In[1]:=

In[2]:=
```
(*Лабораторная работа №3*)
(*По курсу «Защита информационных процессов в компьютерных системах»*)
(*ИСокрытие информации в текстовом документе методом изменения формата текста.*)

(*
    Кутузов Илья
      А-12м-20
*)
nnom = 10;
(*Задание 1*)
(*Подготовить незаполненный (пустой) контейнер cont в виде текстового файла,
взяв за основу файл plaintext.doc и начиная со страницы Ncгр
 (номер по списку в группе) скопировать текст без рисунков приблизительно
 3 Кб в блокнот Windows,а затем сохранить его в кодировке ANSI.*)
```

In[3]:=
```
(*Задание 2*)
(*Импортировать текстовый файл в свой ноутбук (документ) Mathematica,
следующим образом:plntext=
 FromCharacterCode[Import["File Path...","Byte"],"WindowsCyrillic"],
 символ по его коду   импорт  Файл путь   байт
указав путь к выбранному файлу в меню Insert//File Path//.*)
                                          вписать  Файл путь
plntext = FromCharacterCode[
          символ по его коду
   Import["D:\\GitHub Repos\\stud\\mag\\Sem9\\ЦТЗИ\\Lab3\\plain.txt", "Byte"],
   импорт  дифференциировать                                    байт
   "WindowsCyrillic"];
```

In[4]:=
```
(*Задание 3*)
(*Определить общее число возможных точек встраивания-
 подсчитать количество символов "а","к","х","е","р","о",
 "с" в контейнере,а также подсчитать полный размер контейнера*)
letters = {"а", "к", "х", "е", "р", "о", "с"};
s = {};
For[i = 0, i < Length[letters], i++;
 AppendTo[s, {letters[[i]], StringCount[plntext, letters[[i]]]}]
]
s // Grid
Sum[s[[i]][[2]], {i, Length[s]}]
```

Out[7]=
```
а  229
к  89
х  23
е  242
р  147
о  270
с  120
```

Out[8]=
```
1120
```

In[9]:=
```
(*Задание 4*)
(*.Сформировать стегопуть-
 ключ (key) встраивания информации в виде случайной двоичной последовательности
  длиной равной числу символов с одинаковым начертанием (см.п.3)*)
SeedRandom[nnom]
keypath = RandomInteger[{0, 1}, StringLength[plntext]];
```

In[11]:=
```
(*Задание 5*)
(*Подготовить двоичный эквивалент для встраиваемого
 сообщения:прилетаю Ncгр (например для Ncгр=21-прилетаю двадцать первого).Это
   сообщение ввести непосредственно в ноутбуке и,
последовательно применяя функции ToCharacterCode[],IntegerDigits[],Flatten[],
                                  код символа      цифры целого числа  уплостить
получить двоичную строку (cod).Обратить внимание на необходимость выравнивания
 данных до 11 разрядов (PadLeft[]).Определить длину строки cod:r.*)
                          заполнить слева
nnom
mes = "прилетаю десятого";
ToCharacterCode[mes]
код символа
Grid[mesc = IntegerDigits[ToCharacterCode[mes], 2, 11]]
таблица   цифры целого ч…  код символа
cod = Flatten[mesc]
      уплостить
Length[cod]
длина
```

Out[11]= 10

Out[13]= {1087, 1088, 1080, 1083, 1077, 1090, 1072,
 1102, 32, 1076, 1077, 1089, 1103, 1090, 1086, 1075, 1086}

Out[14]=
```
1  0  0  0  0  1  1  1  1  1  1
1  0  0  0  1  0  0  0  0  0  0
1  0  0  0  0  1  1  1  0  0  0
1  0  0  0  0  1  1  1  0  1  1
1  0  0  0  0  1  1  0  1  0  1
1  0  0  0  1  0  0  0  0  1  0
1  0  0  0  0  1  1  0  0  0  0
1  0  0  0  1  0  0  1  1  1  0
0  0  0  0  0  1  0  0  0  0  0
1  0  0  0  0  1  1  0  1  0  0
1  0  0  0  0  1  1  0  1  0  1
1  0  0  0  1  0  0  0  0  0  1
1  0  0  0  1  0  0  1  1  1  1
1  0  0  0  1  0  0  0  0  1  0
1  0  0  0  0  1  1  1  1  1  0
1  0  0  0  0  1  1  0  0  1  1
1  0  0  0  0  1  1  1  1  1  0
```

Out[15]= {1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0,
 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0,
 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0,
 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1,
 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0}

Out[16]= 187

In[17]:=
```
(*Задание 6*)
(*Проверить правильность формирования строки cod-
 восстановить из двоичной строки текст встраиваемого сообщения.*)
x = {};
Do[AppendTo[x, FromCharacterCode[FromDigits[Partition[cod, 11][[i]], 2]]],
   {i, Length[cod] / 11}];
StringJoin[
 x]
```

Out[19]=
```
прилетаю десятого
```

In[20]:=
```
(*Задание 7*)
(*Разработать модуль встраивания на базе
 оператора Which[].Встраивание проводить посимвольно,
в соответствии со следующим правилом:если текущий символ symb[i]∈Lkyr=
 {"а","к","х","е","р","о","с"},и key[i]=1,то при cod[imod r]=
 0 заменить символ кириллицы на такой же по начертанию символ латиницы.*)
```

In[21]:=
```
(*Задание 8*)
(*Отлаженный блок встраивания оформить в виде функции пользователя
   на основе оператора Module[]:входные параметры-пустой контейнер cont,
                              программный модуль
 стартовое значение для формирования ключа,сообщение для встраивания;
 выходной параметр-заполненный контейнер.*)
Steger[container0_, numstart0_, mes0_] := Module[{container = container0,
                                                   программный модуль
   numstart = numstart0, msg = mes0, outpt, key, msgbin, lets, j},
  SeedRandom[numstart];
  инициализация генератора псевдослучайных чисел
  key = RandomInteger[{0, 1}, StringLength[container]];
        случайное целое число      длина строки
  lets = {"a", "к", "x", "e", "p", "o", "c"};
  msgbin = Flatten[IntegerDigits[ToCharacterCode[msg], 2, 11]];
           уплостить цифры целого ч… код символа
  outpt = Characters[container];
          символы
  j = 1;
  For[i = 1, i <= StringLength[container], i++,
  цикл ДЛЯ              длина строки
   If[(MemberQ[lets, outpt[[i]]]) && (key[[i]] == 1),
       элемент списка?
     If[msgbin[[Mod[j, Length[msgbin], 1]]] == 0,
     условный оп… остато… длина
      Which[
      условный оператор с множественными ветвями
        outpt[[i]] == "a", outpt[[i]] = "a",
        outpt[[i]] == "к", outpt[[i]] = "k",
        outpt[[i]] == "x", outpt[[i]] = "x",
        outpt[[i]] == "e", outpt[[i]] = "e",
        outpt[[i]] == "p", outpt[[i]] = "p",
        outpt[[i]] == "o", outpt[[i]] = "o",
        outpt[[i]] == "c", outpt[[i]] = "c"
       ];
    ]; j++
   ];
  ];
  StringJoin[outpt]
  соединить строки
 ]
```

In[22]:=
```
(*Задание 9*)
(*Провести операцию встраивания сообщения из п.5.*)
inserted = Steger[plntext, nnom, mes];
```

In[23]:=
```
(*Задание 10*)
(*Разработать модуль восстановления скрытого сообщения:входные параметры-
 заполненный контейнер и стартовое значение для формирования ключа;
 выходной параметр-восстановленное скрытое сообщение.*)
InSteger[container0_, numstart0_] := Module[
                                     программный модуль
  {container = container0, numstart = numstart0, tmp, letsru, letsen, key, outpt, x},
  letsru = {"а", "к", "х", "е", "р", "о", "с"};
  letsen = {"a", "k", "x", "e", "p", "o", "c"};
  SeedRandom[numstart];
  инициализация генератора псевдослучайных чисел
  key = RandomInteger[{0, 1}, StringLength[container]];
        случайное целое число      длина строки
  tmp = Characters[container];
        символы
  outpt = {};
  For[i = 1, i <= StringLength[container], i++,
  цикл ДЛЯ          длина строки
   If[key[[i]] == 1,
   условный оператор
    If[MemberQ[letsru, tmp[[i]]], AppendTo[outpt, 1]];
    у… элемент списка?          добавить в конец к
    If[MemberQ[letsen, tmp[[i]]], AppendTo[outpt, 0]]
    у… элемент списка?          добавить в конец к
   ]
  ];
  (*FromDigits[Transpose[Partition[outpt,11]],2]*)
    число по ря… транспози… разбиение на блоки
  FromCharacterCode[FromDigits[Transpose[Partition[outpt, 11]], 2]]
  символ по его коду      число по ря… транспози… разбиение на блоки
 ]
```

In[24]:=
```
(*Задание 11*)
(*Восстановить скрытое сообщение*)
restored = InSteger[inserted, nnom]
```

Out[24]= прилетаю десятогоприлетаю десятогоприлетаю десятого

In[25]:=
```
(*Задание 12*)
(*Провести анализ различия пустого и заполненного контейнеров,
сравнив файлы в прогамме WinHex.*)
q = ToCharacterCode[plntext] - ToCharacterCode[inserted];
    код символа                код символа
x = {};
Do[If[q[[i]] ≠ 0, AppendTo[x, i]], {i, Length[q]}]
   условный оператор добавить в конец к  длина
x
```

Out[28]=
```
{7, 17, 23, 29, 72, 74, 78, 90, 91, 92, 98, 107, 110, 116, 122, 131, 132, 167, 168, 182,
 195, 202, 210, 211, 220, 252, 254, 255, 260, 283, 288, 306, 308, 324, 330, 338, 341,
 354, 365, 384, 394, 395, 404, 431, 432, 435, 444, 456, 470, 478, 496, 499, 509, 513,
 518, 528, 556, 558, 566, 570, 575, 578, 581, 592, 594, 608, 610, 632, 641, 642, 650,
 656, 658, 663, 685, 701, 709, 715, 718, 726, 730, 735, 743, 744, 763, 768, 774, 785,
 788, 828, 830, 846, 874, 876, 878, 885, 898, 913, 921, 924, 927, 947, 952, 970,
 972, 984, 1003, 1012, 1026, 1035, 1046, 1057, 1092, 1111, 1114, 1131, 1132, 1177,
 1180, 1183, 1203, 1206, 1218, 1227, 1228, 1230, 1253, 1260, 1261, 1271, 1300, 1302,
 1328, 1347, 1355, 1356, 1358, 1388, 1401, 1410, 1413, 1415, 1423, 1432, 1449, 1450,
 1455, 1458, 1463, 1471, 1473, 1499, 1504, 1519, 1525, 1528, 1565, 1568, 1573, 1576,
 1580, 1582, 1594, 1602, 1606, 1655, 1663, 1673, 1679, 1696, 1699, 1705, 1713, 1716,
 1718, 1719, 1746, 1747, 1749, 1752, 1786, 1799, 1801, 1810, 1828, 1835, 1839, 1851,
 1882, 1912, 1925, 1926, 1938, 1945, 1948, 1951, 1959, 1968, 1971, 1982, 2011, 2038,
 2074, 2075, 2079, 2087, 2089, 2092, 2097, 2105, 2110, 2117, 2121, 2125, 2154, 2166,
 2167, 2176, 2185, 2195, 2202, 2217, 2220, 2222, 2225, 2248, 2262, 2267, 2278, 2283,
 2338, 2342, 2346, 2354, 2357, 2365, 2386, 2401, 2402, 2418, 2421, 2429, 2438, 2468,
 2482, 2484, 2497, 2503, 2513, 2524, 2547, 2565, 2566, 2583, 2584, 2599, 2614, 2639,
 2646, 2652, 2657, 2659, 2663, 2664, 2692, 2695, 2706, 2710, 2712, 2751, 2757, 2766,
 2784, 2788, 2792, 2802, 2812, 2816, 2840, 2842, 2846, 2860, 2863, 2867, 2883, 2884,
 2888, 2908, 2909, 2930, 2937, 2952, 2956, 2981, 2991, 2999, 3001, 3005, 3012, 3016,
 3035, 3045, 3058, 3060, 3068, 3071, 3076, 3089, 3092, 3095, 3107, 3114, 3116, 3122,
 3128, 3165, 3168, 3174, 3187, 3191, 3192, 3195, 3209, 3218, 3231, 3235, 3242, 3271,
 3282, 3291, 3293, 3296, 3318, 3321, 3355, 3362, 3384, 3385, 3415, 3432, 3434}
```

In[29]:=
```
(*Задание 13*)
(*Провести частичное "стирание" скрытого сообщения путем замены (StringReplace[])
                                                          заменить в строке
 символов латиницы на символы кириллицы для половины заполненого контейнера*)
```

In[30]:=
```
damaged = inserted;
rule = {"a" → "а", "k" → "к", "x" → "х", "e" → "е", "p" → "р", "o" → "о", "c" → "с"};
damaged = StringReplace[damaged, rule, Round[Length[x] / 2]]
          заменить в строке              окру··· длина
```

Out[32]=
Стоимость POS-терминалов в зависимости от комплектации и возможностей может меняться от нескольких сотен до нескольких тысяч долларов, хотя обычно не превышает полутора-двух тысяч долларов. Размеры и вес POS-терминала сопостави-мы с аналогичными параметрами телефонного аппарата.

Схема системы POS приведена на рис. 9.4. Покупатель для оплаты покупки предъявляет свою дебетовую или кредитную карту и вводит значение PIN для подтверждения личности. Продавец, в свою очередь, вводит сумму денег, которую необходимо уплатить за покупку или услуги. Затем в банк-эквайер (банк продавца)

направляется запрос на перевод денег. Банк-эквайер переадресует этот запрос в банк-эмитент для проверки подлинности карты, предъявленной покупателем. Если эта карта подлинная и покупатель имеет право применять ее для оплаты продуктов и услуг, банк-эмитент пере-водит деньги в банк-эквайер на счет продавца. После перевода денег на счет продавца банк-эквайер посылает на POS-терминал извещение, в котором сообщает о завершении транзакции. После этого продавец выдает покупателю товар и извещение.

Следует обратить внимание на тот сложный путь, который должна проделать информация о покупке, прежде чем будет осуществлена транзакция. Во время прохождения этого пути возможны искажения и потеря сообщений.

Для защиты системы POS должны выполняться следующие требования.

Проверка PIN, введенного покупателем, должна производиться системой банка-эмитента. При пересылке по каналам связи значение PIN должно быть зашифровано.

Сообщения, содержащие запрос на перевод денег (или подтверждение о переводе), должны проверяться на подлинность для защиты от замены и внесения изменений при прохождении по линиям связи и обрабатывающим процессорам [22].

Самым уязвимым местом системы POS являются ее POS-терминалы. В отличие от банкома-тов в этом случае изначально предполагается, что POS-терминал не защищен от внешних воздействий. Угрозы для POS-терминала связаны с возможностью раскрытия секретного ключа, который находится в POS-терминале и служит для шифрования информации, передаваемой этим терминалом в банk-эквайер. Угроза раckрытия kлюча терминала достаточно реальна, так как эти терминалы устанавливаются в таkих неохраняемых местах, kak магазины, автозаправочные стан-ции и пр.

Потенциальные угрозы из-за раскрытия ключа получили такие названия.

"Обратное трассирование". Сущность этой угрозы состоит в том, что если злоумышленниk полу-чит kлюч шифрования, то он может пытаться восстановить значения PIN, использованные в пре-дыдущих транзакциях.

"Прямое трассирование". Сущность этой угрозы состоит в том, что если злоумышленник получит ключ шифрования, то он попытается восстановить значения PIN, которые будут использоваться в последующих транзакциях.

Для защиты от угроз обратного и прямого трассирования предложены три метода:

метод выведенного kлюча;

метод ключа транзакции;

метод открытых ключей [22].

Сущность первых двух методов состоит в том, что они обеспечивают модификацию kлюча шифрования передаваемых данных для каждой транзаkции.

Метод выведенного ключа обеспечивает смену ключа при каждой транзакции независимо от ее содержания. Для генерации ключа шифрования используют однонаправленную функцию от теку-щего значения ключа и некоторой случайной величины. Процесс получения (вывода) ключа для шиф-рования очередной транзакции представляет собой известное "блуждание" по дереву (рис. 9.5).

Вершиной дерева рис. 9.5 являетс

In[33]:=
```
(*Задание 14*)
(*Выполнить процедуру восстановления скрытого текста после стирания.*)
InSteger[damaged, nnom]
```

Out[33]= ▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪,десятогоприлетаю десятого