УДК 004.021

РАЗРАБОТКА СИСТЕМЫ ДЛЯ СКРЫТИЯ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ГИПЕРБОЛИЧЕСКИХ ФУНКЦИЙ

А.И. СМОЛЯК
(Представлено: канд. физ.-мат. наук, доц. Д.Ф. ПАСТУХОВ)

Рассматриваются: алгоритм скрытия данных в изображении на основе дискретных преобразований, описание работы основной программы и основные функции разрабатываемого программного продукта.

Введение. Первое появление гиперболических функций историки обнаружили в трудах английского математика Абрахама де Муавра. Современное определение и обстоятельное их исследование выполнил Винченцо Риккати в 1757 году, он же предложил их обозначения: sh, ch. Независимое открытие и дальнейшее исследование свойств гиперболических функций было проведено Иоганном Ламбертом, который установил широкий параллелизм формул обычной и гиперболической тригонометрии. Н. И. Лобачевский впоследствии использовал этот параллелизм, пытаясь доказать непротиворечивость неевклидовой геометрии, в которой круговая тригонометрия заменяется на гиперболическую.

Основная часть. Для разработки программного средства организации и функционирования программы необходимо выбрать среду разработки, с помощью которой будет производиться проектирование.

В основе систем быстрой разработки (RAD-систем, Rapid Application Development – среда быстрой разработки приложений) лежит технология визуального проектирования и событийного программирования, суть которой заключается в том, что среда разработки берет на себя большую часть рутинной работы, оставляя программисту работу по конструированию диалоговых окон и функций обработки событий. Производительность программиста при использовании RAD-систем является рекордно высокой.

Также можно рассмотреть такую среду разработки, как и Microsoft Developer Studio. Студия разработчика фирмы Microsoft (Microsoft Developer Studio) – это интегрированная среда для разработки, позволяющая функционировать различным средам разработки, одна из которых Visual C++, другая - Visual J++. В дальнейшем будет идти речь только о среде разработки Visual C++. Среда разработки Visual Studio позволяет создавать библиотеки dll на языке C++ и подключать основное ядро программы, написанное на C++ к интерфейсу, созданному на FORTRAN и динамически управляемым югославской библиотекой Xtfort.

Основными функциями разрабатываемого программного продукта являются:

– шифрование информации;

– дешифрование информации;

Рассматриваемая методика ляжет в основу разрабатываемого программного продукта «Скрытие информации с использованием гиперболических функций».

Основная идея алгоритма шифрования алгоритма шифрования гиперболическими функциями является шифрование каждого первого, второго и третьего символа символа сообщения отдельными ключами. Каждый из этих ключей имеют 3 параметра. В результате дешифрование сообщений только одним или двумя ключами с правильными параметрами ключей принципиально невозможно.

Подробнее опишем алгоритм работы программы.

1. Каждому английскому символу, цифрам и знакам клавиатуры соответствует некоторое целое число(номер) 0т 0 до 255, с помощью таблицы ASCII можно перевести любой символ клавиатуры в число, например, слово Polotsk шифруется 7 символами 80 111 108 111 116 - 115 107.

2. Всего основных символов в клавиатуре 256 с участием больших и малых букв английского шрифта, невидимые знаки (табуляция, начало и конец строки и т.д.) а также знаки типа - !,»,№,(,) и т.д., цифры 0,1,2,3,4,5,6,7,8,9 пронумерованы от 0 до 255. Мы можем перевести каждый символ после применения команды ASCII в действительное число, заключённое от 0 до 1 по формуле $x = n/255, 0 \le x \le 1$, где n -номер символа в ASCII.

3. Далее с единичным отрезком мы проводим линейное преобразование, растягивающее его в несколько раз и с применением параллельного переноса, смещающее левую точку по формуле: $res2[i] = a + (b - a)res1[i]$, где a, b левая и правая границы нового отрезка. В результате последнего преобразования все точки преобразованного отрезка имеют координаты $x_i \in [a, b]$. Сначала работает прямая функция с вызовом res3[i]=c*f(res2[i],k).В которой по порядку следуют аргументы: число $x = res2[i], x \in [a, b]$,c- коэффициент подобия, k номер гиперболической функций целое k=1 для гипер-

болического синуса и k = 1 для гиперболического косинуса. Поскольку диапазон целых чисел значительно меньше диапазона действительных чисел с плавающей запятой двойной точности, то массив res101[i] (res101[i] = res3[i]) заполняется действительными числами двойной точности, здесь применяется одна важная идея. В файл 1001.txt записываются сначала целые части действительных чисел шифра – все нечётные строки текстового файла sad[2*i+1]. Затем мантиссы шифра переводятся в целые числа (первые 9 знаков) и заполняются все чётные строки текстового файла sad[2*i]. Иначе при считывании чисел из файла 1001.txt может быть потеряна мантисса действительного числа. Это во много раз может снизить размерность пространства шифрующих ключей и надёжность дешифрования данных.

4. Открываем и копируем данные текстового файла1001.txt в массив sad[2*i], затем из каждой соседней пары целых чисел формируем целую часть и мантиссу шифра, создаём массив по формуле res102[i] = double(sad[2*i])+ double(sad[2*i+1])*1e-9
5. Далее столбцы кода до записи в текстовый файл и после чтения из него сравниваются res3[i]; и res102[i] (должны совпадать).
6. Прочитанные данные записываются в файл res102[i].
7. Проводим обратное масштабирование res4[i]= res102[i]/c.
8. Вызываем обратную функцию дешифрования: res5[i]=log(res4[i]+sqrt(res4[i]*res4[i]+L))

$$x = \ln(z) = ln\left(y + \sqrt{y^2 + L}\right), L = \begin{cases} 1, y(x) = sh(x) \\ -1, y(x) = ch(x) \end{cases} \Leftrightarrow y(x) = \frac{e^x - Le^{-x}}{2}, L = -1,1.$$

9. Проводим обратное линейное преобразование и возвращаемся к переменной x по формуле: res6[i]= 255.0*(res5[i]-a)/(b-a)).
10. Переводим действительный массив дешифрованного символа в целочисленный массив: res7[i]= int(res6[i]).
11. Делаем обратное преобразование ASCII. В результате исходная символьная фраза программой возвращается в эту же фразу. Таким образом, в программе использованы математические операции запись чисел с двойной точностью в текстовой файл, композиция из преобразования подобия, нелинейных гиперболических функций, однородного сжатия функции:

$$x_1 = \frac{n}{255}, x_2 = a + (b - a)x_1, y_1 = \frac{\exp(x_2) - l*\exp(-x_2)}{2}, y_2 = cy_1.$$

12. Обратные преобразования:

$$y_1 = \frac{y_2}{c}, x_2 = \ln\left(y_1 + \sqrt{y_1^2 + l}\right), x_1 = \frac{x_2 - a}{b - a}, n = 255x_1, \left(l = -1 \leftrightarrow ch(x)\right).$$

Функция шифрования $f(x,k)$ содержит два аргумента. Первый аргумент x – действительное неотрицательное число, преобразованное шифруемым символом. Второй целочисленный аргумент принимает значение $k = 1$ для гиперболического синуса и $k = 2$ для гиперболического косинуса m - число символов в сообщении.

Листинг 1 – Задание параметров ключей

```
1:  double f(double x, int k){
2:  if(k==1){return (exp(x)-exp(-x))/2.0;}
3:  if(k==2){return (exp(x)+exp(-x))/2.0;}}
4:  int const m=20
5:  main(){
6:  int i,aa,bb,k,l, res7[m];
7:  double x, res1[m], res2[m] , res3[m] ,a1,b1,c1;
8:  double  res4[m], res5[m] , res6[m] ;
9:  double a2,b2,c2,a3,b3,c3,res101[m], res102[m];
10: ]char str[m+1]=("Smolak Igor 2018 FIT");
11: k=2;l=1;
12: for(i=1;i<=k+1;i++) {l=l*(-1);}
13: a1=2.0;b1=3.0;c1=4.0;
14: a2=7.0;b2=10.0;c2=5.0;
15: a3=0.0;b3=3.0;c3=6.0;
```

```
16:  for(i=0;i<=m-1;i++){
17:  res1[i]=double(str[i])/255.0;
```

Листинг 2 – Шифрование

```
1:   if(i%3==0){
2:   res2[i]=a1+(b1-a1)*res1[i];
3:   res3[i]=c1*f(res2[i],k); }
4:   else if(i%3==1){
5:   res2[i]=a2+(b2-a2)*res1[i];
6:   res3[i]=c2*f(res2[i],k); }
7:   else if(i%3==2){
8:   res2[i]=a3+(b3-a3)*res1[i];
9:   res3[i]=c3*f(res2[i],k);}
10:  printf("i=%d %c %d normal=%.16lf coder=%.16lf \n",i,str[i],
11:  str[i],res1[i],res3[i]);}
12:  for(i=0;i<=m-1;i++)  {
13:  printf("%.16lf \n", res3[i]);
14:  res101[i]= res3[i];  }
15:  FILE*file;
16:  int sad0[2*m+1];
17:  remove("1001.txt");
18:  file=fopen("1001.txt","w");
19:  for(i=0;i<=m-1;i++){
20:  aa=int(res3[i]);
21:  bb=int ((res3[i]-aa)*1e9);
22:  printf("coder1(%d)=%.12lf\n",i,res3[i]);
23:  fprintf(file,"%d\n", aa);
24:  sad0[2*i]=aa;
25:  fprintf(file,"%d\n", bb); )
26:  sad0[2*i+1]=bb;}
27:  fclose(file);
28:  int sad[2*m+1];
29:  char arr00[2*m+1];
30:  file=fopen("1001.txt","r");
31:  if(file==NULL){
32:  printf(" not open file");}
33:  else{
34:  for(i=0; i<=2*m-1;i++){
35:  fgets(arr00,2*m-1,file);
36:  sad[i]=atoi(arr00);
37:  printf(" coder2(%d)=%d coder1(%d)=%d\n", i, sad[i],i, sad0[i]);}}
38:  fclose(file);
39:  for(i=0;i<=m-1;i++){
40:  res102[i] = double(sad[2*i])+ double(sad[2*i+1])*1e-9;
41:  printf(" coder1(%d)=%.8lf  coder2(%d)=%.8lf\n", i, res101[i], i,
42:  res102[i] );}
43:  for(i=0;i<=m-1;i++){
44:  if(i%3==0){
45:  res4[i]= res102[i]/c1;
46:  res5[i]=log(res4[i]+sqrt(res4[i]*res4[i]+l));
47:  res6[i]= 255.0*(res5[i]-a1)/(b1-a1);}
48:  else if(i%3==1){
49:  res4[i]= res102[i]/c2; // res4[i]= res3[i]/c2
50:  res5[i]=log(res4[i]+sqrt(res4[i]*res4[i]+l)) ;// res5[i]= res2[i]
51:  res6[i]= 255.0*(res5[i]-a2)/(b2-a2);} // res6[i]= res1[i]* 255.0
52:  else if(i%3==2){
53:  res4[i]= res102[i]/c3;
54:  res5[i]=log(res4[i]+sqrt(res4[i]*res4[i]+l));
55:  res6[i]= 255.0*(res5[i]-a3)/(b3-a3);}
```

```
56:  res7[i]= int(res6[i])+1;
57:  printf("i=%d text(%d)=%c decoder(%d)=%c \n",i,i,str[i],i,res7[i]);}}
```

Заключение. В данной статье был рассмотрен алгоритм скрытия информации с использованием гиперболических функций, описание работы основной программы и основные функции разрабатываемого программного продукта.

ЛИТЕРАТУРА

1. Жданов, О.Н. Эллиптические кривые: Основы теории и криптографические приложения / О.Н. Жданов, В.А. Чалкин. – М. : Книжный дом ЛИБРИКОМ, 2013. – 200 с.
2. Бутакова, Н.Г. Криптографическая защита информации : учеб. пособие для вузов / Н.Г. Бутакова, В.А. Семененко, Н.В. Федоров. – М. : Изд-во МГИУ, 2011. – 316 с.