Министерство образования Республики Беларусь

Учреждение образования

«Полоцкий государственный университет»

Д.Ф. Пастухов, Ю.Ф. Пастухов, А.Д. Сонич, А.Ю.Ченторицкий

ОБОБЩЕНИЕ МЕТОДА ШИФРОВАНИЯ ВЛАДИМИРА СИЗОВА НА СЛУЧАЙ ПРОИЗВОЛЬНЫХ ПЕРИОДИЧЕСКИХ И АПЕРИОДИЧЕСКИХ ФУНКЦИЙ.

Учебное пособие к лекционным и практическим занятиям
для студентов специальности

1-98 01 01 Компьютерная безопасность

Новополоцк
ПГУ
2019

УДК 519.72

Рецензенты:
А.А. Козлов, кандидат физико-математических наук, доцент, заведующий кафедрой Высшей математики и дифференциальных уравнений Полоцкого государственного университета

Пастухов Д.Ф., Пастухов Ю.Ф.,Сонич А.Д., Ченторицкий А.Ю.
Обобщение метода шифрования Владимира Сизова на случай произвольных периодических и апериодических функций: учебное пособие /Д.Ф. Пастухов, Ю.Ф. Пастухов, Сонич А.Д., Ченторицкий А.Ю.- Новополоцк: ПГУ,2019. - 15 с.

В учебном пособии исследовано обобщение метода шифрования Владимира Сизова на случай произвольных периодических и апериодических функций. Для студентов программистов 98 01 01 (компьютерная безопасность).

Для студентов университетов, педагогических вузов, технических вузов, преподавателей, инженеров, программистов использующих в своей практической деятельности математические методы шифрования.

Одобрено и рекомендовано к изданию
методической комиссией факультета информационных технологий
В качестве учебного пособия

Кафедра технологий программирования

Предисловие

Предлагаемое учебное пособие создано на основе работы авторов со студентами - дипломниками 4 курса специальности Компьютерная безопасность 98 01 01 (математические методы и программные системы) в Полоцком государственном университете.

Метод шифрования текста функцией косинуса Владимиром Сизовым был представлен на Российской научной конференции РусКрипто в 2005 году.

Алгоритм Сизова использовал одновременно функции действительной и целой переменных и групповые свойства остатков по модулю p. Где p – мощность алфавита клавиатуры. Ключами шифрования у Сизова являлись начальная фаза и момент времени в уравнении бегущей волны аргумента функции косинуса.

Нами исследован модифицированный алгоритм Владимира Сизова с произвольным числом ключей для конечного числа слагаемых тригонометрического ряда с аналогичными аргументами бегущей волны, дополнительно использующими частотную модуляцию. То есть аргумент каждого слагаемого получает дополнительный ключ-частотный сдвиг (аналог частотной модуляции в радиотехнике). Тогда для трех слагаемых ряда Фурье, пользователь имеет в распоряжении 9 действительных чисел - ключей для шифрования. Приведен также пример шифрования с помощью неограниченной периодической функции.

Исследован также модифицированный метод Владимира Сизова шифрования текста с помощью конечного числа слагаемых дробно-степенных функций. Идея была заимствована из раздела математики - уравнений математической физики, в которой особые решения обыкновенных дифференциальных уравнений второго порядка получают с помощью обобщенного степенного ряда Фробениуса с дробными показателями степени.

Корректность математической постановки задачи модифицированного метода обсуждался с заведующим кафедрой Высшей математики, кандидатом физико-математических наук, доцентом, Александром Александровичем Козловым.

Представлены коды трех примеров шифрования модифицированными алгоритмами на языке C++. Для краткости комментарии к программам чередуются с частями кода, поясняя их действия сразу после кода.

Пастухов Д.Ф., Пастухов Ю.Ф.

Приведем выдержки из работ и статей Владимира Сизова и Каминского С.П., Степанова В.А. [1-3]:

Алгоритм шифрования

По координатной оси X расставляются компьютерные символы в любом порядке. Каждому символу соответствует свой порядковый номер от 1 до 256. Всего в компьютере используется 256 символов. По оси Y расставляем те же самые символы в любом (таком же или другом) порядке. Им также присваивают порядковые номера от 1 до 256. Функция, посимвольно переводящая исходный текст в шифротекст:

$$Y = X + 256 \cdot (\cos(Z + N \cdot \Delta x)) mod 256 \quad (1)$$

где X – порядковый номер того символа, который нужно зашифровать; Z, Δx – любые числа, являющиеся секретными параметрами нашего ключа. Остальные параметры не являются секретными. Z, Δx ∈ (–∞, +∞); N – номер по счету шифруемого символа в исходном тексте; 256 – мощность исходного алфавита. Мощность исходного алфавита может быть любой. (В нашем случае мощность равна 256, как количество символов в расширенной таблице ASCII.)

Алгоритм дешифрования

Тригонометрический шифр является примером симметричного алгоритма шифрования, следовательно:

$$X = Y - 256 \cdot (\cos(Z + N \cdot \Delta x)) mod 256 \quad (2)$$

Модифицированные алгоритмы

Модифицируем формулы(1),(2) на случай произвольной периодической функции.

$$y(i) = \left[x + p\left(\sum_{k=1}^{n} \left(a_k \cos\left((i + w_{1k}) t_{1k} + \varphi_{1k} \right) + b_k \sin\left((i + w_{2k}) t_{2k} + \varphi_{2k} \right) \right) \right) \right] \bmod(p) \quad (3)$$

$$x(i) = \left[y - p\left(\sum_{k=1}^{n} \left(a_k \cos\left((i + w_{1k}) t_{1k} + \varphi_{1k} \right) + b_i \sin\left((i + w_{2k}) t_{2k} + \varphi_{2k} \right) \right) \right) \right] \bmod(p) \quad (4)$$

Формулы (3), (4) мы назвали модифицированными формулами шифрования и дешифрования Владимира Сизова обобщенными тригонометрическими многочленами.

Где: a_k, b_k – амплитуды гармоник,i – номер символа исходного сообщения, w_{1k}, w_{2k} – частоты модуляции порядкового номера символа в сообщении (в отличие от слагаемых тригонометрического ряда частоты модуляции не соотносятся как числа натурального

ряда и различаются для функций синуса и косинуса), t_{1k}, t_{2k} – моменты времени, $\varphi_{1k}, \varphi_{2k}$ – начальные фазы. Отметим, что суммы в формулах (3),(4) умножаются на мощность алфавита p, чтобы отображение множества символов клавиатуры в то же множество происходило случайным образом и принимало значения всех символов, даже если мы шифруем достаточно длинную строку из одного фиксированного символа (используется, если амплитуды во много раз меньше мощности алфавита $a_k, b_k << p$). Все параметры сумм(3),(4) кроме номера i являются электронными ключами при шифровании. Переменные x,y – целые остатки по модулю p. Квадратные скобки в формулах(3),(4) – взятие целой части от действительного числа.

Приведем пример шифрования с использованием 2 слагаемых обобщенного тригонометрического ряда (для клавиатуры ASCII p=256).

$$f(i) = p(\sin((i+w_1)t_1+\varphi_1)*\cos((i+w_1)t_1+\varphi_1)+\sin(3((i+w_2)t_2+\varphi_2))) \quad (5)$$

Код программы шифрования-дешифрования конечным тригонометрическим рядом ограниченной функции. Комментарии к коду приведены сразу после части кода.

```
#include<stdio.h>
#include<math.h>
int main()
{
int const m=31,p=255,w01=3,t1=10,fi1=7,w02=30,fi2=8,t2=15;
//частоты модуляции, моменты времени, начальные фазы- целые числа
  int i,j,p0,d,s0,s4,s44,s5,s6,w1,w2,baza[m+1],baza3[m+1],baza1[m+1];
double s1,s2,s3, baza2[m+1];
char str[m+2]="Polotsk 2019 Chentorickij Alexej";
p0=8;
 for(i=0;i<=m;i++)
  {
  baza[i]=str[i];
// по кодировке ASCII номер символа из массива str[i] вводится в целый массив
baza[i]
      printf("sim(%d)=%c(%d)\n",i,str[i],str[i]);
      baza2[i]=double(baza[i])+0.5;
//согласно алгоритму В.Сизова целая переменная x переводится в действительную
//переменную – в середину единичного отрезка  x → x + 0.5
      printf("baza(%d)=%lf\n",i,baza2[i]);
  }
for(i=0;i<=m;i++)
{
w1=i+w01;
s4=w1*t1+fi1;
//целый аргумент первой бегущей волны
w2=i+w02;
s44=w2*t2+fi2;
//целый аргумент второй бегущей волны
s1=double(p)*(sin(double(s4))*cos(double(s4))+sin(double(3*s44)));
//функция шифрования согласно формуле(5)
s3=s1+baza2[i];
//шифрование символа по формуле(3) в переменных с двойной точностью
baza1[i]=int(s3)%p;
//шифрование символа по формуле(3) в целый массив baza1[i]
```

```
if(baza1[i]<0)
{
baza1[i] = baza1[i]+p-1;
}
//измененная нами часть алгоритма, если остаток по модулю p baza1[i]меньше нуля
//( номер символа целочисленного шифра), то прибавляем не p, а p-1.
printf("slovo(%d)=%d,shifr(%d)=%d\n",i,str[i],i,baza1[i]);
}
///deshifrovanie
printf("dechifrovanie\n");
for(i=0;i<=m;i++)
{
s2=double(baza1[i])+0.5;
// приведение номера шифрованного символа в действительное число по В.Сизову
w1=i+w01;
s4=w1*t1+fi1;
//целый аргумент первой бегущей волны
w2=i+w02;
s44=w2*t2+fi2;
//целый аргумент второй бегущей волны
s1=double(p)*(sin(double(s4))*cos(double(s4))+sin(double(3*s44)));
//функция шифрования согласно формуле(5)
s3=s2-s1;
// дешифрование по формуле(4) в действительных переменных
baza3[i]=int(s3)%p;
// дешифрование по формуле(4) в целый массив baza3[i]
if(baza3[i]<0)
{
baza3[i]=baza3[i]+p-1;
      }
//измененная нами часть алгоритма, если остаток по модулю p меньше нуля
//(номера дешифрованного символа), то прибавляем не p, а p-1.

      printf("slovo(%d)=%d,deshifr(%d)=%d\n",i,str[i],i,baza3[i]);
}
for(i=0;i<=m;i++)
{
printf("deshifr(%d)=%c\n",i,baza3[i]);
}
//перевод номера дешифрованного символа из массива baza3[i] в сам символ
}
```

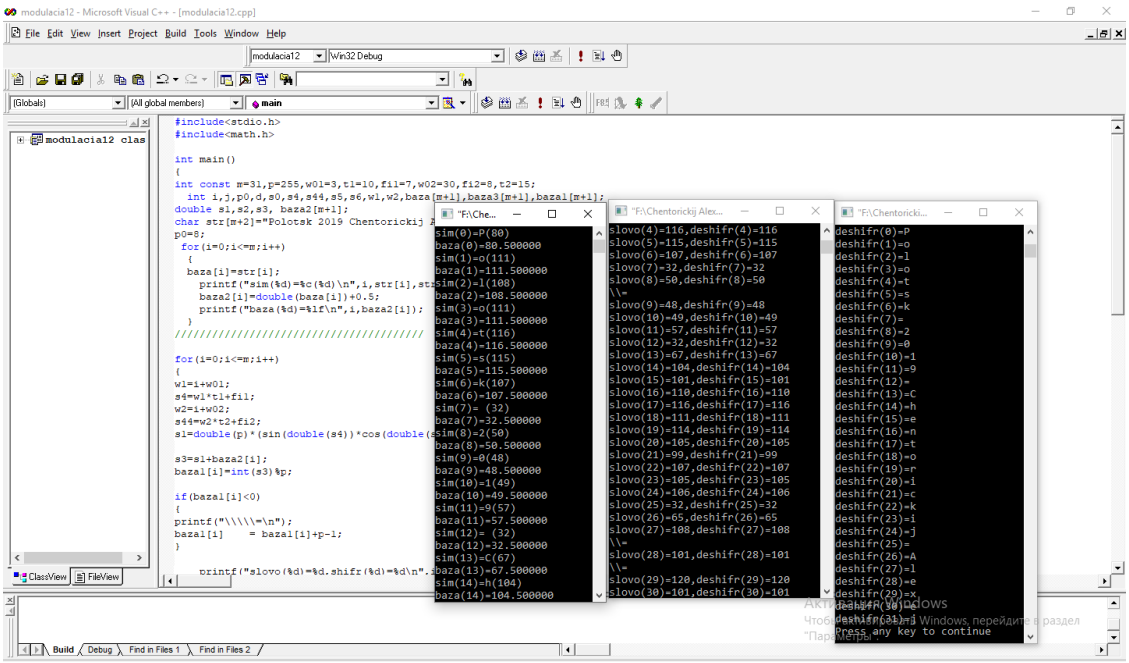

Рис.1. Пример шифрования - дешифрования фразы "Polotsk 2019 Chentorickij Alexej" согласно функции шифрования(5).

Из Рис.1 видно, что символы, а также номера символов исходной и дешифрованной фраз совпадают. В данном примере все ключи - 2 несущих частоты, 2 момента времени, 2 начальных фазы являются целочисленными переменными.

Пример 2. Рассмотрим шифрование периодической неограниченной функцией вида(6)

$$\begin{cases} y = [x + tg((i + w)t + \varphi) + ctg((i + w)t + \varphi)] \bmod(p) \\ x = [y - tg((i + w)t + \varphi) + ctg((i + w)t + \varphi)] \bmod(p) \end{cases} \quad (6)$$

Код программы шифрования-дешифрования конечным тригонометрическим рядом неограниченной функции согласно формуле(6)

```
#include<stdio.h>
#include<math.h>
int main()
{
int const m=40,p=255,w0=300,t=10,fi=7,a=255;
//частоты модуляции, моменты времени, начальные фазы- целые числа
  int i,j,p0,d,s0,b[m+1],s4,s5,s6,w,baza[m+1],baza3[m+1],baza1[m+1];
double s1,s2,s3, baza2[m+1];
char str[m+2]="Polotsk 2019 Chentorickij Alexej Yurevich";
p0=8;
 for(i=0;i<=m;i++)
 {
 baza[i]=str[i];
// по ASCII номер символа из массива str[i] вводится в целый массив baza[i]
 printf("sim(%d)=%c(%d)\n",i,str[i],str[i]);
 baza2[i]=double(baza[i])+0.5;
//согласно алгоритму В.Сизова целая переменная x переводится в действительную
//переменную – в середину единичного отрезка  x → x + 0.5
 printf("baza(%d)=%lf\n",i,baza2[i]);
 }
for(i=0;i<=m;i++)
{
w=i;
```

```
s4=(w)*t+fi;
//целый аргумент первой бегущей волны
s1=tan(double((i)*t+fi))+1.0/tan(double((i)*t+fi));
//функция шифрования согласно формуле(6)
s3=s1+baza2[i];
// шифрование в действительных переменных
baza1[i]=int(s3)%p;
// шифрование в целый массив baza1[i]
if( baza1[i]  <0)
{
baza1[i] = baza1[i]+p-1;
}
//измененная нами часть алгоритма, если остаток по модулю p меньше нуля
//( номера символа целочисленного шифра), то прибавляем не p, а p-1.
printf("slovo(%d)=%d,shifr(%d)=%d\n",i,str[i],i,baza1[i]);
}
///deshifrovani
printf("dechifrovanie\n");
for(i=0;i<=m;i++)
{
s4=baza1[i];
s1=tan(double((i)*t+fi))+1.0/tan(double((i)*t+fi));
//функция шифрования согласно формуле(6)
s2=double(s4)+0.5;
// перевод шифра из целого массива baza1[i] в действительную переменную
s3=s2-s1;
// дешифрование в действительных переменных
baza3[i]=int(s3)%p;
// дешифрование в целый массив baza3[i]
if(baza3[i]<0)
{
baza3[i]=baza3[i]+p-1;
      }
//измененная нами часть алгоритма, если остаток по модулю p меньше нуля
//(номера дешифрованного символа), то прибавляем не p, а p-1.
printf("slovo(%d)=%d,deshifr(%d)=%d\n",i,str[i],i,baza3[i]);
}
for(i=0;i<=m;i++)
{
printf("deshifr(%d)=%c\n",i,baza3[i]);
}
//перевод номера дешифрованного символа из массива baza3[i] в сам символ
}
```

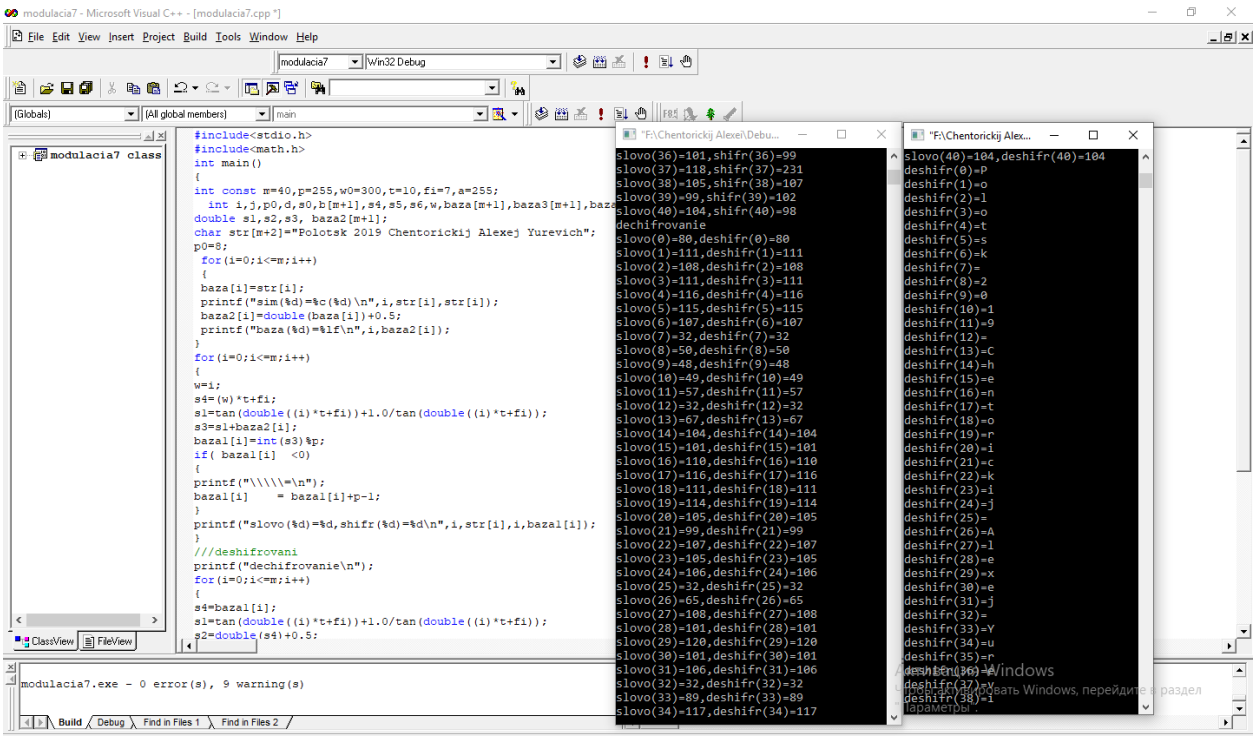

Рис 2. шифрование периодической неограниченной функцией вида(6)

Из рисунка 2 видно, что порядковые номера символов и сами символы исходной фразы "Polotsk 2019 Chentorickij Alexej" шифруются и дешифруются взаимно однозначно. В отличие от формулы (5) в формуле(6) отсутствует множитель p, так как функции тангенса, котангенса, их сумма являются неограниченной функцией. Здесь также ключи начальной фазы, время, несущей частоты - целые числа.

Рассмотрим обобщение алгоритма Владимира Сизова на случай шифрования конечным числом слагаемых обобщенного степенного ряда – рядом Фробениуса. Идея функции шифрования была заимствована из уравнений математической физики[4] для представления особых решений ОДУ второго порядка.

$$f(x) = x^{\nu}\left(a_0 + a_1 x + \ldots + a_n x^n + \ldots\right)$$

$$\begin{cases} y(i) = \left[x + \sum_{k=1}^{n} a_k \left((i + w_k) t_k + \varphi_k \right)^{\beta_k} \right] \bmod(p) \\ x = \left[y - \sum_{k=1}^{n} a_k \left((i + w_k) t_k + \varphi_k \right)^{\beta_k} \right] \bmod(p) \end{cases} \quad (7)$$

Где: x, y – номера исходного и шифрованного символов, a_k - амплитуды слагаемых ряда(7),I – порядковый номер сообщения, β_k - дробные показатели степени, w_k - частоты сдвигов порядкового номера символа i, t_k, φ_k – моменты времени и начальные фазы. В расчете на каждое слагаемое ряда(7)мы имеем 5 электронных ключей – действительных чисел.

Рассмотрим третий пример, в котором у функции шифрования три слагаемых(7), описываемой формулой(8).

$$\begin{cases} y = \left[x + a_1 \left((i + w_1) t_1 + \varphi_1 \right)^{\beta_1} + a_2 \left((i + w_2) t_2 + \varphi_2 \right)^{\beta_2} + a_3 \left((i + w_3) t_3 + \varphi_3 \right)^{\beta_3} \right] \bmod(p) \\ x = \left[y - a_1 \left((i + w_1) t_1 + \varphi_1 \right)^{\beta_1} + a_2 \left((i + w_2) t_2 + \varphi_2 \right)^{\beta_2} + a_3 \left((i + w_3) t_3 + \varphi_3 \right)^{\beta_3} \right] \bmod(p) \end{cases} \quad (8)$$

В которой несущая частота $w_1 = w_2 = w_3 = w$, момент времени $t_1 = t_2 = t_3 = t$, фазы $\varphi_1 = \varphi_2 = \varphi_3 = \varphi$ являются общими для трех слагаемых, амплитуды a_1, a_2, a_3, дробные степени $\beta_1, \beta_2, \beta_3$ являются действительными переменными с двойной точностью.

Код программы шифрования-дешифрования суммой трех слагаемых дробностепенной неограниченной функции приведем согласно формуле(8)

```
#include<stdio.h>
#include<math.h>
int main()
{
int const m=37,p=255,a=255;
  int i,j,p0,d,s0,b[m+1],baza[m+1],baza3[m+1],baza1[m+1];
double s1,s2,s3,s4, baza2[m+1],a1,a2,a3,b1,b2,b3,w,w0,t,fi;
char str[m+2]="Polotsk 2019 Sonich Alexei Dmithievich";
p0=8;
a1=0.3;
a2=0.5;
a3=0.7;
b1=1.0;
b2=2.0;
b3=3.0;
w0=30.0;
t=10.0;
fi=7.0;
  //частоты модуляции, моменты времени, начальные фазы- действительные числа
 for(i=0;i<=m;i++)
 {
  baza[i]=str[i];
  // по ASCII номер символа из массива str[i] вводится в целый массив baza[i]
        printf("sim(%d)=%c(%d)\n",i,str[i],str[i]);
        baza2[i]=double(baza[i])+0.5;
  //согласно алгоритму В.Сизова целая переменная x переводится в действительную
  //переменную – в середину единичного отрезка  x → x + 0.5
   }
for(i=0;i<=m;i++)
{
w=double(i);
s4=(w+w0)*t+fi;
//действительный аргумент первой бегущей волны
s1=b1*pow(double(s4),a1)+b2*pow(double(s4),a2)+b3*pow(double(s4),a3) ;
  // функция шифрования действительной переменной согласно формуле(8)
s3=s1+baza2[i];
  // шифрование в действительных переменных
baza1[i]=int(s3)%p;
  // шифрование в целый массив baza1[i]
if( baza1[i]  <0)
{
printf("\\\\\=\n");
baza1[i]    = baza1[i]+p-1;
}
  //измененная нами часть алгоритма, если остаток по модулю p= baza1[i] меньше
  //нуля (номера дешифрованного символа), то прибавляем не p, а p-1.
```

```
printf("slovo(%d)=%d,shifr(%d)=%d\n",i,str[i],i,baza1[i]);
}
///deshifrovani
printf("dechifrovanie\n");
for(i=0;i<=m;i++)
{
s4=baza1[i];
s1=b1*pow( double((i+w0)*t+fi) ,a1)+b2*pow(  double((i+w0)*t+fi),a2)+b3*pow(double(
(i+w0)*t+fi) ,a3) ;
  // функция шифрования действительной переменной согласно формуле(8)
s2=double(s4)+0.5;
//перевод шифра из целого массива baza1[i] в действительную переменную
s3=s2-s1;
//дешифрование по формуле(8) в действительных переменных
baza3[i]=int(s3)%p;
//дешифрование по формуле(8) в целый массив baza3[i]
if(baza3[i]<0)
{
baza3[i]=baza3[i]+p-1;
}
  //измененная нами часть алгоритма, если остаток по модулю p= baza3[i] меньше
  //нуля (номера дешифрованного символа), то прибавляем не p, а p-1.
printf("slovo(%d)=%d,deshifr(%d)=%d\n",i,str[i],i,baza3[i]);
}
for(i=0;i<=m;i++)
{
printf("deshifr(%d)=%c\n",i,baza3[i]) ;
}}
```

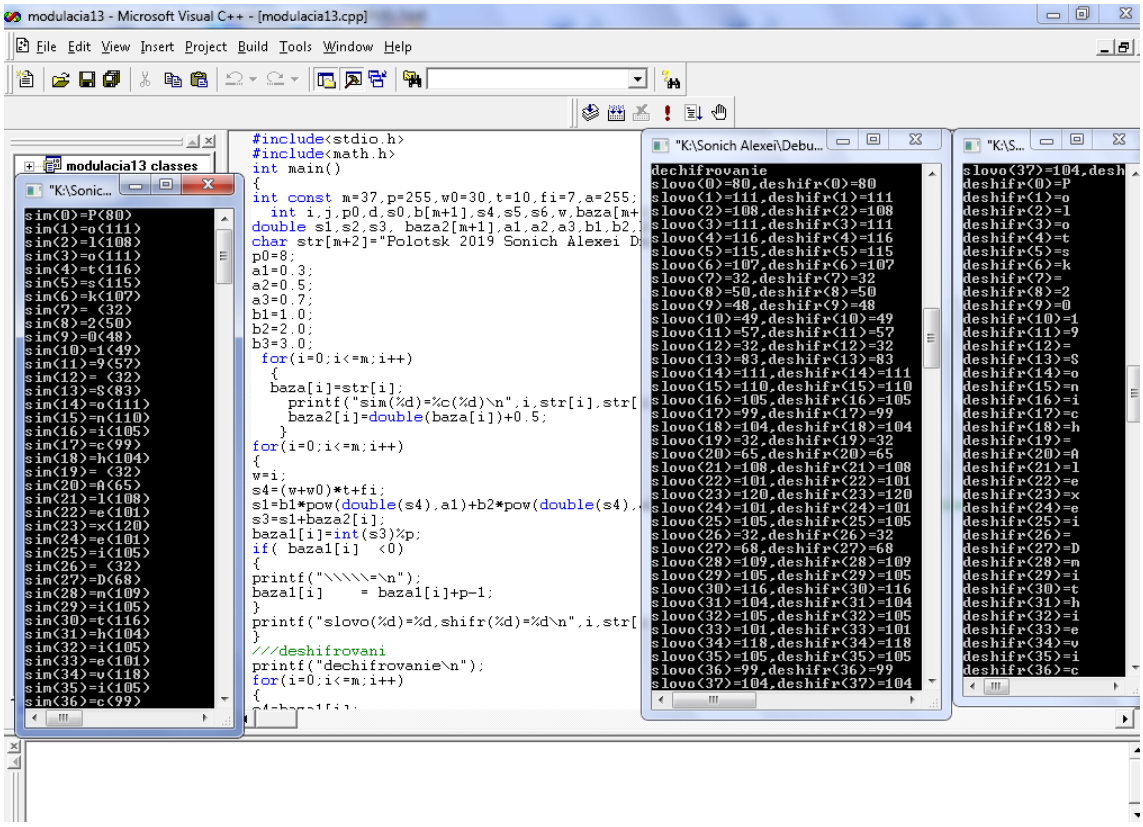

Рис 3. Шифрование суммой дробно степенных функций по формуле (8).

Как видно из Рис.3 номера и значения символов исходной и дешифрованной фраз полностью совпадают.

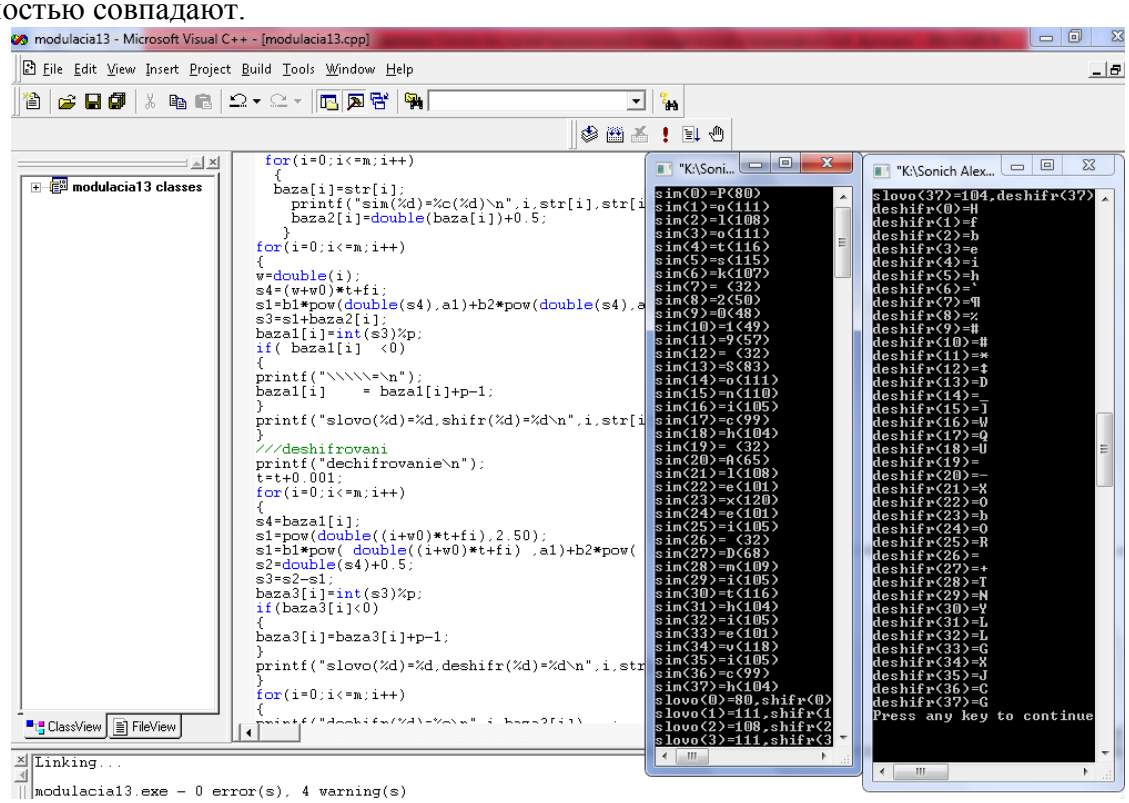

Рис 4. Чувствительность дешифрования суммой дробно степенных функции по формуле (8) с ключами a1=0.3;a2=0.5;a3=1.7;b1=1.0;b2=2.0;b3=3.0;w0=30.0;t=10.0;fi=7.0.

При дешифровании, как видно из рисунка 4, момент времени t изменили на 0.001. При этом ни один символ не был правильно дешифрован (Рис.4).

Сравним число переборов в единичном объёме пространства электронных ключей и прямого перебора слова из 6 символов.

В последней программе используется 9 электронных ключей с чувствительностью 0.001. Число ключей в единичном объеме равно $N_1 = \left(\frac{1}{0.001}\right)^9 = 10^{27}$.

Число перебора слова из 6 независимо расположенных символов клавиатуры ASCII оценим как $N_2 = (256)^6 \approx 2.8 * 10^{14}$, то есть $N_1 / N_2 > 10^{12} >> 1$. Поэтому криптоаналитик имея в своем распоряжении шифрованную фразу из 6 символов имеет возможность разгадать ее прямым перебором быстрее в 10^{12}, чем определить ключи которыми была зашифрована фраза.

Поэтому алгоритмы описываемые формулами (3),(4),(7) имеют высокую криптоустойчивость к простому перебору ключей. Отметим также, что число функций как для конечного тригонометрического ряда(3),(4) так и сумм дробно-степенных функций в формуле (7) бесконечно, что значительно увеличивает возможности шифрования-дешифрования указанными алгоритмами.

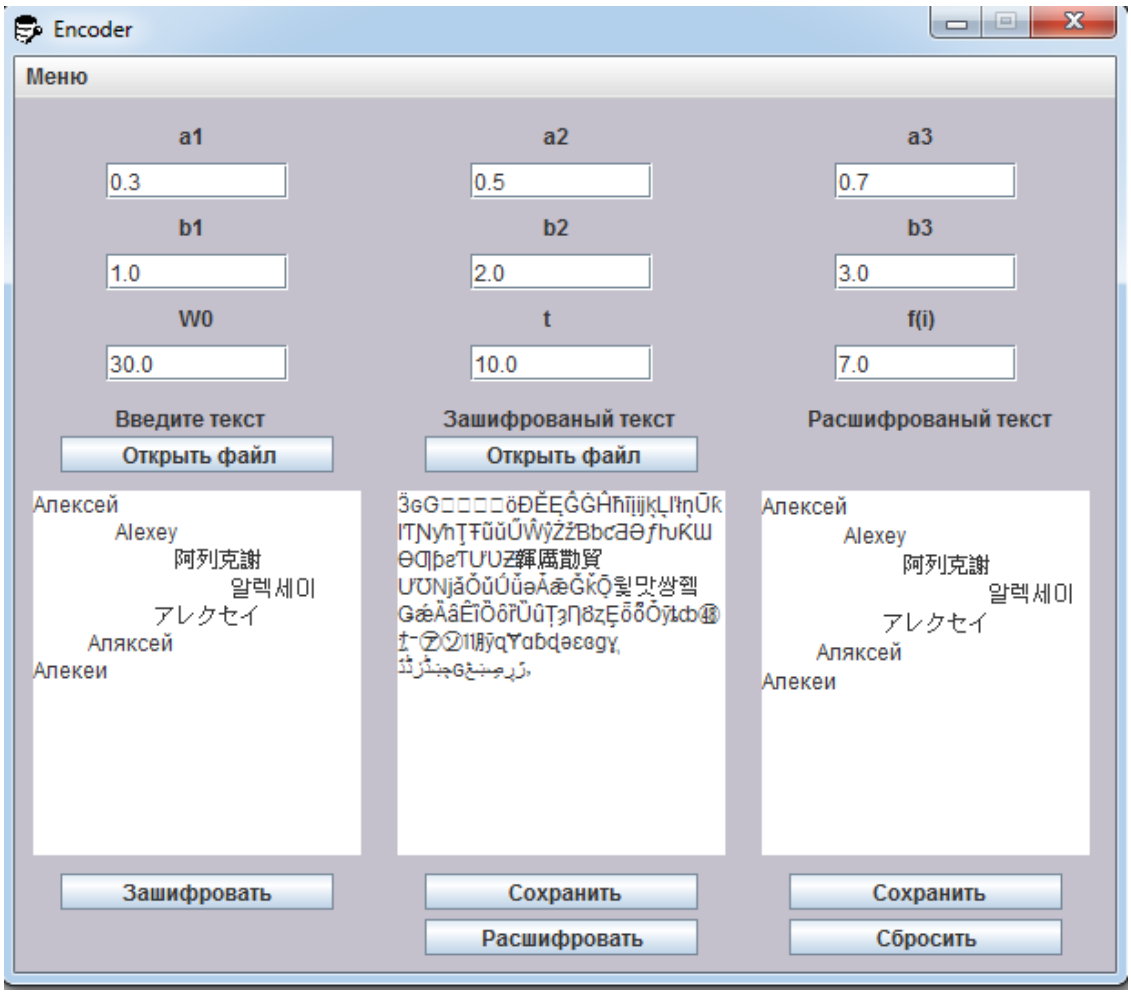

Рис.5. Тестирование шифрования-дешифрования на 7 языках (русском, английском, китайском, корейском, японском, белорусском, греческом) с использованием табуляции и пробелов (мощность алфавита p=10000).
Графический интерфейс программы, использующей формулу(8), принадлежит Сонич А.Д.

Отметим, что модифицированные алгоритмы вида(3),(4),(7) используют нелинейные функции для шифрования текста, как и функции описанные в[5,7]. Однако алгоритмы вида(3),(4),(7) используют дополнительно групповые свойства остатков целых чисел по модулю p. Такие же групповые свойства целых остатков по модулю p использует алгоритм шифрования данных эллиптическими кривыми[6].

Литература

1. Каминский С.П., Степанов В.А. Тригонометрическая криптография. Технологии программирования. №4(1). 2014. С 38-41.
2. Алгоритм защиты информации на основе тригонометрических функций. Диссертация на соискание степени кандидата технических наук. Пермь. 16 с.
3. Сизов В.П. Криптографические алгоритмы на основе тригонометрических функций[HTML](http/www.ruscrypto.ru/sourse/conference/rc2005).
4. Свешников, А.Г. Лекции по математической физике / А.Г. Свешников, А.Н. Боголюбов, В.В. Кравцов. – М.: Изд-во МГУ, 1993 – 332 с.
5. Пастухов Д.Ф.,Пастухов Ю.Ф., Смоляк А.И. Шифрование гиперболическими функциями(http://elib.psu.by:8080/handle/123456789/22089).
6. Пастухов Д.Ф., Пастухов Ю.Ф., Синица П.Р. Шифрование данных на базе эллиптических кривых: учебно-методическое пособие для студентов спец.1-98 01 01 (http://elib.psu.by:8080/handle/123456789/16814).
7. Голубева О.В., Ехилевский С.Г., Пастухов Ю.Ф., Пастухов Д.Ф. Шифрование с помощью нелинейных функций : учебно-методическое пособие для студентов спец.1-98 01 01(http://elib.psu.by:8080/handle/123456789/20320).

УДК 519.72

Кандидат физико-математических наук Пастухов Дмитрий Феликсович

Кандидат физико-математических наук Пастухов Юрий Феликсович

Сонич Алексей Дмитриевич

Ченторицкий Алексей Юрьевич

Полоцкий государственный университет

2019