МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

СЕВЕРО-КАВКАЗСКИЙ ФИЛИАЛ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
СВЯЗИ И ИНФОРМАТИКИ»

С.А. ШВИДЧЕНКО

Методические указания для проведения практических занятий по дисциплине

Б1.В.05 «Разработка безопасного программного обеспечения»

Кафедра **«Информатика и вычислительная техника»**

Направление подготовки **10.03.01 Информационная безопасность**

Профиль **Безопасность компьютерных систем**

Разработала:
Доцент кафедры ИВТ Швидченко С.А.

Ростов-на-Дону
2022

Методические указания для
проведения практических занятий
по дисциплине
«Разработка безопасного программного обеспечения»

Составитель: Швидченко С.А., доц. каф. «ИВТ»

Рассмотрено и одобрено на
заседании кафедры «ИВТ»
Протокол от «26» августа 2022 г.,
№ 1.

Практическое занятие №1. АЛГОРИТМЫ ШИФРОВАНИЯ ДАННЫХ

Алгоритмы шифрования данных

Цель работы: освоить методику работы ассиметричных алгоритмов шифрования, где существует два ключа – один для шифрования, другой для дешифрования.

Теоретические сведения

Алгоритм RSA разработан в 1977 г. Роном Ривестом, Ади Шамиром и Леном Адлеманом и опубликован в 1978 г. С тех пор алгоритм Rivest-Shamir-Adleman (RSA) широко применяется практически во всех приложениях, использующих криптографию с открытым ключом.

Алгоритм RSA:

1. Вычисление ключей

Важным моментом в этом криптоалгоритме является создание пары ключей: открытого и закрытого. Для алгоритма RSA этап создания ключей состоит из следующих операций:

1.1. Выбираются два простых различных числа p и q. Вычисляется их произведение $n = p \cdot q$, называемое модулем. Под простым числом будем понимать такое число, которое делится только на 1 и на само себя. Взаимно простыми числами будем называть такие числа, которые не имеют ни одного общего делителя, кроме единицы.

1.2. Вычисляется функция Эйлера $\Phi(n) = (p - 1) \cdot (q - 1)$.

1.3. Выбирается произвольное число e ($e < n$), такое, что $1 < e < \Phi(n)$ и не имеет общих делителей, кроме 1 (взаимно простое) с числом $(p - 1) \cdot (q - 1)$.

1.4. Вычисляется d методом Евклида таким образом, что $(e \cdot d - 1)$ делится на $(p - 1) \cdot (q - 1)$.

1.5. Два числа (e, n) публикуются как открытый ключ.

1.6. Число d хранится в секрете – закрытый ключ есть пара (d, n), который позволит читать все послания, зашифрованные с помощью пары чисел (e, n).

2. Шифрование

Шифрование с помощью пары чисел производится следующим образом:

2.1. Отправитель разбивает своё сообщение M на блоки m_i. Значение $m_i < n$, поэтому длина блока m_i в битах не больше $k = [\log_2(n)]$ бит, где квадратные скобки обозначают, взятие целой части от дробного числа.

Например, если $n = 21$, то максимальная длина блока $k = [\log_2(21)] = [4.39\ldots] = = 4$ бита.

2.2. Подобный блок может быть интерпретирован как число из диапазона $(0; 2^k - 1)$. Для каждого такого числа m_i вычисляется выражение (c_i – зашифрованное сообщение): $c_i = ((m_i)^e) \bmod n$.

Необходимо добавлять нулевые биты слева в двоичное представление блока c_i до размера $k = [\log_2(n)]$ бит.

3. Дешифрование

Чтобы получить открытый текст, необходимо каждый блок дешифровать отдельно: $m_i = ((c_i)^d) \bmod n$.

Пример:

Выбрать два простых числа: $p = 7$, $q = 17$.

Вычислить $n = p \cdot q = 7 \cdot 17 = 119$.

Вычислить $\Phi(n) = (p - 1) \cdot (q - 1) = 96$.

Выбрать e так, чтобы e было взаимнопростым с $\Phi(n) = 96$ и меньше, чем $\Phi(n)$: $e = 5$.

Определить d так, чтобы $d \cdot e \equiv 1 \bmod 96$ и $d < 96$, $d = 77$, так как $77 \cdot 5 = 385 = 4 \cdot 96 + 1$.

Результирующие ключи открытый {5, 119} и закрытый ключ {77, 119}.

Например, требуется зашифровать сообщение $M = 19$: $19^5 = 66 \pmod{119}$, $C = = 66$. Для дешифрования вычисляется $66^{77} \pmod{119} = 19$.

Варианты заданий

1. Разработать консольное приложение для шифрования/дешифрования произвольных файлов с помощью алгоритма RSA.
2. Разработать визуальное приложение для шифрования/дешифрования изображений.
3. Разработать визуальное приложение для шифрования/дешифрования произвольных файлов.
4. Разработать клиент-серверное приложение для защищённой передачи файлов по сети.
5. Разработать клиент-серверное приложение для защищённого обмена сообщениями по сети.
6. Разработать визуальное приложение для шифрования/дешифрования чисел.
7. Разработать консольное приложение для генерации ключей.
8. Реализовать программу для шифрования / дешифрования текстов, работающую по алгоритму RSA. Программа должна уметь работать с текстом произвольной длины.

Контрольные вопросы к практическим занятиям ПЗ1(ПК-2):

1. Понятие ассиметричных алгоритмов шифрования
2. Поясните суть работы Алгоритма RSA
3. В чем особенности шифрования/дешифрованияизображений?
4. В чем особенности шифрования/дешифрования произвольных файлов?
5. В чем особенности защищённой передачифайлов по сети?
6. В чем особенности защищённой защищённого обменасообщениями по сети?
7. В чем особенности шифрования/дешифрования чисел?
8. В чем особенности генерации ключей?
9. В чем особенности работы с текстом произвольной длины?

***Цель работы*:** ознакомиться с возможностями «привязки» к характеристикам компьютера.

Теоретические сведения

В качестве анализируемых характеристик компьютера могут использоваться:

1. Информация об используемой операционной системе
2. Имя пользователя;
3. Имя компьютера;
4. Наличие звуковой карты;
5. Наличие подключенных принтера, сканера и т.д;
6. Дата создания BIOS;
7. Серийный номер диска;
8. Характеристики процессора.

Для получения подобных характеристик в операционной системе Windows используются API-функции и информация из реестра.

API-функции

API сокращенно Application Programming Interface (интерфейс прикладного программирования). API – набор функций, которые операционная система предоставляет программисту. API обеспечивает относительно простой путь для программистов для использования полных функциональных возможностей аппаратных средств или операционной системы.

32-разрядные версии Windows обычно используют один и тот же набор функций API, хотя имеются некоторые различия между платформами.

Почти все функции, которые составляют Windows API, находятся внутри DLL (Dynamic Link Library). Эти dll-файлы находятся в системной папке Windows. Существует свыше 1000 функций API, которые условно делятся на четыре основные категории:

- работа с приложениями – запуск и закрытие приложений, обработка команд меню, перемещения и изменения размера окон;
- графика – создание изображений;
- системная информация – определение текущего диска, объема памяти, имя текущего пользователя и т.д.
- работа с реестром – манипуляции с реестром Windows.

Реестр Windows

Реестр – база данных операционной системы, содержащая конфигурационные сведения. По замыслу Microsoft реестр должен был полностью заме-

нить файлы ini, которые были оставлены только для совместимости со старыми программами, ориентированными на более ранние версии операционной системы.

Переход от ini файлов к реестру произошел по той причине, что на эти файлы накладывается ряд серьезных ограничений, и главное из них состоит в том, что предельный размер такого файла составляет 64Кб.

Предупреждение: никогда не удаляйте или не меняйте информацию в реестре, если Вы не уверены что это именно то, что нужно. В противном случае некорректное изменение данных может привести к сбоям в работе Windows и, в лучшем случае, информацию придется восстанавливать из резервной копии.

Реестр имеет следующую структуру:

1) HKEY_CLASSES_ROOT. В этом разделе содержится информация о зарегистрированных в Windows типах файлов, что позволяет открывать их по двойному щелчку мыши, а также информация для OLE и операций drag-and-drop;

2) HKEY_CURRENT_USER. Здесь содержатся настройки оболочки поль- зователя (например, Рабочего стола, меню "Пуск", ...), вошедшего в Windows. Они дублируют содержимое подраздела HKEY_USER\name, где name – имя пользователя, вошедшего в Windows. Если на компьютере работает один пользователь и используется обычный вход в Windows, то значения раздела берутся из подраздела HKEY_USERS\.DEFAULT;

3) HKEY_LOCAL_MACHINE. Этот раздел содержит информацию, отно- сящуюся к компьютеру: драйверы, установленное программное обеспечение иего настройки;

4) HKEY_USERS. Содержит настройки оболочки Windows для всех пользователей. Как было сказано выше, именно из этого раздела информация копируется в раздел HKEY_CURRENT_USER. Все изменения в HKCU (сокращенное название раздела HKEY_CURRENT_USER) автоматически переносятся в HKU;

5) HKEY_CURRENT_CONFIG. В этом разделе содержится информация о конфигурации устройств Plug&Play и сведения о конфигурации компьютера с переменным составом аппаратных средств;

6) HKEY_DYN_DATA. Здесь хранятся динамические данные о состоянии различных устройств, установленных на компьютере пользователя. Именно сведения этой ветви отображаются в окне "Свойства: Система" на вкладке "Устройства", вызываемого из Панели управления. Данные этого раздела изменяются самой операционной системой, так что редактировать что-либо вручную не рекомендуется.

Примеры процедур и функций, определяющих параметры компьютера

Определение версии операционной системы

```
BOOL DisplaySystemVersion()
{
```

```
OSVERSIONINFOEX osvi;
BOOL bOsVersionInfoEx;
ZeroMemory(&osvi, sizeof(OSVERSIONINFOEX));
osvi.dwOSVersionInfoSize = sizeof(OSVERSIONINFOEX);
if( !(bOsVersionInfoEx = GetVersionEx ((OSVERSIONINFO *) &osvi)) )
{
  osvi.dwOSVersionInfoSize = sizeof (OSVERSIONINFO);
  if (! GetVersionEx ( (OSVERSIONINFO *) &osvi) )
    return FALSE;
}

switch (osvi.dwPlatformId)
{
  case VER_PLATFORM_WIN32_NT:
    if ( osvi.dwMajorVersion <= 4 )
      printf("Microsoft Windows NT ");
    if ( osvi.dwMajorVersion == 5 && osvi.dwMinorVersion == 0 )
      printf ("Microsoft Windows 2000 ");
    if( bOsVersionInfoEx )
    {
      if ( osvi.wProductType == VER_NT_WORKSTATION )
      {
        if ( osvi.dwMajorVersion == 5 && osvi.dwMinorVersion == 1 )

          printf ("Microsoft Windows XP ");

        if( osvi.wSuiteMask & VER_SUITE_PERSONAL )
          printf ( "Home Edition " );
        else
          printf ( "Professional " );
      }
      else if ( osvi.wProductType == VER_NT_SERVER )
      {
        if ( osvi.dwMajorVersion == 5 && osvi.dwMinorVersion == 2 )
          printf ("Microsoft Windows .NET ");

        if( osvi.wSuiteMask & VER_SUITE_DATACENTER )
          printf ( "DataCenter Server " );
        else if( osvi.wSuiteMask & VER_SUITE_ENTERPRISE )
          if( osvi.dwMajorVersion == 4 )
            printf ("Advanced Server " );
          else
            printf ( "Enterprise Server " );
        else if ( osvi.wSuiteMask == VER_SUITE_BLADE )
          printf ( "Web Server " );
        else
          printf ( "Server " );
```

```
      }
    }
    else
    {
      HKEY hKey;
      char szProductType[BUFSIZE];
      DWORD dwBufLen=BUFSIZE;
      LONG lRet;
      lRet = RegOpenKeyEx( HKEY_LOCAL_MACHINE,
        "SYSTEM\\CurrentControlSet\\Control\\ProductOptions",
        0, KEY_QUERY_VALUE, &hKey );
      if( lRet != ERROR_SUCCESS )
         return FALSE;
      lRet = RegQueryValueEx( hKey, "ProductType", NULL, NULL,
        (LPBYTE) szProductType, &dwBufLen);
      if( (lRet != ERROR_SUCCESS) || (dwBufLen > BUFSIZE) )
         return FALSE;
      RegCloseKey( hKey );
      if ( lstrcmpi( "WINNT", szProductType) == 0 )
        printf( "Professional " );
      if ( lstrcmpi( "LANMANNT", szProductType) == 0 )
        printf( "Server " );
      if ( lstrcmpi( "SERVERNT", szProductType) == 0 )
        printf( "Advanced Server " );

    }
    if ( osvi.dwMajorVersion <= 4 )
    {
      printf ("version %d.%d %s (Build %d)\n",
        osvi.dwMajorVersion,
        osvi.dwMinorVersion,
        osvi.szCSDVersion,
        osvi.dwBuildNumber & 0xFFFF);
    }
    else
    {
      printf ("%s (Build %d)\n",
        osvi.szCSDVersion,
        osvi.dwBuildNumber & 0xFFFF);
    }
    break;
  case VER_PLATFORM_WIN32_WINDOWS:
    if (osvi.dwMajorVersion == 4 && osvi.dwMinorVersion == 0)
    {
       printf ("Microsoft Windows 95 ");
       if ( osvi.szCSDVersion[1] == 'C' || osvi.szCSDVersion[1] == 'B' )
          printf("OSR2 " );
```

```
        }
        if (osvi.dwMajorVersion == 4 && osvi.dwMinorVersion == 10)
        {
           printf ("Microsoft Windows 98 ");
           if ( osvi.szCSDVersion[1] == 'A' )
              printf("SE " );
        }
        if (osvi.dwMajorVersion == 4 && osvi.dwMinorVersion == 90)
        {
           printf ("Microsoft Windows Millennium Edition ");
        }
        break;
     }
     return TRUE;
  }
```

Определение серийного номера раздела диска

```
  TCHAR    szVolName[256];
  DWORD    dwNum;
  DWORD    dwMaxComSize;
  DWORD    dwFlags;
  TCHAR    szFS[256];
  BOOL     bRes;
  bRes = GetVolumeInformation ( "c:\\", szVolName, sizeof(szVolName),
&dwNum, &dwMaxComSize, &dwFlags, szFS, sizeof(szFS));
```

Определение имени компьютера

```
  const int WSVer = 0x101;
  WSADATA wsaData;
  char Buf[128];
  if (WSAStartup(WSVer, &wsaData) == 0)
  {
    gethostname(&Buf[0], 128);
    MessageBox(0, Buf,0,0);
    WSACleanup;
  }
```

Определение имени пользователя

```
  char         buffer[UNLEN+1];
  DWORD                    size;
  size=sizeof(buffer);
  GetUserName(buffer,&size);
```

Определение версии BIOS

```
  LPSTR GetSystemBiosVersion()
  {
    HKEY hKey;
```

```
        LONG Res1, Res2;
        DWORD cData=255;
        TCHAR SystemBiosVersion[255]={'\0'};

Res1=RegOpenKeyEx(HKEY_LOCAL_MACHINE,"HARDWARE\\DESCRIPTI
ON\\System",NULL, KEY_QUERY_VALUE, &hKey);
        if(Res1==ERROR_SUCCESS)
          {

        Res2=RegQueryValueEx(hKey,"SystemBiosVersion",NULL,NULL,…
           (LPBYTE)SystemBiosVersion,&cData);
           if(Res2==ERROR_SUCCESS)
             {
              for (const char* p = SystemBiosVersion; *p; p += strlen(p)+1)
               {
                printf("%s\n", p);
               }

             return SystemBiosVersion;
             }
             else
               {
                  MessageBox(NULL,"RegQueryValueEx:
      SystemBiosVesion","ERROR",MB_OK);
                  return NULL;
               }
          }
        else
          {

           MessageBox(NULL,"RegOpenKeyEx:
SystemBiosVersion","ERROR",MB_OK);
           return NULL;
          }
        RegCloseKey(hKey);
      }
```

Определение частоты процессора (способ №1)

```
      double CPUSpeed(void)
        {
          DWORD dwTimerHi, dwTimerLo;
          asm
           {
             DW 0x310F
             mov dwTimerLo, EAX
             mov dwTimerHi, EDX
           }
```

```
  Sleep (500);
  asm
   {
     DW 0x310F
     sub EAX, dwTimerLo
     sub EDX, dwTimerHi
     mov dwTimerLo, EAX
     mov dwTimerHi, EDX
   }
  return dwTimerLo/(1000.0*500);
 }
```

Задание на лабораторную работу

Разработать программу, реализующую привязку к компьютеру, используя совокупность характеристик согласно варианту задания. Добиться того, чтобы программа не запускалась на другом компьютере.

Таблица 1. Варианты заданий

№ варианта	Характеристики
1	Серийный номер раздела жесткого диска, MAC-адрес сетевой карты
2	Информация из реестра, тактовая частота процессора
3	Версия операционной системы, MAC-адрес сетевой карты
4	Имя пользователя, серийный номер раздела жесткого диска
5	Название компьютера, информация из реестра
6	Версия БИОС, имя пользователя
7	Серийный номер раздела жесткого диска, имя пользователя
8	Имя пользователя, тактовая частота процессора
9	MAC-адрес сетевой карты, тактовая частота процессора

Контрольные вопросы ПЗ2(ПК-2):

1. Что может использоваться в качестве анализируемых характеристик компьютера?
2. Что такое API-функции?
3. Что такое Реестр Windows?
4. Приведите примеры процедур и функций, определяющих параметры компьютера
5. Как определить версию операционной системы?
6. Как определить имя компьютера?
7. Как определить имя пользователя?
8. Как определить частоту процессора?

Практическое занятие №3. ЗАЩИТА ОТ ПОБОЧНОГО ЭЛЕКТРОМАГНИТНОГО ИЗЛУЧЕНИЯ И НАВОДОК ВИДЫ ШИФРОВ.

Побочные излучения — это радиоизлучения, возникающие в результате любых нелинейных процессов в радиоэлектронном устройстве, кроме процессов модуляции. Побочные излучения возникают как на основной частоте, так и на гармониках, а также в виде их взаимодействия. Радиоизлучение на гармонике — это излучение на частоте (частотах), в целое число раз большей частоты основного излучения. Радиоизлучение на субгармониках — это излучение на частотах, в целое число раз меньших частоты основного излучения. Комбинационное излучение — это излучение, возникающее в результате взаимодействия на линейных элементах радиоэлектронных устройств колебаний несущей (основной) частоты и их гармонических составляющих.

Как известно, любая передача электрического сигнала сопровождается электромагнитным излучением. Если электромагнитный сигнал сам не используется как носитель информации (радиоволны), то подобное излучение оказывается крайне нежелательным с точки зрения безопасности. «В русскоязычной специализированной литературе используется определение «Побочные электромагнитные излучения и наводки» (ПЭМИН). За рубежом пользуются аббревиатурой TEMPEST (сокращение от Transient Electromagnetic Pulse Emanation Standard) или понятием «компрометирующие излучения» (compromising emanations)» . Во всех случаях речь идет исключительно о таком явлении, как переходные электромагнитные импульсные излучения работающей радиоэлектронной аппаратуры.

По правде говоря, для пытливого исследователя изучение проблемы побочных излучений может превратиться в потрясающий детектив. Одна история чего стоит! На саму проблему ПЭМИН впервые обратили внимание еще в 20-х годах прошлого века, в ходе разработки армейских средств телефонной и радиосвязи. Полномасштабные (но закрытые) исследования побочных «компрометирующих» электромагнитных излучений начались только в конце 40-х - начале 50-х годов. Причем это тот самый случай, когда практические изыскания даже опережали теоретическую часть проблемы. Вот только несколько наиболее известных исторических примеров.

С конца 80-х годов охотники за чужими секретами часто перехватывают изображение прямиком с компьютерных мониторов при помощи весьма незамысловатого устройства - обычного бытового телевизора, в котором синхронизаторы заменены генераторами, перестраиваемыми вручную.

Осознание опасности побочных электромагнитных излучений привело к тому, что в наши дни правительственные службы используют дорогое металлическое экранирование отдельных устройств, помещений, а иногда и отдельных зданий. Однако даже для внутренних экранированных помещений существует принцип разделения оборудования на так называемое «красное» и

«черное». «Красное» оборудование, используемое для обработки конфиденциальной информации (например, мониторы), должно быть изолировано фильтрами и экранами от «черного» (например, радиомодемов), которое передает данные без грифа «секретно».

«Оценочно, по каналу ПЭМИН (побочных электромагнитных излучений и наводок) может быть перехвачено не более 1-2 процентов данных, хранимых и обрабатываемых на персональных компьютерах и других технических средствах передачи информации (ТСПИ)» . На первый взгляд может показаться, что этот канал действительно менее опасен, чем, например, акустический, по которому может произойти утечка до 100% речевой информации, циркулирующей в помещении. Однако, нельзя забывать, что в настоящее время практически вся информация, содержащая государственную тайну или коммерческие, технологические секреты, проходит этап обработки на персональных компьютерах. Специфика канала ПЭМИН такова, что те самые два процента информации, уязвимые для технических средств перехвата - это данные, вводимые с клавиатуры компьютера или отображаемые на дисплее, то есть, парадоксально, но весьма значительная часть сведений, подлежащих защите, может оказаться доступна для чужих глаз.

Традиционно считается, что перехват ПЭМИН и выделение полезной информации - весьма трудоемкая и дорогостоящая задача, требующая применения сложной специальной техники. Методики контроля эффективности защиты объектов информатизации созданы в расчете на использование противником так называемых оптимальных приемников. Во времена, когда эти документы разрабатывались, приемные устройства, приближающиеся по своим характеристикам к оптимальным, были громоздкими, весили несколько тонн, охлаждались жидким азотом... Ясно, что позволить себе подобные средства могли лишь технические разведки высокоразвитых государств. Они же и рассматривались в качестве главного (и едва ли не единственного) противника.

2. Защита от побочных излучений и наводок

2.1 Известно два основных метода защиты: активный и пассивный.

Активный метод предполагает применение специальных широкополосных передатчиков помех. Метод хорош тем, что устраняется не только угроза утечки информации по каналам побочного излучения компьютера, но и многие другие угрозы. Как правило, становится невозможным также и применение закладных подслушивающих устройств. Становится невозможной разведка с использованием излучения всех других устройств, расположенных в защищаемом помещении. Но этот метод имеет и недостатки. Во-первых, достаточно мощный источник излучения никогда не считался полезным для здоровья. Во-вторых, наличие маскирующего излучения свидетельствует, что в данном помещении есть серьезные секреты. Это само по себе будет привлекать к этому помещению повышенный интерес ваших недоброжелателей. В-третьих, при определенных условиях метод не обеспечивает гарантированную защиту компьютерной информации.

Обоих этих недостатков лишен пассивный метод. Заключается он в экранировании источника излучения (доработка компьютера), размещении источника излучения (компьютера) в экранированном шкафу или в экранировании помещения целиком. В целом, конечно, для защиты информации пригодны

оба метода. Но при одном условии: если у вас есть подтверждение того, что принятые меры действительно обеспечивают требуемую эффективность защиты.

Применяя активный метод, то имейте в виду, что уровень создаваемого источником шума излучения никак не может быть рассчитан. В одной точке пространства уровень излучения источника помех превышает уровень излучения компьютера, а в другой точке пространства или на другой частоте это может и не обеспечиваться. Поэтому после установки источников шума необходимо проведение сложных измерений по всему периметру охраняемой зоны и для всех частот. Процедуру проверки необходимо повторять всякий раз, когда вы просто изменили расположение компьютеров, не говоря уж об установке новых. Это может быть настолько дорого, что, наверное, стоит подумать и о других способах.

Если такие измерения не проводились, то это называется применить меры защиты «на всякий случай». Как правило, такое решение даже хуже, чем решение не предпринимать никаких мер. Ведь будут затрачены средства, все будут считать, что информация защищена, а реальная защита может вовсе и не обеспечиваться.

Каким бы путем вы ни шли, обязательным условием защиты является получение документального подтверждения эффективности принятых мер.

Если это специальное оборудование помещения (экранирование, установка генераторов шума), то детальному обследованию подлежит очень большая территория, что, конечно, недешево. В настоящее время на рынке средств защиты предлагают законченные изделия - экранированные комнаты и боксы. Они, безусловно, очень хорошо выполняют свои функции, но и стоят тоже очень хорошо.

Поэтому в наших условиях реальным остается только экранирование самого источника излучения - компьютера. Причем экранировать необходимо все. У некоторых сначала даже вызывает улыбку то, что мы экранируем, например, мышь вместе с ее хвостиком. Никто не верит, что из движения мыши можно выудить полезную информацию. А я тоже в это не верю. Мышь экранируется по той причине, что хотя она сама, может, и не является источником информации, но она своим хвостиком подключена к системному блоку. Этот хвостик является великолепной антенной, которая излучает все, что генерируется в системном блоке. Если хорошо заэкранировать монитор, то гармоники видеосигнала монитора будут излучаться системным блоком, в том числе и через хвостик мыши, поскольку видеосигналы вырабатываются видеокартой в системном блоке.

Десять лет назад экранированный компьютер выглядел настолько уродливо, что ни один современный руководитель не стал бы его покупать, даже если этот компьютер вообще ничего не излучает.

Современные же технологии основаны на нанесении (например, напылении) различных специальных материалов на внутреннюю поверхность существующего корпуса, поэтому внешний вид компьютера практически не изменяется.

Экранирование компьютера даже с применением современных технологий - сложный процесс. В излучении одного элемента преобладает электриче-

ская составляющая, а в излучении другого – магнитная, следовательно необходимо применять разные материалы. У одного монитора экран плоский, у другого - цилиндрический, а у третьего с двумя радиусами кривизны. Поэтому реально доработка компьютера осуществляется в несколько этапов. Вначале осуществляется специсследование собранного компьютера. Определяются частоты и уровни излучения. После этого идут этапы анализа конструктивного исполнения компьютера, разработки технических требований, выбора методов защиты, разработки технологических решений и разработки конструкторской документации для данного конкретного изделия (или партии однотипных изделий). После этого изделие поступает собственно в производство, где и выполняются работы по защите всех элементов компьютера. После этого в обязательном порядке проводятся специспытания, позволяющие подтвердить эффективность принятых решений. Если специспытания прошли успешно, заказчику выдается документ, дающий уверенность, что компьютер защищен от утечки информации по каналам побочного радиоизлучения.

Комплектующие для сборки ПК поставляются из-за рубежа. С периодичностью 3-6 месяцев происходит изменение их конструкторских решений, технических характеристик, форм, габаритов и конфигураций. Следовательно, технология, ориентированная на защиту каждой новой модели ПК, требует высочайшей маневренности производства. При этом возможен вариант изготовления из металла набора универсальных корпусных изделий и размещения в них комплектующих ПК, а также периферийных устройств зарубежного производства. Недостатком этого подхода является то, что он приемлем толь- ко для полигонного или катастрофоустойчивого исполнения. Другой вариант - это выбор комплектующих для ПК из большого количества однотипных изде- лий по признаку минимальной излучательной способности. Этот вариант не- обходимо рассматривать как непрофессиональный подход к проблеме, так как он противоречит нормативной документации.

2.2 Активный метод защиты компьютерной информации от утечки по ПЭМИН.

Вариант защиты компьютерной информации методом зашумления (радиомаскировки) предполагает использование генераторов шума в помещении, где установлены средства обработки конфиденциальной информации.

Обеспечивается зашумление следующими типами генераторов.

Генератор шума SEL SP-21 "Баррикада"

Система пространственного электромагнитного зашумления (система активной защиты) SEL SP-21B1 'Баррикада' предназначена для предупреждения перехвата информативных побочных электромагнитных излучений и наводок при обработке информации ограниченного распространения в средствах вычислительной техники. Устройство генерирует широкополосный шумовой электромагнитный сигнал и обеспечивает маскировку побочных электромагнитных излучений средств офисной техники, защиту от подслушивающих устройств, передающих информацию по радиоканалу (некварцованных, мощностью до 5 мВт).

Отличительные особенности: малогабаритность и наличие двух телескопических антенн позволяют оперативно устанавливать систему и обойтись без прокладки рамочных антенн по периметру помещений; возможность питания от аккумуляторов позволяет использовать систему вне помещений (например, в автомобиле).

Генератор шума SEL SP-21B2 "Спектр"

Обеспечивает защиту от утечки информации за счет побочных излучений и наводок средств офисной техники и при использовании миниатюрных радиопередающих устройств мощностью до 20 мВт.

Отличительные особенности: использование двух телескопических ан- тенн для формирования равномерного шумового спектра; возможность пита- ния от аккумулятора автомобиля.

Генератор шума "Равнина-5И"

Широкополосный искровой генератор "Равнина-5И" предназначен для маскировки побочных электромагнитных излучений персональных компьюте-

ров, рабочих станций компьютерных сетей и комплексов на объектах вычислительной техники путем формирования и излучения в пространство электромагнитного поля шума.

Отличительные особенности: искровой принцип формирования шумового сигнала; наличие 2-х телескопических антенн, позволяющих корректировать равномерность спектра; наличие шумового и модуляционного (с глубиной модуляции 100%) режимов работы.

Генератор шума "Гном-3"

Предназначен для защиты от утечки информации за счет побочных электромагнитных излучений и наводок средств офисной техники.

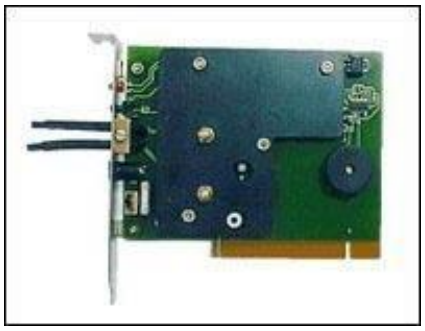
Отличительные особенности: использование рамочных антенн, располагаемых в 3-х взаимно перпендикулярных плоскостях для создания пространственного распределения шумового сигнала; возможность использования для защиты как персональных ЭВМ, так больших ЭВМ.

Генератор шума ГШ-1000М

Предназначен для защиты от утечки информации за счет побочных электромагнитных излучений и наводок средств офисной техники на объектах 2 и 3 категорий.

Отличительная особенность: использование рамочной антенны для создания пространственного зашумления.

Генератор шума ГШ-К-1000М



Предназначен для защиты от утечки информации за счет побочных электромагнитных излучений и наводок средств офисной техники на объектах 2 и 3 категорий.

Отличительные особенности: использование рамочной антенны для создания пространственного зашумления; установка в свободный слот персонального компьютера; выпускаются для слотов PCI и ISA.

Комбинированный генератор шума "Заслон"

Предназначен для использования в качестве системы активной защиты информации от утечки за счет побочных излучений и наводок средств офис- ной техники.

Отличительные особенности: использование 6-и независимых источников для формирования сигналов зашумления: в сети электропитания, шине заземления, 4-х проводной телефонной линии и в пространстве.

2.3 Пассивный метод защиты компьютерной информайии от утечки по ПЭМИН.

Новый подход к решению задач защиты информации базируется на пассивном методе (экранирование и фильтрация), но в отличие от прежних универсальных вариантов его применения, мы предлагаем индивидуальный под- ход к закрытию каналов утечки информации. В основу индивидуального под- хода положен анализ устройств и комплектующих ПК с целью определения общих конструкторских и схемотехнических решений исполнения, определе- ния параметров побочных излучений и на основании анализа этих данных осуществляются мероприятия по защите. В общем случае ПК состоит из:

системного блока;

монитора; клавиатуры;
манипулятора (мышь);
принтера; акустической
системы.

Анализ конструктивного исполнения устройств ПК позволил определить у них обобщенные признаки подобия (ОПП) и различия в зависимости от функционального назначения.

1. Системный блок. Большое многообразие корпусов вертикального и горизонтального исполнения.

ОПП: каркас, кожух, передняя панель, органы управления и индикации, блок питания и ввод-вывод коммуникаций.

2. Монитор. Различные геометрические формы корпусов из пластмассы, три типа экранов (ЭЛТ): плоский, цилиндрический и с двумя радиусами кри- визны в различных плоскостях.

ОПП: пластмассовые корпусные детали, ввод коммуникаций, органы управления и сигнализации.

3. Клавиатура. Незначительные различия в геометрии корпусов из пластмассы (у некоторых типов поддон из метала).

ОПП: пластмассовые корпусные детали, ввод коммуникаций и органы сигнализации.

4. Манипулятор (мышь). Незначительные различия в геометрии корпус- ных деталей из пластмассы.

ОПП: пластмассовые корпусные детали, ввод коммуникаций.

5. Принтер (лазерный, струйный). Корпуса различной геометрии из пластмассы, органы управления и различные разъемные соединения.

ОПП: пластмассовые корпусные детали, ввод коммуникаций, органы управления и сигнализации.

6. Акустические системы. Большое многообразие геометрических форм корпусов из пластмассы и дерева.

ОПП: ввод-вывод коммуникаций, органы управления и сигнализации, а для отдельных групп - пластмассовые корпусные детали.

Таким образом, обобщенные признаки подобия образуют три основные группы, присущие базовому составу ПК, с которым приходится работать при решении задач защиты информации, такие как:

- корпусные детали из
  пластмассы;ввод-вывод
  коммуникаций;
- органы управления и сигнализации.

При этом учитываются и общесистемные проблемные вопросы, как-то:

- разводка и организация электропитания и шин заземления;
  согласование сопротивлений источников и нагрузок;
- блокирование взаимного ЭМИ устройств

ПК; исключение влияния
электростатического поля;эргономика
рабочего места и т.д.

Следующий этап - это разработка типовых конструкторско-технологических решений, реализация которых направлена на предотвраще- ние утечки информации за счет расширения функций конструктивов уст- ройств ПК. Набор типовых конструкторско-технологических решений варьи- руется в зависимости от состава устройств ПК, но для базовой модели ПК с учетом обобщенных признаков подобия он содержит решения по:

- металлизации внутренних поверхностей деталей из пластмассы;
- экранированию проводных коммуникаций;
- согласованию сопротивлений источников и нагрузок;
- экранированию стекол для монитора и изготовлению заготовок различных форм из стекла;
- фильтрации сетевого электропитания и его защите от перенапряжений;
- нейтрализации влияния электростатического поля;
- расположению общесистемных проводных связей;
- точечной локализации ЭМИ;
- исключению ЭМИ органами управления и сигнализации;
- радиогерметизирующим уплотнителям из различных материалов;
- исключению взаимного влияния ЭМИ устройств ПК.

Контрольные вопросы ПЗ3 (ПК-2):

1. Что такое Побочные излучения?
2. Что такое перехват ПЭМИН?
3. Как выполняется Защита от побочных излучений и наводок?
4. Какие есть два основных метода защиты?
5. Поясните активный метод защиты компьютерной информации от утечки по ПЭМИН
6. Поясните Пассивный метод защиты компьютерной информации от утечки по ПЭМИН

Практическое занятие 4. *ВИДЫ ШИФРОВ. АЛГОРИТМЫ РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ*

Задание 1: Виды шифров.

Цель работы: получение навыков создания простейшей криптосистемы симметричного шифрования.

Краткие теоретические сведения Основные понятия криптографии

Криптография представляет собой совокупность методов преобразования данных, направленных на то, чтобы сделать эти данные бесполезными для противника. Такие преобразования позволяют решить две главные проблемы защиты данных: *проблему конфиденциальности* (путем лишения противника возможности извлечь информацию из канала связи) и *проблему целостности* (путем лишения противника возможности изменить сообщение так, чтобы изменился его смысл, или ввести ложную информацию в канал связи).
Система защиты информации, основанная на методах криптографии обычно называется *криптографической системой*, или более коротко – *криптосистемой*. Обобщенная схема криптографической системы показана на рис. 1.1.

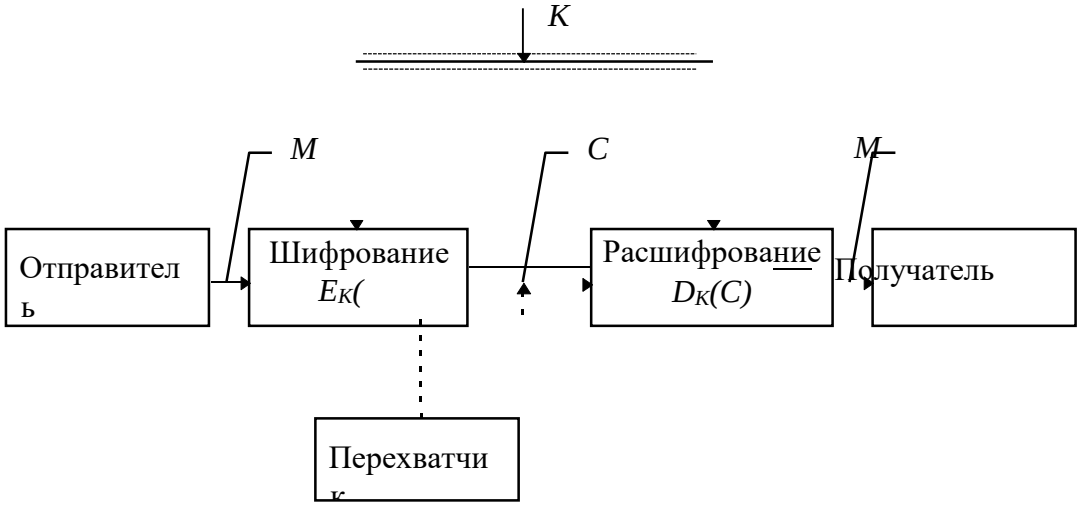


Рис. 1.1. Обобщенная схема криптосистемы

Согласно этой схемы, отправитель генерирует открытый текст исходного сообщения M, которое должно быть передано законному получателю по незащищенному каналу. За каналом следит перехватчик с целью перехватить и раскрыть передаваемое сообщение. Для того чтобы перехватчик не смог узнать содержание сообщения M, отправитель шифрует его с помощью обратимого преобразования E_K и получает *шифртекст* (или *криптограмму*) $C=E_K(M)$, который отправляет получателю.

Преобразование E_K выбирается из семейства криптографических преобразований, называемых *криптоалгоритмами*. Параметр, с помощью которого выби-рается отдельное используемое преобразование, называется *криптографиче- ским ключом K*. Таким образом, процесс преобразования открытого текста с целью сделать не-понятным его смысл для посторонних называется *шифрованием*. В результате шифрования получается шифртекст. Процесс обратного преобразования шиф-ртекста в открытый текст называется *расшифрованием*.

Под *шифром* понимают совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, задаваемых ключоми алгоритмом криптографического преобразования.

Криптосистема имеет разные варианты реализации: набор инструкций, аппа-ратные средства, комплекс программ компьютера, которые позволяют зашиф-ровать открытый текст и расшифровать шифортекст различными способами, один из которых выбирается с помощью конкретного ключа *K*.

Криптосистемы, в которых применятся один и тот же секретный ключ, как при шифровании, так и при расшифровании сообщений, называются *симмет-ричными.*

Проблема криптографического преобразования информации интересует чело-вечество на протяжении нескольких тысячелетий. Шифры, дошедшие до нас с древнейших времен, основаны на двух основных принципах: *перестановке* и *замене* (*подстановки*).

Шифры перестановки

Шифрование перестановкой заключается в том, что символы открытого текста переставляются по определенному правилу в пределах некоторого бло- ка этого текста. Рассмотрим перестановку, предназначенную для шифрования сообщения длиной *n* символов. Его можно представить с помощью таблицы где i_1 – номер места шифртекста, на которое попадает первая буква открытого текста при выбранном преобразовании, i_2 — номер места для второй буквы и т. д. В верхней строке таблицы выписаны по порядку числа от 1 до *n,* а в ниж-ней те же числа, но в произвольном порядке. Такая таблица называется пере-становкой степени *n*.

Зная перестановку, задающую преобразование, можно осуществить как шиф-рование, так и расшифрование текста. В этом случае, сама таблица переста-новки служит ключом шифрования.

Число различных преобразований шифра перестановки, предназначенного для шифрования сообщений длины *n*, меньше либо равно *n*! (*n* факториал). Заме- тим, что в это число входит и вариант преобразования, оставляющий все сим-волы на своих местах.

С увеличением числа *n* значение *n*! растет очень быстро. Для использования на практике такой шифр не удобен, так как при больших значениях *n* прихо- дится работать с длинными таблицами. Поэтому широкое распространение получили шифры, использующие не саму таблицу перестановки, а некотороеправило,

порождающее эту таблицу. Рассмотрим несколько примеров таких шифров.
Шифр перестановки "скитала". Известно, что в V веке до нашей эры правители Спарты, наиболее воинственного из греческих государств, имели хорошо отработанную систему секретной военной связи и шифровали свои послания с помощью *скитала,* первого простейшего криптографического устройства, реализующего метод простой перестановки.
Шифрование выполнялось следующим образом. На стержень цилиндрической формы, который назывался скитала, наматывали спиралью (виток к витку) полоску пергамента и писали на ней вдоль стержня несколько строк текста сообщения (рис. 1.2). Затем снимали со стержня полоску пергамента с написанным текстом. Буквы на этой полоске оказывались расположенными хаотично.

Рис. 1.2. Шифр "Скитала"

Такой же результат можно получить, если буквы сообщения писать по кольцуне подряд, а через определенное число позиций до тех пор, пока не будет исчерпан весь текст. Сообщение "**НАСТУПАЙТЕ**" при размещении его по окружности стержня по три буквы дает шифртекст: "**НУТАПЕСА_ТЙ**".

Для расшифрования такого шифртекста нужно не только знать правило шифрования, но и обладать ключом в виде стержня определенного диаметра. Зная только вид шифра, но не имея ключа, расшифровать сообщение было непросто.
Шифрующие таблицы. С начала эпохи Возрождения (конец XIV столетия) начала возрождаться и криптография. В разработанных шифрах перестановки того времени применяются шифрующие таблицы, которые, в сущности, зада-ют правила перестановки букв в сообщении.
В качестве ключа в шифрующих таблицах используются:

- размер таблицы;
- слово или фраза, задающие перестановку;
- особенности структуры таблицы.

Одним из самых примитивных табличных шифров перестановки является простая перестановка, для которой ключом служит размер таблицы. Этот ме- тод шифрования сходен с шифром скитала. Например, сообщение "**ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ**" записывается в таблицу поочередно по столбцам. Результат заполнения таблицы из 5 строк и 7 столбцов показан на рис. 1.3.
После заполнения таблицы текстом сообщения по столбцам для формирова- ния шифртекста считывают содержимое таблицы по строкам. Если шифртекст записывать группами по пять букв, получается такое шифрованное сообще- ние: "**ТНПВЕ ГЛЕАР АДОНР ТИЕЬВ ОМОБТ МПЧИР ЫСООЬ**".

Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

Рис. 1.3. Заполнение шифрующей таблицы из 5 строк и 7 столбцов

Естественно, отправитель и получатель сообщения должны заранее условиться об общем ключе в виде размера таблицы. Следует заметить, что объединение букв шифртекста в 5-буквенные группы не входит в ключ шифра и осуществляется для удобства записи несмыслового текста. При расшифрова- нии действия выполняют в обратном порядке.

Несколько большей стойкостью к раскрытию обладает метод шифрования, называемый **одиночной перестановкой по ключу**. Этот метод отличается от предыдущего тем, что столбцы таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

Применим в качестве ключа, например, слово "**ПЕЛИКАН**", а текст сообщения возьмем из предыдущего примера. На рис. 1.4 показаны две таблицы, заполненные текстом сообщения и ключевым словом, при этом левая таблица соответствует заполнению до перестановки, а правая таблица – заполнению после перестановки.

Ключ →

П	Е	Л	И	К	А	Н
7	2	5	3	4	1	6
Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

До перестановки

А	Е	И	К	Л	Н	П
1	2	3	4	5	6	7
Г	Н	В	Е	П	Л	Т
О	А	А	Д	Р	Н	Е
В	Т	Е	Ь	И	О	Р
П	О	Т	М	Б	Ч	М
О	Р	С	О	Ы	Ь	И

После перестановки

Рис. 1.4. Шифрующие таблицы, заполненные ключевым словом и текстом сообщения

В верхней строке левой таблицы записан ключ, а номера под буквами ключа определены в соответствии с естественным порядком соответствующих букв ключа в алфавите. Если бы в ключе встретились одинаковые буквы, они бы были понумерованы слева направо. В правой таблице столбцы переставлены в соответствии с упорядоченными номерами букв ключа.

При считывании содержимого правой таблицы по строкам и записи шифртекста группами по пять букв получим шифрованное сообщение: "**ГНВЕП ЛТООА ДРНЕВ ТЕЬИО РПОТМ БЧМОР СОЫЬИ**".

Для обеспечения дополнительной скрытности можно повторно зашифровать сообщение, которое уже прошло шифрование. Такой метод шифрования называется **двойной перестановкой**. В случае двойной перестановки столбцов и строк таблицы перестановки определяются отдельно для столбцов и отдельно для строк. Сначала в таблицу записывается текст сообщения, а потом поочередно переставляются столбцы, а затем строки. При расшифровании порядок перестановок должен быть обратным.

Пример выполнения шифрования методом двойной перестановки показан на рис. 1.5. Если считывать шифртекст из правой таблицы построчно блоками по четыре буквы, то получится следующее: "**ТЮАЕ ООГМ РЛИП ОЬСВ**".

	4	1	3	2
3	П	Р	И	Л
1	Е	Т	А	Ю
4	В	О	С	Ь
2	М	О	Г	О

Исходная таблица

	1	2	3	4
3	Р	Л	И	П
1	Т	Ю	А	Е
4	О	Ь	С	В
2	О	О	Г	М

Перестановка столбцов

	1	2	3	4
1	Т	Ю	А	Е
2	О	О	Г	М
3	Р	Л	И	П
4	О	Ь	С	В

Перестановка строк

Рис. 1.5. Пример выполнения шифрования методом двойной перестановки

Ключом к шифру двойной перестановки служит последовательность номеров столбцов и номеров строк исходной таблицы (в нашем примере последовательности 4132 и 3142 соответственно).

Число вариантов двойной перестановки быстро возрастает при увеличении размера таблицы:

- для таблицы 3x3 36 вариантов;
- для таблицы 4x4 576 вариантов;
- для таблицы 5x5 14400 вариантов.

Шифрование с помощью магических квадратов. В средние века для шифрования перестановкой применялись и магические квадраты. *Магическими квадратами* называют квадратные таблицы с вписанными в их клетки последовательными натуральными числами, начиная от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число.

Шифруемый текст вписывали в магические квадраты в соответствии с нумерацией их клеток. Если затем выписать содержимое такой таблицы по строкам, то получится шифртекст, сформированный благодаря перестановке букв исходного сообщения.

Пример магического квадрата и его заполнения сообщением "**ПРИЛЕТАЮ**

ВОСЬМОГО" показан на рис. 1.6.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

О	И	Р	М
Е	О	С	Ю
В	Т	А	Ь
Л	Г	О	П

Рис. 1.6. Пример магического квадрата 4x4 и его заполнение сообщением

Шифртекст, получаемый при считывании содержимого правой таблицы по строкам, имеет вполне загадочный вид: "**ОИРМ ЕОСЮ ВТАЪ ЛГОП**".

Число магических квадратов быстро возрастает с увеличением размера квадрата. Существует только один магический квадрат размером 3x3 (если не учитывать его повороты). Количество магических квадратов 4x4 составляет уже 880, а количество магических квадратов 5x5 – около 250000.

Магические квадраты средних и больших размеров могли служить хорошей базой для обеспечения нужд шифрования того времени, поскольку практиче- ски нереально выполнить вручную перебор всех вариантов для такого шифра.

Шифры замены

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.

ние a_i (соответствующая a_i буква алфавита шифртекста).

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

Алфавиты A_n и B_n не обязательно должны быть различными. В практической криптографии очень часто применяются шифры, в которых алфавиты A_n и B_n 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 тек-ста применяют свой шифр простой замены. Для реализации шифров сложной замены последовательно и циклически меняют используемые таблицы подстановки.

Полибианский квадрат. Одним из первых шифров простой замены считается

так называемый *полибианский квадрат.* За два века до нашей эры греческий писатель и историк Полибий изобрел для целей шифрования квадратную таблицу размером 5x5, заполненную буквами греческого алфавита в случайном порядке (рис. 1.7).

λ	ε	υ	ω	γ
ρ	ζ	δ	σ	ο
μ	η	β	ξ	τ
ψ	π	θ	α	x
χ	ν		φ	ι

Рис. 1.7. Полибианский квадрат

При шифровании в этом полибианском квадрате находили очередную букву открытого текста и записывали в шифртекст букву, расположенную ни- же ее в том же столбце. Если буква текста оказывалась в нижней строке таб- лицы, то для шифртекста брали самую верхнюю букву из того же столбца. Например, для слова "ταυροσ" получается шифртекст "Χφδμτξ".

Концепция полибианского квадрата оказалась плодотворной и нашла применение в криптосистемах последующего времени.

Система шифрования Цезаря. Шифр Цезаря является частным случаем шифра простой замены. Свое название этот шифр получил по имени римского императора Гая Юлия Цезаря, который использовал этот шифр при переписке с Цицероном (около 50 г. до н.э.).

При шифровании исходного текста каждая буква заменялась на другую букву того же алфавита по следующему правилу. Заменяющая буква определялась путем смещения по алфавиту от исходной буквы на *K* букв. При достижении конца алфавита выполнялся циклический переход к его началу. Цезарь использовал шифр замены при смещении *K* = 3. Такой шифр замены можно задать таблицей подстановки, содержащей соответствующие пары букв открытого текста и шифртекста. Совокупность возможных подстановок для *K*=3 показана в табл. 1.1.

Таблица 1.1.

Таблица подстановкИ шифра Цезаря

A→D	J→M	S→V
B→E	K→N	T→W
C→F	L→O	U→X
D→G	M→P	V→Y

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

Например, послание Цезаря "**VENI VIDI VICI**" ("Пришел, Увидел, Победил") выглядело бы в зашифрованном виде так: "**YHQL YLGL YLFL**".

Система шифрования Цезаря с ключевым словом. Особенностью этой системы является использование ключевого слова для смещения и изменения порядка символов в алфавите подстановки.

Выберем некоторое число k, $0 < k < 25$, и слово или короткую фразу в качест-ве ключевого слова. Желательно, чтобы все буквы ключевого слова были различными. Пусть выбраны слово DIPLOMAT в качестве ключевого слова и число $k = 5$.

Ключевое слово записывается под буквами алфавита, начиная с буквы, числовой код которой совпадает с выбранным числом k:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	2
A	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***	***K***	***L***	***M***	***N***	***O***	***P***	***Q***	***R***	***S***	***T***	***U***	***V***	***W***
					D	***I***	***P***	***L***	***O***	***M***	***A***	***T***										

Оставшиеся буквы алфавита подстановки записываются после ключево-го слова в алфавитном порядке:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	2
A	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***	***K***	***L***	***M***	***N***	***O***	***P***	***Q***	***R***	***S***	***T***	***U***	***V***	
V	***W***	***X***	***Y***	***Z***	***D***	***I***	***P***	***L***	***O***	***M***	***A***	***T***	***B***	***C***	***E***	***F***	***G***	***H***	***J***	***K***	***N***	

Теперь мы имеем подстановку для каждой буквы произвольного сообщения. Исходное сообщение "**SEND MORE MONEY**" шифруется как "**HZBY TCGZ TCBZS**".

Следует отметить, что требование о различии всех букв ключевого слова не обязательно. Можно просто записать ключевое слово (или фразу) без повторения одинаковых букв.

Шифрующие таблицы Трисемуса. В 1508 г. аббат из Германии Иоганн Трисемус написал печатную работу по криптологии под названием "Полиграфия".В этой книге он впервые систематически описал применение шифрующих таблиц, заполненных алфавитом в случайном порядке. Для получения такого шифра замены обычно использовались таблица для записи букв алфавита и ключевое слово (или фраза). В таблицу сначала вписывалось по строкам клю- чевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. Поскольку ключевое слово или фразу легко хранить в памяти, то такой подход упрощал процессы шифрования и расшифрования.

Поясним этот метод шифрования на примере. Для русского алфавита

шифрующая таблица может иметь размер 4х8. Выберем в качестве ключа сло- во БАНДЕРОЛЬ. Шифрующая таблица с таким ключом показана на рис. 1.8.

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

Рис. 1.8. Шифрующая таблица Трисемуса с ключевым словом "БАНДЕРОЛЬ"

Как и в случае полибианского квадрата, при шифровании находят в этой таблице очередную букву открытого текста и записывают в шифртекст букву, расположенную ниже ее в том же столбце. Если буква текста оказывается в нижней строке таблицы, тогда для шифртекста берут самую верхнюю букву из того же столбца.

Например, при шифровании с помощью этой таблицы сообщения "**ВЫЛЕ-ТАЕМПЯТОГО**" получаем шифртекст "**ПДКЗЫВЗЧШЛЫЙСЙ**".

Система шифрования Вижинера впервые была опубликована в 1586 г. и яв-ляется одной из старейших и наиболее известных шифров сложной замены.

Свое название она получила по имени французского дипломата XVI века Бле- за Вижинера, который развивал и совершенствовал криптографические систе-мы.

Система Вижинера подобна такой системе шифрования Цезаря, у которой ключ подстановки меняется от буквы к букве. Этот шифр можно описать таб-лицей шифрования, называемой таблицей (квадратом) Вижинера. Пример квадрата Вижинера для русского языка приведен в табл. 1.2.

Таблица Вижинера используется для шифрования и расшифрования. Таблица имеет два входа: верхнюю строку подчеркнутых символов, используемую для считывания очередной буквы исходного открытого текста; крайний левый столбец ключа. Последовательность ключей обычно получают из числовых значений букв ключевого слова.

При шифровании исходного сообщения его выписывают в строку, а под ним записывают ключевое слово (или фразу). Если ключ оказался короче сообще-ния, то его циклически повторяют. В процессе шифрования находят в верхней строке таблицы очередную букву исходного текста и в левом столбце очеред-ное значение ключа. Очередная буква шифртекста находится на пересечении столбца, определяемого шифруемой буквой, и строки, определяемой число- вым значением ключа.

Рассмотрим пример получения шифртекста с помощью таблицы Вижинера. Пусть выбрано ключевое слово "**АМБРОЗИЯ**". Необходимо зашифровать со-общение "**ПРИЛЕТАЮ СЕДЬМОГО**".

Выпишем исходное сообщение в строку и запишем под ним ключевое слово с повторением. В третью строку будем выписывать буквы шифртекста, определяемые из таблицы Вижинера.

Сообщение ***ПРИЛЕТАЮ СЕДЬМОГО***
Ключ ***АМБРОЗИЯ АМБРОЗИЯ***

Шифртекст ***ПЪЙЫУЩИЭ ССЕКЬХЛН***

Таблица 1.2.

Квадрат Вижиненра для русского языка

Ключ	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	ъ	э	ю	я
0	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	ъ	э	ю	я
1	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	ъ	э	ю	я	а
2	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	ъ	э	ю	я	а	б
3	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	ъ	э	ю	я	а	б	в
4	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г
5	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д
6	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е
7	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж
8	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з
9	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и
10	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й
11	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к
12	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л
13	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м
14	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н
15	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о
16	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
17	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р
18	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с
19	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т
20	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у
21	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф
22	ц	ч	ш	щ	ь	ы	ъ	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
23	ч	ш	щ	ь	ы	ъ	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
24	ш	щ	ь	ы	ъ	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
25	щ	ь	ы	ъ	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
26	ь	ы	ъ	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
27	ы	ъ	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
28	ъ	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
29	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	ъ
30	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	ъ	э
31	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	ъ	э	ю

Содержание работы

1. В соответствии со своим вариантом разработать программы для шифрова- ния и расшифровывания русскоязычного текста при помощи шифра пере- становки и шифра замены. Программы должны обеспечивать:
   - шифрование информации, находящейся в текстовом файле, с записью ре- зультата в другой файл;
   - шифрование информации, вводимой с клавиатуры, с записью в файл только зашифрованного текста;
   - ввод ключа шифрования/расшифрования с клавиатуры без отображения его на экране;
   - расшифровку текста, содержащегося в файле, с выводом результатов на экран или в другой файл (ключ расшифровывания вводится с клавиатуры без эха).
2. При помощи созданной программы подготовить зашифрованный русский текст объемом не менее 2000 символов.

Варианты заданий

№ варианта	Шифры
1	Шифр скитала, шифр Трисемуса
2	Шифр скитала, шифр Цезаря
3	Шифр скитала, шифр Цезаря с ключевым словом
4	Шифр скитала, шифр Вижиненра
5	Простая шифрующая таблица, шифр Трисемуса
6	Простая шифрующая таблица перестановки, шифр Цезаря
7	Простая шифрующая таблица перестановки, шифр Цезаря с ключевым словом
8	Простая шифрующая таблица перестановки, шифр Вижиненра
9	Одиночная перестановка по ключу, шифр Трисемуса
10	Одиночная перестановка по ключу, шифр Цезаря
11	Одиночная перестановка по ключу, шифр Цезаря с ключевым словом
12	Одиночная перестановка по ключу, шифр Вижиненра
13	Двойная перестановка по ключу, шифр Трисемуса
14	Двойная перестановка по ключу, шифр Цезаря
15	Двойная перестановка по ключу, шифр Цезаря с ключевым словом
16	Двойная перестановка по ключу, шифр Вижиненра

Задание 2: АЛГОРИТМЫ РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ

Алгоритм обмена ключами.

Протокол Ди ффи-Хе ллмана (англ. Diffie-Hellman, DH) — криптографический протокол, позволяющий двум и более сторонам получить общий секретный ключ, используя незащищенный от прослушивания канал связи. Полученный ключ используется для шифрования дальнейшего обмена с помощью алгоритмов симметричного шифрования.

Схема открытого распределения ключей, предложенная Диффи и Хеллманом, произвела настоящую революцию в мире шифрования, так как снимала основную проблему классической криптографии — проблему распределения ключей.

Передача ключа по открытым каналам была большой проблемой криптографии XX века. Но эту проблему удалось решить после появления алгоритма Диффи-Хеллмана. Данный алгоритм позволил дать ответ на главный вопрос:

«Как при обмене зашифрованными посланиями уйти от необходимости передачи секретного кода расшифровки, который, как правило, не меньше самого послания?» Открытое распространение ключей Диффи-Хеллмана позволяет паре пользователей системы выработать общий секретный ключ, не обмениваясь секретными данными.

Первая публикация данного алгоритма появилась в 70-х годах XX века в статье Диффи и Хеллмана, в которой вводились основные понятия криптографии с открытым ключом. *Алгоритм Диффи-Хеллмана* не применяется для шифрования сообщений или формирования электронной подписи. Его назначение – в распределении ключей. Он позволяет двум или более пользователям обменяться без посредников ключом, который может быть использован затем для симметричного шифрования. Это была первая криптосистема, которая позволяла защищать информацию без использования секретных ключей, передаваемых по защищенным каналам. Схема открытого распределения ключей, предложенная Диффи и Хеллманом, произвела настоящую революцию в мире шифрования, так как снимала основную проблему классической криптографии – проблему распределения ключей.

Алгоритм основан на трудности вычислений *дискретных логарифмов*. Попробуем разобраться, что это такое. В этом алгоритме, как и во многих других алгоритмах с открытым ключом, вычисления производятся по модулю некоторого большого простого числа P. Вначале специальным образом подбирается некоторое натуральное число A, меньшее P. Если мы хотим зашифровать значение X, то вычисляем

$Y = A^X \bmod P$.

Причем, имея X, вычислить Y легко. Обратная задача вычисле- ния X из Y является достаточно сложной. Экспонента X как раз и называет- ся *дискретным логарифмом Y*. Таким образом, зная о сложности вычисле- ния *дискретного логарифма*, число Y можно открыто передавать по любому каналу связи, так как при большом модуле P исходное значение X подобрать будет

практически невозможно. На этом математическом факте осно- ван *алгоритм Диффи-Хеллмана* для формирования ключа.

Формирование общего ключа

Пусть два пользователя, которых условно назовем пользователь 1 и пользователь 2, желают сформировать общий ключ для алгоритма симметричного шифрования. Вначале они должны выбрать большое простое число P и некоторое специальное число A, 1 < A < P-1, такое, что все числа из интервала [1, 2, ..., P-1] могут быть представлены как различные степени A mod P. Эти числа должны быть известны всем абонентам системы и могут выбираться открыто. Это будут так называемые *общие параметры*.

Затем первый пользователь выбирает число X_1 (X_1<P), которое желатель- но формировать с помощью датчика случайных чисел. Это будет закрытый ключ первого пользователя, и он должен держаться в секрете. На основе за-крытого ключа пользователь 1 вычисляет число

$$Y_1 = A^{X_1} \, mod \, P$$

которое он посылает второму абоненту.

Аналогично поступает и второй пользователь, генерируя X_2 и вычисляя

$$Y_2 = A^{X_2} \, mod \, P$$

Это значение пользователь 2 отправляет первому пользователю.

После этого у пользователей должна быть информация, указанная в сле-дующей таблице:

общий секретный ключ Z для сеанса симметричного шифрования. Воткак это должен сделать первый пользователь:

$$Z = (Y_2)^{X_1} \, mod \, P$$

Никто другой кроме пользователя 1 этого сделать не может, так как чис- ло X_1 секретно. Второй пользователь может получить то же самое число Z, используя свой закрытый ключ и открытый ключ своего абонента следующим образом:

$$Z = (Y_1)^{X_2} \, mod \, P$$

Если весь протокол формирования общего секретного ключа выполнен верно, значения Z у одного и второго абонента должны получиться одинаковыми. Причем, что самое важное, противник, не зная секретных чисел X_1 и X_2, не сможет вычислить число Z. Не зная X_1 и X_2, злоумышленник может попы- таться вычислить Z, используя только передаваемые открыто P, A, Y_1 и Y_2. Безопасность формирования общего ключа в алгоритме Диффи-Хеллмана вытекает из того факта, что, хотя относительно легко вычислить экспоненты по модулю простого числа, очень трудно вычислить дискретные логарифмы. Для больших простых чисел размером сотни и тысячи бит задача считается неразрешимой, так как требует колоссальных затрат вычислительных ресурсов.

Пользователи 1 и 2 могут использовать значение Z в качестве секретного ключа для шифрования и расшифрования данных. Таким же образом любая пара абонентов может вычислить секретный ключ, известный только им.

Пример вычислений по алгоритму

Пусть два абонента, желающие обмениваться через Интернет зашифрованными сообщениями, решили сформировать секретный ключ для очередного сеанса связи. Пусть они имеют следующие общие параметры:

P = 11, A = 7.

Каждый абонент выбирает секретное число X и вычисляет соответствующее ему открытое число Y. Пусть выбраны

$X_1 = 3$, $X_2 = 9$.

Вычисляем

$Y_1 = 7^3 \bmod 11 = 2$,

$Y_2 = 7^9 \bmod 11 = 8$.

Затем пользователи обмениваются открытыми ключами Y_1 и Y_2. После этого каждый из пользователей может вычислить общий секретный ключ:

пользователь 1: $Z = 8^3 \bmod 11 = 6$.
пользователь 2: $Z = 2^9 \bmod 11 = 6$.

Теперь они имеют общий ключ 6, который не передавался по каналу связи.

Контрольные вопросы ПЗ4 (ПК-3):

1. Поясните понятие криптографии.
2. постройте Обобщенную схему криптосистемы.
3. Что такое Шифры перестановки?
4. Что такое Шифрование с помощью магических квадратов?
5. Что такое Шифры замены?
6. Что означает Система шифрования Цезаря? Таблица подстановки шифра Цезаря?
7. Что означает Квадрат Вижиненра для русского языка?
8. Поясните алгоритмы распределения ключей

Практическое занятие №5. СИММЕТРИЧНЫЕ АЛГОРИТМЫ.

Симметричные криптосистемы (также симметричное шифрование, симметричные шифры) — способ шифрования, в котором для шифрования и расшифровывания применяется один и тот же криптографический ключ. До изобретения схемы асимметричного шифрования единственным существовавшим способом являлось симметричное шифрование. Ключ алгоритма должен сохраняться в секрете обеими сторонами. Алгоритм шифрования выбирается сторонами до начала обмена сообщениями.

Рассмотрим общую схему симметричной, или традиционной, криптографии.

Рис. Общая схема симметричного шифрования

В процессе шифрования используется определенный алгоритм шифрования, на вход которому подаются исходное незашифрованное сообщение, называемое также plaintext, и ключ. Выходом алгоритма является зашифрованное сообщение, называемое также ciphertext. Ключ является значением, не зависящим от шифруемого сообщения. Изменение ключа должно приводить к изменению зашифрованного сообщения.

Зашифрованное сообщение передается получателю. Получатель преобразует зашифрованное сообщение в исходное незашифрованное сообщение с помощью алгоритма дешифрования и того же самого ключа, который использовался при шифровании, или ключа, легко получаемого из ключа шифрования.

Незашифрованное сообщение будем обозначать P или M, от слов plaintext и message. Зашифрованное сообщение будем обозначать C, от слова ciphertext.

Характерная особенность симметричных блочных алгоритмов заключается в том, что в ходе своей работы они производят преобразование блока входной информации фиксированной длины и получают результирующий блок того же объема, но не доступный для прочтения сторонним лицам, не владеющим ключом. Таким образом, схему работы симметричного блочного шифра можно описать функциями:

Функция.

C = EK (M),

M = DK (C),

где M – исходный (открытый) блок данных;C

– зашифрованный блок данных.

Ключ К является параметром симметричного блочного криптоалгоритма и представляет собой блок двоичной информации фиксированного размера. Исходный М и зашифрованный С блоки данных также имеют равную фиксированную разрядность (но не обязательно равную длине ключа К).

ГОСТ 28147-89 - отечественный стандарт на шифрование данных. Стандарт включает три алгоритма зашифровывания (расшифровывания) данных: режим простой замены, режим гаммирования, режим гаммирования с обратной связью - и режим выработки имитовставки.

С помощью имитовставки можно зафиксировать случайную или умышленную модификацию зашифрованной информации. Вырабатывать имитовставку можно или перед зашифровыванием (после расшифровывания) всего сообщения, или одновременно с зашифровыванием (расшифровыванием) по блокам. При этом блок информации шифруется первыми шестнадцатью циклами в режиме простой замены, затем складывается по модулю 2 со вторым блоком, результат суммирования вновь шифруется первыми шестнадцатью циклами и т. д.

Алгоритмы шифрования ГОСТ 28147-89 обладают достоинствами других алгоритмов для симметричных систем и превосходят их своими возможностями. Так, ГОСТ 28147-89 (256-битовый ключ, 32 цикла шифрования) по сравнению с такими алгоритмами, как DES (56-битовый ключ, 16 циклов шифрования) и FEAL-1 (64-битовый ключ, 4 цикла шифрования) обладает более высокой криптостойкостью за счет более длинного ключа и большего числа циклов шифрования.

Следует отметить, что в отличие от DES, у ГОСТ 28147-89 блок подстановки можно произвольно изменять, то есть он является дополнительным 512-битовым ключом.

DES (англ. data encryption standard) — алгоритм для симметричного шифрования, разработанный фирмой IBM и утверждённый правительством США в 1977 году как официальный стандарт (FIPS 46-3). Размер блока для DES равен 64 бита. В основе алгоритма лежит сеть Фейстеля с 16 циклами (раундами) и ключом, имеющим длину 56 бит. Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP-1) преобразований. Для DES рекомендовано несколько режимов.

Контрольные вопросы ПЗ5 (ПК-2):

1. Что такое Симметричные криптосистемы?
2. Постройте Общую схему симметричного шифрования
3. Характерная особенность симметричных блочных алгоритмов?
4. Опишите алгоритм для симметричного шифрования DES.
5. Опишите отечественный стандарт на шифрование данных.
6. Абсолютный и относительный адрес ячейки (ссылки) ГОСТ 28147-89.
7. Какой алгоритм используется в процессе шифрования?

Практическое занятие №6. АСИМЕТРИЧНЫЕ АЛГОРИТМЫ.

Асимметричные алгоритмы шифрования называются также алгоритмами с открытым ключом. В отличие от алгоритмов симметричного шифрования (алгоритмов шифрования с закрытым ключом), в которых для шифрования и расшифрования используется один и тот же ключ, в ассиметричных алгоритмах один ключ используется для шифрования, а другой, отличный от первого, – для расшифрования. Алгоритмы называются асимметричными, так как ключи шифрования и расшифрования разные, следовательно, отсутствует симметрия основных криптографических процессов. Один из двух ключей является открытым (public key) и может быть объявлен всем, а второй – закрытым (private key) и должен держаться в секрете. Какой из ключей, открытый или закрытый, используется для шифрования, а какой для расшифрования, определяется назначением криптографической системы.

В настоящее время асимметричные алгоритмы широко применяются на практике, например, для обеспечения информационной безопасности телекоммуникационных сетей, в том числе сетей, имеющих сложную топологию; для обеспечения информационной безопасности в глобальной сети Internet; в различных банковских и платежных системах (в том числе использующих интеллектуальные карты).

Алгоритмы шифрования с открытым ключом можно использовать для решения, как минимум, трех задач:

- Для шифрования передаваемых и хранимых данных в целях их защиты от несанкционированного доступа.
- Для формирования цифровой подписи под электронными документами.
- Для распределения секретных ключей, используемых потом при шифровании документов симметричными методами.

Алгоритм шифрования с открытым ключом RSA был предложен од- ним из первых в конце 70-х годов XX века. Его название составлено из первых букв фамилий авторов: Р.Райвеста (R.Rivest), А.Шамира (A.Shamir) и Л.Адлемана (L.Adleman). Алгоритм RSA является, наверно, наиболее попу- лярным и широко применяемым *асимметричным алгоритмом* в криптографи-ческих системах.

Алгоритм основан на использовании того факта, что задача разложения большого числа на простые сомножители является трудной. Криптографиче-ская система RSA базируется на следующих двух фактах из теории чисел:

1. задача проверки числа на простоту является сравнительно легкой;
2. задача разложения чисел вида n = pq (p и q — простые числа); на множители является очень трудной, если мы знаем только n, а p и q — большие числа (это так называемая задача факторизации,

Алгоритм RSA представляет собой блочный алгоритм шифрования, где зашифрованные и незашифрованные данные должны быть представлены в ви- де

целых чисел между 0 и n -1 для некоторого n.

Шифрование

Итак, рассмотрим сам алгоритм. Пусть абонент А хочет передать зашифрованное сообщение абоненту Б. В этом случае абонент Б должен подготовить пару (открытый ключ; закрытый ключ) и отправить свой открытый ключ пользователю А.

Первым этапом является генерация открытого и закрытого ключей. Для этого вначале выбираются два больших простых числа P и Q. Затем вычисляется произведение N:

$N = PQ$.

После этого определяется вспомогательное число f:f =

(P - l)(Q - 1).

Затем случайным образом выбирается число d < f и взаимно простое с f.

Далее необходимо найти число e, такое, чтоed

mod f = 1.

Числа d и N будут открытым ключом пользователя, а значение e – закрытым ключом.

Таким образом, на этом этапе у пользователя должна быть информация, указанная в следующей таблице:

Б хочет получить зашифрованное сообщение от пользователя А, значит пользователь Б должен отправить свой открытый ключ (d, N) пользователю А. Числа P и Q больше не нужны, однако их нельзя никому сообщать; лучше все- го их вообще забыть.

На этом этап подготовки ключей закончен и можно использовать основ- ной протокол RSA для шифрования данных.

Второй этап – шифрование данных. Если абонент А хочет передать некоторые данные абоненту Б, он должен представить свое сообщение в цифровом виде и разбить его на блоки m_1, m_2, m_3, ... , где $m_i < N$. Зашифрованное сообщение будет состоять из блоков c_i.

Абонент А шифрует каждый блок своего сообщения по формуле c_i

$= m_i^d \bmod N$

используя *открытые параметры* пользователя Б, и пересылает зашифрованное сообщение $C=(c_1, c_2, c_3, ...)$ по открытой линии.

Абонент Б, получивший зашифрованное сообщение, расшифровывает все блоки полученного сообщения по формуле

$m_i = c^e \bmod N$

Все расшифрованные блоки будут точно такими же, как и исходящие от пользователя А.

Злоумышленник, перехватывающий все сообщения и знающий всю открытую информацию, не сможет найти исходное сообщение при больших значениях P и Q.

Асимметричный алгоритм, предложенный в 1985 году Эль-Гамалем (T. ElGamal), универсален. Он может быть использован для решения всех трех

основных задач: для шифрования данных, для формирования цифровой подписи и для согласования общего ключа. Кроме того, возможны модификации алгоритма для схем проверки пароля, доказательства идентичности сообщения и другие варианты. Безопасность этого алгоритма, так же как и *алгоритма Диффи-Хеллмана*, основана на трудности вычисления *дискретных логариф- мов*. Этот алгоритм фактически использует схему Диффи-Хеллмана, чтобы сформировать общий секретный ключ для абонентов, передающих друг другу сообщение, и затем сообщение шифруется путем умножения его на этот ключ.

И в случае шифрования, и в случае формирования цифровой подписи каждому пользователю необходимо сгенерировать пару ключей. Для этого, так же как и в схеме Диффи-Хеллмана, выбираются некоторое большое простое число P и число A, такие, что различные степени Апредставляют собой раз- личные числа по модулю P. Числа P и A могут передаваться в открытом виде и быть общими для всех абонентов сети.

Затем каждый абонент группы выбирает свое секретное число X_i, $1 < X_i < P-1$, и вычисляет соответствующее ему открытое чис- ло $Y_i : Y_i = A^{X_i} \, mod \, P$. Таким образом, каждый пользователь может сгенерировать закрытый ключ X_i и открытый ключ Y_i.

Информация о необходимых параметрах системы сведена в следующую таблицу.

Шифрование

Теперь рассмотрим, каким образом производится шифрование данных. Сообщение, предназначенное для шифрования, должно быть представлено в виде одного числа или набора чисел, каждое из которых меньше P. Пусть пользователь 1 хочет передать пользователю 2 сообщение m. В этом случае последовательность действий следующая.

1. Первый пользователь выбирает случайное число k, взаимно про- стое с P-1, и вычисляет числа

$$r = A^k \, mod \, P, \qquad e = m \times Y_2^k \, mod \, P$$

где Y_2 – открытый ключ пользователя 2. Число k держится в секрете.

2. Пара чисел (r, e), являющаяся шифротекстом, передается второму пользователю.

3. Второй пользователь, получив (r,e), для расшифрования сообщения вычисляет

$$m = e \times r^{P-1-X_2} \, mod \, P$$

где X_2 – закрытый ключ пользователя 2. В результате он получает исходное сообщение m.

Если злоумышленник узнает или перехватит P, A, Y_2, r, e, то он не сможет по ним раскрыть m. Это связано с тем, что противник не знает параметр k, выбранный первым пользователем для шифрования сообщения m. Вычислить каким-либо образом число k практически невозможно, так как это задача дискретного логарифмирования. Следовательно, злоумышленник не может вы-

числить и значение m, так как m было умножено на неизвестное ему число. Противник также не может воспроизвести действия законного получателя сообщения (второго абонента), так как ему не известен закрытый ключ X_2 (вычисление X_2 на основании Y_2 — также задача дискретного логарифмирования).

По аналогичному алгоритму может производиться и согласование ключа, используемого для симметричного шифрования больших объемов данных. Более того, алгоритм Эль-Гамаля на практике целесообразно использовать именно для согласования общего *ключа сессии*, а не прямого шифрования больших сообщений. Это связано с тем, что в алгоритме используются опера- ции возведения в степень и умножения по большому модулю. Так же как и в алгоритмах RSA и Диффи-Хеллмана, операции производятся над большими, состоящими из нескольких сотен или тысяч бит, числами. Поэтому шифрование больших сообщений производится крайне медленно.

Контрольные вопросы ПЗ6 (ПК-2):

1. Что такое Асимметричные алгоритмы шифрования.
2. Что такое Алгоритмы шифрования с открытым ключом?
3. Что такое Алгоритм шифрования с открытым ключом RSA?
4. Что такое согласование ключа?
5. Опишите алгоритм алгоритм RSA
6. Опишите алгоритм Диффи-Хеллмана