БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Гладков Л.А., Курейчик В.М, Курейчик В.В.* Генетические алгоритмы. − Ростов-на-Дону: Ростиздат, 2004.
2. *Ахо А., Хопкрофт Дж., Ульман Д.* Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979.
3. *Гладков Л.А.* Генетические операторы. − Таганрог Изд-во: ТРТУ, 2005
4. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1985.

УДК 687.06

Х.А. Кажаров

РАЗРАБОТКА ГЕНЕТИЧЕСКОЙ МОДЕЛИ ПОИСКА ПРОСТЫХ ЧИСЕЛ ДЛЯ КРИПТОАНАЛИЗА RSA НА ОСНОВЕ КЛИЕНТ-СЕРВЕРНОЙ СТРУКТУРЫ

Введение. Современные протоколы передачи данных, такие как SSL, TLS, PGP, обеспечивающие защиту передаваемой информации, используют криптосистему RSA для установления защищенного соединения и распределения ключей шифрования. Надежность криптосистемы RSA обуславливает защищенность передачи данных.

В современном мире наблюдается тенденция перехода от использования классических детерминированных алгоритмов к интеллектуальным поисковым системам, системам автоматизированного управления и проектирования. Основными направлениями в разработках интеллектуальных систем в настоящее время являются эволюционные вероятностно-направленные методы. Ярким примером таких методов являются генетические алгоритмы (ГА), ориентированные на поиск глобального оптимума в пространстве возможных решений с многоэкстремальным ландшафтом.

Сегодня генетические алгоритмы успешно применяются для решения классических NP-полных задач, задач оптимизации в пространствах с большим количеством измерений, ряда экономических задач.

Постановка задачи. Все асимметричные криптосистемы характеризуются тем, что используют два ключа: один для шифрования, другой для дешифрования, которые имеют определенную зависимость. В асимметричных криптосистемах наибольшее распространение получила RSA. Функция, реализующая схему RSA, представлена формулой (1):

$$f : x \rightarrow x^e \ (mod\ m). \tag{1}$$

Для дешифрования сообщения $a=f(x)$ достаточно решить уравнение (2):

$$x^e \equiv a \ mod\ m. \tag{2}$$

При некоторых условиях, налагаемых на m и e, это сравнение имеет единственное решение x [1]. Если показатель степени e в сравнении с (2) взаимно прост с $\varphi(m)$ (функция Эйлера), то сравнение (2) имеет единственное решение. Чтобы его найти, необходимо определить число d, удовлетворяющее условию (3):

$$\begin{aligned} &de \equiv 1(\mathrm{mod}(\varphi(\mu))), \\ &1 \le \delta < \varphi(\mu). \end{aligned} \tag{3}$$

Решением задачи будет следующее уравнение:

$$a^d \equiv x^{de} \equiv x \bmod m.$$

При этом *(e, m)* – открытый ключ, который передается по незащищенным каналам, а *(d, m)* – секретный ключ. Для вычисления секретного ключа *d* необходимо вычислить функцию Эйлера для *m*. Функцией Эйлера *φ(m)* называется количество положительных целых чисел, меньших *m* и взаимно простых с *m*:

$$\varphi = (p-1)(q-1). \quad (4)$$

При этом если $m = pq$, где p и q – простые числа, то функция Эйлера вычисляется с помощью (4) [2].

Таким образом, для криптоанализа системы RSA необходимо знать разложение модуля *m*. Самые лучшие алгоритмы факторизации (разложения) имеют субэкспоненциальную сложность:

$$L_m[\gamma, c] = e^{(c+o(1))(\ln m)^{\gamma}(\ln\ln m)^{1-\gamma}},$$

где c и γ – постоянные, $o(1) \to 0$, $x \to +\infty$. В [3] подробно описываются алгоритмы разложения составного числа на простые множители.

Таким образом, для применения ГА при криптоанализе RSA, с учетом его специфики, необходимо разработать модифицированный ГА для нахождения простых множителей заданного составного числа.

Основные положения генетических алгоритмов. Основная идея заключается в комплексной адаптации многочисленной группы организмов – популяции к изменяющимся внешним условиям. Адаптация организмов зависит от разнообразия особей популяции. Популяция тождественных особей обладает очень низкой адаптивностью. Популяция индивидуальных особей – индивидов, напротив, способна адаптироваться к любым внешним условиям. Таким образом, эволюция, необходимая организмам для их выживания в среде обитания, представляет собой процесс оптимизации органических систем.

Метод оптимизации технических систем, в котором реализована эта идея, получил название «Генетические алгоритмы». Генетические алгоритмы позволяют получать хорошие результаты при решении NP – полных задач [4].

Структура хромосом. Структура хромосом представляет собой битовый стринг, который хранит информацию о простом множителе. Второй множитель находим из равенства $m = pq$.

В данной работе используются бинарные хромосомы. Бинарная хромосома является гомологичной числовой хромосомой, каждый ген которой может принимать целые значения в интервале [0, 1].

Приведем пример кодировки хромосомы для $x \in N$, $0 \le x \le 1023$. В этом случае одно число кодируется бинарной хромосомой из 10 ген, как показано на рис. 1.

0	0	1	0	1	1	0	0	1	1

Рис. 1. Кодировка хромосомы

При декодировке рассматриваем хромосому как бинарное число. Хромосому просматриваем слева направо, поэтому разряды увеличиваются в том же порядке. Далее применяем простой алгоритм перевода из двоичной системы счисления в десятичную:

$$G = A_0*2^0+A_1*2^1+A_2*2^2+\ldots+A_n*2^n = 820.$$

где A_i – значение бита в i-м разряде.

Формирование исходной популяции. В этом блоке возможно использование различных стратегий формирования исходной популяции:

а) одеяло – покрываем равномерно пространство поиска n решениями, где n – размерность популяции;

б) дробовик – случайный выбор n потенциальных решений;

в) фокусировка – все решения концентрируются в относительно узком диапазоне чисел.

Простое число генерируется случайным выбором значения битов. Последний бит всегда устанавливается равным 1. Затем вычисляем среднее расстояние между простыми числами r, после чего в диапазоне $[G-r;G+r]$ осуществляется поиск наиболее вероятного простого числа.

В заданном диапазоне ведем поиск наиболее вероятного простого числа следующим образом. Для заданного диапазона строим решето Эратосфена. Каждое число $x \in [G-r;G+r]$ последовательно проверяем на делимость с простыми числами в диапазоне $[3; 2r]$. В данном случае простые числа в этом диапазоне необходимо вычислить заранее, причем любым методом. Поскольку диапазон в худшем случае даже для чисел порядка 2^{1024} составляет около семисот, то выведение простых чисел в диапазоне $[3; 2r]$ не будет влиять на временную сложность алгоритма, т.е. им можно пренебречь. Такой быстрый алгоритм позволит значительно сузить пространство поиска решений:

$$r=ln(G). \tag{5}$$

Пусть G порядка 2^{512}, тогда $r \approx 355$. Поскольку вычисления производятся итерационно, то для ускорения вычислений преобразуем формулу (5) следующим образом:

$$r = \frac{\log_2 G}{\log_2 e} \approx \frac{\log_2 G}{1{,}442695}. \tag{6}$$

Зная число бит n выражение (6) принимает следующий вид:

$$r = \frac{n}{1{,}442695}. \tag{7}$$

После сужения пространства поиска решений к оставшемуся множеству чисел применяем тест Миллера-Рабина [4]. В случае, когда остается несколько чисел, удовлетворяющих тесту, выбор наиболее вероятного (одного или нескольких чисел) возлагается на ЛПР (лицо, принимающее решение).

Приведем пример работы данной схемы. Пусть n = 6, а сформированная хромосома имеет следующий вид:

1	0	1	0	1	1

Рис. 2. Пример хромосомы

Отсюда видно, что G = 43. Вычислим радиус:

$$r = \frac{6}{1{,}442695} = 4{,}15888 \approx 5.$$

Отметим, что в данном случае округление делается в большую сторону.

Рассматриваем следующий диапазон чисел [38; 48]. Далее проверяем делимость этих чисел на простые числа в диапазоне [2; 10], т.е. {2, 3, 5, 7}. Согласно методу реализации решета Эратосфена исключаем из диапазона [38;48] числа, де-

лящиеся на {2, 3, 5, 7}. В итоге получаем следующие простые числа {41, 43, 47}. Очевидно, что все три числа пройдут вероятностный тест Миллера-Рабина, а выбор остается за ЛПР.

Приведем еще один пример, показывающий необходимость применения вероятностного теста Миллера-Рабина. Примем $G = 121$. Тогда $r \approx 5$. Получаем следующий диапазон [118; 128], в котором исключаются числа, делящиеся на {2,3,5,7}. В итоге получаем следующие числа {121,127}. После применения к ним теста Миллера-Рабина останется число 127, а число 121 будет исключено, так как 121 = 11*11.

Расчет целевой функции. Декодировка хромосомы дает значение первого потенциально простого множителя *p*, для которого есть лишь один однозначный сомножитель:

$$q = \frac{m}{p}.$$

Далее, к полученному результату применяем вероятностный тест Миллера-Рабина с целью получения информации о вероятности простоты числа q. Поскольку каждая хромосома получается в результате локального поиска с применением вышеуказанного вероятностного теста, то можно судить о вероятности простоты каждого сомножителя. Таким образом, значение целевой функции (ЦФ) определяется произведением вероятностей двух сомножителей:

$$P = P(q)*P(p).$$

Выбор родительской пары хромосом. Как правило, здесь используется три метода выбора родительской пары хромосом: а) случайный; б) элитный; в) «колесо рулетки» [5].

Экспериментальные исследования показали, что наиболее эффективным является случайный выбор. Это обусловлено спецификой задачи. Поскольку вероятность того, что $P(q)=0$ и, соответственно, ЦФ в целом для многих хромосом в популяции будет равна 0, высока, то элитный выбор и «колесо рулетки» будут способствовать локализации пространства поиска, а ГА войдет в состояние стагнации.

Кроссинговер. В данной работе применялись модифицированные простые операторы кроссинговера: а) одноточечный; б) двухточечный; в) многоточечный [5].

Модификация заключалась в следующем. Работа оператора разбивается на два этапа. Первый этап отражает работу стандартного оператора кроссинговера. Второй этап заключается в локальном поиске наиболее вероятного простого числа в окрестности декодированного числа Y. Радиус окрестности определяется формулой (5), и, следовательно, пространство поиска соответствует интервалу $[G-r; G+r]$. Соотношение (5) подтверждено экспериментальными исследованиями.

Рассмотрим результат работы программы для диапазона [1; 1024]. Находим экстремум функции распределения простых чисел в заданном диапазоне. В диапазоне [1; 1024] он равен 20, для [1; 10240] получили значение 36, для [1; 102400] – 72. Другими словами, получили максимальное расстояние между простыми числами в заданном диапазоне.

На рис. 2 показаны два графика: зависимость расстояния между двумя простыми числами от размера числа и среднее значение. Из графиков видно, что среднее расстояние между простыми числами в зависимости от размерности распределено по логарифмическому закону. Это подтверждается законом Валле-Пуссена [4].

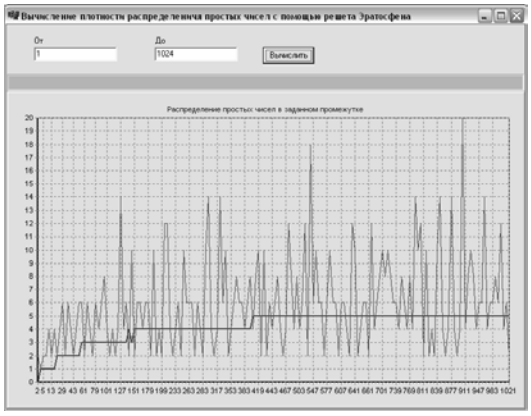

Рис. 2. Вычисление плотности распределения простых чисел

Найденное число и будет являться потомком.

Мутация. В качестве оператора мутации используются обмен и инверсия, к которым также применяется локальный поиск наиболее вероятного простого числа. Отметим, что для определения простоты числа также используем алгоритм вероятностного теста Миллера-Рабина, благодаря которому становится известным вероятность простоты.

Оператором отбора могут служить стандартные операторы: а) случайный; б) элитный; в) «колесо рулетки» [5].

Применение экспертной системы. Как известно, для криптоанализа RSA применяются распределенные вычисления. Такой подход реализуем и для ГА на основе метаэволюции. Эта форма эволюции характеризуется существованием нескольких популяций, развивающихся независимо друг от друга. Как правило, такой ГА очень удобен для реализации экспертной системы (ЭС). ЭС предназначается для выявления параметров ГА, при которых ЦФ быстрее всего достигает своего оптимума. В данном же случае каждая популяция может развиваться с различными параметрами (вероятности кроссинговера и мутации, вариации генетических операторов (ГО), некоторые количественные характеристики). Таким образом, ЭС на основе анализа информации об индивидах из различных популяций выделяет «лучшие» параметры ГА. При этом ЭС может также влиять на параметры ГА, которые являются динамическими.

На рис. 3 показан пример взаимодействия 8 популяций с ЭС.

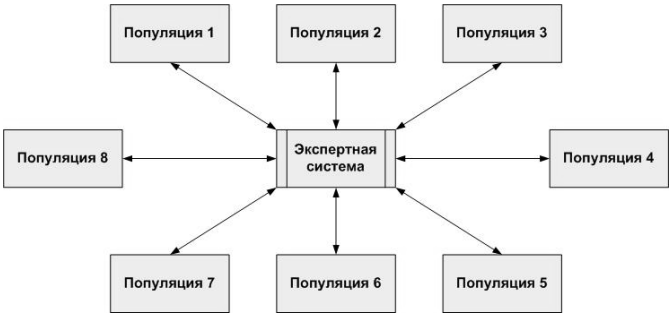

Рис. 3. Модель взаимодействия ЭС с 8 популяциями

Связь между популяциями и блоком экспертной системы двусторонняя (см. рис. 3). Каждая популяция посылает в блок ЭС информацию о лучших индивидах (прямая связь) и получает новых индивидов (обратная связь). Также на основе анализа полученных решений ЭС возвращает каждой популяции новые параметры, которые в дальнейшем определят их адаптируемость.

Структура алгоритма. ГА на основе метаэволюции имеет клиент-серверную структуру (см. рис. 3). Так как каждый клиент представляет одну популяцию со своими специфическими индивидами, то экспертная система необходима, чтобы периодически производить обмен видов между различными популяциями, как показано на рис. 4, внося, тем самым, разнообразие. Изначально в серверной части комплекса нет информации о клиентах. Клиентские модули присоединяются по сети к серверу и через него обогащают свой генофонд. Экспертная система периодически получает информацию о лучших результатах, что позволяет отслеживать работу системы.

Рис. 4. Работа сервера при взаимодействии с клиентом

При взаимодействии клиент посылает определенное количество случайно выбранных решений из всей популяции (см. рис. 4). Экспертная система объединяет всех индивидов, полученных от клиентов, затем каждое решение мигрирует в случайно выбранную популяцию. Представленная организация многопопуляционного генетического алгоритма может иметь произвольное количество клиентских модулей, которое задается ЛПР.

Параметры интеллектуальной системы. Представленная интеллектуальная система характеризуется следующими параметрами: а) количество популяций; б) размер популяций; в) количество индивидов для выборки при миграции.

Каждый из этих параметров задается ЛПР. Важной частью представленной интеллектуальной системы является работа ЭС. Логика работы экспертной системы отражена в следующей формуле:

$$P_i = P_i + \Delta P, i \in N, i \le n, P_i \ne P_b,$$

где $P_i=\{p_{cross}, p_{mut}, o_{cross}, o_{mut}, o_r, o_s\}$ – кортеж параметров i-й популяции; p_{cross} и p_{mut} – вероятности, соответственно, кроссинговера и мутации; o_{cross} и o_{mut} – операторы, соответственно, кроссинговера и мутации; o_r – оператор редукции; o_s – оператор селекции.

Заключение. Результатом данной работы является описание основных этапов генетического алгоритма для решения задачи криптоанализа RSA. В работе проведен анализ криптосистемы RSA, рассмотрены различные подходы к криптоанализу. В основе идеи предложенного метода лежит разложение составного числа на два простых методами генетического поиска. Разработана структурная схема генетического алгоритма для параллельного криптоанализа RSA. Определен метод формирования исходной популяции, в котором разработан алгоритм быстрой проверки на простоту, разработана целевая функция, позволяющая определять адаптируемость отдельных видов популяции. Определен метод выбора родительской пары хромосом в генетическом алгоритме, а также разработан модифицированный оператор кроссинговера, учитывающий специфику задачи.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Ященко В.В.* Введение в криптографию. – М.: МЦНМО, 2003.
2. *Баричев С.*, Криптография без секретов [Электронный ресурс] – http://forum.armkb.com.
3. *Василенко О.Н.* Теоретико-числовые алгоритмы в криптографии. – М.: МЦНМО, 2003.
4. *МакКоннелл Дж.* Основы современных алгоритмов. – М.: Техносфера, 2004.
5. *Гладков Л.А., Курейчик В.М., Курейчик В.В.* Генетические алгоритмы. – Ростов-на-Дону: ООО «Ростиздат», 2004.

УДК 681.31

П.В. Афонин, О.В. Кокшагина

ГИБРИДНЫЕ ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ ДЛЯ ЗАДАЧИ СОСТАВЛЕНИЯ РАСПИСАНИЯ ПРОЕКТА*

Введение. Интеграция и гибридизация различных методов и компьютерных технологий позволяет решать сложные задачи, которые трудно или нельзя решить на основе каких-либо отдельных методов и технологий. В гибридной системе (ГС), объединяющей несколько методов или подходов, эффективность одного подхода может компенсировать слабость другого. Комбинируя различные подходы, можно обойти недостатки, присущие каждому из них в отдельности.

Классификация гибридных систем по уровню интеграции приведена в [1]. Согласно этой классификации выделяются автономные, трансформационные, слабосвязанные, сильно связанные и полностью интегрированные системы. Идея гибридных генетических алгоритмов (hybrid genetic algorithms) заключается в сочетании генетического алгоритма (ГА) с некоторым другим методом поиска [2].

В статье рассматривается решение задачи составления расписания проекта с помощью гибридных генетических алгоритмов. Предложены гибридные схемы с использованием улучшающих эвристических алгоритмов, гибридная схема формирования начальной популяции, модифицированные операторы скрещивания и мутации для случая кодирования строки-хромосомы как последовательности работ в расписании. Основное внимание уделено построению гибридного алгоритма на основе трансформационной схемы. Программная реализация алгоритмов и проведение экспериментальных исследований осуществлялось в программной среде MatLab 7.1.

Постановка задачи. Рассмотрим следующую постановку задачи: проект представляет собой совокупность работ и ресурсов. Ресурс характеризуется нормальным или сверхурочным потреблением в день (в единицах ресурса), стоимостью единицы ресурса при нормальном потреблении и сверхурочном потреблении соответственно. Работа характеризуется длительностью, списком назначенных ресурсов, списком предшествующих и последующих работ, задержками между работами (например, в случае закладки фундамента).

Необходимо составить расписание работ проекта. Под расписанием понимается функция, которая каждому ресурсу l и моменту времени t сопоставляет работу, обслуживаемую ресурсом l в момент времени t, либо указывает, что прибор l в момент времени t ресурс простаивает.

* Работа выполнена при поддержке РФФИ (гранты № 08-07-00337, 08-07-00343).