Шифр Виженера

Кевролетин В.В.

12 декабря 2011 г.

Задание 3.1

Условие

Какова методика криптоанализа шифра Виженера (реализовать (Mathematica, Scheme, Sage, ...) и продемонстрировать методику криптоанализа на достаточно длинном шифрованном тексте.)? Землянникова

Решение

Реализация шифрования/дешифрования. Язык реализации: Перл

```
use warnings;
use strict;

sub table {
    my $j = 0;
    my $alph = ['A' .. 'Z'];
    my ($rows, $cols, $rows_sym_pos, $cols_sym_pos);
    for my $i (0 .. 25 ) {
        $cols_sym_pos->{$alph->[$i]} = $i;
    }
    for my $j (0 .. 25) {
        my ($t, $h);
        for my $i (0 .. 25) {
            my $sym = $alph->[($i + $j) % 26];
            $rows->[$j]->{array}->[$i] = $sym;
            $rows->[$j]->{hash}{$sym} = $i;
            $cols->[$i]->{array}->[$j] = $sym;
            $cols->[$i]->{hash}{$sym} = $j;
        }
        $rows_sym_pos->{$alph->[$j]} = $j;
    }
    { rows => $rows,
      cols => $cols,
      rows_sym_pos => $rows_sym_pos,
      rows_pos_sym => ['A' .. 'Z'],
      cols_sym_pos => $cols_sym_pos,
      cols_pos_sym => ['A' .. 'Z'] }
}
```

```
sub encode_symbol {
    my ($table, $key_sym, $sym) = @_;
    my $row = $table->{rows_sym_pos}{$sym};
    my $col = $table->{cols_sym_pos}{$key_sym};
    $table->{rows}->[$row]->{array}->[$col];
}

sub decode_symbol {
    my ($table, $key_sym, $sym) = @_;
    my $row = $table->{cols_sym_pos}{$key_sym};
    my $col = $table->{rows}->[$row]->{hash}{$sym};
    $row = $table->{cols}->[$col]->{hash}{$sym};
    $table->{rows_pos_sym}->[$row];
}

sub apply_circulary {
    my ($key, $text, $sub) = @_;
    my $table = table();
    my @a = split //, $key;
    my $i = 0;
    my $proc = sub {
        my $res = $sub->($table, $a[$i], $_[0]);
        $i = ($i + 1) % @a;
        $res
    };
    join '', map { $proc->($_) } split //, $text
}

sub encode {
    my ($key, $text) = @_;
    apply_circulary($key, $text, \&encode_symbol)
}

sub decode {
    my ($key, $text) = @_;
    apply_circulary($key, $text, \&decode_symbol)
}

my $key = 'CRYPTO';
my $text = 'PURPLEPURPLE';
my $res = encode($key, $text);
print "$res\n";
print  decode($key, $res);
```

Если известна длина ключа, то задачу криптоанализа можно свести к задаче криптоанализа одноалфавитной системы. Для этого можно записать криптотекст в несколько столбцов, так чтобы количество столбцов было равно длине ключа. Тогда буквы в одном столбце зашифрованы с использованием одинаковой буквы ключа, т.е. одной и той же букве откры-

того текста будет соответствовать одна и та же буква криптотекста будет и можно будет использовать подсчет частот вхождения символов в текст. Если длина ключа неизвестно необходимо иметь метод для определния его длины.

Примерно в 1860 г. немецким криптоаналитиком Ф.У. Казизки был изобретен метод для вскрытия периодических криптосистем с неиз- вестным периодом. Метод Казизки выявляет период с помощью об- наружения одинаковых слов в криптотексте. [Саломаа А. "КРИПТОГРАФИЯ С ОТКРЫТЫМ КЛЮЧОМ"]

Допустим, мы нашли слово повторяющееся дважды, с 20 буквами между двумя появлениями: Это может означать тот факт, что одинаковая часть сообщения зашифрована, начиная с той же пози- ции ключа. Тогда расстояние между 2мя вхождениями кратно длине ключа. Поэтому возможна длина ключа 2, 4, 5, 10 или 20. Проверяя полученную гипотизу можно попытаться вскрыть шифр как описано выше.

Таким образом первым шагом криптоанализа является определение наиболее вероятной длины ключа.

Для этого можно найти все вхождения каждой подстроки криптотекста. Затем необходимо проанализировать полученную информацию и выделить наиболее длинные и наиболее повторяющиеся подстроки. В предложенной ниже программе этот выбор делается следующием образом: длина подстроки складывается с количеством её вхождений в текст, из всех таких сумм выбирается наибольшая и выбирается соответствующая этой сумме подстрока(такая оценка взята на угад и не притендует на самою справедливую или походящую оценку)

Язык реализации: Перл.

```
use warnings;
use strict;
use List::Util 'reduce';

sub analize_periods {
    my ($periods) = @_;
    my %periods;
    for (keys %$periods) {
        my ($pattern, $positions) = ($_, $periods->{$_});
        my $scores = length($pattern) + @$positions;
        for my $i (1 .. $#{$positions}) {
            my $period = $positions->[$i] - $positions->[$i - 1];
            $periods{$period} ||= 0;
            $periods{$period} += $scores;
        }
    }
    my $ans = reduce { $periods{$a} > $periods{$b} ? $a : $b }
        keys %periods;
    my @rating = sort { $periods{$a} <=> $periods{$b} }
        keys %periods;
    ($ans, \@rating);
}
```

```
sub find_period {
    my ($text) = @_;
    my %res;
    for my $p_len (1 .. length($text) - 1) {
        for my $i (0 .. length($text) - $p_len) {
            my $pattern = substr($text, $i, $p_len);
            next if $res{$pattern};
            $res{$pattern} ||= [];
            while ($text =~ m/$pattern/g) {
                push @{$res{$pattern}}, pos($text)
            }
        }
    }
    my ($ans, $rating) = analize_periods(\%res);
    $ans
}

print find_period('ababab');          # -> 2
print find_period('zabcabc435');      # -> 3
```

Данная программа так же тестировалась на примере из книги Саломаа А. "КРИПТОГРАФИЯ С ОТКРЫТЫМ КЛЮЧОМ"стр. 47:

AVXZHHCKALBRVIGLGMOSTHBWMSXSPELHHLLABABKVXHIGHPN
PRALFXAVBUPNPZWVCLTOESCGLGBMHKHPELHBWZXVTLACFJOG
UCGSSLZWMZHBUURDSBZHALVXHFMVMOFHDKTASKVBPFULQHTS
LTCKGAVHMLFRVITYILFBLBVLPHAVHHBUGTBBTAVXZWPBZPGG
VHWPXTCLAQHTAHUAPBZHGTBBTPGMOLACOLKBAVMVALGMVJXP
GHUZUMFLHTSPHEKBCGLGHUHHWHALMISALOMLRIYCPGOLFRZA
TSZGWMOHALVXHFMVTLHIMOSLLVNTSMOIWYMTHFMVGVBGBZHT
VVTCCYLKRHABAVTJBMOSILFELJXYTLHIGHACWWLHURTLLLRS
SMLAZSCNKVFIPXGH

И выдаёт верный результат - 3 (стоит отметить, что 100% ответа для данной задачи дать нельзя и пример показывает что программа выдаёт неплохие результаты, но в любом случае для реально примера пришлось бы изучать статистуку более детально и производить перебор).

Как было отмечено выше, после определения периода задача сводится к задаче криптоанализа модулярного шифра, процесс криптоанализа которого был описан в предыдущие заданиях.