Е.Д. Дунаев, Е.Р. Кирколуп

РАЗРАБОТКА ПРИЛОЖЕНИЯ, РЕАЛИЗУЮЩЕГО КРИПТОАНАЛИЗ ШИФРА ВИЖЕНЕРА

В статье описывается компьютерное приложение, которое может осуществлять автоматизированную атаку на шифр Виженера с известной или неизвестной длиной ключа. Приложение дополнительно имеет функции шифрования с помощью метода Виженера. Отличительной особенностью данного приложения является то, что оно само находит правильно расшифрованное сообщение и записывает его в отдельный текстовый файл. Разработанное приложение может быть использовано при проведении лабораторных занятий по информационной безопасности в вузе.

Ключевые слова: шифр Виженера, информационная безопасность, криптоанализ шифра Виженера, автоматизированная атака на шифр Виженера.

E.D. Dunaev, E.R. Kirkolup

THE APPLICATION DEVELOPMENT TO REALIZE THE VIGENERE CIPHER CRYPTANALYSIS

The article describes the computer application that can realize the automatic attack of the Vigenere cipher with a determined or undetermined key length. The application has also a possibility to cipher using Vigenere method. A distinctive feature of this application is that it finds itself correctly deciphered messages and writes them down into a separate text file. The elaborated application can be used while managing laboratory works of information security at the university.

Key words: Vigenere cipher, information security, the Vigenere cipher cryptanalysis, the automatic attack of the Vigenere cipher.

До сих пор одной из существенных проблем при хранении, передаче информации, разработке и использовании различных информационных технологий, web-технологий является защита информации. Например, информация, которая хранится на сервере, в целях безопасности шифруется с применением разных криптоалгоритмов: шифра Виженера, алгоритма DES, генератора псевдослучайных чисел, алгоритма RSA и других [1, с. 171; 2, с. 68–71]. Шифрование информации в данном случае – это единственное надёжное решение задачи защиты информации. Поэтому в настоящее время число подходов к шифрованию постоянно увеличивается [3], а существующие алгоритмы шифрования модифицируются и усложняются.

Шифр Виженера относится к историческим шифрам, с изучения которых начинается практически каждый курс криптографии [2, с. 66]. Тем не менее его до сих пор используют при защите информации, например на web-серверах. Шифр Виженера является сложным для криптоанализа и какое-то время даже считался невзламываемым. И сейчас шифр Виженера можно сделать невзламываемым, если использовать случайный ключ, хотя такой системе затруднительно найти практическое применение. При изучении основ информационной безопасности в вузе очень важно знакомиться с подобными шифрами и проводить их криптоанализ, но так как вручную осуществлять криптоатаку на шифр Виженера достаточно трудно, то актуальным является создание компьютерных приложений, которые были бы способны осуществлять автоматическую атаку на этот шифр. В данной статье описано компьютерное приложение, которое может осуществлять автоматизированную атаку на шифр Виженера с известной или неизвестной длиной ключа и дополнительно имеет функции шифрования с помощью метода Виженера. Разработанное приложение значительно сокращает время криптоанализа шифра Виженера и может быть использовано при проведении лабораторных занятий по информационной безопасности.

Приложение, реализующее автоматическую атаку на шифр Виженера, разработано в среде программирования Delphi и может работать в двух режимах:

1) в режиме шифрования информации, в частности слов, строк или текста, написанного при помощи букв русского алфавита;

2) в режиме автоматического дешифрования текстовой информации разной сложности ключа (длиной до 6 символов).

В режиме шифрования приложение работает следующим образом. Текст, который необходимо зашифровать, вводится в первое текстовое поле, расположенное слева (рис. 1). Затем необходимо придумать и ввести ключ (кодовое слово). Чем длиннее будет ключ, тем сложнее дешифровать текст. Ключевое слово необходимо запомнить и надежно хранить, не передавая третьим лицам, так как без ключа расшифровать текст будет очень сложно, а с ним не составит большого труда. После введения текста и ключевого слова зашифрованный текст будет записан во второе текстовое поле, расположенное справа. За указанный процесс в работе приложения отвечает следующий блок кода.

```
for i:=1 to length(text2) do
  for j:=1 to length(alf) do
    begin
      s:=(i-1) mod length(kluch)+1;
      t:=key[s];
      k:=(j+2*t-3) mod length(alf)+1;
      if text2[i]=alf[j] then text3:=text3+alf[k];
    end;
```

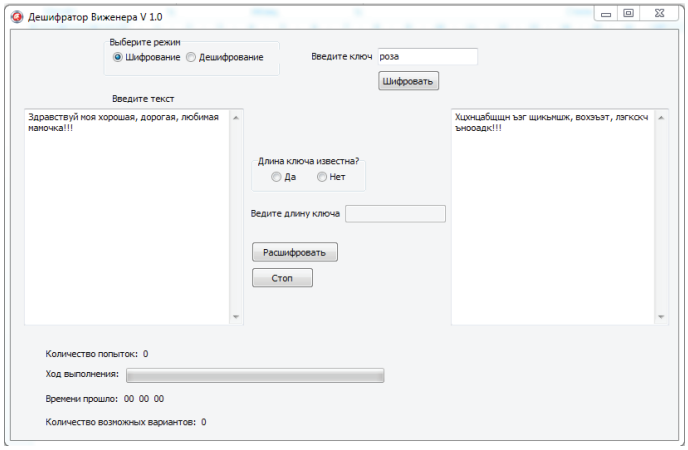

Рис. 1. Режим шифрования

Чтобы использовать приложение в режиме дешифрования, необходимо переключить соответствующий указатель в верхней части программы (рис. 2). Затем в первое текстовое поле вставить исходный зашифрованный текст. Далее следует указать: известна ли длина ключа или неизвестна. Если пользователю известна длина ключа, к примеру, определена с помощью теста Касиски [4, с. 34–37], то ее необходимо указать. Данный выбор пользователя значительно ускорит работу нахождения ключа и расшифровку данного сообщения. После нажатия кнопки «Расшифровать» начнется процесс дешифрования. Каждая попытка перебора ключа и расшифрованного по нему сообщения будет записана в текстовые файлы, которые создаются автоматически в том же каталоге, где расположено само приложение. В один такой файл записывается по 50 тысяч попыток. Отличительной особенностью данного приложения является то, что оно само пробует найти правильно расшифрованное сообщение и записывает эти варианты в отдельный текстовый файл. Распознавание текста происходит по словарю, состоящему из 1–2 символьных слов (рис. 3).

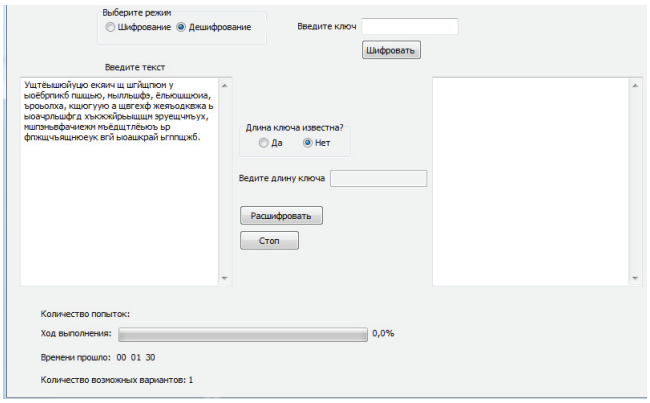

Рис. 2. Режим дешифрования

```
const slov2:array [1..303] of string=('АА','АД','АЕ','АЖ','АЗ','АИ','АЙ','АК',
'АЛ','АМ','АН','АО','АР','АС','АТ','АУ','АХ','АЧ','АШ','ГА','ГЕ','ГИ','ГМ',
'ГО','ГЫ','ГЭ','ЕЕ','ЕЁ','ЁЖ','ЕЙ','ЕЛ','ЕМ','ЕЮ','БЫ','ВО','ВЫ','ДО','КА',
'КО','КУ','ЗА','ИЗ','ИЛ','ИМ','ИХ','ЖЕ','МЫ','НА','НЕ','НИ','НО','НУ','ОТ',
'ОХ','ЛИ','РА','УЖ','УМ','УС','УХ','СО','ТА','ТЕ','ТС','ТУ','ТЫ','ФУ','ЩА',
'ЩИ','ЭЙ','ЭХ','ЮГ','ЯД','ЯМ','ЯН','ЕЕ','аа','ад','ае','аж','аз','аи','ай',
'ак','ал','ам','ан','ао','ар','ас','ат','ау','ах','ач','аш','га','ге','ги',
'гм','го','ты','гэ','ее','её','ёж','ей','ел','ем','ею','бы','во','вы','до',
'ка','ко','ку','за','из','ил','им','их','же','мы','на','не','ни','но','ну',
'об','ой','он','от','ох','ли','ра','по','уж','ум','ус','ух','со','та','те',
'тс','ту','ты','фу','ща','щи','эй','эх','юг','яд','ям','ян','ее','Аа','Ад',
'Ае','Аж','Аз','Аи','Ай','Ак','Ал','Ам','Ан','Ао','Ар','Ас','Ат','Ау','Ах',
'Ач','Аш','Га','Ге','Ги','Гм','Го','Ты','Гэ','Еб','Её','Ёж','Ей','Ел','Ем',
'Ею','Бы','Во','Вы','До','Ка','Ко','Ку','За','Из','Ил','Им','Их','Же','Мы',
'На','Не','Ни','Но','Ну','Об','Ой','Он','От','Ох','Ли','Ра','По','Уж','Ум',
'Ус','Ух','Со','Та','Те','Тс','Ту','Ты','Фу','Ща','Щи','Эй','Эх','Юг','Яд',
'Ям','Ян','Ее','ОБ','ОЙ','ОН','ПО','ТО','то','То','А','а','Б','б','В','в',
'Г','г','Д','д','Е','е','Ё','ё','Ж','ж','З','з','И','и','Й','й','К','к','Л',
'л','М','м','Н','н','О','о','П','п','Р','р','С','с','Т','т','У','у','Ф','ф',
'Х','х','Ц','ц','Ч','ч','Ш','ш','Щ','щ','Э','э','Ю','ю','Я','я');
```

Рис. 3. Фрагмент словаря

Работа приложения в режиме дешифрования осуществляется следующим образом. Первым делом приложение форматирует текст в удобный вид: сканирует всю информацию, удаляет пробелы и знаки препинания, преобразует весь текст в одно слово. Это происходит с помощью цикла, который проверяет каждый символ в тексте. Если символа нет в русском алфавите, то он отбрасывается и цикл переходит дальше, при этом все выброшенные символы и их расположение сохраняются в специальном массиве. Блок цикла, который выполняет эти действия, выглядит так:

```
for i:=1 to length(text) do
 begin
  flag:=false;
  for j:=1 to length(alf) do
   if text[i]=alf[j] then
    begin
     text2:=text2+alf[j];
     flag:=true;
    end;
  if flag=false then
   begin
   sym[i]:=text[i];
   sym2[i]:=' ';
   end;
 end;
```

Следующий шаг работы приложения – это перебор всех возможных вариантов ключей. При этом каждая попытка перебора делает обратную шифровку текста по ключу, который в свою очередь генерируется программой. Если пользователь в самом начале не указал определенную длину ключа, то программа начинает свою проверку с односимвольного ключа «А», далее «Б» и т. д. После того, как все возможные варианты односимвольного ключа были использованы, дешифратор начинает генерировать двухсимвольные ключи, начиная с «АА». В настоящее время приложение способно дешифровать сообщения, зашифрованные ключом длиной 6 символов. Это означает, что последней будет попытка с ключом «ЯЯЯЯЯЯ». Так как в русском алфавите содержатся 33 буквы, из этого следует, что число всех возможных попыток перебора равно 336 = 1291467969. Для выполнения такой большой работы необходимо немалое количество времени (не менее 20 часов) и компьютер средней (по современным меркам),мощности. Поэтому если пользователю сразу известна длина ключа, то время атаки значительно сократится. Например, будет указана длина ключа 3, следовательно, программа начнет перебор ключа не с «А», а «ААА». Код блока перебора имеет такой вид:

```
if kluch>7 then exit;
for I := 1 to kluch do
finish:=finish*33;
finish2:=finish2+finish;
for i1:=1 to 33 do
begin
  Application.ProcessMessages;
 for i2:=1 to 33 do
  for i3:=1 to 33 do
   for i4:=1 to 33 do
    for i5:=1 to 33 do
     for i6:=1 to 33 do
      for i7:=1 to 33 do
     begin
     SetLength(key,kluch+5);
      key[1]:=i7;
      key[2]:=i6;
      key[3]:=i5;
      key[4]:=i4;
      key[5]:=i3;
      key[6]:=i2;
      key[7]:=i1;
      d:='';
```

```
for i:=1 to kluch do
d:=d+inttostr(key[i])+' ';
d:=Trimright(d);
//определение индексов бугв ключа
for i:=1 to kluch do
dd:=dd+alf2[key[i]];
if x=0 then
 begin
  rewrite(f1);
  x:=x+1;
 end;
 if x>0 then
  append(f1);
for i:=1 to length(text2) do
  for j:=1 to length(alf) do
    begin
      s:=(i-1) mod kluch+1;
      t:=key[s];
      k:=(j+2*t-3) mod length(alf)+1;
  if text2[i]=alf[j] then text3:=text3+alf[k];
    end;
label12.Caption:=inttostr(x3);
 if stop=true then
 begin
 Timer1.Enabled:=false;
 exit;
end;
```

После завершения процедуры перебора запускается автоматизированный поиск правильных вариантов. Сначала программа сканирует текст на наличие слов, содержащих одну или две буквы, и сохраняет их в отдельный массив. Далее идет сравнение слов массива со словарем. Если все расшифрованные слова находят своих двойников в словаре, то данный вариант перебора помечается как возможно правильный и заносится в отдельный текстовый файл, созданный только для помеченных попыток. По завершении работы приложения пользователь может открыть текстовый файл с возможно правильными вариантами и найти расшифрованное сообщение. Процесс отбора подходящих слов и сравнение со словарем происходит в следующем блоке:

```
//поиск 1 и 2 буквенных слов
SetLength(slov,length(cc2)+5);
n:=1;
 for i:=1 to length(cc2) do
   begin
    flag:=false;
    for j:=1 to length(alf) do
     if cc2[i]=alf[j] then
       begin
         slov[n]:=slov[n]+alf[j];
         flag:=true;
       end;
     if flag=false then
      if n<length(cc2) then n:=n+1;
   end;
//проверка слов
m:=0;
 for I := 1 to Length(cc2) do
  begin
if (length(slov[i])=1) or (length(slov[i])=2) then
    begin
     flag:=false;
     for j := 1 to Length(slov2) do
      begin
       if slov[i]=slov2[j] then
        flag:=true;
      end;
     if flag=false then
       begin
        m:=1;
       end;
    end;
  end;
 writeln(f1,'Попытка №',x3);
 writeln(f1,'Ключ в цифрах[',d,']');
 writeln(f1,'Ключ в буквах[',dd,']');
 writeln(f1,cc);
 writeln(f1,'');
 if regim=1 then
 begin
   progress:=(x3*100)/finish;
 ProgressBar1.Position:=round(progress);
   Label9.Caption:=FloatToStrf(progress,
ffFixed,3,1)+'%';
  End;
```

Разработанное приложение было апробировано в ходе изучения студентами основ информационной безопасности на кафедре информационных технологий АлтГПУ. При выполнении лабораторной работы «Криптографическая атака на шифр Виженера» студентам предлагалось расшифровать несколько сообщений, зашифрованных с помощью шифра Виженера. Без автоматизированных дешифраторов выполнить лабораторную работу смогли лишь несколько студентов, затратив на это порядка 8 дней каждый. С использованием приложения процесс дешифрации сообщений, приведенных в лабораторной работе, занимал значительно меньше времени (5–6 часов) при условии, что студентами предварительно была вычислена длина ключа.

Описанное в статье компьютерное приложение, осуществляющее автоматизированную атаку на шифр Виженера с известной или неизвестной длиной ключа и имеющее дополнительную функцию шифрования с помощью данного метода, значительно сокращает время криптоанализа шифра Виженера. При этом криптоатака на шифр проходит полностью автоматизированно, путем перебора всевозможных вариантов и распознавания текста. Разработанное приложение может быть использовано при проведении лабораторных занятий по информационной безопасности в вузе.

Библиографический список

1. Корсунов, Н. И. Повышение эффективности защиты информации модификацией шифра Вижинера / Н. И. Корсунов, А. И. Титов // Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика. — 2010. — № 7—1 (78). — С. 171—175.

2. Марков, А. С. Основы криптографии: подготовка к cissp / А. С. Марков, В. Л. Цирлов // Вопросы кибербезопасности. — 2015. — № 1 (9). — С. 65—73.

3. Бабенко, Л. К. Развитие криптографических методов и средств защиты информации / Л. К. Бабенко, Е. А. Ищукова, Е. А. Маро и др. // Известия ЮФУ. Технические науки. — 2012. — № 4. — С. 40—50.

4. Габидулин, Э. М. Защита информации : учебное пособие / Э. М. Габидулин, А. С. Кшевецкий, А. И. Колыбельников. — Москва : МФТИ, 2011. — 262 с.