конечное множество открытых текстов,

C — конечное множество шифротекстов,

K — можество ключей

Процедура шифрования задается функцией $Ek : T \rightarrow C$, которая отображает множество открытых текстов во множество шифротекстов в зависимости от некоторого ключа $k \in K$. Аналогично, процедура дешифрования $Dk : C \rightarrow T$ также зависит от ключа k.

Любой текст, записанный с помощью букв, например, русского алфавита, можно представить в виде последовательности целых чисел (10-х, 2-х или 16-х). Это удобно для потроения алгоритмов шифрования и расшифрования, т.к.

числовые функции хорошо изучены. Для большинства современных систем шифрования множества T, C, K представлены в алфавите {0,1}, т.е. это последовательности нулей и единиц.

В 18 веке Пль Буль доказал, что код из нулей и единиц – самый плотный код. Идеи Буля реализовались в 20 веке.

При реализаци алгоритмов шифрования и дешифрования считается, что длина ключа равна длине текста. Но тексты имеют различную длину. Как быть?

Секретный ключ — это ключ фиксированной длины, т.е. это последовательность символов, длина которой не зависит от длины текста.

На его основе формируют *ключ шифрования, имеющий необходимую длину*. Это осуществляется периодическим повторением символов секретного ключа. Например, из секретного ключа $k = (k_1, k_2, \ldots, k_m)$ можно получить ключ шифрования $(k_1, k_2, \ldots, k_m, k_1, k_2, \ldots, k_m, k_1, k_2, \ldots)$ произвольной длины. Такую последовательность принято называть *гаммой*, а сам процесс — *гаммированием*. Полученный ключ используют для преобразования текста.

Современные системы шифрования можно разделить на два больших класса:

• симметричные (одноключевые) системы — в них для шифрования и расшифрования текста используется один и тот же ключ.

• асимметричные (двухключевые) системы используют различные ключи для шифрования и расшифрования текста.

Основы криптографирования

В качестве информации, подлежащей *шифрованию* и *дешифрованию,* будут рассматриваться *тексты*, построенные на некотором *алфавите*.

Алфавит - конечное множество знаков. Примеры алфавитов, используемых в современных информационных системах (ИС):

• алфавит Z33 — 32 буквы русского алфавита (без ё) и пробел;

• алфавит Z44– 32 буквы русского алфавита (без ё), арабские цифры 0,1,…9 точка и пробел;

• алфавит Z256 – символы, входящие в стандартные коды ASCII (стандарт кодов для обмена информаией);

• бинарный алфавит Z2 = {0,1};

• восьмеричный алфавит Z8 = {0,1,2,3,4,5,6,7};

• шестнадцатеричный алфавит Z16 = {0,1,2,3,4,5,6,7,8,9,10,11, 12,13, 14, 15};

алфавит Z256. Таблица кодов состоит из 2-х частей:

1) базовая таблица: коды 0-127 — невидимые управляющие символы (cs), специальные символы, арабские цифры, большие и маленькие буквы латинского алфавита.

2) Таблица для национального алфавита: Коды 128-255 невидимые управляющие символы (cs), специальные символы, большие и маленькие буквы русского алфавита (т.е. это дополнительный алфавит).

Порядковый номер символа в таблице – это 10-й код символа. Это целое число, лежащее в диапазоне 0-255 и занимающее в памяти 1 байт = 8 бит. Клод Шеннон предложил использовать слово бит для обозначения наименьшей единицы информации.

10-й код символа может быть записан в 2-ой системе, в 16-й системе счисления.

Простые операции над двоичным кодом

Клод Шеннон предложил использовать слово бит для обозначения наименьшей единицы информации.

10-я система счисления {0;1;2;3;4;5;6;7;8;9}

2- я система счисления {0;1}

16 - я система счисления {0;1;2;3;4;5;6;7;8;9;A;B;C;D;E;F}

Для нас, в криптографии, любой символ – это двоичный код. Для удобства записи используют 16-е представление двоичного числа (более сжатое).

Перевод чисел из 2-й системы в 10-ю.

$101010_2 = 1\cdot2^5+1\cdot2^3+1\cdot2=32+8+2=42_{10}$

Decimal	Binary	Hex
1	0000 0001	01
2	0000 0010	02
3	0000 0011	03
4	0000 0100	04
5	0000 0101	05
6	0000 0110	06
7	0000 0111	07
8	0000 1000	08

Decimal	Binary	Hex
9	0000 1001	09
10	0000 1010	0A
11	0000 1011	0B
12	0000 1100	0C
13	0000 1101	0D
14	0000 1110	0E
15	0000 1111	0F
16	0001 1111	10

Перевод чисел из 2-й системы в 16-ю.

$1010\ 0000_2 = A0_{16}$

Перевод чисел из 16-й системы в 10-ю.

$A0_{16} = A \cdot 16^0 + 0 \cdot 16^1 = 160 + 0 = 160_{10}$

Действия над доичным кодом

Сложение

0110 1101	$2^6 + 2^5 + 2^3 + 2^2 + 2^0 = 109_{10}$
0010 1110	$2^5 + 2^3 + 2^2 + 2^1 = 46_{10}$
1001 1011	$2^7 + 2^4 + 2^3 + 2^1 + 2^0 = 155_{10}$

Сумма по модулю 2: a+b(mod 2) — остаток от деления на 2, единица переходит в старший разряд

Простейший способ кодирования информации — выписывать 2-ые коды букв.

Логические поразрядные функции

Представляют особый интерес для криптографирования. К ним относятся:

AND – логическое И (умножение);

OR - логическое ИЛИ (сложение);

XOR – функция ИСКЛЮЧАЮЩАЯ ИЛИ (сумма по модулю 2);

NOT – отрицание НЕ (инверсия).

Последние две функции являются обратимыми.

1 – значение «истина»

0 – значение «ложь»

Таблица истинности

x	**y**	**x AND y**	**x OR y**	**x XOR y**	**NOT** x
0	0	0	0	0	1
1	0	0	1	1	0
0	1	0	1	1	1
1	1	1	1	0	0

Функция AND принимает значение «истина», когда оба операнда имеют значение «истина».

	0100 0011	C	(англ.)
AND	0100 0100	D	
	0100 0000	@	

Если выполнить туже операцию между символами @ и D, то не получим символ C. Эта операция не является обратимой.

Функция OR принимает значение «истина», когда хотя бы один операнд имеет значение «истина»

	0100 0011	C
OR	0100 0100	D
	0100 0111	G

Эта операция не является обратимой.

Функция NOT

NOT(C)	1011 1100	J
NOT(J)	0100 0011	C

Операция является обратимой.

Функция XOR или ⊕ принимает значение «истина», когда только один операнд имеет значение «истина».

	0100 0011	C	
XOR	0100 0100	D	
	0000 0111	(cs)	сумма по модулю 2, здесь единица не переходит в старший разряд.

Если выполнить туже операцию между символами (cs) и D, то получим символ C.

	0000 0111	(cs)
XOR	0100 0100	D
	0100 0011	C

Эта функция является обратимой и представляет наибольший интерес в системах шифрования.

Пример. Зашифровать слово “СЕКРЕТ”. Ключ шифра — цифровая последовательность – 123456 (см. таблицу ASCII).

С	Е	К	Р	Е	Т
1101 0001	1100 0101	1100 1010	1101 0000	1100 0101	1101 0010
1	2	3	4	5	6
0011 0001	0011 0010	0011 0011	0011 0100	0011 0101	0011 0110
1110 0000	1111 0111	1111 1001	1110 0100	1111 0000	1110 0100
а	ч	щ	д	р	д
224	247	249	228	240	228

В результате шифрования получен набор букв русского алфавита “ачщдрд”. Усложним шифр заменой символов на их эквивалентные десятичные коды (номера символов).

10-е коды могут быть заменены на 2-ые коды, поэтому шифротекст может быть сохранен, либо передан получателю в виде текстового файла, или в 10-х кодах, или в виде бинарного файла.

Расшифрование файла выполняется в обратной последовательности. Получатель должен знать ключ. Длина текста должна быть равна длине ключа, но ключ запомнить невозможно, поэтому используется *гаммирование.*

Логические сдвиги

Левый сдвиг *SHL число позиций* — сдвиг всех битов на указанное число позиций влево. Недостающие значения битов дополняются нулями.

Пример. $0010\ 0000_2 \rightarrow 2^5=32_{10}=20_{16}$

SHL 1 $0100\ 0000_2 \rightarrow 2^6=64_{10}=40_{16}$

Левый сдвиг на 1 разряд равносилен умножению числа на 2

на 2 разряда равносилен умножению числа на 4

на 3 разряда равносилен умножению числа на 8 и т.д.

Это целочисленное умножение. Если 1 выскакивает, то все нули.

Правый сдвиг *SHR число позиций* — сдвиг всех битов на указанное число позиций вправо. Недостающие значения битов дополняются нулями.

Пример. $0010\ 0000_2 \rightarrow 2^5=32_{10}=20_{16}$

SHR 1 $0001\ 0000_2 \rightarrow 2^4=16_{10}=10_{16}$

SHR 1 $0000\ 1000_2 \rightarrow 2^3=8_{10}=08_{16}$

Правый сдвиг на 1 разряд равносилен дедению числа на 2

на 2 разряда равносилен делению числа на 4 (видно в 10-й системе.)

Пример. Каким будет результат в 16-й системе X=($C8 and $A7) SHR2

$-признак того, что число представлено в 16-й системе счисления (м.б. H).

B 1100 1000

AND B 1010 0111

1000 0000

SHR 2 $0010\ 0000_2 \rightarrow 2^5=32_{10}=20_{16}$

Тема 4. Симметричные системы шифрования

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

Симметричные системы шифрования можно разделить на:

- *шифры перестановки* — для получения шифротекста символы открытого текста переставляются местами;
- *шифры замены* — символы открытого текста заменяются другими символами или группами символов;
- *композиционные шифры* — наиболее распространённые шифры, представляют собой последовательное применение нескольких алгоритмов шифрования разных типов.

Шифры перестановки

Шифры перестановки — это получение шифротекта путём перестановки символов в открытом тексте.

Открытый текст записывается в некоторую геометрическую фигуру (например матрицу размерностью m×n) по некоторой траекории, а затем, выписывая символы из этой фигуры по другой траектории, получаем шифротекст.

Пример 1. Зашифровать фразу «*это маршрутная перестановка*».

а) простая перестановка

Порядок действий:

1. Выбираем *ключ* – количество столбцов (пусть 9), столбцы нумеруются попорядку.

2. Открытый текст записываем в матрицу слева направо без пробелов, либо в качестве разделителей может быть пробел или *. Если букв не хватает, то дописываем любые буквы либо * (см. рис.2).

1	2	3	4	5	6	7	8	9
э	т	о	м	а	р	ш	р	у
т	н	а	я	п	е	р	е	с
т	а	н	о	в	к	а		

Рис. 2. Пример простой перестановки

3. **Для шифрования** выписываем буквы, двигаясь по столбцам сверху вниз.

этт *1-й столбец в 1-ую строку*

тна *2-й столбец во 2-ю строку*

оан *……. и т.д*

В качестве усложнения выписываем всё в 1 строку без промежутков:

эттт наоанмяоапврекршареус — это шифротекст

Расшифрование:

- подсчитать количество символов в шифротексте (27),
- необходимо знать ключ (количество столбцов 9); количество строк 27/9=3;

- выписываем по 3 символа в столбец согласно нумерации столбцов.

Криптоанализ. Если не известно количество столбцов, то перебираем пары чисел такие, что бы m×n =27.

Это шифры низкой стойкости, они используются только как составная часть композиционных шифров.

б) Маршрутная транспозиция

Выбираем слово-ключ. Требования к ключу:

1) в качестве ключа должно быть слово, в котором нет повторяющихся букв;
2) количество столбцов матрицы = количеству букв ключа. Число строк любое.

Пример 2. Зашифровать фразу «*это маршрутная перестановка*», ключ — ПОБЕДА (6 столбцов).

Порядок действий.

1. Столбцы нумеруются в порядке следования букв ключа в алфавите.
2. Открытый текст записывается в матрицу, содержащую 6 столбцов.

п	**о**	**б**	**е**	**д**	**а**
6	**5**	**2**	**4**	**3**	**1**
э	т	о	м	а	р
ш	р	у	т	н	а
я	п	е	р	е	с
т	а	н	о	в	к
а	*	*	*	*	*

3. Для шифрования выписывается текст по столбцам в 1 строку без промежутков:

раск*оуен*анев*мтро*трпа*эшята это шифротекст

Расшифрование:

- необходимо знать слово-ключ, по которому определяется количество столбцов (6) и нумерация столбцов;
- считаем количество символов в шифротексте (30) → количество строк 5.
- выписывать по 5 символов в столбцы, согласно нумерации столбцов.

Криптоанализ.

- подсчитывается количество символов в шифротексте;
- не зная ключа, подбирается количество столбцов (m×n=30);
- подбирается порядок следования столбцов;
- количества итераций $= \left(\frac{\textit{длина текста}}{2}\right)^2$. Для нашего случая $(\frac{30}{2})^2 = 225$

При современной выч. технике это реально

Шифры замены

Шифры замены — это получение шифротекста путём замены фрагментов открытого текста (символов, групп символов) другими символами.

Простейшим из шифров замены является *одноалфавитная подставновка*, называемая также шифром *простой замены*.

Пусть X — алфавит открытого текста; Y — алфавит шифротекста.

Ключом такого шифра является взаимно однозначное отображение F: $X \leftrightarrow Y$,

Этот ключ может быть задан либо таблицей, либо с помощью формулы.

Примером шифра простой замены является *шифр Цезаря* (1 век н.э.).

X — латинский алфавит. При этом:

- каждая буква алфавита с номером i заменяется буквой, стоящей на три позиции правее нее в алфавите, то есть смещение на 3 буквы, i→i+3; ключом шифра является число 3;
- ключ для шифра Цезаря можно задать в виде таблицы (рис. 3). В первой строке записан алфавит открытого текста, во второй — соответствующие им буквы алфавита шифротекста.

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

Рис . 3. Ключ для шифра Цезаря

Алфавит считается записанным по кругу, то есть после буквы ‘z’ идет буква ‘a’

Открытый текст ‘secret’ будет преобразован в ‘vhfuhw’.

Возможен *обобщенный вариант шифра Цезаря*, при котором буква с номером *i* заменяется на букву с номером *i+k,* то есть i→i+к. В этом случае ключом шифра является число *k*. Зная *k* можно составить подобную таблицу, шифрование аналогично.

Иначе: ключ может быть задан с помощью формулы, для этого

- пронумеруем буквы алфавита от 1 до 26 (с каждой буквой связывается число); $t_i \rightarrow t_{i+k}$

a	**b**	**c**			**t**		**x**	**y**	**z**
1	**2**	**3**			**20**		**24**	**25**	**26**

- букву алфавита шифротекста получаем по правилу:

$$c_i = \begin{cases} t_{i+k}, & i+k \le 26 \\ t_{i+k-26}, & i+k > 26 \end{cases}$$ буква ↔ номер

c_i - буква с номером i в шифротексте;

t_{i+k} - буква с номером i+к в открытом тексте

Проверим формулу при к=3: вместо 1-ой буквы **a** — буква с номером 4 (**d**)
вместо буквы **x** с номером 24 будет буква с номером 24+3-26=1 (**a)**
Пусть к=10, тогда вместо буквы **t** с номером 20 будет буква с номером 30-26=4 (**d)**

Для удобства реализации на ПК пронумеруем буквы латинского алфавита числами от 0 до 25:

a	**b**	**c**		**x**	**y**	**z**
0	**1**	**2**		**23**	**24**	**25**

Тогда правило замены для шифра Цезаря:
$c_i = t_i + 3 (\mathrm{mod}\, 26)$, где операция ‘mod 26’ означает вычисление остатка от деления на 26. Это числа от 0 до 25; 26 — количество символов в алфавите.
Для обобщённого варианта Шифра Цезаря:
$c_i = t_i + k (\mathrm{mod}\, 26)$ буква с номером *i* заменяется на букву с номером *i+к*

Криптоанализ. Стойкость шифров простой замены невелика, поэтому в настоящее время они не используются. Методы взлома таких шифров основаны на анализе частоты появления символов. Например, в текстах на русском языке чаще всего встречается буква ‘О ’, затем, в порядке убывания частоты, идут буквы ‘Е ’ (считая, что ‘Е ’ и ‘Ё ’ — одна и та же буква), ‘А ’, ‘И ’,‘Т’ и т. д. Для английского языка аналогичная последовательность самых частых букв: ‘E’, ‘T’, ‘A’, ‘I’, ‘N’. Самым частым символом в текстах является, однако, не буква, а символ пробела.

Частота повторений символов в шифротексте совпадает с частотой повторений соответствующих символов в открытом тексте. Это позволяет достаточно легко вскрыть такой шифр. Таким образом *атакой на шифр замены является* анализ частот вхождения символов в шифротекст.

Для того чтобы затруднить взлом шифра замены, можно скрыть частотные свойства исходного текста. Для этого необходимо, чтобы частоты появления разных символов тексте совпадали. Такие шифры замены называются *гомофоническими*.

Для того чтобы увеличить стойкость шифров замены, применяют *многоалфавитную замену*. Примером является шифр Виженера.

Шифр Виженера

При шифровании каждого i-го символа открытого текста применяется новый алфавит.

Для удобства шифрования и расшифрования вручную используется *таблица Виженера*. Устройство таблицы:

- в 1-ой строке выписывается алфавит, Z31 (без ё, ъ);
- в каждой следующей строке осуществляется сдвиг на одну букву в алфавите, то есть каждая следующая строка это новый алфавит.

Так получается квадратная таблица, число строк в которой равно числу столбцов и равно количеству букв в алфавите.

Таблица Виженера

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

Идея шифрования :

- выбирается слово-ключ (лозунг) и подписывается под буквами сообщения с повторением;
- букву сообщения выбираем из 1-ой строки (горизонтальный алфавит);
  букву ключа выбираем из 1-го столбца (вертикальный алфавит);
  на пересечении — буква шифротекста.

Может быть наоборот: буква сообщения – 1-ый столбец, буква ключа – 1-я строка

В шифре Виженера 2 секретные составляющие:

- алфавит может быть любой, например 06А9….
- слово-ключ (лозунг).

Пример. Зашифровать слово ТЕЛЕФОН, лозунг ГОЛОС.

Т Е Л Е Ф О Н открытый текст Т

Г О Л О С Г О ключ К

Х У Ц У Ж С Ы шифротекст С

Расшифрование

1. Под шифротекстом подписываем ключ (лозунг) с повторением.
2. Выбираем букву ключа в 1-ом столбце, букву шифротекста — в таблице, букву открытого текста в 1-ой строке.

Х У Ц У Ж С Ы шифротекст С

Г О Л О С Г О ключ К

Т Е Л Е Ф О Н открытый текст Т

Если не пользоваться таблицей, то пронумеруем символы алфавита от 1 до 31.

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Ф	Х	Ц	Ч	Ш	Щ	ь	ы	Э	Ю	Я									
21	22	23	24	25	26	27	28	29	30	31									

Для получения буквы шифротекста воспользуемся формулой:

$$c_i = \begin{cases} t_i + k_i - 1, & t_i + k_i - 1 \le 31 \\ t_i + k_i - 1 - 31, & t_i + k_i - 1 > 31 \end{cases}$$

c_i - буква в шифротексте;

t_i - буква в открытом тексте; k_i - буква ключа

Т	Е	Л	Е	Ф	О	Н
Г	О	Л	О	С	Г	О
19	6	12	6	21	15	14
4	15	12	15	18	4	15
22	20	23	20	7	18	28
Х	У	Ц	У	Ж	С	Ы

В действительности мы работаем не с символами алфавита, а с чисами (номерами букв). Существует взаимооднозначное отображение *символ* ↔ *число.*

Для удобства реализации шифра Виженера на ПК пронумеруем буквы алфавита от 0 до 30.

В качестве секретного ключа выберем набор из m целых чисел $k = (k_1, k_2, \ldots, k_m)$. В качестве ключа шифрования используется бесконечная последовательность $k_1, k_2, \ldots, k_m, k_1, k_2, \ldots, k_m, k_1, k_2, \ldots$ — *гамма.*

Преобразования открытого текста $t = (t_1, t_2, \ldots)$ в шифртекст $c = (c_1, c_2, \ldots)$ строим на основе обобщенного шифра Цезаря $c_i = t_i + k_i \ (mod\ 31)$,

где операция ‘mod31’ означает вычисление остатка от деления на 31. Это числа от 0 до 30. 31 — количество символов в алфавите

Модифицированный шифр Цезаря

Был предложен в 1518 г. аббатом Тритемеусом в первой печатной книге о тайнописи. Этот шифр можно считать усовершенствованным шифром Цезаря. Идея шифра:

- все буквы русского алфавита нумеруются по порядку от 1 до 31 (всего 33 буквы, отсутствуют ъ и ё).
- выбирается слово-ключ и подписывается под сообщением с повторением;
- складывают номер очередной буквы открытого текста с номером соответствующей буквы ключа:
  - если полученная сумма меньше или равна 31, то её оставляют,
  - если полученная сумма больше 31, то из нее вычитают 31.
- заменяем числа полученной последовательности на буквы, получаем шифрованный текст.
- разбиваем этот текст на группы одной длины (для удобства), получаем шифрованное сообщение.

Пример 1. Зашифровать слово СЕКРЕТ. Слово-ключ "ПРАВДА".

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Ф	Х	Ц	Ч	Ш	Щ	Ы	Ь	Э	Ю	Я									
21	22	23	24	25	26	27	28	29	30	31									

Шифрование:

$$c_i = \begin{cases} t_i + k_i, & t_i + k_i \le 31 \\ t_i + k_i - 31, & t_i + k_i > 31 \end{cases}$$

Т:	С	Е	К	Р	Е	Т
К:	П	Р	А	В	Д	А
	18	6	11	17	6	19
	16	17	1	3	5	1
	3	23	12	20	11	20
С:	В	Ц	Л	У	К	У

c_i - номер буквы в шифротексте;

t_i - номер буквы в открытом тексте;

k_i - номер буквы ключа

Расшифрование: Т = С-К $\qquad t_i = \begin{cases} c_i - k_i, & c_i - k_i \geq 0 \\ c_i - k_i + 31, & c_i - k_i \leq 0 \end{cases}$

С:	В	Ц	Л	У	К	У	
К:	П	Р	А	В	Д	А	
	3	23	12	20	11	20	3-16+31=18
	16	17	1	3	5	1	
	18	6	11	17	6	19	
Т:	С	Е	К	Р	Е	Т	

Отличие от обобщённого шифра Цезаря: там не было слова-ключа.

Для удобства реализации на ПК пронумеруем буквы русского алфавита от 0 до 30.

А	**Б**	**В**		**Э**	**Ю**	**Я**
0	**1**	**2**		**28**	**29**	**30**

Тогда $c_i = t_i + k_i (\mathrm{mod}\, 31)$

Это целые числа от 0 до 30. Для полного русского алфавита (33 б.) всё аналогично.

Замечания.

1. Если секретный ключ меньше длины открытого текста, то ключ шифрования — повторение секретного ключа (гамма).

Зашифровать сообщение "У Лукоморья дуб зелёный". Ключ — ПОСОБИЕ

Ключ подписывается под сообщением с повторением:

У л у к о м о р ь я д у б з е л ё н ы й

п о с о б и е п о с о б и е п о с о б и

Далее аналогично.

2. Если под ключом шифрования понимать 1 букву, то получим обобщённый шифр Цезаря.

Например С Е К Р Е Т

Л Л Л Л Л Л

Криптоанализ. Взломать шифры многоалфавитной замены (Виженера) и модифицированный шифр Цезаря сложнее, чем шифры простой замены, но тоже легко. Если известен период гаммы (т.е. число m), то к каждой такой части можно применить любой из методов взлома шифров простой замены. Если период гаммы

не известен, то сложнее. Но и для этих случаев разработаны эффективные методы, позволяющие с достаточной вероятностью определить период гаммы.

Гаммирование

Благодаря удобству реализации шифры гаммирования широко используются и их выделяют в отдельный класс.

Шифр гаммирование — это разновидность шифра замены реализованная путём наложения символов ключа на открытый текст.

Идея метода:

1. генерируется последовательность целых чисел g_1, g_2,…, g_i,… (гамма);
2. при шифровании гамма *накладывается* на открытый текст $t = t_1, t_2, \ldots t_i \ldots,$;
3. символы шифротекста получаются с помощью некоторой обратимой операции: $c_i = t_i \bullet g_i$, $i=1,2,...$

Знак (•) – некоторая обратимая операция.

c_i - номер буквы в шифротексте;

t_i - номер буквы в открытом тексте;

k_i - номер буквы ключа.

В качестве обратимой операции может быть:

1. сложение по модулю *N*, где *N* —количества букв в алфавите

$$c_i = t_i + g_i (\mathrm{mod}\, N)$$ (шифр Виженера, модиф. Шифр Цезаря)

2. операция XOR (поразрядное суммирование по модулю 2), при этом символы открытого текста и ключа д.б. в виде двоичного кода.

$$c_i = t_i \oplus g_i$$

Расшифрование осуществляется с помощью обратных операций:

1. $t_i = c_i - g_i (\mathrm{mod}\, N)$,

(если $c_i < g_i$, то $t_i = c_i + N - g_i (\mathrm{mod}\, N)$)

2. $t_i = c_i \oplus g_i$ (операция XOR является обратной к самой себе).

Стойкость шифра зависит от характеристик гаммы. Наиболее стойким является гаммирование, удовлетворяющее следующим условиям:

1) все символы гаммы полностью случайны и появляются в гамме с равными вероятностями (периода нет);

2) длина гаммы равна длине открытого текста или превышает ее;

3) каждый ключ (гамма) используется для шифрования только одного текста, а потом уничтожается.

Такой шифр не может быть взломан в принципе, то есть является *абсолютно стойким,* т.к. любая буква шифротекста получается случайно. Однако абсолютно стойкие шифры очень не удобны в использовании, и поэтому почти не применяются на практике.

Обычно гамма получается периодическим повторением секретного ключа фиксированного размера (период известен), либо генерируется по некоторому правилу.

Для генерации гаммы используются генераторы псевдослучайных чисел, которые основаны на рекуррентных математических формулах. Например

$$g_i = ag_{i-1} + b(\mathrm{mod}\, m) \qquad \text{i=1, 2,…}$$

Где g_i — i-й член последовательности псевдослучайных чисел;

a, b, m и g_0 — ключевые параметры.

Свойства последовательности:

1. последовательность $\{g_i\}$ состоит из целых чисел от 0 до m–1. Чем больше m, тем больше разных чисел.

2. последовательность $\{g_i\}$ является периодической и ее период не превышает m. Если элементы g_i и g_j совпадут, то последующие участки последовательности также совпадут, т.е. если: $g_i = g_j$, то $g_{i+1} = g_{j+1}$, $g_{i+2} = g_{j+2}$, и т.д.

Чем больше период гаммы, тем лучше.

3. Для того чтобы период последовательности псевдослучайных чисел, был максимальным (равным *m*), параметры должны удовлетворять следующим условиям:

b и *m* —взаимно простые числа, т.е. НОД(b;m)=1;

a–1 делится на любой простой делитель числа m;

a–1 кратно 4, если *m* кратно 4.

Одноразовый блокнот

Почти все используемые на практике шифры характеризуются как условно надежные, т.к. они могут быть раскрыты при наличии неограниченных вычислительных возможностей. Абсолютно надежные шифры разрушить нельзя.

В 70-х годах XX века Клод Шеннон (амер. учёный) доказал существование и единственность абсолютно надежного шифра.

На практике таким шифром является так называемый *одноразовый блокнот*, в основе которого лежит та же идея, что и шифре Цезаря:

1) используется расширенный алфавит Z44:

0 1 2 3 4 5 6 7 8 9 А Б В …Ю Я (без ё)· **.** **,**

2) все символы алфавита нумеруются числами от 0 до 43. Поэтому каждый символ сообщения есть число $\alpha_n \in A=\{0,1,2,\ldots,43\}$ n=1,2, … Передаваемое сообщение — есть последовательность чисел.

3) создаётся ключ той же длины что и передаваемый текст (за счёт гаммирования). Каждый символ ключа есть случайное число $c_n \in A$, n=1,2,…

4) символы шифротекста b_n получаются путём сложения по модулю 44 символов открытого текста a_n и ключа c_n.

Алгоритма шифрования принятый в записи теории чисел:

$$a_n+c_n \equiv b_n (\mathbf{mod}\ 44),\ 0 \le b_n \le 43,$$

Дешифрование с помощью того же ключа:

$$a_n \equiv b_n - c_n (\mathbf{mod}\ 44),\ 0 \le a_n \le 43.$$

Если $b_n < c_n$, то $a_n = b_n + 44 - c_n (\mathrm{mod}\, 44)$

У двух абонентов, находящихся в секретной переписке, имеются два одинаковых блокнота, составленных из отрывных страниц. На каждой странице напечатан алфавит, символы которого пронумерованы *случайным* образом.

Отправитель шифрует свой текст указанным выше способом при помощи первой страницы блокнота, ключ известен заранее. Зашифровав сообщение и отправив его абоненту, он уничтожает использованную страницу. Получатель

шифрованного текста расшифровывает его и также уничтожает использованный лист блокнота.

Шифр не раскрывается в принципе, так как любая буква алфавита представляет собой число случайным образом, а значит и буква в шифротексте — случайна.

Чтобы изготовить страницу одноразового блокнота используются генераторы случайных чисел. Вариант результата работы генератора для одной строки приведен в таблице случайных отношений.

Исходная таблица

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
0	1	2	3	4	5	6	7	8	9	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	.	,

Таблица случайных отношений

5	16	22	7	32	1	24	8	31	21	6	33	19	23	3	37	30	15	42	9	41	0	43	18	17	40	39	10	34	4	35	20	25	14	2	27	36	26	12	38	28	13	29	11
0	1	2	3	4	5	6	7	8	9	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	.	,

Как видно, генератор воздействовал только на цифровую последовательность. В реальных условиях генератор воздействует и на символьную последовательность, что приводит к случайному распределению символов и во второй строке таблицы.

Неизвестные составляющие:

- алфавит, на каждой странице случайная нумерация символов;
- ключ — случайная последовательность, равная длине сообщения.

Одноразовый блокнот, он же “Русский Блокнот” эффективно использовался в годы Великой Отечественной Войны. Вскрыть его не удалось. Он остается по–прежнему актуальным, т.к. легко аппаратно реализуется. Передача информации осуществляется по свободным каналам связи в виде передаваемых групп цифр.

Криптография с одним ключом (симметричные системы)

Один и тот же ключ используется как для шифрования, так и для расшифровки сообщения. Современный криптографический ключ — это последовательность чисел определённой длины, созданная с помощью генератора случайных чисел.

Фундаментальное правило криптоанализа — стойкость шифра должна определяться только секретностью ключа, т.е. весь алгоритм шифрования, кроме ключа, известен противнику.

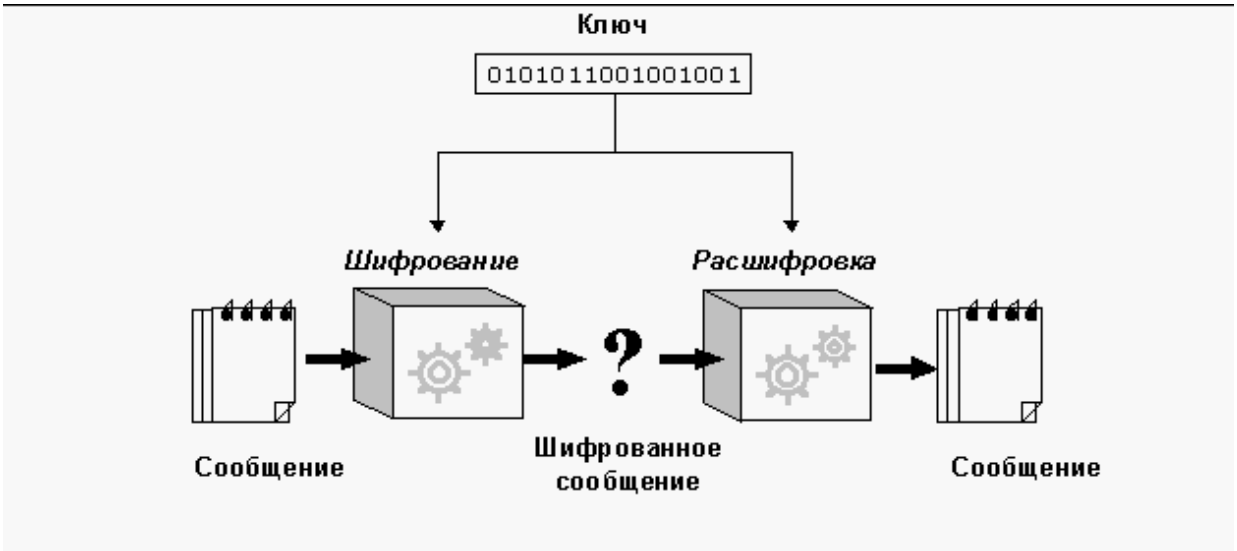

Рис. 1 Общая схема шифрования с 1 секретным ключом

Неудобства.

1. Прежде чем начать обмен информацией, надо выполнить передачу ключа, а для этого нужна защищённая связь.

2. Симметричные алгоритмы работают быстро и сравнительно просты, но они требуют знания ключа по крайней мере двумя людьми, что повышает риск доступа посторонних.

3. Пусть одно и тоже сообщение должно быть передано многим адресатам. Для этого оно должно много раз шифроваться разными ключами (затраты времени).

Поэтому в настоящее время в Интернете используются несимметричные криптографические системы, основанные на использовании не одного, а двух ключей.

Тема 5. Асимметричные криптографические системы

Основы асимметричных систем

В 1976 году была опубликована работа американских математиков У.Диффи и М.Хеллмана «Новые направления в криптографии». Ими была предложена абсолютно «новая» криптография — криптография с открытым ключом, ассиметричное шифрование. Современный период развития криптографии (70-е годы и до наших дней) — криптография с открытым ключом.

В ассиметричных системах для обмена данными используются два ключа, один *открытый* (public - публичный) а другой *закрытый* (private - личный). Открытый ключ доступен любому желающему, передаётся по незащищённому каналу связи. Закрытый ключ известен только владельцу ключа (обычно сохраняется на месте генерации).

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

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

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

2) не существует *эффективного алгоритма инвертирования функции*, т.е. алгоритма, позволяющего определить значение *x* по значению y=*F(x),* используя современные вычислительные средства за обозримый промежуток времени..

В качестве примера односторонней функции рассмотрим функцию дискретного возведения в степень $y = \alpha^x (\mathrm{mod}\, p)$, где

p– очень большое простое число; $y \in [0; p-1]$

α – натуральное число, $a \in (1; p-1]$

x– натуральное число, $x \in [1; p-1]$

Пример. Возьмем небольшое простое число p=7; пусть α=3, тогда при x=1;2; …

$\alpha^1 (\mathrm{mod}\, 7) = 3$; $\alpha^2 (\mathrm{mod}\, 7) = 9 (\mathrm{mod}\, 7) = 2$;

$\alpha^3 (\mathrm{mod}\, 7) = 6$; $\alpha^4 (\mathrm{mod}\, 7) = 4$; $\alpha^5 (\mathrm{mod}\, 7) = 5$; $\alpha^6 (\mathrm{mod}\, 7) = 1$

Функция $y = \alpha^x (\mathrm{mod}\, p)$ вычисляется просто, но если P велико, то вычисление обратной к ней функции $x = \log_a y (\mathrm{mod}\ \mathrm{p})$ оказывается невозможным за приемлемое время. Поэтому функция считается односторонней.

Пример хранения паролей в компьютере. Любой пользователь имеет свой пароль в сети. При входе он указывает имя и вводит пароль. Но если хранить пароль на диске, то кто-нибудь его может считать (администратор) и получить доступ к секретной информации. Для решения этой проблемы используется односторонняя функция.

В компьютере сохраняется не сам пароль, а результат вычисления функции от этого пароля и имени пользователя f(имя_пароль)=необратимая функция. Файл паролей может быть просмотрен другими пользователями без потери секретности.

Пусть *x*— код символа в T; y — код символа в C.

Если преобразовать открытый текст T с помощью односторонней функции, то расшифровать его, то есть по C восстановить T, не сможет уже никто, в том числе и законный получатель.

Для того что бы текст можно было восстановить, используются односторонние функции с *секретом* (то есть нечто, что помогает расшифровать).

Односторонняя функция с секретом — это функция, имеющая параметр K, при котором осуществляется алгоритм инвертирования.

Множество классов односторонних функций порождает всё разнообразие систем с открытым ключом.

Строгое существование односторонних функций и функций с секретом не доказано. Но существуют функции, для которых инвертирование является сложной задачей.

Проблема распределения ключей

Два абонента находятся на значительном расстоянии. Что бы осуществлять обмен информацией необходимо сгенерировать ключ и передать его. Как?

Пусть в системе имеется *N* абонентов. Для обеспечения секретного обмена информацией между любыми двумя абонентами потребуется:

1. сгенерировать и распределить (передать) N(N–1)/2 секретных ключей;
2. каждый абонент будет вынужден хранить N–1 секретный ключ для обмена информацией с остальными абонентами.

Проблема *распределения секретных ключей* между абонентами и проблема *хранения этих ключей* является одной из основных проблем на практике. В

симметричных систем шифрования, для распределения ключей необходимо устанавливать специальный защищенный канал.

Асимметричные системы позволяют распределять ключи по открытым каналам, т.е. каналам, которые потенциально могут быть прослушаны противником.

Процедура открытого распределения ключей

Процедура открытого распределения ключей была впервые опубликована в 1976 году У. Диффи и Э. Хеллманом. Эта процедура позволяет двум и более сторонам получить *общий секретный ключ*, используя открытый канал связи.

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

$Y(x) = g^x \pmod p$, где

p — большое простое число,

g – первообразный корень по модулю p, $g < p$;

x– натуральное число, $x \in [1; p-1]$ — ключ

g, p — числа, не являющиеся секретом; чем больше p, тем больше ключ X.

Идея процедуры.

1. Каждый абонент в качестве *своего закрытого* ключа генерирует случайное число x, по которому вычисляет свой *открытый* ключ $y = g^x \pmod p$
2. Все абоненты помещают свои открытые ключи в общедоступный справочник.
3. Если два абонента, A и B, захотят обменяться секретным сообщением, они берут из справочника открытые ключи друг друга (соответственно, Y_A и Y_B) и вычисляют *общий секретный ключ*:

$Y_A = g^{x_A} \pmod p$ — открытый ключ абонента A

$Y_B = g^{x_B} \pmod p$ — открытый ключ абонента B

абонент A вычисляет

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

абонент В вычисляет

$$z_{\mathrm{B}} = (y_{\mathrm{A}})^{x_{\mathrm{B}}} = (g^{x_{\mathrm{A}}})^{x_{\mathrm{B}}} \pmod{p} = g^{x_{\mathrm{A}} x_{\mathrm{B}}} \pmod{p}.$$

Таким образом, у абонентов А и В есть общее число

$K = Z_A = Z_B = g^{x_A x_B} \pmod{p}$ — общий секретный ключ. Он может быть использован в качестве ключа для шифрования в алгоритмах симметричного шифрования. Например

C=T xor K —шифрование T=C xor K — расшифрование

Выводы. Процедура распределения ключей Диффи–Хеллмана позволяет:

- передавать ключи по открытому каналу связи;
- хранить каждому абоненту только 1 закрытый ключ;
- получить общий секретный ключ.

Криптоанализ. Противник знает открытые ключи $Y_A = g^{x_A} \ (mod \ p)$ и $Y_B = g^{x_B} \ (mod \ p)$, но если число P выбрано достаточно большим, то определить закрытый ключ, он не сможет за разумное время.

На современных ПК генерация открытого ключа по известному секретному ключу производится за секунды, а на обратное преобразование, в зависимости от длины ключа, может уйти до сотен (тысяч) лет.

Тема 6. Криптосистема RSA

Основные понятия

Компания для работы с клиентами создаёт два ключа: один *открытый* (public-публичный) а другой *закрытый* (private - личный). Это две «половинки» одного целого ключа, связанные друг с другом.

Ключи устроены так, что сообщение, зашифрованное одной половинкой, можно расшифровать только другой половинкой. Создав пару ключей, торговая компания широко распространяет публичный ключ (открытую половинку) и надёжно сохраняет закрытый ключ (свою половинку).

Публичный ключ компании может быть опубликован на её сервере, откуда каждый желающий может его получить. Если клиент хочет сделать фирме заказ, он возьмёт её публичный ключ и с его помощью закодирует своё сообщение о заказе и данные о своей кредитной карте. Это сообщение может прочесть только владелец закрытого ключа. Никто из участников цепочки, по которой пересылается информация, не в состоянии это сделать. Даже сам отправитель не может прочитать собственное сообщение, хотя ему хорошо известно его содержание.

Если фирме надо будет отправить клиенту квитанцию о том, что заказ принят к исполнению, она закодирует его своим закрытым ключом. Клиент сможет прочитать квитанцию, воспользовавшись имеющимся у него публичным ключом данной фирмы. Он может быть уверен, что квитанцию отправила именно эта фирма, поскольку никто другой доступа к закрытому ключу фирмы не имеет.

Алгоритм RSA

Является самым известным ассиметричным алгоритмом шифрования на сегодняшний день. Он назван по первым буквам фамилий разработчиков этой криптосистемы (Rivest, Shamir, Adleman). Этот алгоритм позволяет общаться через не защищённый канал, уже не требуя, что бы каждая сторона имела копию одного и того же секретного ключа.

Эта система используется как для шифрования данных, так и для формирования цифровой подписи. В основе — использование односторонних функций. Система шифрования RSA устроена следующим образом:

1. Каждый абонент формирует для себя ключевую пару: закрытый и открытый ключи:

- генерируется пара больших простых чисел *p* и *q*. Вычисляется $n = pq$ и m=(p-1)(q-1)*;*
- выбираются целые числа (показатели степени) e и d, удовлетворяющие условиям: $ed = 1 (\mathrm{mod}\ m)$, e<m,
  НОД(e, m)=1, т.е. e и m — взаимнопростые числа;
- в качестве *открытого ключа* абонента *выступает пара чисел (n, e),* а в качестве *закрытого ключа — число d.*

2. Открытые ключи всех абонентов помещаются в общедоступный справочник.

Шифрование идёт открытым ключом получателя, а расшифрование — закрытым ключом.

Функция шифрования сообщения в системе RSA:

T — открытый текст, представленный в виде числа; C — шифротекст

Для шифрования используется открытый ключ адресата *(n, e)*:

$$C = T^e (\mathrm{mod}\ n) \qquad C \in [0; n-1]$$

Функция расшифрования: получатель использует свой закрытый ключ *число d:*

$C^d (\mathrm{mod}\ n) = (T^e)^d (\mathrm{mod}\ n) = T (\mathrm{mod}\ n) = T$ (по теореме Эйлера)

Шифрование и расшифрование представляют собой операции над целыми числами. Считается, что все целые числа ≥ нуля и меньше некоторого заданного числа m. (Возведение в степень осуществляется как ряд умножений). Возможности ЭВМ имеют границы.

Приходится длинные сообщения разбиваеть на блоки длиной $\log_2 n$ (чтобы каждый блок представлял собой число, меньшее *n*). Каждый блок шифруется и расшифровывается отдельно.

Алгоритмы для построения больших простых чисел (длиной 512 и более бит), удовлетворяющие условию, сложны для понимания.

Криптоанализ. Как определить закрытый ключ d (ed =1(mod m)) **?**

Если числа p и q известны, то ищется m, а по открытому ключу (n, e) и полученному m находится секретный ключ d.

Пусть числа p и q не известны, но n=pq. Не существует эффективного алгоритма разложения числа n на простые множители (задача факторизации). Функция n=pq — односторонняя.

Если завтра гениальный математик найдёт способ факторизации больших чисел, то мы лишимся ассиметричной криптографии, но, по оценке специалистов, вероятность этого ≈ 0.

Длины чисел p и q должны состоять $\approx$из 100 десятичных знаков каждое. При этом число n оказывается состоящим уже из 200 10-х знаков (1024-2048 бит). Для современной вычислительной техники производить такие расчёты задача не простая.

Ключи. 512-битный ключ RSA может быть вскрыт за 6 месяцев (1999 г.), поэтому он не обеспечивает дотаточной безопасности (используется для краткосрочных задач).

Для общих задач рекомендуются ключи размером 1024 бита, а для особо важных задач — 2048 бит.

Обычно ключ индивидуального пользования имеет определённый срок жизни, например 1 год. Поэтому существует потребность регулярно заменять ключи и обеспечивать необходимый уровень безопасности.

Особенности использования асимметричных криптосистем на практике

1. По эффективности асимметричные системы проигрывают симметричным, они работают на 2-3 порядка медленнее и требуют больших вычислительных ресурсов.
2. Стойкость асимметричных систем основана на предположении о существовании односторонних функций. Это предположение пока не доказано, хотя и не опровергнуто.
3. Абсолютно стойкие системы шифрования есть только в классе симметричных систем.

Зачем использовать ассиметричные системы шифрования?

1. Упрощается процедура распределения ключей между абонентами.
2. Становится возможным использование электронной цифровой подписи (ЭЦП).
3. Решаются вопросы аутентификации и др.

Платой за это является время и длина ключа (1024 бит и больше).

Поэтому обычно асимметричные системы используют не самостоятельно, а в комплексе с симметричными системами.

Тема 7. Способы повышения стойкости шифров

Стойкость большинства алгоритмов шифрования можно существенно повысить, модифицировав сам алгоритм.

Определение. Система шифрования называется *поточной*, если символы исходного текста последовательно (побитово) заменяются на символы шифротекста в соответствии с некоторым алгоритмом: $c_i = E_k(t_i)$ (E_k – алгоритм шифрования по ключу *к*).

Шифры замены и гаммирования относятся к поточным системам шифрования.

Система шифрования называется *блочной,* если исходный текст разбивается на блоки, и алгоритм шифрования применяется к каждому блоку, как к единому целому. Блочные шифры — разновидность симметричных шифров. Шифр перестановки является блочным. Обычно размер блока — 64 бита. Сейчас ключ шифрования— 32 байта.

«Классическим шифрованием» принято называть симметричные блочные шифры.

Блочный шифр DES, разработанный в 70-х годах XX века, долгое время служил стандартом для шифрования в США. Потом появились IDEA, советский ГОСТ 28147 89 (является стандартом).

Блочные шифры надёжны, но медленнее поточных.

Недостаток блочных шифров: одинаковые блоки открытого текста преобразуются в одинаковые блоки шифротекста. Это понижает стойкость шифра.

Сцепление блоков

Пример 1. Текущий блок открытого текста (T_i) суммируется побитово функцией XOR (по модулю два) с предыдущим блоком шифртекста (C_{i-1}), и к результату применяется алгоритм шифрования:

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

В качестве начального блока C_0 используется произвольный случайный блок (он включается в шифротекст).

Пример 2. Алгоритм шифрования применяется к предыдущему блоку шифртекста, а затем берется функцией XOR с текущим блоком:

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

Эти схемы обеспечивают зависимость всех последующих блоков шифротекста от всех предыдущих блоков открытого текста. Поэтому изменение какого-то блока открытого текста приводит к изменению не только соответствующего блока шифротекста, но и всех последующих блоков шифртекста.

Добавление случайных данных

Является эффективным способом затруднить криптоанализ шифра. *Идея шифрования:*

1. Перед шифрованием текста T генерируется случайный блок данных R заранее определенной длины, и дописывается к тексту.
2. Получившийся блок данных $R|T$, (знак '|' означает конкатенацию (сцепление) двоичных наборов данных), преобразуется в шифротекст с помощью алгоритма шифрования по ключу k: $C = E_k(R|T)$.

Применив к шифротексту алгоритм расшифрования, мы получаем блок данных $R|T$. Так как длина блока R известна, то исходный текст T однозначно восстанавливается.

Достоинство такого метода: при шифровании одного и того же блока данных получаются различные блоки шифротекста. А это сильно затрудняет атаку на шифр.

Недетерминированные шифры

При оценке стойкости шифра предполагается, что алгоритм шифрования известен лицу, пытающемуся взломать шифр. Это основывается на том, что

1. удержать алгоритм в секрете не реально, так как программные средства шифрования распространяются;
2. что бы оценить стойкость шифра, алгоритмы открыто изучаются экспертами.

Очевидно, что знание алгоритма шифрования облегчает криптоанализ. Для того чтобы этого избежать, используются *недетерминированные* (гибкие) шифры.

Идея метода. Существует набор известных алгоритмов шифрования $E_1, E_2, ..., E_n$. По секретному ключу k формируется последовательность чисел $a_1, a_2, ..., a_n$ (например 3; 5; 1; …). К тексту Т применяется процедура шифрования E_i в порядке, определяемом a_i.

$$C = E_{\alpha_i}(...(E_{\alpha_2}(E_{\alpha_1}(T))...)$$

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

Тема 8. Система конфиденциального обмена информацией PGP

Криптография с открытым ключом (асимметричное шифрование)

Создаётся ключевая пара: закрытый и открытый ключи. *Открытый ключ используется для шифрования сообщений, закрытый ключ — для расшифрования.*

Отправить сообщение может кто угодно, зашифровав его открытым ключом получателя. Расшифровать сообщение может, только получатель с помощью своего закрытого ключа.

Открытый ключ выглядит как небольшой текстовый блок, и его можно разместить на своей Web странице или послать электронной почтой своему партнеру. Схема шифрования с открытым ключом (рис.2).

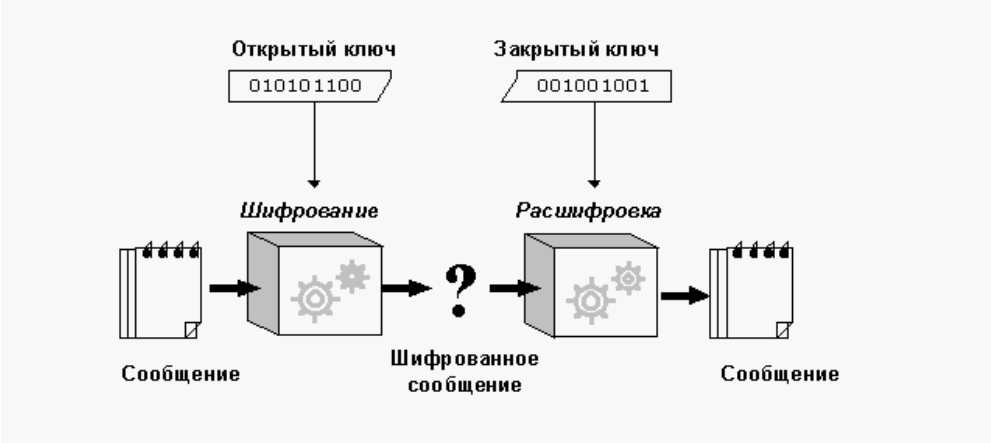

Рис. 2 Схема шифрования с открытым ключом

Мы знаем, что алгоритмы шифрования с двумя ключами работают значительно медленнее, чем алгоритмы симметричного шифрования, поэтому обычно ассиметричные системы используются не самостоятельно, а в комплексе с симметричными системами.

В 1991 г. Филиппом Циммерманном была разработана программа PGP (Pretty Good Privacy “Почти полная приватная безопасность”), которая относится к классу систем с двумя ключами, открытым (публичным) и закрытым (секретным).

PGP — это компьютерная программа, позволяющая выполнять операции шифрования и создания цифровой подписи (ЦП) для сообщений, файлов и другой информации, представленной в электронном виде.

Как шифруются файлы и сообщения

1. Для шифрования сообщения используется качественный алгоритм *симметричного шифрования* (с 1 секретным ключом). Для этого генерируется временный случайный ключ специально для данного "сеанса". Поэтому шифрование происходит быстро.

В PGP для генерации временных сеансовых ключей используется генератор псевдослучайных чисел.

2. Сам сеансовый ключ шифруется по ассиметричному алгоритму с помощью открытого ключа получателя. Так как ключ – это небольшой объём данных, то времени затрачивается не много.

3. Зашифрованный сеансовый ключ включается в зашифрованное сообщение и отправляется получателю.

Сеансовый ключ используется только один раз (см. рис. 3).

Расшифрование. Процесс расшифровки обратен по отношению к шифрованию:

- адресат ищет сеансовый ключ и расшифровывает его с помощью своего закрытого ключа;
- полученный сеансовый ключ используется для расшифровки самого сообщения (см. рис. 4).

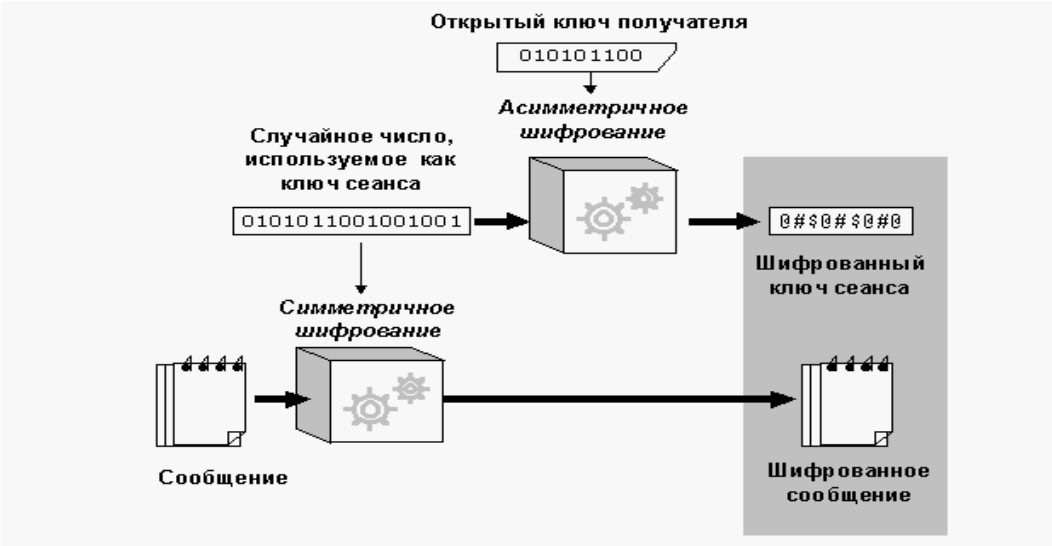

Рис. 3 Процесс шифрования

Ключи. Так как открытый и закрытый ключи взаимосвязаны, то сложно получить закрытый ключ исходя из наличия открытого ключа, но это возможно. Поэтому важно выбрать длину ключа подходящего размера: достаточно большого размера

— для обеспечения безопасности (для длительного хранения), достаточно малого размера — для быстрого режима работы.

Удобства. Пусть одно сообщение должно быть передано многим адресатам. Для этого:

1. сообщение шифруется симметричным алгоритмом с помощью сеансового ключа *один раз*. Затем копируется.
2. сеансовый ключ шифруется ассиметричным алгоритмом с помощью открытых ключей адресатов и рассылается вместе с сообщением.

Архивация при шифровании данных

Доказано, что чем меньше корреляция (взаимосвязь) между блоками входной информации, тем труднее взломать шифр. Архивация (сжатие), устраняя избыточную информацию, как раз и ликвидирует корреляции во входном потоке.

PGP сжимает данные до того, как их шифровать, так как зашифрованные данные уже не сжимаются. Таким образом архивация данных

- снижает избыточность данных, при этом значительно увеличивая устойчивость к криптоанализу;
- экономит дисковое пространство.

На сжатие исходного текста требуется дополнительное время, но с точки зрения безопасности это оправдано.

Симметричные алгоритмы PGP

PGP предоставляет три симметричных блочных шифра: CAST, тройной DES и IDEA. Эти алгоритмы разработаны командами криптографов с выдающейся репутацией.

Дайджест сообщения

Дайджест сообщения (или хэш-образ) — это компактная (160- или 128-битная) "выжимка" сообщения. Дайджест сообщения вычисляется с использованием *односторонней хэш-функции*.

Если сообщение подвергнется какому-либо изменению, ему будет соответствовать другой дайджест. Это позволяет использовать дайджест для контроля целостности данных. Идея:

- при передаче информации сообщение хэшируется;
- хэш-образ передаётся получателю вместе с сообщением;

- получатель вычисляет хэш-образ сообщения повторно, и если оба хэша совпали, то это означает, что информация была передана без изменений.

Строгое существование односторонних функций не доказано, поэтому все используемые хэш-функции являются кандидатами в односторонние.

В PGP для получения дайджеста сообщения используется алгоритм SHA (Алгоритм защищенного хеширования).

Электронная цифровая подпись

Основные понятия

Рассматриваем электронный документооборот.

Клиент может общаться с банком, отдавая ему распоряжения о перечислении своих средств на счета других лиц. Ему не надо ездить в банк, всё можно сделать не отходя от компьютера. Но здесь возникает проблема: как банк узнает, что распоряжение поступило именно от данного лица, а не от злоумышленника, выдающего себя за него? Эта проблема решается с помощью так называемой *электоронной цифровой подписи (ЭЦП).*

Если абонент А хочет создать себе ЭЦП, то с помощью специальной программы он должен создать пару ключей: закрытый и открытый. Открытый ключ передаётся банку, закрытый хранится у себя. Абонент А отправляет банку *электронный документ = текстовый документ + ЭЦП.*

Текстовый документ шифруется *публичным* ключом банка, а расшифровывается его закрытым ключом. ЭЦП кодируется *закрытым* ключом абонента А, а читается банком открытым ключом отправителя.

Если подпись читаема, банк может быть уверен, что поручение ему отправил именно абонент А и никто другой.

На практике ЭЦП — это зашифрованный дайджест сообщения.

Электронная подпись позволяет:

1. Контролировать целостность документа.

При любом изменении документа подпись станет недействительной, так как вычисляется на основе исходного состояния документа.

2. Удостовериться в личности отправителя сообщения.

Создать подпись можно только зная закрытый ключ, а он известен только владельцу. Поэтому он не сможет отказаться от своей подписи под документом.

Таким образом решаются вопросы *о невозможности отказа от авторства,* а также о *невозможность приписать себе авторство*.

Схема создания дайджеста сообщения и электронной подписи

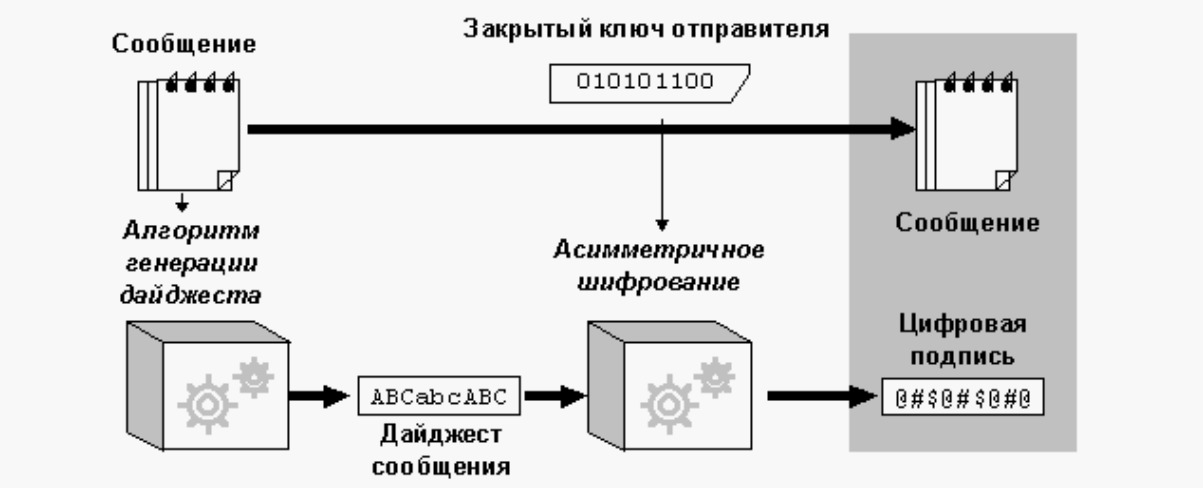

Рис 5. Схема создания дайджеста сообщения и электронной подписи

Схема создания ЭП охватывает 3 процесса:

1. генерация пары ключей (закрытый и соответствующий ему открытый, несущий идентификатор пользователя);
2. создание дайджеста сообщения с помощью односторонней хэш-функции;
3. шифрование дайджеста сообщения *закрытым ключом (*формирование ЭЦП).

Схема проверки подлинности документа и авторства

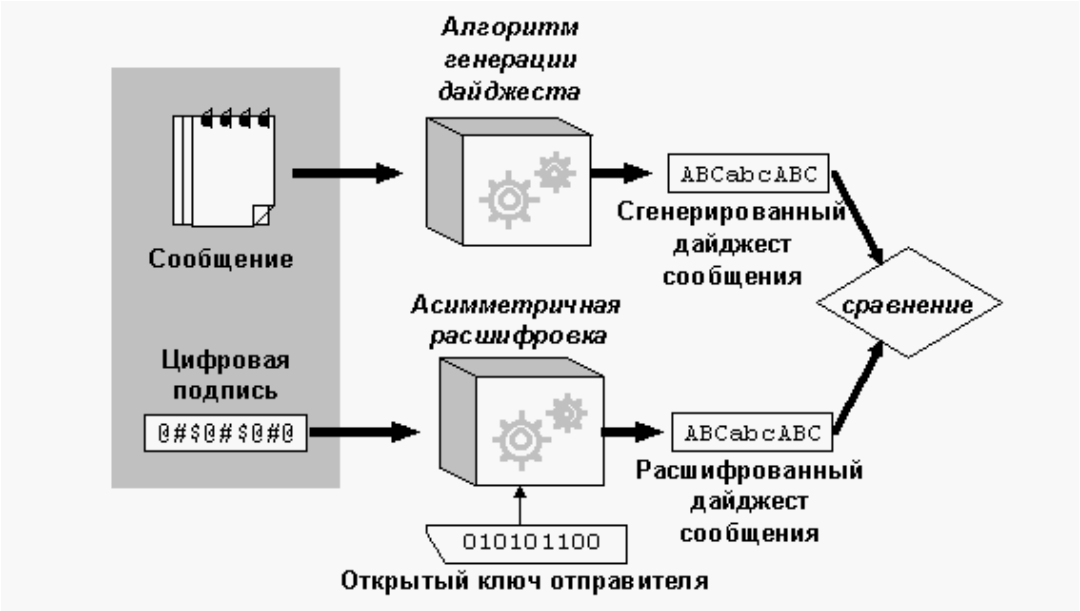

Рис. 6 Схема проверки подлинности документа

Проверка ЭЦП:

1. генерация дайджеста исходного сообщения;
2. расшифрование ЭЦП открытым ключом, получение дайджеста сообщения;
3. если значения сгенерированного и расшифрованного дайджестов совпали, то это значит, что
   - сообщение не было изменено другим человеком;
   - отправитель является создателем сообщения.

Вывод. В ассиметричных алгоритмах:

1. шифрование сообщений осуществляется с помощью открытого ключа, а расшифрование с помощью закрытого ключа получателя;
2. ЭЦП — реквизит электронного документа, создаётся *закрытым ключом автора сообщения. Проверка ЭЦП осуществляется его открытым ключом;* текстовый документ без ЭЦП — это не более как обычный текст, который не имеет юридической силы.
3. ЭЦП выполняет ту же функцию, что и ручная подпись. Но ручную подпись легко подделать. ЦП подделать почти невозможно.

Понятие об электронных сертификатах

В ассиметричных криптосистемах чем шире распространяются открытые ключи, тем лучше. Но важно защитить их от подделки. (Злоумышленник может сгенерировать новую пару ключей: закрытый и открытый, несущий идентификатор абонанта.)

Цель: исключить возможность подделки открытых ключей.

Пути решения этой проблемы:

- получение открытого ключа непосредственно от абонента. Это может быть затруднительным;
- использование сертификатов. Сертификат позволяет удостоверить данные о владельце и его открытом ключе подписью доверенного лица. На практике подписанный сертификат выдаётся Центрами сертификации ключей, которые поддерживаются доверенными организациями.

Сертификат обеспечивает надёжную связь между открытым ключом и абонентом, которому он принадлежит.

Задача защиты открытых ключей от подделки — единственная серьезная проблема практического приложения криптографии с открытыми ключами.

Уязвимые места

Защита закрытого ключа

Никакая система защиты данных не является неуязвимой. Украденный закрытый ключ может быть использован:

1. для расшифровки всех сообщений, адресованных вам;
2. для создания фальшивых сертификатов открытых ключей.

Защита закрытого ключа: держать его на домашнем персональном компьютере или на съёмном носителе и забирать его с собой.

Не до конца удаленные файлы

Проблема безопасности связана со способом, которым большинство операционных систем удаляет файлы.

1. Если вы зашифровали файл и затем удаляете его с исходным открытым текстом, операционная система не стирает данные физически. Она просто помечает соответствующие блоки на диске, как свободные, допуская тем самым повторное использование этого пространства.

Если злоумышленник прочитает эти блоки данных вскоре после того, как они помечены как свободные, он сможет восстановить ваш исходный открытый текст.

Это может произойти и случайно, если запустят программу восстановления, а она восстановит ранее стертые файлы.

2. При создании исходного секретного сообщения с использованием текстового редактора, программа может оставить промежуточные временные файлы, просто потому, что она так работает. Эти временные файлы обычно удаляются редактором при его закрытии, но фрагменты текста остаются где-то на диске.

Единственный способ предотвратить восстановление открытого текста — это обеспечить перезапись места, занимаемого удаленными файлами. Это можно осуществить с помощью любой утилиты, которая способна перезаписать все неиспользованные блоки на диске.

Вирусы и закладки

Другая атака может быть предпринята с помощью специально разработанного компьютерного вируса, который инфицирует PGP или операционную систему. Такой гипотетический вирус может перехватывать

пароль, закрытый ключ или расшифрованное сообщение, а затем тайно сохранять их в файле или передавать по сети своему создателю.

Вирус также может модифицировать PGP таким образом, чтобы она перестала надлежащим образом проверять подписи. Такая атака обойдется дешевле, чем криптоаналитическая.

Существуют антивирусные программы, которые снижают риск заражения вирусами.

Радиоатака

Хорошо оснащенным противником может быть предпринята атака, которая осуществляет перехват электромагнитного излучения, испускаемого вашим компьютером. Эта дорогая и трудоемкая атака, также является более дешевой, чем криптоанализ.

Соответствующим образом оборудованный фургон может припарковаться рядом с вашим офисом и издалека перехватывать нажатия клавиш и сообщения, отображаемые на мониторе.

Такую атаку можно предотвратить экранированием всего компьютерного оборудования и сетевых кабелей с тем, чтобы они не испускали излучения. Технология такого экранирования известна под названием "Tempest" и используется службами, выполняющими оборонные заказы.

Тема 9. Сокрытие передачи и хранения информации. Стеганография

Из истории древнего мира, V век до н.э. Голову раба брили, на коже головы писали сообщение и после отрастания волос раба отправляли к адресату. Так Геродот описывает один из первых случаев применения *стеганографии — искусства скрытого письма.*

Искусство развивалось, превратившись в науку, помогавшую людям на протяжении многих веков скрывать от посторонних глаз сам факт передачи информации. Еще древние римляне писали между строк невидимыми чернилами, в качестве которых использовались фруктовые соки, молоко. При нагревании невидимый текст проявлялся.

Известен метод "микроточка". Сообщение записывается с помощью современной техники на очень маленький носитель (микроточку), который посылается с обычным письмом в заранее оговоренном месте (например под маркой). При увеличении микроточка даёт чёткое изображение печатной страницы.

В настоящее время существует много методов «спрятать» информацию больших объёмов, включая чертежи. Обнаружить «микроточки» очень трудно.

Стеганографические программные продукты

Компьютерная стеганография базируется на двух принципах.

1. Файлы, содержащие оцифрованные изображение или звук, могут быть незначительно видоизменены без потери функциональности, в отличие от других типов данных, требующих абсолютной точности.
2. Неспособность органов чувств человека различать незначительные изменения в цвете изображения или качестве звука. Изменение значений наименее важных битов, отвечающих за цвет пиксела, не приводит к сколь-нибудь заметному для человека изменению цвета.

Стеганографические системы — это системы сокрытия информации.

Один из лучших продуктов в этой области для платформы Windows9x/NT - это S-Tools. Программа позволяет прятать любые файлы как в изображениях формата gif и bmp, так и в аудио файлах формата wav.

При этом S-Tools — это стеганография и криптография "в одном флаконе", потому что файл, подлежащий сокрытию, еще и шифруется с помощью одного из криптографических алгоритмов с симметричным ключом:

- в окно программы перетаскивается Файл-носитель (изображение или музыка), затем в этот файл перетаскивается файл с данными любого формата;
- вводится пароль;
- выбирается алгоритм шифрования.

Внешне графический файл остается практически неизменным, меняются лишь кое-где оттенки цвета. Звуковой файл также не претерпевает заметных изменений.

Для безопасности следует использовать неизвестные изображения. Например фотография вашего песика вполне подойдет.

Соотношения размеров файла-носителя и текстового файла, который нужно спрятать может быть различным. Иногда размер текстового файла даже превышает размер графического. Но, даже если подозрения у кого-то и возникнут, то не зная пароля, сам факт использования S-Tools доказать нельзя.

Другая распространенная стеганографическая программа — Steganos for Windows 95. Она обладает теми же возможностями, что и S-Tools, кроме того, способна прятать данные не только в файлах формата bmp и wav, но и в обычных текстовых и HTML файлах, причем весьма оригинальным способом: в конце каждой строки добавляется определенное число пробелов.

Везете вы на дискете "Анну Каркнину" Л. Толстого, а в ней — чертежи секретной макаронной фабрики…

Цифровые водяные знаки

Если рассматривать коммерческие применения стеганографии, то одним из перспективных направлений является digital watermarking, т.е. создание невидимых глазу водяных знаков для защиты авторских прав на графические и аудио файлы.

Помещенные в файл цифровые водяные знаки несут много полезной информации: когда создан файл, кто владеет авторскими правами, как вступить в контакт с автором.

Сегодня на рынке существует много фирм, предлагающих продукты для создания водяных знаков. Один из лидеров — фирма Digimarc.

Но цифровые водяные знаки оказались нестойкими. Они могут перенести многое — изменение яркости, использование спецэффектов, печать и последующее сканирование, но они не могут перенести воздействие специальных программ-стирателей, таких как UnZign и StirMarK. Водяные знаки всех производителей уничтожаются без заметного ухудшения качества изображения.

Компьютерные вирусы и их классификация. Разрушительные действия вирусов

Понятие вируса

Компьютерный вирус — это программа. Но она отличается от привычных программ тем, что, во-первых, запускается без ведома пользователя, а во-вторых, после своего запуска начинает самовоспроизводиться, то есть создавать вредоносные копии и внедрять их в файлы, системные области дисков, вычислительные сети.

Компьютерный вирус представляет собой программный код, который обладает возможностями несанкционированного запуска и самовоспроизведения.

Важно обнаружить вирус до того, как он успеет проявить себя. На этом принципе сроится работа всех антивирусных программ.

Классификация вирусов

Объекты, в которые внедряются вирусы, называются средой обитания вирусов. В зависимости от среды обитания различают следующие типы вирусов:

- программные вирусы;
- загрузочные вирусы;
- макровирусы;
- сетевые вирусы.

Программные вирусы. — внедряются в исполняемые файлы (exe, com, ,bat), в системные файлы (sys), файлы библиотек. После внедрения программные вирусы начинают размножаться при каждом *запуске* файла.

Программные вирусы поступают на компьютер при запуске непроверенных программ. В связи с этим все данные, принятые из Интернета, должны обязательно проходить проверку на безопасность, а если получены данные из незнакомого источника, их следует уничтожать, не рассматривая.

Загрузочные вирусы. Они поражают не программные файлы, а определённые системные области магнитных носителей (жёстких дисков).

Обычно заражение происходит при попытке загрузки компьютера с магнитного носителя, системная область которого содержит загрузочный вирус.

Макровирусы. Внедряются в документы Word и другие файлы, выполненные в программах, имеющие свой язык макрокоманд (например, Excel). Заражение происходит при открытии файла документа.

Сетевые вирусы. Распространяются по компьютерной сети. Их особенность состоит в том, что они заражают только оперативную память компьютера и не записываются на носители информации.

Возможна классификация вирусов не только по среде их обитания, но и по другим характеристикам, например:

- по способу заражения;
- по разрушительным возможностям;
- по алгоритму работы.

По способу заражения вирусы делятся на резидентный и нерезидентный. *Резидентные вирусы* попадают в оперативную память компьютера и могут постоянно проявлять свою активность вплоть до выключения компьютера.
Нерезидентные вирусы, напротив, в память не внедряются и активны ограниченное время.

По разрушительным возможностям вирусы можно разделить на:

- безвредные. Их влияние ограничивается уменьшением свободной памяти на диске и графическими, звуковыми и пр. эффектами;
- опасные вирусы, которые могут привести к серьезным сбоям в работе компьютера;
- очень опасные, которые могут привести к потере программ, уничтожить данные, стереть информацию, записанную в системных областях памяти.

По особенностям алгоритма. Алгоритмы работы новых вирусов намного изощрённее алгоритмов десятилетней давности**.** К вирусам со сложным алгоритмом работы относятся:

- *полиморфные вирусы*. Эти вирусы трудно обнаружить, поскольку они имеют зашифрованный программный код. Расшифровка кода производится в процессе его выполнения.
- стелс-вирусы (вирусы-невидимки). Это вирусы, которые не видимы при просмотре файлов средствами операционной системы. При открытии

поражённого файла они немедленно удаляют из него свой программный код, а при закрытии файла восстанавливают свой код на прежнее место.

К компьютерным вирусам примыкают и так называемые *троянские кони (троянские программы). Троянский конь,* а так же *вирусы-черви* представляют собой программы–шпионы, основная цель которых сбор и передача информации получателю.

Троянский конь узнаёт из сведений о компьютере

- адрес электронной почты пользователя;
- считывает адреса e-mail из его адресной книги;
- саморазмножается по данным адресам;
- похищает сведения о паролях удалённых пользователей

и возвращает эту информацию по ЭП пользователю, активизирующему его. С точки зрения информационной безопасности такие программы относятся к особо опасным.

Средства антивирусной защиты

1. Основным средством является резервное копирование наиболее ценных данных. Резервные копии должны храниться отдельно от компьютера (на внешних носителях в сейфе, в Web-папках на удалённых сервера Интернете.)

2. Создание образа жёсткого диска на внешних носителях. В случае выхода из строя данных сохранённый «образ диска» может позволить восстановить если не все данные, то их большую часть.

3. Использование программных средств антивирусной защиты. Регулярное сканирование жёстких дисков в поисках компьютерных вирусов. При сканировании антивирусная программа ищет вирус путём сравнения кода программ с кодами известных ей вирусов, хранящимися в базе данных.

Если база данных устарела, а вирус является новым, сканирующая программа его не обнаружит. Антивирусную программу желательно обновлять один раз в 2 недели, в крайнем случае — один раз в 3 месяца.

4. Не запускать не проверенные файлы.