УДК 004.021

Л. П. Каминский, В. А. Степанов
Сибирский федеральный университет, г. Красноярск, Россия

ТРИГОНОМЕТРИЧЕСКАЯ КРИПТОГРАФИЯ

Отмечаются основные моменты работы тригонометрического алгоритма шифрования. Приводятся математические уязвимости алгоритма, а также идеи улучшения имеющейся модели. Рассказывается о генетическом алгоритме (алгоритме оптимизации) и его применении для тестирования улучшенных моделей тригонометрического алгоритма шифрования.

Ключевые слова: тригонометрический алгоритм шифрования, генетический алгоритм.

L. P. Kaminskii, V. A. Stepanov
Siberian Federal University, Krasnoyarsk, Russia

TRIGONOMETRIC CRYPTOGRAPHY

The main points of a trigonometric encryption algorithm highlighted. Mathematical algorithm vulnerability as well as ideas to improve the existing model are provided. The genetic algorithm (optimization algorithm) and using it to test improved models of trigonometric encryption algorithm is described.

Keywords: trigonometric encryption algorithm, genetic algorithm.

1. Описание работы тригонометрического шифра

Уравнения волны $y = \cos(x + N\cdot\Delta x)$ – пример одной из многих функций, имеющих постоянную амплитуду и непрерывных на всем промежутке $x \in (-\infty, +\infty)$. Важным моментом является то, что если для $y = \cos(x + \Delta x)$ параметр Δx не равен $2\pi/N$, 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 в компьютере используется 256 символов. По оси Y расставляем те же самые символы в любом (таком же или другом) порядке. Им также присваиваются порядковые номера от 1 до 256. Функция, посимвольно переводящая исходный текст в шифротекст:

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

где X – порядковый номер того символа, который нужно зашифровать; Z, Δx – любые числа, являющиеся секретными параметрами нашего ключа. Остальные параметры не являются секретными. Z, $\Delta x \in (-\infty, +\infty)$; 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 дешифровки

Тригонометрический шифр является примером симметричного алгоритма шифрования, следовательно:

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

2. Проблематика. Актуальность. Цель

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

В 2011 г. вышла статья [2], в которой говорится, 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 этом направлении нет. Следовательно, появляется интерес сначала создать сильный генетический алгоритм, а затем опробовать улучшенные версии тригонометрического шифра на надежность. Тем самым получим ответ на пригодность и конкурентоспособность данного шифра.

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:

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

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

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:

$$0 < Z < 2\pi,\ 0 <\cdot \Delta x < 2\pi.$$

Из статьи следует, что для получения текста, близкого к исходному, в качестве решения можно рассматривать не точку (пару секретных параметров), а некоторую ее окрестность [2]. 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 $m = 500$ символов эти величины имеют порядок 10^{-4} и 10^{-6} соответственно.

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

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

Итак, проблема с конечностью пространства решена. На прямоугольнике

$$0 < Z < 2\pi,\ 0 <\cdot \Delta x < 2\pi.$$

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

$$(2\pi\cdot 10^5)^2 \approx 4\cdot 10^{11}.$$

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 улучшение алгоритма шифрования с помощью генетического алгоритма, нужно сначала создать хороший генетический алгоритм и достичь результатов статьи [2].

Приведем пошаговое описание работы алгоритма.

1. Формируется начальная популяция. Количество особей N 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 $\log_2 100\ 000 \approx 16{,}684$, то для хранения 5 десятичных разрядов (именно данную точность, согласно главе «математические уязвимости», мы считаем достаточной) потребуется 17 двоичных. Вывод – особь есть упорядоченная последовательность 34 бит, хранящая дроб-

ные части ключа. В нашем случае каждая начальная особь задается псевдослучайно.)

2. Создается вектор значений фитнесс-функции (функция, позволяющая оценить пригодность популяции к решению задачи). Далее ко всей популяции применяется данная функция. В нашем случае фитнесс-функция «расшифровывает» текст для каждой особи в популяции и считает среднее значение суммы вероятностей встречи пары подряд идущих символов в полученном тексте, согласно данным алфавита. Затем происходит нормализация значений сумм вероятностей в пределах минимального и максимального значений и этими значениями заполняется вектор. Таким образом, каждой особи ставится в соответствие число, показывающее ее приспособленность.

3. Выбирается число M – количество поколений. Параметр вновь задается пользователем.

4. Далее выполняются M поколений, что заключается в N раз создании новой популяции из текущей, которая получится выбором особым образом двух особей, кроссинговере этих особей и последующей мутации полученной популяции. Далее полученная популяция становится «рабочей» и начинается следующее поколение.

Теперь подробнее про генетические операторы [3]:

1) Селекция – оператор случайного выбора одного индивида из популяции, основанный на значениях функции пригодности всех индивидов текущей популяции, для использования его в операторе скрещивания. При этом вероятность выбора у индивидов с более высокой пригодностью выше, чем у индивидов с более низкой пригодностью. Селекция бывает трех видов: пропорциональная, ранговая и турнирная. Пропорциональная селекция заключается в выборе особи пропорционально значению пригодности особи. Ранговая селекция работает не с массивом пригодностей напрямую, а массивом нормированных рангов, присваиваемых индивидам на основе значений пригодности. Далее для выбора индивидов используется пропорциональная селекция, работающая с массивом рангов. При турнирной селекции из популяции с равной вероятностью выбираются индивиды в количестве T (размер турнира), где $2 \le T \le N$. При этом каждый индивид может попасть в группу (турнир) только один раз (турнирная селекция без возвращения). Из данной группы выбирается индивид с наибольшей пригодностью.

2) Кроссинговер – оператор случайного формирования нового индивида из двух выбранных родителей с сохранением признаков обоих родителей. Нами были рассмотрены три вида кроссинговера: одноточечный, двухточечный и равномерный. Пусть у нас имеются два выбранных селекцией родителя. Тогда в одноточечном кроссинговере в случайном месте происходит разрыв между двумя позициями генов в обоих родителях. После этого происходит обмен частями, в результате чего образуются два потомка. Из них выбирается случайно один потомок, который и передается в качестве результата оператора скрещивания. Двухточечный кроссинговер представляет собой тот же одноточечный кроссинговер, только не с одним разрывом между позициями генов, а двумя. Равномерный кроссинговер – когда потомок состоит из генов, каждый из которых выбран случайно из генов родителей на соответствующих позициях.

3) Мутация – оператор случайного изменения всех потомков из популяции. Цель данного оператора не получить лучшее решение, а разнообразить многообразие рассматриваемых индивидов. Обычно мутация предполагает незначительное изменение потомков. При выполнении оператора каждый ген каждого индивида с некоторой заданной вероятностью мутирует, то есть меняет свое значение на противоположное. Обычно вероятность мутации задается исходя из трех вариантов: слабая (вероятность мутирования равна $1/(3 \cdot N)$), средняя (вероятность равна $1/N$) или сильная мутация (вероятность равна $3/N$).

5. Если количество поколений не превышает M, то эволюция продолжается (возвращаемся к шагу 3).

6. Решением объявляется особь, имеющая лучшую приспособленность за все время работы алгоритма.

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

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

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

В данный момент созданный нами алгоритм столкнулся со специфической проблемой. Так как создание генетического алгоритма – это обычно решение задачи оптимизации по нахождению минимума или максимума определенной функции (в нашем случае это фитнесс-функция), то алгоритм действует именно так и находит максимум фитнесс-функции. Максимум зачастую не совпадает с решением нашей задачи расшифровки, так как всегда можно взять таблицу вероятностей биграмм и составить специальным образом набор символов заданной длины, имеющей наибольшую вероятность встречи по таблице. Реализация алгоритма находит как раз такой текст, и наша задача – улучшить работу алгоритма, которая заключается в обучении алгоритма способности отличать осмысленный набор символов от неосмысленного.

5. Способы улучшения

На данный момент есть два основных способа улучшения 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, чтобы найти тройку параметров с точностью 10^{-5}, потребуется уже не 10^{10}, а 10^{15} переборов. Учитывая, что и период функции 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.

Благодарность

Авторы выражают благодарность научному руководителю доктору физико-математических наук Кытманову Алексею Александровичу за помощь в подготовке научной работы.

Библиографические ссылки

1. Сизов В. П. Криптографические алгоритмы на основе тригонометрических функций. URL: http://www.ruscrypto.ru/sources/conjBrsnce/rc2005/
2. Городилов А. Ю., Митраков А. А. Криптоанализ тригонометрического шифра с помощью генетического алгоритма // Вестник Пермского университета, 2011. Вып. 4(8).
3. Сергиенко А. Б. Генетический алгоритм. Стандарт. v.3.6. URL: https://github.com/Harrix/Standard-Genetic-Algorithm

Статья поступила в редакцию 19.11.2013 г.