УДК 004.021

ТРИГОНОМЕТРИЧЕСКАЯ КРИПТОГРАФИЯ

Каминский Л.П., Степанов В.А.
научный руководитель доктор физ.-мат. наук Кытманов А.А.
Сибирский Федеральный Университет

1. Описание работы тригонометрического шифра

Уравнения волны $y = \cos(x + N \cdot \Delta x)$ - пример одной из многих функций, имеющих постоянную амплитуду и являющихся непрерывными на всем промежутке $x \in (-\infty, +\infty)$. Важным моментом является то, что если для $y = \cos(x + \Delta x)$ параметр Δx не равен $\frac{-2\pi}{\mathrm{N}}$, где N - любое целое число, то период гаммирования данной конкретной функции бесконечен.

Алгоритм шифрования:

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

$$Y = X + 256 \cdot [\cos(Z + N \cdot \Delta x)] \bmod (256)$$

Где: X – порядковый номер того символа который нужно зашифровать;

$\mathrm{Z}, \Delta x$ - любые числа, являющиеся секретными параметрами нашего ключа. Остальные параметры не являются секретными. $\mathrm{Z}, \Delta x \in (-\infty, +\infty)$

N - номер по счету шифруемого символа в исходном тексте;

256 – мощность исходного алфавита. На самом деле мощность исходного алфавита может быть любой. (В нашем случае мощность равна 256, как количество символов в расширенной таблице ASCII)

Алгоритм дешифровки:

Тригонометрический шифр является примером симметричного алгоритма шифрования, следовательно:

$$X = Y - 256 \cdot [\cos(Z + N \cdot \Delta x)] \bmod (256)$$

2. Проблема. Цель. Актуальность.

Сам шифр был разработан В.П.Сизовым и успешно представлен на Всероссийскую конференцию «РусКрипто» в 2005 году [1].

Однако в 2011 году вышла статья [2], утверждающая, что данный шифр стоек лишь относительно прямого перебора, а не против специально разработанного генетического алгоритма. Согласно данной статье, шифр, в таком представлении, является уязвимым. На данный момент существует два способа улучшения данного шифра, однако работ в данном направлении нет. У нас появился интерес опробовать улучшенные версии тригонометрического шифра на надежность с помощью генетического алгоритма, тем самым дать ответ на пригодность и конкурентоспособность данного шифра вообще.

3. Математические уязвимости

Рассмотрим свойства криптосистемы с математической точки зрения. Вместо тригонометрических функций можно взять любые периодические непрерывные функции, определённые на всей числовой прямой. В нашем примере мы выбрали косинус, имеющий период 2π. Тогда рассмотрим следующие выражения:

$\cos((Z + 2\pi) + N \cdot \Delta x) = \cos(Z + N \cdot \Delta x)$,

$\cos(Z + N(\Delta x + 2\pi)) = \cos(Z + N \cdot \Delta x)$.

Второе выражение справедливо только для целого N, что, вообще говоря, выполняется. Таким образом, задача имеет не одно решение, а целое множество, каждое из которых отличается на 2π по любой координате. Это "уязвимое место" справедливо и для остальных модификаций криптосхемы - достаточно лишь знать период функции.

Этот факт снижает пространство поиска с R^2 до прямоугольника

$$0 < Z < 2\pi \text{ на } 0 < \Delta x < 2\pi.$$

Из статьи [2] следует, что для получения текста, близкого к исходному, в качестве решения можно рассматривать не точку (пару секретных параметров), а некоторую ее окрестность. Теоретически радиус такой окрестности должен находиться в пределах 1/(2N) для параметра Z и в пределах 1/(2Nm) для параметра Δx. Для алфавита из N= 256 символов и текстов длиной порядка m = 500 символов эти величины имеют порядок $\mathbf{10^{-4}}$ и $\mathbf{10^{-6}}$соответственно.

Очевидно, что чем больше длина текста, тем меньше требуется радиус окрестности для корректной его дешифровки.

Простые практические исследования показали, что начальный фрагмент текста уже является читабельным в окрестности $\mathbf{10^{-4}}$ истинного решения. В окрестности $\mathbf{10^{-5}}$ в тексте уже легко прослеживается смысл (200-символьные тексты расшифровываются полностью), а в окрестности $\mathbf{10^{-6}}$ полностью расшифровываются даже 400-символьные тексты.

Итак, проблема с конечностью пространства решена. На прямоугольнике

$$0 < \Delta x < 2\pi \; и \; 0 < Z < 2\pi.$$

построим равномерную сетку с шагом h = 10^5. Решениями будут служить точки в узлах сетки. Для их представления потребуется хранить 5 разрядов после запятой по каждой координате. Нетрудно посчитать, что количество элементов в пространстве решений составит:

$$[(2\pi \cdot 10^5)^2] \lessapprox 4 \cdot 10^{11}$$

Однако решить даже такую задачу полным перебором, в отличие от генетического алгоритма, за приемлемое время не представляется возможным.

4. Генетический алгоритм

Для того, чтобы оценивать улучшения алгоритма шифрования с помощью генетического алгоритма – нужно сначала самим создать хороший генетический алгоритм и достичь результатов статьи [2]. В данный момент созданный нами алгоритм не показывает столь впечатляющих результатов.

Приведем пошаговое описание работы алгоритма:

1) Формируется начальная популяция. Количество особей задается пользователем. (Для кодирования мы будем использовать двоичный алфавит {0,1}. Хромосома будет представлять собой конкатенацию двух битовых строк. В структуре

особи будет храниться дробная часть чисел Δx и Z .Так как $\log_2 100000 \approx 16.684$, то для хранения 5 десятичных разрядов (а именно данную точность мы считаем достаточной) потребуется 17 двоичных. Вывод - особь есть упорядоченная последовательность 34 бит, хранящая дробные части ключа. В нашем случае каждая начальная особь задается псевдослучайно)

2) Выбирается число M - количество поколений. Параметр вновь задается пользователем.

3) Каждая особь расшифровывает шифр-текст, и далее фитнесс-функция применяется либо ко всему тексту, либо к его части (порядок фитнесс-функции $\boldsymbol{m}$ задается пользователем. В нашем случае фитнесс-функция считает среднее значение суммы вероятностей встречи пары подряд идущих символов в полученном тексте, согласно данным алфавита. В дальнейшем существует цель использовать вместо биграмм – триграммы.) Таким образом, каждой особи ставится в соответствие число, показывающее ее приспособленность.

4) Происходит сортировка всей популяции.

5) Отсортированная популяция делится на 5 групп (их размер также задается пользователем).

6) Первые 2 группы (с лучшими хромосомами или с лучшим значением фитнесс-функции) допускаются к кроссинговеру.

(В ходе скрещивания два родителя дают единственного потомка. Биты родителей с вероятностью 1/3 складываются по модулю 2 либо с такой же вероятностью наследуются от одного из них)

7) Четвертая группа (с особями низкой приспособленности) претерпевает мутацию. (инвертирование битов хромосомы с вероятностью ½)

8) Последняя группа (худшие особи) удаляется из популяции, а их место занимают потомки, полученные после скрещивания первых двух групп. В нашем случае каждые два родителя дают одного потомка, так что размер популяции не изменится.

9) Если количество поколений не превышает $\boldsymbol{M}$, то эволюция продолжается (возвращаемся к шагу 3).

Напомним, что особи хранят только дробную часть чисел Z и Δx. Так как мы рассматриваем прямоугольник

$$0 < Z < 2\pi \text{ и } 0 < \Delta x < 2\pi,$$

то целая часть каждого параметра варьируется от 0 до 6 и таким образом генетический алгоритм запускается 49 раз (по одному разу для каждой пары целых частей чисел Z и Δx).

5. Способы улучшения

На данный момент существует два основных способа улучшения самого алгоритма тригонометрического шифра:
1) Использование функции с большим периодом, так как период влияет на количество переборов вариантов ключа с нужной точностью. (Если период функции $k\pi$, то количество вариантов ключа пропорционально k^2. Например: функция с периодом 8π будет иметь в 16 раз больше вариантов ключа, чем обычная функция $y = \cos(x + \Delta x)$) Однако не достаточно просто получить функцию с большим периодом – важное значение имеют «биения» функции. Математическая задача состоит в том, чтобы функции имели как можно более широкий разброс в спектре частот, содержащихся в функции.

2) Введение третьего параметра ключа. Данное улучшение позволит перейти от плоскости, на осях которой расположены параметры ключа, к объему. Теперь для того, что бы найти тройку параметров с точностью 10^{-5}, потребуется уже не 10^{10}, а 10^{15} переборов. Учитывая, что и период функции теперь будет возводиться в третью, а не во вторую степень, можем предполагать, что данное улучшение позволит тригонометрическому алгоритму быть неуязвимым и для генетического алгоритма за приемлемое время.

6. Список литературы

1. Сизов В.П.

Криптографические алгоритмы на основе тригонометрических функций. URL: http://www.ruscrypto.ru/sources/conjBrsnce/rc2005/

2. А. Ю. Городилов, А. А. Митраков

Криптоанализ тригонометрического шифра с помощью генетического алгоритма, Вестник Пермского Университета 2011, Вып. 4(8)